UNIVERSITÄT LEIPZIG

**Institut für Produktionswirtschaft
und Industrielle Informationswirtschaft**

Marschnerstraße 31, 04109 Leipzig
Tel.: 0341/4941-182, Fax: -125

Arbeitsbericht Nr. 8

Petrinetzbasierte Modellierung komplexer Produktionssysteme

Band 4: Verfeinerungen von Stelle/Transition-Netzen

von
Univ.-Prof. Dr. Stephan Zelewski

<zelewski @ hpswifa.wifa.uni-leipzig.de>

Leipzig 1995

Inhaltsverzeichnis zu Band 4

4 Verfeinerungen des Konzepts der Stelle/Transition-Netze

4.1 Algebraische Beiträge zur Konzeptverfeinerung

4.1.1 Überblick über das algebraische Signaturkonzept

Basiskonzept für die Definition strukturierter Marken mit der Hilfe eines algebraischen Kalküls ist das Signaturkonzept[1)]. Es wird später zum Konzept signaturbezogener Algebren[2)] mit zugehörigen Termauswertungen erweitert. Algebraische Spezifikationen und prädikatenlogische Dynamiken runden das Signaturkonzept ab. Das algebraisch ausgerichtete - und später prädikatenlogisch angereicherte[3)] - Signaturkonzept wird in dieser Arbeit vorrangig dazu benutzt[4)], die strukturlosen Marken aus Stelle/Transition-Netzen durch interne Strukturierung zu den sortierten Marken von Synthetischen Netzen zu erweitern[5)].

Mit der Hilfe des Signaturkonzepts wird zunächst die Syntax einer formalen Sprache definiert. Sie dient dazu, Stelle/Transition-Netze mit formalsprachlichen Anschriften zu versehen. Die Netzbeschriftungen konstituieren eine formale Semantik der vormals rein syntaktisch eingeführten Stelle/Transition-Netze. Die Komplexe aus zugrundeliegenden Stelle/Transition-Netzen und ihren Beschriftungen gehören im Prinzip[6)] zur Klasse der Prädikat/Transition-Netze. Die derart spezifizierten Netze vereinigen mehrere Basiskonzepte in sich. Als Stelle/Transition-Netze gehören sie zur Arithmetik und Graphentheorie. Als Prädikat/Transition-Netze erschließen sie die Ausdruckskraft der Prädikatenlogik. Infolge ihres Signaturbezugs überlagern sie der konventionellen Prädikatenlogik eine verfeinernde algebraische Sortenstruktur. Zugleich schlagen die algebraischen Signaturen eine Brücke zum Konzept der abstrakten Datentypen[7)]. Diese Datentypen verweisen ihrerseits auf den objektorientierten Gestaltungsansatz[8)] und auf den Bereich konzeptueller Datenschemata[9)]. Aufgrund dieser Vielfalt inhärenter Konzepte werden die hier entwickelten Netze unter den Begriff der Synthetischen Netze subsumiert.

Die arithmetischen und graphentheoretischen Grundlagen von Stelle/Transition-Netzen wurden bereits früher dargelegt. Daher konzentrieren sich die nachfolgenden Ausführungen auf das algebraische Signaturkonzept und seine prädikatenlogischen Erweiterungen. Der Bezug auf abstrakte Datentypen[10)] und konzeptuelle Datenschemata[11)] wird dagegen nicht weiter vertieft.

Die Kombination von Prädikat/Transition-Netzen mit algebraischen Signaturen findet - gemessen an der reichhaltigen Literatur zu Prädikat/Transition-Netzen - derzeit noch selten statt[12)]. Trotzdem präferiert der Verf. diesen Ansatz aus drei Gründen.

Erstens stellt das Signaturkonzept seiner Einschätzung nach das zur Zeit präziseste formalsprachliche Instrument dar[13)], mit dessen Hilfe sich die formale Semantik von Prädikat/Transition-Netzen definieren läßt[14)].

Zweitens eröffnet das oben skizzierte Konzeptamalgam die Perspektive, aus der Verknüpfung sehr unterschiedlicher Konzepte befruchtende Impulse für die Gestaltung oder Anwendung Synthetischer Netze zu gewinnen[15)].

Drittens vermutet der Verf. ein beträchtliches - sowohl theoretisches als auch praktisches - Entwicklungspotential für die Kombination von Petrinetzen mit algebraischen Signaturen. Als Indiz hierfür wertet er, daß vor kurzem ein Forschungsprogramm der Europäischen Gemeinschaften initiiert wurde, das auf eben dieser Vereinigung zwischen Netz- und Signaturkonzept beruht. Es handelt sich um das ESPRIT-Projekt[16)] Nr. 125 "GRASPIN"[17)]. Hierbei soll eine Entwicklungsumgebung für die Problemanalyse, Spezifizierung und inkrementelle Programmierung von komplexen, verteilten und nebenläufig operierenden Softwaresystemen erarbeitet werden. Die Systemspezifizierung erfolgt in der semi-graphischen Sprache SEGRAS[18)]. Sie wurde speziell dafür ausgelegt, Petrinetze vom Typ der Prädikat/Transition-Netze[19)] mit dem algebraischen

Signaturkonzept für die Spezifikation abstrakter Datentypen zu kombinieren[20]. Damit werden frühere Ansätze, Spezifikationssprachen für Softwaresysteme auf der Basis von Signaturen und signaturbezogenen Algebren zu konzipieren[21], in Kombination mit dem Petrinetz-Konzept vertieft fortgeführt.

Anmerkungen zum Kapitel:

1) Die nachfolgenden Ausführungen beruhen auf den Darstellungen von Signaturen und signaturbezogenen Algebren (many-sorted algebras) bei EHRIG (1984a), S. 10ff.; EHRIG (1984b), S. 4ff., insbesondere S. 13ff.; EHRIG (1985a), S. 9ff., insbesondere S. 14ff.; VAUTHERIN (1987b), S. 294ff.; EHRIG (1987), S. 4f.; KRÄMER (1987a), S. 270f.; BEIERLE (1988a), S. 447ff.; RECK (1988), S. 39f. u. 42ff.; ASTESIANO (1988), S. 140ff., insbesondere S. 142 u. 147ff. (mit besonderer Berücksichtigung von prädikatenlogischen Erweiterungen); BATTISTON (1988), S. 22 u. 24ff.; EHRICH (1988), S. 123ff.; KIRCHNER (1988), S. 289f.; REISIG (1989a), S. 5ff.

2) Der Begriff einer Algebra wird in der einschlägigen Literatur unterschiedlich definiert. Mitunter wird z.B. unter "der" Algebra derjenige Kalkül verstanden, der aus allen reellen Zahlen zusammen mit einer Klasse ausgezeichneter Funktionen - den "algebraischen" Funktionen - über diesen Zahlen besteht. Als solche Funktionen kommen vor allem Polynome mit ganzzahligen Koeffizienten in Betracht. Dann sind die algebraischen Zahlen als die Nullstellen dieser Polynome definiert. Vgl. zu diesem eng gefaßten, an Polynomen ausgerichteten Algebraverständnis z.B. LORENZEN,P. (1962), S. 153; MESCHKOWSKI (1967), S. 28.
Im folgenden wird ein abstrakterer, nicht auf das reellzahlige "Rechnen" beschränkte Begriff für Algebren eingeführt. In einer ersten groben Annäherung lassen sich diese Algebren als Kalküle charakterisieren, in denen beliebige formale Operationen über beliebigen formalen Objekten erklärt werden. "Die" o.a. Algebra ist dann nur derjenige Sonderfall, in dem die Operationen als "algebraische" Funktionen und die formalen Objekte als reelle Zahlen definiert sind.
Eine Präzisierung dieser informalen Annäherung findet sich bei RICHTER,M.M. (1978), S. 16; RAUTENBERG (1979), S. 319: Dort wird eine Algebra A als ein 2-Tupel $A=(OB_A,OP_A)$ erklärt. Ihre erste Komponente OB_A ist eine beliebige, nicht-leere Menge formaler Objekte. Die zweite Komponente stellt eine (nicht-leere) Menge von K_j-stelligen Operationssymbolen Op_j mit $K_j \in \mathcal{N}_0$ mit $j \in \{1,...,J\}$ und $J \in \mathcal{N}_+$ dar. Zu jedem Operationssymbol gehört eine Operation, die über den formalen Objekten der Menge OB definiert ist. Weiterführende inhaltliche Festlegungen der Operationsvorschriften erfolgen nicht. Diese abstrakte Algebradefinition bildet auch die Grundlage der später eingeführten SIG-Algebren. Diese SIG-Algebren unterscheiden sich von RICHTER's und RAUTENBERG's Definition im wesentlichen dadurch, daß sie ihr noch die Sortenstruktur einer Signatur überlagern. Darüber hinaus verwendet das Signaturkonzept einen vollkommen anderen Signaturbegriff als RICHTER. Ähnliche Ansätze, eine Algebra als ein Paar aus einer Objekt- und einer Operationenmenge aufzufassen, finden sich auch bei URSPRUNG (1982), S. 28.
Einen anderen abstrakten, hier nicht weiter verfolgten Algebrabegriff vertreten EHRIG und MAHR in EHRIG (1985a), S. 9. Sie fassen Algebren als mathematische Strukturen auf, die durch formale Operationen und Gleichungen konstituiert werden. Aus der Sicht des Verf. irritiert an diesem Definitionsvorschlag einerseits, daß die formalen Objekte, auf denen die Operationen ausgeführt werden können, überhaupt nicht berücksichtigt werden. Andererseits erscheint ihm die Beschränkung auf Gleichungen für die Entfaltung Synthetischer Netze als zu eng, in denen auch Ungleichungen und andere Relationen eine Rolle spielen werden. Der Gleichungsbezug von EHRIG und MAHR rührt dagegen aus deren spezieller Beschäftigung mit der algebraischen Spezifikation von abstrakten Datentypen und Softwaresystemen her, bei der es tatsächlich ausreicht, ausschließlich Gleichungssysteme zu betrachten. Allerdings wird diese enge Sichtweise in EHRIG (1985a), S. 25f., zugegeben ("equational specification" anstatt "algebraic specification"). Zugleich erfolgt dort (S. 25f.) eine Perspektivenerweiterung auf alle Spezifikationen, die mit beliebigen Formeln aus der Prädikatenlogik 1. Stufe für die Spezifikationsformulierung arbeiten.

3) Andeutungen auf Verknüpfungen des algebraischen Signaturkonzepts mit der Prädikatenlogik finden sich bereits in den algebraisch ausgerichteten Beiträgen zum Signaturkonzept; vgl. z.B. EHRIG (1985a), S. 19f.

4) Das Signaturkonzept wird überwiegend in das informationstechnische Konzept der algebraischen Spezifikation von Datenstrukturen und Softwaresystemen eingebettet. Eine kurze Einführung in den inhaltlichen und historischen Hintergrund des Signaturkonzepts findet sich bei EHRIG (1985a), S. 1ff. u. 8. Die informationstechnische Konzeptausrichtung wird hier jedoch an die speziellen Anforderungen des Petrinetz-Konzepts angepaßt.

5) Näheres zu sortierten Marken an späterer Stelle.

6) Allerdings verdeutlicht diese einschränkende Formulierung den Vorbehalt, daß die Freizügigkeit der Prädikat/Transition-Netze hier zugunsten der speziellen Behandlungsweise von Marken als explizit eingeführte Basiskonstrukte eingeschränkt wird. Vgl. dazu die Erläuterungen im voranstehenden Kapitel.

7) Vgl. zur engen Beziehung zwischen dem algebraischen Signaturkonzept und abstrakten Datentypen EHRIG (1984a), S. 2; EHRIG (1984b), S. 3; EHRIG (1985a), S. 1ff. u. 33f.; KRÄMER (1987a), S. 270f.; ASTESIANO (1988), S. 142 u. 147ff., insbesondere S. 150ff.; BEIERLE (1988a), S. 447ff., insbesondere S. 447 u. 452; RECK (1988), S. 39ff., 53f. u. 60ff.; EHRICH (1988), S. 119ff., insbesondere S. 123f.
Nähere Kennzeichnungen abstrakter Datentypen finden sich bei ZILLES (1974); EHRIG (1985a), S. 2f. u. 33ff., insbesondere S. 34; RECK (1988), S. 39ff.; BEIERLE (1988a), S. 447; BARR,R. (1989), S. 23f.

8) Ein abstrakter Datentyp läßt sich grob beschreiben als eine Kombination, die einen gewöhnlichen Datentyp und alle darauf anwendbare Operationen zusammenfaßt. Dies entspricht dem objektorientierten Ansatz, Objekte einer-

seits mit einer reichhaltigen inneren Struktur auszustatten. Andererseits wird ebenso die Erweiterung von Objekten um Operationen eingeschlossen, die auf diese Objekte angewendet werden dürfen. Vgl. zur engen Verwandtschaft zwischen objektorientierten Gestaltungsideen - insbesondere der objektorientierten Programmierung - und abstrakten Datentypen BARR,R. (1989), S. 23.

9) Vgl. dazu auch die Anmerkungen zur Beziehung zwischen strukturierten Marken und Datentypen.

10) Die Ausklammerung des Aspekts abstrakter Datentypen liegt darin begründet, daß sich die prädikatenlogische Implementierung Synthetischer Netze vermittels der Implementierungssprache Turbo-PROLOG an das Konzept der logischen Programmierung anlehnt. Logische Programmierung und abstrakte Datentypen verhalten sich aber komplementär zueinander. Dies wird von BEIERLE (1988a), S. 454, näher ausgeführt. Daher verdrängt die implementierungsbezogene Präferenz für das logische Programmieren die Perspektive abstrakter Datentypen.
Vgl. dagegen zur Verknüpfung zwischen Petrinetzen mit strukturierten Marken einerseits und abstrakten Datentypen andererseits SCHMIDT,H.W. (1984a), S. 330ff.; KRÄMER (1985a), S. 308ff.; KRÄMER (1987a), S. 269ff.; VAUTHERIN (1987b), S. 293ff.; FONIO (1987); BATTISTON (1988), S. 20ff.; REISIG (1989a), S. 43ff.

11) Allerdings ließe sich dieser Bezug auf konzeptuelle Datenschemata anhand des Programmpakets PASIPP vertiefen, das in dieser Arbeit der Implementierung von Synthetischen Netzen zugrundegelegt worden ist. Denn PASIPP ist neuerdings in die relationale Datenbankumgebung INGRES eingebettet. Eine Übersicht hierüber und die dabei verwendeten konzeptuellen Datenschemata findet sich bei OBERWEIS (1989a), S. 11ff.

12) Vgl. zu ähnlichen, aber im Detail - mitunter erheblich - divergierenden Ansätzen, Petrinetze mit dem algebraisch definierten Signaturkonzept zu kombinieren, SCHMIDT,H.W. (1984a), S. 332ff., insbesondere S. 335ff.; SCHMIDT,H.W. (1984b); SCHMIDT,H.W. (1985); KRÄMER (1987a), S. 272ff. u. 276ff.; KRÄMER (1987b), S. 116ff.; VAUTHERIN (1987b), S. 295ff.; FONIO (1987); GERLACH,H. (1987); GERLACH,H. (1988); RECK (1988), S. 81ff.; BATTISTON (1988), S. 22ff., insbesondere S. 28ff.; REISIG (1989a), S. 11ff.
Einen Vorläufer, der bereits die algebraische Fundierung von Petrinetzen betont, hierbei aber noch nicht das Signaturkonzept verwendt, stellt KRÄMER (1985a), S. 310ff., dar.

13) Die Präzision algebraischer (Signatur-)Konzepte wird auch von SCHMIDT,H.W. (1984a), S. 331, und KRÄMER (1985a), S. 308, hervorgehoben. REISIG (1989a), S. I u. 1, betrachtet algebraische (Signatur-)Konzepte als adäquate und flexible Instrumente für den Umgang mit beliebigen strukturierten Objekten. Sogar im Bereich der KI-Forschung werden algebraische (Signatur-)Konzepte neuerdings als ein Ansatz diskutiert, der zur Präzisierung mehrschichtiger Wissensrepräsentationen dienen soll; vgl. VOß,A. (1989b), S. 370ff., insbesondere S. 374ff.

14) In den meisten Beiträgen zu Prädikat/Transition-Netzen wird deren algebraische Beschriftung zumeist nur in halbformaler, großenteils natürlichsprachlicher Diktion eingeführt. Dies unterstreicht die Feststellung von SCHMIDT,H.W. (1984a), S. 333: "PrT-nets are 'open-ended' with respect to specifying the interpretation of inscriptions."

15) Auf Befruchtungsaspekte wird später in einem eigenen Kapitel näher eingegangen. Die wechselseitige Befruchtung zwischen der präzisen algebraischen Spezifikationen von (abstrakten) Datentypen einerseits und Petrinetzen für nebenläufige Systemmodellierungen andererseits heben SCHMIDT,H.W. (1984a), S. 331, und KRÄMER (1985a), S. 308, hervor.

16) ESPRIT steht für: European Strategic Programme for Research and Development in Information Technology.

17) GRASPIN steht für: Incremental Graphical Specification and Formal Implementation on Non-Sequential Systems (mitunter auch: ... Nonsequential Systems). Vgl. zum GRASPIN-Projekt HEYDERHOFF (1984); ENDRES (1988); O.V. (1989h), "Foreword"; vgl. auch die Abstract-Sammlung von Arbeitsberichten zum GRASPIN-Projekt in O.V. (1989h), S. 1ff.

18) SEGRAS steht für: Semi-Graphical Specification Language; vgl. SCHMIDT,H.W. (1984a), S. 330ff.; SCHMIDT,H.W. (1984b); KRÄMER (1985a), S. 307ff., insbesondere S. 313ff.; KRÄMER (1986a), S. 107ff.; KRÄMER (1987a), S. 269ff.; KRÄMER (1987b), S. 116ff.; REISIG (1989a), S. 43f.
Auch die Abstract-Sammlung von Arbeitsberichten zum GRASPIN-Projekt in O.V. (1989h), S. 1ff., enthält vielfache Hinweise auf die Sprache SEGRAS.

19) Strenggenommen handelt es sich um Prädikat/Ereignis-Netze; vgl. z.B. SCHMIDT,H.W. (1984a), S. 330; KRÄMER (1987a), S. 269. Diese gehören jedoch auch zur Klasse der Prädikat/Transition-Netze im Sinne der oben eingeführten weiten Begriffsfassung (Höhere Netze). Prädikat/Ereignis-Netze stellen im wesentlichen Prädikat/Transition-Netze dar, bei denen die zulässigen Markierungen je einer Stelle von Markenmultimengen auf Markenmengen eingeschränkt worden sind. Vgl. zur speziellen Definition von Prädikat/Ereignis-Netzen SCHMIDT,H.W. (1984a), S. 332ff.; REISIG (1985b), S. 111ff., insbesondere S. 114ff.; REISIG (1986a), S. 131ff., insbesondere S. 135ff.; REISIG (1989a), S. 43 i.V.m. S. 36.

20) Vgl. zur Einbindung der netzorientierten Sprache SEGRAS in das GRASPIN-Projekt vor allem KRÄMER (1985a), S. 325f.; KRÄMER (1987a), S. 269.

21) Einen solchen Ansatz stellt z.B. die Spezifikationssprache ACT ONE dar (ACT steht für: Algebraic specification techniques for Correct Design of Trusty software systems). Sie wird seit dem Jahr 1983 an der Technischen Universität Berlin von der ACT-Studiengruppe entwickelt; vgl. EHRIG (1985a), S. 244ff., insbesondere S. 268ff. u. 273ff. Hinweise auf andere algebraische Spezifikationssprachen finden sich bei EHRIG (1985a), S. 305f.

4.1.2 Signaturen

Definition: Signatur

Eine Signatur ist ein geordnetes 2-Tupel SIG=(SO,OP), für das gilt[1]:

- SO ist eine endliche, nicht-leere Menge $\{sort_1,...,sort_I\}$ mit $i \in \{1,...,I\}$ und $I \in \mathcal{N}_+$. Ihre Elemente $sort_i$ heißen Sorten.
- OP ist eine endliche, nicht-leere Menge $\{Op_1,...,Op_J\}$ mit $j \in \{1,...,J\}$ und $J \in \mathcal{N}_+$. Ihre Elemente Op_j werden als Operationssymbole bezeichnet.
- Zwischen den Mengen SO und OP besteht die Beziehung $OP \subseteq (SO^* \times SO)$.

Erläuterungen und Ergänzungen zur Signatur-Definition:

a) Die Sortenmenge SO wird als ein Alphabet interpretiert, das eine formale Sprache für die Signatur SIG erzeugt. Jede Sorte $sort_i$[2] stellt einen Buchstaben des Alphabets SO dar.

b) Jeder Buchstabe $sort_i$ ist zugleich ein 1-stelliges Wort wo aus dem Alphabet SO mit $wo=sort_i$. Ein K-stelliges Wort wo aus dem Alphabet SO mit $k \in \mathcal{N}_+$ und $K \geq 2$ ist ein K-Tupel $wo=(sort_{i(1)},...,sort_{i(K)})$. Es wird auch kurz als Konkatenation $sort_{i(1)} \ldots sort_{i(K)}$ notiert. Das Symbol "λ" ist für jedes Alphabet das leere oder 0-stellige Wort, das aus keinem Buchstaben besteht[3]. SO^K ist die Menge aller K-stelligen Worte mit $K \in \mathcal{N}_0$, die sich durch Aneinanderreihung von Buchstaben des Alphabets SO erzeugen lassen.

c) Der Monoid[4] SO^+ über der Sortenmenge SO ist die Menge aller nicht-leeren und endlichen Worte, die aus dem Alphabet SO gebildet werden können. Der freie Monoid SO^* entspricht dem Monoid über demselben Alphabet SO zuzüglich des leeren Worts: $SO^*=SO^+ \cup \{\lambda\}$[5]. Jede Teilmenge des freien Monoids über dem Alphabet SO stellt eine Sortensprache SOS_{SIG} dar. Folglich gilt[6]:

$$SO^0 = \{\lambda\}$$
$$SO^1 = SO = \{sort_1,...,sort_I\}$$
$$SO^K = SO \times_1 \ldots \times_{K-1} SO \quad \text{für alle } K \in \mathcal{N}_+ \text{ mit } K \geq 2$$
$$SO^+ = \bigcup (K \in \mathcal{N}_+)\text{: } SO^K$$
$$SO^* = SO^+ \cup SO^0 = SO^+ \cup \{\lambda\}$$
$$SOS_{SIG} \subseteq SO^*$$

Falls das leere Wort λ durch $SOS_{SIG} \subseteq SO^+$ ausgeschlossen ist, wird von einer positiven Sortensprache gesprochen.

d) Durch die Einführung von Sorten wird das früher skizzierte formallogische Fundament der Synthetischen Netze erheblich ausgeweitet. Denn die Kalküle konventioneller Formallogiken kennen für ihre Objektsprachen nur unstrukturierte Mengen formaler Objekte. Durch eine Sortensprache kann dagegen einer Menge formaler Objekte - den logischen Individuen - eine komplexe innere Struktur als Sortenstruktur aufgeprägt werden[7]. Wenn eine Objektsprache in dieser Weise durch eine Sortensprache überlagert wird, resultiert eine relativ anspruchsvolle "sortierte" oder "getypte" Logik[8]. Die formalen Objekte ihrer Objektsprache sind kraft ihrer Sortenzugehörigkeiten auf verschiedene, jeweils sortenspezifische Objektklassen aufgeteilt. Eine solche sortierte Logik liegt den hier entfalteten Synthetischen Netzen zugrunde. Sie wird auch bei der späteren prädikatenlogischen Implementierung der Synthetischen Netze durch die ausgewählte Sprache Turbo-PROLOG explizit berücksichtigt[9)].

e) Operationssymbole Op_j sind Platzhalter für konkrete Operationen op_j[10)]. In einer Signatur werden die Operationssymbole grundsätzlich noch nicht konkretisiert. Dies bleibt den später eingeführten SIG-Algebren vorbehalten.

f) Jedes Operationssymbol Op_j aus der Menge OP ist ein formales Konstrukt. Es ordnet einem endlichen, K_j-stelligen Wort wo_j aus der Sortensprache SO^{Kj} mit $wo_j \in SO^{Kj}$ und $K_j \in \mathcal{N}_0$ ein einstelliges Wort $sort_{i(j.Kj+1)}$ aus dem zugrundeliegenden Alphabet SO zu[11)]:

$$Op_j: \quad wo_j \rightarrow sort_{i(j.Kj+1)}$$

Bei Auflösung des Worts wo_j in seine Buchstaben aus dem Sortenalphabet SO gilt:

$$Op_j: \quad sort_{i(j.1)} \ldots sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)}$$

Die Ausdrücke "$wo_j \rightarrow sort_{i(j.Kj+1)}$" und "$sort_{i(j.1)} \ldots sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)}$" bilden die Deklaration des Operationssymbols Op_j. Dabei stellen das Wort "wo_j" und die Buchstabenkonkatenation $sort_{i(j.1)} \ldots sort_{i(j.Kj)}$ alternative Formulierungen für das K_j-stellige Argument des Operationssymbols dar. Die Sorte "$sort_{i(j.Kj+1)}$" ist die Zielsorte des Operationssymbols. Das Operationssymbol Op_j mit dem K_j-stelligen Argument $wo_j \in SO^{Kj}$ wird als K_j-stelliges Operationssymbol bezeichnet.

g) Jedes K_j-stellige Operationssymbol Op_j läßt sich als symbolhafte Darstellung einer K_j-stelligen rechtseindeutigen Abbildung auffassen. Aus dieser Perspektive handelt es sich um ein K_j-stelliges Funktionssymbol Op_j. Daher wird zugelassen, Operationssymbole analog zur etablierten Notationsweise für Funktionen darzustellen[12)]. Hierfür wird vereinbart[13)]:

$$sort_{i(j.Kj+1)} = Op_j(sort_{i(j.1)} \ldots sort_{i(j.Kj)})$$
$$:\Leftrightarrow \quad Op_j: \quad sort_{i(j.1)} \ldots sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)}$$

h) Mehrere Operationssymbole können dieselbe Zielsorte besitzen. In diesem Fall wird die Zielsorte durch die verschiedenen Argumente der jeweils involvierten Operationssymbole auf unterschiedliche Weisen gebildet. Es erfolgt eine multiple Zielsortenspezifizierung:

$$Op_1: \quad sort_{i(1.1)} \ldots sort_{i(1.K1)} \rightarrow sort_{i(1.K1+1)}$$
$$Op_2: \quad sort_{i(2.1)} \ldots sort_{i(2.K2)} \rightarrow sort_{i(2.K2+1)}$$

mit:

$$sort_{i(1.K1+1)} = sort_{i(2.K2+1)}$$

i) Dasselbe Argument "wo_j" kann durch unterschiedliche Operationssymbole "Op_1" und "Op_2" auf verschiedene Zielsorten "$sort_{i(1.K1+1)}$" bzw. "$sort_{i(2.K2+1)}$" mit $sort_{i(1.K1+1)} = sort_{i(2.K2+1)}$ abgebildet werden. Daher ist die Menge OP aller Operationssymbole für die Signatur SIG=(SO,OP) keine Abbildung $SO^* \rightarrow SO$[14)]. Vielmehr stellt sie eine zweistellige Relation OP über dem freien Monoid SO^* und über dem Alphabet SO der Signatur SIG dar:

$$OP \subseteq (SO^* \times SO)$$

j) Die Sortensprache SOS_{SIG} einer Signatur SIG=(SO,OP) ist die Menge aller Worte aus der formalen Sprache des freien Monoids SO* über dem Alphabet SO, die in mindestens einem Operationssymbol Op_j aus der Menge OP als Argument oder Zielsorte vorkommen:

$$SOS_{SIG} = \{wo_j \colon \exists(Op_j \in OP) \colon Op_j \colon wo_j \to sort_{i(j.Kj+1)}\} \cup \ldots$$
$$\{sort_{i(j.Kj+1)} \colon \exists(Op_j \in OP) \colon Op_j \colon wo_j \to sort_{i(j.Kj+1)}\}$$

k) 0-stellige Operationssymbole "$Op_j \colon \lambda \to sort_{i(j)}$"[15] mit dem leeren Wort wo=λ als Argument stellen Konstantensymbole Ko_j mit $Ko_j = Op_j$ dar. Die Rückführung eines Konstantensymbols auf ein 0-stelliges Operationssymbol hat den Vorzug, die Sorte des Konstantensymbols als Zielsorte des Operationssymbols explizit anzugeben. Da das leere Wort "λ" für alle 0-stelligen Operationssymbole identisch ist, braucht es nicht explizit aufgeführt werden[16]. Daher gilt für alle 0-stelligen Operationssymbole die Vereinbarung:

$$Op_j \colon \quad \to sort_{i(j)}$$
$$:\Leftrightarrow \quad Op_j \colon \quad \lambda \to sort_{i(j)})$$

Für die funktionsorientierte Notationsweise eines 0-stelligen Operationssymbols gilt entsprechend::

$$sort_{i(j)} = Op_j()$$
$$:\Leftrightarrow \quad Op_j \colon \quad \lambda \to sort_{i(j)})$$

l) Mehrere 0-stellige Operationssymbole mit derselben Zielsorte $sort_i$[17] können der Einfachheit halber in einer Zeile zusammengezogen werden ($n \in \mathcal{N}_+$ und $n \geq 2$)[18]:

$$Op_{j(1)} \,;\, Op_{j(2)} \,;\, \ldots \,;\, Op_{j(n)} \colon \; \to sort_i$$

Wegen der Deutung 0-stelliger Operationssymbole als Konstantensymbole kann dies mit $Ko_j = Op_j$ auch notiert werden als:

$$Ko_{j(1)} \,;\, Ko_{j(2)} \,;\, \ldots \,;\, Ko_{j(n)} \colon \; \to sort_i$$

m) Mehrere K-stellige Operationssymbole mit $K \in \mathcal{N}_+$ und derselben Zielsorte $sort_1$ lassen sich in analoger Weise zusammenziehen:

$$Op_{j(1)}(sort_{i(1.1)} \ldots sort_{i(1.K1)});$$
$$Op_{j(2)}(sort_{i(2.1)} \ldots sort_{i(2.K2)});$$
$$\ldots$$
$$Op_{j(n)}(sort_{i(n.1)} \ldots sort_{i(n.Kn)}) \; \to sort_i$$

n) 1-stellige Operationssymbole können durch multiple Zielsortenspezifizierung zur Bildung von Subsorten[19] herangezogen werden. Um eine Sorte $sort_i$ aus n Subsorten mit $n \in \mathcal{N}_+$ und $n \geq 2$ zusammenzusetzen, werden n 1-stellige Operationssymbole mit derselben Zielsorte $sort_i$ benötigt. Jedes von ihnen bildet eine der Subsorten auf diese Zielsorte ab:

$Op_{j(1)}$: $sort_{i(j(1))} \rightarrow sort_i$
...
$Op_{j(n)}$: $sort_{i(j(n))} \rightarrow sort_i$

oder in Kurznotation:

$Op_{j(1)}(sort_{i(j(1))})$; ... ; $Op_{j(n)}(sort_{i(j(n))}) \rightarrow sort_i$

o) Die Sortenmenge SO der Signatur SIG umfaßt genau alle Sorten $sort_i$, die zum Argument oder zur Zielsorte mindestens eines Operationssymbols $Op_j \in OP$ gehören[20)].

p) Signaturen stellen das formale Basiskonzept dar, auf dem alle nachfolgend entfalteten Konzepte bis hin zu den Synthetischen Netzen beruhen. Aus den Sorten und den Operationssymbolen einer Signatur SIG wird jeweils eine abstrakte formale Struktur[21)] geformt, die allen hieraus abgeleiteten formalen Konzepten zugrundeliegt. Aus dieser Perspektive wird auch von der Sortenstruktur gesprochen, die eine Signatur SIG jedem ihrer SIG-Derivate überlagert.

q) Eine Signatur SIG=(SO,OP) mit einem konkreten Signaturnamen sowie mit konkret definierten Sorten und Operationssymbolen wird zwecks größerer Transparenz als eine strukturierte Liste nach folgendem Schema notiert[22)]:

<<u>Signaturname</u>> =

<u>sorts</u>: $sort_1$
...
$sort_I$

<u>Ops</u>: Op_1: $sort_{i(1.1)} \ldots sort_{i(1.K1)} \rightarrow sort_{i(1.K1+1)}$
...
Op_J: $sort_{i(J.1)} \ldots sort_{i(J.KJ)} \rightarrow sort_{i(J.KJ+1)}$

<u>Beispiel zur Signatur-Definition:</u>

Die Signatur mit dem Namen "String" besteht aus den Sorten "zeichen" und "zeichenfolge". 26 Konstantensymbole Ko_k mit $k \in \{1,...,26\}$ sind für die Sorte "zeichen" definiert. Drei weitere Operationssymbole "Erzeuge0", "Erzeuge1" und "Verknüpfe" besitzen dieselbe Zielsorte "zeichenfolge", aber unterschiedliche Argumente. Durch die ersten beiden 0- und 1-stelligen Operationssymbole werden die leere Zeichenfolge bzw. beliebige 1-stellige Zeichenfolgen konstituiert. Das dritte, 2-stellige Operationssymbol dient der Verknüpfung (Konkatenation) beliebiger Zeichenfolgen zu neuen Zeichenfolgen. Die vollständige Definition der Signatur "String" wird notiert als:

<u>String</u> =

<u>sorts</u>: zeichen
zeichenfolge

<u>Ops</u>:

$Ko_1,...,Ko_{26}$:	$\rightarrow$ zeichen
Erzeuge0:	$\rightarrow$ zeichenfolge
Erzeuge1:	zeichen $\rightarrow$ zeichenfolge
Verknüpfe:	zeichenfolge zeichenfolge $\rightarrow$ zeichenfolge

Für die Sorte "zeichenfolge" liegt eine multiple Zielsortenbeschreibung durch die Operationssymbole "Erzeuge0", "Erzeuge1" und "Verknüpfe" vor. Gleiches gilt für die Zielsorte "zeichen" durch die 26 0-stelligen Operationssymbole (Konstantensymbole) $Ko_1,\ldots,Ko_{26}$.

Anmerkungen zum Kapitel:

1) Vgl. EHRIG (1985a), S. 14f.; RECK (1988), S. 43; REISIG (1989a), S. 5. Die vorgenannten Quellen enthalten allerdings nicht die dritte Definitionskonstituente, welche die Beziehung zwischen den Mengen der Sorten und der Operationssymbole betrifft. Grundsätzlich ist zu unterscheiden zwischen der Definition eines allgemeinen Signatur-Schemas und der Definition konkreter Signaturen, die jeweils das allgemeine Signatur-Schema erfüllen. Definitionen beziehen sich in dieser Arbeit bei der grundlegenden Einführung neuer Konzepte - wenn nicht ausdrücklich anders vermerkt - stets auf die allgemeinen Schemata. (Vgl. dazu auch die Anmerkung zur Definition von Netzen durch allgemeine Netz-Schemata). Daher wird der Einfachheit halber nicht zwischen einem allgemeinen Definitions-Schema und einer einzelnen konkreten Definition unterschieden, sofern aus dem Kontext ersichtlich ist, welches von beiden gerade gemeint ist. So wird nachfolgend das allgemeine Signatur-Schema definiert, während später eine konkrete Signatur exemplarisch definiert wird. Auf die Differenzierung zwischen allgemeinem Definitions-Schema und konkreter Definition wird fortan nicht mehr gesondert hingewiesen.

2) Auf den identifizierenden Index "i" der Sorte "$sort_i$" kann verzichtet werden, wenn jeweils nur eine Sorte "sort" betrachtet wird. Gleiches gilt, falls die Sortenbezeichnungen durch paarweise verschiedene Sortennamen ersetzt werden: z.B. die Sorten "$sort_1$" und "$sort_2$" durch die Namen "attr" bzw. "mark".
Um natürlichsprachliche Namen für Operationssymbole vom benachbarten Text deutlich zu unterscheiden, können die Namen - wie voranstehend - durch Anführungszeichen gekennzeichnet werden. Eine analoge Notationskonvention gilt auch für alle später eingeführten formalsprachlichen Konstrukte, wenn sie in natürlichsprachlichen Kontexten verwendet werden und aufgrund ihrer Bezeichnungsweise mit natürlichsprachlichen Ausdrücken verwechselt werden könnten. (Da der formalsprachliche Ausdruck "$K \in \mathcal{N}_+$" infolge des Element-Notators "$\in$" unmittelbar als formalsprachliches Konstrukt erkannt werden kann, erfolgt in diesem Fall keine Notation mit Anführungszeichen). Die Anführungszeichen zur Kennzeichnung formalsprachlicher Ausdrücke in *natürlich*sprachlichen Kontexten sind nicht mit der Kennzeichnung von Zeichenfolgen zu verwechseln, die in *formal*sprachlichen Kontexten als formale Objekte vom Typ "string" verwendet werden (vgl. dazu die Signatur "String" und die zugehörige Algebra "string").

3) In Theorien formaler Sprachen wird das leere Wort im allgemeinen mit dem griechischen Buchstaben "λ" bezeichnet. Das leere Wort ist erforderlich, um später 0-stellige Operationssymbole formal als Konstrukte einer formalen Sprache über der Sortenmenge SO einführen zu können.

4) Monoide wurden bereits an früherer Stelle definiert.

5) Wird die Sortenmenge SO nicht als Alphabet einer formalen Sprache, sondern als eine gewöhnliche Menge aufgefaßt, so entspricht der freie Monoid SO^* der Vereinigungsmenge aller K-stelligen kartesischen Produkte über der Sortenmenge SO mit $K \in \mathcal{N}_0$. Dabei wird ein erweiterter Begriff des kartesischen Produkts vorausgesetzt, der auch null- und einstellige "kartesische Produkte" umfaßt, welche die nachfolgenden Definitionen für SO^0 bzw. SO^1 erfüllen.

6) Mit "x" wird der Operator für die Bildung des kartesischen Produkts bezeichnet. "x_k" notiert das k-te Vorkommnis des kartesischen Produktoperators in einer Formel.

7) Im Rahmen des Signaturkonzepts sind zwar noch keine formalen Objekte definiert, die den Sorten zugeordnet sind. Doch dies geschieht im nachfolgenden Kapitel durch die Einführung von SIG-Algebren. Vgl. dort vor allem die Erläuterungen zur Überlagerung des algebraischen Universums durch eine Sortenstruktur.

8) Näheres dazu später im Kontext der Prädikatenlogik. Die sortierten sind zu unterscheiden von mehrwertigen Logiken, deren Wahrheitswerte mehr als zwei unterschiedliche Ausprägungen annehmen können. Diese Multiplizität der Wahrheitswerte besitzt keinen Bezug zur hier angesprochenen Sortierung der objektsprachlichen Konstrukte (Individuen) durch sortierte Logiken. Vgl. zu solchen mehrwertigen Logiken z.B. RESCHER (1968b), S. 74ff.; RESCHER (1969); RAUTENBERG (1979), S. 103ff.; GINSBERG (1986), S. 243ff.

9) Dies geschieht im Rahmen der "domains"-Sektion; Näheres dazu später. Der Verf. hat seiner Auswahl einer Programmiersprache für die prädikatenlogische Implementierung von Synthetischen Netzen die Möglichkeit, eine sortierte Logik zu benutzen, als ein wesentliches Selektionskriterium zugrundegelegt. Insbesondere aus diesem Grund fiel die Auswahl zugunsten des Dialekts "Turbo-PROLOG" aus. Denn die Mehrzahl etablierter PROLOG-Dialekte kann solche sortierten Logiken nicht explizit repräsentieren. Dies wird besonders deutlich von BEIERLE (1988b), S. 161 u. 164f., dargelegt.
Dagegen existieren im Kontext der logischen Programmierung durchaus mehrere Ansätze zur Verwendung von Sorten. Sie werden dort zumeist als "Typen" thematisiert. Allerdings haben diese Beiträge bisher noch kaum Eingang in die zuvor erwähnten etablierten PROLOG-Dialekte gefunden. Statt dessen bleiben sie auf unkonventionelle PROLOG-Erweiterungen beschränkt, die zumeist auf spezielle Applikationsdomänen zugeschnitten und nur schwer zugänglich sind. Vgl. zu solchen "sortierten" oder "getypten" Varianten von logischer Programmierung und PROLOG VARSEK (1986), S. 214 u. 219f.; GOOS (1987a), S. 89 u. 91ff., insbesondere S. 92ff.; CORRENZ (1987), S. 236; BEIERLE (1988b), S. 160ff.; SMOLKA (1988a), S. 183ff.; ESTER (1989), S. 34ff., 51ff. u. 126.

Darüber hinaus erfolgen im Rahmen des europäischen Forschungsprogramms ESPRIT Anstrengungen, in das Konzept der logischen Programmierung die Berücksichtigung von Sorten einzubinden; vgl. COMMISSION OF THE EUROPEAN COMMUNITIES (1989), S. I-14.

10) Symbole werden in dieser Arbeit von den konkreten Entitäten, welche die Symbole vertreten, durch verschiedene Notationsweisen unterschieden: Konkrete Entitäten beginnen stets mit Kleinbuchstaben. Die zugehörigen Symbole werden dagegen von Großbuchstaben eingeleitet. Dies gilt allerdings nur als Rahmenregel, von der Abweichungen in begründeten Sonderfällen zugelassen werden. Solche Besonderheiten liegen vor allem in zwei Fällen vor. Erstens kann die o.a. Unterscheidung zwischen initialen Groß- und Kleinbuchstaben nur vorgenommen werden, falls die Bezeichnungen (Namen) der betroffenen Symbole bzw. konkreten Entitäten jeweils mit einem Buchstaben beginnen. Dies muß keineswegs so sein, weil Bezeichnungen auch mit numerischen Zeichen oder Sonderzeichen anstelle von Buchstaben eingeleitet werden können, wie z.B. das Symbol "$\varnothing$" als Bezeichnung für die leere Menge. Zweitens wird später als Zuordnungskonvention eingeführt, Konstantensymbole mit ihren konkreten Konstanten zu identifizieren. Dann kann eine unterschiedliche Notation nicht mehr aufrecht gehalten werden.
Die oben eingeführte Notationskonvention entspricht der Programmiersprache PROLOG. In ihr werden Variablennamen mit Großbuchstaben eingeleitet, während alle anderen Namen mit Kleinbuchstaben beginnen. Allerdings können Variablennamen auch durch das Zeichen "_" für die anonyme Variable ersetzt werden. Namen für Konstrukte, die keine Variablen sind, dürfen darüber hinaus mit Sonderzeichen oder dem speziellen Zeichen "'" beginnen. Vgl. zu diesen Notationskonventionen PROLOG (o.J.), S. 197; KINNEBROCK (1988), S. 119f.; CORDES (1988), S. 185 u. 187. Auf diese Besonderheiten, welche die o.a. Notationskonvention im wesentlichen unberührt lassen, wird nicht weiter eingegangen.

11) Statt dessen kann ein Operationssymbol Op_j auch als eine Menge $\{wo_j, sort_{i(j.Kj+1)}\}$ aus den zugehörigen Worten wo_j und $sort_{i(j.Kj+1)}$ aufgefaßt werden. Dann ist die Menge OP der Operationssymbole Op_j die Vereinigungsmenge aller Mengen $\{wo_j, sort_{i(j.Kj+1)}\}$, sofern sichergestellt ist, daß diese Mengen jeweils paarweise disjunkt sind. Diese Interpretation der Operationssymbole einer Signatur scheint RECK (1988), S. 42f., zu unterstellen. Allerdings bezeichnet er Operationssymbole nur als "paarweise disjunkte Teilmengen", ohne die Bezugsmenge dieser Teilmengen und ohne die Elemente dieser Teilmengen explizit zu definieren.
In dieser Arbeit braucht die Disjunktheitsforderung nicht erhoben zu werden. Die Definition der Operationssymbole als formale Konstrukte, die Elemente der Menge OP sind, gewährleistet bereits die paarweise Unterschiedlichkeit aller Operationssymbole. Denn der Mengenbegriff impliziert, daß alle Elemente einer Menge paarweise verschieden sind.

12) Diese funktionsorientierte Notationsweise von Operationssymbolen wird später intensiv genutzt, um die Sektion "Marken/Operationssymbole" der Netzlegenden Synthetischer Netze kompakt zu formulieren.

13) Die nachfolgende Gleichung ist nur für Sorten und Operationssymbole erklärt. Sie besitzt daher nicht die Qualität von Gleichungen, die bei der Definition der Abbildungsvorschrift einer Funktion zwischen Termen aufgestellt wird. Für die definitorische Rückführung einer neuen Formel "for_n" auf eine bereits definierte Formel "for_a" wird hier der zweistellige metasprachliche Definitionsoperator "$:\Leftrightarrow$" in Infixnotation eingeführt. Die Formel "for_n" wird durch eine andere, bereits eingeführte Formel "for_a" und den Ausdruck "$for_n :\Leftrightarrow for_a$" nach der Maßgabe definiert, daß der linksstehenden Formel "for_n" genau dann der Gültigkeitsstatus "(un)gültig" zugeschrieben wird, wenn die rechtsstehende Formel "for_a" als (un)gültig bekannt ist. Es handelt sich um einen Aspekt der formalen Semantik. Denn die Bedeutung des formalen Konstrukts "for_a" wird durch die bereits eingeführte Bedeutung des anderen formalen Konstrukts "for_a" erklärt. Vgl. dazu auch die analoge Festlegung des gleichen Definitionsoperators im Rahmen der prädikatenlogischen Semantik.

14) Bei einer Abbildung darf jedem Element des Vorbereichs nur höchstens ein Element des Nachbereichs zugeordnet werden. Hiergegen würde durch die Zuordnung der zwei Elemente "$sort_{i(1.K1+1)}$" und "$sort_{i(2.K2+1)}$" aus dem Nachbereich SO zum selben Element "wo_j" aus dem Vorbereich SO^* verstoßen.

15) Die ausführliche Bezeichnung "$sort_{i(j.1)}$" der Zielsorte wird für 0-stellige Operationssymbole Op_j mit K_j=0 zu "$sort_{i(j)}$" vereinfacht.

16) Es wird dann jeweils implizit als Argument des 0-stelligen Operationssymbols unterstellt.

17) Die identischen Zielsorten $sort_{i(j(1))},\ldots,sort_{i(j(n))}$ der Operationssymbole $Op_{j(1)},\ldots,Op_{j(n)}$ werden mit $sort_{i(j(1))}=\ldots=sort_{i(j(n))}=sort_i$ durch die Sorte $sort_i$ vertreten.

18) Das Zeichen ";" besitzt in dieser Notation die Bedeutung des "exklusiven oder".

19) Die Subsortenbildung ist bei der logischen Programmierung noch weit seltener üblich als die Sortenbildung. Sie ist jedoch im Rahmen des Turbo-PROLOG-Dialekts, der später zur prädikatenlogischen Implementierung von Synthetischen Netzen benutzt wird, prinzipiell möglich. Bei BEIERLE (1988b), S. 162, 165 u. 170, werden Subsorten für einen neueren PROLOG-Dialekt (PROTOS-L) sogar unmittelbar zugelassen. SMOLKA (1988a), S. 183 u. 187ff., behandelt Subsorten im allgemeineren Fall der logischen Programmierung im Kontext sortierter Logiken.

In konventionellen PROLOG-Dialekten, die keine direkte Sortenbildung kennen, müssen Subsorten durch zusätzliche Hornklauseln definiert werden. Meist hat dies aufwendigere Programmauswertungen durch zusätzliche backtracking-Schritte zur Folge. Vgl. zu solchen Hilfskonstruktionen und deren Schwierigkeiten BEIERLE (1988b), S. 164(f.).

20) Dadurch werden abundante Sorten ausgeschlossen, die zwar in der Sortenmenge SO einer Signatur SIG=(SO,OP) enthalten sind, aber zu keiner Deklaration eines Operationssymbols aus der Menge OP benötigt werden. Daher stellt die Menge SO diejenige Sortenmenge dar, die zur Deklaration aller Operationssymbole der Signatur SIG *mindestens* erforderlich ist. Dies entspricht dem Sparsamkeitsprinzip von "OCKHAM's razor".

21) Der Strukturbegriff wird hier im intuitiven, vor-formalen Sinn verstanden. Er besitzt keinen Bezug auf den systemtheoretischen Strukturbegriff.

22) Die Notation "<text>" wird in dieser Arbeit als ein Platzhalter für einen natürlich- oder formalsprachlichen Ausdruck benutzt. Er wird vorläufig durch die Bezeichnung "text" vertreten und muß noch einzelfallspezifisch konkretisiert werden.

4.1.3 SIG-Algebren

Eine SIG-Algebra geht aus einer Signatur SIG=(SO,OP) dadurch hervor, daß den Sorten $sort_i$ aus der Sortenmenge SO einer Signatur SIG Mengen OB_i formaler Objekte zugeordnet werden. Die Sorten werden durch diese formalen Objektmengen interpretiert. Die Operationssymbole $Op_j \in OP$ aus der Signatur SIG werden entsprechend als Operationen op_j erklärt, die sich auf den Mengen formaler Objekte ausführen lassen. Die abstrakten formalen Konstrukte einer Signatur - Sorten und Operationssymbole - werden also durch konkrete formale Konstrukte - Objekte bzw. Operationen - interpretiert. Eine SIG-Algebra stellt somit eine formale Semantik der Signatur SIG dar[1]. Je nach Festlegung der sortenspezifischen Objektmengen und der hierüber definierten Operationen resultieren unterschiedliche SIG-Algebren für dieselbe Signatur SIG. Einer Signatur SIG können beliebig viele verschiedene SIG-Algebren zugeordnet sein.

Definition: Algebra

Eine SIG-Algebra bezüglich der Signatur SIG=(SO,OP) ist ein geordnetes 2-Tupel A_{SIG}=(OBF,OPF), für das gilt[2]:

- OBF ist eine Mengenfamilie[3] (OB_i: $i=1,...,I$). Ihre Mitglieder OB_i sind jeweils nicht-leere und endliche Mengen beliebiger formaler Objekte ob.
- Jeder Sorte $sort_i \in SO$ aus der Signatur SIG ist genau eine sortenspezifische Objektmenge $OB_i \in OBF$ zugeordnet[4].
- OPF ist eine Familie (op_j: $j=1,...,J$) Operationen op_j[5]. Die Vor- und Nachbereiche der Operationen erstrecken sich über diejenigen Objektmengen OB_i aus der Mengenfamilie OBF, die den Argument- und Zielsorten des Operationssymbols Op_j zugeordnet sind.
- Jedem Operationssymbol $Op_j \in OP$ aus der Signatur SIG entspricht genau eine Operation $op_j \in OPF$.

<u>Erläuterungen und Ergänzungen zur Algebra-Definition:</u>

a) Die sortenspezifischen Objektmengen OB_i werden auch als Definitionsbereiche, Trägermengen oder Domänen der jeweils zugehörigen Sorten "$sort_i$" bezeichnet. Nicht-leere Objektmengen müssen vorausgesetzt werden, da andernfalls die hiervon abhängigen Operationen nicht mehr wohldefiniert wären. Die geforderte Endlichkeit der Objektmengen ist dagegen theoretisch nicht notwendig. Grundsätzlich könnte auch mit unendlichen Objektmengen umgegangen werden. Aus praktischen Gründen wird jedoch auf diese Option verzichtet. Erstens beziehen sich einige der später eingeführten Konstrukte auf die Kombinationsmöglichkeiten der formalen Objekte aus verschiedenen Objektmengen. Die Behandlung dieser Konstrukte würde zum Teil erheblich erschwert, wenn unendliche Mengen zulässiger Objektekombinationen zugelassen wären[6]. Zweitens erzwingt die spätere Implementierung von Netzmodellen auf realen Automatischen Informationsverarbeitungssystemen eine Beschränkung auf endliche Objektmengen[7].

b) Die Operationen op_j, die K_j-stellige Operationssymbole Op_j mit $K_j \in \mathcal{N}_+$[8] aus der zugrundeliegenden Signatur interpretieren, stellen rechtseindeutige Abbildungen (Funktionen[9]) dar[10]. Das Operationssymbol Op_j mit:

$$Op_j: \quad sort_{i(j.1)} \ldots sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)}$$

wird interpretiert durch die Operation op_j mit:

$$op_j: \quad OB_{i(j.1)} \text{ x...x } OB_{i(j.Kj)} \rightarrow OB_{i(j.Kj+1)}$$
$$(ob_1,...,ob_{Kj}) \rightarrow op_j(ob_1,...,ob_{Kj}) = ob_{Kj+1}$$

Der Ausdruck "... →..." bildet die Deklaration der Operation op_j. Das kartesische Produkt $OB_{i(j.1)}\text{x...x}OB_{i(j.Kj)}$ stellt den Vorbereich VB_j, die Menge $OB_{i(j.Kj+1)}$ den Nachbereich NB_j der Operation op_j dar[11]. Jedes K_j-Tupel $(ob_1,...,ob_{Kj})$ aus dem Vorbereich VB_j stellt ein mögliches Argument der Operation op_j dar. Der Definitionsbereich DB_j der Operation op_j umfaßt alle K_j-Tupel $(ob_1,...,ob_{Kj})$ aus dem Vorbereich VB_j, für welche die Operationsanwendung zulässig ist. Es handelt sich daher um alle zulässigen Argumente der Operation op_j. Für den Definitionsbereich gilt stets $DB_j \subseteq VB_j$. Als Regelfall wird unterstellt, daß die Operation op_j auf alle Tupel aus ihrem Vorbereich angewendet werden darf[12]. Dann fallen ihr Vor- und ihr Definitionsbereich zusammen ($DB_j = VB_j$). Andernfalls ist der Definitionsbereich eine echte Teilmenge ihres Vorbereichs. In diesem letzten Fall mit $DB_j \subset VB_j$ liegt eine partiell definierte - oder kurz: partielle - Operation vor[13]. Der Ausdruck $op_j(ob_1,...,ob_{Kj})$, der aus der Anwendung der Operation op_j auf die formalen Objekte $ob_1,...,ob_{Kj}$ aus den Mengen $OB_{i(j.1)},...,OB_{i(j.Kj)}$ hervorgeht, wird als Operatorapplikation bezeichnet. Die formalen Objekte $ob_1,...,ob_{Kj}$ bilden das Argument $arg = (ob_1,...,ob_{Kj})$ der Operatorapplikation. Der Ausdruck $op_j(...)$ ohne Argumentspezifizierung stellt den Operator für die Operation op_j dar. Das formale Objekt ob_{Kj+1} aus der Menge $OB_{i(j.Kj+1)}$ heißt das Ergebnis der Operatorapplikation, wenn $ob_{Kj+1} = op_j(ob_{j.1},...,ob_{j.Kj})$ gilt. Das formale Objekt ob_{Kj+1} wird auch als das Bild des Arguments $(ob_{j.1},...,ob_{j.Kj})$ unter der Operation op_j bezeichnet. Alle Elemente des Nachbereichs, die Ergebnisse von Operatorapplikationen darstellen, definieren die Bildmenge oder den Wertebereich WB_j der Operation op_j. Es gilt stets $WB_j \subseteq NB_j$.

c) Jedes 0-stellige Operationssymbol $Op_j \in OP$ aus der Signatur SIG mit $Op_j: \rightarrow sort_{i(j)}$ wird durch eine 0-stellige Operation $op_j: \rightarrow OB_{i(j)}$ interpretiert. Sie legt für das Operationssymbol Op_j genau ein formales Objekt "ob" aus der sortenspezifischen Objektmenge $OB_{i(j)}$ mit $op_j() = op_j(\lambda) = ob$[14] fest. Das formale Objekt "ob" ist das Bild der Operation op_j für das leere Argument $() = (\lambda)$. Anstelle der formal vollständigen Notation "$op_j: \rightarrow OB_{i(j)}$, $op_j() = ob$" kann für diese Festlegung auch die Kurznotation "ob: $\rightarrow OB_{i(j)}$" benutzt werden. Falls mehrere 0-stellige Operationssymbole $Op_{j(k)}$ mit $k \in \{1,...,K\}$ und $K \geq 2$ dieselbe Zielsorte $sort_{i(j)}$ besitzen, werden als Kurznotationen vereinbart:

$op_{j(1)}: \rightarrow OB_{i(j)},\ op_{j(1)}() = ob_1$

...

$op_{j(K)}: \rightarrow OB_{i(j)},\ op_{j(K)}() = ob_K$

kann vereinfacht notiert werden als:

$op_{j(1)}, \dots, op_{j(K)}: \rightarrow OB_{i(j)}$

mit: $op_{j(1)}() = ob_1, \dots, op_{j(K)}() = ob_K$;

dies kann wiederum vereinfacht notiert werden als:

$ob_1, \dots, ob_K: \rightarrow OB_{i(j)}$

d) Die 0-stellige Operation $op_j()$ und das hierdurch definierte formale Objekt "ob" werden gemeinsam als eine Konstante[15)] bezeichnet. Beide repräsentieren dieselbe Konstante, die das Operationssymbol Op_j aus der Signatur SIG in der SIG-Algebra interpretiert. Daher können alle Konstanten einer SIG-Algebra in zwei formal gleichwertigen Weisen eingeführt werden: Entweder werden sie als formale Objekte "ob" in den sortenspezifischen Trägermengen OB_i definiert. Ebenso lassen sie sich aber auch als die Bilder von 0-stelligen Operationen op_j mit $op_j() = ob$ oder ob: $\rightarrow OB_{i(j)}$ darstellen. Welche dieser äquivalenten Repräsentationsformen gewählt wird, ist im Rahmen des Signaturkonzepts nicht präzise festgelegt. Beispielhaften Verdeutlichungen des Signaturkonzepts kann jedoch entnommen werden, daß durch die sortenspezifischen Objektmengen niemals besondere formale Objekte spezifiziert werden[16)]. Die Objektmengen OB_i werden immer nur als Mengen allgemein definierter Objekte behandelt, wie z.B. die Objektmenge "INTEGER" aller Ganzzahlen[17)]. Dieser Konvention, die zumeist nicht explizit dokumentiert ist, schließt sich der Verf. als Regelfall an[18)]. Falls eine Objektmenge dagegen in spezieller Weise definiert werden soll, so geschieht dies immer als Bildmenge von Operationen: Wenn die mengenzugehörigen Objekte atomar sind, werden diese Konstanten als die Bilder von 0-stelligen Operationen unmittelbar eingeführt[19)]. Es wird dann auch von originär definierten Objekten gesprochen. Sofern es sich um zusammengesetzte Objekte handelt, werden sie als Bilder von mehrstelligen Operationen mittelbar etabliert. Bei diesen Bildern handelt es sich um Komplexe, die aus anderen Objekten und aus Operatoren aufgebaut sind. Sie werden auch als derivativ definierte Objekte bezeichnet. In den beiden letztgenannten Fällen brauchen die betroffenen Objektmengen in der Mengenfamilie OBF nicht näher spezifiziert zu werden. Statt dessen reicht es aus, die Objektmengen als originär oder derivativ definierte Bildmengen innerhalb der Operationenfamilie OPF festzulegen[20)].

e) Da sich alle Konstantensymbole Ko_j auf 0-stellige Operationssymbole Op_j mit $Op_j \in OP$ zurückführen lassen, gilt analog zu den beiden voranstehenden Erläuterungen: Jedes Konstantensymbol $Ko_j \in OP$ aus der Signatur SIG mit $Ko_j: \rightarrow sort_{i(j)}$ wird durch eine 0-stellige Operation $ko_j: \rightarrow OB_{i(j)}$ interpretiert. Sie ordnet dem Konstantensymbol Ko_j genau ein formales Objekt "ob" aus der sortenspezifischen Objektmenge $OB_{i(j)}$ mit $ko_j() = ko_j(\lambda) = ob$ zu. Anstelle der formal vollständigen Notation "$ko_j: \rightarrow OB_{i(j)},\ ko_j() = ob$" kann für diese Zuordnung auch die Kurznotation "ob: $\rightarrow OB_{i(j)}$" benutzt werden. Die 0-stellige Operation $ko_j()$ und das hierdurch definierte formale Objekt "ob" werden gemeinsam als eine Konstante bezeichnet. Beide repräsentieren dieselbe Konstante, die das Konstantensymbol Ko_j aus der Signatur SIG in der SIG-Algebra interpretiert. Solche Konstanten können einerseits als formale Objekte "ob" in den sortenspezifischen Trägermengen OB_i definiert werden. Andererseits lassen sie sich ebenso als 0-stellige Operationen ko_j mit $ko_j() = ob$ oder ob: $\rightarrow OB_{i(j)}$ darstellen.

f) Die formalen Objekte aus einer sortenspezifischen Objektmenge OB_i können atomar oder zusammengesetzt sein. Atomare oder einfache Objekte liegen vor, wenn sie als Elemente der Objektmenge OB_i originär eingeführt werden. Ein zusammengesetztes oder komplexes Objekt wird dagegen aus anderen, bereits eingeführten formalen Objekten durch mindestens eine Operatorapplikation in derivativer Weise gebildet. Beispielsweise stellt der Ausdruck $op_j(ob_1,...,ob_{Kj})$ ein komplexes formales Objekt dar, dessen interne Struktur aus den einfacheren formalen Objekten $ob_1,...,ob_{Kj}$ und dem Operator $op_j(...)$ besteht. Solche komplexen Objekte heißen daher auch strukturierte Objekte.

g) Jeder Sorte $sort_i$ wird durch die Objektmengenfamilie OBF genau eine Objektmenge OB_i originär zugeordnet. Neben dieser eindeutigen originären Objektmengendefinition kann eine Sorte durch beliebig viele weitere Objektmengen in derivativer Weise interpretiert werden. Dies ist immer dann der Fall, wenn diese Sorte die Zielsorte von Operationssymbolen aus der zugrundeliegenden Signatur ist. Dabei wird die Sorte durch die Bildmengen derjenigen Operationen interpretiert, die jenen Operationssymbolen zugeordnet sind. Wenn mindestens eine solche derivative Objektmengendefinition erfolgt, liegt eine multiple Sorteninterpretationen durch mehrere Objektmengen vor[21)].

h) Bei der multiplen Interpretation einer Sorte $sort_i$ kann das gleiche formale Objekt ob_j ebenso als atomares Objekt wie auch als zusammengesetztes Objekt aufgefaßt werden. Im ersten Fall wird auf die originäre Sorteninterpretation durch die unmittelbar zugeordnete Objektmenge OB_i Bezug genommen. Dem zweiten Fall liegt dagegen eine derivative Interpretation derselben Sorte durch die Bildmenge einer Operation zugrunde. Daher kann es durchaus zulässig sein, das gleiche formale Objekt - je nach Bezugspunkt - einmal als atomar und ein anderes Mal als zusammengesetzt zu behandeln[22)]. Es hängt von der jeweils gewählten formalen Darstellungsweise ab, ob das betroffene Objekt entweder atomar oder aber zusammengesetzt erscheint.

i) Darüber hinaus ist es möglich, dem gleichen formalen Objekt unterschiedliche interne Strukturen zuzuordnen[23)]. Hierdurch wird das gleiche Objekt aus verschiedenen Blickwinkeln in differierenden formalen Gestalten erfaßt[24)]. Daher erweist sich das Signaturkonzept als bemerkenswert flexibles Instrument für die formale Objektmodellierung.

j) Die sortenspezifischen Objektmengen OB_i brauchen nicht paarweise disjunkt zu sein. Daher können sich die Objektmengen zweier verschiedener Sorten auch partiell überlappen, ineinander enthalten oder sogar identisch sein. Folglich darf dasselbe formale Objekt zugleich verschiedenen Objektmengen OB_i für unterschiedliche Sorten angehören. Die unterschiedlichen Sorten lassen sich auch hier als verschiedene Perspektiven für dasselbe formale Objekt interpretieren. Im Gegensatz zur voranstehenden Erläuterung steht aber nicht der Aspekt alternativer Strukturierungen des gleichen Objekts im Vordergrund. Statt dessen dominiert hier der Gesichtspunkt, daß dasselbe Objekt als Bestandteil der Interpretationen verschiedener Sorten in unterschiedlichen Kontexten verwendet werden kann[25)]. Beide Varianten des formalen Perspektivismus - alternative interne Objektstrukturen und alternative Objektkontexte - lassen sich miteinander kombinieren. Dann wird das gleiche formale Objekt in verschiedenen Kontexten mit unterschiedlichen internen Strukturen verwendet[26)].

k) Beliebig komplex strukturierte formale Objekte lassen sich induktiv definieren, indem die komplementären Objektbildungsprinzipien der horizontalen und der vertikalen Objektzusammensetzung miteinander kombiniert werden. Bei der horizontalen Objektzusammensetzung werden *mehrere* formale Objekte auf *derselben* Stufe durch genau einen Operator zu einem formalen Objekt der nächsthöheren Stufe zusammengesetzt. Das formale Objekt der höheren Stufe besitzt eine flache Struktur[27)]. Durch die vertikale Objektzusammensetzung wird dagegen ein formales Objekt *höherer Stufe* durch *mindestens* einen Operator aus *mindestens einem* formalen Objekt *tieferer Stufe(n)* gebildet[28)]. Wenn die vertikale Zusammensetzung mehrfach erfolgt, erhält das

Objekt auf der obersten Ebene eine tiefe Struktur[29]. Horizontale und vertikale Objektbildung schließen sich nicht gegenseitig aus. Vielmehr stellen sie unterschiedliche Strukturierungsprinzipien dar, die bei der Zusammensetzung formaler Objekte im Regelfall kombiniert auftreten[30].

l) Die Vereinigung aller Objektmengen OB_i aus der Mengenfamilie OBF ist das Universum OB, das durch die Algebra A_{SIG} auf der Signatur SIG errichtet wird[31]:

$$OB = \bigcup (i \in \{1,...,I\}): OB_i$$

Das algebraische Universum OB ist die Menge aller formalen Objekte, die in der SIG-Algebra A_{SIG} definiert sind[32].

m) Dem Universum OB einer SIG-Algebra wird durch die Überlagerung der Sortensprache SOS_{SIG} aus der zugrundeliegenden Signatur SIG eine beliebig komplexe "innere" Struktur aufgeprägt[33]. Diese Sortenstruktur des algebraischen Universums OB ist mit der bereits eingeführten Familie OBF aller sortenspezifischen Objektmengen OB_i identisch. Die Sortenstruktur $OBF = (OB_i: i = 1,...I)$ konstituiert die algebraische Ontologie der Signatur SIG. Da eine induktive Objektzusammensetzung nach vertikalen und horizontalen Objektbildungsprinzipien zugelassen wird, besitzt das Universum OB eine hierarchisch geschichtete Ontologie[34]. Durch eine große Vielfalt der Operationssymbole Op_j aus der Menge OP, deren Argumente Worte aus der Sortensprache SOS_{SIG} darstellen, kann trotz einer begrenzten Sortenmenge SO der Signatur SIG = (SO,OP) die Struktur des Universums OB unbeschränkt ausdifferenziert werden. Diese komplexe Ontologie des algebraischen Universums ist der *wesentliche Ansatzpunkt* für die spätere Einführung strukturierter Marken.

n) SIG-Algebren lassen sich zunächst als Algebren allgemeinster Art[35] auffassen, in denen beliebige formale Operationen über beliebigen formalen Objekten erklärt sind. Diese Operationen werden hier als die Klasse aller K-stelligen rechtseindeutigen Abbildungen (Funktionen) mit $K \in \mathcal{N}_0$ eingeschränkt. Insofern handelt es sich um funktionale Algebren[36]. Darüber hinaus werden nicht beliebige Definitionen der Vor- und Nachbereiche für die Operationen einer Algebra zugelassen. Statt dessen dürfen sich die Vor- und Nachbereiche nur auf solche formalen Objekte erstrecken, die der überlagerten Sortensprache aus der zugrundeliegenden Signatur SIG entsprechen. Daher können SIG-Algebren als sortierte funktionale Algebren[37] bezeichnet werden. Der Sortierungscharakter einer SIG-Algebra wird besonders dadurch deutlich, daß ihr Universum OB durch die Objektmengen OB_i aus der Familie OBF in sortenspezifische Teiluniversen aufgespalten wird[38]. Da die sortenspezifischen Objektmengen OB_i nicht disjunkt zu sein brauchen, können die Teiluniversen auch einander überschneiden[39].

o) Die Ausdruckskraft der hier eingeführten SIG-Algebren geht deutlich über die Ausdruckskraft derjenigen Algebren hinaus, die konventionellen Programmen des Operations Research (OR-Programmen) gewöhnlich zugrunde liegen. Erstens lassen SIG-Algebren formale Objekte mit beliebig komplexen inneren Strukturen zu. Solche internen Objektstrukturierungen sind für die formalen Objekte von OR-Programmen grundsätzlich unbekannt. Zweitens verfügen SIG-Algebren über eine deutlich höhere Vielfalt unterschiedlicher Objektmengen als OR-Programme. Denn OR-Programme beruhen im Regelfall[40] auf höchstens zwei[41] verschiedenen numerischen Objektmengen. Es handelt sich um die Objektmenge "INTEGER" aller Ganzzahlen und die Objektmenge "REAL" aller Rationalzahlen[42]. Solche Algebren werden fortan verkürzt als NUM-Algebren bezeichnet. Bei NUM-Algebren handelt es sich um Primitiv-Algebren in dem Sinne, daß sie von allen hier eingeführten SIG-Algebren als Basiskonstrukt umschlossen werden.

p) SIG-Algebren bleiben nicht auf die NUM-Algebren beschränkt. Statt dessen können die Objektmengen von SIG-Algebren aus formalen Objekten *beliebiger* Art bestehen, sofern sie die o.a. Einschränkungen sortierter funktionaler Algebren erfüllen. Diese Öffnungsklausel erschließt den SIG-Algebren außerordentlich reichhaltige Universen formaler Objekte. Die möglichen Erweiterungen von SIG-Algebren gegenüber NUM-Algebren von OR-Programmen werden anhand dreier neuartiger Objektmengen verdeutlicht. Sie spielen auch bei der späteren Definition von SIG-Interpretationen und von Synthetischen Netzen eine bedeutsame Rolle.

Hierbei handelt es sich erstens um die Objektmenge OB_{wahr_wert} für die Sorte "wahr_wert". Diese Objektmenge umfaßt diejenigen Wahrheitswerte, die in der jeweils zugrundegelegten Formallogik definiert sind. In dieser Arbeit werden nur die prädikatenlogischen Wahrheitswerte benötigt. Sie konstituieren die BOOL'sche Objektmenge "BOOL" mit $BOOL = OB_{wahr\_wert} = \{gültig, ungültig\}$[43]. Zweitens beruhen SIG-Algebren im Kontext der Theorie formaler Sprachen oftmals auf einer Sorte "zeichenfolge"[44]. Die zugeordnete Objektmenge $OB_{zeichenfolge} = STRING$ besteht aus allen Zeichenfolgen (strings)[45], die aus den Alphabeten der jeweils betrachteten formalen Sprachen erzeugt werden können. Drittens ist die Objektmenge "SYMBOL" betroffen. Sie enthält *alle* formalen Objekte, denen kein bestimmter formaler Charakter zugeordnet ist (symbolische Objekte). Der Begriff des "bestimmten formalen Charakters" ist nicht eindeutig definiert, sondern hängt vom Kontext der jeweils betrachteten formalen Konzepte ab. Im allgemeinen werden aber alle formalen Objekte, die Zahlen[46], Zeichenfolgen oder Wahrheitswerte darstellen, als formale Objekte mit numerisch bzw. zeichenhaft bzw. logisch bestimmtem Charakter angesehen. Alle formalen Objekte, die nicht in einer der drei vorgenannten Weisen bestimmt sind, werden in dieser Arbeit als symbolische Objekte aus der Menge "SYMBOL" behandelt. Diese Objektmenge eröffnet infolge ihrer Definitionsfreiheit ein unbegrenztes Universum algebraischer Objekte.

q) Die formalen Objekte, die Elemente des algebraischen Universums OB sind, stellen aus formallogischer Sicht "Individuen" dar[47]. Die Strukturierung des algebraischen Universums durch die Sorten aus der Signatur SIG bedeutet aus dieser Perspektive eine entsprechende Strukturierung des Individuenbereichs. Daher korrespondiert das hier eingeführte Konzept der SIG-Algebren mit dem Konzept "sortierter" oder "getypter" Logiken.

r) Wenn die Signatur SIG=(SO,OP) der SIG-Algebra A_{SIG} als eine strukturierte Liste nach dem früher eingeführten Schema "<Signaturname> = sorts: ... Ops: ..." notiert wird, läßt sich jede zugeordnete SIG-Algebra als eine homomorphe strukturierte Liste darstellen:

<algebraname> =

OBs: OB_1
...
OB_I

ops: op_1: $OB_{i(1.1)} \times ... \times OB_{i(1.K1)} \rightarrow OB_{i(1.K1+1)}$
$(ob_1,...,ob_{K1}) \rightarrow ob_{K1+1} = op_1(ob_1,...,ob_{K1})$
...
op_J: $OB_{i(J.1)} \times ... \times OB_{i(J.KJ)} \rightarrow OB_{i(J.KJ+1)}$
$(ob_1,...,ob_{KJ}) \rightarrow ob_{KJ+1} = op_J(ob_1,...,ob_{KJ})$

Beispiele zur Algebra-Definition:[48)]

a) Der Signatur "String", die früher als exemplarische Verdeutlichung der Signatur-Definition angeführt wurde, läßt sich die SIG-Algebra "string" zuordnen. Sie interpretiert die Sorte "zeichen" durch die Objektmenge $OB_{zeichen}=Z$ und die Sorte "zeichenfolge" durch die Objektmenge $OB_{zeichenfolge}=Z^*$ als freiem Monoid über der Zeichenmenge Z. Jedem 0-stelligen Operationssymbol (Konstantensymbol) Ko_k aus der Signatur "String" wird die Konstante $z_k \in Z$ zugeordnet. Diese Konstanten werden durch 0-stellige Operationen in Kurznotation eingeführt. Jedem anderen Operationssymbol $Op_j \in OP$ aus der Signatur SIG werden Operationen op_j zugeordnet, die Objekte aus der Trägermenge der Sorte "zeichenfolge" - einschließlich der leeren Zeichenfolge λ - erzeugen oder verknüpfen:

string =

OBs:	$OB_{zeichen} = Z$		
	$OB_{zeichenfolge} = Z^*$		
ops:	$z_1,...,z_{26}$:	$\rightarrow Z$	
	erzeuge0:	$\rightarrow Z^*$,	erzeuge0() = λ
	erzeuge1:	$Z \rightarrow Z^*$,	erzeuge1(z_k) = z_k
	verknüpfe:	$Z^* \times Z^* \rightarrow Z^*$,	verknüpfe(x,y) = xy

b) Das Konzept der SIG-Algebren ist so leistungsfähig, daß es die Menge aller kombinatorisch möglichen Netztopologien $TOP_{STN}=(S,T;F)$ für Stelle/Transition-Netze STN zu entfalten vermag. Ausgangspunkt ist eine topologiebezogene Signatur $SIG_{top}=(SO_{top},OP_{top})$[49)], die auch unter der Bezeichnung "Topologie" angesprochen wird. Sie definiert fünf Sorten: je eine Sorte für stellen- und für transitionsartige Knoten (s-Knoten bzw. t-Knoten), zwei richtungsspezifische Kantensorten und eine Sorte für Netztopologien. Der Signatur SIG_{top} wird anschließend eine SIG_{top}-Algebra $A_{top}=(OBF_{top},OPF_{top})$="topologie" zugeordnet. Dabei bedeuten OBF_{top} die Familie der sortenspezifischen Trägermengen und OPF_{top} die Operationenfamilie für die Operationssymbole aus der Signatur SIG_{top}.

$OB_{s\text{-}knoten}$ und $OB_{t\text{-}knoten}$ sind die Trägermengen aller Stellen bzw. Transitionen. Da *beliebig große* Netztopologien möglich sein sollen, werden *unendlich viele* Konstantensymbole Ks_m und Kt_n als 0-stellige Operationssymbole mit $m \in \mathcal{N}_+$ bzw. $n \in \mathcal{N}_+$ in der Signatur SIG_{top} eingeführt. Sie werden in der SIG_{top}-Algebra durch beliebig zahlreiche s-Knoten s_m (Stellen) und t-Knoten t_n (Transitionen) als zugeordnete Konstanten interpretiert[50)]. Daher werden die symbolischen Trägermengen $OB_{s\text{-}knoten}$ und $OB_{t\text{-}knoten}$ für die atomaren formalen Objekte der Stellen bzw. Transitionen in der SIG_{top}-Algebra als potentiell unendliche Mengen behandelt. Folglich ist auch die Topologienmenge $OB_{netztop}$ unbegrenzt. Zwei Operationssymbole "Kante_st" und "Kante_ts" dienen dazu, Netzkanten zu generieren. Durch die beiden kombinatorisch möglichen Anordnungen der zwei definierten Knotensorten werden zwei Kantenrichtungen festgelegt. Zwei Operationssymbole "Erzeuge_st" und "Erzeuge_ts" dienen als Induktionsbasis. Ihre zugeordneten Erzeugungsoperationen "erzeuge_st" bzw. "erzeuge_ts" bringen in der SIG_{top}-Algebra die beiden Netztopologien top_{st} bzw. top_{ts} hervor. Diese Netztopologien stellen die kleinsten zulässigen[51)] Topologien für Stelle/Transition-Netze dar. Als komplexe formale Objekte besitzen sie jeweils eine innere Struktur aus zwei sortenverschiedenen Knoten und einer Kante. Durch sechs Ergänzungsoperationen werden bereits bestehende Netztopologien jeweils um eine Kante oder um eine Kombination aus einer Kante und aus einem neuen adjazenten Knoten erweitert. Aufgrund dieser Festlegungen wird die Menge $OB_{netztop}$ aller zulässigen Topologien von Stelle/Transition-Netzen durch die Signatur SIG_{top} und ihre SIG_{top}-Algebra auf folgende induktive Weise definiert:

Signatur $SIG_{top}=(SO_{top},OP_{top})$="Topologie":

<u>Topologie</u> =

<u>sorts</u>: s-knoten
t-knoten
st-kante
ts-kante
netztop

<u>Ops</u>:

$Ks_1, ..., Ks_m, ...$:	$\rightarrow$ s-knoten
$Kt_1, ..., Kt_n, ...$:	$\rightarrow$ t-knoten
Kante_st:	s-knoten t-knoten $\rightarrow$ st-kante
Kante_ts:	s-knoten t-knoten $\rightarrow$ ts-kante
Erzeuge_st:	s-knoten t-knoten st-kante $\rightarrow$ netztop
Erzeuge_ts:	s-knoten t-knoten ts-kante $\rightarrow$ netztop
Ergänze_st:	netztop st-kante $\rightarrow$ netztop
Ergänze_ts:	netztop ts-kante $\rightarrow$ netztop
Ergänze_s/st:	netztop s-knoten st-kante $\rightarrow$ netztop
Ergänze_s/ts:	netztop s-knoten ts-kante $\rightarrow$ netztop
Ergänze_t/ts:	netztop t-knoten ts-kante $\rightarrow$ netztop
Ergänze_t/st:	netztop t-knoten st-kante $\rightarrow$ netztop

SIG_{top}-Algebra $A_{top} = (OBF_{top},OPF_{top})$ = "topologie"[52]:

<u>topologie</u> =

<u>OBs</u>: $OB_{s\text{-knoten}}$
$OB_{t\text{-knoten}}$
$OB_{st\text{-kante}}$
$OB_{ts\text{-kante}}$
$OB_{netztop}$

<u>ops</u>:

$s_1, ..., s_m, ...$:	$\rightarrow OB_{s\text{-knoten}}$
$t_1, ..., t_n, ...$:	$\rightarrow OB_{t\text{-knoten}}$
kante_st:	$OB_{s\text{-knoten}} \times OB_{t\text{-knoten}} \rightarrow OB_{st\text{-kante}}$
	$(s_m,t_n) \rightarrow \text{kante\_st}(s_m,t_n) = (s_m,t_n)$
kante_ts:	$OB_{s\text{-knoten}} \times OB_{t\text{-knoten}} \rightarrow OB_{ts\text{-kante}}$
	$(s_m,t_n) \rightarrow \text{kante\_ts}(s_m,t_n) = (t_n,s_m)$
erzeuge_st:	$OB_{s\text{-knoten}} \times OB_{t\text{-knoten}} \times OB_{st\text{-kante}} \rightarrow OB_{netztop}$
	$(s_m,t_n,(s_m,t_n)) \rightarrow (\{s_m\},\{t_n\},\{(s_m,t_n)\})$

erzeuge_ts: $OB_{s\text{-knoten}} \times OB_{t\text{-knoten}} \times OB_{ts\text{-kante}} \rightarrow OB_{netztop}$
$(s_m,t_n,(t_n,s_m)) \rightarrow (\{s_m\},\{t_n\},\{(t_n,s_m)\})$

ergänze_st: $OB_{netztop} \times OB_{st\text{-kante}} \rightarrow OB_{netztop}$
$((S,T;F),(s_m,t_n)) \rightarrow (S,T;F \cup \{(s_m,t_n)\})$; falls $s_m \in S \wedge t_n \in T$

ergänze_ts: $OB_{netztop} \times OB_{ts\text{-kante}} \rightarrow OB_{netztop}$
$((S,T;F),(t_n,s_m)) \rightarrow (S,T;F \cup \{(t_n,s_m)\})$; falls $s_m \in S \wedge t_n \in T$

ergänze_s/st: $OB_{netztop} \times OB_{s\text{-knoten}} \times OB_{st\text{-kante}} \rightarrow OB_{netztop}$
$((S,T;F),s_m,(s_m,t_n)) \rightarrow (S \cup \{s_m\},T;F \cup \{(s_m,t_n)\})$; falls $s_m \notin S \wedge t_n \in T$

ergänze_s/ts: $OB_{netztop} \times OB_{s\text{-knoten}} \times OB_{ts\text{-kante}} \rightarrow OB_{netztop}$
$((S,T;F),s_m,(t_n,s_m)) \rightarrow (S \cup \{s_m\},T;F \cup \{(t_n,s_m)\})$; falls $s_m \notin S \wedge t_n \in T$

ergänze_t/ts: $OB_{netztop} \times OB_{t\text{-knoten}} \times OB_{ts\text{-kante}} \rightarrow OB_{netztop}$
$((S,T;F),t_n,(t_n,s_m)) \rightarrow (S,T \cup \{t_n\};F \cup \{(t_n,s_m)\})$; falls $s_m \in S \wedge t_n \notin T$

ergänze_t/st: $OB_{netztop} \times OB_{t\text{-knoten}} \times OB_{st\text{-kante}} \rightarrow OB_{netztop}$
$((S,T;F),t_n,(s_m,t_n)) \rightarrow (S,T \cup \{t_n\};F \cup \{(s_m,t_n)\})$; falls $s_m \in S \wedge t_n \notin T$

Die Menge $OB_{netztop}$ umfaßt alle Netztopologien $TOP_{STN}=(S,T;F)$, die für Stelle/Transition-Netze überhaupt gebildet werden können. Dabei werden die Stellen- und die Transitionenmengen S bzw. T jeweils mit einer konkret bestimmten Objektmenge $OB_{s\text{-knoten}}$ bzw. $OB_{t\text{-knoten}}$ identifiziert. Ebenso werden die Kantenmengen F jeweils mit der Vereinigung zweier konkret bestimmter Objektmengen $OB_{st\text{-kanten}}$ und $OB_{ts\text{-kanten}}$ identifiziert.

Die drei Integritätsbedingungen IB_D, IB_E und IB_V, die von jedem Petrinetz erfüllt werden müssen, werden von der SIG_{top}-Algebra A_{top} eingehalten, sofern die beiden Trägermengen $OB_{s\text{-knoten}}$ und $OB_{t\text{-knoten}}$ für atomare formale Objekte disjunkt sind. Diese Anforderung ist durch die unterschiedlichen Konstanten s_m und t_n erfüllt. Die Gültigkeit der Disjunktheitsbedingung IB_D: $S \cap T = \emptyset$ folgt unmittelbar aus den Identifizierungen $S=OB_{s\text{-knoten}}$ und $T=OB_{t\text{-knoten}}$ sowie aus der Disjunktheit der involvierten Objektmengen. Die Einhaltung der Existenzbedingung IB_E: $S \cup T \neq \emptyset$ ergibt sich daraus, daß die Trägermengen $OB_{s\text{-knoten}}$ und $OB_{t\text{-knoten}}$ jeweils als nichtleere, ja sogar potentiell unendliche Mengen definiert wurden. Die Verknüpftheitsbedingung IB_V: $S \cup T = VB(F) \cup NB(F)$ wird per constructionem immer erfüllt. Denn die beiden kleinstmöglichen Netztopologien werden explizit als verknüpfte Knotenpaare eingeführt. Die hieraus induktiv abgeleiteten komplexeren Netztopologien ergänzen bereits vorliegende Netztopologien nur entweder um zusätzlich verknüpfende Kanten. Oder es wird jeweils ein neuer Knoten hinzugefügt, der aufgrund der Definitionen der vier letzten "ergänze_..."-Operationen stets durch eine Kante mit schon vorhandenen Knoten verknüpft wird.

In einem sehr weit gefaßten Sinn[53)] kann die SIG_{top}-Algebra $A_{top}=(OBF_{top},OPF_{top})$ als die Syntax einer vollständig formalisierten topologischen Netzsprache TNS aufgefaßt werden. Die Vereinigungsmenge der beiden Konstantenmenge $OB_{s\text{-knoten}}$ und $OB_{t\text{-knoten}}$ ist das Alphabet dieser Netzsprache. Die Kanten-, Erzeugungs- und Ergänzungsoperationen stellen Formierungsregeln dar. Mit ihrer Hilfe lassen sich alle zulässigen Ausdrücke (Worte) der Netzsprache bilden, die in der ausgezeichneten Menge $OB_{netztop}$ zusammengefaßt werden. Andere Elemente als diese Netzworte umfaßt die Topologienmenge $OB_{netztop}$ nicht. Daher gilt $TNS=OB_{netztop}$. Jedes Wort aus der Netzsprache TNS stellt eine zulässige Netztopologie $TOP_{STN}=(S,T;F)$ für ein Stelle/Transition-Netz STN dar. Da sich die Topologien von Petrinetzen aller Netzklassen als Topologien von Stelle/Transition-Netzen darstellen lassen, gilt allgemein: Die oben vorgestellte SIG_{top}-Algebra

A_{top} erzeugt durch ihre topologische Netzsprache TNS die Gesamtheit aller Netztopologien, die im Rahmen des Petrinetz-Konzepts definiert sind.

Über diese Generatoreigenschaft wird einerseits eine Brücke zwischen dem Signaturkonzept und den früher dargestellten Stelle/Transition-Netzen geschlagen. Die Topologien solcher Netze werden als formale Objekte aus einer speziellen SIG-Algebra - der SIG_{top}-Algebra A_{top} - erklärt. Alle anderen Komponenten von Stelle/Transition-Netzen STN, die nicht schon in ihren Netztopologien TOP_{STN} enthalten sind, lassen sich als abstrakte Beschriftungen der topologischen Elemente - der Stellen, Transitionen oder gerichteten Kanten - einführen. Diese Beschriftungen stammen aus formalen Sprachen, die ihrerseits aus anderen Signaturen hergeleitet werden können[54)]. Daher kann das gesamte Konzept der Stelle/Transition-Netze als ein algebraisches Konstrukt aufgefaßt werden, das auf der Basis der Signatur SIG_{top} und ihrer SIG_{top}-Algebra errichtet ist. Damit wird die originär arithmetische Konstruktion von Stelle/Transition-Netzen in das allgemeinere und abstraktere Konstruktionskonzept sortierter Algebren eingebettet.

Andererseits wird durch die Signatur SIG_{top} und ihre SIG_{top}-Algebra ebenso ein Bezug zu den später eingeführten Synthetischen Netzen hergestellt. Denn diese besitzen denselben topologischen Netzkern $TOP_{STN} = (S,T;F)$ wie Stelle/Transition-Netze. Daher läßt sich das Signaturkonzept auch als eine algebraische Schnittstelle zwischen Stelle/Transition-Netzen und Synthetischen Netzen auffassen.

Anmerkungen zum Kapitel:

1) Vgl. EHRIG (1985a), S. 25f.; RECK (1988), S. 48; BEIERLE (1988a), S. 449.

2) Vgl. EHRIG (1985a), S. 16; RECK (1988), S. 43f.; REISIG (1989a), S. 5.

3) Mengenfamilien stellen geordnete Tupel dar, deren Komponenten jeweils Mengen sind. Die Tupelkomponenten werden als Mitglieder der Mengenfamilie bezeichnet. Eine Mengenfamilie kann dieselbe Menge mehrfach enthalten. Im Gegensatz zur Vereinigungsmenge derselben Mitglieder gehen in einer Mengenfamilie keine Informationen über die Extensionen der beteiligten Mitgliedsmengen verloren.

4) Da eine Mengenfamilie dieselbe Menge mehrfach enthalten darf, können verschiedenen Sorten dennoch dieselben Objektmengen zugeordnet sein.

5) Operationenfamilien stellen geordnete Tupel dar, deren Komponenten jeweils Operationen sind. Operationenfamilien entsprechen ansonsten den voranstehend definierten Mengenfamilien.

6) Vgl. dazu beispielsweise die Ausführungen zu den Übergangsmodi (tr_v,vb_c) von Übergangsoperationen tr_v. Bei unendlichen Objektmengen könnten für eine Übergangsoperation unendlich viele Übergangsmodi existieren. Diese Übergangsmodi werden später als Schaltmodi von Transitionen in Synthetischen Netzen gedeutet. Um herauszufinden, ob eine Transition unter einer Netzmarkierung aktiviert ist, müßten dann widrigstenfalls unendlich viele Schaltmodi untersucht werden. Dies ist im allgemeinen - wenn keine besonderen vereinfachenden Umstände hinzukommen - mit endlichen Informationsverarbeitungsressourcen aber nicht möglich.

7) Automatische Informationsverarbeitungssysteme müssen keineswegs auf endliche Mengen eingeschränkt werden. Aber für den Umgang mit echten Unendlichkeiten sind aufwendige und unkonventionelle Konstruktionen - wie etwa symbolische Formelmanipulationen - erforderlich. Diese werden bei der Implementierung von Netzmodellen nicht herangezogen. Die konventionell definierten Objektmengen von Automatischen Informationsverarbeitungssystemen besitzen dagegen endlichen Charakter, auch wenn sie mitunter auf den ersten Blick als unendliche Mengen erscheinen. Vgl. dazu die Ausführungen hinsichtlich der Endlichkeit der scheinbar unendlichen Menge "REAL", welche die reellen Zahlen umfassen soll. Es wird dort aufgezeigt, daß diese Menge bei Automatischen Informationsverarbeitungssystemen im Regelfall nur endlich viele Rationalzahlen umfaßt.

8) Unter einer Operation ohne qualifizierenden Zusatz wird stets eine K_j-stellige Operation mit $K_j \in \mathcal{N}_+$ verstanden.

9) Daher gelten die nachfolgend angeführten terminologischen Festlegungen für Funktionen analog. Beispielsweise lassen sich anstelle der Begriffe Operator und Operatorapplikation ebenso die Bezeichnungen Funktor bzw. Funktorapplikation verwenden.

10) Unter einer K_j-stelligen Abbildung mit $K_j \in \mathcal{N}_+$ wird hier eine K_j+1-stellige Relation REL_j verstanden, für die gilt: Die Relation REL_j stellt eine Teilmenge des kartesischen Produkts der Trägermengen OB_k mit $k \in \{1,...,K_j+1\}$ dar: $REL_j \subseteq OB_1 x ... x OB_{Kj+1}$. Die Relation kann daher synonym als Relationsmenge REL_j angesprochen werden. Bei den Elementen jeder Trägermenge OB_k handelt es sich vorerst um formale Objekte ob_k. Sie werden später zu Termen te_k verallgemeinert. Hinzu kommt als charakteristische Abbildungseigenschaft: Die relationszugehörigen K_j+1-Tupel $(ob_1,...,ob_{Kj+1})$ besitzen eine innere Struktur derart, daß zwischen dem Teiltupel $(ob_1,...,ob_{Kj})$ und der letzen Tupelkomponente ob_{Kj+1} differenziert wird. Erstes heißt ein Argument oder Original, zweite ein Bild oder Wert der Abbildung. Entsprechend wird das kartesische Teilprodukt $OB_1x...xOB_{Kj}$ von der Menge OB_{Kj+1} unterschieden. Erstes stellt den Vorbereich (domain) VB_j, zweites den Nachbereich (codomain) NB_j der Abbildung REL_j dar. Die Vereinigungsmenge von Vor- und Nachbereich heißt das Feld (field) der Abbildung. Jedes Element $(ob_1,...,ob_{Kj})$ aus dem Vorbereich heißt ein K_j-stelliges Argument der Abbildung. Die Menge AM_j aller Argumente (Argumentemenge) einer Abbildung ist immer eine - echte oder unechte - Teilmenge ihres Vorbereichs: $AM_j \subseteq VB_j$. Die Menge BM_j aller Bilder (Bildmenge) stellt immer eine - abermals echte oder unechte - Teilmenge des Nachbereichs der Abbildung dar: $BM_j \subseteq NB_j$.
Eine Abbildung REL_j wird eine Funktion genannt, sofern sie sich rechtseindeutig verhält. Rechtseindeutigkeit bedeutet, daß die Abbildung jedem Element ihres Vorbereichs höchstens ein Element aus ihrem Nachbereich zuordnet:

$$(ob_1,...,ob_{Kj},ob_{Kj+1.1}) \in REL_j \wedge (ob_1,...,ob_{Kj},ob_{Kj+1.2}) \in REL_j$$

$$\Rightarrow \quad ob_{Kj+1.1} = ob_{Kj+1.2}$$

oder äquivalent:

$$\forall(ob_1 \in DB_1) \ldots \forall(ob_{Kj+1} \in DB_{Kj+1}) : \#(\{ob_{Kj+1} : (ob_1,...,ob_{Kj+1}) \in REL_j\}) \leq 1$$

Rechtseindeutige Abbildungen (Funktionen) werden gewöhnlich nicht als ausgezeichnete Relationen oder Relationsmengen REL_j behandelt, sondern als formale Konstrukte sui generis. Eine Funktion REL_j erhält dann einen eigenständigen Funktionsnamen f_j, der stets mit Kleinbuchstaben notiert wird. Es wird auch kurz von einer Funktion

f_j gesprochen. Jede Funktion besitzt als rechtseindeutige Abbildung die Eigenschaft, daß jedem Argument genau ein Bild zugeordnet wird. Gleiche Argumente besitzen also stets gleiche Bilder. Die Zuordnung zwischen Argumenten und Bildern wird mit der Hilfe eines Funktionsoperators - oder kurz Funktors - $f_j(...)$ ausgedrückt. $f_j(ob_1,...,ob_{Kj})$ stellt die Anwendung des Funktionsoperators $f_j(...)$ auf das Funktionsargument $(ob_1,...,ob_{Kj})$ dar. Die Zuordnung zwischen Funktionsbildern und -argumenten nimmt die allgemeine Form $ob_{Kj+1}=f_j(ob_1,...,ob_{Kj})$ an. Die Abbildungs- oder Funktionsvorschrift definiert eindeutig, welche konkrete Gestalt diese Zuordnung für eine bestimmte Funktion annimmt. Die Funktionsvorschrift kann explizit oder implizit angegeben werden. Im ersten - weithin vorherrschenden - Fall wird die Funktionsvorschrift durch einen formalsprachlichen Ausdruck "vorschrift" spezifiziert, der angibt, wie das Objekt ob_{Kj+1} aus bekannten Objekten $ob_1,...,ob_{Kj}$ ermittelt wird. Die Funktionsvorschrift wird dagegen implizit definiert durch die Menge F_j aller (K_j+1)-Tupel $(ob_1,...,ob_K,ob_{Kj+1})$, welche die - nicht explizit genannte - Funktionsvorschrift erfüllen. Diese Erfüllungsmenge F_j fällt notwendig mit der Relationsmenge REL_j zusammen, die der Funktion f_j als rechtseindeutige Abbildung zugrundeliegt: $F_j=REL_j$. Alle vorgenannten Konstituenten der Definition einer Funktion f_j werden in ihrer Deklaration zusammengefaßt. Die Deklaration einer Funktion f_j nimmt bei explizit notierter Funktionsvorschrift folgende Gestalt an:

$$f_j: \quad OB_1 \times ... \times OB_{Kj} \rightarrow OB_{Kj+1}$$
$$(ob_1,...,ob_{Kj}) \rightarrow ob_{Kj+1} = f_j(ob_1,...,ob_{Kj}) = \text{<vorschrift>}$$

Bei implizit notierter Funktionsvorschrift gilt dagegen:

$$f_j: \quad OB_1 \times ... \times OB_{Kj} \rightarrow OB_{Kj+1}$$
$$(ob_1,...,ob_{Kj}) \rightarrow ob_{Kj+1} = f_j(ob_1,...,ob_{Kj})$$

mit:

$$f_j(ob_1,...,ob_{Kj}) = ob_{Kj+1} \Leftrightarrow (ob_1,...,ob_K,ob_{Kj+1}) \in F_j$$

Vgl. zur Definition rechtseindeutiger Abbildungen (Funktionen) ROGERS,H. (1967), S. XVI; STEGMÜLLER (1970a), S. 44f.; KANITSCHEIDER (1971), S. 68; KUYPERS (1973), S. 97 u. 110; SCHNORR (1974), S. 12; GENRICH (1976b), S. 30; STEGMÜLLER (1984b), S. 38f.
Die Rechtseindeutigkeit schließt auch den Fall ein, daß mindestens einem Vorbereichselement überhaupt kein Nachbereichselement zugeordnet wird. Wenn mit $AM_j \subseteq VB_j$ dieser Fall zugelassen wird, liegt eine *partielle* Funktion vor; vgl. KLEENE (1952), S. 325; ROGERS,H. (1967), S. XVI; KELLER,R. (1972a), S. 3; SCHNORR (1974), S. 12; RIEDEMANN (1979), S. 2. Dann darf die Argumentemenge der Abbildung eine echte Teilmenge ihres Vorbereichs sein ($OM_j \subset VB_j$), muß es aber nicht. (Abweichender Ansicht ist STEGMÜLLER (1984b), S. 39, mit der Forderung, es müsse stets eine echte Teilmenge vorliegen.) Andernfalls, wenn Argumentemenge und Vorbereich der Abbildung f_j wegen $AM_j=VB_j$ identisch sein müssen, wird von einer *totalen* oder *vollständigen* Funktion gesprochen. Bei einer solchen vollständigen Abbildung wird jedem Vorbereichselement genau ein Nachbereichselement zugeordnet; vgl. ROGERS,H. (1967), S. XVI; KANITSCHEIDER (1971), S. 69; SCHNORR (1974), S. 12; RIEDEMANN (1979), S. 2; STEGMÜLLER (1984b), S. 39.
Die Deklaration einer partiellen Funktion f_j läßt sich durch den Zusatz "; falls $prä_j(te_1,...,te_{Kj})$" präzisieren. Dabei gibt die abbildungsspezifische Formel $for_j(ob_1,...,ob_{Kj})$ den Geltungsbereich der Funktion an. Er umfaßt alle Argumente aus dem Vorbereich der Funktion, welche die Formel erfüllen:

$$f_j: \quad OB_1 \times ... \times OB_{Kj} \rightarrow OB_{Kj+1}$$
$$(ob_1,...,ob_{Kj}) \rightarrow ob_{Kj+1} = f_j(ob_1,...,ob_K) = \text{<vorschrift>}; \quad \text{falls } for_j(ob_1,...,ob_{Kj})$$

Wegen $AM_j \subseteq VB_j$ für partielle und $AM_j=VB_j$ für vollständige Funktionen bilden partielle Funktionen das allgemeinste Funktionskonzept. Vollständige Funktionen sind ein Sonderfall davon. Fortan werden vollständige Funktionen vorausgesetzt, sofern keine ausdrücklich abweichenden Festlegungen erfolgen.
Wenn eine rechtseindeutige Abbildung zusätzlich die Eigenschaft der Linkseindeutigkeit erfüllt, dann entspricht jedem ihrer Bilder genau ein Original. In diesem Fall liegt eine eineindeutige Abbildung oder *injektive* Funktion vor. Bei ihr besitzen gleiche Bilder stets dieselben Originale.
Die Bildmenge (der Wertebereich) einer Abbildung ist immer eine - echte oder unechte - Teilmenge ihres Nachbereichs. Falls die Bildmenge und der Nachbereich einer rechtseindeutigen Abbildung gleich sind, wird von einer *surjektiven* Funktion gesprochen. Dann ist jedes Nachbereichselement zugleich auch ein Bild der Funktion. Der Verf. folgt hier der Terminologie von LEBNER (1980), S. 38. Dagegen teilt er nicht die Festlegung von DINKELBACH (1973), S. 158, den Abbildungsbegriff nur auf den Sonderfall surjektiver Funktionen zu beziehen.
Eine Funktion heißt bijektiv, wenn sie sowohl in- als auch surjektiv ist. Dies bedeutet, daß jedem ihrer Nachbereichselemente genau ein Original entspricht und der Nachbereich mit der Bildmenge zusammenfällt. Eine bijektive Funktion ist daher immer eineindeutig und schöpft ihren Nachbereich durch Bilder vollständig aus.

11) Die Nachbereiche aller Operationen op_j einer SIG-Algebra werden explizit als formale Objektmengen definiert. Denn das zugehörige Operationssymbol Op_j besitzt in der zugrundeliegenden Signatur SIG=(SO,OP) die Zielsorte $sort_{i(j.Kj+1)} \in SO$. Diese Sorte wird in der SIG-Algebra A_{SIG}=(OBF,OPF) durch die Objektmenge $OB_{i(j.Kj+1)}$ interpretiert. Diese Objektmenge ist per constructionem der Nachbereich der Operation op_j. Hierdurch unterscheiden sich die hier vorgestellten SIG-Algebren von der später eingeführten prädikatenlogischen Implementierung mittels des Turbo-PROLOG-Dialekts. Dort werden die Nachbereiche von Operationen und somit die Zielsorten der zugrundeliegenden Operationssymbole nur implizit definiert; vgl. KINNEBROCK (1988), passim, z.B. S. 44.

12) Falls dieser Regelfall besonders hervorgehoben werden soll, wird von einer vollständig definierten - oder kurz: vollständigen - Operation gesprochen.

13) Dieser Sachverhalt kann durch die Notation "$\rightarrow_{part}$" in der Deklaration der Operation op_j formal verdeutlicht werden. Diejenigen Fälle, in denen die Operation op_j auf Tupel ihres Vorbereichs angewendet werden darf, werden durch eine "falls..."-Klausel am Deklarationsende spezifiziert. Vgl. dazu die Erläuterungen partieller Funktionen.

14) Die Operatoranwendungen $op_j()$ stellen wegen $op_j()=op_j(\lambda)$ jeweils Applikationen des Operators $op_j(...)$ auf das leere Argument λ dar. Bei dem zugeordneten formalen Objekt "ob" kann es sich z.B. um den Ausdruck "nil" handeln. Er wird später z.B. benutzt, um nicht-definierte Attributausprägungen oder um nicht vorhandene Transportobjekte zu vertreten.

15) Diese algebraischen Konstanten werden in mathematischen oder modelltheoretischen Argumentationskontexten oftmals als Parameter bezeichnet. Vgl. zu diesem Parameterbegriff z.B. KOSIOL (1961a), S. 332; VON WEIZSÄCKER (1985), S. 225 u. 229; MALIK (1986), S. 117; MÜLLER,A. (1987), S. 367f. (hinsichtlich ihres Festlegungsarguments); MERTENS (1988e), S. 9, sowie die frühere Erläuterung parametrischer Modelle. Mitunter wird der Parameterbegriff auch im Sinne von Variablen gebraucht; vgl. z.B. WOLLNIK (1986), S. 126; MÜLLER,A. (1987), S. 367f. (bezüglich ihres externen Variationsarguments). Der Verf. vermeidet diese schwankende Terminologie, indem er Parameter mit Konstantencharakter von vornherein als Konstanten bezeichnet. Parameter, die von den vorgenannten Quellen als Variablen behandelt werden, stellen dagegen hier Konstantensymbole dar. Gegenüber den vorgenannten Variablen besteht nur eine scheinbare Abweichung, weil in dem dort zugrundeliegenden Variablenbegriff Konstantensymbole und Variablen unreflektiert vermengt werden. (Der präzise Variablenbegriff wird in dieser Arbeit im Zusammenhang mit dem Termkonzept eingeführt.) Aufgrund der voranstehenden Klärungen entspricht die "Parametrisierung" eines konkreten Modells der Ersetzung einer Teilmenge seiner Konstanten durch Konstantensymbole. Umgekehrt erfolgt die Konkretisierung eines parametrisierten Modells u.a. durch die algebraische Interpretation seiner Konstantensymbole, bei der alle Konstantensymbole in der o.a. Weise durch Konstanten ersetzt werden.

16) Vgl. z.B. die Konstantendefinitionen bei RECK (1988), S. 86 u. 138; REISIG (1989a), S. 7 i.V.m. S. 14. Sie werden nicht als Elemente spezieller Objektmengen eingeführt, sondern stellen jeweils die Bilder von 0-stelligen Operationen dar. Sie interpretieren hierdurch die zugrundeliegenden 0-stelligen Operationssymbole (Konstantensymbole).

17) Dies entspricht der Syntaxfestlegung des Turbo-PROLOG-Dialekts, in seiner "domains"-Sektion den Sorten nur sechs ausgezeichnete allgemeine Objektmengen zuordnen zu können. Hierzu gehören z.B. die Mengen aller ganzen oder reellen Zahlen (innerhalb eines Intervalls) oder die Menge aller Zeichenfolgen aus einem weit definierten Alphabet. Vgl. zu diesen zulässigen Objektmengendeklarationen PROLOG (o.J.), S. 34ff.; TOWNSEND,C. (1987), S. 37; KINNEBROCK (1988), S. 30ff. u. 128.

18) Beispielsweise werden in der unten vorgestellten SIG-Algebra "string" die 26 zulässigen alphabetischen Zeichen als 0-stellige Operationen (Konstanten) z_k mit $k \in \{1,...,26\}$ eingeführt. Die Objektmenge $OB_{zeichen}$ für die Zielsorte "zeichen" der 0-stelligen Operationssymbole, die in der zugrundeliegenden Signatur "String" mit den Operationen z_k korrespondieren, wird dagegen nur als allgemeine Menge $OB_{zeichen}=Z$ definiert. Zugleich wird für die Konstanten z_k mit $k \in \{1,...,26\}$ die Kurznotation $z_1,...,z_{26}: \rightarrow Z$ mit $Z=OB_i$ gewählt. In dieser Darstellungsweise tritt der Operationscharakter der Konstantendefinition zugunsten ihrer Gleichsetzung mit formalen Objekten zurück. Statt dessen hätte aber auch die Objektmenge $OB_{zeichen}$ von vornherein durch die Konstanten z_k als $OB_{zeichen}=\{z_1,...,z_{26}\}$ festgelegt werden können. Diese Alternative ist aber in der Literatur zum Signaturkonzept unüblich, bei Anwendung der Programmiersprache Turbo-PROLOG sogar unzulässig. Gleiches gilt für die Knoten von Stelle/Transition-Netz-Topologien, die als 0-stellige Operationen und nicht als Elemente entsprechender Objektmengen eingeführt werden.

19) Allgemein läßt sich jede spezielle Objektmenge $OB_i=\{ob_1,...,ob_{Ki}\}$ aus endlich vielen formalen Objekten (Konstanten) ob_k mit $k \in \{1,...,K_i\}$ und $K_i \in \mathcal{N}_+$ einführen als Auflistung von K_i 0-stelligen Operationen $op_{j(k)}$. Dann gilt für die Konstantendefinition:

$op_{j(1)}$: $\rightarrow OB_i$, $op_{j(1)}() = ob_1$

...

$op_{j(Ki)}$: $\rightarrow OB_i$, $op_{j(Ki)}() = ob_{Ki}$

oder in Kurznotation:

$ob_1,...,ob_{Ki}$: $\rightarrow OB_i$.

20) Es ist aber auch weiterhin möglich, eine Objektmenge zunächst in der Mengenfamilie OBF durch eine Menge allgemein definierter Objekte zu spezifizieren. Erst später werden die Objekte aus der Objektmenge auf eine echte Teilmenge von speziell definierten Objekten eingeschränkt, indem entsprechende Operationen für originäre oder derivative Objektdefinitionen ergänzt werden. In diesem Fall wird vereinbart, daß die formalen Objekte nur aus der zuletzt eingeführten echten Teilmenge der speziell definierten Objekte stammen dürfen. Die zunächst spezifizierte Menge allgemein definierter Objekte besitzt dagegen nur den Charakter einer echten Obermenge.
Die voranstehend erläuterte, zweistufige Objektmengendefinition ist erforderlich, um die spätere Implementierung von algebraisch spezifizierten Netzmodellen vorzubereiten. Denn die dort verwendete Programmiersprache Turbo-PROLOG erfordert es, in der "domains"-Sektion die Objektmenge jeder originär definierten Sorte mit einer der sechs allgemeinen Objektmengen zu identifizieren, die bereits in einer früheren Anmerkung angesprochen wurden. Dies läuft speziell definierten Objektmengen insofern zuwider, als diese Objektmengen nicht mit jenen allgemeinen Objektmengen zusammenfallen. Vielmehr sind die erstgenannten echte Teilmengen der letztgenannten. Dieser Sachverhalt wird etwa beim unten angeführten Beispiel der "string"-Algebra deutlich. Ihre beiden Objektmengen Z und Z* sind in spezieller Weise definiert. In der "domains"-Sektion der Programmiersprache Turbo-PROLOG müssen sie aber mit allgemeinen Objektmengen identifiziert werden. Die allgemeine Objektmenge "STRING" läßt aber für die Objektmenge "Z*" weit mehr Zeichenfolgen zu, als in der u.a. "string"-Algebra aus den Konstanten "z_k" kombiniert werden können. Daher wäre die PROLOG-typische Identifizierung "Z*=STRING" in der "domains"-Sektion strenggenommen unzutreffend. Statt dessen gilt korrekt: $Z^* \subset STRING$. Allerdings wirkt sich dieser Fehler in PROLOG-Implementierungen von Netzmodellen nicht aus. Denn das Inferenzkonzept der Programmiersprache PROLOG gestattet nur solche Zeichenfolgen abzuleiten, die aus den Konstanten "z_k" zusammengesetzt sind. Wegen dieser Unwirksamkeit der fehlerhaften Objektmengenfestlegungen in PROLOG-Programmen wird hierauf nachfolgend nicht weiter eingegangen. In implementierungsbezogenen Ausführungen wird daher die PROLOG-typische Gleichsetzung der Objektmengen mit einer der sechs allgemeinen Objektmengen auch dann zugelassen, wenn die erstgenannten Objektmengen später auf die Bildmengen von Operationen in spezieller Weise eingeschränkt werden. Als Objektmenge gilt dann jeweils nur diejenige Menge von Objekten, die mit Hilfe der zugehörigen Operationen durch Inferenzen der Programmiersprache PROLOG tatsächlich abgeleitet werden können.

21) Bei der multiplen Sorteninterpretation durch mehrere Objektmengen handelt es sich entweder um verschiedene intensionale Beschreibungen extensionsgleicher Mengen. Diese extensionsgleiche Mengen sind per definitionem identisch. Oder die Extensionen der Objektmengen, welche dieselbe Sorte interpretieren, sind nicht identisch. Dann wird die betroffene Sorte durch die Vereinigungsmenge aller definitionsbeteiligten Objektmengen interpretiert.
Eine multiple Sorteninterpretation liegt in folgendem verdeutlichenden Beispiel in zweifacher Hinsicht vor. Erstens wird die betrachtete Sorte $sort_2$ sowohl originär durch die Objektmenge OB_2 der natürlichen Zahlen als auch derivativ durch die Bildmengen der Operationen op_1 und op_2 interpretiert. Im derivativen Fall ist die Sorte $sort_2$ die Zielsorte zweier unterschiedlicher Operationssymbole Op_1 und Op_2, denen die beiden Operationen op_1 bzw. op_2 zugeordnet sind. Die Operationen op_1 und op_2 definieren zwei intensional verschiedene Bildmengen. Dies ist der zweite Aspekt der hier vorliegenden multiplen Sorteninterpretation. Allerdings handelt es sich um extensional identische Bildmengen, weil in beiden Fällen die Menge aller geraden natürlichen Zahlen derivativ definiert werden. Wenn für die Deklaration der Operation op_2 eine partielle Operation mit "$mod_2(...)$" als Operator der Restklassenoperation "modulo 2" gewählt wird, gilt für dieses Beispiel:

Signatur SIG =

sorts: $sort_1$
$sort_2$

Ops: Op_1: $sort_1 \rightarrow sort_2$
Op_2: $sort_1 \rightarrow sort_2$

SIG-Algebra =

Obs: $OB_1 = \mathcal{N}_+$
$OB_2 = \mathcal{N}_+$

ops: op_1: $OB_1 \rightarrow OB_2$
$n \rightarrow op_1(n) = 2n$

op_2: $OB_1 \rightarrow OB_2$
$n \rightarrow op_2(n) = n$; falls $mod_2(n) = 0$

Extensionsverschiedene Objektmengen liegen dagegen für die derivative Interpretation derselben Sorte oftmals dann vor, wenn diese Sorte - die Referenzsorte - in der zugrundeliegenden Signatur aus Subsorten aufgebaut wird. Eine Subsorte ist das 1-stellige Argument eines Operationssymbols, dessen Zielsorte die Referenzsorte ist. Zusätzlich müssen in der zugehörigen Algebra die Operationssymbole aller Subsorten durch die identische Operation interpretiert werden.

Ein besonders anschauliches, vom Verf. ausgebautes Beispiel hierfür findet sich bei KINNEBROCK (1988), S. 44. Da sich das Beispiel auf die Programmiersprache PROLOG bezieht, wird jede Objektmenge zunächst mit einer der sechs PROLOG-spezifischen allgemeinen Objektmengen gleichgesetzt. Später werden diese Objektmengen jedoch durch die Bildmengen von Operationen in spezieller Weise definiert. Die Sorte "zeitpunkt" wird aus den Subsorten "absoluter_zeitpunkt" und "relativer_zeitpunkt" aufgebaut. Beide Subsorten sind Argumente zweier Operationssymbole "Abs_zeitpunkt" und "Rel_zeitpunkt", die in ihrer Zielsorte "zeitpunkt" übereinstimmen. Die erste Subsorte "absoluter_zeitpunkt" ist die Zielsorte des Operationssymbols "Datum", dessen Argument aus den Sorten "tag", "monat" und "jahr" besteht. Dem Operationssymbol "Datum" ist die Operation "datum" zugeordnet. Ihr Vorbereich ist die Menge aller geordneten 3-Tupel (tt,mm,jj), deren Komponenten jeweils einen Term für die 2-ziffrige numerische Tages-, Monats- bzw. Jahresangabe darstellen. Die Objektmenge der ersten Subsorte "absoluter_zeitpunkt" wird derivativ definiert durch die Menge aller zusammengesetzten Objekte datum(tt,mm,jj)="tt.mm.jj" aus der Bildmenge der Operation "datum". Die zweite Subsorte "relativer_zeitpunkt" stellt die Zielsorte des Operationssymbols "Wöchentlich" mit der Sorte "wochentag" als Argument dar. Die Objektmenge dieser zweiten Subsorte wird abermals derivativ definiert, und zwar als die Menge aller zusammengesetzten Objekte wöchentlich(ww)="jeden ww" aus der Bildmenge der Operation "wöchentlich". Dann besitzen die formalen Objekte für die Sorte "zeitpunkt" entweder die tupuläre Struktur datum(tt,mm,jj) oder aber die unäre Struktur wöchentlich(ww). Die derivativ definierte Objektmenge dieser Sorte ist die Vereinigung aller vorgenannten heteromorphen formalen Objekte aus den Objektmengen der beiden involvierten Subsorten. Formal gilt für diese algebraische Spezifizierung von Zeitpunkten mit "SYMBOL" und "STRING" als allgemeinen Objektmengen aller symbolischen bzw. stringartigen Ausdrücke:

Signatur TERMIN =

sorts:
tag
monat
jahr
absoluter_zeitpunkt
wochentag
relativer_zeitpunkt
zeitpunkt

Ops:		
	$TT_1, ..., TT_{31}$:	$\rightarrow$ tag
	$MM_1, ..., MM_{12}$:	$\rightarrow$ monat
	$JJ_1, ..., JJ_{100}$:	$\rightarrow$ jahr
	Datum:	tag monat jahr $\rightarrow$ absoluter_zeitpunkt
	$WW_1, ..., WW_7$:	$\rightarrow$ wochentag
	Wöchentlich:	wochentag $\rightarrow$ relativer_zeitpunkt
	Abs_zeitpunkt:	absoluter_zeitpunkt $\rightarrow$ zeitpunkt
	Rel_zeitpunkt:	relativer_zeitpunkt $\rightarrow$ zeitpunkt

TERMIN-Algebra =

Obs:
OB_{tag} = STRING
OB_{monat} = STRING
OB_{jahr} = STRING
$OB_{absoluter\_zeitpunkt}$ = SYMBOL
$OB_{wochentag}$ = STRING
$OB_{relativer\_zeitpunkt}$ = SYMBOL
$OB_{zeitpunkt}$ = SYMBOL

ops:		
	"01", ..., "31":	$\rightarrow OB_{tag}$
	"01", ..., "12":	$\rightarrow OB_{monat}$
	"00", ..., "99":	$\rightarrow OB_{jahr}$
	datum:	OB_{tag} x OB_{monat} x $OB_{jahr} \rightarrow OB_{absoluter\_zeitpunkt}$ (tt,mm,jj) $\rightarrow$ datum(tt,mm,jj) = "tt.mm.jj"
	"Montag", ..., "Sonntag":	$\rightarrow OB_{wochentag}$
	wöchentlich:	$OB_{wochentag} \rightarrow OB_{relativer\_zeitpunkt}$ ww $\rightarrow$ wöchentlich(ww) = "jeden ww"
	abs_zeitpunkt:	$OB_{absoluter\_zeitpunkt} \rightarrow OB_{zeitpunkt}$ datum(tt,mm,jj) $\rightarrow$ ab_zeitpunkt(datum(tt,mm,jj)) = datum(tt,mm,jj)
	rel_zeitpunkt:	$OB_{relativer\_zeitpunkt} \rightarrow OB_{zeitpunkt}$ wöchentlich(ww) $\rightarrow$ rel_zeitpunkt(wöchentlich(ww)) = wöchentlich(ww)

Dabei wird für den Umgang mit Strings eine intuitiv übersichtlich Notation gewählt, die sich jedoch nicht unmittelbar in der sonst zugrundegelegten Programmiersprache Turbo-PROLOG implementieren läßt. Für eine Implementierung in dieser Programmiersprache müßte dagegen der dreistellige Konkatenationsoperator concat(X_1,X_2,X_3) verwendet werden, der z.B. von KINNEBROCK (1988), S. 136, definiert wird. Er verknüpft die beiden stringartigen Objekte X_1 und X_2 zu einem neuen stringartigen Objekt X_3. Da in PROLOG Terme nicht mit der gleichen Präzision wie im Signaturkonzept oder im strengen prädikatenlogischen Kalkül definiert werden, müssen die teilevaluierten Terme ww, tt, mm und jj durch entsprechende Variablen X_0, X_1, X_2 bzw. X_3 ersetzt werden. Dann gilt mit weiteren Hilfsvariablen Y_0, Y_1, Y_2 und Y_3 z.B.:

a) concat("jeden ",X_0,Y_0)
wöchentlich(X_0)=Y_0

b) concat(X_1,".",Y_1)
concat(Y_1,X_2,Y_2)
concat(Y_2,".",Y_3)
concat(Y_3,X_3,Y_0)
datum(X_1,X_2,X_3)=Y_0

Die multiple Interpretation einer Sorte durch extensional verschiedene derivative Objektmengen für ihre Subsorten kann zwar vermieden werden. Für diesen Zweck wird der Sorte von vornherein die Vereinigungsmenge aller derivativ definierten Objektmengen oder eine ihrer Obermengen als originär definierte Objektmenge zugeordnet. Doch erweist sich die multiple Sortendefinition durch extensionsverschiedene Subsorten bei der praktischen Anwendung des Signaturkonzepts als wesentlich flexibler. Denn bei der originären Sorteninterpretation müßten im allgemeinen alle Elemente aus der umfassenden Objektmenge explizit aufgelistet werden. Statt dessen reicht es bei der Verwendung von Subsorten aus, die Sorte zunächst durch eine allgemeine Objektmenge ohne explizite Elementeaufzählung originär zu interpretieren. Hierfür bietet sich vor allem die - bereits oben eingeführte - allgemeine Objektmenge "SYMBOL" aller symbolischen Ausdrücke an. Später können die tatsächlich benötigten formalen Objekte durch Subsorten eingeführt werden, die durch entsprechend eingeschränkte Objektmengen interpretiert werden. Mittels der derivativ definierten Objektmengen der Subsorten wird die originäre allgemeine Objektmenge OB_{symbol} in übersichtlich strukturierter Weise präzisiert. Genau dies war auch im o.a. Beispiel der Zeitpunktspezifizierung der Fall.
Darüber hinaus vereinfacht sich die multiple Sorteninterpretation bei der späteren prädikatenlogischen Implementierung von Synthetischen Netzen auf Turbo-PROLOG-Basis noch. Denn bei dieser Implementierungsweise werden die Objektmengen von Zielsorten niemals explizit angeführt. Sie werden nur implizit durch die Objektmengen der Sorten aus den Argumenten derjenigen Operationssymbole definiert, deren Zielsorten sie darstellen. Daher wird die vorgenannte Objektmenge OB_{symbol} überhaupt nicht originär definiert, sondern ergibt sich implizit aus den Objektmengen der involvierten Subsorten; vgl. dazu die Turbo-PROLOG-Implementierung des o.a. Beispiels für die Zeitpunktspezifizierung bei KINNEBROCK (1988), S. 44.

22) Beispielsweise wird eine Sorte $sort_i$ betrachtet, die durch die Objektmenge OB_i originär interpretiert wird. Ihre derivative Interpretation erfolgt durch ein Operationssymbols Op_j und eine zugehörige Operation op_j mit:

$$Op_j: \quad sort_{i(j.1)} \dots sort_{i(j.Kj)} \rightarrow sort_i$$

$$op_j: \quad OB_{i(j.1)} \times \dots \times OB_{i(j.Kj)} \rightarrow OB_i$$

$$(ob_1,\dots,ob_{Kj}) \rightarrow op_j(ob_1,\dots,ob_{Kj}) = ob_{Kj+1}$$

Dann wird das gleiche formale Objekt einmal mit $ob_{Kj+1} \in OB_i$ als strukturloses atomares Objekt betrachtet. Ein anderes Mal weist es mit $op_j(ob_1,\dots,ob_{Kj})$ eine innere Struktur auf: Es ist mit Hilfe des Operators $op_j(\dots)$ aus anderen formalen Objekten $ob_1,\dots,ob_{Kj}$ zusammengesetzt. Die Sorten aus dem Argument des Operationssymbols Op_j können ihrerseits originär oder ebenfalls derivativ interpretiert werden. Auf diese Weise ist es möglich, formale Objekte mehrfach ineinander verschachtelt aus anderen formalen Objekten und Operatorapplikationen zusammenzusetzen. Infolge dieser verschachtelten internen Struktur werden solche Objekte auch als hierarchisch strukturierte Objekte bezeichnet. Sie entsprechen dem Konstruktionsprinzip der objektorientierten Systemgestaltung, bei der komplexe Objekte sukzessiv auf weniger komplexe Subobjekte zurückgeführt werden.

23) Zu diesem Zweck wird die Sorte $sort_i$, der das betrachtete formale Objekt ob_k mit $ob_k \in OB_i$ zugeordnet ist, in der zugrundeliegenden Signatur mehrfach als Zielsorte von Operationssymbolen mit verschiedenartig strukturierten Argumenten angeführt. In der zugeordneten Algebra werden diese Operationssymbole jeweils durch die identische Operation interpretiert. Dies entspricht der Vorgehensweise bei der Subsortenbildung. Nur werden die gleichartigen 1-stelligen Operationssymbole, die bei der Subsortenbildung verwendet wurden, zu Operationssymbolen mit unterschiedlich strukturierten Argumenten verallgemeinert. Vgl. dazu das Beispiel der Zeitpunktspezifizierung in der voranstehenden Fußnote. Dort führen die Operationssymbole "Datum" und "Wöchentlich" mit ihren differierenden 3- bzw. 1-stelligen Argumenten zu unterschiedlichen Strukturen der formalen Objekte aus der Sorte "zeitpunkt" je nachdem, ob diese Objekte den Subsorten absoluter bzw. relativer Zeitpunktformulierungen zugeordnet sind.

24) Dieser Perspektivismus entspricht auf formaler Ebene der früher aufgestellten materialen Anforderung an Modellierungskonzepte, die modellierten Objekte in "pluralistischer" oder "multiparadigmatischer" Weise verschiedenartig darstellen zu können.

25) Dieser Aspekt wird später bei der modularen Gestaltung von Maschinenbelegungsmodellen für Flexible Fertigungssysteme genutzt. Auf diese Weise können Module integriert werden, die verschiedene Perspektiven auf mindestens ein gemeinsames identisches formales Objekt darstellen.

26) Es könnte eingewendet werden, es läge überhaupt kein "gleiches" Objekt vor, weil seine Strukturen bzw. Anwendungskontexte variierten. Dieser Einwand ist jedoch nicht berechtigt. Denn einerseits wird die Gleichheit von formalen Objekten, die auf verschiedene Weise intern strukturiert sind, bei SIG-Algebren durch Gleichungen in den Deklarationen der zugehörigen Operationen explizit festgelegt. So wurde in dem Beispiel, das an früherer Stelle präsentiert wurde, durch die Gleichung $op_j(ob_1,..,ob_{Kj})=ob_{Kj+1}$ expliziert, daß das zusammengesetzte Objekt $op_j(ob_1,..,ob_{Kj})$ und das atomare Objekts ob_{Kj+1} als Darstellungsformen des gleichen formalen Objekts gelten. Andererseits wird die Identität eines formalen Objekts ob_k nicht dadurch tangiert, daß es in verschiedenen Kontexten verwendet wird. Dies drückt sich dadurch aus, daß es in allen Kontexten als dasselbe Objekt ob_k notiert wird.

27) Z.B. stellt das zusammengesetzte Objekt $op_j(ob_1,..,ob_{Kj})$ ein solches flach strukturiertes Objekt dar.

28) Wenn ein Objekt höherer Stufe nur aus genau einem Objekt niedrigerer Stufe und genau einem Operator zusammengesetzt wird, liegt ein komplexes Objekt mit minimaler innerer Struktur vor. Dies ist z.B. für das Objekt $ob_2=op_j(ob_1)$ der Fall. Minimal strukturierte formale Objekte gehören zu den Bildmengen von 1-stelligen Operationen op_j, deren Definitionsbereiche Mengen atomarer Objekte darstellen.

29) Beispielsweise handelt es sich bei dem zusammengesetzten Objekt $op_1(op_2(...op_n(ob_k)))$ um ein tief strukturiertes, n-fach verschachteltes Objekt.

30) Eine Kombination von horizontaler und vertikaler Strukturierung liegt beispielsweise für das Objekt $op_1(ob_1,op_2(op_3(ob_2)),op_3(ob_3))$ vor.

31) Es wird daher auch von einem algebraischen Universum gesprochen, um es von dem später eingeführten prädikatenlogischen Universum abzuheben.

32) Damit entspricht das algebraische Universum der ontologischen Grundmenge, die LEE,R. (1988a), S. 226, im Kontext von Datenbankschemata als "universal set" und als "domain of individuals" bezeichnet.

33) Die Strukturaufprägung geschieht zunächst durch die originär definierten, jeweils sortenspezifischen Objektmengen OB_i. Hinzu kommen die inneren Strukturen der Mengen zusammengesetzter Objekte, die als Bildmengen von sortenbezogen eingeführten Operationen derivativ definiert werden.

34) Ähnlich hebt RICHTER,M.M. (1988), S. 25, den "sauberen und schrittweisen Aufbau der Begriffswelten vom Einfachen zum Komplexen" als die herausragende Eigenschaft von Konzepten hervor, welche die Prädikatenlogik durch eine Sortenstruktur überlagern.

35) Vgl. dazu die Anmerkung zum Begriff der Algebren.

36) Im Gegensatz zu anderen Definitionen von Algebren werden hier aber nicht bestimmte "algebraische" Funktionen vorgeschrieben, sondern grundsätzlich alle Funktionen zugelassen, die als K-stellige rechtseindeutige Abbildungen definiert sind.

37) In dieser Arbeit stehen die Überlagerungen von Sortenstrukturen im Vordergrund des Interesses, nicht aber die überlagerten Funktionen. Daher wird auch kurz von sortierten Algebren gesprochen.

38) Dabei wird als Normalfall aller SIG-Algebren unterstellt, daß die zugrundeliegenden Signaturen in ihren Sortenmengen SO mindestens zwei verschiedene Sorten enthalten, daß also $I \geq 2$ für $SO=\{sort_1,...,sort_I\}$ gilt. Der degenerierte Fall mit $I=1$ wird fortan - wenn nicht ausdrücklich anders festgelegt - nicht berücksichtigt.

39) Dies entspricht bei der Modulintegration, die bereits in einer früheren Anmerkung angesprochen wurde, der Modulverknüpfung über gemeinsame Modulschnittstellen.

40) Dieser nachfolgend konkretisierte Regelfall wird fortan für OR-Programme - sofern nicht ausdrücklich anders vermerkt - implizit vorausgesetzt.

41) Die Anzahl der Objektmengen von OR-Programmen läßt sich erweitern, indem Teilmengen der - nachfolgend angeführten - Ganz- oder Rationalzahlen jeweils als eigenständige Objektmengen definiert werden. Dabei handelt es sich jedoch nicht um tatsächlich neuartige Objektmengen, sondern lediglich um Derivate der originären Objektmengen "INTEGER" bzw. "REAL", die durch spezielle Einschränkungen gewonnen werden. So resultiert z.B. die "Objektmenge" der natürlichen Zahlen aus der Objektmenge der Ganzzahlen durch die Restriktion, daß die Ganzzahlen positiv ($\mathcal{N}_+$) oder nicht-negativ ($\mathcal{N}_0$) sein sollen. Da sich auf diese Weise beliebig viele derivative Teilmengen - wie etwa auch die Menge der Dualzahlen - bilden lassen, ohne neuartige mathematische Operationen auf diesen Teilmengen einzuführen, werden diese Teilmengen hier nicht als Objektmengen sui generis gezählt. Von einer neuartigen Objektmenge wird vielmehr erst dann gesprochen, wenn auf ihr neuartige mathematische Operatio-

nen definiert werden. Neuartige Objektmengen lägen erst dann vor, wenn z.B. irrationale, komplexe oder transfinite Zahlen zugelassen würden. Dies ist aber für OR-Programme im allgemeinen nicht der Fall.

42) Obwohl die Objektmenge "REAL" prima facie die Menge aller reellen Zahlen notiert, ist dies für OR-Programme zumeist nicht der Fall. Denn in OR-Programmen werden im allgemeinen keine irrationalen Zahlen verwendet. Dies gilt noch strenger für die Programmiersprachen und Automatischen Informationsverarbeitungsanlagen, mit deren Hilfe die Ausführung solcher Programme implementiert wird. Diese Sprachen und Anlagen lassen nur Zahlen endlicher Stelligkeit zu. Im numerischen Bereich können daher mit Automatischen Informationsverarbeitungssystemen nur endliche Dezimalzahlen bewältigt werden. Folglich bleibt hier die Implementierung von OR-Programmen zwangsläufig auf die Menge aller Rationalzahlen beschränkt. Dagegen lassen sich irrationale Zahlen, die unendliche Dezimalbrüche darstellen, so lange nicht erfassen, wie sie als numerische Ausdrücke behandelt werden. Der Ausweg, mit irrationalen Zahlen als nicht-numerischen Symbolen umzugehen, besteht zwar grundsätzlich im Rahmen der symbolischen Programmierung. Dieser Ansatz wird jedoch seitens des konventionellen Operations Research nicht verfolgt. Er wird daher hier aus der Regelfallbetrachtung ausgeschlossen. Folglich umfaßt die Objektmenge "REAL" für OR-Programme im allgemeinen nur die Rationalzahlen. Strenggenommen umfaßt sie sogar bei der konkreten Implementierung eines OR-Programms auf einer Automatischen Informationsverarbeitungsanlage nur diejenige echte Teilmenge der Rationalzahlen, die auf jener Anlage mit der zur Verfügung stehenden maximalen Stellenanzahl für numerische Objekte bewältigt werden kann. Vgl. zur Ausgrenzung reeller Zahlen aus dem Bereich der Automatischen Informationsverarbeitung auch EHRIG (1985a), S. 13. Vgl. zur algebraischen Definition der Menge aller Rationalzahlen EHRIG (1985a), S. 149f.

43) Eine präzise Definition der BOOL'schen Objektmenge durch eine algebraische Spezifikation findet sich bei EHRIG (1985a), S. 13f. u. 26f. Allerdings wird dort die Bezeichnung "bool" für die Sorte verwendet, die dieser Objektmenge zugrundeliegt und hier als "wahr_wert" bezeichnet wird. Eine Ausweitung auf das umfassender spezifizierte Konzept BOOL'scher Algebren erfolgt in EHRIG (1985a), S. 29ff.

44) Signaturen, SIG-Algebren und daraus abgeleitete Konzepte, die auf der Sorte "zeichenfolge" und ihrer Objektmenge $OB_{zeichenfolge}$ beruhen, werden z.B. bei EHRIG (1985a), S. 35ff.; RECK (1988), S. 43f., 49 u. 51ff., behandelt. Vgl. dazu auch das nachfolgende Beispiel der Signatur "String" und ihrer "string"-Algebra.
Die Analyse formaler Objekte aus "string"-Algebren hat im Rahmen der Petrinetz-Theorie eine breite Rezeption gefunden. Sie führte zu einer breit ausdifferenzierten Theorie netzbasierter formaler Sprachen.

45) Der Begriff der Zeichenfolge umgreift den des Einzelzeichens (Buchstabens) als Grenzfall atomarer Zeichenfolgen; vgl. dazu das Beispiel der "string"-Algebra.

46) Unter Zahlen werden hier alle formalen Objekte aus den bereits eingeführten Objektmengen "INTEGER" und "REAL" verstanden. Der Zahlbegriff schließt Ziffern als atomare Zahlen mit ein.

47) Diese formallogischen "Individuen" sind nicht mit individuellen realen Objekten zu verwechseln. Reale Objekte wurden bisher überhaupt noch nicht eingeführt. Zwecks eindeutiger Unterscheidung werden "Individuen" in diesem formallogischen Sinne hier durch Anführungszeichen gekennzeichnet. Die früher kritisch angesprochenen "Individuen" in Prädikat/Transition-Netzen entsprechen den hier betrachteten formallogischen "Individuen".

48) In Anlehnung an EHRIG (1985a), S. 23, und RECK (1988), S. 44 (dort umfassender und abweichende Notation).

49) Der Wechsel von der Großschreibung "TOP" im Ausdruck TOP_{STN} aus dem Kontext der Stelle/Transition-Netze zur Kleinschreibung "top" in den hier signaturbezogenen Ausdrücken - wie z.B. SIG_{top} - besitzt keine materielle Bedeutung. Sie dient nur der Angleichung an die Notationskonvention des Signaturkonzepts, Konstanten durchweg mit Kleinbuchstaben auszudrücken. Jede Netztopologie stellt eine solche Konstante dar.

50) In der bisher eingeführten Notation würden die Konstanten, die aus den Konstantensymbole Ks_m und Kt_n abgeleitet sind, als "ks_m" bzw. "kt_n" notiert. Um die Übereinstimmung mit den Notationskonventionen des Petrinetz-Konzepts zu wahren, werden die Konstanten der s- und t-Knoten in der oben eingeführten Weise notiert.

51) Statt dessen hätten als Basis der Netztopologie auch die beiden atomaren Netztopologien $top_s=(\{s_m\},\emptyset,\emptyset)$ und $top_t=(\emptyset,\{t_n\},\emptyset)$ gewählt werden können. Sie bestehen jeweils nur aus genau einer Stelle bzw. Transition. Diese beiden komplexen Objekte top_s und top_t stellen aber nur für allgemeine Netze zulässige topologische Strukturen dar. Für Petrinetze verbietet dagegen die Verknüpftheitsbedingung IB_V, daß ein Netz aus einem einzigen, infolgedessen auch unverknüpften Knoten besteht. Da Stelle/Transition-Netze einen Unterfall der Petrinetze darstellen, wurden oben nicht die verbotenen atomaren, sondern die kleinsten zulässigen Topologien von Petrinetzen als Konstanten top_{st} und top_{ts} eingeführt.

52) Als Besonderheit der mehrstelligen "ergänze..."-Operationen ist zu vermerken, daß es sich infolge der nachgestellten Einschränkungen "falls..." um partielle Funktionen handelt. Diese partielle Definition der "ergänze..."-Operationen ist notwendig, um zu gewährleisten, daß nur solche Kanten zu einer bereits bestehenden Netztopologie hinzugefügt werden, deren adjazenten Knoten entweder in dieser Topologie schon enthalten sind oder uno actu ebenfalls ergänzt werden.

53) Strenggenommen erfüllt die SIG_{top}-Algebra für Netztopologien nicht die Struktur jener formalen Sprachen, die bereits mit Hilfe der Sortensprache exemplarisch skizziert wurden. Es fehlt der Bezug auf ein Alphabet und den hierüber definierten freien Monoid. Der Kerngedanke formaler Sprachen, einen Generator zur Erzeugung aller innersprachlich zulässigen Ausdrücke ("Worte") zu definieren, wird jedoch auch von der o.a. SIG_{top}-Algebra erfüllt. Daher wird sie hier ebenfalls als eine - großzügig ausgelegte - formale Sprache angesehen.

54) Dies wird hier nicht für die einfache Netzklasse der Stelle/Transition-Netze nachgewiesen, sondern später für die komplexe Klasse der Synthetischen Netze ausgeführt. Da die letzten die ersten als Grenzfall umfassen, belegen jene Ausführungen mittelbar auch die hier aufgestellte Beschriftungsbehauptung für Stelle/Transition-Netze.

4.1.4 SIG-Termfamilien

Durch die Einführung einer SIG-Algebra wurde die zugrundeliegende Signatur SIG um sortenspezifische Objektmengen einer SIG-Algebra erweitert. Eine zweite Entfaltung des Signaturkonzepts erfolgt durch das Konzept der SIG-Termfamilie. Es führt sortenspezifische Termmengen als neuartige formalsprachliche Konstrukte ein. Hierbei wird die Objektorientierung von SIG-Algebren durch eine Variablenorientierung ergänzt. Den Variablen entspricht im Kontext der SIG-Algebra kein Pendant. Sie stellen formalsprachliche Ausdrücke sui generis dar, die das Ausdrucksvermögen des Signaturkonzepts beträchtlich erweitern.

Definition: Termfamilie

Eine SIG-Termfamilie ist eine Mengenfamilie $TMF_{SIG}(VAF)$, die bezüglich der Signatur $SIG=(SO,OP)$ und mit Hilfe einer Variablenfamilie VAF in induktiver Weise[1)] aufgebaut ist[2)]:

- Jedes nullstellige Operationssymbol Op_j: $\rightarrow sort_i$ aus der Menge OP der Signatur SIG ist als sortenspezifisches Konstantensymbol Ko_j mit $Ko_j=Op_j$ ein Basisterm der Sorte $sort_i$.
- KO_i ist die Menge aller Konstantensymbole Ko_j, die einer Sorte $sort_i$ aus der Signatur SIG zugeordnet sind.
- Jede sortenspezifische Variable $X_{i.v}$ ist ein Basisterm der Sorte $sort_i$. Zu jeder Sorte können beliebig viele[3)] sortenspezifische Variablen eingeführt werden.
- VA_i ist die Menge aller Variablen $X_{i.v}$, die einer Sorte $sort_i$ aus der Signatur SIG zugeordnet sind.
- $VAF=(VA_i\colon i=1,...,I)$ ist die Familie aller sortenspezifischen Variablenmengen VA_i, die für die Signatur SIG vorgesehen sind.
- Die Vereinigungsmenge aller Basisterme der Sorte $sort_i$ ist die sortenspezifische Basistermmenge BT_i mit $BT_i=KO_i{\cup}VA_i$.
- Jedes K_j-stellige Operationssymbol Op_j aus der Menge OP der Signatur SIG, das wegen Op_j: $sort_{i(j.1)} \dots sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)}$ kein Konstantensymbol darstellt ($k \in \mathcal{N}_+$), ordnet jedem K_j-Tupel aus Termen $te_1,...,te_{Kj}$ der Sorten $sort_{i(j.1)},...,sort_{i(j.Kj)}$ einen zusammengesetzten Term $te=Op_j(te_i,...,te_{Kj})$ der Sorte $sort_{i(j.Kj+1)}$ zu. Jeder Term te_k mit $k \in \{1,...,K_j\}$ kann sowohl ein Basisterm als auch ein zusammengesetzter Term sein.
- ZT_i ist die Menge aller zusammengesetzten Terme te, die für die Sorte $sort_i$ definiert sind.
- Die sortenspezifische Termmenge $TM_i(VA_i)$ umfaßt als Vereinigungsmenge $TM_i(VA_i)=BT_i{\cup}ZT_i$ alle zuvor definierten Basisterme und zusammengesetzten Terme derselben Sorte $sort_i$. Darüber hinaus enthält sie keine anderen Terme. Jedes Element aus der Termmenge $TM_i(VA_i)$ heißt ein Term te der Sorte $sort_i$.
- Die SIG-Termfamilie $TMF_{SIG}(VAF)$ ist die Zusammenfassung der sortenspezifischen Termmengen $TM_i(VA_i)$ für alle Sorten $sort_i$ der Signatur SIG: $TMF_{SIG}(VAF) = (TM_i(VA_i)\colon i=1,...,I)$.

Erläuterungen und Ergänzungen zur Termfamilien-Definition:

a) Durch Termfamilien wird das Konzept der algebraischen Operationssymbole Op_j wesentlich bereichert: Ursprünglich wurden Operationssymbole Op_j so eingeführt, daß ihre Argumente stets Tupel $(sort_{i(j.1)} \dots sort_{i(j.Kj)})$ aus Sorten $sort_{i(j.k)}$ darstellten. Nunmehr kann das Argument der gleichen Operationssymbole Op_j jeweils ein beliebiges Termtupel $(te_1,\dots,te_{Kj})$ sein. Seine Komponenten te_k sind jeweils Konstantensymbole, Variablen oder die Ergebnisse der Anwendung anderer Operationssymbole auf andere Terme. Dabei gehören die Konstantensymbole, Variablen und Ergebnisse der Operationssymbolanwendungen jeweils zur Sorte $sort_{i(j.k)}$.

b) Die sortenspezifischen Konstantensymbol- und Variablenmengen KO_i bzw. VA_i können sowohl leer als auch unendlich sein. Unendliche Konstantensymbolmengen wurden z.B. bei der Definition aller zulässigen Topologien von Stelle/Transition-Netzen verwendet. Unendliche Variablenmengen werden in dieser Arbeit nicht betrachtet.

c) Durch die zugrundeliegende Signatur SIG sind nur die Konstantensymbole als nullstellige Operationssymbole definiert. Die sortenspezifischen Variablen $X_{i.v}$ sind dagegen in der Signatur SIG noch nicht erklärt[4]. Sie stellen neuartige formale Konstrukte[5] dar, die durch die SIG-Termfamilie erstmals eingeführt werden. Daher werden die sortenspezifischen Termmengen als $TM_i(VA_i)$ notiert, um die sortenzugehörigen Variablenmengen VA_i auszuweisen[6].

d) Häufig werden Variablen als exogen vorgegebene Konstrukte betrachtet, die innerhalb des Signaturkonzepts nicht mehr spezifiziert zu werden brauchen. Statt dessen werden nur noch die Symbole VA_i der Variablenmengen notiert, ohne ihre Elemente - die Variablen $X_{i.v}$ - zu definieren[7]. Es mutet jedoch inkonsequent an, einerseits Konstanten- und Operationssymbole in Signaturen explizit einzuführen, aber andererseits bei Variablen nicht so zu verfahren. Daher werden in dieser Arbeit alle Variablen $X_{i.v}$, die mit $v \in \{1,\dots,V_i\}$ und $V_i \in \mathcal{N}_+$[8] für eine Sorte $sort_i$ definiert sind, als Elemente der sortenspezifischen Variablenmenge explizit aufgelistet: $VA_i \circ \dagger \{X_{i.1},\dots,X_{i.Vi}\}$[9].

e) Die sortenspezifischen Variablenmengen VA_i werden so definiert, daß sie sowohl untereinander als auch mit der Menge OP aus der Signatur SIG paarweise disjunkt sind[10]. Auf diese Weise sind alle Variablen X voneinander und von den Operationssymbolen $Op_j \in OP$ wohlunterschieden.

f) Die Gesamtheit aller sortenspezifischen Variablenmengen VA_i wird als Variablenfamilie $VAF=(VA_i\colon i=1,\dots,I)$ dargestellt. Hierdurch wird die Sortenstruktur einer Signatur - im Gegensatz zur sonst üblichen Bildung einer Vereinigungsmenge[11] - reflektiert. Analog ist die Gesamtheit aller Konstantensymbole der SIG-Termfamilie $TMF_{SIG}(VAF)$ als Familie $KOF=(KO_i\colon i=1,\dots,I)$ aller sortenspezifischen Konstantensymbolmengen KO_i definiert. Nur wenn die Sortenstruktur der Variablen- oder Konstantensymbolgesamtheit nicht interessiert, können auch die Vereinigungsmengen $VA=\cup(i\in\{1,\dots,I\})\colon VA_i$ bzw. $KO=\cup(i\in\{1,\dots,I\})\colon KO_i$ aller sortenspezifischen Variablen- bzw. Konstantensymbolmengen betrachtet werden.

g) Der kategoriale Unterschied zwischen Variablen und Konstantensymbolen ist im konventionellen Signaturkonzept nicht präzise definiert. Beide besitzen die Qualität von Platzhaltern, die durch formale Objekte aus SIG-Algebren ersetzt werden können. Darauf wird später im Zusammenhang mit dem Konzept der Termauswertungen ausführlicher eingegangen. Die Termauswertungen sind für Konstantensymbole und Variablen in der gleichen Weise definiert[12]. Daher werden oftmals Konstantensymbole mit Variablen vermengt[13]. Dennoch werden diese beiden Termkategorien hier deutlich separiert, um eine spätere - prädikatenlogisch motivierte - Differenzierung vorzubereiten. Sie besteht im wesentlichen darin, daß in jeder Interpretation eines prädikatenlogischen Objektmodells zunächst nur die Konstantensymbole auf formale Objekte

(Konstanten) abgebildet werden. Die Variablen werden erst auf einer zweiten Interpretationsstufe durch spezielle Variablenbindungsfunktionen durch Konstanten ersetzt[14]. Diese Zweistufigkeit ist keineswegs eine prädikatenlogische Kuriosität, sondern liegt - zumeist unbemerkt - allen hinreichend ausdrucksstarken[15] formalsprachlichen Modellierungskonzept zugrunde[16].

h) Die Terme einer jeden Sorte $sort_i$ lassen sich in zwei Dimensionen gliedern. Die erste Dimension erstreckt sich auf die Zusammengesetztheit, die zweite auf die Variabilität der Terme. Aus der ersten Perspektive wird zwischen atomaren und zusammengesetzten Termen unterschieden. Die atomaren Terme fallen mit den oben eingeführten Basistermen zusammen. Es handelt sich also entweder um Konstantensymbole oder aber um Variablen. Die zusammengesetzten Terme sind dagegen aus anderen Termen und aus mindestens einstelligen Operationssymbolen aufgebaut. Aus der zweiten Perspektive wird differenziert, ob ein Term entweder mindestens eine ober aber keine Variable enthält. Ein Term, der mindestens eine Variable umfaßt, wird als variabler Term bezeichnet[17]. Ein Term, der keine Variable enthält, heißt ein variablenfreier Term oder Grundterm[18]. Ein Grundterm kann sowohl ein Basisterm als auch ein zusammengesetzter Term sein. Im ersten Fall handelt es sich um ein einzelnes Konstantensymbol te mit $te \in KO_i$. Im zweiten Fall liegt ein zusammengesetzter Term te mit $te = Op_j(te_1,...,te_{Kj})$ und $K_j \in \mathcal{N}_+$ vor. Seine Teilterme te_k mit $k \in \{1,...,K_j\}$ stellen entweder unmittelbar Konstantensymbole dar oder sind ihrerseits durch weitere Operationssymbole Op_j' aus Konstantensymbolen mittelbar zusammengesetzt.

i) Mit GT_i wird die Menge aller Grundterme für die Sorte $sort_i \in SO$ bezeichnet. Die Familie der sortenspezifischen Grundtermmengen GT_i stellt die Grundtermfamilie GTF der SIG-Termfamilie $TMF_{SIG}(VAF)$ dar:

$$GTF = (GT_i: i = 1,...,I)$$

Da Grundterme als variablenfreie Terme definiert sind, kann die Grundtermfamilie mit $VAF = () = \varnothing$ auch äquivalent als $GTF = TMF(\varnothing)$ notiert werden.

j) Die Darstellungsweise von zusammengesetzten Termen ist mehrdeutig. Ein zusammengesetzter Term $te = Op_j(te_i,...,te_{Kj})$ kann als Term te oder auch als Term $Op_j(te_i,...,te_{Kj})$ angesprochen werden. Darüber hinaus gilt diese Ambiguität auch für die Terme te_k aus dem Argument $(te_i,...,te_{Kj})$ des Operators $Op_j(...)$, sofern es sich bei diesen Termen te_k mit $k \in \{1,...,K_j\}$ ebenfalls um zusammengesetzte Terme handelt.

k) Die Termdarstellung ist aber eindeutig, sobald alle zusammengesetzten Terme mittels der Deklarationen derjenigen Operationssymbole, die an der Termdefinition beteiligt sind, vollständig auf Basisterme und Operatoren zurückgeführt sind. Terme, die entweder schon Basisterme sind oder nur aus Basistermen und Operatoren bestehen, heißen vollständig explizierte Terme. Alle zuvor eingeführten Terme können als vollständig explizierte Terme notiert werden. Die Terme und die entsprechenden Termmengen können jeweils durch das Superskript "^{ex}" gekennzeichnet werden[19]. Beispielsweise handelt es sich bei der Menge $TM_i^{ex}(VA_i)$ um alle vollständig explizierten Terme, die für eine Sorte $sort_i$ definiert sind. Ein Term aus dieser Menge wird durch die Notation te^{ex} ausdrücklich als vollständig explizierter Term dargestellt.

Formal lassen sich die vollständig explizierten Terme einer Signatur SIG=(SO,OP) als Ausdrücke einer speziellen Termsprache TES_{SIG} darstellen. Die Termsprache ist mit TE^n als beliebiger Termmenge n-ter Stufe und $n \in \mathcal{N}_+$ rekursiv definiert durch:

für alle $n \in \mathcal{N}_+$ mit $n \geq 2$:

$$TE^n = \{te: (\exists(j \in \{1,...,J\}): Op_j \in OP \wedge Op_j: sort_{i(j.1)} \ldots sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)})$$
$$\wedge (\exists(n_1 \in \{1,...,n\text{-}1\}): te_i \in TE^{n1})$$
$$\ldots$$
$$\wedge (\exists(n_{Kj} \in \{1,...,n\text{-}1\}): te_{Kj} \in TE^{nKj})$$
$$\wedge \; te = Op_j(te_i,...,te_{Kj})\}$$

für n=1:

$$TE^1 = BT_{SIG}$$

Daraus folgt allgemein:

$$TES_{SIG} = \cup \, (n \in \mathcal{N}_+): TE^n$$

Terme aus den Termmengen TE^n sind für $n \geq 2$ jeweils über n-1 Schritte durch Operationssymbole aus Konstantensymbolen oder Variablen als Basistermen zusammengesetzt[20]. Terme 1. Stufe stellen unmittelbar Basisterme dar. Jede Menge vollständig explizierter Terme aus derselben Sorte $sort_i$ kann als Schnittmenge zwischen der sortenspezifischen Termmenge $TM_i(VA_i)$ und der Termsprache TES_{SIG} eingeführt werden: $TM_i(VA_i)^{ex} = TM_i(VA_i) \cap TES_{SIG}$.

l) Die eingangs angeführte Definition von SIG-Termfamilien besitzt den Vorzug, die induktive Komplexitätssteigerung von Termen besonders anschaulich darzustellen. Statt dessen kann aber auch von Interesse sein, Termfamilien so kompakt wie möglich zu definieren[21]. Dies leistet die nachstehende äquivalente Definition. Eine SIG-Termfamilie ist die kleinste[22] Familie $TMF_{SIG}(VAF)$, für die gilt[23]:

- Jedes nullstellige Operationssymbol $Op_j: \rightarrow sort_i$ aus der Menge OP der Signatur SIG ist als Konstantensymbol Ko_j mit $Ko_j = Op_j$ ein Term der Sorte $sort_i$. KO_i ist die Menge aller Konstantensymbole der Sorte $sort_i$[24].
- Jede Variable $X_{i.v}$ ist ein Term der Sorte $sort_i$. VA_i ist die Menge aller Variablen $X_{i.v}$, die mit $v \in \{1,...,V_i\}$ und $V_i \in \mathcal{N}_0$ für die Sorte $sort_i$ definiert sind[25]. Die sortenspezifischen Variablenmengen sind nicht innerhalb der Signatur SIG definiert, sondern müssen explizit eingeführt werden.
- Wenn $Op_j: sort_{i(j.1)} \ldots sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)}$ ein K_j-stelliges Operationssymbol aus der Menge OP der Signatur SIG mit $K_j \in \mathcal{N}_+$ ist und wenn $te_1,...,te_{Kj}$ Terme der Sorten $sort_{i(j.1)},...,sort_{i(j.Kj)}$ sind, dann ist der Ausdruck te_{Kj+1} mit $te_{Kj+1} = Op_j(te_i,...,te_{Kj})$ ein Term der Sorte $sort_{i(j.Kj+1)}$.
- $TM_i(VA_i)$ ist die Menge aller Terme der Sorte $sort_i$.
- $VAF = (VA_i: i = 1,...,I)$ ist die Familie aller sortenspezifischen Variablenmengen VA_i für die Signatur SIG.
- $TMF_{SIG}(VAF) = (TM_i(VA_i): i = 1,...,I)$ ist die Familie aller sortenspezifischen Termmengen $TM_i(VA_i)$ für die Signatur SIG.

m) Eine spezielle Sektion "Terms" erlaubt die rein formale Definition einer SIG-Termfamilie $TMF_{SIG}(VAF)$. Der Übersichtlichkeit halber wird dieser Sektion eine zweite Sektion "Vars" vorangestellt. Sie umfaßt alle Variablen, die zur Definition der Signatur SIG hinzukommen[26]. Dagegen können alle Konstantensymbole, die in der Operationssymbolemenge OP der zugrundeliegenden Signatur SIG als nullstellige Operationssymbole definiert sind, unmittelbar aus der bereits eingeführten Sektion "Ops" entnommen werden. Sie werden in der Sektion "Terms" in den sortenspezifischen Konstantensymbolemengen KO_i für alle Sorten $sort_i$ mit $i \in \{1,...,I\}$ zusammengefaßt. Daraus folgt für die rein formale Definition einer SIG-Termfamilie $TMF_{SIG}(VAF)$:

<u>Vars</u>: $VA_1 = \{X_{1.1},...,X_{1.V1}\}$

...

$VA_I = \{X_{I.1},...,X_{I.VI}\}$

$VAF = (VA_i: i=1,...,I)$

<u>Terms</u>: $KO_1 = \{Ko_j: \exists(Op_j \in OP): Ko_j = Op_j \wedge Op_j: \rightarrow sort_1\}$

$TM_1(VA_1)$ minimal mit:

$KO_1 \subseteq TM_1(VA_1)$

$VA_1 \subseteq TM_1(VA_1)$

$(\forall(Op_j \in OP): (Op_j: sort_{i(j.1)} \ldots sort_{i(j.KJ)} \rightarrow sort_{i(j.Kj+1)}$

$\wedge (\forall(k \in \{1,...,K_j\}): te_k \in TM_{i(j.k)}(VA_{i(j.k)})) \wedge i(j.K_j+1) = 1 \wedge Op_j(te_1,...,te_{KJ}) = te_{KJ+1})$

$\rightarrow te_{KJ+1} \in TM_1(VA_1))$

...

$KO_I = \{Ko_j: \exists(Op_j \in OP): Ko_j = Op_j \; Op_j: \rightarrow sort_I\}$

$TM_I(VA_I)$ minimal mit:

$KO_I \subseteq TM_I(VA_I)$

$VA_I \subseteq TM_I(VA_I)$

$(\forall(Op_j \in OP): (Op_j: sort_{i(j.1)} \ldots sort_{i(j.KJ)} \rightarrow sort_{i(j.Kj+1)}$

$\wedge (\forall(k \in \{1,...,K_j\}): te_k \in TM_{i(j.k)}(VA_{i(j.k)})) \wedge i(j.K_j+1) = I \wedge Op_j(te_1,...,te_{KJ}) = te_{KJ+1})$

$\rightarrow te_{KJ+1} \in TM_I(VA_I))$

$TMF_{SIG}(VAF) = (TM_i(VA_i): i=1,...,I)$

n) Die Termfamilie TMF_{SIG} enthält nur formale Ausdrücke, die aus Konstantensymbolen, Variablen und Operationssymbolen bestehen. Das algebraische Universum OB_{SIG} umfaßt dagegen ausschließlich formale Objekte, die sich aus atomaren formalen Objekten (Konstanten) und Operationen (Funktionen) zusammensetzen. Dies bedeutet:

- Termfamilie und algebraisches Universum derselben Signatur sind aufgrund der Verschiedenartigkeit ihrer Komponenten grundsätzlich disjunkt.
- Konstantensymbole und Konstanten einerseits sowie Operationssymbole und Operationen andererseits korrespondieren miteinander. Dies unterstreicht die eingangs festgestellte Dualität zwischen SIG-Termfamilien und SIG-Algebren.
- Mit den Variablen einer SIG-Termfamilie korrespondiert kein formales Konstrukt einer SIG-Algebra. Die Dualität zwischen beiden gilt daher nur eingeschränkt.

Die kompromißlose Separation zwischen Konstantensymbolen, Variablen und Operationssymbolen einerseits sowie Konstanten und Funktoren andererseits ist ein Charakteristikum des Signaturkonzepts. Sie entspricht der Trennung einer abstrakten Term-Konzeptebene, auf der nur mit Variablen und Symbolen gearbeitet wird, von einer konkreten algebraischen Konzeptschicht, innerhalb derer Konstanten und Funktionen angewendet werden. Dies reflektiert eine wesentliche konzeptionelle Wurzel des Signaturkonzepts, im Bereich Automatischer Informationsverarbeitungssysteme zwischen abstrakten Datentypen einerseits und konkreten Daten[27] andererseits zu unterscheiden[28].

Dagegen werden in konventionellen mathematischen und formallogischen Darstellungen die abstrakte termbezogene und die konkrete algebraische Konzeptebene miteinander vermengt. Dies gilt insbesondere auch für übliche prädikatenlogische Abhandlungen, die betriebswirtschaftlich vertrauteren OR-Programme sowie Implementierungssprachen der Automatischen Informationsverarbeitung. Denn dort wird erstens zwischen Konstantensymbolen und Konstanten sowie zweitens zwischen Operationssymbolen und Operationen (Funktionen) überhaupt nicht differenziert. Statt dessen werden von vornherein nur Konstanten und Funktionen verwendet. Diese konkreten algebraischen Konstrukte werden auf derselben Darstellungsebene mit Variablen aus der abstrakten Termfamilie kombiniert. So geht durch das konventionelle Amalgam aus Variablen, Konstanten und Funktionen die oben erläuterte klare Trennung zwischen SIG-Termfamilien und SIG-Algebren verloren. Bemerkenswert ist in diesem Zusammenhang, daß die konventionelle Konfundierung zwischen Termen und algebraischen Ausdrücken mitunter auch erklärten Anhängern des Signaturkonzepts unterläuft[29].

Die konzeptionelle Präzision des Signaturkonzepts und ein Anschluß an konventionelle Darstellungsweisen schließen sich gegenseitig aus. Der Verf. präferiert zwar das Signaturkonzept aufgrund seiner - bereits früher angesprochenen - Reichhaltigkeit und Klarheit. Doch läßt es sich aufgrund der späteren Implementierung Synthetischer Netze mittels der Sprache PROLOG nicht vermeiden, die Trennung zwischen termbezogenen und algebraischen Ausdrücken zugunsten ihrer konventionellen Vermengung bis zum Implementierungsschritt aufzugeben. Darüber hinaus wird das intuitive Verständnis des formalen Apparats Synthetischer Netze erschwert, wenn an der vorgenannten Separation des Signaturkonzepts festgehalten wird. Dies gilt zumindest für Rezipienten, die mit dem Signaturkonzept nicht vertraut sind[30].

Aufgrund der vorgenannten Anschluß- und Verständnisprobleme geht der Verf. folgenden Kompromiß zwischen signaturbezogener Präzision und konventioneller Darstellungsweise ein: Zunächst wurde das Signaturkonzept bis an diese Stelle in seiner strengen Fassung entfaltet. Mit Hilfe des nachfolgenden Konzepts der SIG-Termauswertungen wird aber eine Brücke zur konventionellen Kombination von Variablen, Konstanten und Funktionen geschlagen[31]. In den daran anschließenden Erläuterungen des Signaturkonzepts wird diese konventionelle Ebenenvermengung vorausgesetzt, sofern nicht ausdrücklich andere Festlegungen erfolgen.

Anmerkungen zum Kapitel:

1) Ein Definitionsschema heißt induktiv, wenn aus Basiskonstrukten und Konstruktionsoperationen, die jeweils als bekannt vorausgesetzt werden, schrittweise immer komplexere Konstrukte zusammengesetzt werden. Bei dieser Sukzession werden - beginnend mit den Basiskonstrukten als Induktionsbasis - aus bereits definierten Konstrukten die Definitionen neuer Konstrukte abgeleitet. Dabei werden die konstruktiven Operationen derart hintereinander ausgeführt, daß die Ergebnisse früherer Operationsausführungen die Argumente jeweils unmittelbar folgender Operationsausführungen bilden (können). Vgl. zu diesem induktiven Definitionsschema z.B. LORENZEN,P. (1962), S. 51.
Das Sukzessionsschema induktiver Definitionen entspricht dem Konzept der Definitionskette von CARNAP (1968), S. 22. Es beruht auf dem allgemeinen Konzept diskursiver Argumentation, von bereits Bekanntem zu noch Unbekanntem voranzuschreiten. Dies führt u.a. im Rahmen konventioneller Definitionslehren zu der Auffassung, daß das noch unbekannte Definiendum nicht im Definiens enthalten sein dürfe; vgl. z.B. WHITEHEAD (1925), S. 11; CARNAP (1931c), S. 100; CARNAP (1968), S. 22. Vgl. allgemein zur Betonung diskursiver Schemata (Denkfiguren, Konzepte, Argumentationsweise usw.) HABERMAS (1973b), S. 20ff.; LORENZEN,P. (1975a), S. 14; PERELMAN (1979), S. 11; GETHMANN (1980b), S. 39f.; RESCHER (1980a), S. 7; POPPER (1984b), S. 135f.; RESCHER (1985a), S. 157; BRAUN,W. (1985), S. 25 (ohne Verwendung der Bezeichnung "diskursiv"); GETHMANN (1987), S. 270 u. 273f.; HABERMAS (1988), S. 46 u. 124; KLEIN,S. (1989), S. 120f.
Die Argumentationsnorm der Diskursivität ist allerdings keineswegs denknotwendig; vgl. GETHMANN (1987), S. 277f. Dies wird auch in dieser Arbeit anhand des rekursiven Definitionsschemas aufgezeigt. Es läßt im Gegensatz zu konventionellen Definitionslehren zu, daß das Definiendum auch - aber in einer besonderen Weise - im Definiens enthalten sein darf. Vgl. dazu die Erläuterungen zur rekursiven Konstruktdefinition.
Das induktive Definitionsschema geht allerdings deutlich über iterative Konstruktdefinitionen hinaus. Zwar beruhen auch Iterationen auf der wiederholten Ausführung einer Konstruktionsoperation, die auf unterschiedliche Objekte angewendet wird. Dabei erfolgen jedoch die Operationsausführungen voneinander unabhängig. Daher können sich nachfolgende Operationsausführungen - im Gegensatz zum induktiven Definitionsschema - nicht auf die Resultate der jeweils vorangehenden Operationsausführungen beziehen. Die konzeptionell einfachere iterative Konstruktdefinition findet in betriebswirtschaftlichen Kontexten des öfteren Anwendung, z.B. als Wiederholungsfunktion für Elementarkombinationen im Rahmen der Produktionsfunktion vom Typ "C"; vgl. HEINEN (1983), S. 286ff.. Das iterative Definitionsschema wird auch später bei der Formulierung eines Pseudocode-Moduls für einen Algorithmus benutzt, der zur Konstruktion von Erreichbarkeitsgraphen dient.

2) Vgl. EHRIG (1985a), S. 17f.; RECK (1988), S. 45; REISIG (1989a), S. 5.

3) Der Begriff "beliebig viele" wird in dieser Arbeit stets als Oberbegriff zu den Begriffen "keine", "genau eine" und "mehrere, aber endlich viele" gebraucht.

4) Daher kann dieselbe Signatur SIG mit verschiedensten sortenspezifischen Variablenmengen kombiniert werden. Folglich ist die Familie VAF dieser Variablenmengen nicht signaturspezifisch. Aus diesem Grund wird in der Notation $TMF_{SIG}(VAF)$ das Symbol "VAF" nicht mit dem Index "SIG" versehen. Alle anderen Konstituenten einer Termfamilie - die Konstanten- und Operationssymbole - sind dagegen bereits durch die zugrundeliegende Signatur SIG als 0-stellige bzw. durch mindestens 1-stellige Operationssymbole definiert.

5) Vgl. dazu auch die Quellen, die zum prädikatenlogischen Variablenbegriff angeführt werden. Variablen aus SIG-Termfamilien und aus prädikatenlogischen Kalkülen stellen gleichartige Konstrukte dar.

6) Um sich nicht darauf festlegen zu müssen, welche Variablen konkret benutzt werden, wird im allgemeinen eine beliebig umfassend definierte Variablenfamilie VAF implizit vorausgesetzt. Dies wird erreicht, indem als Variablen zunächst alle mindestens einstelligen Zeichenfolgen aus der SIG-Algebra "string" zugelassen werden, sofern gilt: Sie beginnen jeweils mit einem Großbuchstaben, bestehen nur aus den 26 Buchstaben des lateinischen Alphabets und umfassen weder Sonderlaute (z.B. "ß" oder "ö") noch Sonderzeichen (z.B. "/" oder ";"). Hinzu kommt die genau einstellige, degenerierte Zeichenfolge "_", die für die anonyme Variable reserviert ist. Dann gilt für die Menge VA aller Variablen, die für eine beliebige Signatur SIG zulässig sind, folgende algebraische Definition:

String-var =

sorts: grosszeichen
kleinzeichen
sonderzeichen
zeichen
zeichenfolge
variable

Ops:	$Ko_{01}, ..., Ko_{26}$:	$\rightarrow$ grosszeichen
	$Ko_{27}, ..., Ko_{52}$:	$\rightarrow$ kleinzeichen
	Ko_{53}:	$\rightarrow$ sonderzeichen
	Erzeuge1g:	grosszeichen $\rightarrow$ zeichen
	Erzeuge1k:	kleinzeichen $\rightarrow$ zeichen
	Erzeuge1 :	zeichen $\rightarrow$ zeichenfolge
	Verknüpfe:	zeichenfolge zeichenfolge $\rightarrow$ zeichenfolge
	ErzeugeXs:	sonderzeichen $\rightarrow$ variable
	ErzeugeXg:	grosszeichen $\rightarrow$ variable
	ErzeugeX :	grosszeichen zeichenfolge $\rightarrow$ variable

string-var =

OBs: $OB_{grosszeichen} = ZG$
$OB_{kleinzeichen} = ZK$
$OB_{sonderzeichen} = ZS$
$OB_{zeichen} = Z$
$OB_{zeichenfolge} = ZS \cup Z^*$
$OB_{variable} = VA$

ops:	"A", "B", ..., "Z":	$\rightarrow$ ZG	
	"a", "b", ..., "z":	$\rightarrow$ ZK	
	"_":	$\rightarrow$ ZS	
	erzeuge1g:	$ZG \rightarrow Z$,	erzeuge1g(z) = z
	erzeuge1k:	$ZK \rightarrow Z$,	erzeuge1k(z) = z
	erzeuge1 :	$Z \rightarrow Z^*$,	erzeuge(z) = z
	verknüpfe:	$Z^* \times Z^* \rightarrow Z^*$,	verknüpfe(x,y) = xy
	erzeugeXs:	$ZS \rightarrow VA$,	erzeugeXa(z) = z
	erzeugeXg:	$ZG \rightarrow VA$,	erzeugeX1(z) = z
	erzeugeX :	$ZG \times Z^* \rightarrow VA$,	erzeugeX(z,x) = zx

Die Variablenmenge VA stellt nur die Gesamtheit aller Variablen dar, die für SIG-Termfamilien *grundsätzlich* zulässig sind. Sie braucht durch die konkret verwendeten Terme nicht vollständig ausgeschöpft zu werden.

7) Vgl. zu dieser Einstellung, die Variablen selbst nicht zu spezifizieren, sondern sich nur noch auf exogen gegebene Variablenmengen zu beziehen, z.B. EHRIG (1985a), S. 17; RECK (1988), S. 45; REISIG (1989a), S. 5.

8) Für $V_i = 0$ ist keine Variable der Sorte $sort_i$ definiert, so daß $VA_i = \emptyset$ gilt.

9) Dies führt zu einer speziellen Sektion "Vars", in der alle Variablen einer Termfamilie spezifiziert werden.

10) Diese Festlegung dient der Konzeptklarheit. Sie ist jedoch nicht unbedingt notwendig. Denn die Variablengesamtheit wurde nicht als Vereinigungsmenge, sondern als Familie der sortenspezifischen Variablenmengen definiert.

11) Vgl. EHRIG (1985a), S. 17; RECK (1988), S. 45; REISIG (1989a), S. 5.

12) Vgl. dazu die strukturell übereinstimmenden Auswertungsfunktionen $eval_{i.KO}$ und $eval_{i.VA}$ für Konstantensymbole bzw. Variablen.

13) Vgl. dazu die exemplarischen Erläuterungen zu parametrischen Modellen und zum Parameterbegriff.

14) Diese Zweistufigkeit prädikatenlogischer Interpretationen wird noch ausführlich dargestellt. Im Konzept der Termauswertungen wird sie bereits dadurch antizipiert, daß das unkonventionelle Konzept der teilevaluierten Terme mit zugehörigen Teil- und Komplementauswertungsfunktionen eingeführt wird.

15) Dieser Vorbehalt stellt sicher, daß die Modellierungskonzepte hinreichend ausdrucksfähig sind, um Konstantensymbole und Variablen sowohl darstellen als auch voneinander differenzieren zu können. Aussagenlogische Modellierungskonzepte erlauben z.B. die o.a. Zweistufigkeit nicht, weil sie überhaupt keine Variablen auszudrücken vermögen.

16) Dies äußert sich zumeist darin, daß in solchen Konzepten zunächst Modellschemata entworfen werden, die sowohl Konstantensymbole als auch Variablen umfassen. Falls ein solches Konzept herangezogen wird, um ein aktuelles Modellierungsproblem zu lösen, wird das konzeptspezifische Modellschema durch ein konkretes Modell ersetzt. Das konkrete Modell ist eine zulässige Interpretation des abstrakten Modellschemas, bei der alle Konstantensymbole des Schemas durch Konstanten ersetzt werden. Infolge der Zweistufigkeit der Modellinterpretation werden die Variablen des Modellschemas im konkreten Modell jedoch zunächst noch nicht ersetzt. Daher ist es möglich, im konkreten Modell Entscheidungs- oder Handlungsalternativen durch Variablen zu repräsentieren, die durch mindestens zwei verschiedene formale Objekte ersetzt werden können. Erst auf der zweiten Interpretationsstufe werden auch diese Variablen durch formale Objekte ersetzt. Das Ersetzungsergebnis ist jeweils eine zulässige Lösung des konkreten Modells.
Diese Zweistufigkeit der Interpretation des Modellschemas eines Modellierungskonzepts liegt im Prinzip z.B. jedem entscheidungstheoretischen Modellierungsansatz zugrunde. Sie bleibt dort allerdings zumeist im Dunkeln, weil zwischen Modellschemata als abstrakten Modellen und konkreten Modellen als Schemaausprägungen nicht differenziert wird. Folglich werden dort Konstantensymbole und Variablen auch nicht unterschieden, sondern in einem unreflektierten Variablenbegriff konfundiert. Dies wurde bereits in bezug auf parametrische Modelle kritisiert.
Die hier skizzierte Differenzierung von Modellschemata, die noch Konstantensymbole besitzen, und Modellen, in denen alle Konstantensymbole durch Konstanten ersetzt worden sind, wurde schon durch die analoge Unterscheidung zwischen Netzschemata und Netzen vorweggenommen. Beide Unterscheidungen sind kohärent, weil Netze und ihre zugrundeliegenden Netzschemata in dieser Arbeit herangezogen werden, um prädikatenlogisch basierte Modelle für Flexible Fertigungssysteme zu entwerfen.
Darüber hinaus erlangt die Differenzierung zwischen Konstantensymbolen und Variablen auch Bedeutung, wenn prädikatenlogische Modelle mit Hilfe entsprechender Programmiersprachen in Automatischen Informationsverarbeitungssystemen implementiert werden sollen. Solche Programmiersprachen sehen im allgemeinen keine Konstantensymbole als formale Konstrukte sui generis vor. Vielmehr unterscheiden sie gewöhnlich nur zwischen Konstanten und Variablen. Daher können sie ohne Schwierigkeiten nur für die Implementierung konkreter Modelle herangezogen werden. In diesem eingeschränkten Anwendungsbereich erweisen sich die Programmiersprachen allerdings als bemerkenswert leistungsfähig. Denn sie stellen Mechanismen zur Verfügung, durch die auf der zweiten Interpretationsstufe alternative Variablenbindungsfunktionen für die Variablen von konkreten Modellen automatisch realisiert werden. Vgl. dazu die Anmerkungen zur Realisierung prädikatenlogischer Variablenbindungsfunktionen durch das Unifizierungskonzept der Programmiersprache PROLOG, die in dieser Arbeit für die Implementierung von Netzmodellen verwendet wird.
Falls dagegen der voranstehend skizzierte Bereich der Implementierung konkreter Modelle überschritten wird, um mit der Hilfe der Automatischen Informationsverarbeitung auch abstrakte Modellschemata zu implementieren, treten Schwierigkeiten auf. Da die üblichen Programmiersprachen keine Konstantensymbole kennen, müssen sie die Konstantensymbole eines Modellschemas durch Variablen ersetzten. Es resultiert ein abstraktes, "parametrisiertes" Modell, das sich von einem beliebigen konkreten Modell nicht formal unterscheiden läßt. Denn sowohl parametrisierte als auch konkrete Modelle bleiben in der gleichen Weise aus Konstanten und Variablen aufgebaut. Diese Vermengung von Konstantensymbolen und Variablen ist nicht nur formal bedenklich, sondern kann auch zu materiellen Fehlschlüssen verleiten. Ein Beispiel hierfür wurde bereits in einer früheren Anmerkung angeführt: Dort wurde die Ansicht vertreten, in konkreten Modellen müßten bereits alle Variablen durch ihre jeweils optimalen Werte ersetzt sein. Entscheidungs- und Handlungsalternativen könnten in solchen konkreten Modellen folglich überhaupt nicht mehr repräsentiert werden.

17) Ein variabler Term kann, muß aber nicht eine Variable darstellen. Denn ein variabler Term kann - neben seiner mindestens einen Variablen - ebenso Konstanten- und Operationssymbole enthalten. Letztes ist allerdings nur dann möglich, wenn es sich um einen variablen zusammengesetzten Term handelt. Ein variabler Basisterm fällt dagegen notwendig mit einer Variablen zusammen.

18) Es kann auch von einem konstanten Term gesprochen werden. Der Verf. bevorzugt aber die Bezeichnung "Grundterm". Auf diese Weise wird vermieden, einen variablenfreien Term mit jenen Konstanten zu assoziieren, die als formale Objekte "ob" eingeführt wurden. Jene formalen Objekte spielen für die hier definierten Terme *keine* Rolle. Denn die Terme sind allenfalls als Konstantensymbole definiert oder aus Konstantensymbolen zusammengesetzt. Diese Konstantensymbole sind aber als nullstellige Operationssymbole von allen Konstanten (formalen Objekten) wohlunterschieden.

19) Bei Termmengen, die nicht durch dieses Superskript als Mengen vollständig explizierter Terme ausgewiesen sind, wird der Explizierungsstatus ihrer Elemente nicht konkret festgelegt. Gleiches gilt für Terme, die dieses Superskript nicht tragen: Sie brauchen nicht, können aber vollständig expliziert sein.

20) Beispielsweise läßt sich der vollständig explizierte Term $te = Op_1(Op_2(Op_3(te_1), te_2, Op_4(te_3, te_4))$ 4. Stufe aus den Basistermen te_1, te_2, te_3 und te_4 einer Signatur SIG rekursiv gewinnen durch:

4. Stufe:	$te = Op_1(te_5, te_2, te_6) \in TE^4$
3. Stufe:	$te_5 = Op_2(te_7) \in TE^3$
2. Stufe:	$te_6 = Op_4(te_3, te_4) \in TE^2$, $te_7 = Op_3(te_1) \in TE^2$
1. Stufe:	$te_1, te_2, te_3, te_4 \in TE^1$ und $TE^1 = BT_{SIG}$

21) Dies ist z.B. der Fall, wenn später Termmengen für die formale Definition von Synthetischen Netzen benötigt werden. Vgl. auch die analoge Definition von Termen im prädikatenlogischen Kalkül an späterer Stelle.

22) Es muß auf die kleinste Familie Bezug genommen werden, weil die nachfolgenden Anforderungen von zahlreichen Mengenfamilien erfüllt werden, deren Mengen jeweils Obermengen der gesuchten Termfamilie sind. Jene Obermengen können neben Termen noch beliebige andere Konstrukte enthalten.

23) Vereinfacht läßt sich eine Termfamilie auch definieren als die kleinste Familie $TMF_{SIG}(VAF)$ von sortenspezifischen Termmengen $TM_i(VA_i)$, für die gilt:

- Jedes Konstantensymbol ist ein Term aus genau einer sortenspezifischen Termmenge $TM_i(VA_i)$.
- Jede Variable ist ein Term aus genau einer sortenspezifischen Termmenge $TM_i(VA_i)$.
- Wenn Op_j ein K_j-stelliges Operationssymbol mit $K_j \in \mathcal{N}_+$ ist und wenn $te_i, \ldots, te_{Kj}$ Terme sind, dann ist auch $te = Op_j(te_i, \ldots, te_{Kj})$ ein Term aus genau einer sortenspezifischen Termmenge $TM_i(VA_i)$.

24) Die sortenspezifische Konstantensymbolemenge KO_i kann auch leer sein.

25) Im Grenzfall $V_i = 0$ ist für die Sorte $sort_i$ überhaupt keine Variable definiert. Für die sortenspezifische Variablenmenge gilt dann: $VA_i = \emptyset$.

26) Vgl. MURATA,TA. (1988b), S. 483f. Auch dort werden Variablenmengen explizit verwendet. Sie werden benutzt, um prädikatenlogische Formelsysteme zu gestalten und auf Prädikat/Transition-Netze abzubilden.

27) Die taxonomische Inkonsistenz zwischen Informations- und Datenbegriff wird hier nicht aufgelöst. Der Verf. hält den Informationsbegriff für geeigneter. Er hat dies in ZELEWSKI (1986a), S. 101ff., insbesondere S. 104f., ausführlicher dargelegt. Dennoch handelt es sich im Fall abstrakter Datentypen um einen etablierten Terminus technicus. Daher verwendet der Verf. vornehmlich den Informationsbegriff, greift aber auf den Datenbegriff als Synonym überall dort zurück, wo der Bezug zu eingeführten sprachlichen Konventionen aufrechterhalten werden soll.

28) Vgl. zu dieser klaren Separation zwischen abstrakten und konkreten Datenbeschreibungen im Kontext der Theorie abstrakter Datentypen BEIERLE (1988a), S. 447.

29) Vgl. z.B. REISIG (1989a), S. 21. Er läßt für die Markierungen und Kantenbeschriftungen von Netzen Tupel aus *Konstanten und Variablen* zu, obgleich bei korrekter Ausdrucksweise von Konstanten*symbolen* gesprochen werden müßte. Dies wird im unmittelbaren Kontext (S. 21) deutlich, in dem REISIG für die Spezifikation dieser Tupel auf Variablen und 0-stellige Operationssymbole zurückgreift. Letzte stellen aber genau die erforderlichen Konstanten*symbole* dar.

30) Vor allem der Gebrauch von Konstantensymbolen und Operationssymbolen anstelle von Konstanten bzw. Funktionen trägt nach Erfahrungen des Verf. zur Verwirrung Dritter bei.

31) Zu diesem Zweck werden die Termauswertungen in einer Weise in zwei Komponenten aufgespalten, die für das Signaturkonzept selbst nicht bekannt ist.

4.1.5 SIG-Termauswertungen

Die beiden zuvor entfalteten Erweiterungen des Signaturkonzepts - die sortenorientierten Objektmengen und die operationssymbolbezogenen Termfamilien - werden durch das Konzept der Termauswertung zusammengeführt. Termauswertungen bilden Terme auf Komplexe aus formalen Objekten ab. In einer Zwischenstufe transformieren sie Terme in Kombinationen aus Variablen, Konstanten und Operatoren (Funktoren).

Definition: Termauswertung

Eine SIG-Termauswertung bezüglich der Signatur $SIG=(SO,OP)$ ist eine rekursiv definierte Funktionenfamilie $TA_{SIG}=(eval_i\colon i=1,\dots,I)$. Jede ihrer sortenspezifischen Auswertungsfunktionen $eval_i$ ist eine Abbildung der sortenspezifischen Termmenge $TM_i(VA_i)$ aus der SIG-Termfamilie $TMF_{SIG}(VAF)$ auf die sortenspezifische Objektmenge OB_i aus der SIG-Algebra A_{SIG}. Diese Auswertungsfunktionen wird rekursiv definiert durch[1]:

$$eval_i\colon \quad TM_i(VA_i) \rightarrow OB_i$$

$$te \rightarrow eval_i(te) = ob = \begin{cases} eval_{i.KO}(te); \\ \text{falls Term te ein Konstantensymbol ist mit} \\ te = Ko \text{ und } Ko \in KO_i \\[1ex] eval_{i.VA}(te); \\ \text{falls Term te eine Variable X ist mit} \\ te = X \text{ und } X \in VA_i \\[1ex] op_j(eval_{h(1)}(te_1),\dots,eval_{h(Kj)}(te_{Kj})); \\ \text{falls Term te ein zusammengesetzter Term ist mit} \\ te = Op_j(te_1,\dots,te_{Kj}),\ te \in ZT_i,\ Op_j \in OP_{SIG} \text{ und } K_j \in \mathcal{N}_+ \end{cases}$$

$$eval_{i.KO}\colon \quad KO_i \rightarrow OB_i$$
$$\qquad Ko \rightarrow eval_{i.KO}(Ko) = ob$$

$$eval_{i.VA}\colon \quad VA_i \rightarrow OB_i$$
$$\qquad X \rightarrow eval_{i.VA}(X) = ob$$

<u>Erläuterungen und Ergänzungen zur Termauswertungs-Definition:</u>

a) Auf einen Term te, der wegen $te \in TM_i(VA_i)$ zur Sorte $sort_i$ gehört, können nur Auswertungsfunktionen $eval_i$ derselben Sorte angewendet werden[2]. Es erfolgen also stets sortengerechte Termevaluierungen[3]. Da die Auswertungsfunktionen durch $eval_i\colon TM_i(VA_i) \rightarrow OB_i$ definiert sind, stellen die sortengerechten Termevaluierungen zugleich sicher, daß auch das Auswertungsergebnis - das zugeordnete formale Objekt ob - aus derselben Sorte $sort_i$ wie der evaluierte Term te stammt.

b) Für Terme derselben Sorte $sort_i$ steht in der Funktionenfamilie der Termauswertung TA_{SIG} mindestens eine Auswertungsfunktion $eval_i$ derselben Sorte zu Verfügung. Das formale Konzept der Funktionenfamilie läßt es zu, für dieselbe Sorte $sort_i$ auch mehrere Auswertungsfunktionen $eval_{i.q}$ mit $q \in \{1,..,Q_i\}$, $Q_i \in \mathcal{N}_+$ und $Q_i \geq 2$ zu definieren[4)]. Die sortengerechte Termauswertung kann daher eindeutig bestimmt sein, muß es aber nicht. Im Regelfall enthält eine Termauswertung TA_{SIG} für jede Sorte "sort" mehrere Auswertungsfunktionen $eval_i$. Im allgemeinen ist daher die Abbildung eines Terms te mit $te \in TM_i(VA_i)$ auf ein formales Objekt durch die Angabe einer Termauswertung TA_{SIG} unterbestimmt. Erst die konkrete Angabe einer Auswertungsfunktion $eval_i$ aus der Funktionenfamilie TA_{SIG} determiniert die Termabbildung auf ein formales Objekt eindeutig.

c) Die Abbildung eines Terms te aus der Menge $TM_i(VA_i)$ auf ein formales Objekt ob aus der Menge OB_i durch die Anwendung einer Auswertungsfunktion wird auch als Evaluation des Terms te oder als Ersetzung des Terms te durch das formale Objekt ob bezeichnet. Dabei kann die Applikation einer bestimmten Auswertungsfunktion $eval_i$ als $eval_i$-Evaluation des Terms te hervorgehoben werden. Die Ersetzung aller Terme te aus einer Termmenge durch Anwenden von Auswertungsfunktionen aus der Funktionenfamilie TA_{SIG} heißt eine Evaluation der jeweils betrachteten Termmenge. Falls die Evaluation auf die SIG-Termfamilie $TMF_{SIG}(VAF)$ bezogen wird, wird auch kurz von einer Termauswertung gesprochen[5)].

d) Die Definition der Auswertungsfunktion $eval_i$ befolgt ein rekursives Schema. Denn die Auswertungsfunktion $eval_i$ des Definiendums ist auch im Definiens - und zwar in dessen dritten Unterfall - enthalten[6)].

e) Durch die Auswertungsfunktion $eval_i$ wird jedem Term te mit $te \in TM_i(VA_i)$ aus der Sorte $sort_i$ ein sortengleiches formales Objekt ob mit $ob \in OB_i$ und $ob = eval_i(te)$ zugeordnet. Falls te einen Basisterm darstellt, erfolgt die Zuordnung des formalen Objekts ob unmittelbar durch eine der beiden Subfunktionen $eval_{i.KO}$ und $eval_{i.VA}$. Andernfalls wird jeder zusammengesetzte Term te mit $te = Op_j(te_1,..,te_{Kj})$ durch ein mehrstufiges rekursives Auswertungsschema auf das formale Objekt ob abgebildet. Durch die Subfunktionen $eval_{i.KO}$ und $eval_{i.VA}$ wird die Rekursionsbasis definiert. Die formale Struktur des Rekursionsschemas stellt sicher, daß der Wert der Auswertungsfunktion $eval_i$ für jedes Argument aus ihrem Vorbereich $TM_i(VA_i)$ in einem endlichen Evaluationsprozeß aus der Rekursionsbasis und einer endlichen Anzahl von Anwendungen des Rekursionsschemas berechnet werden kann[7)].

f) Das formale Objekt ob, das einem Term te durch sortengerechte Evaluation zugeordnet wird, kann entweder atomaren oder zusammengesetzten Charakter besitzen. Dies hängt jeweils davon ab, auf welche Weise der Term te und das Objekt ob in der zugrundeliegenden Signatur bzw. SIG-Algebra durch Operationssymbole bzw. Operationen definiert worden sind. Durch das Rekursionsschema der Termauswertung werden Basisterme - d.h. Konstantensymbole und Variablen - stets auf atomare formale Objekte abgebildet. Zusammengesetzten Termen werden zusammengesetzte formale Objekte in der Weise zugeordnet, daß jedem Operationssymbol Op_j aus einem Term im korrespondierenden formalen Objekt die Operation op_j entspricht. Daher müssen der Term te und das zugeordnete formale Objekt ob die gleiche Struktur besitzen. Die sortengerechte Evaluation von Termen erfolgt daher strukturerhaltend. Dies bedeutet erstens, daß unstrukturierte Basisterme auf atomare Objekte ohne innere Struktur abgebildet werden. Zweitens sind zusammengesetzte Terme und zusammengesetzte formale Objekte jeweils paarweise in der gleichen Weise zusammengesetzt.

g) Aus der Kombination von Sortengerechtigkeit und Strukturerhaltung folgt, daß die Evaluation von Termen deren Sortenstruktur[8)] vollständig erhält. Dieser Sachverhalt wird auch als Sortentreue von Termauswertungen bezeichnet[9)]. Der Aspekt der Sortengerechtigkeit garantiert nur an der "Termoberfläche", daß ein Term te aus der Sorte $sort_i$ mit $te \in TM_i(VA_i)$ auf ein formales

Objekt ob aus derselben Sorte mit $ob = eval_i(te)$ und $ob_i \in OB_i$ abgebildet wird. Er sagt aber nichts darüber aus, in welcher Weise bei zusammengesetzten Termen die Sorten der Subterme aus der internen Termstruktur berücksichtigt werden[10]. Erst durch Hinzunahme der Strukturerhaltung wird sichergestellt, daß die interne Termstruktur auf eine interne Struktur des zugeordneten komplexen formalen Objekts während der Evaluation so abgebildet wird, daß die Sortenstrukturen von zusammengesetztem Term und zusammengesetztem formalen Objekt einander entsprechen. Die Anwendung von Auswertungsfunktionen aus einer Termauswertung TA_{SIG} bedeutet daher eine sortentreue Abbildung von Termen aus der SIG-Termfamilie $TMF_{SIG}(VAF)$ auf formale Objekte aus der SIG-Algebra A_{SIG}. Hierin liegt der *erste wesentliche Gehalt* des Konzepts der SIG-Termauswertung.

h) Durch das Rekursionsschema der Termauswertung wird jede Termabbildung auf ein formales Objekt letztlich auf die Ersetzung von Basistermen (Konstantensymbolen und Variablen) durch atomare formale Objekte (Konstanten) zurückgeführt. Die Ersetzung eines Konstantensymbols durch eine Konstante ist aus dem Blickwinkel der praktischen Anwendung des Signaturkonzepts trivial. Sie besitzt nur die theoretische Bedeutung, aus dem abstrakten Bereich der SIG-Termfamilien in den konkreten Bereich der SIG-Algebren hinüberzuleiten. Dagegen spielt die sortentreue Zuordnung von Konstanten zu Variablen eine wesentliche Rolle. Hierauf wird später bei der Auswahl bestimmter Modi für das Ausführen von Übergangsoperationen ausführlich zurückgekommen. Die Anwendung einer Auswertungsfunktion auf einen Term wird deshalb auch als Belegung seiner Variablen mit Konstanten bezeichnet. Ebenso wird vom Substituieren oder Binden seiner Variablen durch Konstanten gesprochen. Der *zweite wesentliche Gehalt* von SIG-Termauswertungen erstreckt sich auf diese Variablensubstitution durch Konstanten.

i) Zwei sortengleiche Terme[11] te_1 und te_2 mit $te_1, te_2 \in TM_i$ werden als kongruent bezeichnet, wenn sie unter derselben Auswertungsfunktion auf dieselben formalen Objekte abgebildet werden. Die binäre Kongruenzrelation "$\approx$" ist definiert durch[12]:

$$te_1 \approx te_2 \;:\Leftrightarrow\; eval_i(te_i) = eval_i(te_2)$$

Die Termkongruenz kann sich sowohl auf den trivialen Fall $te_1 = te_2$ für identische Terme als auch auf den Regelfall $te_1 \neq te_2$ für verschiedene Terme beziehen[13].

j) Die termbezogene und zugleich sortenspezifische Auswertungsfunktion $eval_i$ läßt sich ohne Schwierigkeiten auf K_j-stellige Termtupel $(te_1,..,te_{Kj})$ mit $K_j \in \mathcal{N}_+$ erweitern. Dabei ist im Signaturkonzept jedes Termtupel $(te_1,..,te_{Kj})$ als zulässiges Argument eines K_j-stelligen Operationssymbols Op_j definiert. Durch die Sorten $sort_{i(j.k)}$ aus dem Argument dieses Operationssymbols Op_j mit $k \in \{1,...,K_j\}$ sind die Termmengen $TM_{i(j.k)}(VA_{i(j.k)})$ für die Komponenten te_k des Termtupels $(te_1,..,te_{Kj})$ festgelegt. Die kanonisch[14] erweiterte Auswertungsfunktion $eval_{Kj}$ ist nunmehr sortenunspezifisch, aber operationssymbolspezifisch definiert durch:

$$\begin{aligned} eval_j\colon\quad & TM_{i(j.1)}(VA_{i(j.1)}) \times ... \times TM_{i(j.Kj)}(VA_{i(j.Kj)}) \rightarrow OB_{i(j.1)} \times ... \times OB_{i(j.Kj)} \\ & (te_1,...,te_{Kj}) \rightarrow eval_j(te_1,...,te_{Kj}) = (eval_{i(j.1)}(te_1),...,eval_{i(j.Kj)}(te_{Kj})) \end{aligned}$$

k) Das Konzept der SIG-Termauswertung verknüpft nicht nur die früher eingeführten Konzepte der SIG-Termfamilie und der SIG-Algebra formal. Vielmehr läßt sich anhand von Termevaluationen ein neuartiges Auswertungskonzept einführen. Es bildet den wesentlichen Ansatzpunkt, aus dem heraus später eine formale operationale Logik für Synthetische Netze entwickelt wird. Diese Logik betrifft in ihrem Kern die Schaltregeln von Transitionen in Synthetischen Netzen. Für das Schalten dieser Transitionen interessiert, welche Marken[15] - als formale Objekte - die Aktivierungsbedingungen der jeweils betrachteten Transitionen erfüllen. Eine Antwort auf diese Frage wird durch das anschließend entfaltete neuartige Auswertungskonzept für SIG-Termfamilien und SIG-Algebren konzeptionell vorbereitet.

l) Die Termmengen $TM_i(VA_i)$ einer SIG-Termfamilie und die Objektmengen OB_i einer SIG-Algebra können zwar beliebig umfassend formuliert werden. Auch lassen sich die SIG-Termfamilie und das algebraische Universum intern beliebig strukturieren, indem auf die Sortensprache SOS_{SIG} der zugrundeliegenden Signatur SIG zurückgegriffen wird. Doch werden diese Term- bzw. Objektmengen bei jeder Entfaltung des Signaturkonzepts als irgendwie *gegeben* vorausgesetzt. In gleicher Weise könnte ebenso in bezug auf eine SIG-Termausweitung vorgegangen werden. Auch ihre Funktionenfamilie ließe sich als gegeben vorstellen. Von dieser Sichtweise wird hier jedoch durch das Auswertungskonzept der konstruktiven Existenzbeweise abgewichen.

m) Die SIG-Termauswertung wird statt dessen als eine offene, potentiell unendliche Funktionenfamilie vorgestellt. Sie umfaßt für jede Sorte $sort_i$ aus der Sortenmenge SO der zugrundeliegenden Signatur SIG *potentiell alle* Auswertungsfunktionen $eval_i$, die das o.a. rekursive Definitionsschema erfüllen. Daher ist die Evaluation von Termen hochgradig unterbestimmt. Auf jeden präsentierten Term te mit $te \in TM_i(VA_i)$ aus der SIG-Termfamilie $TMF_{SIG}(VAF)$ kann eine Vielzahl unterschiedlicher Auswertungsfunktionen $eval_i$ derselben Sorte angewendet werden.

n) Aus der Sicht des konventionellen algebraischen Auswertungskonzepts besteht ein kombiniertes Auswahl- und Ermittlungsproblem. Es gilt, aus einer Vielzahl von gegebenen sortengerechten Auswertungsfunktionen genau eine auszuwählen, diese auf den Term anzuwenden und hierdurch genau ein formales Objekt aus der vorgegebenen Objektmenge OB_i zu ermitteln, das dem Term zugeordnet ist. Charakteristisch für dieses konventionelle Auswertungskonzept ist es erstens, daß von *fest vorgegebenen* Auswertungsfunktionen ausgegangen wird. Zweitens ist die applizierte Auswertungsfunktion als Lösung des Auswahlproblems *genau bestimmt.* Drittens wird das Ermittlungsproblem durch *genau ein* formales Objekt für die Zuordnung zum präsentierten Term gelöst. Dieses Auswahl- und Ermittlungsproblem interessiert beim Auswertungskonzept des konstruktiven Existenzbeweises jedoch nicht[16)].

o) Vielmehr geht es jetzt um ein kombiniertes Existenz- und Konstruktionsproblem. Ausgangspunkt ist abermals ein präsentierter Term te mit $te \in TM_i(VA_i)$ aus der Termfamilie $TMF_{SIG}(VAF)$. Die sortengerechten Auswertungsfunktionen sind aber nicht fest vorgegeben, weil die offene Funktionenfamilie TA_{SIG} der SIG-Termauswertung prinzipiell unendliche viele solcher Funktionen enthalten kann[17)]. Daher ist das o.a. Auswahlproblem hier nicht wohldefiniert. Statt dessen besteht aber das Existenzproblem, ob für den präsentierten Term te mindestens ein formales Objekt ob in der vorgegebenen Objektmenge OB_i und mindestens eine Auswertungsfunktion $eval_i$ aus der offenen Funktionenfamilie TA_{SIG} in der Weise existieren, daß $ob = eval_i(te)$ gilt.

p) Eine negative Lösung dieses Existenzproblems besteht in dem Beweis, daß keine Auswertungsfunktion und kein formales Objekt mit den voranstehend definierten Eigenschaften existieren. Die Familie TA_{SIG} aller zulässigen Auswertungsfunktionen schließt infolge ihrer Offenheit keine denkmöglichen Auswertungsfunktionen aus[18)]. Daher kann die negative Lösung des Existenzproblems hier nur dadurch verursacht sein, daß sich in der gegebenen Objektmenge OB_i kein "passendes" formales Objekt befindet, das sich dem präsentierten Term te durch irgendeine zulässige Auswertungsfunktion zuordnen ließe[19)].

q) Eine positive Lösung des Existenzproblems weist dagegen nach, daß mindestens eine Auswertungsfunktion existiert, die den präsentierten Term auf ein formales Objekt aus der vorgegebenen Objektmenge abbildet. Ein solcher Existenzbeweis ist aber nicht notwendig konstruktiv. Es kann zwar bekannt sein, daß ein solches Paar aus Auswertungsfunktion und formalem Objekt *existiert*, ohne aber zu wissen, um *welche* Auswertungsfunktion und *welches* formales Objekt es sich handelt[20]. Daher wird das Existenzproblem, sofern es sich positiv lösen läßt, mit einem Konstruktionsproblem kombiniert.

r) Das Konstruktionsproblem besteht aus der Aufgabe, *mindestens ein* formales Objekt ob anzugeben, das durch die Anwendung mindestens einer Auswertungsfunktion $eval_i$ auf den präsentierten Term te als $ob = eval_i(te)$ resultiert (schwaches Konstruktionsproblem). Dieses Konstruktionsproblem kann verschärft werden durch die Anforderung, *alle* formalen Objekte aus der vorgegebenen Objektmenge aufzufinden, welche die voranstehende Spezifizierung erfüllen (starkes Konstruktionsproblem). In beiden Fällen interessiert aus der Perspektive des hier formulierten Konstruktionsproblems nicht, welche Auswertungsfunktion(en) jeweils die Zuordnung eines formalen Objekts zum präsentierten Term ermöglicht hat (haben)[21]. Hinsichtlich der Auswertungsfunktion reicht der Existenzbeweis aus. Nur bezüglich des mindestens einen zugeordneten formalen Objekts interessiert die konstruktive Angabe dieses Objekts.

s) Das Auswertungskonzept des konstruktiven Existenzbeweises unterscheidet sich vom konventionellen Auswertungskonzept also in drei charakteristischen Veränderungen: Erstens werden die Auswertungsfunktionen nicht als fest vorgegeben betrachtet. Statt dessen können sie innerhalb einer *offenen* Funktionenfamilie jede denkmögliche (rekursive) Gestalt annehmen. Zweitens bleibt im Falle einer positiven Lösung von Existenz- und Konstruktionsproblem die *Auswertungsfunktion* selbst *unbestimmt*; es besteht kein Interesse, ihre konkrete Gestalt kennenzulernen[22]. Drittens wird nicht genau ein formales Objekt für die Zuordnung zum präsentierten Term gesucht, sondern entweder *mindestens ein* formales Objekt oder *alle* formalen Objekte, für das bzw. für die eine Zuordnung aus der vorgegebenen Objektmenge möglich ist.

t) Das voranstehend skizzierte Auswertungskonzept auf der Basis konstruktiver Existenzbeweise kann ohne *grundsätzliche* theoretische Schwierigkeiten realisiert werden. Realisierungsinstrument ist im wesentlichen das prädikatenlogische Unifizierungskonzept[23]. Es stellt die wesentliche Grundlage des hier skizzierten neuartigen Auswertungskonzepts dar. Mit seiner Hilfe lassen sich Termevaluationen durch konstruktive Existenzbeweise so ausführen, daß nur rein formale Beweistechniken eingesetzt zu werden brauchen. Dies hat den praktischen Vorteil, die Termauswertungen mit Hilfe der Automatischen Informationsverarbeitung implementieren zu können.

u) Im Unifizierungskonzept wird allerdings auf Auswertungsfunktionen der oben vorgestellten Art überhaupt kein Bezug genommen. Statt dessen werden die Argumente von teilevaluierten prädikatenlogischen Formeln betrachtet. Es wird versucht, Variablen[24] aus den Formelargumenten so durch neue teilevaluierte Terme zu ersetzen, daß die betroffenen Formelargumente identisch ausfallen[25]. Dabei werden die betroffenen Variablen oftmals durch Konstanten ersetzt. Sie können aber auch durch andere Variablen substituiert werden. Ebenso kommen formale Ausdrücke in Betracht, die aus Konstanten, Variablen und Funktoren zusammengesetzt sind[26]. Bei dieser Variablensubstitution spielen Auswertungsfunktionen überhaupt keine Rolle[27]. Dennoch lassen sich die Unifizierungsoperationen so anwenden, daß sie zu den gleichen Substitutionsergebnissen führen, die aus dem Gebrauch von Auswertungsfunktionen resultieren würden[28].

v) Trotz seiner eminenten Bedeutung für Termevaluationen wird das Unifizierungskonzept hier aus zwei Gründen nicht im Detail behandelt[29]. Einerseits erlangt das Unifizierungskonzept seine volle Wirksamkeit erst dann, wenn es in das umfassendere Resolutionskonzept[30] der prädikatenlogischen Beweistheorie eingebettet wird. Diese Konzeptkombination[31] verhält sich zwar im Prinzip[32] vollständig und korrekt[33]. Doch erweist sie sich hinsichtlich ihrer formalen Ausarbeitung als intuitiv schwer zugänglich und relativ aufwendig[34]. Andererseits verfügen alle Varianten der Programmiersprache PROLOG über Implementierungen der Kombination aus Resolutions- und Unifizierungskonzept[35]. Denn diese Konzeptkombination stellt die Basis des logischen Programmierstils schlechthin dar, zu dessen Realisierung die Sprache PROLOG entwickelt wurde[36]. Somit gestatten die PROLOG-Varianten, die konstruktiven Existenzbeweise des Unifizierungsansatzes automatisch ausführen zu lassen[37].

w) Die Programmiersprache PROLOG enthält in ihrer fortentwickelten Version PROLOG II sogar ein besonders komfortables reserviertes Prädikat "val(te_1,te_2)". Es gestattet, den ersten Term te_1[38] aus seinem Argument automatisch auswerten zu lassen und das Auswertungsergebnis in den zweiten Term te_2 für weitere Zugriffe zu schreiben[39].

x) Darüber hinaus kann die Unifizierung bei einer PROLOG-Implementierung sortentreu erfolgen. Die Sortentreue erfordert allerdings, daß der PROLOG-Dialekt es zuläßt, für die jeweils auszuwertenden Terme unterschiedliche Sorten festzulegen. Dies ist für die meisten PROLOG-Varianten, die zur Zeit vorherrschen, noch nicht möglich[40]. Aber der Dialekt Turbo-PROLOG, der in dieser Arbeit ausgewählt wurde, sieht in der "domains"-Sektion eine Deklaration der Termsorten vor[41]. Auf dieser Grundlage vollzieht Turbo-PROLOG sortentreue Unifizierungen. Dieser Sachverhalt war ein wesentlicher Bestimmungsgrund für die Konzeptualisierungsentscheidung, die später eingeführte Notation von Synthetischen Netzen - insbesondere die Netzlegenden - an die Struktur von Turbo-PROLOG-Programmen anzulehnen.

y) Die voranstehenden Anmerkungen verdeutlichen, welch große Bedeutung der später entfalteten prädikatenlogischen Implementierung von Synthetischen Netzen auf PROLOG-Basis zukommt[42]. Diese Implementierung des Unifizierungskonzepts gestattet es, die beträchtliche Ausdruckskraft des algebraischen Signaturkonzepts durch praktische Auswertungen von Netzmodellen auch konkret zu nutzen. Die PROLOG-basierte Netzimplementierung schlägt somit eine Brücke: Sie vermittelt zwischen der primär interessierenden Mächtigkeit des hier entfalteten netzgestützten Modellierungskonzepts einerseits und seiner praktischen Anwendungseffizienz andererseits[43]. Zugleich kann wegen der PROLOG-basierten Implementierung des Unifizierungskonzepts darauf verzichtet werden, seine algorithmischen Details hier vertieft zu erörtern.

z) Im Signaturkonzept wird im allgemeinen klar unterschieden zwischen einerseits Termen, die aus Konstantensymbolen, Variablen und Operationssymbolen bestehen, sowie andererseits formalen Objekten, die aus Konstanten und Operatoren zusammengesetzt sind. Bei der Kombination des Signaturkonzepts mit anderen formalen Konzepten kann es jedoch erforderlich sein, diese scharfe Differenzierung aufzuheben. Denn jene Konzepte räumen den Variablen eine Zwitterstellung zwischen Termen und formalen Objekten derart ein, daß sie bei einer Termauswertung nicht vollständig durch Konstanten ersetzt zu werden brauchen. Die Ergebnisse solcher unvollständigen Termauswertungen sind dann formalsprachliche Ausdrücke, in denen Konstanten und Operationen als Konstituenten formaler Objekte mit Variablen als Konstituenten von Termen koexistieren. Von dieser Möglichkeit wird später bei der Überlagerung der Prädikatenlogik durch eine Sortenstruktur Gebrauch gemacht. Ebenso wird sie vom Übergangsschema für Operationen, die Faktenmengen ineinander transformieren können, vorausgesetzt. Schließlich wird auf diese Möglichkeit auch bei der Formulierung der Schaltregel von Synthetischen Netzen zurückgegriffen. Formal konkretisiert wird diese Option einer besonderen Variablenbehandlung durch das Konzept der Teilevaluierung.

A) Unter der Teilevaluierung[44] eines Terms te aus der Sorte $sort_i$ mit $te \in TM_i$ wird die sortengerechte Anwendung einer Teilauswertungsfunktion verstanden. Eine Teilauswertungsfunktion wird qualitativ charakterisiert durch folgende Eigenschaften:

- alle Konstantensymbole werden auf Konstanten abgebildet[45];
- alle K-stelligen Operationssymbole mit $K \in \mathcal{N}_+$ werden durch entsprechende Operationen ersetzt;
- Variablen können, müssen aber nicht durch Konstanten ersetzt werden.

Im allgemeinen werden im Rahmen der Teilevaluierung nicht alle Variablen eines Terms durch Konstanten ersetzt. Das Teilevaluierungsergebnis ist dann ein Ausdruck, der aus Konstanten und Variablen sowie aus Zusammensetzungen von Variablen, Konstanten und Operatoren bestehen kann. Das Ergebnis der Anwendung einer Teilauswertungsfunktion auf einen Term "te" wird als ein teilevaluierter Term "te'" bezeichnet.

B) Jeder teilevaluierte Term kann einerseits noch Variablen enthalten, andererseits aber ebenso formale Objekte umfassen. Daher gehört er im allgemeinen Fall weder zur sortenspezifischen Termmenge $TM_i(VA_i)$ noch zur sortenspezifischen Objektmenge OB_i. Vielmehr sind in einem teilevaluierten Term Aspekte der beiden vorgenannten Mengen miteinander vermengt. Daher wird die Familie $TTMF_{SIG}(VAF)$ aller teilevaluierten Terme, die durch Anwenden von Teilauswertungsfunktionen auf Terme $te \in TM_i(VA_i)$ für alle Sorten $sort_i$ einer Signatur SIG mit $i \in \{1,...,I\}$ gebildet werden können, separat - aber wiederum induktiv[46] - definiert:

- Jede Konstante, die ein atomares formales Objekt ob der Sorte $sort_i$ darstellt, ist ein teilevaluierter Term der Sorte $sort_i$. Die Menge aller Konstanten der Sorte $sort_i$ ist zunächst eine Menge OB_i aus allgemein definierten formalen Objekten. Falls mindestens eine nullstellige Operation op_j existiert, deren Nachbereich die Objektmenge OB_i ist, wird die Menge aller Konstanten der Sorte $sort_i$ auf eine Menge KB_i[47] von speziell definierten Objekten eingeschränkt. Es handelt sich dann um die Bildmenge aller nullstelligen Operationen aus der Familie OPF, die denselben Nachbereich OB_i besitzen.
- Jede Variable, die zur Variablenmenge VA_i der Sorte $sort_i$ gehört, ist ein teilevaluierter Term der Sorte $sort_i$.
- Jede K_j-stellige Operation op_j, die mit $K_j \in \mathcal{N}_+$ einem K_j-stelligen Operationssymbol $Op_j: sort_{i(1)} \dots sort_{i(Kj)} \rightarrow sort_{i(Kj+1)}$ zugeordnet ist, bildet jedes K_j-Tupel $(te_1,...te_{Kj})$ aus teilevaluierten Termen te_k, die mit $k \in \{1,...,K_j\}$ jeweils zur Sorte $sort_{i(k)}$ gehören, auf einen teilevaluierten Term te_{Kj+1} mit $te_{Kj+1} = op_j(te_1,...,te_{Kj})$ der Sorte $sort_{i(Kj+1)}$ ab.
- $TTM_i(VA_i)$ ist die Menge aller teilevaluierten Terme der Sorte $sort_i$.
- $VAF = (VA_i: i = 1,...,I)$ ist die Familie aller sortenspezifischen Variablenmengen für die Signatur SIG.
- $TTMF_{SIG}(VAF) = (TTM_i(VA_i): i = 1,...,I)$ ist die Familie aller sortenspezifischen Mengen $TTM_i(VA_i)$ teilevaluierter Terme für die Signatur SIG.

Die Schreibweise $te_{Kj+1} = op_j(te_1,...,te_{Kj})$ weist darauf hin, daß die operation op_j auf ein Argument aus teilevaluierten Termen angewendet wird. Die vormals eingeführte Notation $te_{Kj+1} = Op_j(te_1,..., te_{Kj})$ verdeutlicht dagegen, daß das Operationssymbol Op_j auf ein Argument aus Termen appliziert wird, die noch keiner (Teil-)Evaluierung unterworfen worden sind.

C) Als Grenzfälle der Teilevaluierung werden die beiden denkmöglichen Antipoden maximaler und minimaler Variablensubstitution zugelassen. Bei maximaler Substitution werden alle[48] Variablen des evaluierten Terms durch Konstanten ersetzt (Vollevaluierung). In diesem Grenzfall fallen gewöhnliche Auswertungsfunktionen $eval_i$ und Teilauswertungsfunktionen $teval_i$ zusammen. Als Evaluierungsergebnis resultiert dann ein formales Objekt, das entweder eine Konstante darstellt oder aber aus mindestens einer Konstanten und mindestens einem Operator zusammengesetzt ist. In diesem Sonderfall einer vollständigen Variablensubstitution stellt der teilevaluierte "Term" ein formales Objekt aus der Menge OB_i dar. Bei minimaler Substitution bleiben dagegen alle Variablen des evaluierten Terms erhalten[49]. Falls keine ausdrücklich abweichenden Festlegungen erfolgen, wird unter einer Teilevaluierung fortan stets der Grenzfall *minimaler* Variablensubstitution verstanden.

D) Speziell für den Fall minimaler Variablensubstitution wird eine rein formale Definition für die teilevaluierte SIG-Termfamilie $TTMF_{SIG}(VAF)$ vereinbart. Sie lehnt sich eng an die rein formale Definition der SIG-Termfamilie $TMF_{SIG}(VAF)$ an, die im vorigen Kapitel für nichtevaluierte Terme eingeführt wurde. Die beiden Definitionen weichen nur insofern voneinander ab, als bei einer minimalen Variablensubstitution von Operationssymbolen Op_j zu Operationen op_j und von Konstantensymbolen zu Konstanten übergegangen wird. Entsprechend wird die früher vorgestellte Sektion "Terms" durch die nachfolgend eingeführte Sektion "terms" ersetzt. Die Festlegung der Variablen in der Sektion "Vars" bleibt dagegen qua Voraussetzung unberührt. Die Familie $TTMF_{SIG}(VAF)$ aller sortenspezifischen Mengen $TTM_i(VA_i)$ teilevaluierter Terme für die Signatur SIG bei minimaler Variablensubstitution läßt sich daher definieren durch:

vars:

$VA_1 = \{X_{1.1},...,X_{1.V1}\}$

...

$VA_I = \{X_{I.1},...,X_{I.VI}\}$

$VAF = (VA_i: i=1,...,I)$

terms:

$KB_1 = \{te: \exists(op_j \in OPF): op_j: \rightarrow OB_1 \wedge op_j() = te\}$

$TTM_1(VA_1)$ minimal mit:

$KB_1 = \emptyset \rightarrow OB_1 \subseteq TTM_1(VA_1)$

$KB_1 \neq \emptyset \rightarrow KB_1 \subseteq TTM_1(VA_1)$

$VA_1 \subseteq TTM_1(VA_1)$

$(\forall(op_j \in OPF): (op_j: OB_{i(j.1)} \times ... \times OB_{i(j.KJ)} \rightarrow OB_{i(j.Kj+1)}$

$\wedge (\forall(k \in \{1,...,K_j\}): te_k \in TTM_{i(j.k)}(VA_{i(j.k)}))$

$\wedge \iota(j.K_j+1) = 1 \wedge op_j(te_1,...,te_{KJ}) = te_{KJ+1})$

$\rightarrow te_{KJ+1} \in TTM_1(VA_1))$

...

$KB_I = \{te: \exists(op_j \in OPF): op_j: \rightarrow OB_I \wedge op_j() = te\}$

$TTM_I(VA_I)$ minimal mit:

$KB_I = \emptyset \rightarrow OB_I \subseteq TTM_I(VA_I)$

$KB_I \emptyset \rightarrow KB_I \subseteq TTM_I(VA_I)$

$VA_I \subseteq TTM_I(VA_I)$

$(\forall(op_j \in OPF): (op_j: OB_{i(j.1)} \times ... \times OB_{i(j.KJ)} \rightarrow OB_{i(j.Kj+1)}$

$\wedge (\forall(k \in \{1,...,K_j\}): te_k \in TTM_{i(j.k)}(VA_{i(j.k)}))$

$\wedge \iota(j.K_j+1) = I \wedge op_j(te_1,...,te_{KJ}) = te_{KJ+1})$

$\rightarrow te_{KJ+1} \in TTM_I(VA_I))$

$TTMF(VAF) = (TTM_i(VA_i): i=1,...,I)$

E) Formal werden die sortenspezifischen Teilauswertungsfunktionen $teval_i$ durch folgendes Schema[50] definiert, das - abgesehen von der Wahlfreiheit bei der Variablensubstitution - aus der oben eingeführten sortenspezifischen Auswertungsfunktion $eval_i$ abgeleitet ist:

$$teval_i\colon\quad TM_i(VA_i) \rightarrow TTM_i(VA_i)$$

$$te \rightarrow teval_i(te) = te' = \begin{cases} teval_{i.KO}(te); \\ \text{falls Term te ein Konstantensymbol ist mit} \\ te = Ko \text{ und } Ko \in KO_i \\ \\ teval_{i.VA}(te); \\ \text{falls Term te eine Variable X ist mit} \\ te = X \text{ und } X \in VA_i \\ \\ op_j(teval_{h(1)}(te_1),\dots,teval_{h(Kj)}(te_{Kj})); \\ \text{falls Term te ein zusammengesetzter Term ist mit} \\ te = Op_j(te_1,..,te_{Kj}),\ te \in ZT_i,\ Op_j \in OP_{SIG} \text{ und } K_j \in \mathcal{N}_+ \end{cases}$$

$$teval_{i.KO}\colon\quad KO_i \rightarrow OB_i$$
$$Ko \rightarrow teval_{i.KO}(Ko) = ob$$

$$teval_{i.VA}\colon\quad VA_i \rightarrow (VA_i \cup OB_i)$$
$$X \rightarrow teval_{i.VA}(X) \in \{X, ob\}$$

F) Im Grenzfall der minimalen Variablensubstitution wird keine Variable X durch eine Konstante $ob \in OB_i$ ersetzt. Dann vereinfacht sich die variablenbezogene Teilauswertungsfunktion $teval_{i.VA}$ zur identischen Abbildung "id":

$$teval_{i.VA}\colon\quad VA_i \rightarrow VA_i$$
$$X \rightarrow teval_{i.VA}(X) = id(X) = X$$

Teilauswertungsfunktionen $teval_i$ mit minimaler Variablensubstitution werden als minimale Teilauswertungsfunktionen bezeichnet und als $xeval_i$ notiert[51]. Für sie gilt die vereinfachte Definition:

$$xeval_i\colon\quad TM_i(VA_i) \rightarrow TTM_i(VA_i)$$

$$te \rightarrow xeval_i(te) = te' = \begin{cases} xeval_{i.KO}(te); \\ \text{falls Term te ein Konstantensymbol ist mit} \\ te = Ko \text{ und } Ko \in KO_i \\ \\ id(te) = te = X; \\ \text{falls Term te eine Variable X ist mit} \\ te = X \text{ und } X \in VA_i \\ \\ op_j(xeval_{h(1)}(te_1),\dots,xeval_{h(Kj)}(te_{Kj})); \\ \text{falls Term te ein zusammengesetzter Term ist mit} \\ te = Op_j(te_1,..,te_{Kj}),\ te \in ZT_i,\ Op_j \in OP_{SIG} \text{ und } K_j \in \mathcal{N}_+ \end{cases}$$

$$xeval_{i.KO}\colon\quad KO_i \rightarrow OB_i$$
$$Ko \rightarrow xeval_{i.KO}(Ko) = ob$$

G) Die kanonische Erweiterung $teval_j$ der sortenspezifischen Teilauswertungsfunktionen $teval_i$ auf K_j-stellige Termtupel $(te_1,...,te_{Kj})$ mit $K_j \in \mathcal{N}_+$ ist analog zur kanonischen Erweiterung $eval_j$ der sortenspezifischen Auswertungsfunktionen $eval_i$ definiert. Für die resultierende sortenunspezifische, aber operationssymbolspezifische und termtupelbezogene Teilauswertungsfunktion $teval_j$ gilt:

$$\begin{aligned} teval_j\text{:}\quad & TM_{i(j.1)}(VA_{i(j.1)}) \times ... \times TM_{i(j.Kj)}(VA_{i(j.Kj)}) \\ & \qquad \rightarrow TTM_{i(j.1)}(VA_{i(j.1)}) \times ... \times TTM_{i(j.Kj)}(VA_{i(j.Kj)}) \\ & (te_1,...,te_{Kj}) \rightarrow teval_j(te_1,...,te_{Kj}) = (teval_{i(j.1)}(te_1),...,teval_{i(j.Kj)}(te_{Kj})) \end{aligned}$$

Eine kanonisch erweiterte minimale Teilauswertungsfunktion $xeval_j$ ist analog definiert:

$$\begin{aligned} xeval_j\text{:}\quad & TM_{i(j.1)}(VA_{i(j.1)}) \times ... \times TM_{i(j.Kj)}(VA_{i(j.Kj)}) \\ & \qquad \rightarrow TTM_{i(j.1)}(VA_{i(j.1)}) \times ... \times TTM_{i(j.Kj)}(VA_{i(j.Kj)}) \\ & (te_1,...,te_{Kj}) \rightarrow xeval_j(te_1,...,te_{Kj}) = (xeval_{i(j.1)}(te_1),...,xeval_{i(j.Kj)}(te_{Kj})) \end{aligned}$$

H) Das Komplement zu einer sortenspezifischen (minimalen) Teilauswertungsfunktion $teval_i$ ($xeval_i$) und ihrer kanonischen Erweiterung $teval_j$ ($xeval_j$) wird als komplementäre Teilauswertungsfunktion - oder kurz: Komplementauswertungsfunktion - $keval_i$ bzw. $keval_j$ bezeichnet. Die Komplementauswertungsfunktion ist so definiert, daß folgende Identität gilt[52)]: Wenn zunächst eine (minimale) Teilauswertungsfunktion auf ein Argument angewandt wird und danach auf das Resultat dieser Teilevaluierung die Komplementauswertungsfunktion appliziert wird, dann ist das Ergebnis der zweistufigen Auswertung identisch mit der Anwendung einer Auswertungsfunktion auf das ursprüngliche Argument. Das Komplement einer (minimalen) Teilauswertungsfunktion vervollständigt diese also immer auf eine Vollevaluierung.

I) Falls das Ergebnis einer Teilevaluierung ein Ausdruck[53)] war, der noch mindestens eine Variable enthielt, so bedeutet die Anwendung der Komplementauswertungsfunktion auf diesen variablen Ausdruck eine vollständige, sortengerechte Ersetzung aller seiner Variablen durch Konstanten. Die Komplementauswertungsfunktion ist dann eine vollständige Variablenersetzung für einen teilevaluierten Ausdruck. Dies gilt insbesondere bei der Teilevaluation durch minimale Teilauswertungsfunktionen. Damit entspricht die Komplementauswertungsfunktion der Zuweisungsfunktion[54)], die mitunter im Rahmen des Signaturkonzepts eingeführt wird, um Variablen in Termen auf Konstanten abzubilden. Wenn dagegen die Teilevaluierung als Grenzfall einer Vollevaluierung erfolgte, hat sie notwendig einen vollevaluierten, konstanten Ausdruck hervorgebracht. Dann ist die Komplementauswertungsfunktion die identische Abbildung "id".

J) Formal gilt für die sortenspezifischen Komplementauswertungsfunktionen $keval_i$, die über teilevaluierten Termen te' mit $TTM_i(VA_i)$ definiert sind, und ihre kanonischen Erweiterungen $keval_j$ auf K_j-Tupel aus teilevaluierten Termen:

$$keval_i: \ TTM_i(VA_i) \rightarrow OB_i$$

$$te' \rightarrow keval_i(te') = ob = \begin{cases} te'; & \text{falls der teilevaluierte Term te' eine Konstante} \\ & \text{ist mit } te' = op_j() = ob \text{ und } ob \in OB_i \\ keval_{i.VA}(te'); & \text{falls der teilevaluierte Term te' eine Variable X} \\ & \text{ist mit } te' = X \text{ und } X \in VA_i \\ op_j(keval_{h(1)}(te_1), \ldots, keval_{h(Kj)}(te_{Kj})); & \text{falls der teilevaluierte Term te' ein zusammengesetzter Term} \\ & \text{ist mit } te' = op_j(te_1, .., te_{Kj}),\ te' \in ZT_i \text{ und } K_j \in \mathcal{N}_+ \end{cases}$$

$$keval_{i.VA}: \ VA_i \rightarrow OB_i$$
$$X \rightarrow keval_{i.VA}(X) = ob$$

mit:

$$\forall (te \in TM_i(VA_i)) \ \forall (eval_i: TM_i(VA_i) \rightarrow OB_i): \ldots$$
$$(teval_i: TM_i(VA_i) \rightarrow TTM_i(VA_i) \wedge teval_i(te) = te'$$
$$\wedge \ keval_i: TTM_i(VA_i) \rightarrow OB_i \wedge keval_i(te') = ob)$$
$$\rightarrow \ eval_i(te) = keval_i(teval_i(te)) = ob \qquad [55)]$$

sowie:

$$keval_j: \ TTM_{i(j.1)}(VA_{i(j.1)}) \times \ldots \times TTM_{i(j.Kj)}(VA_{i(j.Kj)}) \rightarrow OB_{i(j.1)} \times \ldots \times OB_{i(j.Kj)}$$
$$(te_1, \ldots, te_{Kj}) \rightarrow keval_j(te_1, \ldots, te_{Kj}) = (keval_{i(j.1)}(te_1), \ldots, keval_{i(j.Kj)}(te_{Kj}))$$

J) Durch das Konzept der Teilevaluierung wird die SIG-Termfamilie $TMF_{SIG}(VAF)$ auf die SIG-Ausdrucksmenge $AM_{SIG}(VAF)$ erweitert. Jedes Element aus der SIG-Ausdrucksmenge ist ein formalsprachlicher Ausdruck, der sich aus einer zugrundeliegenden Signatur SIG in zulässiger Weise ableiten läßt. Dabei kann auf die SIG-Termfamilie $TMF_{SIG}(VAF)$, auf das algebraische Universum U_{SIG} der zugehörigen SIG-Algebra A_{SIG} sowie auf Teilauswertungsfunktionen $teval_i$ und deren kanonischen Erweiterungen teval zurückgegriffen werden. Ein zulässig abgeleiteter Ausdruck[56)] kann ein Element aus der SIG-Termfamilie $TMF_{SIG}(VAF)$ oder die Teilevaluierung eines solchen Terms darstellen. Ebenso kann es sich um ein K-stelliges Tupel solcher Terme bzw. teilevaluierten Terme handeln. Da das Konzept der Teilevaluierung die Vollevaluierung als Grenzfall umschließt, ist auch jedes formale Objekt aus der SIG-Algebra A_{SIG} ein zulässiger Ausdruck aus der Menge AM_{SIG}. Andere zulässige Ausdrücke als die voranstehend aufgeführten existieren in der SIG-Ausdrucksmenge nicht. Alle Ausdrücke aus der Menge $AM_{SIG}(VAF)$ werden auch kurz als SIG-Ausdrücke bezeichnet.

Anmerkungen zum Kapitel:

1) Vgl. EHRIG (1985a), S. 21; KRÄMER (1987a), S. 275; RECK (1988), S. 46; REISIG (1989a), S. 5f. Der Verf. benutzt nicht die Auswertungsfunktion, wie sie in den o.a. Quellen von EHRIG, RECK und REISIG originär definiert wird. Jene Auswertungsfunktion i.e.S. erstreckt sich nur auf Grundterme. Um beliebige Terme erfassen zu können, die auch Variablen enthalten dürfen, müssen jene Autoren zusätzlich eine erweiterte Zuweisungsfunktion festlegen. Diese Funktionsfestlegung benötigt zu ihrer vollständigen Herleitung drei Funktionsbegriffe (Auswertungsfunktion, Zuweisungsfunktion und erweiterte Zuweisungsfunktion). Darüber hinaus sind die Definitionen von Auswertungs- und erweiterter Zuweisungsfunktion partiell redundant. Diese Schwierigkeiten vermeidet der Verf. durch die gleichwertige, aber kompaktere Definition einer Auswertungsfunktion i.w.S., die auf beliebige Terme angewendet werden kann. Damit folgt er dem umfassenden Definitionsansatz für Auswertungsfunktionen bei KRÄMER (1987a), S. 275.

2) Die sortengerechte Anwendung einer Auswertungsfunktion $eval_i$ auf einen Term te mit $te \in TM_i(VA_i)$ wird auch kurz als Applikation einer Auswertungsfunktion auf den Term te bezeichnet. Die Sortengerechtigkeit der Funktionsanwendung braucht also nicht explizit erwähnt zu werden, weil sie per definitionem immer erfüllt wird.

3) Die Begriffe der Evaluierung und der Auswertung werden synonym verwendet.

4) Für $Q_i = 1$ ist für eine Sorte $sort_i$ nur genau eine Auswertungsfunktion $eval_{i.1} = eval_i$ definiert.

5) Daher erstreckt sich der Ausdruck "Termauswertung" - vereinfachend - sowohl auf die SIG-Termauswertung, die eine Funktionenfamilie darstellt, als auch auf die Anwendung von Mitgliedern dieser Funktionenfamilie auf alle Terme einer SIG-Termfamilie.

6) Die ersten beiden Unterfälle verweisen auf die Rekursionsbasis. Die konstantensymbol- und variablenbezogenen Auswertungsfunktionen $eval_{i.KO}$ bzw. $eval_{i.VA}$ sind ohne weitere Bezugnahme auf andere Auswertungsfunktionen originär definiert. Im dritten Unterfall muß zwar die Auswertungsfunktion $eval_i$ nicht unbedingt auf sich selbst bezogen sein. Denn die Auswertungsfunktionen $eval_{h(k)}$ mit $k \in \{1,...,K_j\}$ können jeweils wegen $sort_{h(k)} \neq sort_i$ von der Auswertungsfunktion $eval_i$ verschieden sein. Dennoch liegt hier eine echte Selbstbezüglichkeit des Definiendums vor. Einerseits ist es möglich, daß die zu definierende Auswertungsfunktion $eval_i$ im Definiens $op_j(eval_{h(1)}(te_i),...,eval_{h(Kj)}(te_{Kj}))$ als Auswertungsfunktion $eval_{h(k)}$ mit $sort_{h(k)} = sort_i$ für mindestens ein $k \in \{1,...,K_j\}$ tatsächlich vorkommt. Zweitens werden alle anderen Auswertungsfunktionen $eval_{h(k)}$ mit $sort_{h(k)} \neq sort_i$ ihrerseits durch Definitionen eingeführt, welche das gleiche rekursive Schema befolgen wie die Definition der betrachteten Auswertungsfunktion $eval_i$. Daher nehmen im Gesamtkomplex der Funktionenfamilie TA_{SIG} insgesamt alle Auswertungsfunktionen $eval_i$ für alle Sorten $sort_i \in SO$ gegenseitig auf sich Bezug. Daher ist zumindest die Definitionsgesamtheit der Termauswertung TA_{SIG} für alle Auswertungsfunktionen eine im strengen Sinne selbstbezügliche rekursive Definition. Da hier die Definition einer einzelnen Auswertungsfunktion $eval_i$ für eine *beliebige* Sorte $sort_i \in SO$ zugleich das allgemeine Definitions*schema* für alle Auswertungsfunktionen aus der Funktionenfamilie der Termauswertung TA_{SIG} darstellt, ist dieses Schema ein rekursives Definitionsschema.

7) Zunächst werden die Teilterme $te_1,..,te_{Kj}$ des zusammengesetzten Terms te ausgewertet, indem für jeden Teilterm te_k mit $k \in \{1,...,K_j\}$ der Wert $te_k' = eval_{h(k)}(te_k)$ der jeweils zugehörigen Auswertungsfunktion $eval_{h(k)}$ ermittelt wird. Dabei können die Teilterme te_k aus verschiedenen Sorten $sort_{h(1)} \in SO$ stammen, so daß zu ihrer Auswertung entsprechend unterschiedliche Auswertungsfunktionen $eval_{h(k)}$ herangezogen werden müssen. Wenn es sich beim Teilterm te_k um einen Basisterm handelt, kann er als Konstantensymbol oder Variable unmittelbar durch die Funktionen $eval_{h(k)} = eval_{h(k).KO}$ bzw. $eval_{h(k)} = eval_{h(k).VA}$ ausgewertet werden. Falls der Teilterm te_k jedoch seinerseits zusammengesetzt ist, wird das Auswertungsverfahren so lange rekursiv fortgesetzt, bis jeweils der letzte der mittelbar gebildeten Teilterme ein Basisterm ist. Jeder unmittelbar oder mittelbar erhaltene Basisterm wird durch ein sortengleiches formales Objekt ersetzt. Auf diese Weise werden alle Teilterme te_k aus dem Argument $(te_1,..,te_{Kj})$ des Operationssymbols Op_j durch sortengerecht zugeordnete formale Objekte $ob_k \in OB_{h(k)}$ mit $eval_{h(k)}(te_k) = ob_k$ ersetzt. Dann liegt das Argument $arg = (eval_{h(1)}(te_1),...,eval_{h(Kj)}(te_{Kj}))$ in der evaluierten Form $arg' = (ob_1,...,ob_{Kj})$ vor. Auf dieses Argument wird der Operator $op_j(...)$ derjenigen Operation op_j: $OB_{h(1)}x..xOB_{h(Kj)} \rightarrow OB_{h(Kj+1)}$ angewendet, die in der SIG-Algebra dem Operationssymbol Op_j aus der Signatur SIG zugeordnet ist. Schließlich liefert die Operatorapplikation $op_j(arg') = ob$ dasjenige formale Objekt ob, das dem zusammengesetzten Term te mit $te \in TE_i$ durch die Auswertungsfunktion $eval_i$ als Funktionswert $eval_i(te) = ob$ mit $ob \in OB_i$ zugeordnet ist.

8) Unter der Sortenstruktur eines Terms oder eines formalen Objekts wird das Ergebnis folgender Konstruktion verstanden: Jeder involvierte Term te bzw. jedes beteiligte formale Objekt ob wird durch die Bezeichnung $sort_i$ derjenigen Sorte ersetzt, die dem Term mit $te \in TM_i(VA_i)$ bzw. dem formalen Objekt mit $ob \in OB_i$ zugeordnet ist. Basisterme und atomare formale Objekte werden unmittelbar durch jeweils genau eine Sortenbezeichnung ersetzt. Sie besitzen jeweils eine simple Sortenstruktur. Zusammengesetzte Terme und zusammengesetzte formale Objekte werden durch Kombinationen aus Sortenbezeichnungen und Operationssymbolen ersetzt. Solche Ersetzungen stellen komplexe Sortenstrukturen dar. Sie sind entsprechend der Term- bzw. Objektzusammensetzung durch die Operationssymbole ineinander verschachtelt.

9) Die Sortentreue folgt aus der sortengerechten Anwendung von Auswertungsfunktionen und aus dem Rekursionsschema der Definition jener Auswertungsfunktionen. Jeder Basisterm te mit $te \in TM_i(VA_i)$ aus einer Sorte $sort_i$ wird durch die Subfunktionen $eval_{i.KO}$ oder $eval_{i.VA}$ unmittelbar auf ein formales Objekt ob aus derselben Sorte $sort_i$ abgebildet. Also wird jedem Konstantensymbol Ko mit $Ko \in KO_i$ und jeder Variable X mit $X \in VA_i$ jeweils ein sortengleiches formales Objekt ob mit $ob \in OB_i$ zugeordnet. Das formale Objekt kann ebenso als eine sortengleiche Konstante oder sortengleiche 0-stellige Operation bezeichnet werden. Durch das Rekursionsschema der Auswertungsfunktionen wird die Auswertung von zusammengesetzten Termen schrittweise und strukturerhaltend auf die Auswertung von Basistermen zurückgeführt. Daher wird auch jeder zusammengesetzte Term mit einer komplexen Sortenstruktur auf ein zusammengesetztes formales Objekt mit derselben komplexen Sortenstruktur abgebildet.

10) Die hier behandelte Sortentreue der Anwendung von Auswertungsfunktionen ist daher deutlich zu unterscheiden von der o.a. Sortengerechtigkeit der Evaluation von Termen. Eine Termevaluation erfolgt schon dann sortengerecht, wenn auf einen Term te der Sorte $sort_i$ eine Auswertungsfunktion $eval_i$ für dieselbe Sorte angewendet wird. Dies garantiert nur, daß das zugeordnete formale Objekt $ob = eval_i(te)$ aus derselben Sorte $sort_i$ stammt. Falls der Term te zusammengesetzt ist, sagt die sortengerechte Funktionsanwendung jedoch noch nichts über das Verhältnis der internen Sortenstrukturen von Term te und zugeordnetem formalen Objekt ob aus. Auf diesen weitergehenden, die Sortenstruktur betreffenden Sachverhalt erstreckt sich erst der Begriff der Sortentreue. Nur für Basisterme fallen Sortentreue und Sortengerechtigkeit zusammen.

11) Im Kontext des Signaturkonzepts wird die Kongruenzrelation nicht allgemein auf Terme, sondern nur auf Grundterme bezogen; vgl. EHRIG (1985a), S. 39; RECK (1988), S. 50. Eine explizite Begründung der Einschränkung auf Grundterme erfolgt dort jedoch nicht. Sie ist zwar erforderlich, weil die zugrundegelegte Auswertungsfunktion enger definiert ist als die vom Verf. benutzte. Jene Auswertungsfunktion ist nicht für Terme definiert, die Variablen enthalten, sondern nur für variablenfreie Grundterme. Statt dessen wird auf Variablen eine spezielle Zuweisungsfunktion angewendet. Es bleibt jedoch offen, warum diese Zuweisungsfunktion nicht auch zur Definition der Termkongruenz herangezogen wird. Darüber hinaus findet sich bei EHRIG (1985a), S. 21, und RECK (1988), S. 46, ein Ansatz, die Zuweisungsfunktion so zu erweitern, daß sie im Prinzip der umfassend definierten Auswertungsfunktion des Verf. gleicht. Der Verf. vermag nicht nachzuvollziehen, warum RECK diese erweiterte Zuweisungsfunktion nicht benutzt, um eine Termkongruenz zu definieren, die sich auch auf variablenenthaltende Terme erstreckt.

12) Vgl. EHRIG (1985a), S. 39; RECK (1988), S. 50 u. 79f.; REISIG (1989a), S. 6.

13) In der Programmiersprache PROLOG wird zwischen formalen Objekten einerseits und Termen andererseits nicht differenziert. Statt dessen werden dort alle Terme als gleich behandelt, wenn sie sich im algebraischen Sinne kongruent verhalten. Daher wird im Kontext der Programmiersprache PROLOG - wie auch in den meisten anderen formalsprachlichen, aber nicht algebraisch fundierten Konzepten - die algebraische Kongruenzrelation "≈" durch die Gleichheitsrelation "=" ersetzt. Dann drückt die Gleichung $te_1 = te_2$ aber nicht mehr - wie oben bei der algebraischen Kongruenzdefinition - notwendig die Identität, sondern nur noch die Kongruenz der involvierten Terme aus. Dieser kongruenzbezogene Gebrauch der Gleichheitsrelation wird später auch vom Verf. zugrundegelegt, wenn die Gleichheitsrelation auf Terme angewendet wird. Dies ist zwar aus algebraischer Perspektive unglücklich, weil die Differenzierung zwischen Termkongruenz und -identität verloren geht. Doch wird hierdurch ein unkomplizierter Anschluß an die vorherrschende Verwendung der Gleichheitsrelation in formalsprachlichen Konzepten ermöglicht.

14) Von einer kanonischen Erweiterung wird hier immer dann gesprochen, wenn ein Konzept ursprünglich für artgleiche formale Objekte eingeführt wurde und nachträglich auf K_j-stellige Tupel $tup = (tu_1, \ldots, tu_{Kj})$ mit $K_j \in \mathcal{N}_+$ so erweitert wird, daß jede Stelle tu_k des Tupels mit $k \in \{1, \ldots, K_j\}$ durch Objekte der ursprünglich eingeführten Objektart eingenommen werden kann.
Im Regelfall wird die kanonische Erweiterung f_j^+ einer ursprünglich einstelligen Funktion "f" durch folgendes Schema geleistet:

$$f: \quad ob \rightarrow f(ob)$$

$$f_j^+: \quad (ob_1, \ldots, ob_{Kj}) \rightarrow f_j^+(ob_1, \ldots, ob_{Kj}) = (f_j(ob_1), \ldots, f_j(ob_{Kj}))$$

Im Grenzfall einstelliger Tupel fallen die Funktionswerte f(ob) einer objektbezogenen Funktion "f" und die Funktionswerte $f_j^+((obj))$ ihrer tupelbezogenen kanonischen Erweiterung f_j^+ wegen $f_j^+((ob_1)) = (f_j(ob_1))$ im wesentlichen zusammen. Denn es gilt $f_j^+((ob_1)) = f(ob)$, sofern in $(f_j(ob_1))$ von der äußeren Klammersetzung und der Indizierung abstrahiert wird. Klammern und Indizes besitzen *hier* nur unwesentliche formale Bedeutung.
Falls eine einstellige Funktion "f" kanonisch zur K_j-stelligen Funktion f_j^+ erweitert wird, resultiert daraus strenggenommen eine Doppelklammerung der Funktionswerte. Sie beruht auf der inneren Einklammerung der Funktionsargumente als Tupel $(tu_1, \ldots, tu_{Kj})$ und auf der äußeren Einklammerung durch den Funktor $f_j(\ldots)$. Die formal präzisere Notation $f_j((tu_1, .., tu_{Kj}))$ wird durch Reduzierung der Doppel- auf Einfachklammerung ohne Bedeutungsverlust zu $f_j(tu_1, .., tu_{Kj})$ vereinfacht. Diese Kurznotation für die Anwendung von Funktionen (Operationen) auf Tupel wird in

dieser Arbeit - ohne besondere Hinweise - stets als vereinbart vorausgesetzt. Zwischen der ursprünglichen Funktion "f" für Objekte und der kanonisch erweiterten Funktion f_j^+ für Objekttupel braucht auch nicht durch die unterschiedliche Notation "f" bzw. f_j^+ differenziert zu werden. Wenn für beide Fälle dieselbe Notation benutzt wird, geht aus dem jeweils aktuellen Anwendungskontext der Funktion und aus der Notation ihrer Argumente hervor, ob es sich um die ursprünglich eingeführte oder um die kanonisch erweiterte Variante handelt.

15) Strenggenommen handelt es sich um Markenkopien.

16) Dennoch wurde das konventionelle Auswahl- und Ermittlungsproblem den voranstehenden Erläuterungen zur Definition der SIG-Termauswertung zugrundegelegt, um ihre Bedeutung im Rahmen des Signaturkonzepts transparent darstellen zu können. Eine direkte Anknüpfung an das Auswertungskonzept konstruktiver Existenzbeweise hätte nach Einschätzung des Verf. dagegen das Verständnis erschwert. Denn dieses Konzept erweist sich erstens komplizierter als die o.a. konventionelle Ermittlungsaufgabe. Zweitens ist es im literarischen Kontext von Signaturen unüblich.

17) Dies ist immer dann der Fall, wenn mindestens eine der involvierten Objektmengen OB_i unendlich ist, wie z.B. für OB_i = "integer", aber die SIG-Termfamilie selbst nur endlich viele Basisterme umfaßt. Dann können unendlich viele verschiedene Auswertungsfunktionen $eval_{i,q}$ definiert werden. Sie unterscheiden sich zumindest dadurch, daß sie mindestens ein Konstantensymbol Ko mit $Ko \in KO_i$ oder mindestens eine Variable X mit $X \in VA_i$ auf jeweils ein anderes formales Objekt $ob \in OB_i$ als Konstante abbilden.

18) Dies gilt strenggenommen nur insofern, als die denkmöglichen Auswertungsfunktionen mit denjenigen Auswertungsfunktionen identifiziert werden, die das oben eingeführte rekursive Definitionsschema einhalten. Rekursive Schemata stellen jedoch die derzeit leistungsfähigsten Definitionskonzepte für Funktionen überhaupt dar. Dies gilt zumindest so lange, wie die definierten Funktionen in einem intuitiven Sinne noch "berechenbar" sein sollen. Diesen Sachverhalt drückt die CHURCH-These aus, der zufolge jede Funktion rekursiv definiert sein muß, die sich in irgendeiner, aber endlichen Weise "berechnen" läßt. Vgl. dazu ZELEWSKI (1989c), S. 30ff., und die dort angeführte vertiefende Literatur; vgl. ebenso die Anmerkungen zur inhaltlich erweiterten CHURCH/POST/TURING-These. Vgl. schließlich die entsprechende Anmerkung zur Leistungsfähigkeit rekursiver Definitionsschemata. Der Einfachheit halber werden hier denkmögliche mit rekursiv definierten Auswertungsfunktionen gleichgesetzt.
Darüber hinaus entfällt bei der späteren Bezugnahme auf das Unifizierungskonzept die Rückbindung an rekursive Auswertungsfunktionen ohnehin. Die dort erfolgenden Unifizierungsoperationen beruhen auf dem formalen Konzept der Termsubstitution. Es ist so allgemein formuliert, daß es für *alle* endlichen Terme gilt, ohne an irgendeiner Stelle auf rekursive Funktionen zurückgreifen zu müssen. Daher ist das hier behandelte Auswertungskonzept konstruktiver Existenzbeweise leistungsfähig genug, um noch nicht einmal durch die Betrachtung rekursiv definierter Auswertungsfunktionen ernsthaften Einschränkungen unterworfen zu sein. Das rekursive Definitionsschema der Auswertungsfunktionen wurde lediglich aus didaktischen Gründen gewählt. Denn es ist - verglichen mit dem Unifizierungskonzept - noch relativ einfach verständlich. Auch läßt es sich im Rahmen des algebraischen Signaturkonzepts unmittelbar definieren. Das Unifizierungskonzept ist dagegen in algebraischen Kontexten unüblich.

19) Dieser Fall einer negativen Lösung des Existenzproblems spielt im Rahmen des bisher eingeführten Signaturkonzepts kaum eine Rolle. Denn die Objektmengen OB_i enthalten im allgemeinen jeweils mindestens ein formales Objekt, das einem präsentierten Term durch irgendeine denkmögliche Auswertungsfunktion zugeordnet werden kann. Negative Lösungen des Existenzproblems erlangen aber im Anschluß an die spätere Einführung von SIG-Spezifikationen eine erhebliche Bedeutung. Für die dort gebildeten Formeln $for(te_1,...,te_K)$ über Termen muß untersucht werden, ob sie durch formale Objekte $ob_1,...,ob_K$ erfüllt werden können. Auch diese Untersuchung läßt sich als ein Existenzproblem formulieren. Es ist zu entscheiden, ob mindestens eine Kombination aus Auswertungsfunktionen und aus einem Objekttupel $(ob_1,...,ob_K)$ existiert, für das gilt: Die Anwendung der Auswertungsfunktionen auf die Terme aus dem Argument $(te_1,...,te_K)$ der Formel "for" liefern das Objekttupel $(ob_1,...,ob_K)$ als entsprechende Kombination der Funktionswerte. Dieser Sachverhalt wird bei der späteren Formulierung von Schaltregeln für Transitionen in Synthetischen Netzen eine erhebliche Rolle spielen. Dort muß untersucht werden, ob eine jeweils präsentierte Formel $for(te_1,...,te_K)$ als Beschriftung einer Transition durch mindestens ein Objekttupel $(ob_1,...,ob_K)$, das eine Marke darstellt, erfüllt werden kann.
Darüber hinaus könnte die Formel $for(te_1,...,te_K)$ inkonsistent formuliert sein. Dann läßt sie sich grundsätzlich nicht durch irgendein Objekttupel $(ob_1,...,ob_K)$ erfüllen. Eine solche Inkonsistenz ist für die bislang eingeführte Termauswertung zwar nicht definiert. Denn Terme te, auf welche die Auswertungsfunktionen angewendet werden sollen, können nicht in sich widersprüchlich sein. Doch wird diese Inkonsistenzoption später durch die Einführung aussagenlogischer Formelverknüpfungen eröffnet. Dann ist es durchaus möglich, daß eine Transition grundsätzlich niemals schalten kann, weil bereits ihre Formelbeschriftung inkonsistent ist. Als Regelfall werden aber in dieser Arbeit stets konsistente Formeln für die Transitionsbeschriftungen vorausgesetzt. Daher wird diese mögliche Ursache negativer Lösungen von Existenzproblemen nicht weiter berücksichtigt.

20) Nicht-konstruktive Existenzbeweise beruhen auf dem Refutationskonzept indirekter Schlüsse. Ausgangspunkt ist die zu beweisende Behauptung, ein Objekt mit bestimmten Eigenschaften existiere. Ein indirekter Beweis beginnt dann mit der kontradiktorischen Annahme, es gebe kein Objekt mit den geforderten Eigenschaften. Es wird zu zeigen versucht, daß diese Annahme in sich widersprüchlich ist. Falls dieser Versuch gelingt, wird auf das Prinzip des ausgeschlossenen Dritten (tertium non datur) zurückgegriffen: Aus dem Wissen über die Widersprüchlichkeit der kontradiktorischen Annahme wird gefolgert, daß die ursprünglich zu beweisende Behauptung doch gültig sein muß. Folglich existiert mindestens ein Objekt mit den geforderten Eigenschaften; q.e.d. Aus der Inkonsistenz der kontradiktorischen Annahme folgt aber noch kein positives Wissen über die konkrete Gestalt des mindestens einen existenten Objekts. Denn das logisch erschlossene Existenzwissen alleine beinhaltet noch kein Wissen darüber, um welche Objekte es sich dabei tatsächlich handelt.
Vgl. zur indirekten Beweistechnik Meschkowski (1967), S. 149 u. 159ff.; Smullyan (1968), S. 110; Loveland (1978), S. 30f.; Stegmüller (1984b), S. 100f.; Zelewski (1986a), S. 944ff., insbesondere S. 951f.
Auch das unten angesprochene Resolutionskonzept beruht auf dem indirekten Beweisansatz der Refutation; vgl. Yates (1970), S. 259; Loveland (1978), S. 30f.; Nilsson,N. (1980a), S. 161; Clocksin (1981), S. 220f.; Apt (1982), S. 846 u. 849ff.; Cohen,P. (1982), S. 90; Graham,N. (1983), S. 176; Stede (1984), S. 83; Zelewski (1986a), S. 944ff., insbesondere S. 950ff.; Beetz (1986), S. 27.
Sogar das alternative Inferenzkonzept der algebraischen Vervollständigung, das an anderer Stelle skizziert wird, benutzt die indirekte Beweistechnik; vgl. z.B. Hsiang (1983a), S. 337 u. 339.
Die Problematik der Nonkonstruktivität mancher formaler Beweistechniken wird eingehender im Rahmen der konstruktivistischen oder intuitionistischen Ausrichtung von Logik und Mathematik thematisiert. Dort wird die voranstehend skizzierte indirekte Beweistechnik als unzulässig kritisiert. Darüber hinaus werden zumeist noch das Prinzip des ausgeschlossenen Dritten bezweifelt, das der indirekten Beweistechnik zugrundeliegt, und Aussagen - insbesondere Existenzaussagen - über unendlichen Argumentemengen kritisch beleuchtet. Vgl. zu Überblicken über diese konstruktivistisch-intuitionistische Konzeption Brouwer (1913), S. 81ff.; Brouwer (1919), S. 203ff.; Brouwer (1925), S. 1ff.; Brouwer (1928), S. 48ff.; Fraenkel (1930), S. 289ff.; Heyting (1931), S. 106ff.; Church (1932), S. 348ff.; Gentzen (1936), S. 497 u. 522ff.; Gentzen (1938), S. 12ff.; Lorenzen,P. (1962), S. 9ff.; Mendelson (1964), S. 4f.; Meschkowski (1967), S. 220ff.; Heyting (1968a), S. 312ff.; Heyting (1968b), S. 316ff.; Körner,S. (1968), S. 142ff. u. 162ff.; Myhill (1968), S. 324ff.; Carnap (1968), S. 41ff. u. 114f.; Stegmüller (1976b), S. 438ff.; Rautenberg (1979), S. 249ff.; Gethmann (1980b), S. 17ff.; Putnam,H. (1982b), S. 25ff.; Mittelstaedt (1983), S. 30f. u. 34f.; Machover (1983), S. 5, 7f. u. 10f.; Popper (1984b), S. 132ff. u. 319f.; Bunge (1985a), S. 58ff.; Richter,M.M. (1988), S. 22f.; Wedekind (1989c), S. 23ff. Diese Konzeption wird fortan auch als logischer Konstruktivismus angesprochen. Zwischen ihm und dem früher skizzierten epistemologischen Konstruktivismus bestehen keine inhaltlichen Überschneidungen. Dagegen ist der linguistische Konstruktivismus durch die Arbeiten der "Erlanger" und der eng verwandten "Konstanzer Schule" aus dem gedanklichen Fundament des logischen Konstruktivismus hervorgegangen. Der linguistische Konstruktivismus bestärkt die oben skizzierte Kritik an allen nicht-konstruktiven Argumentationsführungen.
Vgl. zur Kritik an indirekten Existenzbeweisen und ihrem nicht-konstruktiven Charakter Fraenkel (1930), S. 290; Gentzen (1938), S. 18; Körner,S. (1968), S. 149ff.; Stegmüller (1976b), S. 439f.; Rautenberg (1979), S. 249 u. 251f.; Bachem (1980), S. 816; Nelson,R.J. (1982), S. 547; Davis,M. (1983), S. 7; Bunge (1985a), S. 58f.; Zelewski (1986a), S. 194; vgl. auch zur entsprechenden Bevorzugung von konstruktiven Beweisführungen Machover (1983), S. 7.
Vgl. des weiteren zur analogen Unzulänglichkeit von nicht-konstruktiven Fragebeantwortungen Raphael (1976), S. 130ff.; Dilger (1979), S. 277f.; Janas (1979), S. 431f. Solche Fragebeantwortungen gehören zu dem Typ: "Kennen Sie den Preis dieser Ware?" - "Ja!". Die bloß assertorische Preiskenntnis bleibt für den Fragesteller aber solange wertlos, wie er nicht konkret den Preis erfährt.
Vgl. schließlich zum Konzept des linguistischen Konstruktivismus Kamlah,W. (1967), S. 23ff.; Mittelstraß (1973), S. 57ff.; Lorenzen,P. (1973a), S. 74ff.; Lenk (1973), S. 98ff.; Janich (1974), S. 20f., 25ff., 39ff., insbesondere S. 42ff., 55ff. u. 70ff.; Mittelstraß (1974), S. 183ff.; Kambartel (1974b), S. 9ff.; Lorenzen,P. (1975a), insbesondere S. 9ff., 18f.; Kambartel (1975), S. 179 u. 182; Gethmann (1977), S. 356ff.; Lorenzen,P. (1978a), S. 109ff.; Alexy (1978a), S. 178ff.; Wohlrapp (1979), S. 359ff., insbesondere S. 362ff.; Gerum (1979), S. 206ff.; Lorenzen,P. (1980), S. 43ff., insbesondere S. 48; Seiffert (1983a), S. 27ff. u. 144ff.; Mittelstraß (1984a), S. 125ff., insbesondere S. 130ff.; Mittelstraß (1984b), S. 12ff. Vgl. auch weitere Beiträge zu Detailaspekten in den Sammelwerken Kambartel (1974a); Lorenz,Ku. (1979a); Gethmann (1980a) und Schwemmer (1981a).

21) Es ist möglich, daß demselben präsentierten Term te wegen $ob_1 = eval_{i.1}(te)$, $ob_2 = eval_{i.2}(te)$ und $ob_1 = ob_2$ durch mindestens zwei verschiedene Auswertungsfunktionen dennoch dasselbe formale Objekt zugeordnet wird. Denn die Unterschiedlichkeit der angewandten Funktionen impliziert keineswegs, daß ihre Bilder für alle Argumente identisch sein müßten.

22) Dies gilt für die Lösung des Existenzproblems notwendig, solange ein nicht-konstruktiver Beweis erfolgt. Der Existenzbeweis kann aber auch konstruktiv geschehen. Davon unabhängig könnte das Konstruktionsproblem - abweichend von der o.a. Formulierung - auch so spezifiziert werden, daß es auch die Angabe mindestens einer (bzw.

aller) beweiserfüllenden Auswertungsfunktion(en) verlangt. Für das hier entwickelte Termauswertungskonzept besitzt dieses Wissen über konkret bestimmte Auswertungsfunktionen aber keine sachliche Relevanz. Wenn das Konzept später auf die Formulierung der Schaltregeln von Transitionen in Synthetischen Netzen angewendet wird, interessieren nur die zugeordneten formalen Objekte - d.h. die Marken -, nicht aber die zuordnungsermöglichenden Auswertungsfunktionen. Auswertungsfunktionen der hier thematisierten Art sind im Petrinetz-Konzept im allgemeinen noch nicht einmal definiert. Eine der seltenen Ausnahmen hiervon stellen die Ausführungen von RECK (1988), S. 82f. ("Belegung ass"), und REISIG (1989a), S. 12f. ("occurence mode ... ß"), dar. Sie binden Auswertungsfunktionen explizit in die Schaltregel von Netzen ein, die zur weit definierten Klasse der Prädikat/Transition-Netze zählen. Aber auch dort interessiert nur die Existenz der betreffenden Auswertungsfunktionen. Die konkreten Funktionsdeklarationen werden auch von RECK und REISIG nicht expliziert.
Die Irrelevanz der Auswertungsfunktionen zeigte sich bereits, als in einer früheren Anmerkung Transitionsbeschriftungen thematisiert wurden. Für die dort angesprochene Erfüllung der Formel aus einer Transitionsbeschriftung spielt nur eine Rolle, welche Marken als formale Objekte die Formel erfüllen (sofern mindestens eine solche Markenkombination überhaupt existiert). Es bleibt aber aus der Perspektive des Petrinetz-Konzepts vollkommen unerheblich, welche Auswertungsfunktionen diese Formelerfüllung ermöglichen könnten.

23) Vgl. zum Unifizierungskonzept SMULLYAN (1963), S. 828ff.; ROBINSON,J. (1965), S. 32ff.; YATES (1970), S. 260ff.; CHANG,C.L. (1973), S. 74ff.; ITZINGER (1976), S. 47ff.; LOVELAND (1978), S. 73ff., insbesondere S. 77; RICHTER,M.M. (1978), S. 171ff.; BIBEL (1982a), S. 88ff.; SIEKMANN (1982a), S. 102ff., insbesondere S. 113ff.; STEDE (1984), S. 85ff.; BAST (1985), S. 14ff.; BÜRCKERT (1986), S. 186f.; SIEKMANN (1987), S. 365ff.; DELAHAYE (1987), S. 113ff.; BÜRCKERT (1987a), S. 104ff.; COHN (1987), S. 189f.; CORDES (1988), S. 19ff., 39ff. u. 46ff.; RUZICKA (1988), S. 501ff. (mit einer besonders effizienten algorithmischen Darstellung auf S. 505); BEZEM (1988), S. 17f.; PREIß (1989), S. 19f. Auf das Unifizierungskonzept wird im Zusammenhang mit der operationalen Dimension der Prädikatenlogik zurückgekommen.

24) Mitunter wird die Unifizierung nicht als Variablen-, sondern als Termersetzung thematisiert; vgl. z.B. CORDES (1988), S. 39ff. Dabei werden jedoch - in der prädikatenlogisch üblichen Weise - teilevaluierte Terme unterstellt. Da hierin bereits alle Konstanten- und Funktionssymbole durch Konstanten bzw. Funktionen ersetzt sind, bleibt die Termersetzung auf eine Variablensubstitution beschränkt.

25) Auf diesem "Vereinheitlichen" der Formelargumente beruht die Konzeptbezeichnung "Unifizierung". Grob gesprochen bezieht sich jede Unifizierung auf zwei teilevaluierte Terme. Zwei solche Terme lassen sich genau dann unifizieren, wenn es möglich ist, eine Substitution anzugeben, die folgende Anforderungen erfüllt: Jeder Variablen wird genau ein Substitutionsterm so zugeordnet, daß die beiden ursprünglichen Terme identisch werden, nachdem alle ihre Variablenvorkommnisse durch die variablenspezifischen Substitutionsterme ersetzt worden sind. Die Substitutionsterme können Konstanten oder Variablen sein oder aus diesen und aus Funktoren zusammengesetzt sein. Vgl. dazu die übersichtlichen Darstellungen bei SIEKMANN (1987), S. 366; CORDES (1988), S. 39ff.
Darüber hinaus kann die Variablensubstitution sortengerecht erfolgen. Dies ist zwar bei konventionellen Darstellungen des Unifizierungskonzepts nicht vorgesehen, da sie jeweils auf einem konventionellen prädikatenlogischen Kalkül ohne überlagerte Sortenstruktur beruhen. Doch realisiert die hier verwendete Programmiersprache Turbo-PROLOG - im Gegensatz zu vielen anderen PROLOG-Dialekten - eine sortengerechte Variablensubstitution. Dies rechtfertigt nochmals die Auswahlentscheidung zugunsten dieser Sprache. Vgl. auch die explizite Darstellung sortengerechter Ersetzungsoperationen im Rahmen sortierter prädikatenlogischer Kalküle bei OBERSCHELP (1962), S. 297 u. 308f. (sortengerechte Termersetzungen); COHN (1987), S. 189f. (sortengerechte Unifizierungen); BOLLINGER (1989), S. 208ff. (sortengerechte Unifizierungen).
In seinem Kern beruht das Unifizierungskonzept auf dem Unifizierungstheorem von ROBINSON und PRAWITZ. Es weist einen Algorithmus nach, der zwei wesentliche Auswertungsleistungen realisiert: Erstens vermag er für je zwei beliebige Terme zu entscheiden, ob sie überhaupt im oben skizzierten Sinne unifiziert werden können. Wenn dies der Fall ist, dann gibt er in *konstruktiver* Weise einen Unifikator beider Terme an. Dabei liefert der Unifizierungsalgorithmus nicht nur einen beliebigen, sondern sogar einen größtmöglichen Unifikator. Vgl. zu diesem Unifizierungstheorem die bereits angeführten Quellen zum Unifizierungskonzept, insbesondere RICHTER,M.M. (1978), S. 172f.
Aus einem anderen Blickwinkel läßt sich die Unifizierung im Rahmen der logischen Programmierung auf PROLOG-Basis als konsistente Zuordnung ("Matching") von Konstanten zu Variablen charakterisieren. Dabei wird versucht, zulässige Belegungen für die Variablen einer prädikatenlogischen Formel zu finden. Bei diesen Zuordnungsversuchen werden die Variablen durch Konstanten aus anderen "passenden" Formeln - Fakten desselben Formeltyps (Prädikats) oder Formeln aus der jeweils anderen Hälfte (Subziel) einer Hornklausel - ersetzt ("gebunden"). Daher wird die oben eingeführte Charakterisierung von Termauswertungen, daß sie im wesentlichen Belegungen von Variablen mit formalen Objekten (Konstanten) darstellen, durch das Unifizierungskonzept unterstützt. Näheres zu dieser Matchingperspektive findet sich bei PROLOG (o.J.), S. 51ff.; NILSSON,N. (1980a), S. 378ff.; CLOCKSIN (1981), S. 219 u. 224; KONAGAYA (1984), S. 48ff.; BEETZ (1986), S. 84; TOWNSEND,C. (1987), S. 53; ESTENFELD (1987), S. 68; SIEKMANN (1987), S. 389f.; MURATA,TA. (1988b), S. 482; vgl. aber auch zu einer ver-

feinernden Unterscheidung zwischen Unifizierung einerseits und konventionellen Matching-Konzepten der Künstlichen Intelligenz andererseits LEVI,G. (1986), S. 408f.

26) Vgl. dazu die Beschreibung der Unifizierungsoperationen.

27) Dies rechtfertigt nachträglich, bei der oben erfolgten Formulierung des Konstruktionsproblems keine Kenntnis einer konkret bestimmten Auswertungsfunktion vorauszusetzen.

28) Strenggenommen gilt diese Ergebnisübereinstimmung nur in bezug auf Komplementauswertungsfunktionen, da die prädikatenlogische Unfizierungsoperationen bereits die Applikation von Teilauswertungsfunktionen voraussetzen.

29) Vgl. dazu die Darstellungen des Unifizierungskonzepts in den bereits angeführten Quellen. Eine besonders umfassende Übersicht über das Unifizierungskonzept findet sich bei SIEKMANN (1987), S. 365ff., insbesondere S. 375ff.

30) Auf das Resolutionskonzept wird im Zusammenhang mit der operationalen Dimension der Prädikatenlogik näher zurückgekommen.

31) Eine abweichende Terminologie vertritt LEVI,G. (1986), S. 400f.: Dort wird ein weit gefaßter Resolutionsbegriff zugrundegelegt, der das Unifizierungskonzept und ein Ersetzungskonzept (goal rewriting process) als Teilkonzepte umfaßt. Der Verf. stuft dagegen Unifizierungs- und Resolutionskonzept zunächst als unabhängige und gleichrangige Konzepte ein, die zur Konstitution eines Inferenzkonzepts miteinander kombiniert werden. Dabei wird ein engeres Verständnis des Resolutionskonzepts vertreten, das dem vorgenannten Ersetzungskonzept entspricht.

32) Diese Einschränkung berücksichtigt, daß Vollständigkeit und Korrektheit nur unter der Prämisse *unendlicher* Ressourcenverfügbarkeit gelten. Denn bei Versuchen, prädikatenlogische Theoreme auf der Grundlage des Resolutionskonzepts zu beweisen, müssen unter Umständen unendlich große abstrakte Suchräume unifizierend durchforscht werden. Vollständigkeit und Korrektheit der Prädikatenlogik 1. Stufe lassen sich dann nur noch garantieren, wenn auch unendlich ausgedehnte Beweisprozeduren zugelassen werden. Da in realen Beweissystemen jedoch nur *endliche* Ressourcen zur Verfügung stehen, können unvollständige oder unkorrekte "Beweise" resultieren. Darauf wird an späterer Stelle zurückgekommen. Vgl. des weiteren die Anmerkungen zum Backtracking-Scheitern infolge endlicher Ressourcenverfügbarkeit.

33) Auf die Vollständigkeits- und Korrektheitseigenschaft prädikatenlogischer Beweissysteme wird noch näher eingegangen.

34) Dies gilt zumindest im Vergleich zum hier betrachteten algebraischen Signaturkonzept.

35) Vgl. zur Bedeutung des Unifizierungskonzepts für die Programmiersprache PROLOG und seinen Implementierungen in diversen PROLOG-Dialekten CLOCKSIN (1981), S. 219; ESTENFELD (1987), S. 68f.; TOWNSEND,C. (1987), S. 53; CORDES (1988), S. 39ff. u. 45ff.; PROLOG (o.J.), S. 51ff., insbesondere S. 54ff.

36) Vgl. zur engen Beziehung zwischen der logischen Programmierung und der Sprache PROLOG z.B. CLOCKSIN (1981), S. 225ff.; KOWALSKI (1982a), S. 2f.

37) Die Konstruktivität der Programmiersprache PROLOG wird explizit von FUTO (1990), S. 89f., gewürdigt.

38) Der Term darf allerdings nur einen arithmetischen Ausdruck darstellen.

39) Vgl. COLMERAUER (1983), S. 308; DELAHAYE (1987), S. 179.

40) Vgl. zur fehlenden Einbindung von Sorten oder "Typen" in vorherrschenden PROLOG-Varianten IGEL (1989a), S. 177.

41) Zwar wird in der "domains"-Sektion nicht explizit von Sorten gesprochen. Aber es handelt sich inhaltlich um eine Festlegung von Sorten.

42) Dies betrifft vor allem die Implementierung der Schaltregel für Transitionen in Synthetischen Netzen. Hierauf wurde bereits kurz hingewiesen; ausführlicher wird die prädikatenlogische Schaltregelimplementierung an späterer Stelle dargelegt. Darüber hinaus werden später etliche Erweiterungen des algebraischen Kernkonzepts Synthetischer Netze vorgestellt, die nicht mehr im Rahmen des hier entwickelten algebraischen Signaturkonzepts ausgedrückt werden (können). Statt dessen werden sie ohne größere Schwierigkeiten auf PROLOG-Basis im Rahmen der prädikatenlogischen Implementierung von Synthetischen Netzen eingeführt. Vgl. dazu beispielsweise die Konstruktion von Nulltestkanten, von Absorber- und Distributorkanten, von Schaltprioritäten für Transitionen und von Zeitattributen für Marken.

43) Allerdings muß auch auf theoretische Inferenz- und praktische Implementierungsprobleme von der logischen Programmiersprache PROLOG hingewiesen werden. Sie werden an anderer Stelle ausführlicher behandelt.

44) Das Konzept der Teilevaluierung findet im allgemeinen keine Beachtung. Es muß hier aber eingeführt werden, um später eine Synthese aus algebraischem Signaturkonzept und Prädikatenlogik bilden zu können. Eine der überaus seltenen Ausnahmen, in denen die Möglichkeit einer Teilevaluierung anklingt, ist die Feststellung von STACHOWIAK (1973), S. 251: "Bei syntaktischen Logikkalkülen besteht die Interpretation im stärksten Fall in einer systematischen Belegung der ... nicht-logischen Grundzeichen des abstrakten Kalküls, in schwächeren Fällen einer wenigstens teilweisen ... In-Beziehung-Setzung des Kalküls zu einem ... Bereich von Designata." STACHOWIAK's stärkster Fall entspricht der Vollevaluierung von Termen, wie sie im Signaturkonzept üblich ist. Zu seinen schwächeren Fällen gehört dagegen die prädikatenlogische Teilevaluierung von Termen, die im folgenden näher erläutert wird.

45) Falls nicht ausdrücklich anders vereinbart, wird hierfür die identische Abbildung vorausgesetzt.

46) Dabei wird die kompakte Definitionsvariante von Termfamilien zugrundegelegt, die schon an früherer Stelle eingeführt wurde.

47) Die Notation KB_i erinnert einerseits daran, daß den involvierten nullstelligen Operationen nullstellige Operationssymbole zugrundeliegen, die den Charakter von Konstantensymbolen besitzen ("K"). Andererseits werden diese Konstantensymbole durch die Teilevaluierung auf atomare formale Objekte aus einer Objektmenge OB_i abgebildet ("B").

48) Falls ein Grundterm vorliegt, der per definitionem überhaupt keine Variablen enthält, bezieht sich die Ersetzung "aller" Variablen auf "alle" Elemente aus der leeren Variablenmenge.

49) Dieser Grenzfall stimmt mit einer Auswertungsfunktion $eval_i$ nur dann überein, wenn Teilauswertungsfunktion $teval_i$ auf einen variablenfreien Grundterm angewendet wird.

50) Es handelt sich nur um ein Definitionsschema für Teilauswertungsfunktionen, da weder das Ausmaß, in dem Variablen durch Konstanten ersetzt werden, noch die tatsächlich eingesetzten Konstanten konkret bestimmt sind. Jede solche Bestimmung führt zu einer konkreten Teilauswertungsfunktion $teval_{sort}$.

51) Die minimalen Teilauswertungsfunktionen entsprechen weitgehend den Auswertungsfunktionen, die von Autoren im Kontext des Signaturkonzepts gewöhnlich verwendet werden. Allerdings sind jene Auswertungsfunktionen nur für Grundterme definiert, während die hier verwendeten minimalen Teilauswertungsfunktionen auf alle Terme angewendet werden können. Vgl. zu solchen Auswertungsfunktionen EHRIG (1985a), S. 21; RECK (1988), S. 46; REISIG (1989a), S. 5f.

52) Die nachfolgenden Ausführungen gelten mutatis mutanda nicht nur für sortenspezifische Teilauswertungsfunktionen, sondern auch für deren kanonischen Erweiterungen. Der Einfachheit halber werden die kanonischen Erweiterungen aber nicht explizit erwähnt.

53) Im Fall einer sortenspezifischen Teilauswertungsfunktion handelt es sich bei einem solchen Ausdruck um eine Konstante oder um eine Variable. Bei ihrer kanonischen Erweiterung auf eine sortenunspezifische Teilauswertungsfunktion liegt der Ausdruck dagegen als Termtupel aus Konstanten, Variablen und Operatoren vor. Konstanten, Variablen und Tupel werden der Einfachheit halber hier gemeinsam als (formale) Ausdrücke angesprochen.

54) Vgl. zu Zuweisungsfunktionen RECK (1988), S. 46; REISIG (1989a), S. 6.

55) Dies impliziert: $te' \in OB_i \rightarrow keval_i(te') = id(te') = te'$.

56) Das Adverb "zulässig" kann fortan vernachlässigt werden, wenn durch den - expliziten oder impliziten - Bezug auf die Ausdrucksmenge AM_{SIG} sichergestellt ist, daß nur zulässige Ausdrucksableitungen erfolgt sind.

4.1.6 SIG-Spezifikationen

Auf der Basis einer SIG-Algebra, einer SIG-Termmenge TM_{SIG} und einer SIG-Termauswertung TA_{SIG} lassen sich beliebige algebraische Formeln für die Signatur SIG einführen. Diese Formeln legen Anforderungen fest, die von allen formalen Objekten erfüllt werden müssen, die durch eine SIG-Termauswertung aus einer SIG-Algebra ausgewählt und den Termen aus einer SIG-Termmenge zugeordnet werden. Daher bedeuten die algebraischen Formeln Restriktionen für jede zulässige Termauswertung. Die Gesamtheit aller dieser Restriktionsformeln für eine Signatur stellt eine algebraische Spezifikation[1)] dar.

Definition: algebraische Spezifikation

Eine SIG-Spezifikation für eine Signatur SIG=(SO,OP) ist ein geordnetes 3-Tupel $SPEC_{SIG}$=(SO,OP,RES). RES ist eine nicht-leere Menge algebraischer Formeln mit Restriktionscharakter.

Erläuterungen und Ergänzungen zur algebraischen Spezifikations-Definition:

a) Eine SIG-Spezifikation wird als algebraisch bezeichnet, wenn die zugrundeliegende Signatur SIG durch eine SIG-Algebra interpretiert wird. Je eine signaturzugehörige SIG-Algebra, SIG-Termmenge und SIG-Termauswertung werden für jede SIG-Spezifikation als gegeben vorausgesetzt.

b) Die algebraischen Formeln werden auch als Restriktionsformeln, Restriktionen[2)], Axiome[3)], Gesetze[4)] oder als algebraische Formeln bezeichnet. Die Formelmenge RES wird hier als Restriktionsmenge angesprochen[5)].

c) Die Formelmenge RES wird in induktiver Weise aus algebraischen Basisformeln und aussagenlogischen Formeloperationen gebildet[6)]. Jede algebraische Formel ist eine mindestens zweistellige Formel $for(te_1,...,te_K)$[7)] mit $K \in \mathcal{N}_+$ und $K \geq 2$. Ihr Argument $(te_1,...,te_K)$ besteht aus Termen te_k mit $te_k \in TMF_{SIG}(VAF)$ für alle $k \in \{1,...,K\}$.

d) Eine algebraische Basisformel für die SIG-Spezifikation einer Signatur SIG kann durch jede mehrstellige Relation definiert werden, die für formale Objekte ob aus dem Universum OB_{SIG} der zugehörigen SIG-Algebra gilt. Prinzipiell werden alle Relationen zugelassen[8)], hier aber nicht umfassend eingeführt. Statt dessen werden nur diejenigen algebraischen Basisformeln durch Relationen über formalen Objekten exemplarisch vorgestellt, die bei der späteren Gestaltung Synthetischer Netze am häufigsten verwendet werden. Sie reichen zur Verdeutlichung des allgemeinen Schemas für die Definition algebraischer Formeln aus.

e) Eine K-stellige Relation REL mit $K \in \mathcal{N}_+$ und $K \geq 2$ (mehrstellige Relation) ist eine Teilmenge des K-fachen kartesischen Produkts aus sortenspezifischen Objektmengen $OB_{i(k)}$ mit $k \in \{1,...,K\}$:

$$REL \subseteq (OB_{i(1)} \times ... \times OB_{i(K)})$$

Zusätzlich kann die Sortengleichheit der Objektmengen gefordert werden[9)], muß es aber nicht[10)]. Sortengleiche Objektmengen erleichtern den Auswertungsprozeß der hierauf fußenden algebraischen Basisformeln[11)]. Darüber hinaus erhöhen sie das Formelverständnis[12)].

f) Zur Verdeutlichung des Konzepts algebraischer Basisformeln $for(te_1,..,te_K)$ wird eine Familie zweistelliger algebraischer Vergleichsformeln $for_{\equiv}(te_1,te_2)$ eingeführt[13)]. Es werden jeweils zwei sortengleiche[14)] Terme[15)] te_1 und te_2 mit $te_1,te_2 \in TM_i(VA_i)$ für eine beliebige Sorte $sort_i$ betrachtet. Das Symbol "$\equiv$" vertritt ein beliebiges Element aus der Symbolmenge $\{=,<,\leq,\geq,>,\neq\}$. Jedes dieser Symbole dient der Infixnotation einer zweistelligen Gleichheits-, Ungleichheits- oder Anordnungsrelation $REL_{\equiv}$. Unter diesen Voraussetzungen lassen sich die zweistelligen algebraischen Vergleichsformeln definieren durch:

$$
\begin{array}{ll}
 & (REL_{\equiv} \subseteq (OB_i x OB_i) \wedge eval_i \in TA_{SIG}) \\
\Rightarrow & (for_{\equiv}(te_1,te_2) :\Leftrightarrow eval_i(te_1) \equiv eval_i(te_2))
\end{array}
$$

g) Fortan wird für die Vergleichsformeln $for_{\equiv}(te_1,te_2)$ die vereinfachte Infixnotation "$\equiv$" zugelassen, für die gilt[16)]:

$$te_1 \equiv te_2 :\Leftrightarrow for_{\equiv}(te_1,te_2)$$

Beispielsweise gilt für die Gleichungsrelation $REL_=$ und ihre zugehörige Basisformel $for_=(te_1,te_2)$ unter Ausnutzung der Transitivität von Definitionsbeziehungen:

$$te_1 = te_2 :\Leftrightarrow eval_i(te_1) = eval_i(te_2))$$

Damit wird die früher eingeführte Termkongruenz $te_1 \approx te_2$ vereinfacht als Termgleichheit $te_1 = te_2$ notiert[17)]. Es wird aber ausdrücklich darauf hingewiesen, daß es sich um keine Gleichheit im Sinne der Identität von Termen handeln muß. Vielmehr liegt nur eine Gleichheit im Sinne identischer Ergebnisse von Termauswertungen vor. Die Terme te_1 und te_2 brauchen also selbst keineswegs identisch zu sein[18)]. Die hier definierte Termgleichheit erfordert statt dessen nur, daß die Bilder ihrer zugehörigen Auswertungsfunktion $eval_i$ dieselben formalen Objekte darstellen.

h) Als weitere algebraische Basisformel spielt die 2-stellige "...ist Element von..."-Formel eine größere Rolle. Sie wird mit Hilfe des vertrauten mengentheoretischen Mengenbildungsoperators "$\in$" dargestellt. Diese algebraische "$\in$"-Formel bedarf allerdings einer besonderen formalen Interpretation. Denn sie ist nicht - wie die voranstehend eingeführten algebraischen Basisformeln - über Termtupeln definiert. Statt dessen wird diese Formel im Signaturkonzept benutzt, um auszudrücken, daß ein Term te zur Termmenge $TM_i(VA_i)$ einer ausgezeichneten Sorte $sort_i$ aus der Sortenmenge SO der Spezifikation $SPEC_{SIG} = (SO,OP,RES)$ stammen muß. Anstatt der formal korrekten mengentheoretischen Notation $te \in TM_i(VA_i)$ wird derselbe Sachverhalt in algebraischen Spezifikationen durch die algebraische "$\in$"-Formel in der Infixnotation "$te \in sort_i$" ausgedrückt[19)]. Obwohl die Verwendung der algebraischen "$\in$"-Formel problematisch ist[20)], besitzt sie den Vorzug, die geforderte Sortenzugehörigkeit besonders deutlich hervortreten zu lassen[21)]. Darüber hinaus konnte sie sich in der einschlägigen Literatur etablieren[22)]. Daher wird sie auch in dieser Arbeit verwendet.

i) Eine algebraische Basisformel $for_{REL}(te_1,...,te_K)$ ist bezüglich einer Termevaluation genau dann gültig, wenn gilt: Ihr termbezogenes Argument $arg = (te_1,...,te_K)$ wird durch die Auswertungsfunktionen der Termevaluation auf ein objektbezogenes Argument $arg' = (eval_{i(1)}(te_1),..., eval_{i(K)}(te_K)) = (ob_1,...,ob_K)$ abgebildet, das ein Element aus der zugrundeliegenden Relationen REL ist.

j) Als aussagenlogische Formeloperationen[23] werden die Negation einzelner Formeln sowie die konjunktive, adjunktive, disjunktive, subjunktive und bijunktive Verknüpfung jeweils mehrerer Formeln zugelassen[24]. Es gilt induktiv[25]:

- Induktionsbasis: Jede algebraische Basisformel ist eine Formel "for". Die Gültigkeit dieser Formel bezüglich einer Termevaluation wurde in der voranstehenden Erläuterung i) definiert.
- Negation ("nicht"): Wenn "for" eine Formel ist, dann ist auch ihr Negat "$\neg$for" eine Formel. Das Negat "$\neg$for" ist bezüglich einer Termevaluation genau dann gültig, wenn die Formel "for" bezüglich derselben Termevaluation ungültig ist.
- Konjunktion ("und"): Wenn for_1 und for_2 Formeln sind, dann ist auch das Konjugat $for_i \wedge for_j$ eine Formel. Das Konjugat $for_1 \wedge for_2$ ist bezüglich einer Termevaluation genau dann gültig, wenn beide Formeln for_1 und for_2 bezüglich derselben Termevaluation jeweils gültig sind.
- Adjunktion ("oder"): Wenn for_1 und for_2 Formeln sind, dann ist auch das Adjugat $for_1 \vee for_2$ eine Formel. Das Adjugat $for_1 \vee for_2$ ist bezüglich einer Termevaluation genau dann gültig, wenn mindestens eine der Formeln for_1 und for_2 bezüglich derselben Termevaluation gültig ist.
- Disjunktion ("entweder...oder..."): Wenn for_1 und for_2 Formeln sind, dann ist auch das Disjugat $for_1 \underline{\vee} for_2$ eine Formel. Das Disjugat $for_1 \underline{\vee} for_2$ ist bezüglich einer Termevaluation genau dann gültig, wenn $((\neg for_1 \wedge for_2) \vee (for_1 \wedge \neg for_2))$ bezüglich derselben Termevaluation gültig ist.
- Subjunktion ("wenn...dann...): Wenn for_1 und for_2 Formeln sind, dann ist auch das Subjugat $for_1 \rightarrow for_2$ eine Formel. Das Subjugat $for_1 \rightarrow for_2$ ist bezüglich einer Termevaluation genau dann gültig, wenn $(\neg for_1 \vee for_2)$ bezüglich derselben Termevaluation gültig ist.
- Bijunktion ("genau dann...wenn...): Wenn for_1 und for_2 Formeln sind, dann ist auch das Bijugat $for_1 \leftrightarrow for_2$ eine Formel. Das Bijugat $for_1 \leftrightarrow for_2$ ist bezüglich einer Termevaluation genau dann gültig, wenn $((for_1 \rightarrow for_2) \wedge (for_2 \rightarrow for_1))$ bezüglich derselben Termevaluation gültig ist.

k) Die Restriktionsmenge RES einer Spezifikation ist die Menge aller algebraischen Formeln, die sich in der voranstehend exemplifizierten Weise mit der Hilfe von algebraischen Relationen induktiv einführen lassen. Die zugelassenen Relationen werden auf keine endliche Relationenmenge eingeschränkt. Das kombinatorische Potential der aussagenlogischen Formeloperationen kann beliebig komplexe algebraische Formeln hervorbringen. Aus beiden vorgenannten Gründen ist die Restriktionsmenge nicht abgeschlossen, d.h. potentiell unendlich.

l) Der Restriktionscharakter der algebraischen Formeln beruht auf der Formelrückbindung an Relationen. Zulässige Termevaluationen sind genau solche, unter denen alle algebraischen Formeln aus einer SIG-Spezifikation gültig sind[26]. Termevaluationen, unter denen das Argument mindestens einer algebraischen Formel for_{REL} nicht mehr aus der Objekttupelmenge der zugehörigen Relation REL stammt, sind dagegen unzulässig und werden daher verworfen. In dieser relationsfundierten Ausgrenzung von unzulässigen Termevaluationen mit ihren Auswertungsfunktionen $eval_i \in TA_{SIG}$ liegt der restriktive Charakter der Restriktionsmenge RES einer SIG-Spezifikation begründet.

m) In der algebraisch ausgerichteten Literatur zu SIG-Spezifikationen werden gewöhnlich alle Restriktionen in der Gestalt von Gleichungen notiert. Diese Uniformität wird durch formale Konstrukte ermöglicht, die algebraische mit logischen Ausdrucksmitteln kombinieren[27]. Der Verf. sieht hingegen in der uniformen Gleichungsnotation von Restriktionen keinen Selbstzweck. Statt dessen bevorzugt er für die Formulierung von Restriktionen jeweils diejenige Darstellungsform, die ihm am kompaktesten oder transparentesten erscheint[28].

n) Notiert wird die Restriktionsmenge RES als eine Erweiterung der schon früher definierten schematischen Darstellungsweise von Signaturen. Hinter die "sorts"- und "Ops"-Bereiche gibt eine "Res"-Sektion alle restringierenden algebraischen Formeln einer SIG-Spezifikation an.

Beispiel zur Definition von SIG-Spezifikationen[29]:

Das weiter oben angeführte Beispiel einer Signatur "String" für die algebraische Definition beliebiger Zeichenfolgen läßt sich durch Formeln präzisieren, die restriktive Anforderungen an das Operationssymbol für die Konkatenation von Zeichenfolgen ausdrücken[30]. Es resultiert mit te, te_1, te_2, te_3 als beliebigen Termen aus der SIG-Termfamilie $TMF_{SIG}(VAF)$ für beliebige Zeichenfolgen die "Signatur"-Spezifikation:

String =

sorts:	zeichen	
	zeichenfolge	
Ops:	$Ko_1, ..., Ko_{26}$:	$\rightarrow$ zeichen
	Erzeuge0:	$\rightarrow$ zeichenfolge
	Erzeuge1:	zeichen $\rightarrow$ zeichenfolge
	Verknüpfe:	zeichenfolge zeichenfolge $\rightarrow$ zeichenfolge
Res:	Verknüpfe(te,Erzeuge0()) = te	
	Verknüpfe(Erzeuge0(),te) = te	
	Verknüpfe(Verknüpfe(te_1,te_2),te_3) = Verknüpfe(te_1,Verknüpfe(te_2,te_3))	

Anmerkungen zum Kapitel:

1) Vgl. zu algebraischen Spezifikationen EHRIG (1984b), S. 3ff., insbesondere S. 6ff.; EHRIG (1985a), S. 9ff., 25ff. u. 246ff.; KRÄMER (1987a), S. 274ff.; EHRIG (1987), S. 4ff.; RECK (1988), S. 48ff.; ASTESIANO (1988), S. 142ff.; BEIERLE (1988a), S. 449ff.; EHRICH (1988), S. 123ff.; REISIG (1989a), S. 6, 8ff. u. 34f.

2) Vgl. EHRICH (1988), S. 129f. u. 132 ("constraints").

3) Vgl. EHRIG (1985a), S. 26; ASTESIANO (1988), S. 147ff.; RECK (1988), S. 48; VOß,A. (1989b), S. 374ff.

4) Vgl. RECK (1988), S. 79.

5) Diese Restriktionsformeln werden zumeist als Axiome thematisiert; vgl. dazu die Quellenangaben in den voranstehenden Fußnoten. Diese Bezeichnung hebt den formalen Status der Formeln hervor. In dieser Arbeit interessiert dagegen mehr der Modellierungsbezug des Signaturkonzepts. Aus dieser Perspektive besitzen die Formeln aus einer SIG-Spezifikation den Charakter von Bedingungen, deren Erfüllung bestimmte Strukturen oder Verhaltensweisen eines algebraisch spezifizierten Modells ausschließt. Daher werden diese Formeln hier als Restriktionen bezeichnet. Die gleichen restriktionsorientierten Ansatz vertritt auch EHRICH (1988), S. 129.

6) In die gleiche Richtung weisen die SIG-Spezifikationen für Netze bei RECK (1988), S. 79f., und REISIG (1989a), S. 6.

7) Die hier behandelten algebraischen Formeln werden später als Sonderfälle der umfassender definierten prädikatenlogischen Formeln "prä" erklärt.

8) Dagegen beschränken sich die meisten Werke, die sich mit algebraischen Spezifikationen befassen, weitgehend auf die Gleichheitsrelation und (implizite) aussagenlogische Formelverknüpfungen. Die nachfolgend ebenso zugelassenen Ungleichheits- und Anordnungsrelationen werden dort - ohne Erläuterung ihres Ausschlusses - im allgemeinen nicht berücksichtigt. Die spätere Verallgemeinerung algebraischer Formeln zu prädikatenlogischen Formeln, die bei der Ersetzung von SIG-Spezifikationen durch SIG-Übergangsschemata erfolgt, wird von den vorgenannten rein algebraischen Ansätzen zumeist nicht abgedeckt. Nur EHRIG (1985a), S. 25 u. 89, und EHRICH (1988), S. 123, 129 u. 132, weisen darauf hin, daß auch andere Restriktionstypen als Gleichungen verwendet werden könnten: SIG-Spezifikationen ließen sich grundsätzlich mit beliebigen Restriktionsformeln aus der Prädikatenlogik 1. Stufe vereinbaren. Doch wird dieser begrüßenswerte Ansatz von den o.a. Quellen mehrheitlich nicht konkret ausgeführt. Als einzige Ausnahmen sind die Ausführungen von ASTESIANO (1988), S. 144ff., und von EHRICH (1988), S. 129ff., insbesondere S. 132f., hervorzuheben. Dort findet sich ein freizügiger Umgang mit verschiedensten Formelausprägungen in algebraischen Spezifikationen.
EHRIG (1985a), S. 89, KRÄMER (1987a), S. 275; EHRICH (1988), S. 123, lassen den Grund für die Präferenz von Gleichungen durchblicken. Sie garantieren eine spezielle Klasse algebraischer Spezifikationen (Initialalgebren), die bei beliebigen prädikatenlogischen Formeln nicht mehr erfüllt sein muß. Die Bevorzugung dieser Spezifikationsklasse interessiert aber in der hier vorgelegten Arbeit nicht weiter. Daher erfolgt - wie bei ASTESIANO - auch keine Festlegung auf Gleichungen.

9) Bei sortengleichen Objektmengen, die alle zur selben Sorte $sort_i \in SO$ gehören, vereinfacht sich die Relationsdefinition zu $REL \subseteq (OB_i)^K$.

10) Beispielsweise findet sich diese Einschränkung nicht bei RECK (1988), S. 50.

11) In dieser Arbeit werden grundsätzlich keine Relationen verwendet, die durch sortenverschiedene Grundtermkombinationen erfüllt werden können. Daher wird der Auswertungsprozeß verkürzt, wenn solche sortenverschiedenen Grundtermkombinationen von vornherein durch die Forderung nach Sortengleichheit aus dem Auswertungsprozeß ausgeschlossen werden. Denn sie können ohnehin keine der intendierten relationserfüllenden Grundtermkombinationen darstellen.
Allerdings lassen sich grundsätzlich Relationen vorstellen, die auch durch sortenverschiedene Grundtermkombinationen erfüllt werden können. Dies wäre immer dann der Fall, wenn die sortenspezifischen Objektmengen OB_1 und OB_2 zweier unterschiedlicher Sorten $sort_1$ und $sort_2$ nicht disjunkt sind. Dann können z.B. Termauswertungen $eval_1(te_1)$ und $eval_2(te_2)$ für zwei Terme te_1 und te_2 mit $eval_1(te_1) \in OB_1$ und $eval_2(te_2) \in OB_2$ existieren, für die $eval_1(te_1) \in (OB_1 \cap OB_2)$ und $eval_2(te_2) \in (OB_1 \cap OB_2)$ gilt. In diesem Fall *können* algebraische Basisformeln wie z.B. $te_1 = te_2$ trotz Sortenverschiedenheit der beiden Terme wegen $eval_1(te_1) = eval_2(te_2)$ durchaus erfüllt sein.

12) Durch den expliziten Sortenbezug wird unmittelbar deutlich, welche formalen Objekte durch eine algebraische Formel angesprochen werden.

13) Vgl. zur Bedeutung, die diesen Vergleichsformeln für die Repräsentation von Wissen bei der Koordinierung von Prozessen zukommt, DORN (1989), S. 56ff.

14) Die Einschränkung auf sortengleiche Terme wird hier vorgenommen, weil der Vergleich sortenverschiedener Terme in dieser Arbeit nicht interessiert. Dennoch ist diese Restriktion im allgemeinen Fall nicht unbedingt erfor-

derlich. Denn die Mengen formaler Objekte, die als Nachbereiche von sortenverschiedenen Auswertungsfunktionen definiert sind, dürfen grundsätzlich einander überlappen oder sogar zusammenfallen. Daher ist es möglich, daß die Bilder von sortenverschiedenen Auswertungsfunktionen, die auf sortenverschiedene Terme angewendet werden, dennoch formale Objekte darstellen, die sich durch eine der o.a. Relationen miteinander vergleichen lassen und unter Umständen sogar identisch sein können.

15) In der Literatur werden zumeist nur Grundterme als Komponenten der Argumente von algebraischen Formeln $for(te_1,...,te_K)$ berücksichtigt. Darauf wurde bereits anläßlich der Definition kongruenter Terme hingewiesen. Die dort vorgetragenen Argumente zugunsten des allgemeineren Ansatzes beliebiger Terme gelten auch hier. Daher brauchen die nachfolgenden Erläuterungen nicht auf Grundterme beschränkt zu werden.

16) Da es sich ausschließlich um einfache zweistellige Basisformeln handelt, wird die allgemeine Präfixnotation für K-stellige Relationen und Formeln durch die übersichtlichere Infixnotation ersetzt. Hierfür gilt: $te_1 \ for_{REL} \ te_2 :\Leftrightarrow for_{REL}(te_1,te_2)$ und $te_1 \ REL \ te_2 :\Leftrightarrow (te_1,te_2) \in REL$. Der Übersichtlichkeit halber werden für die Infixe "for_{REL}" und "REL" dieselben Symbole benutzt. Ihr Charakter als Formel- bzw. Relationsnamen wird jeweils durch den Ausdruckskontext determiniert.

17) Die (sortenspezifische) Termgleichheit anstelle der Termkongruenz verwendet z.B. auch KRÄMER (1987a), S. 271 u. 275.

18) Sie müssen noch nicht einmal zur selben Sorte gehören.

19) Vgl. z.B. EHRIG (1985a), S. 13f. u. 30; RECK (1988), S. 47 u. 49; REISIG (1989a), S. 8.

20) Die Formulierung $te \in sort_i$ ist zweifelhaft, weil sie einen Mengencharakter von Sorten suggeriert, obwohl Sorten keine Mengen darstellen. Eine ähnlich problematische Bezugnahme auf Sorten in den algebraischen Formeln von SIG-Spezifikationen findet sich bei EHRIG (1984b), S. 4. Dort werden Sortenbezeichnungen in Gleichungen wie Konstantensymbole behandelt. Dagegen wird in EHRIG (1985a), S. 10f., deutlich darauf hingewiesen, daß die "Gleichungen" von SIG-Spezifikationen keine Gleichungen im strengen Sinne sind, sondern erst dann zu echten Gleichungen führen, wenn ihre Komponenten durch Konstanten und Operationen einer SIG-Algebra ersetzt werden. So sprechen auch EHRIG (1985a), S. 25f., und RECK (1988), S. 47f., exkulpierend von "Gleichungen in vereinfachter Form" (RECK). Die formal korrekte, aber auch aufwendigere Notation von Gleichungen und ihren zugrundeliegenden Sorten findet sich bei EHRIG (1985a), S. 25; RECK (1988), S. 47.

21) Die Benutzung der "$\in$"-Formel ist in Spezifikationen für die Kennzeichnung der Sortenzugehörigkeit von Termen te mit $te \in TM_i$ keineswegs notwendig. Sie läßt sich dadurch vermeiden, daß eine Restriktionsgleichung $te = Op_j$ formuliert und die Zielsorte der 0-stelligen Operation Op_j durch $Op_j: \rightarrow sort_i$ festgelegt wird. Oder es wird in den Formeln einer SIG-Spezifikation überhaupt nicht auf Terme, sondern nur auf das einfachere Konstrukt der Konstantensymbole Bezug genommen. Die Sortenzugehörigkeit dieser Konstantensymbole Ko_j wird dann wieder in der Sektion "Ops" durch Ausdrücke der Form $Ko_j: \rightarrow sort_i$ festgelegt. Vorteilhaft hierbei ist, daß wegen des Verzichts auf Terme - bis auf die neu hinzugefügten Gleichungen - überhaupt keine Erweiterungen gegenüber Signaturen SIG=(SO,OP) erfolgen zu brauchen. Diese formal korrekte Vorgehensweise findet sich z.B. bei EHRIG (1984a), S. 11. Sie hat aber den Nachteil, daß in der <u>Res</u>-Sektion der restriktive Charakter der Anforderung, das Konstantensymbol Ko_j müsse zur Sorte $sort_i$ gehören, nicht unmittelbar offensichtlich wird. Darüber hinaus ist die formale Ausdruckskraft geringer als bei der Verwendung von Termen. Insbesondere der Aspekt, in den Formelargumenten Variablen ansprechen zu können, geht verloren.

22) Vgl. EHRIG (1985a), S. 11ff.; EHRIG (1987), S. 4f. (dort allerdings "<u>in</u>" statt "$\in$"); RECK (1988), S. 49; REISIG (1989a), S. 8f. u. 23.

23) Das Attribut "aussagenlogisch" bezieht sich hier nur auf den Aspekt der Formelverknüpfung oder -negation. Bei diesen Operationen wird von den inneren Strukturen der Formeln abstrahiert. Dies bedeutet insbesondere, daß Formelquantifizierungen durch Existenz- oder Allquantoren hier noch nicht berücksichtigt werden. Jenseits dieser rein aussagenlogischen Perspektive besitzen die jeweils betrachteten Formeln innere Strukturen, da sie als mehrstellige Relationen über Termen eingeführt wurden. Insofern stellen die Formeln selbst bereits prädikatenlogische Ausdrücke dar. Diese prädikatenlogisch definierte innere Formelstruktur spielt aber anläßlich der hier betrachteten Formeloperationen noch keine Rolle.

24) Die redundante Einführung aussagenlogischer Formeloperationen dient der kompakteren Formelnotation.

25) Vgl. SMULLYAN (1968), S. 10; RECK (1988), S. 80.

26) Die algebraischen Formeln werden in der Regel nur als Elemente der Restriktionsmenge RES aufgezählt. Diese Aufzählung gilt aber implizit stets als eine Konjunktion der Restriktionsformeln. Eine Konjunktion ist genau dann gültig, wenn *alle* ihre Konjuganden gültig sind.

27) Besonders deutlich wird dies bei DIMITROVICI (1990a), S. 188: Dort wird die Restriktion, daß die Länge ("length") einer Warteschlange ("q") einen bestimmten Wert ("n-1") nicht überschreiten soll, nicht etwa als die

naheliegende Ungleichung "length(q)≤n-1" formuliert. Vielmehr wird auf das logische Ausdrucksmittel einer Wahrheitsfunktion zurückgegriffen. Sie bildet die voranstehende Ungleichung auf den Wahrheitswert "true" ab, falls die Warteschlangenlänge den spezifizierten Wert tatsächlich nicht überschreitet. Durch die Kombination von algebraisch formulierter Ungleichung und logischer Wahrheitsfunktion resultiert dann bei DIMITROVICI (1990a), S. 188, die Restriktionsspezifikation in der Gleichungsnotation "(length(q)≤n-1)=true". Abgesehen davon, daß der Funktor der implizit benutzten Wahrheitsfunktion fehlt, empfindet der Verf. diese Notationsweise als unnötig kompliziert und artifiziell.

28) Vgl. dazu das Beispiel, das bereits in einer früheren Anmerkung diskutiert wurde. Diesbezüglich zieht der Verf. die Restriktionsformulierung "length(q)≤n-1" der Gleichungsnotation "(length(q)≤n-1)=true" vor.

29) Ein weiteres anschauliches Beispiel findet sich bei ALLEN,J. (1984), S. 128ff. Er stützt sich zwar nicht unmittelbar auf das hier vorgestellte Signaturkonzept. Statt dessen verwendet er den Darstellungsmodus einer sortierten Logik. Diese läßt sich jedoch auf eine Überlagerung der Prädikatenlogik durch das Signaturkonzept zurückführen. Besonders interessant an ALLEN's Exempel ist, daß er das oben eröffnete unbeschränkte Formulierungspotential für algebraische Relationen weitgehend ausschöpft. Anstelle der sonst dominierenden Vergleichsrelationen "≡" formuliert er eine vollständige Liste von Zeitrelationen. Mit ihrer Hilfe ist es möglich, alle Präzedenzbeziehungen zwischen endlichen Prozessen prädikatenlogisch auszudrücken. Die Repräsentation solcher zeitlicher Abhängigkeitsverhältnisse wird später bei der Entfaltung Synthetischer Netze noch eine erhebliche Rolle spielen. Im Gegensatz zur dort erfolgenden Anwendung spezieller Netzkonstrukte verknüpft ALLEN seine Zeitrelationen in genau der aussagenlogischen Weise, wie sie oben für SIG-Spezifikationen eingeführt wurde; vgl. ALLEN,J. (1984), S. 129ff. Daher kann sein Beispiel - auch wenn es sich nicht explizit auf das Signaturkonzept bezieht - als instruktive Veranschaulichung des Ausdruckspotentials von SIG-Spezifikationen auf prädikatenlogischer Basis betrachtet werden.

30) Vgl. RECK (1988), S. 49, dort allerdings umfangreicher.

4.2 Prädikatenlogische Beiträge zur Konzeptverfeinerung

4.2.1 Überblick

Die Prädikatenlogik 1. Stufe[1)] stellt - neben Petrinetzen und Signaturen - das dritte Hauptkonzept der vorliegenden Ausarbeitung dar. Ihre Bedeutung resultiert aus der Modellierungsprämisse, daß sich in einer ersten Annäherung[2)] alle logischen Aspekte der Koordinierung komplexer Produktionssysteme mit Hilfe der Prädikatenlogik 1. Stufe ausdrücken lassen. Daher wird das Objektwissen[3)], das eine vorgegebene Koordinierungsaufgabe beschreibt und als bekannt vorausgesetzt wird, grundsätzlich durch prädikatenlogische Formeln ausgedrückt. Der gesamte Realitätsausschnitt, der auf diese Weise prädikatenlogisch repräsentiert wird, stellt das prädikatenlogische Objektmodell der interessierenden Koordinierungsaufgabe dar. Dieses Objektmodell bildet den Ausgangspunkt für die Konstruktion von Netzmodellen.

Durch die prädikatenlogische Strukturierung des vorausgesetzten Modellierungswissens wird das hier entfaltete Modellierungskonzept wesentlich geprägt. Diese Art der Wissensstrukturierung wird nicht durch die zu bewältigende Koordinierungsaufgabe determiniert. Vielmehr handelt es sich um eine konzeptionelle Basisentscheidung des Verfassers. An ihrer Stelle hätte grundsätzlich auch eine andere Logikkonzeption[4)] für die Repräsentation der logischen Modellierungsaspekte ausgewählt werden können[5)]. Angesichts dieser Wahlfreiheit wird die Basisentscheidung zugunsten der Prädikatenlogik 1. Stufe durch einen Katalog von Plausibilitätsargumenten gerechtfertigt:

- Die Prädikatenlogik liegt bereits den Höheren Netzen zugrunde, zu denen auch die später vorgestellten Synthetischen Netze gehören. Daher kann auf eine breit und tief ausgearbeitete prädikatenlogische Basis des Petrinetz-Konzepts zurückgegriffen werden. Dies erspart Entwicklungsaufwand, der für andere, noch nicht derart ausgereifte formallogische Konzepte anfallen würde.

- Darüber hinaus widmen sich schon mehrere Ansätze der speziellen Aufgabe, (prädikaten)logische[6)] Sachverhaltsbeschreibungen unmittelbar in Repräsentationen durch Petrinetze zu übersetzen, um sie hierdurch weiteren netztheoretischen Analysen zu erschließen[7)].

- Das Signaturkonzept läßt sich benutzen, um den Kalkül der Prädikatenlogik strukturell anzureichern. Es resultiert ein sortierter prädikatenlogischer Kalkül. Daher besitzt die Prädikatenlogik den Vorzug, nicht isoliert neben dem bereits eingeführten Signaturkonzept zu stehen. Vielmehr liefert die Integration von beiden einen strukturell besonders reichhaltigen Konzeptverbund.

- Die Prädikatenlogik ist eine wesentliche Stütze der Wissensrepräsentation im Bereich der Künstlichen Intelligenz[8)]. Dies liegt vor allem daran, daß sich die meisten Inferenzkonzepte[9)], mit deren Hilfe "intelligente" Automaten aus implizitem Wissen neues explizites Wissen erschließen[10)], auf dem prädikatenlogischen Kalkül beruhen. Daher kann die Prädikatenlogik quasi als "natürliche" formallogische Basis der Künstlichen Intelligenz betrachtet werden[11)]. Daneben besitzt die Prädikatenlogik eine enge Verwandtschaft mit einem zweiten herausragenden Repräsentationskonzept der Künstlichen Intelligenz: den Produktionsregelsystemen[12)]. Beide Aspekte eröffnen ein Potential befruchtender Impulse für die Ausgestaltung[13)] und Fortentwicklung[14)] von Synthetischen Netzen durch Adaption von Konzepten der KI-Forschung.

- Auch zwischen Prädikatenlogik und Systemtheorie besteht eine "natürliche Entsprechung[15]. Denn die systemtheoretischen Konstrukte lassen sich unmittelbar in prädikatenlogische Konstrukte überführen[16]. Diese Korrespondenz ist hier insofern bedeutsam, als auch der Strukturierung des Realproblems in dieser Arbeit der systemtheoretische Ansatz zugrundegelegt wurde.

- Es wurden bereits einige Arbeiten präsentiert, welche die Modellierung Flexibler Fertigungssysteme mit Hilfe der Prädikatenlogik versucht haben[17]. Ihre Ergebnisse deuten auf die grundsätzliche Eignung und Ausbaufähigkeit des prädikatenlogischen Ansatzes für diese spezielle Modellierungsaufgabe hin.

- Die Beziehungen der Prädikatenlogik zu Petrinetzen, Signaturkonzept, Künstlicher Intelligenz, Systemtheorie und Modellierungen Flexibler Fertigungssysteme zeichnen diese formale Logik aus. Sie bildet eine konzeptionelle Schnittstelle[18] zwischen bemerkenswert unterschiedlichen Konzepten. Zugleich stellt sie hierdurch den Ansatzpunkt für eine formallogisch gesicherte Konzeptintegration zur Verfügung.

- Die Prädikatenlogik besitzt einen annähernd "optimalen" Komplexionsgrad[19]. Dieses Optimalitätsurteil wird im Sinne des "faute de mieux"-Kriteriums[20] praktischer Rationalität[21] verstanden: Eine alternative Logikkonzeption mit noch besserem Komplexionsgrad steht nach vorherrschender Meinung in der einschlägigen Literatur nicht zur Verfügung[22]. Dabei wird dem Komplexionsvergleich[23] einerseits die einfachere Aussagenlogik[24] und die monadische Prädikatenlogik 1. Stufe[25] als Referenzkonzepte zugrundegelegt. Andererseits dienen die Prädikatenlogiken 2. oder höherer Stufe[26] sowie die Varianten der Modallogik[27] als Bezugspunkte höherer Komplexität.

- Die Prädikatenlogik stellt ein formallogisches Konzept dar, das infolge seiner transparenten Darstellungsweise und seiner leistungsfähigen Analyseinstrumente breite Akzeptanz genießt[28].

- Die Programmiersprache PROLOG[29] erlaubt es, prädikatenlogisch formulierte Modelle unmittelbar auf Automatischen Informationsverarbeitungssystemen zu implementieren[30]. Hierauf baut auch das Softwarepaket PASIPP auf, das für die prädikatenlogische Implementierung von Netzmodellen umfangreiche Darstellungs- und Analysemöglichkeiten bietet[31].

- Die Sprache PROLOG erweist sich als ein "interessantes" Instrument für die Implementierung von prädikatenlogischen Modellierungen[32]. Insbesondere zeichnet sich diese Sprache durch ein hohes Integrationspotential aus[33]. Sie gestattet es, drei weitere Konzepte in die automatengestützte Implementierung prädikatenlogischer Objektmodelle einzubeziehen, die in dieser Arbeit eine größere Rolle spielen. Hierbei handelt es sich erstens um Konstruktionen, mit deren Hilfe sich *objektorientierte* System*strukturierungen* unmittelbar in objektorientiert strukturierte PROLOG-Programme übersetzen lassen[34]. Zweitens bestehen Bestrebungen, das kombinierte Unifizierungs- und Resolutionskonzepts durch *nebenläufige* Informationsverarbeitungs*prozesse* zu realisieren[35]. Drittens läßt sich die prädikatenlogische Syntax von PROLOG-Programmen durch eine *Sortenstruktur* überlagern[36].

- Die Implementierung prädikatenlogischer Formelsysteme wird durch ausgereifte relationale Datenbanksysteme[37] unterstützt[38]. Daher bereitet es grundsätzlich keine Schwierigkeiten, auch großvolumige prädikatenlogische Modellierungen realer Systeme mittels der Automatischen Informationsverarbeitung zu beherrschen[39]. Auch die vorgenannte Programmiersprache PROLOG gestattet es, den prädikatenlogischen Modellimplementierungen relationale Datenbanksysteme zugrundezulegen[40]. Für das Softwarepaket PASIPP steht eine solche relationale Datenbankanbindung bereits zur Verfügung[41].

Es lassen sich weitere Vorteile der Prädikatenlogik 1. Stufe - wie etwa deren Modularität[42] - anführen, die jedoch im Rahmen dieser Arbeit nicht näher interessieren[43]. Statt dessen besitzt jedoch das Argument des optimalen Komplexionsgrads aus der Perspektive, ein Modellierungskonzept für Realprobleme zu entwickeln, eine besondere Rolle[44]. Daher wird es nachfolgend auf drei Ebenen konkretisiert. Die erste betrifft die Ausdruckskraft einer Logik. Die zweite erstreckt sich auf die logische Inferenzqualität[45]. Die dritte berührt eine abstrakte "metalogische"[46] Ebene. Auf ihr wird untersucht, welche globalen Eigenschaften sich von einem formallogischen Konzept *gleichzeitig* erfüllen lassen[47].

Die Ausdruckskraft der Prädikatenlogik 1. Stufe erweist sich im Rahmen dieser Arbeit sowohl erforderlich als auch nahezu ausreichend. Daher kommt ihr für das hier entfaltete netzbasierte Modellierungskonzept eine fast[48] optimale Ausdruckskomplexion zu:

- Einerseits reicht das Formulierungspotential der Prädikatenlogik 1. Stufe aus[49], um nahezu alle formallogischen Bedürfnisse für die Definition Synthetischer Netze und die spätere Modellierung Flexibler Fertigungssysteme zu erfüllen. Diese Ausdrucksmächtigkeit kommt der einfacheren Aussagenlogik dagegen bei weitem nicht zu[50]. Das prädikatenlogische Ausdruckspotential muß lediglich in zwei Richtungen ergänzt werden:
  - Erstens werden geringfügige Anleihen bei der Modallogik erforderlich[51].
  - Zweitens ist es notwendig, die Monotonieprämisse der konventionellen Prädikatenlogik aufzugeben, um Wissen über zeitlich veränderliche Sachverhalte adäquat repräsentieren zu können.
- Andererseits ist die prädikatenlogische Ausdruckskraft 1. Stufe auch *notwendig*, um Synthetische Netze auf logischer Grundlage zu definieren. Beispielsweise lassen sich die strukturierten Marken von Synthetischen Netzen auf dem Fundament der Aussagenlogik grundsätzlich nicht darstellen[52].

Auch hinsichtlich ihrer Inferenzqualität befindet sich die Prädikatenlogik 1. Stufe in der engen Nähe eines optimalen Komplexionsniveaus[53]. Diese Niveaubewertung läßt sich auf zwei Dimensionen für die Inferenzqualität beziehen: die Inferenzmächtigkeit und die Inferenzeffizienz.

Die Inferenzeffizienz wird hinsichtlich der Möglichkeit beurteilt, Inferenzprozesse zu automatisieren. Dieses Automatisierungspotential einer Logik hängt von der Vollständigkeit und Korrektheit ihrer Beweissysteme ab. Diese Beweissysteme beruhen auf rein syntaktisch definierten Inferenzkonzepten. Falls mindestens ein Beweissystem die Eigenschaften der Vollständigkeit[54] und Korrektheit[55] erfüllt, dann können alle Schlußfolgerungen, die in der untersuchten Logik semantisch zulässig sind, durch die syntaktischen Inferenzkonzepte dieses Beweissystems äquivalent ersetzt werden[56]. Nur solche syntaktischen, "mechanischen" Inferenzkonzepte lassen sich ohne Schwierigkeiten vollautomatisch implementieren[57]. Daher besitzen nur vollständige und korrekte Logiken[58] das Potential, Inferenzprozesse durch Automatisierung relativ[59] effizient abzuwickeln[60].

- Für die Prädikatenlogik 1. Stufe stehen tatsächlich vollständige und korrekte Inferenzkonzepte zur Verfügung[61]. Daher lassen sich prädikatenlogische Beweisführungen grundsätzlich vollautomatisch ausführen[62]. Insbesondere die Programmiersprache PROLOG bietet relativ effiziente Implementierungen[63] vollautomatischer Inferenzprozesse auf prädikatenlogischer Basis an[64]. Es bleiben zwar derzeit durchaus noch Wünsche hinsichtlich der absoluten Ausführungsdauer umfangreicher PROLOG-Programme offen[65]. Doch geben mehrere Ansätze berechtigten Anlaß zur Hoffnung, daß sich diese Effizienzschwächen mittelfristig überwinden lassen. Dazu gehört die bereits erwähnte Möglichkeit, PROLOG-Programme in nebenläufiger Weise abarbeiten zu lassen[66] und mit einer *Sortenstruktur* zu überlagern[67]. Hinzu kommen weitere Bemühungen, die Ausführungseffizienz durch Verbesserungen der algorithmischen Implementierungen von Unifizierungs- und Resolutionskonzept zu steigern[68]. Es erfolgen sogar Anstrengungen, mit PROLOG-Programmen Realzeitanforderungen für die kontinuierliche Steuerung realer Prozesse zu erfüllen[69].

- Bereits für die nächstkomplexeren formallogischen Konzepte stehen keine vollständigen und korrekten Inferenzkonzepte mehr zur Verfügung. So erweisen sich die Prädikatenlogik 2. Stufe[70)] und verschiedene Varianten der Modallogik[71)] als unvollständig[72)]. Darüber hinaus kennt der Verf. für beide Logikkonzeptionen auch keine umfassenden Automatenimplementierungen. Die wenigen bisher vorliegenden Ansätze, Inferenzkonzepte für höherstufige logische Kalküle zumindest teilweise zu implementieren, haben sich bisher als enttäuschend ineffizient herausgestellt[73)].

- Die einfachere Aussagenlogik erweist sich dagegen ebenso wie die Prädikatenlogik 1. Stufe als vollständig und korrekt[74)]. Darüber hinaus lassen sich ihre Inferenzkonzepte sogar wesentlich effizienter implementieren[75)] als die vollständigen und korrekten Inferenzkonzepte der Prädikatenlogik 1. Stufe. Doch schlägt hier abermals das schon oben angeführte Argument der allzu kargen aussagenlogischen Ausdruckskraft durch. Die Ausdrucksdefizite der Aussagenlogik übersteigen nach Einschätzung des Verf. ihre Effizienzvorteile so sehr, daß sich die Reduzierung des logischen Komplexionsniveaus auf die Aussagenlogik nicht rechtfertigen läßt[76)].

Die Inferenzmächtigkeit einer Logik wird hier hinsichtlich ihrer Universalität bewertet. Eine Logik besitzt universelle Inferenzmächtigkeit, falls für sie mindestens ein Inferenzkonzept existiert, welches das allgemeine Gültigkeitsproblem löst. Das Gültigkeitsproblem[77)] besteht in der Aufgabe, für *jede* wohlgeformte[78)] logische Formel mit beliebig großem, aber endlichem Ressourceneinsatz entscheiden zu können, ob sie allgemeingültig[79)] ist oder nicht[80)]. Dieses Gültigkeitsproblem wird oftmals als "das" logische Entscheidungs- oder Entscheidbarkeitsproblem bezeichnet[81)]. Daher wird - in vereinfachter Diktion - von der Entscheidbarkeit einer Logik genau dann gesprochen, wenn mindestens ein Inferenzkonzept existiert, das ihr Gültigkeitsproblem löst. Andernfalls heißt die Logik unentscheidbar.

- Die Prädikatenlogik verhält sich leider nicht universell. Ihre Unentscheidbarkeit ist seit längerem bekannt[82)].

- Die nächsteinfacheren Logiken zeichnen sich zwar durch ihre universelle Inferenzmächtigkeit aus. Die Entscheidbarkeit ist sowohl für die monadische Prädikatenlogik 1. Stufe[83)] als auch für die Aussagenlogik[84)] nachgewiesen. Aber die aussagenlogischen Inferenzen bleiben auf das spärliche Ausdruckspotential der Aussagenlogik beschränkt. Gehaltreiche Beweise lassen sich auf dieser Basis nicht ausführen[85)]. Auch die monadische Prädikatenlogik 1. Stufe leidet unter einem beträchtlich restringierten Ausdrucksvermögen. Da sie nur einstellige Prädikate zuläßt, vermag sie keine relationalen Sachverhalte in "natürlicher" Weise auszudrücken[86)]. Bereits einfache alltägliche Schlußweisen des "gesunden Menschenverstands" lassen sich mit der monadischen Prädikatenlogik nicht erfassen[87)].

- Beim Übergang zu Logiken, die noch komplexer als die Prädikatenlogik 1. Stufe sind, stellt sich zunächst heraus, daß sie ebenso unentscheidbar sind. Dies gilt z.B. für die Varianten der Modallogik[88)]. Darüber hinaus erweisen sich bei komplexeren Logiken die Defizite mangelnder Entscheidbarkeit sogar als wesentlich gravierender als in der Prädikatenlogik. Denn in der Prädikatenlogik erlangt ihre generelle Unentscheidbarkeit nur für einige wenige spezielle Entscheidungsprobleme unmittelbare Bedeutung[89)]. In der exemplarisch angeführten Modallogik wirkt sich dagegen die Unentscheidbarkeit auf alle nicht-trivialen Entscheidungsprobleme erheblich komplizierend aus[90)].

Die Relevanz der prädikatenlogischen Unentscheidbarkeit für die hier vorgelegten Untersuchungen wird nachfolgend aus vier unterschiedlichen, paarweise gegenläufigen Perspektiven näher beleuchtet.

Erstens kann die Unentscheidbarkeit des Gültigkeitsproblems von keinem logischen Konzept vermieden werden, daß mindestens die Ausdruckskraft der Prädikatenlogik 1. Stufe besitzt. Auf diesen Formulierungsreichtum soll für die hier interessierende Modellierung Flexibler Fertigungssysteme angesichts des früher begründeten Primats der Modellierungsgüte grundsätzlich nicht verzichtet werden. Darüber hinaus wurde schon dargelegt, daß die Prädikatenlogik für die Formulierung Synthetischer Netze notwendig ist. Daher resultiert die prädikatenlogische Unentscheidbarkeit zwangsläufig aus den beiden vorgenannten Basisentscheidungen, für die Realproblemabbildung und das dazu eingesetzte Petrinetz-Konzept die prädikatenlogische Ausdruckskraft (mindestens) zur Verfügung zu stellen. Dieser Sachverhalt läßt sich auch in der Diktion des oben erwähnten "faute de mieux"-Kriteriums reformulieren: Neben der Prädikatenlogik 1. Stufe ist keine alternative Logik bekannt, die entscheidbar ist, ohne diese Universalität durch eine nicht mehr akzeptierte Einbuße ihrer Ausdrucksmächtigkeit zu realisieren[91)].

Zweitens wird die praktische Bedeutung der prädikatenlogischen Unentscheidbarkeit zunächst dadurch erheblich geschmälert[92)], daß sie sich nicht unmittelbar auf betriebswirtschaftlich interessante Problemstellungen bezieht[93)]. Denn sie betrifft prima facie nur die Existenz unentscheidbarer Entscheidungsprobleme oder die Nichtexistenz von Entscheidungsalgorithmen, die beide so abstrakt formuliert sind, daß sie keine Verwandtschaft mit Problemen aus der betrieblichen Entscheidungspraxis erkennen lassen[94)]. Daher bleibt die Unentscheidbarkeit der Prädikatenlogik in einer ersten Annäherung ein theoretisch bemerkenswertes, aber praktisch irrelevant anmutendes Kuriosum.

Drittens folgen aber aus der prädikatenlogischen Unentscheidbarkeit Konsequenzen, die im Prinzip *jedes* prädikatenlogisch formulierte Entscheidungsproblem betreffen können[95)]. Damit gelten diese indirekten Wirkungen der Unentscheidbarkeit auch für alle betriebswirtschaftlich interessanten Problemmodellierungen auf prädikatenlogischer Basis[96)]. Dies betrifft insbesondere auch die hier angestrebte Modellierung der Koordinierung von Flexiblen Fertigungssystemen mit prädikatenlogisch fundierten Synthetischen Netzen. Denn zusammen mit der semantischen Vollständigkeit und Korrektheit der Prädikatenlogik führt deren Unentscheidbarkeit dazu, daß *alle* prädikatenlogischer Inferenzkonzepte unter dem Phänomen ihrer Semi-Entscheidbarkeit leiden. Es bedeutet in einer ersten Annäherung[97)], daß sich Entscheidungsprobleme, in denen die Allgemeingültigkeit mindestens einer prädikatenlogischen Formel behauptet wird, grundsätzlich nur dann mit endlichem Ressourceneinsatz lösen lassen, wenn diese Formel tatsächlich allgemeingültig ist. Falls die behauptete Allgemeingültigkeit der Formel dagegen tatsächlich nicht zutrifft, so kann dies von keinem endlichen Entscheidungsalgorithmus *mit Sicherheit* erkannt werden[98)]: Das untersuchte Entscheidungsproblem ist *potentiell* unlösbar[99)].

Bei der Formulierung eines Entscheidungsproblems ist es in der Regel[100)] zunächst unbekannt, ob die dabei behauptete Allgemeingültigkeit einer Formel tatsächlich zutrifft oder nicht. Daher bedeutet die Semi-Entscheidbarkeit, daß es im allgemeinen unmöglich ist, einem Entscheidungsproblem aufgrund seiner Formulierung anzusehen, ob seine prinzipielle Lösbarkeit mit beliebig großem, aber endlichem Ressourceneinsatz gesichert ist. Erschwerend kommt in der Praxis hinzu, daß die Entscheidungsressourcen nicht nur endlich sind, sondern auch nicht in jedem beliebigen endlichen Ausmaß zur Verfügung gestellt werden können[101)]. Daher beeinträchtigt die Semi-Entscheidbarkeit die praktische Zuverlässigkeit *grundsätzlich aller* Lösungskonzepte, die auf prädikatenlogisch formulierte Entscheidungsprobleme angewendet werden. Die Zuverlässigkeitsmängel können als Ineffektivität, Inkorrektheit und Formulierungssensitivität der Bearbeitung von Entscheidungsproblemen praktisch bedeutsam werden. Dies gilt insbesondere auch für das Unifizierungs- und Resolutionskonzept der prädikatenlogischen Programmiersprache PROLOG, die in dieser Arbeit zur Implementierung von Netzmodellen herangezogen wird. Wegen dieser eminenten Bedeutung wird das Phänomen der Semi-Entscheidbarkeit später als wesentlicher Ausschnitt aus der operationalen Dimension der Prädikatenlogik ausführlicher diskutiert.

Viertens bedeutet die prädikatenlogische Semi-Entscheidbarkeit aber nur, daß sich die vorgenannten Zuverlässigkeitsmängel bei allen prädikatenlogisch formulierten Entscheidungsproblemen auswirken *können.* Ob sie beim jeweils untersuchten Entscheidungsproblem *tatsächlich* wirken werden, kann nur empirisch durch Bearbeiten konkreter Entscheidungsprobleme festgestellt werden. Im Rahmen der Untersuchungen dieser Arbeit haben sich die potentiellen Mängel an *keiner* Stelle tatsächlich ausgewirkt. Das Untersuchungsdesign wurde jedoch vornehmlich auf die Ausdrucksmächtigkeit der vorgelegten Modellierungen und nicht auf die Lösung konkret modellierter Probleme ausgerichtet. Daher kommt der fehlenden Manifestation von Zuverlässigkeitsmängeln keine repräsentative Aussagekraft zu. Entsprechend breit angelegte Zuverlässigkeitsuntersuchungen für umfangreiche Sammlungen von praktisch interessanten Entscheidungsproblemen liegen außerhalb des Erkenntnisinteresses dieser Arbeit.

Schließlich lassen sich die voranstehenden speziellen Facetten optimaler formallogischer Komplexion verallgemeinern. Aus metalogischer Perspektive weist die Prädikatenlogik 1. Stufe die derzeit reichste Kollektion "wesentlicher" Erkenntnisse auf[102)]. Insbesondere kann gezeigt werden, daß die Prädikatenlogik 1. Stufe die ausdrucksstärkste Logik darstellt, in der gerade noch zwei zentrale Theoreme über die Interpretationen von logischen Kalkülen gelten[103)]: das LÖWENHEIM-SKOLEM-Theorem[104)] und das Kompaktheits-Theorem[105)]. Auch diese metalogische Erkenntnis stützt die eingangs vorgetragene These, die Prädikatenlogik 1. Stufe repräsentiere ein nahezu optimales Komplexionsniveau.

Trotz der oben aufgeführten breiten Palette von Vorzügen besitzt die Prädikatenlogik durchaus auch Nachteile[106)]. Auf die Notwendigkeit einer modallogischen Erweiterung wurde bereits hingewiesen. Die Probleme der prädikatenlogischen Unentscheidbarkeit und Semi-Entscheidbarkeit wurden ebenso schon thematisiert. Auf andere Mängel wird später zurückgekommen[107)]. Jedoch wird jeweils aufgezeigt werden, daß sich diese zusätzlichen Schwächen im Rahmen der Prädikatenlogik und ihrer PROLOG-basierten Implementierung überwinden lassen.

Die konventionelle Komponente der prädikatenlogischen Basis von Synthetischen Netzen wird anschließend in drei Dimensionen entfaltet. Die erste erstreckt sich auf den objektsprachlichen Kalkül der Prädikatenlogik 1. Stufe. Mit seiner Hilfe wird die Gesamtheit aller zulässigen prädikatenlogischen Formeln rein syntaktisch definiert. Die semantische Dimension betrifft dagegen die Interpretation solcher Formeln. Sie führt als wesentliche metasprachliche Neuerung das Konzept der Prädikatsextensionen ein. Diese Extensionen werden später mit Hilfe des daraus abgeleiteten Faktenbegriffs die Brücke zu Synthetischen Netzen schlagen. In der operationalen Dimension wird schließlich näher untersucht, wie sich prädikatenlogische Formelsysteme durch die Anwendung prädikatenlogischer Inferenzkonzepte und ihrer automatengestützten Implementierung praktische handhaben lassen.

Anmerkungen zum Kapitel:

1) Vgl. zu Überblicken über die Prädikatenlogik 1. Stufe, die mitunter auch als Quantorenlogik bezeichnet wird, FREGE (1879), als erste vollständige' Präsentation; WHITEHEAD (1925), S. 14ff. u. 38ff., mit einer der frühesten umfassenden prädikatenlogischen Ausarbeitungen in ihrer heute üblichen Darstellungsweise; CHURCH (1932), S. 350ff.; CHURCH (1956), S. 47ff. u. 168ff.; CARNAP (1960a), S. 34ff. u. 79ff.; ACKOFF (1962), S. 10ff.; LORENZEN,P. (1962), S. 34ff.; MENDELSON (1964), S. 45ff. u. 56f.; RAMSEY,F. (1965), S. 25ff. u. 34ff.; CARNAP (1968), S. 12ff. u. 139ff.; STEGMÜLLER (1968), S. 10ff.; SMULLYAN (1968), S. 43ff.; CHANG,C.L. (1973), S. 26ff.; ROBINSON,J. (1975), S. 6ff.; ITZINGER (1976), S. 29ff.; OPP,K. (1976), S. 25ff. u. 115ff., der eine Einführung aus sozialwissenschaftlicher Richtung mit niedrigem Formalisierungsgrad vorlegt; ESSER,H. (1977a), S. 38ff.; RICHTER,M.M. (1978), S. 58ff.; KOWALSKI (1978), S. 77ff.; KOWALSKI (1979a), S. 425ff.; BULLERS (1980), S. 354ff.; NILSSON,N. (1980a), S. 131ff.; CLOCKSIN (1981), S. 207ff.; ATTARDI (1981), S. 504ff.; BARR,A. (1981), S. 154f. u. 160ff.; BIBEL (1982a), S. 58ff.; ESSLER (1982a), S. 34ff.; STEGMÜLLER (1983), S. 52ff. u. 87ff.; HABEL (1983), S. 121ff.; GRAHAM,N. (1983), S. 167ff.; STEGMÜLLER (1984b), S. 73ff.; TROST (1984), S. 49ff.; DUTTA,A. (1984), S. 90ff.; REITER (1984), S. 194ff.; SOWA (1984), S. 17ff. u. 386ff.; ZELEWSKI (1986a), S. 181ff.; DELAHAYE (1987), S. 66ff., mit einer inhaltlich umfassenden und zugleich kompakten Darstellung; BUCHER (1987), S. 161ff.; GREWENDORF (1987), S. 334ff.; PREIß (1989), S. 13ff.
Die Prädikatenlogik 1. Stufe wird in dieser Arbeit auch vereinfacht als Prädikatenlogik angesprochen. Die Prädikatenlogik und ihr Grenzfall - die Aussagenlogik - werden auch als konventionelle Logiken bezeichnet.

2) Diese Einschränkung berücksichtigt zwei Aspekte. Erstens wird später die prädikatenlogische Modellierung mit Hilfe des Signaturkonzepts durch Überlagerung einer Sortenstruktur verfeinert. Auf diese Weise wird die Ausdrucksmächtigkeit der Prädikatenlogik zwar nicht in dem grundsätzlichen Sinne vergrößert, daß die resultierende sortierte Prädikatenlogik gestattete, Sachverhalte zu repräsentieren, die sich im Rahmen der konventionellen Prädikatenlogik überhaupt nicht ausdrücken lassen. Aber die sortierte Prädikatenlogik erlaubt kompaktere und zugleich subtilere Formulierungen als ihr konventionelles Pendant. Zweitens werden später echte Erweiterungen der Prädikatenlogik eingeführt, um spezielle Modellierungserfordernisse erfüllen zu können. Sie erstrecken sich einerseits auf modallogische Ergänzungen und andererseits auf zeitlich variable Faktenmengen.

3) Es wird wieder der weit gefaßte Objektbegriff der objektorientierten Systemgestaltung zugrundegelegt. Daher kann der gesamte modellierungsrelevante Realitätsausschnitt als ein - komplex strukturiertes - Objekt angesehen werden.

4) Vgl. zu einem Überblick über alternative Logikkonzeptionen z.B. STEGMÜLLER (1986a), S. 152ff.

5) Daß die Auswahl einer Logikkonzeption gerechtfertigt werden sollte, hat GETHMANN (1980b), S. 20ff., überzeugend dargelegt. Er zeigt auf, daß die Auswahlentscheidung die Ergebnisse zu beeinflussen vermag, die mit formallogisch fundierten Konzepten erarbeitet werden. Daher handelt es sich bei der Entscheidung für eine bestimmte Logik keineswegs um eine konsequenzenlose Konvention, sondern um eine Basisentscheidung von epistemischer Relevanz. Dies gilt vor allem in Hinblick auf die Entstehung von Antinomien (Paradoxien), die in manchen Logiken auftreten - in anderen dagegen nicht; vgl. GETHMANN (1980b), S. 22ff.; HEGSELMANN (1982), S. 349ff. (allerdings mit einer schwer nachvollziehbaren Konklusion auf S. 357f.). Vgl. ebenso zur bewußten Auswahl einer Logikkonzeption DORN (1989), S. 82 u. 85.

6) Gemeint sind hier nicht nur prädikaten-, sondern auch aussagenlogische Sachverhaltsbeschreibungen. Hierauf wird in der anschließenden Anmerkung zurückgekommen.
Zwar läßt sich zeigen, daß bei zahlreichen praktischen Repräsentationsaufgaben die prädikatenlogische Sachverhaltsbeschreibung nachträglich in eine aussagenlogische transformiert werden kann. Dies gilt zumindest dann, wenn die Definitionsbereiche prädikatenlogischer Semantiken endlich sind. Denn derart finite prädikatenlogische Formelsysteme lassen sich grundsätzlich in entsprechend vervielfachte aussagenlogische Formelsysteme äquivalent transformieren. Dabei wird für jedes prädikatenlogische Individuum eine separate aussagenlogische Formel eingeführt. Dem entspricht die netztheoretische Operation der Entfaltung von prädikatenlogisch fundierten Prädikat/Transition-Netzen. Diese lassen sich , jedes in ein äquivalentes, aber aussagenlogisch fundiertes Stelle/Transition-Netz übersetzen, falls die Stellenprädikate jeweils über endlichen Individuenmengen definiert sind. Vgl. zu dieser finiten Netzentfaltung z.B. LAUTENBACH (1985b), S. 11f. u. 17. Vgl. insgesamt zur Möglichkeit, prädikatenlogisch ausgerichtete auf aussagenlogisch basierte Problembeschreibungen zurückzuführen, ZELEWSKI (1989c), S. 12 u. 24.
Allerdings handelt es sich bei den vorgenannten Reduzierungen der Prädikaten- auf die Aussagenlogik erstens um keine allgemeingültige und zweitens um oftmals sehr unhandliche Operationen. Der Allgemeingültigkeitsmangel folgt daraus, daß eine äquivalente Transformationsmöglichkeit für prädikatenlogische Formeln über unendlichen Definitionsbereichen nicht mehr garantiert ist. Die Unhandlichkeit manifestiert sich in zugeordneten aussagenlogischen Formelsystemen, die aufgrund ihrer Voluminösität vollkommen intransparent und praktisch unzugänglich werden. Daher widmet sich der Verf. von vornherein einer transparenten und kompakten, allgemeingültigen Wissensrepräsentation durch Synthetische Netze auf prädikatenlogischer Basis. An späterer Stelle wird ein Algorithmus

vorgestellt, mit dessen Hilfe sich jede prädikatenlogische Wissensrepräsentation in ein äquivalentes Netzmodell transformieren läßt, das sich auf Synthetische Netze stützt.

7) Vgl. THIELER-MEVISSEN (1975), S. 2ff.; THIELER-MEVISSEN (1977), S. 6ff., insbesondere S. 33ff.; ZISMAN (1978a), S. 57ff., insbesondere 61ff.; AZEMA (1984), S. 510ff.; MAINZ (1984), S. 11ff.; GIORDANA (1985), S. 3ff., insbesondere S. 8ff.; LAUTENBACH (1985b), S. 7ff. u. 29f.; FIDELAK (1986a), S. 19ff. u. 108ff.; FIDELAK (1986b), S. 32ff.; MURATA,TA. (1986), S. 132ff.; FIDELAK (1987a), S. 95ff.; OBERWEIS (1987b), S. 21ff.; LIU,N. (1987), S. 121ff.; MURATA,TA. (1987a), S. 381ff.; FRIEDRICHS,J. (1987), S. 171ff.; SAHRAOUI (1987), S. 1161ff.; OBERWEIS (1988b), S. 300ff.; MURATA,TA. (1988a), S. 73ff., insbesondere S. 82ff.; MURATA,TA. (1988b), S. 483ff.; BANDINI (1988), S. 869ff.; VON MARTIAL (1988a), S. 293ff.; FREEDMAN (1988a), S. 1276ff.; DUGGAN (1988), S. 239 u. 245f.; ATANASSOV (1989), S. 303ff.; FLEISCHHACK (1989), S. 23f. u. 28ff.; ZELEWSKI (1989c), S. 17ff.; O.V. (1989g), S. 2ff.; BAIER,W. (o.J.), S. 3ff.
Mehrere der vorgenannten Quellen beziehen sich auf die Aussagen- anstelle der Prädikatenlogik. Daher liegt in der Entwicklung Synthetischer Netze, welche die Prädikatenlogik 1. Stufe voll abdecken, durchaus ein Beitrag zur Erweiterung der netztheoretischen Ausdruckskraft. In diese Richtung weisen bereits die vornehmlich prädikatenlogisch ausgerichteten Arbeiten von MURATA, OBERWEIS und THIELER-MEVISSEN (1977).

8) Vgl. KOWALSKI (1978), S. 77ff.; KOWALSKI (1979a), S. 425ff.; BULLERS (1980), S. 354ff.; NILSSON,N. (1980a), S. 131ff.; ATTARDI (1981), S. 504ff.; BARR,A. (1981), S. 154f. u. 160ff.; MOORE,R. (1982), S. 428ff.; SCHEFE (1982), S. 50ff.; KOBSA (1982), S. 11ff.; HABEL (1983), S. 119 u. 121ff.; GRAHAM,N. (1983), S. 185ff.; MYLOPOULOS (1983), S. 144f. u. 151f.; BRACHMAN (1983a), S. 46; STEFIK (1983a), S. 61; DUTTA,A. (1984), S. 90ff.; TROST (1984), S. 49ff.; MYLOPOULOS (1984), S. 4f.; HARMON (1985), S. 46ff.; REINFRANK (1985b), S. 4; BEETZ (1986), S. 36ff.; ZELEWSKI (1986a), S. 181ff.; SCHARF,A. (1987), S. 23f.; RICHTER,M.M. (1988), S. 18; KLEINE BÜNING (1988a), S. 56; SPECHT,D. (1989), S. 18ff.; DORN (1989), S. 11 u. 13.

9) Dabei handelt es sich im wesentlichen um das kombinierte Unifizierungs- und Resolutionskonzept, das bereits eingeführt wurde.

10) Vgl. zu dem paradigmatischen Konzept der Künstlichen Intelligenz, durch Wissensverarbeitung (symbol processing u.ä.) aus implizitem Wissen explizites Wissen durch Inferenzen abzuleiten, ZELEWSKI (1986a), S. 171ff. u. 342ff. (mit vertiefenden Quellenangaben).

11) Vgl. RAPHAEL (1976), S. 134; BARR,A. (1981), S. 155 u. 170; COHEN,P. (1982), S. 83; NEWELL (1982a), S. 121f.; MYLOPOULOS (1983), S. 144; MYLOPOULOS (1984), S. 4f.; ZELEWSKI (1986a), S. 189.

12) Denn jede Produktionsregel läßt sich im Prinzip auf ein prädikatenlogisches Subjugat zurückführen. Auf Produktionsregelsysteme und deren prädikatenlogische Einbindung in das Konzept Synthetischer Netze wird an späterer Stelle ausführlicher zurückgekommen.

13) Vgl. dazu die Einbeziehung von Produktionsregelsystemen in die Formulierung von Transitionen mit komplexer Schaltlogik.

14) Vgl. die Ausführungen zur Einbindung von KI-Konzepten in die Gestaltung der Lösungssuche in Erreichbarkeitsgraphen.

15) Vgl. ZELEWSKI (1986a), S. 188f.

16) Jedes atomare Objekt (Element) eines Systems entspricht einem prädikatenlogischen Individuum. Jedes zusammengesetzte Objekt läßt sich als ein prädikatenlogischer Term induktiv aus Individuen aufbauen. Jede Objekteigenschaft kann mittels eines einstelligen Prädikats ausgedrückt werden. Alle Beziehungen zwischen Objekten lassen sich durch mehrstellige Prädikaten erfassen.

17) Vgl. BULLERS (1980), S. 354ff., insbesondere S. 357f.; DANGELMAIER (1987a), S. 83ff. (dort wird sogar die Programmiersprache PROLOG zur Modellimplementierung verwendet, die auch dieser Arbeit zugrundeliegt).

18) Vgl. zu diesem Schnittstellencharakter der Prädikatenlogik auch ZELEWSKI (1986a), S. 191.

19) Vgl. zu einer ähnlichen Abwägung "ideeller Kosten und Nutzen" bei der Auswahl unter verschiedenen formallogischen Konzepten HABEL (1983), S. 124ff. Vgl. ebenso die analoge Abwägung zwischen größerer Ausdruckskraft und ihrem "Preis" der Unentscheidbarkeit, wenn von ein- zu mehrstelligen prädikatenlogischen Kalkülen übergegangen wird, bei BOOLOS (1980), S. 251. Schließlich präsentiert LUSTI (1990), S. 259f., eine breiter angelegte Übersicht, in der er die Gegenläufigkeit von Ausdruckskraft und Inferenzmächtigkeit ("Ableitungsstärke") bei alternativen Logikarten miteinander konfrontiert. Allerdings äußert er sich nicht über ein optimale Vermittlung dieser entgegengesetzt gerichteten Größen.
Vgl. darüber hinaus die Erläuterungen, die an früherer Stelle zum Konzept eines optimalen Komplexionsgrads von Modellierungen vorgetragen wurden. Die dort vorgetragenen prinzipiellen Schwierigkeiten dieses modelltheoretischen Ansatzes gelten auch hier. Daher handelt es sich - wie oben expliziert - nur um ein *Plausibilitäts*argument mit schwachem Rechtfertigungscharakter. Vgl. dazu auch die Erörterungen hinsichtlich der Rechtfertigungsqualität von

Plausibilitätsargumenten. Vgl. ebenso die exemplarische Problematisierung der nachfolgend thematisierten optimalen Inferenzkomplexion.

20) Dieses Rationalitätskriterium wurde von RESCHER (1987a), S. 73ff., zur Rechtfertigung induktiver Argumentationsverfahren besonders deutlich dargestellt. Es besagt im wesentlichen, daß ein Konzept so lange als "beste" Wahl gerechtfertigt werden kann, wie sich keine besserstellende Alternative präsentieren läßt. Hierdurch wird jedoch keine Feststellung über die theoretische Existenzmöglichkeit - bislang unbekannter - besserer Konzepte getroffen. Vgl. daneben zu weiteren Erwähnungen des faute de mieux-Kriteriums RESCHER (1977a), S. 34 u. 296; RESCHER (1980a), S. 129; RESCHER (1982a), S. 259; COOMANN (1983), S. 154; RESCHER (1987a), S. 73ff., 135 u. 146ff.

21) Eine detaillierte Kennzeichnung praktischer Rationalitätsauffassungen liegt jenseits der Thematik dieser Untersuchungen. Vgl. statt dessen die Anmerkungen und Quellenangaben, die bereits anläßlich der Erörterung des thematischen Bezugsrahmens erfolgten.

22) Mit euphorischen Anklängen stellt QUINE (1975b), S. 156 heraus: "Die klassische Quantorenlogik erfreut sich einer außergewöhnlichen Kombination von Tiefe und Einfachheit, Schönheit und Nützlichkeit. Ihr kühn abgestecktes Gebiet erfüllt sie mit strahlender Klarheit. Abweichungen von ihr wirken dagegen eher ziemlich willkürlich." Dabei entspricht QUINE's Begriff der klassischen Quantorenlogik der hier eingeführten Bezeichnungsweise der Prädikatenlogik 1. Stufe. Vgl. auch BUNGE (1985a), S. 71, der im Hinblick auf alle Komplizierungen gegenüber der klassischen (Prädikaten-)Logik feststellt: "... modal logic and other extended logical systems failed to solve the problems they set out to solve. ... The ... non-standard logics have proved useless ..." (ähnlich S. 74f.).

23) Er wird weiter unten detailliert ausgeführt.

24) Vgl. zur Aussagenlogik, die auch als Junktorenlogik bezeichnet wird, CARNAP (1960a), S. 7ff.; LORENZEN,P. (1962), S. 15ff.; RAMSEY,F. (1965), S. 5ff.; SMULLYAN (1968), S. 4ff.; CHANG,C.L. (1973), S. 6ff.; OPP,K. (1976), S. 98ff. u. 306ff.; ESSER,H. (1977a), S. 30ff.; RICHTER,M.M. (1978), S. 12ff. u. 38ff.; RAUTENBERG (1979), S. 5ff.; BIBEL (1982a), S. 159ff.; STEGMÜLLER (1983), S. 44ff. u. 76ff.; GRAHAM,N. (1983), S. 159ff.; STEGMÜLLER (1984b), S. 49ff.; SOWA (1984), S. 384ff.; DELAHAYE (1987), S. 49ff.; LORENZEN,P. (1987), S. 55ff.; BUCHER (1987), S. 43ff.; GREWENDORF (1987), S. 319ff.; TIETZE,J. (1988), S. 4ff.

25) Die monadischen Prädikatenlogik 1. Stufe zeichnet sich dadurch aus, daß sie nur für einstellige Prädikate definiert ist. Alle einstelligen Prädikate lassen sich als Definitionen von Klassen (Mengen) auffassen. Daher stellt die monadische Prädikatenlogik eine reine Klassenlogik dar. Zugleich handelt es sich um die aristotelische Logikkonzeption. Nähere Ausführungen zur monadisch-aristotelischen Logik finden sich bei REICHENBACH (1977), S. 323f.; BUCHER (1987), S. 133ff.; HABEL (1983), S. 123ff.

26) Vgl. zu prädikatenlogischen Kalkülen 2. und höherer Stufen WHITEHEAD (1925), S. 52ff.; KLEENE (1952), S. 180; CARNAP (1960a), S. 65ff.; HERMES (1961), S. 173ff.; RAMSEY,F. (1965), S. 214ff.; CARNAP (1968), S. 74ff., insbesondere S. 75; MCCARTHY,J. (1968), S. 412f.; RAPHAEL (1976), S. 135; BOOLOS (1980), S. 197ff.; GRABOWSKI,J. (1980c), S. 49; THIEL,C. (1980c), S. 120; FORBUS (1981), S. 326ff.; BIBEL (1982a), S. 254ff.; COHEN,P. (1982), S. 82ff.; OHSUGA (1983), S. 306ff.; TROST (1984), S. 50; DELAHAYE (1987), S. 70; RICHTER,M.M. (1988), S. 24f. u. 29; LEE,R. (1988a), S. 226; LUSTI (1990), S.259.

27) Vgl. zu Darstellungen verschiedener modallogischer Konzepte KRIPKE (1959), S. 1ff.; MCCARTHY,J. (1977), S. 1041ff.; VAN EMDE BOAS (1978), S. 2ff. u. 8ff.; RAUTENBERG (1979), S. 144ff. u. 161ff.; MANNA (1979), S. 385ff.; ABRAHAMSON (1979), S. 24ff.; MITTELSTAEDT (1983), S. 28ff.; WEBBER (1983), S. 44f.; BIBEL (1984), S. 153f. u. 159; BUNGE (1985a), S. 63(ff.); STEGMÜLLER (1986a), S. 152ff.; ZELEWSKI (1986a), S. 197; BUCHER (1987), S. 240ff.; LORENZEN,P. (1987), S. 106ff.; RICHTER,M.M. (1988), S. 26ff.; DORN (1989), S. 79ff.; BIBEL (1989), S. 55ff., insbesondere S. 58.

28) Vgl. WAHLSTER (1977), S. 75; MYLOPOULOS (1983), S. 144; MYLOPOULOS (1984), S. 5; NEWELL (1982a), S. 118 u. 121; ZELEWSKI (1986a), S. 189f.; BEETZ (1986), S. 40f.

29) Fortan wird die Programmiersprache PROLOG als Standardvariante prädikatenlogischer Programmiersprachen behandelt. Es wird nicht weiter unterschieden, ob sich die angeführten Quellen auf die prädikatenlogische Programmierung im allgemeinen oder aber die Sprache PROLOG im besonderen beziehen. Dies gilt allerdings nur in dem Ausmaß, wie die Sprache PROLOG hinsichtlich der jeweils thematisierten Aspekte den vorherrschenden Charakter prädikatenlogischer Programmierung repräsentativ vertritt.

30) Vgl. ZELEWSKI (1986a), S. 189.

31) Hierauf wurde bereits hingewiesen.

32) Allerdings leidet die Implementierung prädikatenlogischer Modellierungen durch die Programmiersprache PROLOG unter praktischen Ausdrucks- und Inferenzmängeln. Diese werden in der hier vorliegenden Untersuchung jeweils dort herausgestellt, wo sie besondere Bedeutung erlangen. Vgl. beispielsweise die Anmerkungen zur beschränkten Ausdruckskraft von HORN-Klauseln sowie zu Inferenzmängeln der Kontrollstrukturen von PROLOG-

Dialekten. Auf derzeit noch bestehende Effizienzprobleme wird in Kürze eingegangen. Weitere Hinweise auf oder ausführliche Erörterungen von Schwächen der Programmiersprache PROLOG finden sich bei KLEINE BÜNING (1988a), S. 59ff.; BEZEM (1988), S. 21f.; BOSSI (1989), S. 96f.; ESTER (1989), S. 48.
Diese Schwierigkeiten sind jedoch nicht so gravierend, daß sie den Einsatz von PROLOG von vornherein vereiteln. So wirkt sich die begrenzte Ausdruckskraft von HORN-Klauseln auf die Formulierung und Implementierung von Netzmodellen nicht aus. Dies wird an anderer Stelle begründet. Die o.a. Inferenzmängel beruhen oftmals darauf, daß in den Kontrollstrukturen der PROLOG-Dialekte Vereinfachungen vorgenommen werden, um Effizienzengpässe zu überwinden. Solche problematischen Vereinfachungen können jedoch unterbleiben, wenn es gelingt, die Auswertungseffizienz von PROLOG-Programmen zu steigern. Darauf wird in Kürze zurückgekommen.

33) Daneben wird die "Interessantheit" der Programmiersprache PROLOG durch zwei weitere Aspekte unterstrichen. Einerseits gewinnt sie bei der Implementierung von Koordinierungskonzepten für Produktionssysteme in jüngster Zeit zunehmende Beachtung. Vgl. dazu die Ausführungen bei FISHER,E. (1985), S. 179ff.; ERSCHLER (1986), S. 1651ff.; HUTCHINSON,S. (1986), S. 904ff.; FREEDMAN (1988b), S. 330ff., insbesondere S. 333 u. 335f.; SAUER (1988a), S. 366ff.; SAUER (1988b), S. 247ff.; ESTER (1989), S. 44ff.
Andererseits wird die Programmiersprache PROLOG des öfteren eingesetzt, um Netzmodelle zu implementieren, die mit der Hilfe von Höheren Netzen gestaltet worden sind. Dies trifft insbesondere auf die Verwendung von Prädikat/Transition-Netzen zu. Wegen der gemeinsamen prädikatenlogischen Basis von PROLOG und Prädikat/Transition-Netzen liegt eine solche Implementierungsweise unmittelbar auf der Hand. Vgl. zu solchen PROLOG-basierten Implementierungen von Prädikat/Transition-Netzen und anderen Höheren Netzen - über das bereits erwähnte Softwarepaket PASIPP hinaus - LIND (1988), S. 1, 6f. u. Anhang C.

34) Vgl. zu objektorientierten PROLOG-Konzepten GALLAIRE (1986), S. 754ff.; HÖSS (1986), S. 5ff.; ROGGENBUCK (1989), S. 182ff.
Vgl. insbesondere zu Ansätzen, das spezielle objektorientierte Strukturierungskonzept der semantischen Rahmen (frames) in PROLOG zu verwirklichen, BINOT (1988), S. 10ff.; THUY (1989), S. 87ff.; SCHNUPP (1989), S. 125ff.

35) Vgl. zu Konzepten für die nebenläufige Implementierungen von logischen Programmiersprachen im allgemeinen und von PROLOG im besonderen KOWALSKI (1974), S. 572f.; KOWALSKI (1979a), S. 428 u. 430; ELCOCK (1983a), S. 115; SHAPIRO,E. (1983), S. 25ff.; MOTO-OKA (1984), S. 479ff.; KUNG (1985), S. 324ff.; YAMAGUCHI,T. (1985), S. 1178ff.; WESTPHAL (1986), S. 227ff., der einen fundierten und kritischen Überblick über mehrere Forschungsansätze bietet; LEVI,G. (1986), S. 414ff. u. 428ff.; PLÜMER (1986), S. 142; KOWALSKI (1987a), S. 132f. u. 139; QADAH (1987), S. 267 u. 270ff.; BARTH,G. (1987), S. 225; POWERS (1988), S. 958 u. 964; MURATA,TA. (1988b), S. 481 u. 490ff.; VAGIN (1988), S. 98ff.; VARNEY (1988), S. 2; GILOI (1989), S. 43ff.
Von besonderem Interesse ist, daß MURATA und ZHANG im o.a. Beitrag MURATA,Ta. (1988b) die nebenläufige Kontrollstruktur von PROLOG-Programmen durch ein Petrinetz vom Typ der Prädikat/Transition-Netze modellieren. Dabei werden zunächst PROLOG-Programme in äquivalente Netzmodelle transformiert (S. 484ff.). Darauf aufbauend läßt sich zeigen, daß bestimmte Schaltfolgen in den Netzmodellen invertierte Ausführungen von kombinierten Unifizierungs- und Resolutionsprozeduren darstellen (S. 488). Schließlich werden diese Netzmodelle herangezogen, um ein Konzept für nebenläufige Unifizierungs- und Resolutionsprozeduren zu entwickeln (S. 490ff.) Dies verdeutlicht, daß sich Netzmodelle nicht nur auf PROLOG-Basis implementieren lassen, sondern auch zur Verbesserung der Implementierungseffizienz beitragen können. Da auf diese Weise das Petrinetz-Konzept mittelbar auf sich selbst angewendet wird, liegt ein "reflexives Modellierungskonzept" vor.
Auf die Möglichkeit, das Petrinetz-Konzept für die Gestaltung nebenläufiger prädikatenlogischer Inferenzprozeduren einzusetzen, wird später noch einmal zurückgekommen.

36) Auf die Erweiterung der Sprache PROLOG um das Sortenkonzept wurde bereits hingewiesen.

37) Dieses Architekturkonzept wurde schon im Kontext sortierter Marken kurz angesprochen.

38) Vgl. zur engen Verknüpfung zwischen relationalen Datenbanksystemen und prädikatenlogischen Wissensrepräsentationen CHANG,C.L. (1976), S. 121ff.; KOWALSKI (1978), S. 77ff.; KOWALSKI (1979a), S. 432; OHSUGA (1983), S. 305; DUTTA,A. (1984), S. 90; MYLOPOULOS (1984), S. 6; LEE,R. (1985), S. 58ff., insbesondere S. 64; LEVI,G. (1986), S. 410f.; ZELEWSKI (1986a), S. 191 u. 596f.; KOWALSKI (1987a), S. 135ff. u. 142f.; vgl. darüber hinaus die Hinweise in der nachfolgenden Fußnote zur speziellen Kombination relationaler Datenbanksysteme mit der Programmiersprache PROLOG.
Darüber hinaus läßt sich auf den Relationenkalkül verweisen, der von CODD (1971b), S. 39ff.; WEDEKIND (1981), S. 232ff.; SCHLAGETER (1983), S. 141ff., beschrieben wird. Bei der Verwendung dieses Kalküls formuliert der Benutzer eines relationalen Datenbanksystems seine Informationseingaben direkt in der Gestalt von Prädikaten. Vgl. zur engen Beziehung zwischen Relationenkalkül und Prädikatenlogik vor allem REITER (1984), S. 192, 195, 199ff. u. 206ff.
Die Abbildung prädikatenlogischer Formelsysteme auf relationale Datenbanksysteme ist keineswegs trivial. Denn die formalsprachlichen Strukturen weichen zunächst voneinander ab. Prädikatenlogische Formeln können jeweils eine komplizierte, hierarchisch ineinander verschachtelte, "tiefe" Struktur besitzen, in die verschiedenartige logische

Operatoren einfließen können. Relationale Ausdrücke beruhen dagegen auf zweidimensionalen, "flachen" Matrixstrukturen, die im Regelfall nur mittels des einen logischen Operators der Konjunktion miteinander verknüpft sind. Daher liegt es zunächst auf der Hand, den Konnex zwischen Prädikatenlogik und relationaler Datenbankarchitektur anhand von atomaren Prädikaten herzustellen, die in Konjugaten zu Formelsystemen zusammengefaßt werden. Daher läßt sich die Extension jedes K-stelligen Prädikats als eine K-stellige Relation auffassen. Alle prädikatserfüllenden K-stelligen Objekttupel, die zum jeweils betrachteten Prädikat gehören, bilden dann - linear aneinandergereiht - eine Liste. Falls diese Liste insgesamt L Elemente enthält, besitzt die gesamte Liste die Gestalt einer KxL-Matrix mit K Spalten und L Zeilen. Diese matrixförmige Listendarstellung der Prädikatsextension stellt eine Relation dar, die sich in einem relationalen Datenbanksystem verwalten läßt. Vgl. zu dieser Abbildung atomarer Prädikate auf Datenbank-Relationen KOWALSKI (1978), S. 79; LEE,R. (1985), S. 64; ZELEWSKI (1986a), S. 597, Fn. 2. Voraussetzung ist allerdings, daß nur endliche Prädikatsextensionen vorkommen. Denn Relationen mit unendlichen Listen relationszugehöriger Objekttupel lassen sich in relationalen Datenbanksystemen nicht benutzen.
Der voranstehend skizzierte Weg läßt sich auch auf die Implementierung Synthetischer Netze anwenden. Dort wird jeder Stelle ein atomares Prädikat zugeordnet. Die aktuelle Prädikatsextension ist die Menge aller Marken, die unter der aktuellen Netzmarkierung die jeweils betrachtete Stelle belegen. Jede Marke ist ein Objekttupel, welches das stellenspezifische atomare Prädikat erfüllt. Die Gesamtheit dieser Marken liefert also alle prädikatserfüllenden Objekttupel, die - in einer Liste angeordnet - eine stellen- und prädikatspezifische Relation in einem relationalen Datenbanksystem formen. Diese relationale Darstellungs- und Verwaltungsweise liegt auch der Datenbankkomponente des hier verwendeten Programmpakets PASIPP zugrunde; vgl. OBERWEIS (1987b), S. 21.
Die prädikatenlogischen und netzbezogenen Konstrukte, die oben vorausgesetzt werden, erfahren an anderer Stelle eine nähere Erläuterung. Die relationalen Matrixkonstrukte werden dagegen hier kurz skizziert, da auf sie nicht mehr zurückgekommen wird. Die Erläuterungen lehnen sich an CODD (1970), S. 379; CODD (1971a), S. 3f., und SCHLAGETER (1983), S. 4 u. 82, an: Jede darzustellende Information wird als ein Objekt (Entity) aufgefaßt. Die Objektstruktur wird durch die Zugehörigkeit des Objekts zu einer Objektklasse (Entity-Typ) determiniert. Jede Objektklasse wird durch eine endliche Menge von Attributen charakterisiert. Die Struktur der Objektklasse ist die lineare Anordnung dieser Attribute. Die Objektklasse stellt eine Relation dar. Sie ist eine Teilmenge des kartesischen Produkts der Definitionsbereiche zulässiger Ausprägungen für alle klassenzugehörigen Attribute. Jedes Objekt ist ein Element dieser klassenspezifischen Relation, das durch eine Kombination von Ausprägungen aller Attribute seiner Objektklasse eindeutig bestimmt ist. Wenn eine Objektklasse durch K Attribute charakterisiert wird mit $K \in \mathcal{N}_+$ und insgesamt L Objekte umfaßt mit $L \in \mathcal{N}_0$, dann läßt sich die gesamte Information über diese relational strukturierte Objektklasse als eine (K+1)x(L+1)-Matrix ausdrücken. Die Matrixspalten codieren die K Attribute zuzüglich einer Spalte für die Namen der jeweils attributtragenden Objekte. Die Matrixzeilen codieren die L relationszugehörigen Objekte einschließlich einer Kopfzeile für die Attributnamen und für den Namen der repräsentierten Objektklasse.

39) Bemerkenswert ist, daß auch WINTER,RO. (1991), S. 12f., 121f. u. 213, für die Fortentwicklung von PPS-Systemen anregt, konventionelle Koordinierungskonzepte und Beiträge der KI-Forschung auf der Basis von relationalen Datenbanksystemen miteinander zu kombinieren. Vgl. auch die vertiefende Rechtfertigung und detaillierte Ausführung dieses Vorschlags bei WINTER,RO. (1991), S. 189ff. bzw. 296ff.

40) Vgl. zur engen Beziehung zwischen relationalen Datenbanksystemen und der Programmiersprache PROLOG JARKE (1984), S. 72f. u. 82; LEE,R. (1988a), S. 223f. u. 234ff.

41) Vgl. OBERWEIS (1987b), S. 21f.

42) Vgl. WAHLSTER (1977), S. 75; BARR,A. (1981), S. 170; BEETZ (1986), S. 41; ZELEWSKI (1986a), S. 190.

43) Sie spielen vor allem im Rahmen der Künstlichen Intelligenz-Forschung bei der Wissensrepräsentation eine Rolle; vgl. KOWALSKI (1979a), S. 425 u. 429; BARR,A. (1981), S. 170; MYLOPOULOS (1983), S. 144; DAISER (1984), S. 36; HARMON (1985), S. 48 (Eleganz); BEETZ (1986), S. 41; ZELEWSKI (1986a), S. 190f.

44) Vgl. dazu die Thematisierung des optimalen Komplexionsgrads von Modellierungen sowie die Anmerkungen zum Dilemma zwischen Ausdruckskraft und Inferenzmächtigkeit.

45) Hier wird noch nicht zwischen syntaktischen oder beweistheoretischen Inferenzen einerseits und semantischen oder modelltheoretischen Schlußfolgerungen andererseits unterschieden. Diese Differenzierung wird in dieser Arbeit nur im speziellen Kontext der prädikatenlogischen Semantik verwendet. Ansonsten wird die sprachliche Vereinfachung zugelassen, Inferenz- und Schlußfolgerungsbegriff synonym zu verwenden.

46) Vgl. zu dieser Begriffsbildung RESCHER (1968a), S. 32 u. 35.

47) Dadurch werden formallogische "trade offs" berücksichtigt: Bestimmte Eigenschaftskombinationen können grundsätzlich nicht verwirklicht werden, weil sich einzelne der kombinierten Aspekte wechselseitig ausschließen.

48) Unter Einschluß der geringfügigen modallogischen Erweiterung, die bereits in einer früheren Anmerkung angesprochen wurde, stellt sie sogar den optimalen Komplexionsgrad dar.

49) Vgl. zur beträchtlichen Ausdruckskraft der Prädikatenlogik 1. Stufe CARNAP (1960a), S. 79 u. 105; LUXEMBURG (1973), S. 42 i.V.m. S. 41f.; RAPHAEL (1976), S. 122 ("Predicate calculus is an adequate language for expressing almost any mathematical concept...") u. 134; NILSSON,N. (1980a), S. 131; PYLYSHYN (1980), S. 441; BIBEL (1982a), S. 58 u. 254; COHEN,P. (1982), S. 121; HABEL (1983), S. 121 u. 127, der u.a. feststellt, die Prädikatenlogik 1. Stufe sei "für die meisten mathematischen - aber auch viele aussermathematische - Anwendungen ... hinreichend mächtig" (S. 121); REITER (1984), S. 221f.; BEETZ (1986), S. 37.
Eine ausführliche und detaillierte Diskussion des prädikatenlogischen Ausdruckspotentials hat HASENJAEGER (1968), S. 242ff., vorgelegt. Dabei kommt er zu dem Schluß, daß das "'Weltbild der klassischen Prädikatenlogik' ... den passenden Rahmen für 'sehr viele' Erfahrungen geliefert hat."
Mitunter werden aber auch Zweifel an der Ausdruckskraft der Prädikatenlogik 1. Stufe geäußert; vgl. OPP,K. (1976), S. 357; COHEN,P. (1982), S. 82f.; ISRAEL (1983), S. 38; OHSUGA (1983), S. 306; WINOGRAD (1984), S. 98f.
Der Verf. wird anhand einer modallogischen Erweiterung selbst eine solches Formulierungsdefizit aufdecken. Auch räumt er ein, daß sich in komplexeren Logiken durchaus Sachverhalte ausdrücken lassen, die sich in der Prädikatenlogik 1. Stufe entweder überhaupt nicht oder nur wesentlich aufwendiger darstellen lassen. Zum ersten Aspekt der mangelnden Ausdrucksmöglichkeit gehören die bereits angesprochenen Modalitäten und Ausdrücke höherer Ordnung, die sich selbst auf Ausdrücke tieferer Ordnung erstrecken. Auf mehrstufige Ausdrücke wird im Kontext der Prädikatenlogik 2. Stufe zurückgekommen; vgl. auch die Beispiele bei OHSUGA (1983), S. 306, und COHEN,P. (1982), S. 83. Zum zweiten Aspekt der aufwendigeren Ausdrucksmöglichkeit zählt die Repräsentation zeitlich variablen Wissens. Der Verf. vertritt daher keineswegs die naive Überzeugung, mit Hilfe der Prädikatenlogik 1. Stufe lasse sich *alles* formulieren. Vielmehr wird die vorsichtigere These verteidigt, daß alle netzbezogenen und betriebswirtschaftlich relevanten Sachverhalte, die in dieser Arbeit von Interesse sind, im Rahmen der - modallogisch ergänzten - Prädikatenlogik 1. Stufe formuliert werden können; vgl. dazu ZELEWSKI (1986a), S. 187f., insbesondere Fn. 4 auf S. 187 und Fn. 1 auf S. 188. Ebenso wird auf das prädikatenlogische Konzept Linearer Beweise verwiesen, das von BIBEL entwickelt wurde. Dort wird erstens gezeigt, daß sich *jedes* Planungsproblem in prädikatenlogischer Weise repräsentieren läßt, wenn es das problemtheoretische Strukturierungsparadigma befolgt; vgl. BIBEL (1989), S. 52. Zweitens kann jedes derart repräsentierte Planungsproblem durch prädikatenlogische Theorembeweise gelöst werden, sofern es sich um ein lösbares Problem handelt; vgl. BIBEL (1989), S. 51ff. Drittens besitzen die prädikatenlogische Problemrepräsentation und -lösung eine wohlfundierte modallogische Semantik; vgl. BIBEL (1989), S. 55ff. Ausführliche Darstellungen des Konzepts Linearer Beweise und seiner Anwendung auf Planungsprobleme bieten BIBEL (1986), S. 115ff.; BIBEL (1989), S. 50ff.; HERTZBERG (1989), S. 34ff. Vgl. darüber hinaus zum problemtheoretischen Ansatz die Ausführungen im Kapitel 2.6 dieser Arbeit.
Abgesehen von den vorgenannten Einschränkungen vermag sich der Verf. den allgemein vorgetragenen Vorbehalten gegenüber dem prädikatenlogischen Ausdruckspotential jedoch nicht anzuschließen. Es würde allerdings den Rahmen dieser Ausarbeitung übersteigen, alle diese Zweifel zu zerstreuen. Daher beschränkt sich der Verf. darauf, die Kritik OPP's an der Prädikatenlogik als pars pro toto zu diskutieren. Dabei hat er die Ausführungen von OPP,K. (1976), ausgewählt, weil dieser Autor zu einem der profundesten, durchaus "freundlich gestimmten" Kenner des prädikatenlogischen Konzepts im sozialwissenschaftlichen Bereich zählt.
OPP betrachtet die Prädikatenlogik als "im Rahmen der Sozialwissenschaften *nur begrenzt* anwendbar" und prädikatenlogische Sprachen als "*relativ ausdrucksarme* ... Sprachen" (S. 357, kursive Hervorhebungen durch den Verf.). Dies führt OPP zunächst auf die Behauptung zurück, mittels der Prädikatenlogik ließen sich "bestimmte Theoreme" (S. 357) nicht ableiten. Seine Unmöglichkeitsthese trifft zwar in bezug auf spezielle prädikatenlogische Konstrukte zu, die etwa GÖDEL bei der Herleitung der Unentscheidbarkeit der Prädikatenlogik 1. Stufe ersonnen hat. Es lohnt aber, diese unentscheidbaren Konstrukte genauer zu studieren; vgl. GÖDEL (1931), S. 188f.; NAGEL,E. (1964), S. 86ff.; MENDELSON (1964), S. 143f. i.V.m. S. 141; KLEENE (1976), S. 765; RHEINWALD (1989), S. 7. Dabei zeigt sich, daß Formeln *dieser Art*, die per constructionem ihre eigene Unbeweisbarkeit aussagen, bei der Modellierung realer Systeme - insbesondere auch in sozialwissenschaftlichen Kontexten - überhaupt keine Rolle spielen. Daher ist OPPs Unmöglichkeitsbehauptung zwar grundsätzlich gültig, aber für den hier interessierenden Anwendungsbereich *irrelevant*.
Darüber hinaus versucht OPP,K. (1976), S. 357 i.V.m. S. 324f., seine Kritik an der prädikatenlogischen Ausdruckskraft durch ein konkretes Beispiel zu untermauern. Dies offenbart jedoch ein fundamentales Mißverständnis. Denn OPP behauptet, aus seinem Beispiel lasse sich eine bestimmte alltägliche Schlußfolgerung nicht ziehen, *weil* "der Prädikatenkalkül ... nicht geeignet" (S. 357) sei. Der Vorwurf, die Prädikatenlogik sei zur Repräsentation von Alltagswissen nicht in der Lage, ist durchaus weit verbreitet; vgl. z.B. COHEN,P. (1982), S. 82f.; ISRAEL (1983), S. 38; WINOGRAD (1984), S. 98f. Eine Analyse von OPPs Beispiel zeigt jedoch sofort, daß die fehlende Schlußfolgerungsmöglichkeit nicht an der angeblich mangelhaften prädikatenlogischen Ausdruckskraft liegt, sondern an einem Informationsdefizit der Beispielkonstruktion. Würde auf S. 324 anstelle der relativen Einkommens-Spar-These eine absolute Einkommens-Spar-These axiomatisch vorgegeben, so wäre die von OPP vermißte Schlußfolgerung möglich. OPP räumt diese Reformulierung zwar selbst ein (S. 325). Doch scheint er sie aus dem prädikatenlogischen Kalkül ausgrenzen zu wollen, weil ihre Darstellung die Gleichheitsrelation erfordert. Der Glaube, die Prädikatenlogik schließe diese Relation aus, verkennt deren Ausdrucksfähigkeit vollkommen. An anderer Stelle wird dargelegt, daß sich mit prädikatenlogischen Mitteln grundsätzlich alle, zumindest alle rekursiven Relationen als (Extensionen von)

prädikatenlogische Formeln darstellen lassen. Andere als rekursive Konstrukte erlangen aber weder in dieser Arbeit noch in anderen sozial- oder realwissenschaftlichen Kontexten irgendeine Bedeutung. Folglich erweist sich OPPs Unmöglichkeitsthese für diesen Argumentationszusammenhang als irrelevant.
Des weiteren glaubt OPP,K. (1976), S. 357, mittels der Prädikatenlogik sei es unmöglich, stochastische Sachverhalte oder Abhängigkeiten zwischen quantitativen Variablen auszudrücken. Beide Unmöglichkeitsbehauptungen verkennen jedoch das tatsächliche prädikatenlogische Ausdruckspotential. Stochastische Sachverhalte lassen sich ohne Schwierigkeiten als zweistellige Prädikate über Ereignissen und deren Eintrittswahrscheinlichkeiten formulieren. Auch die vorgenannten Variablenabhängigkeiten können durch zweistellige Prädikate dargestellt werden, die jeweils über quantitativen Argumenten definiert sind.
Durch - nunmehr dreistellige - Prädikate ist es sogar möglich, unscharfe Eigenschaften von Objekten formal präzise darzustellen. Dies erfolgt im Rahmen der Theorie unscharfer Mengen dadurch, daß entsprechende Prädikate über Tripeln aus Objekten, deren Eigenschaftsausprägungen und unscharfen Zugehörigkeitsfunktionen definiert werden; vgl. RUSPINI (1987a), S. 232f. Dieses unscharfe Formulierungspotential, das für inexakte sozialwissenschaftliche Sachverhaltsbeschreibungen hochwillkommen ist, hat OPP anscheinend noch nicht einmal im Sinne einer Unmöglichkeitsbehauptung in Erwägung gezogen. Zugleich wird hierdurch der Einwand von BEETZ (1986), S. 41, widerlegt, unscharfes (vages) Wissen könne nicht auf prädikatenlogische Weise repräsentiert werden. Auf Erweiterungen der konventionellen Prädikatenlogik durch das Konzept unscharfer Mengen wird an anderer Stelle - einschließlich weiterführender Quellenangaben - zurückgekommen.

50) Die Erweiterungen des prädikatenlogischen Ausdrucksvermögens gegenüber seinem aussagenlogischen Pendant werden z.B. bei KLEENE (1952), S. 142; CHANG,C.L. (1973), S. 26f.; RAPHAEL (1976), S. 120 u. 134; RICHTER,M.M. (1978), S. 58; BIBEL (1982a), S. 58f.; ZELEWSKI (1986a), S. 184, thematisiert. Sie betreffen im wesentlichen die innere Formelstrukturierung durch Terme und Prädikate. Hinzu kommen die Verwendung von Variablen, die Zulässigkeit von Formelquantifizierungen sowie die Erklärung von Funktionen und Relationen. Vgl. allgemein zur Beschränktheit der aussagen- gegenüber der prädikatenlogischen Ausdruckskraft RAPHAEL (1976), S. 120 u. 134 ("predicate calculus makes a great leap in expressive power"); BIBEL (1982a), S. 58; BUCHER (1987), S. 161; LUSTI (1990), S. 253f.

51) Zur logisch präzisen Modellierung von zeitlichen Abhängigkeiten muß die Schaltregel von Synthetischen Netzen um modallogische Operatoren für die Schaltmöglichkeit und für die Schaltnotwendigkeit ergänzt werden. Diese modallogische Ausweitung ist nach Wissen des Verf. in der einschlägigen Literatur zu Prädikat/Transition-Netzen bisher noch nicht erwogen worden. Näheres zur modallogischen Ergänzung der Schaltregelformulierung findet sich an anderer Stelle.
Die modallogische Erweiterung der Prädikatenlogik besitzt zwar *grundsätzlich* erhebliche Bedeutung. Hierauf wird nachfolgend unter dem Aspekt der Vollständigkeit und Automatisierbarkeit noch zurückgekommen. Dennoch bedeutet die modallogische Fortentwicklung des Petrinetz-Konzepts *in dieser Arbeit* nur eine geringfügige Modifizierung des prädikatenlogischen Fundaments. Denn sie zieht im Rahmen des Konzepts Synthetischer Netze keine weiterreichenden formallogischen Konsequenzen nach sich. Statt dessen können die modallogischen Erweiterungen Synthetischer Netze mit Hilfe der prädikatenlogischen Programmiersprache Turbo-PROLOG implementiert werden, ohne deren prädikatenlogisches Ausdrucks- und Inferenzpotential verändern zu müssen. Der tiefere Grund für diese Adaptionsfähigkeit liegt darin, daß die Kontrollstrukturen von PROLOG-Dialekten Steuerungskonzepte für die Ausführung von Inferenzprozeduren auf der Basis des kombinierten Unifizierungs- und Resolutionskonzepts enthalten. Hierdurch weist die prozedurale Semantik der PROLOG-Kontrollstrukturen von vornherein über die rein deklarative Semantik der Prädikatenlogik hinaus. Genau diese prozeduralen Gehaltsüberschüsse werden später ausgenutzt, um die modallogischen Ergänzungen auf "prädikatenlogischer" Basis zu implementieren.
Die Erweiterung der Prädikatenlogik in modallogischer Richtung regt auch HASENJAEGER (1968), S. 243ff., an. Allerdings bezieht er sich auf anders gelagerte Kontexte. Vgl. ebenso MITTELSTAEDT (1983), S. 25ff., der eine formale "'C-Sprache' ... zur angemessenen sprachlichen Erfassung der klassisch-physikalischen Wirklichkeit" (S. 25) präsentiert: Auch dabei handelt es sich um eine prädikatenlogische Sprache 1. Stufe, die um modallogische Ausdrucksmittel angereichert ist (S. 28ff., 35f. u. 41f.). Hinzu kommt BIBEL's Konzept Linearer Beweise, das bereits in einer früheren Anmerkung erwähnt wurde. Es erlaubt die prädikatenlogische Repräsentation und Lösung beliebiger Planungsprobleme. Dabei liegt den problemlösenden Beweisführungen eine ausgefeilte modallogische Semantik zugrunde. Die vorgenannten Beiträge unterstützen die Ansicht, daß für die Modellierung realer Sachverhalte das Ausdruckspotential der Prädikatenlogik 1. Stufe zwar noch nicht ganz ausreicht, sich jedoch durch modallogische Ergänzungen befriedigend abrunden läßt. Gegen diese Einschätzung können aber noch zwei Einwände erhoben werden.
Erstens ist es möglich, die operationale Semantik von Netzen im Sinne einer Prädikatenlogik 2. oder 3. Stufe auszulegen. Aus dieser Perspektive reicht das Ausdruckspotential der Prädikatenlogik 1. Stufe prima facie nicht aus. Doch werden solche höherstufigen Prädikatenlogiken in dieser Arbeit nicht benötigt, um Synthetische Netze zu implementieren. Dies unterstreicht die oben getroffene Feststellung, daß die modallogische Erweiterung des prädikatenlogischen Ausdrucksvermögens konsequenzenlos bleibt, weil sie durch die PROLOG-basierte "prädikatenlogische" Netzimplementierung abgedeckt wird.

Zweitens könnte auf die grundsätzliche Diskrepanz zwischen der deklarativen Semantik der Prädikatenlogik einerseits und der prozeduralen Natur jedes Wissens über zeitliche Abläufe hingewiesen werden. Zeitliche Abläufe spielen aber als einfache Arbeitsgänge oder komplexe Produktionsprozesse in Flexiblen Fertigungssystemen eine herausragende Rolle. Bereits zu Beginn dieser ausführlicheren Anmerkung wurde angedeutet, daß es sich um die Berücksichtigung *zeitlicher* Aspekte von Produktions*prozessen* handelt, die eine modallogische Ergänzung von Schaltregeln erfordern. Die deklarative Semantik der Prädikatenlogik besitzt dagegen einen zeitlosen Charakter: Ihre Formeln gelten unverändert bis in alle Zukunft, wenn sie erst einmal als gültig erkannt worden sind. Hierauf wird unter dem Aspekt der prädikatenlogischen Monotonie zurückgekommen.
Infolge der voranstehend skizzierten Diskrepanz scheint es vor allem unmöglich zu sein, zeitlich variables Wissen über den aktuellen Ausführungszustand von Prozeduren oder über das folgerichtige Fortschreiten bei der Prozedurausführung prädikatenlogisch darzustellen. Auf das allgemeine Problem, beliebiges prozedurales Wissen im Rahmen der Prädikatenlogik zu repräsentieren, weisen z.B. BARR,A. (1981), S. 171; FUCHI (1982), S. 115; MYLOPOULOS (1983), S. 144; MYLOPOULOS (1984), S. 5; WINOGRAD (1984), S. 99; ZELEWSKI (1986a), S. 191f., hin. Vgl. darüber hinaus zu der speziellen Schwierigkeiten, Wissen über zeitlich veränderliche Sachverhalte in prädikatenlogischer Weise abzubilden, FIKES (1972), S. 417f.; STALLMAN (1977), S. 166; MCDERMOTT (1982a), S. 101ff.; BOLOUR (1982), S. 28; LEE,R. (1983), S. 32; WEBBER (1983), S. 44; SCHWIND (1985), S. 240ff.; REINFRANK (1985b), S. 96ff.; HERTZBERG (1986), S. 157f.; ZELEWSKI (1986a), S. 194ff.; MORRIS,P. (1988), S. 384ff.; SHOHAM (1988c), S. 279ff.; DORN (1989), S. 16ff.; KLEINHANS (1989), S. 88; BIBEL (1989), S. 49; OBERWEIS (1990a), S. 22f. (allerdings mit allgemeinem Bezug auf "Logikorientierte Ansätze").
Erst Fortentwicklungen der Prädikatenlogik erlauben es, die Anschauungsform "Zeit" formallogisch präzise zu behandeln. Sie beruhen im allgemeinen auf einer Erweiterung der Prädikatenlogik um modale Operatoren, die es gestatten, Zeitaspekte formallogisch zu erfassen. Der Modalcharakter dieser temporalen Erweiterung resultiert aus der Intensionalität zeitbezogener Formeln. Denn der Wahrheitswert solcher Formeln hängt nicht nur von den Extensionen der Formelbestandteile ab, sondern ebenso vom Zeitpunkt der Formelinterpretation. Die gleiche Formel kann zu einem Zeitpunkt gültig, in einem anderen Zeitpunkt ungültig sein. Sofern der Zeitpunkt der Formelgeltung nicht selbst explizit in die Formeldefinition einbezogen wird, ist daher der Wahrheitswert einer temporalen Formel durch deren Extension noch keineswegs vollständig bestimmt. Die Intensionalität ihrer Formeln ist ein charakteristisches Merkmal modaler Logiken.
Vgl. zur Einbettung von Zeitlogiken in die Modallogik, insbesondere zum voranstehend skizzierten modalen Charakter von temporalen Formeln, VAN EMDE BOAS (1978), S. 1; PNUELI (1977), S. 53; PNUELI (1979), S. 8; ABRAHAMSON (1979), S. 26; BUCHER (1987), S. 275; WEIDEMANN (1988), S. 89ff.; WEDEKIND (1989c), S. 26f. u. 34.; DORN (1989), S. 21f., 49, 79 u. 87ff.; BIBEL (1989), S. 55ff. i.V.m. S. 53 u. 61.
Vgl. zu Zeitlogiken oder temporalen Logiken im allgemeinen z.B. RESCHER (1968c), S. 123ff.; RESCHER (1971); GÜNTHER,A. (1974b), S. 125ff.; PNUELI (1977), S. 46ff.; PNUELI (1979), S. 1ff., insbesondere S. 12ff.; HAREL (1979), S. 12ff.; MANNA (1979), S. 392ff.; RAUTENBERG (1979), S. 159ff. u. 237ff.; GABBAY (1980), S. 163ff.; GRABOWSKI,J. (1980c), S. 58f. (kritisch); MCDERMOTT (1981); BEN-ARI (1981), S. 164ff.; BOLOUR (1982), S. 28ff. (mit einem breiten Überblick über Konzepte für temporale Logiken); MCDERMOTT (1982a), S. 103ff.; QUEILLE (1982b), S. 217 u. 219ff.; LAMPORT,L. (1983), S. 657ff.; FUSAOKA (1983), S. 405ff.; WEBBER (1983), S. 45f.; SAUERWEIN (1984), S. 10ff.; SCHWIND (1985), S. 243ff.; BUNGE (1985a), S. 64 (ablehnend); PNUELI (1986), S. 510ff.; STEGMÜLLER (1986a), S. 191ff.; ARLABOSSE (1986), S. 215ff.; BEETZ (1986), S. 76ff.; ZELEWSKI (1986a), S. 198; HOWELL (1988a), S. 353f.; SAHRAOUI (1988), S. 7ff.; SINACHOPOULOS (1989), S. 9ff., der auf S. 12f., sogar speziell auf temporal-logische Aspekte des Petrinetz-Konzepts eingeht; FAIDT (1989), S. 303ff., insbesondere S. 303f. u. 307; DORN (1989), S. 16.
Sowohl der allgemeine Aspekt des prozeduralen als auch der spezielle Gesichtspunkt des zeitlich veränderlichen Wissens stellen jedoch keine grundsätzlichen Einschränkungen der prädikatenlogischen Ausdruckskraft dar. Der Mangel einer prozeduralen Semantik wird nachträglich durch die PROLOG-Implementierung von prädikatenlogischen Wissensdarstellungen geheilt. Es wurde bereits oben auf die prozedurale Kontrollstruktur der Implementierungssprache PROLOG hingewiesen, welche die prädikatenlogische deklarative durch eine prozedurale Semantik überlagert. Vgl. ZELEWSKI (1986a), S. 192ff.
Auch die Schwierigkeiten bei der Repräsentation zeitlich variabler Sachverhalte lassen sich im Rahmen der Prädikatenlogik 1. Stufe beheben. Anknüpfungspunkt dafür ist vor allem die spezielle Variante einer temporalen Logik, die von ALLEN konzipiert wurde; vgl. ALLEN,J. (1984), S. 127ff.; BEETZ (1986), S. 78f.; DORN (1989), S. 18ff., 26 u. 47ff. Sie kommt ohne modallogische Erweiterungen aus. Statt dessen bewegt sie sich explizit im Rahmen der Prädikatenlogik 1. Stufe (ALLEN,J. (1984), S. 128). Dies widerspricht nicht dem eingangs skizzierten intensional-modallogischen Charakter von zeitbezogenen Formeln. Denn erstens es wurde bereits dort angedeutet, daß sich die intensionsverursachende Anschauungsform "Zeit" explizit in die Definition von zeitbezogenen Formeln als weitere Argumentstelle einbeziehen läßt. Durch einen solchen expliziten "Zeitparameter" wird die intensionale Formeldefinition in eine extensionale Variante transformiert, die sich mit konventionellen prädikatenlogischen Mitteln erfassen läßt. Ansätze dieser Art, Zeitaspekte im Rahmen der Prädikatenlogik durch explizite Zeitparameter zu erfassen, hat es schon des öfteren gegeben; vgl. z.B. BULLERS (1980), S. 355f.; LEE,R. (1983), S. 33; REITER (1984), S. 222ff.; ZELEWSKI (1986a), S. 195f.; DORN (1989), S. 25 u. 46 (distanziert); BIBEL (1989), S. 53 u. 61. Einen verwandten

Ansatz, der auf der komplexeren, aber weiterhin prädikatenlogischen Verwendung einer sortierten Logik beruht, präsentiert DORN (1988a), S. 208ff.; DORN (1988b), S. 205ff.
Die im Bereich der Prädikatenlogik benutzten Zeitparameter entsprechen im Prinzip den "Zeitstempeln", die KNOLMAYER (1989), S. 79 (u. 82ff.), aus informationstechnischer Perspektive für die Erfassung der Dimension "Zeit" in betriebswirtschaftlichen Informationssystemen empfiehlt.
Darüber hinaus wird auf die spätere Integration des Zeitaspekts in Synthetischen Netzen hingewiesen. Dort werden Zeitgrößen analog zu Zeitparametern und -stempeln als Markenattribute dargestellt. Dadurch wird das Ausdruckspotential der Prädikatenlogik wiederum nicht verlassen. Vielmehr wird es im Petrinetz-Konzept ohne konzeptionelle Probleme in die prädikatenlogische Objektmodellierung wie jedes andere quantitative Objektattribut integriert.
Schließlich wird der Ausdrucksreichtum prädikatenlogischer Semantiken genutzt, um die zeitliche Veränderung von Wissen über ein Modellierungsobjekt zu erfassen. Dabei werden unterschiedliche Interpretationen einer syntaktischen Objektmodellierung herangezogen, um die Extensionen von objektbeschreibenden Prädikaten im Zeitablauf zu variieren. Diese zeitvariablen Interpretationen entsprechen jeweils einer Markierung desjenigen Netzes, welches das Modellierungsobjekt abbildet. Daher kann alles zeitlich variable Wissen über das Modellierungsobjekt durch Markierungsfolgen im Erreichbarkeitsgraphen seines Netzmodells erfaßt werden. Für die einzelnen Netzmarkierungen reicht dabei der Formulierungsreichtum der Prädikatenlogik vollkommen aus.
Die voranstehenden Erläuterungen lassen sich wie folgt zusammenfassen. Modal- oder zeitlogische Erweiterungen der Prädikatenlogik sind für das Petrinetz-Konzept zumindest dann nicht notwendig, wenn die Erreichbarkeitsgraphen der Netze vorliegen. Denn die Erreichbarkeitsgraphen enthalten alle modalen und zeitvariablen Wissensbestandteile, die über das Ausdrucksvermögen der Prädikatenlogik (1. Stufe) hinausgehen. Wird dagegen auf das Instrument der Erreichbarkeitsgraphen nicht zurückgegriffen, so muß das prädikatenlogische Formulierungspotential durch Komponenten aus Modal- oder Zeitlogiken erweitert werden.
Dieser Zusammenhang wird mittelbar durch die Ausführungen von GRABOWSKI,J. (1980c), S. 53ff., bekräftigt. Er präsentiert für die Dynamik von Systemen (Vektor-Additions-Systemen), die zunächst durch eine Kombination aus prädikatenlogisch-arithmetischen Formeln und Erreichbarkeitsgraphen beschrieben waren, eine modallogische Reformulierung. Eine nähere Analyse zeigt jedoch, daß diese modallogische Erweiterung der Prädikatenlogik keine dynamischen Systemeigenschaften zu formulieren gestattet, die sich nicht schon in der Kombination aus Prädikatenlogik und Erreichbarkeitsgraphen ausdrücken ließen (S. 56). Die Verwendung der Modallogik gestattet allenfalls eine formal einfachere und übersichtlichere Definition von dynamischen Systemeigenschaften. Dies wird anhand zweier Beispiele verdeutlicht (S. 56), die sich auf die Systembeschränktheit und -reversibilität erstrecken (diese Systemeigenschaften werden in dieser Arbeit an anderer Stelle für Netze definiert).

52) Dies liegt an der fehlenden inneren Strukturierung von atomaren aussagenlogischen Formeln.

53) Betont wird dieser Sachverhalt von COHEN,P. (1982), S. 83: "first-order logic is the most general logic for which the techniques of automatic deduction are at all well developed."
Die Niveauoptimalität der Inferenzkomplexion wird hier nur in einem groben, lokal definierten Sinne verstanden: Ein Komplexionsniveau ist optimal, wenn Übergänge sowohl zu nächsthöheren als auch zu nächstniedrigeren Komplexitätsstufen jeweils zu einer relativen Einbuße der Inferenzqualität führen. Problematisch ist bei den nachfolgenden Plausibilitätsargumenten insbesondere der Umstand, daß nur "wesentliche" Variationen der Komplexität von alternativen Logiken erfolgen, aber keine erschöpfende Untersuchung aller denkmöglichen lokalen Komplexitätsvariationen. Letztes würde den Erkenntnisrahmen dieser Arbeit bei weitem übersteigen. Darüber hinaus gibt die lokale Komplexitätsvariation keine zuverlässige Auskunft über das globale Komplexionsniveau. Schließlich beruhen die nachfolgenden Erläuterungen nicht auf einer reinen Betrachtung der komplexionsabhängigen Inferenzqualität. Denn die Argumente können ihre Plausibilitätskraft erst dann voll entfalten, wenn der Inferenzaspekt mit dem Gesichtspunkt der - gegenläufigen - Ausdrucksmächtigkeit kombiniert wird.

54) Die prädikatenlogische Vollständigkeit bezieht sich hier zunächst immer auf den semantischen Vollständigkeitsbegriff. Er wird in dieser Arbeit stets unterstellt, sofern nicht ausdrücklich auf sein Pendant - den syntaktischen Vollständigkeitsbegriff - Bezug genommen wird. Die unterschiedlichen Bedeutungen der semantischen und der syntaktischen Vollständigkeit werden in den nachfolgenden Anmerkungen im Bedarfsfall näher erläutert. Später werden sie in ein präzises terminologisches Raster eingebettet.

55) Auch der Korrektheitsbegriff wird später präzisiert. Hier reicht zunächst das intuitive Begriffsverständnis aus.

56) Bei der Rückführung semantischer Schlußfolgerungen auf rein syntaktische Inferenzen handelt es sich keineswegs um eine triviale Reduktion, sondern um "eine der wichtigsten Entdeckungen der modernen Kalkültheorie" (STEGMÜLLER (1976b), S. 421). Ähnlich äußert sich HABEL (1983), S. 122. Darüber hinaus gilt diese Reduktionsmöglichkeit - wie nachfolgend dargelegt - keineswegs für alle Logiken.

57) Dies liegt im wesentlichen daran, daß in rein syntaktisch angelegten Kalkülen per definitionem keine Formelbedeutungen beachtet werden. Deswegen lassen sich die syntaktischen Inferenzen "mechanisch" ohne Wissen über die Formelinhalte und damit relativ effizient ausführen. Gerade zu solchen "mechanischen", rein formal definierten

Formeltransformationen bilden die Stärke von informationsverarbeitenden Automaten. Vgl. zur Verknüpfung von rein formal-syntaktischen Konzepten mit ihrer effizienten "mechanischen" Anwendung durch Automaten ("Maschinen") RAPHAEL (1976), S. 110f.; REICHENBACH (1977), S. 336; HABEL (1983), S. 122; STEGMÜLLER (1983), S. 85f.; STEGMÜLLER (1984b), S. 97. Vgl. des weiteren die Anmerkungen zur Vorteilhaftigkeit extensionaler Konstrukte, nur formale Operationen zu erfordern.

58) Der Einfachheit halber wird jede Logik, für die mindestens ein vollständiges und korrektes Beweissystem bekannt ist, selbst als vollständige und korrekte Logik bezeichnet.

59) Bezugspunkt ist hier die Ausführung semantischer Schlußfolgerungen, die sich - zumindest bis heute - nicht unmittelbar automatisieren lassen. Darüber hinaus wird hier unterstellt, daß die Automatisierung von Prozeßausführungen deren Effizienz tendenziell erhöht; vgl. SLAGE (1972), S. 90. Diese Prämisse wird später anläßlich der Erörterung des Automatisierungsziels gerechtfertigt und zugleich problematisiert. Vgl. auch die Effizienzthese bezüglich der PROLOG-basierten Automatisierung von Inferenzprozessen bei SCHARF,A. (1987), S. 24.

60) Vgl. zur Bedeutung der Vollständigkeit und Korrektheit für Inferenzkonzepte auch BEETZ (1986), S. 40.

61) Es handelt sich primär um die Kombination aus Unifizierungs- und Resolutionskonzept, auf die sich die prädikatenlogischen Ausführungen dieser Arbeit konzentrieren. Vgl. zur Vollständigkeit und Korrektheit dieses Inferenzkonzepts ROBINSON,J. (1965), S. 23, 35 i.V.m. S. 30; YATES (1970), S. 257 u. 285; SLAGE (1972), S. 82f. u. 90; CHANG,C.L. (1973), S. 83ff., insbesondere S. 85f.; MELTZER (1975), S. 20; ITZINGER (1976), S. 49, 53 u. 89; RICHTER,M.M. (1978), S. 189 u. 193ff.; CLOCKSIN (1981), S. 220; DIGRICOLI (1981), S. 540; APT (1982), S. 848f.; LEVI,G. (1986), S. 400 u. 402; WEIJLAND (1988), S. 717f. u. 721; MURATA,TA. (1988b), S. 482; PREIß (1989), S. 21. Hinsichtlich eines anderen, weniger subtil gestalteten und praktisch nicht automatisierbaren Inferenzkonzepts wurde die semantische Vollständigkeit der Prädikatenlogik 1. Stufe erstmals von GÖDEL (1930), S. 350ff., nachgewiesen. Vgl. dazu auch MENDELSON (1964), S. 68; STEGMÜLLER (1968), S. 206; KLEENE (1976), S. 761ff.; QUINE (1985), S. 177.
Vgl. darüber hinaus allgemein zur prädikatenlogischen Vollständigkeit und Korrektheit HERMES (1961), S. 30 u. 157; OBERSCHELP (1962), S. 309ff. (für eine sortierte Prädikatenlogik); MENDELSON (1964), S. 59f. u. 62ff.; SMULLYAN (1968), S. 57ff.; MELTZER (1970), S. 189; STEGMÜLLER (1973a), S. 54; RICHTER,M.M. (1978), S. 69 u. 73; BOOLOS (1980), S. 121ff., 125ff. u. 131ff.; POTTHOFF (1981), S. 52 i.V.m. S. 50ff.; BARR,A. (1981), S. 165; CLOCKSIN (1981), S. 220; HABEL (1983), S. 121f.; STEGMÜLLER (1984b), S. 333f.; DUTTA,A. (1984), S. 90; SOWA (1984), S. 164; ZELEWSKI (1986a), S. 186f.; DELAHAYE (1987), S. 78f.; PREIß (1989), S. 21; LUSTI (1990), S. 259.
Es könnte allerdings der Einwand erhoben werden, die prädikatenlogische Vollständigkeit sei unbeachtlich, da bereits die Erweiterung der reinen Prädikatenlogik um arithmetische Operationen der Zahlentheorie zur prädikatenlogischen Unvollständigkeit führe. Dies habe GÖDEL mit seinem Unvollständigkeitsbeweis für formale Systeme aufgezeigt, welche die Ausdruckskraft von arithmetischer Zahlentheorie und Prädikatenlogik in sich vereinen. Dieser Einwand wiegt insofern schwer, als in dieser Arbeit grundsätzlich von einer arithmetisch erweiterten Prädikatenlogik ausgegangen wird. Denn Synthetische Netze und ihre Implementierung in der Programmiersprache PROLOG setzen stets voraus, arithmetische Sachverhalte formulieren zu können. Andernfalls ließen sich z.B. die schaltbedingten Veränderungen von Markenanzahlen und die arithmetischen Formeln der Transitionenbeschriftungen nicht darstellen.
Dennoch trifft der oben skizzierte hypothetische Einwand nicht zu. Denn er übersieht den wesentlichen Unterschied zwischen zwei verschiedenen Vollständigkeitsbegriffen. GÖDELs Unvollständigkeitsbeweis knüpft an dem rein syntaktisch konzipierten Vollständigkeitsbegriff an, dem zufolge ein Beweissystem genau dann (syntaktisch) unvollständig ist, wenn in ihm mindestens eine syntaktisch unentscheidbare Formel existiert; vgl. dazu GÖDEL (1931), S. 191, Fn. 48a. Daher bedeutet syntaktische Vollständigkeit eines Beweissystems, daß in ihm jede Formel entschieden, d.h. durch Inferenzen mit endlichem Ressourceneinsatz entweder bewiesen oder aber widerlegt werden kann; vgl. auch LORENZEN,P. (1962), S. 129ff.; KLEENE (1976), S. 763 u. 768, und STEGMÜLLER (1976b), S. 444.
Die oben thematisierte Vollständigkeit der Prädikatenlogik bezieht sich dagegen auf den semantischen Vollständigkeitsbegriff. Er bezeichnet ein Beweissystem genau dann als (semantisch) vollständig, wenn es in ihm möglich ist, jede im semantischen Sinne allgemeingültige Formel durch syntaktische Inferenzen aus den Axiomen des Beweissystems mit endlichem Ressourceneinsatz abzuleiten. Diese Unterscheidung zwischen syntaktischen und semantischem Vollständigkeitsbegriff wird an anderer Stelle ausführlicher erläutert.
Die subtilen begrifflichen Differenzen führen dazu, daß die arithmetisch erweiterte Prädikatenlogik *semantisch vollständige* Beweissysteme besitzt, obwohl sie *syntaktisch unvollständig* ist. Beispielsweise bleiben Beweissysteme, die das Unifizierungs- und Resolutionskonzept miteinander kombinieren, auch in einer arithmetisch erweiterten Prädikatenlogik semantisch vollständig, obwohl diese Prädikatenlogik syntaktisch unvollständig ist. Dies hat bereits GÖDEL (1931), S. 193, in der Verbindung seines Satzes IX mit seiner Fn. 55) festgestellt. Vgl. dazu auch die Quellen, die in einer früheren Anmerkung zur (semantischen) Vollständigkeit des kombinierten Unifizierungs- und Resolutionskonzepts angeführt wurden. Sie beschränken sich keineswegs auf die reine Prädikatenlogik, aus der alle arithmetischen Sachverhalte ausgeschlossen sind. Vielmehr belegen sie die semantische Vollständigkeit des Unifizierungs- und Resolutionskonzepts für die arithmetisch erweiterte und infolgedessen syntaktisch unvollständige

Prädikatenlogik. Dies gilt allerdings nur unter dem Vorbehalt, daß keine implementierungsbedingten Verkürzungen dieses Inferenzkonzepts erfolgen; vgl. dazu die Erläuterung zu PROLOG-typischen Implementierungsdefiziten.
Der oben vorausgesetzte Nachweis der syntaktischen Unvollständigkeit der arithmetisch erweiterten Prädikatenlogik 1. Stufe findet sich im Original bei GÖDEL (1931), S. 190ff. Vgl. dazu auch POST,E. (1936), S. 103 u. 105; GENTZEN (1938), S. 38; HASENJAEGER (1952), S. 81ff.; LORENZEN,P. (1962), S. 55; NAGEL,E. (1964), S. 85 u. 92; STEGMÜLLER (1968), S. 207; STEGMÜLLER (1973a), S. 24f.; KLEENE (1976), S. 763ff.; BOOLOS (1980), S. 170ff., insbesondere S. 179f.; MACHOVER (1983), S. 7 u. 11; QUINE (1985), S. 177f.; BUNGE (1985a), S. 49; RHEINWALD (1989), S. 5ff.; SMULLYAN (1989), S. 115, 176, 182, 184ff. u. 224.

62) Vgl. zum Vorzug der Prädikatenlogik 1. Stufe, eine umfassende und - relativ zu anderen Logikvarianten höherer Komplexität - einfache Beweisautomatisierung zuzulassen, COHEN,P. (1982), S. 83; DELAHAYE (1987), S. 118; PREIß (1989), S. 20.

63) Das Effizienzurteil bezieht sich auf alternative Implementierungsoptionen für prädikatenlogische Inferenzkonzepte. Vgl. zur - derart relativierten - Effizienz der PROLOG-Implementierungen von prädikatenlogischen Beweissystemen LEVI,G. (1986), S. 399 u. 404; SCHARF,A. (1987), S. 24.
Die Implementierung vollautomatischer prädikatenlogischer Beweissysteme erfolgt bei der Programmiersprache PROLOG mit Hilfe des Unifizierungs- und Resolutionskonzepts. Dessen Vollständigkeit und Korrektheit wurde schon früher dargelegt. Allerdings ist darauf hinzuweisen, daß im Interesse der Effizienzsteigerung die meisten der derzeit angebotenen PROLOG-Dialekte das Unifizierungs- und Resolutionskonzept so implementieren, daß es seine theoretische Vollständigkeit verliert. Auf diese Implementierungsdefizite wird noch näher eingegangen.

64) Vgl. FREEDMAN (1988b), S. 332.

65) Vgl. zur Kritik, PROLOG-Programme und die zugrundeliegenden prädikatenlogischen Inferenzprozeduren - also letztlich das kombinierte Unifizierungs- und Resolutionskonzepte - litten an mangelhafter Ausführungseffizienz, YATES (1970), S. 257; NEWELL (1982a), S. 91; SCHWIND (1982), S. 245; COHEN,P. (1982), S. 121; HSIANG (1983a), S. 331 u. 335ff.; LENAT (1984), S. 181; VARSEK (1986), S. 212; ZELEWSKI (1986a), S. 932f.; BARTH,G. (1987), S. 225; QADAH (1987), S. 267; MURATA,TA. (1988b), S. 490; VAGIN (1988), S. 98; GILOI (1989), S. 42.

66) Skeptisch hinsichtlich der potentiellen Effizienzverbesserungen durch nebenläufige Ansätze äußert sich hingegen YASUURA (1984), S. 235ff., insbesondere S. 235 u. 240. Allerdings führt er eine worst case-Analyse durch, die nur Effizienzschranken für die schlechtest möglichen Fällen bestimmt. Hinsichtlich der durchschnittlichen Verarbeitungseffizienz können solche Analysen keine zuverlässige Aussage treffen. Dies räumt auch YASUURA (1984), S. 242, ein. Der Verf. hat an anderer Stelle die Validitätsmängel solcher worst case-Analysen näher ausgeführt; vgl. ZELEWSKI (1989a), S. 94ff. Weitere kritische Anmerkungen zur Effizienzverbesserung durch Nebenläufigkeit finden sich bei WESTPHAL (1986), S. 228 ("... erweisen sich die meisten dieser Modelle entweder als unvollständig oder als nicht effizient implementierbar. ... Bei weitem nicht alle von diesen scheinen wirklich ernst gemeint zu sein.") u. S. 236f. WESTPHAL's Einwendungen besitzen größere praktische Relevanz, weil sie sich auf das Durchschnittsverhalten nebenläufiger Prolog-Implementierungen beziehen. Allerdings werden keine stringenten average case-Analysen vorgelegt, sondern nur qualitative Abschätzungen auf der Basis von Plausibilitätsüberlegungen vorgenommen.

67) Zu den hiermit verknüpften Hoffnungen auf eine verbesserte Effizienz der Programmausführung vgl. VARSEK (1986), S. 214 u. 219; SMOLKA (1988a), S. 183 u. 186. Vgl. ebenso - allerdings ohne Bezugnahme auf PROLOG - COHN (1987), S. 184; CORRENZ (1987), S. 235.

68) Mit unterschiedlichen Strategien, möglichst effiziente Algorithmen für die Implementierung des Unifizierungs- und Resolutionskonzepts zu entwickeln, setzen sich beispielsweise auseinander: SICKEL (1977), S. 74ff., insbesondere S. 79ff.; PLÜMER (1986), S. 138ff.; VARSEK (1986), S. 213ff.; DELAHAYE (1987), S. 125ff.; GOOS (1987a), S. 89ff., insbesondere S. 98ff.; KOWALSKI (1987a), S. 136f.; SCHÖNFELD,W. (1988), S. 51ff.
Ein herausragendes Exempel für algorithmische Verbesserungen erstreckt sich auf das Unifizierungskonzept. Seine erste algorithmische Formulierung durch ROBINSON besaß noch exponentielle Komplexität. Von CORBIN und BIDOIT wurde im Jahr 1983 ein quadratischer Unifizierungsalgorithmus präsentiert. Vor kurzem - im Jahr 1988 - haben sogar RUZICKA und PRIVARA einen Algorithmus von nahezu linearer Komplexität vorgestellt. Vgl. zu diesen Komplexitätsresultaten für das Unifizierungskonzept RUZICKA (1988), S. 501f. u. 504ff.; vgl. zu den hier vorausgesetzten komplexitätstheoretischen Effizienzbegriffen ZELEWSKI (1989c), S. 42ff. u. 51ff.
Neuerdings wird versucht, die Effizienz von PROLOG-Programmen auch dadurch zu erhöhen, daß das kombinierte Unifizierungs- und Resolutionskonzept mit der Technik der Restriktionspropagierung verknüpft wird; vgl. SMOLKA (1988a), S. 202; COMMISSION OF THE EUROPEAN COMMUNITIES (1989), S. I-13f. (Restriktionspropagierungen wurden bereits im Zusammenhang mit Konzepten für Anpassungsplanungen angesprochen) Insbesondere im Rahmen der fortentwickelten Sprachversion PROLOG II wird versucht, durch Einbeziehen von Restriktionen in Inferenzprozesse deren Effizienz zu erhöhen; vgl. zu dieser "logischen Programmierung mit Restriktionen" in PROLOG II COLMERAUER (1983), S. 292ff. (die dort behandelten Gleichungen und Ungleichungen stellen die hier angesprochenen Restriktionen dar), insbesondere S. 296; PIQUE (1988), S. 4 u. 7ff., insbesondere S. 12ff. u. 16; SMOLKA (1988a), S. 202. Daneben wurde auch eine spezielle Sprachvariante mit dem Namen CHIP (für: Constraints

Handling in Prolog) entwickelt. Sie ist darauf spezialisiert, bereits bei der Wissensrepräsentation durch PROLOG-Programme Restriktionen in besonders übersichtlicher und kompakter Weise zu berücksichtigen. Auch auf diese Weise läßt sich die Effizienz der Programmausführung durch Inferenzprozesse steigern. Vgl. zur PROLOG-Variante CHIP und ihrer Restriktionsorientierung O.V. (1990a), S. 14f. Den vorgenannten Ansätzen gemeinsam ist das Konzept, mit der Hilfe von Restriktionen den logischen Suchraum für das Unifizierungs- und Resolutionskonzept drastisch zu reduzieren. Die gleiche Idee, durch Restriktionen logische Suchräume einzuschränken, wird von HERTZBERG (1989), S. 171ff., ausführlich - allerdings ohne Bezugnahme auf die Sprache PROLOG - erörtert.

69) Vgl. vor allem VARNEY (1988), S. 1f. u. 5ff.; daneben auch LEVI,G. (1986), S. 428. ZELL,A. (1989), S. 96ff., führt zwar Zeitschranken für die Ausführung von PROLOG-Programmen ein, bezieht diese aber nicht auf reale, sondern nur auf logische Zeiteinheiten (Inferenzschritte). Die Bemühungen um die Realzeitfähigkeit von PROLOG erweist sich insbesondere aus der hier verfolgten Perspektive, reale Produktionsprozesse in Flexiblen Fertigungssystemen zu koordinieren, als bemerkenswert.

70) Vgl. zur Unvollständigkeit der Prädikatenlogik 2. Stufe HASENJAEGER (1952), S. 85; HERMES (1961), S. 31 u. 176; QUINE (1975b), S. 154f.; RICHTER,M.M. (1988), S. 24f.
Darüber hinaus existiert für die Prädikatenlogik 2. Stufe kein Algorithmus, mit dessen Hilfe sich alle Paare aus je zwei beliebigen Termen unifizieren ließen; vgl. LUSTI (1990), S. 259. Aufgrund ihrer geringen Inferenzmächtigkeit erfahren Prädikatenlogiken 2. und höherer Stufen bei der Konstruktion wissensbasierter Automatischer Informationsverarbeitungssysteme keine Beachtung; vgl. LUSTI (1990), S. 260.

71) Vgl. zur Unvollständigkeit (mancher Varianten) der Modallogik RAUTENBERG (1979), S. 180, 186ff., 257 u. 259; DORN (1989), S. 80 (unter dem Aspekt mangelnder Wahrheitsfunktionalität) u. 82.
Dagegen bleiben einzelne ausgezeichnete Varianten der Modallogik vollständig (und korrekt); vgl. KRIPKE (1959), S. 11f. (für die Modallogik "S5").

72) Auch andere Komplizierungen der Prädikatenlogik leiden unter - zumindest partiellem - Vollständigkeitsverlust; vgl. RICHTER,M.M. (1988), S. 24.

73) Vgl. RAPHAEL (1976), S. 135.

74) Vgl. zur aussagenlogischen Korrektheit und Vollständigkeit GENTZEN (1936), S. 516; SMULLYAN (1968), S. 25ff.; RICHTER,M.M. (1978), S. 42ff.; RAUTENBERG (1979), S. 58f., 71 u. 85; DELAHAYE (1987), S. 55f. u. 57ff.; GREWENDORF (1987), S. 344 i.V.m. S. 343; LUSTI (1990), S. 259. Vgl. des weiteren die Quellen, die bereits zur Entscheidbarkeit der Aussagenlogik angeführt wurden. Denn bei aussagenlogischen Beweissystemen fallen Vollständigkeit und Korrektheit mit ihrer Entscheidbarkeit zusammen.

75) Vgl. zur Automatisierbarkeit die Anmerkungen zur Entscheidbarkeit sowie zur Vollständigkeit und Korrektheit. Denn im Fall der Aussagenlogik lassen sich die vollständigen und korrekten Algorithmen zur Entscheidung über Formelgültigkeiten ohne Schwierigkeiten auf Automatischen Informationsverarbeitungssystemen implementieren.

76) Dies gilt zumindest dann, wenn alle logisch basierten Modellierungskonzepte dem Primat der Ausdrucksmächtigkeit gegenüber der Lösungseffizienz unterworfen werden. Diese Basisentscheidung wurde bereits früher getroffen und im Rahmen dieser Arbeit gerechtfertigt.

77) Eine der präzisesten und zugleich kompaktesten Formulierungen des Gültigkeitsproblems hat RAMSEY,F. (1965), S. 92, vorgelegt: "The *Entscheidungsproblem* is to find a procedure for determing whether any given formula is valid, or, alternatively, whether any given formula is consistent ..." (kursive Hervorhebung im Original). Vgl. zu ähnlichen Formulierungen des Gültigkeitsproblems ZELEWSKI (1989a), S. 14ff., und die dort angeführten Quellen.

78) Die Wohlgeformtheit einer Formel hängt davon ab, daß die Formel aus einem formalsprachlichen Kalkül regulär - d.h. unter Anwendung der Kalkülregeln - gebildet werden kann.

79) Im angloamerikanischen Sprachgebrauch wird eine allgemeingültige Formel zumeist als "valid" bezeichnet; vgl. z.B. BOOLOS (1980), S. 104. Damit ist aber nicht nur die "Gültigkeit" einer Formel unter einer Interpretation gemeint. Vielmehr ist eine "valide" Formel eine solche, die unter *allen* zulässigen Interpretationen gültig - also: allgemeingültig - ist; vgl. ebenso BOOLOS (1980), S. 104, sowie die Definitionen zur prädikatenlogischen Semantik.

80) Als komplementäre Formulierung des Gültigkeitsproblems kommt auch in Betracht, die Inkonsistenz jeder wohlgeformten Formel feststellen zu können. Die komplementäre Beziehung zwischen Entscheidungen über Allgemeingültigkeit und Inkonsistenz folgt daraus, daß eine Formel genau dann (nicht) allgemeingültig ist, wenn ihr Negat (konsistent) inkonsistent ist. Daher kann die behauptete Allgemeingültigkeit einer Formel ebenso dadurch untersucht werden, daß die Inkonsistenz des Negats dieser Formel behauptet wird. Die Entscheidungsalgorithmen werden zwar für beide Behauptungsalternativen unterschiedlich gestaltet, müssen aber - sofern sie korrekt sind - zu identischen Ergebnissen führen (falls sie überhaupt zu Ergebnissen führen). Die komplementäre Beziehung zwischen den Untersuchungen der Formelallgemeingültigkeit oder der Formelinkonsistenz wird besonders deutlich von

QUINE (1975b), S. 152ff., und BOOLOS (1980), S. 106 u. 121, behandelt. Diese Beziehung scheint auch RAMSEY,F. (1965), S. 92, im Auge zu haben, der allerdings von Konsistenz anstelle von Inkonsistenz spricht. Die oben verwendeten prädikatenlogischen Formeleigenschaften werden an anderer Stelle präzisiert.
Die Betrachtung der oben angeführten Formulierungsalternative des Gültigkeitsproblems ist insofern interessant, als vom Entscheidungsalgorithmus der Programmiersprache PROLOG - dem kombinierten Unifizierungs- und Resolutionskonzept - stets die komplementäre Problemformulierung der Inkonsistenzbehauptung benutzt wird. Vgl. dazu die Charakterisierung des Resolutionskonzepts als eine indirekte Beweistechnik sowie die Erläuterungen zur Bearbeitung von Entscheidungsproblemen mit Hilfe des kombinierten Unifizierungs- und Resolutionskonzepts.
Die komplementären Inferenzkonzepte stellen aber nur Entscheidungsverfahren dar, die in der Lage sind, die Allgemeingültigkeit oder die Inkonsistenz einer Formel zu *nachzuweisen.* Sie bedeuten nicht notwendig Entscheidungsverfahren, die jene Formeleigenschaften auch zurückweisen könnten. Darauf wird unter dem Aspekt der prädikatenlogischen Semi-Entscheidbarkeit durch die Differenzierung zwischen positiven und negativen Entscheidungsverfahren später ausführlich zurückgekommen.

81) Vgl. z.B. das RAMSEY-Zitat in der voranstehenden Anmerkungen; vgl. ebenso CHURCH (1936b), S. 40; BRAUER (1968), S. 4; BRAUER (1990a), S. 66. Der Verf. zieht dagegen die Bezeichnung "Gültigkeitsproblem" vor, um eine Verwechslung mit anderen Entscheidungsproblemen zu vermeiden.

82) Die prinzipielle Unmöglichkeit, für *jede* prädikatenlogische Formel durch *einen universellen* Algorithmus in endlicher Zeit entscheiden zu können, ob sie allgemeingültig ist oder nicht, wurde erstmals unmittelbar von CHURCH (1936b), S. 40f.; CHURCH (1936c), S. 101f.; TURING (1937a), S. 259ff.; TURING (1937b), S. 544ff., bewiesen. Folglich läßt sich das Gültigkeitsproblem der Prädikatenlogik 1. Stufe im allgemeinen nicht lösen. Es stellt ein unentscheidbares Problem dar. Daher ist auch die Prädikatenlogik selbst unentscheidbar.
Von GÖDEL (1931), S. 173ff., insbesondere 187ff. u. 193ff., wurde bereits früher ein eng verwandtes Unmöglichkeitsresultat für eine breitere Klasse von - nicht nur rein prädikatenlogischen - Inferenzkonzepten geliefert. Dabei setzte er Beweissysteme voraus, die so ausdrucksstark sind, daß sie das Formulierungspotential von Prädikatenlogik und arithmetischer Zahlentheorie umfassen. Jedes solche Beweissystem wurde von GÖDEL als syntaktisch unvollständig nachgewiesen. Darauf wurde schon oben eingegangen. Daher läßt sich in allen Beweissystemen für die arithmetisch erweiterte Prädikatenlogik *mindestens eine* Formel konstruieren, für die gilt: Es ist innerhalb jedes Beweissystems unmöglich, die Formel dadurch zu beweisen, daß sie aus den Axiomen des Beweissystems abgeleitet wird, oder dadurch zu widerlegen, daß ihr Negat aus denselben Axiomen abgeleitet wird. Eine solche weder beweisbare noch widerlegbare Formel heißt unentscheidbar. Die Unentscheidbarkeit mindestens einer solchen Formel bedeutet zugleich die Unlösbarkeit des prädikatenlogischen Gültigkeitsproblems. Denn dann ist auch unmöglich, für *jede* Formel zu entscheiden, ob sie allgemeingültig (inkonsistent) ist oder nicht. Dies folgt zumindest unter recht schwachen Zusatzannahmen, auf die anschließend näher eingegangen wird.
Um Mißverständnissen vorzubeugen, wird ausdrücklich darauf hingewiesen, daß die Unentscheidbarkeit der Prädikatenlogik direkt nur aus dem Beweis von CHURCH und TURING folgt. GÖDEL hat dagegen zunächst "nur" die syntaktische Unvollständigkeit eines prädikatenlogisch fundierten Beweissystems, nicht aber die Unentscheidbarkeit der Prädikatenlogik selbst nachgewiesen. Darüber hat sich haben sich CHURCH und TURING auf einen prädikatenlogischen Entscheidungsalgorithmus bezogen, während sich GÖDEL mit einer einzelnen unentscheidbaren Formel auseinandersetzte. Vgl. zu solchen Differenzierungen zwischen der Erkenntnissen von CHURCH/TURING einerseits und GÖDEL andererseits ROSSER (1936), S. 87; TURING (1937a), S. 259; MYHILL (1952), S. 181; ZELEWSKI (1989a), S. 15. Trotz dieser Abweichungen sind aber die Unmöglichkeitsbeweise von CHURCH/TURING und GÖDEL eng miteinander verwandt. Denn GÖDELs Unvollständigkeitserkenntnis führt - wie bereits oben angedeutet wurde - zur Unentscheidbarkeit der Prädikatenlogik 1. Stufe, falls vier ergänzende Annahmen akzeptiert werden.
Erstens gilt GÖDELs Unvollständigkeitsbeweis nur unter der Prämisse, daß die von ihm untersuchten Beweissysteme konsistent sind. Die innere Widerspruchsfreiheit der Prädikatenlogik war aber zu dem Zeitpunkt, als GÖDEL seinen Beweis vorlegte, noch ungeklärt. Längere Zeit bestanden erhebliche Schwierigkeiten, ihre mutmaßliche Konsistenz nachzuweisen; vgl. FRAENKEL (1930), S. 296; VON NEUMANN (1931), S. 119ff.; GENTZEN (1936), S. 498; LORENZEN,P. (1962), S. 65; KLEENE (1976), S. 767; REICHENBACH (1977), S. 334; Erst GENTZEN gelang es, die Konsistenz der Prädikatenlogik schlüssig aufzuzeigen. Vgl. zu diesem und analogen Widerspruchsfreiheitsbeweisen GENTZEN (1936), S. 533ff.; GENTZEN (1938), S. 26ff.; LORENZEN,P. (1962), S. 61f., 65ff. u. 74ff.; MENDELSON (1964), S. 59f. u. 258ff.; NAGEL,E. (1964), S. 95; STEGMÜLLER (1973a), S. 27; KLEENE (1976), S. 767f. Daher ist die Konsistenzprämisse VON GÖDELs Unvollständigkeitsbeweis tatsächlich erfüllt.
Zweitens bezieht sich GÖDEL auf formale Systeme, die neben der Prädikatenlogik auch die Arithmetik umfassen. Daher kann die o.a. Unlösbarkeit des Gültigkeitsproblems im Anschluß an GÖDEL nur für eine erweiterte Prädikatenlogik gelten, die auch die arithmetische Ausdruckskraft umfaßt. Auch diese Voraussetzung bereitet keine Schwierigkeiten. Denn die arithmetische Erweiterung der Prädikatenlogik ist für die später zu entwickelnden Netzmodelle nicht nur akzeptabel, sondern sogar erforderlich.
Drittens hat sich GÖDEL auf die syntaktische Unvollständigkeit von Inferenzen in einem Beweissystem bezogen. Das Gültigkeitsproblem der Prädikatenlogik erstreckt sich dagegen auf die semantischen Kategorien der Allgemeingültigkeit (oder Inkonsistenz) von Formeln, ohne syntaktische Inferenzen in einem Beweissystem direkt zu berüh-

ren. Auf die Unterschiedlichkeit zwischen syntaktischen Inferenzen und semantischen Schlußfolgerungen wird später ausführlicher eingegangen. Hier reicht der Hinweis, daß es unmöglich ist, mittels eines syntaktisch definierten Entscheidungsalgorithmus die semantisch definierte Allgemeingültigkeit (Inkonsistenz) *jeder* prädikatenlogisch-arithmetischen Formel festzustellen, falls die arithmetisch erweiterte Prädikatenlogik syntaktisch unvollständig ist. Denn die Kenntnis auch nur einer syntaktisch unentscheidbaren Formel widerspricht der Existenzmöglichkeit eines solchen syntaktischen Entscheidungsalgorithmus. Da GÖDEL die syntaktische Unvollständigkeit der arithmetisch erweiterten Prädikatenlogik bewiesen hat, kann es einen syntaktisch definierten Entscheidungsalgorithmus für ihr Gültigkeitsproblem nicht geben.
Viertens wird vorausgesetzt, daß Algorithmen immer vollständig formalisiert und somit syntaktisch definiert sein müssen. Dies folgt aus der früheren Basisentscheidung zugunsten formaler, rein extensional gestalteter Konzepte. Der Verf. kennt auch keine informal definierten Entscheidungsalgorithmen, die das prädikatenlogische Gültigkeitsproblem - im Gegensatz zu ihren formalen Pendants - lösen könnten.
Unter allen vier voranstehenden Voraussetzungen resultiert aus der GÖDEL'schen Unvollständigkeitserkenntnis die oben festgestellte Unlösbarkeit des Gültigkeitsproblems für die arithmetisch erweiterte Prädikatenlogik 1. Stufe. Dieses Ergebnis läßt sich auch durch folgende Überlegung veranschaulichen: Aufgrund der Unvollständigkeitserkenntnis von GÖDEL läßt sich mindestens eine arithmetisch-prädikatenlogische Formel - die "GÖDEL-Formel" - konkret konstruieren, die innerhalb eines konsistenten Beweissystems weder bewiesen noch widerlegt werden kann. In jedem korrekten Beweissystem würden der syntaktische Beweis der Formel oder deren Widerlegung bedeuten, daß diese Formel auf der semantischen Ebene allgemeingültig bzw. nicht allgemeingültig ist. (Vgl. dazu die Korrespondenzen zwischen syntaktischen und semantischen Schlußweisen.) Da GÖDEL zeigte, daß sowohl der Beweis als auch die Widerlegung der GÖDEL-Formel in *allen* arithmetisch-prädikatenlogischen Beweissystemen unmöglich sind, läßt sich die Allgemeingültigkeit dieser Formel mit rein syntaktischen Mitteln in *keinem* arithmetisch-prädikatenlogischen Kalkül entscheiden. Folglich kann auch kein syntaktisch definierter Entscheidungsalgorithmus existieren, der das Gültigkeitsproblem für einen Kalkül zu lösen vermag, in dem sich die spezielle GÖDEL-Formel ausdrücken läßt.
CHURCH und TURING haben dagegen keine einzelne ausgezeichnete Formel konstruiert, die auf syntaktische Weise weder durch einen Formelbeweis als allgemeingültig noch durch eine Formelwiderlegung als nicht-allgemeingültig entschieden werden kann. Statt dessen haben sie nachgewiesen, daß es innerhalb eines prädikatenlogischen Beweissystems keinen (syntaktisch definierten) Entscheidungsalgorithmus geben kann, mit dessen Hilfe sich die Allgemeingültigkeit *aller* Formeln überprüfen ließe. Dabei setzen sie voraus, daß jeder denkmögliche, rein syntaktisch definierte Entscheidungsalgorithmus durch einen TURING-Automaten dargestellt werden kann. Für jeden solchen Entscheidungsalgorithmus existiert daher mindestens eine "pathologische" Formel, deren Allgemeingültigkeit sich nicht entscheiden läßt. Dies äußert sich darin, daß der algorithmusrepräsentierende TURING-Automat diese Formel unendlich lange untersucht, ohne mit einem Entscheidungsergebnis terminieren zu können. Von CHURCH/TURING wird aber nicht festgestellt, welche Gestalt diese mindestens eine pathologische Formel besitzen muß (insofern ist ihr Beweis nicht-konstruktiver Art). Ferner läßt es ihr Unentscheidbarkeitsresultat offen, ob alle pathologischen Formeln algorithmusspezifisch ausfallen oder ob sich mindestens eine Formel für alle denkmöglichen Entscheidungsalgorithmen als pathologisch erweist. Daß tatsächlich letztes der Fall ist, folgt erst aus GÖDELs Konstruktion einer Formel, die in keinem arithmetisch-prädikatenlogischen Beweissystem bewiesen oder widerlegt werden kann: Ihre Allgemeingültigkeit läßt sich innerhalb der arithmetisch erweiterten Prädikatenlogik durch keinen syntaktischen Entscheidungsalgorithmus mit endlichem Ressourceneinsatz entscheiden. Auf TURING-Automaten und die Unentscheidbarkeit ihres Halteproblems wird im Zusammenhang mit der operationalen Dimension der Prädikatenlogik näher zurückgekommen.
Vgl. des weiteren zur Unentscheidbarkeit der Prädikatenlogik 1. Stufe, zur daraus resultierenden Unentscheidbarkeit des prädikatenlogischen Gültigkeitsproblems nach CHURCH und TURING sowie zur Unvollständigkeit von prädikatenlogisch-zahlentheoretischen Formelsystemen nach GÖDEL: ROSSER (1936), S. 87 u. 89ff., der die Erkenntnisse von CHURCH/TURING und GÖDEL verallgemeinert; GENTZEN (1938), S. 9f.; MYHILL (1952), S. 180ff.; ACKERMANN,W. (1957), S. 4 u. 6f.; Davis,M. (1958), S. 127ff.; LUCAS (1961), S. 112ff.; LORENZEN,P. (1962), S. 96 u. 98ff., insbesondere S. 124f.; DUMMETT (1963), S. 124ff.; NAGEL,E. (1964), S. 69ff., insbesondere S. 82ff.; MENDELSON (1964), S. 14ff.; STEGMÜLLER (1968), S. 206ff.; KÖRNER,S. (1968), S. 109ff.; STEGMÜLLER (1973a), S. 5ff., insbesondere S. 20ff. u. 27 (bezüglich GÖDEL) und S. 44f. u. 55ff. (hinsichtlich CHURCH); KLEENE (1976), S. 763ff.; ITZINGER (1976), S. 32f.; RAPHAEL (1976), S. 122f.; STEGMÜLLER (1976b), S. 444f.; ESSER,H. (1977a), S. 45; HERMES (1978), S. 165ff.; RICHTER,M.M. (1978), S. 168; BOOLOS (1980), S. 112ff. u. 175ff.; STEGMÜLLER (1983), S. 90; HABEL (1983), S. 123; STEGMÜLLER (1984b), S. 366ff.; HOPCROFT (1984), S. 34 u. 44; QUINE (1985), S. 178ff.; BUNGE (1985a), S. 49f.; ZELEWSKI (1986a), S. 934ff.; BUCHER (1987), S. 197; DELAHAYE (1987), S. 85; SMULLYAN (1989), S. 113ff., insbesondere S. 176 u. 186f.; ZELEWSKI (1989a), S. 14ff.; BRAUER (1990a), S. 66; LUSTI (1990), S. 259.

83) Vgl. zur Entscheidbarkeit der monadischen Prädikatenlogik 1. Stufe BEHMANN (1922), S. 187 u. 201ff.; ACKERMANN,W. (1957), S. 7; RAMSEY,F. (1965), S. 92; BOOLOS (1980), S. 250ff.; STEGMÜLLER (1983), S. 90; HABEL (1983), S. 124; DELAHAYE (1987), S. 85; BUCHER (1987), S. 197 i.V.m. S. 217ff.

Einstellige Prädikate stellen die Grenze zwischen prädikatenlogischer Entscheidbarkeit und Unentscheidbarkeit dar. Denn die Prädikatenlogik wird unentscheidbar, sobald zweistellige Prädikate zugelassen werden. Dafür reichen bereits einfache dyadische Prädikate mit symmetrisch vertauschten Argumentkomponenten aus. Vgl. zur Unentscheidbarkeit der dyadischen Prädikatenlogik 1. Stufe CHURCH (1952), S. 184f.; CRAIG (1952), S. 188; BOOLOS (1980), S. 228, 232, 250f. u. 255ff.; QUINE (1985), S. 198. Die Prädikatenlogik läßt sich erst recht nicht mehr für dreistellige Prädikate entscheiden, in denen ein Existenz- von zwei Allquantoren in pränexer Normalform eingeschlossen wird; vgl. HERMES (1968), S. 258; HABEL (1983), S. 125; QUINE (1985), S. 198. Ein analoges Resultat präsentiert RAMSEY,F. (1965), S. 92ff., mit dem Nachweis, daß die Entscheidbarkeit der Prädikatenlogik für mehrstellige Prädikate nur so lange garantiert werden kann, wie keine Existenzquantoren zugelassen werden. Vgl. zur Grenzqualität einstelliger Prädikate des weiteren DELAHAYE (1987), S. 85.

84) Vgl. zur Entscheidbarkeit der Aussagenlogik LORENZEN,P. (1962), S. 69ff.; SMULLYAN (1968), S. 11 u. 15ff.; RAPHAEL (1976), S. 113ff.; RAUTENBERG (1979), S. 59f.; DELAHAYE (1987), S. 60 u. 107; BUCHER (1987), S. 86 u. 252; LUSTI (1990), S. 259.
Die aussagenlogische Entscheidbarkeit folgt in trivialer Weise aus der Definition endlicher Wahrheitswertetafeln für alle aussagenlogischen Formeloperationen. Falls eine aussagenlogische Formel K verschiedene atomare Formeln enthält mit $K \in \mathcal{N}_+$, dann existieren hierfür insgesamt 2^K Wahrheitswertekombinationen; vgl. z.B. RAMSEY,F. (1965), S. 6, 9 u. 276. Durch sukzessive Aggregation der Wahrheitswertetafeln für alle involvierten Formeloperationen läßt sich der Wahrheitswert der Gesamtformel für jede dieser Wahrheitswertekombinationen in endlicher Zeit bestimmen. Folglich ist nach höchstens 2^K Bestimmungsschritten entschieden, ob die betrachtete aussagenlogische Formel für *alle* kombinatorisch möglichen Wahrheitswertekombinationen ihrer K atomaren Formelkomponenten wahr ist oder nicht. Im ersten Fall ist die untersuchte Formel allgemeingültig; im zweiten dagegen nicht. Also läßt sich die Allgemeingültigkeit jeder aussagenlogischen Formel in endlicher Zeit entscheiden; q.e.d. (Die Entscheidung kann auch schon in weniger als 2^K Bestimmungsschritten im Sinne der Nichtallgemeingültigkeit fallen, sobald die erste Wahrheitswertekombination gefunden ist, für welche die Gesamtformel den Wahrheitswert "falsch" annimmt). Subtilere Entscheidungsalgorithmen als der voranstehend skizzierte finden sich in den o.a. Quellen. Effizientere, aber immer noch anschauliche Beispielalgorithmen stellen z.B. SMULLYAN (1968), S. 16ff., und RAUTENBERG (1979), S. 55ff., vor.

85) Beispielsweise kann das Erreichbarkeitsproblem der Petrinetz-Theorie nicht im Rahmen der Aussagenlogik, wohl aber mit Hilfe der Prädikatenlogik 1. Stufe formuliert werden. Die erhebliche mathematische und netztheoretische Relevanz dieses Erreichbarkeitsproblems wurde vom Verf. an anderer Stelle ausführlich dargelegt; vgl. ZELEWSKI (1989a), S. 111ff., insbesondere S. 120ff.

86) Vgl. zur Beschränktheit der monadisch-aristotelischen Logik gegenüber der "modernen", relational erweiterten Prädikatenlogik CARNAP (1930a), S. 15 u. 17f.; ACKERMANN,W. (1957), S. 3; STEGMÜLLER (1976b), S. 430ff.; REICHENBACH (1977), S. 324ff.; BUCHER (1987), S. 136, 172f. 197ff.
Die wesentliche Einschränkung der aristotelischen Logik liegt in ihrer Überlagerung des prädikatenlogischen Kalküls durch eine epistemologische Prämisse. (BUCHER (1987), S. 197, spricht zwar von einer ontologischen Voraussetzung, doch handelt es sich strenggenommen um einen Aspekt, der nicht das Welt*sein*, sondern das Welt*erkennen* betrifft.) Der aristotelischen Prämisse zufolge läßt sich *alles* Wissen über die Welt dadurch ausdrücken, daß jeweils einer unveränderlichen Substanz eine veränderliche Akzidenz zugeschrieben wird. Die Substanzen werden prädikatenlogisch als Individuen dargestellt, die Akzidenzien als 1-stellige Prädikate über diesen Individuen definiert. Da angeblich kein anderes Weltwissen denkmöglich sein sollte, bedurfte es nicht mehr als der 1-stelligen Prädikate.
Tatsächlich zeigte aber die eingehendere Beschäftigung mit rationalem Wissen, das Beziehungen zwischen Entitäten repräsentiert, die Begrenztheit der aristotelischen Epistemologie. Denn relationale Wissensinhalte können durch 1-stellige Prädikate nicht befriedigend ausgedrückt werden. Anschauliche Beispiele hierfür finden sich bei CARNAP (1930a), S. 17f.; BUCHER (1987), S. 199ff. Relationales Wissen erfordert vielmehr den Übergang zu einer Relationenlogik, die auch mehrstellige Prädikate umfaßt. Denn Relationen lassen sich als Extensionen von mindestens zweistelligen Prädikaten auffassen. Wissen über Relationen spielt auch innerhalb des Petrinetz-Konzepts eine zentrale Rolle. Um das Verhalten von Netzmodellen formal erfassen zu können, sind Erreichbarkeitsrelationen erforderlich. Vgl. auch zur Bedeutung relationalen Wissens für wissenschaftliche Disziplinen im allgemeinen CARNAP (1930a), S. 17f.; STEGMÜLLER (1976b), S. 431; BUCHER (1987), S. 200f.; sowie für die Sozialwissenschaften im besonderen OPP,K. (1976), S. 35ff.

87) Vgl. STEGMÜLLER (1976b), S. 431f.; HABEL (1983), S. 126.

88) Vgl. zur Unentscheidbarkeit der Modallogik BUCHER (1987), S. 252.

89) Darauf wird weiter unten näher eingegangen.

90) Näheres zu diesen modallogischen Komplikationen bei BUCHER (1987), S. (240f. u.) 252ff. Im wesentlichen lassen sich die modallogischen Schwierigkeiten darauf zurückführen, daß die Modallogik nicht mehr rein extensional ausgedrückt werden kann; vgl. dazu auch CARNAP (1960a), S. 41f. u. 114; BUCHER (1987), S. 240. Die modallogischen Komplikationen lassen sich auch daran ablesen, daß die Modallogik - im Gegensatz zur Prädikatenlogik 1.

Stufe - nicht mehr (semantisch) vollständig ist: In der Prädikatenlogik 1. Stufe können die Folgewirkungen der Unentscheidbarkeit durch die prädikatenlogische Vollständigkeit (und Korrektheit) immerhin noch auf das Phänomen der Semi-Entscheidbarkeit begrenzt werden (Näheres dazu anschließend). Da die Modallogik nicht mehr vollständig ist, fehlt ihr diese Eindämmung logischer Folgewirkungen der Unentscheidbarkeit. Daher drohen ihr weit gravierendere Folgen der Unentscheidbarkeit als lediglich die prädikatenlogische Semi-Entscheidbarkeit. Eine Eruierung dieser modallogischen Unentscheidbarkeitskonsequenzen liegt aber außerhalb des hier verfolgten Erkenntnisinteresses.

91) Vgl. dazu die oben erfolgten Ausführungen zur Inferenzmächtigkeit die analogen Argumentationen von HABEL (1983), S. 125 u. 127; ZELEWSKI (1986a), S. 942f.

92) Vgl. MELTZER (1975), S. 29ff.; ZELEWSKI (1986a), S. 937ff.; ZELEWSKI (1989a), S. 16.

93) Dies gilt allerdings nur so lange, wie die prädikatenlogische Unentscheidbarkeit auf das *allgemeine* Gültigkeitsproblem für beliebige prädikatenlogische Formeln bezogen wird. Dagegen existieren andere *spezielle* Entscheidungsprobleme, die ebenso unentscheidbar sind, aber dennoch von praktischem Interesse sein können. Darauf wird in Kürze eingegangen.

94) Besonders deutlich wird dies anhand des von GÖDEL untersuchten Problems, die Ableitbarkeit einer Formel aus einem Beweissystem zu entscheiden, die ihre eigene Unbeweisbarkeit behauptet. Darauf wurde schon hingewiesen. Ebenso sind Entscheidungsalgorithmen betroffen, die als "Super-Algorithmen" das Terminieren jedes beliebigen Algorithmus für jedes beliebige, algorithmisch zu lösende Problem festzustellen oder zu widerlegen vermögen; vgl. MINSKY (1971), S. 202; ZELEWSKI (1989c), S. 16f. Dies folgt aus der Unentscheidbarkeit des Halteproblems für TURING-Automaten, das bereits - neben anderen unentscheidbaren Problemen - vorgestellt wurde.

95) Ein aufschlußreicher Überblick über praktisch interessante, aber dennoch unentscheidbare Probleme findet sich bei BRAUER (1990a), S. 66. (Allerdings bezieht sich BRAUER nicht auf die Unentscheidbarkeit des logischen Gültigkeitsproblems, sondern auf die Unentscheidbarkeit des Halteproblems. Dies spielt hier jedoch keine Rolle, weil die Unentscheidbarkeiten beider Probleme unmittelbar miteinander zusammenhängen; vgl. dazu den Hinweis schon zum Halteproblem der TURING-Automaten.) Dabei gelangt BRAUER zu der überzeugenden Gesamteinschätzung: "Alle diese und viele andere Unmöglichkeitsaussagen sind für die Praxis der Informatik zumindest insoweit interessant, als sie zeigen, daß selbst sehr einfach erscheinende Probleme des symbolischen Rechnens ... nicht automatisch mit Hilfe eines Computers gelöst werden können, sondern daß man sich oft mit Teilantworten ... zufrieden geben muß ..." (S. 66). Die Bezugnahme auf "Teilantworten" spielt vermutlich auf das Phänomen der Semi-Entscheidbarkeit an, das in dieser Arbeit an anderer Stelle ausführlich erörtert wird.
Zu den unentscheidbaren, jedoch praktisch relevanten Problemen zählt beispielsweise HILBERT's 10. Problem. Es besteht in der Aufgabe zu entscheiden, ob für die Nullstellen eines Polynomials mit ganzzahligen Koeffizienten mindestens eine ganzzahlige Lösung existiert. Obwohl dieses Entscheidungsproblem von HILBERT bereits im Jahr 1900 formuliert wurde, konnte MATIJASEVIC seine Unentscheidbarkeit für beliebige Polynomiale erst siebzig Jahre später beweisen. Hieraus läßt sich u.a. die semantische Unvollständigkeit jeder axiomatisierten Zahlentheorie ableiten: Zu jeder Axiomatisierung der Zahlentheorie, die als Kalkül der Addition und Multiplikation über ganzen Zahlen definiert ist, existiert mindestens ein Polynomial, das keine ganzzahlige Lösung besitzt. Dies kann zwar durch semantische Schlußfolgerungen als gültig nachgewiesen werden. Doch es ist aufgrund der voranstehenden Unentscheidbarkeit unmöglich, diese Nichtexistenz innerhalb der zahlentheoretischen Axiomatisierung durch syntaktische Inferenzen abzuleiten; vgl. DAVIS,M. (1973a), S. 263f.; ZELEWSKI (1989a), S. 17. Eine betriebswirtschaftlich interessante Implikation dieses Unentscheidbarkeitsresultats besteht in der Erkenntnis, daß kein allgemeingültiger Algorithmus existieren kann, der *alle* ganzzahligen arithmetischen Entscheidungsprobleme - etwa Zuordnungsprobleme des Operations Research - zu lösen vermag; vgl. BACHEM (1980), S. 841; ZELEWSKI (1986a), S. 940f.
Vgl. allgemein zur Formulierung des 10. HILBERT'schen Problems und seiner Unentscheidbarkeit HILBERT (1900), S. 276; POST,E. (1944), S. 288; DAVIS,M. (1958), S. 102ff.; MENDELSON (1964), S. 254f.; MATIJASEVIC (1970), S. 354ff.; DAVIS,M. (1973a), S. 233f. u. 262 i.V.m. S. 237ff.; DAVIS,M. (1973b), S. 85, 87 u. 90f.; HACK,M. (1975a), S. 95; COHORS-FRESENBORG (1977), S. 47ff.; BACHEM (1980), S. 841; ZELEWSKI (1986a), S. 940f.; ZELEWSKI (1989a), S. 14 u. 17f.; BRAUER (1990a), S. 66.

96) Ein Beispiel dafür wurde schon in der voranstehenden Anmerkung aus dem Bereich des Operations Research vorgelegt.

97) Die Semi-Entscheidbarkeit der Prädikatenlogik wird an anderer Stelle ausführlich besprochen.

98) GÖDEL (1931), S. 193, Fn. 55, hat diesen hat diesen Sachverhalt markant ausgedrückt: Für jede prädikatenlogische Formel kann gezeigt werden, daß sie "entweder als allgemeingültig nachweisbar ist oder ein Gegenbeispiel existiert;" (Anmk. des Verf.: Erstes folgt aus der oben erläuterten semantischen Vollständigkeit der Prädikatenlogik. Zweites resultiert aus dem Prinzip des tertium non datur, dem zufolge für jede Formel, die nicht allgemeingültig ist, mindestens ein Formelinterpretation existieren muß, unter der diese Formel ungültig ist.) "die Existenz dieses Gegenbeispiels ist aber ... nicht immer nachweisbar". Diese Möglichkeit, daß sich ein tatsächlich existierendes Gegen-

beispiel innerhalb eines Beweissystems nicht nachweisen läßt, beruht letztlich darauf, daß sich das Gegenbeispiel in einem unendlich großen logischen Suchraum befindet, der mit den Inferenzversuchen eines Beweissystems untersucht werden muß. Dabei kann es geschehen, daß die Inferenzversuche unendlich lange am tatsächlich existierenden Gegenbeispiel vorbeiführen. Darauf wird später zurückgekommen. Zugleich wird hierdurch nochmals die Kritik gegenüber nicht-konstruktiven Beweistechniken bestärkt, die bereits thematisiert wurde: Das bloße Wissen über die Existenz eines Gegenbeispiels garantiert keineswegs, die behauptete Allgemeingültigkeit einer tatsächlich nicht allgemeingültigen Formel mit endlichen Ressourcen auch wirklich nachweisen zu können.

99) Das schließt nicht aus, daß *einzelne* Entscheidungsprobleme, in denen die Allgemeingültigkeit mindestens einer tatsächlich nicht allgemeingültigen Formel behauptet wird, mit endlichem Ressourceneinsatz als nicht-allgemeingültig erkannt werden. Es steht nur fest, daß dieses Erkennen der Nicht-Allgemeingültigkeit *nicht für alle* Entscheidungsprobleme gewährleistet werden kann, welche die Allgemeingültigkeit mindestens einer tatsächlich nicht allgemeingültigen Formel behaupten.

100) Auf einen Ausnahmefall wird an anderer Stelle hingewiesen.

101) Diese zusätzlichen Komplikationen werden an anderer Stelle thematisiert.

102) Vgl. BEETZ (1986), S. 40; SCHARF,A. (1987), S. 24. Vor allem hebt HABEL (1983), S. 121, "die Existenz zahlreicher interessanter Resultate in Bezug auf syntaktische und semantische Eigenschaften, insbesondere das 'Zusammenspiel' von Modelltheorie und Beweistheorie" hervor. (Vgl. zur Modell- und Beweistheorie die Anmerkungen zur prädikatenlogischen Vollständigkeit und Korrektheit.)

103) Vgl. STEGMÜLLER (1984b), S. 461ff.; RICHTER,M.M. (1988), S. 18.
Beispielsweise gilt das Kompaktheitstheorem für die nonmonotonen Logiken, die an späterer Stelle angesprochen werden, nicht mehr; vgl. REINFRANK (1985b), S. 24 u. 41.

104) Vgl. zu Inhalt und Bedeutung des LÖWENHEIM-SKOLEM-Theorems FRAENKEL (1930), S. 298; GENTZEN (1938), S. 11f.; MENDELSON (1964), S. 69 i.V.m. S. 65ff.; SMULLYAN (1968), S. 61 u. 63f.; KLEENE (1976), S. 761f.; BOOLOS (1980), S. 131, 141f. u. 147ff.; PUTNAM,H. (1982b), S. 10ff.; STEGMÜLLER (1983), S. 89, Fn. 9; HABEL (1983), S. 123; STEGMÜLLER (1984b), S. 321; ALISCH (1987), S. 261f. (allerdings zieht er daraus fehlerhafte Folgerungen); DELAHAYE (1987), S. 79.
Nach dem LÖWENHEIM-SKOLEM-Theorem besitzt jedes logische Formelsystem, das mindestens ein Modell aufweist, auch mindestens ein höchstens abzählbar-unendliches Modell. Unter Vorwegnahme der späteren Erläuterungen zur prädikatenlogischen Semantik bedeutet dies: Falls ein prädikatenlogisches Formelsystem durch mindestens eine Interpretation erfüllt wird, dann gibt es auch mindestens eine Interpretation mit höchstens abzählbar-unendlichem Definitionsbereich, die das Formelsystem erfüllt. Diese Eigenschaft ist insofern bedeutsam, als die prädikatenlogische Semantik keine Rücksicht auf den - logisch und mathematisch problematischen - Bereich überabzählbarer Unendlichkeiten nehmen muß. Die Kritik an diesem Unendlichkeitsbereich des reellen Kontinuums stammt vor allem aus dem Lager des logischen Konstruktivismus; vgl. GENTZEN (1938), S. 12ff. Vgl. darüber hinaus zu den mathematischen Schwierigkeiten beim Umgang mit überabzählbaren Unendlichkeiten HILBERT (1925), S. 169ff., 180ff. u. 190; FRAENKEL (1930), S. 286f. u. 292; ACKERMANN,W. (1957), S. 15f.; MESCHKOWSKI (1967), S. 53f. u. 61ff.; KÖRNER,S. (1968), S. 73ff.; LUXEMBURG (1973), S. 38; STEGMÜLLER (1976b), S. 436ff.; GETHMANN (1980b), S. 18, sowie die Anmerkungen zur Ablehnung aktuell unendlicher Größen.
Diese Probleme führten dazu, daß sich ein eigenständiger Zweig der Mathematik entwickelt hat, der sich speziell dem Umgang mit überabzählbaren Kontinua widmet. Vgl. zu dieser "Nonstandard-Analysis" LUXEMBURG (1973), S. 38ff.; KLEENE (1976), S. 762; ARENS (1985), S. 147ff. (ein Überblicksartikel).

105) Vgl. zu Inhalt und Bedeutung des Kompaktheits-Theorems GÖDEL (1930), S. 358; SMULLYAN (1968), S. 30ff., insbesondere S. 32; KLEENE (1976), S. 762; RICHTER,M.M. (1978), S. 61 u. 76; RAUTENBERG (1979), S. 41ff.; BOOLOS (1980), S. 131 u. 140f.; STEGMÜLLER (1984b), S. 295ff. u. 321; REINFRANK (1985b), S. 19; DELAHAYE (1987), S. 61 u. 110.
Das Kompaktheits-Theorem besagt grob gesprochen: Für eine Formelmenge gibt es genau dann mindestens eine (keine) Interpretation, die alle Formeln aus der Gesamtmenge erfüllt, wenn jede (mindestens eine) endliche Teilmenge dieser Formelmenge durch jeweils mindestens eine (keine) Interpretation erfüllt wird. Dabei darf die Gesamtmenge durchaus abzählbar unendlich viele Formeln enthalten. Dieses Theorem ermöglicht es, mittels *endlicher* Schlußweisen Erkenntnisse über die Erfüllbarkeit (Unerfüllbarkeit) *unendlicher* Formelmengen zu gewinnen; vgl. RAUTENBERG (1979), S. 42. Ebenso folgt daraus, daß jede Formel aus einer *endlichen* Menge von Axiomen (Prämissen) abgeleitet werden kann, sofern sie sich überhaupt beweisen läßt; vgl. REINFRANK (1985b), S. 19. Aufgrund der voranstehenden Erläuterungen zeichnet sich das Kompaktheitstheorem dadurch aus, die Finitheit von Schlußfolgerungen zu garantieren.

106) Die Nachteile betreffen sowohl die Ausdruckskraft als auch die Inferenzmächtigkeit der Prädikatenlogik. In der ersten Hinsicht ist es z.B. unmöglich, nicht-kategorische Quantifikationen wie "meistens" oder "typisch" darzustellen. Aus der anderen Perspektive handelt es sich z.B. um Inferenzmonotonie, der zufolge ein gültiges Prädikat

immer gültig bleiben muß. Diese beiden exemplarisch angeführten Schwächen der Prädikatenlogik 1. Stufe werden besonders anschaulich von HABEL (1983), S. 127ff. u. 131ff., behandelt.
Auf weitere Einschränkungen der Prädikatenlogik 1. Stufe weisen z.B. HASENJAEGER (1968), S. S. 243f., und RICHTER,M.M. (1988), S. 21 u. 23f., hin. Sie werden hier aber nicht mehr im Detail diskutiert, weil sie im Rahmen Synthetischer Netze und Flexibler Fertigungssysteme keine Rolle spielen. So bezieht sich HASENJAEGER im wesentlichen auf transfinite ("aktual unendliche") Sachverhalte, die das prädikatenlogische Ausdrucksvermögen übersteigen. RICHTER beschäftigt sich mit der Begrenzung der Prädikatenlogik auf nur zwei Wahrheitswerte und die rein extensionale Herleitung aller Formelgültigkeiten aus den Gültigkeiten ihrer Teilformeln. Die Möglichkeit, auf eine dreiwertige Logik überzugehen, wird in dieser Arbeit an anderer Stelle diskutiert und schließlich verworfen. Auf die Fragwürdigkeit nicht-extensionaler Konzepte wurde bereits näher eingegangen.
Vgl. zu weiteren Beschränkungen der Prädikatenlogik 1. Stufe OPP,K. (1976), S. 357, BEETZ (1986), S. 41; ZELEWSKI (1986a), S. 191ff.
Einige der dort angeführten Nachteile wurden allerdings schon voranstehend als scheinbar entlarvt, wie z.B. die angebliche Unmöglichkeit, bestimmte Schlußfolgerungen zu ziehen oder unscharfes Wissen zu repräsentieren.

107) Dies betrifft insbesondere die Monotonie der konventionellen Prädikatenlogik, die bereits angesprochen wurde. Sie führt zu erheblichen Schwierigkeiten bei der Repräsentation temporalen Wissens. Später wird sie aber durch die Einführung zeitlich variabler Prädikatsextensionen überwunden.

4.2.2 Die konventionelle Prädikatenlogik

4.2.2.1 Die syntaktische Dimension

Die algebraischen Formeln, die bereits im Kontext von SIG-Spezifikationen eingeführt worden sind, werden nun zu konventionellen prädikatenlogischen Formeln[1] verallgemeinert[2]. Drei wesentliche Merkmale prädikatenlogischer Formeln wurden bereits im Rahmen der SIG-Spezifikationen eingeführt. Es handelt sich um innere Formelstrukturierungen durch Terme, um aussagenlogische Formeloperationen und um Wahrheitswerte von negierten oder verknüpften Formeln. Als echte prädikatenlogische Erweiterung der algebraischen Formeln kommen nunmehr Quantifizierungen über Termen hinzu[3]. Diese Konstituenten des konventionellen prädikatenlogischen Kalküls werden systematisch zusammengefaßt und inhaltlich vertieft.

Definition: Prädikatenlogischer Kalkül 1. Stufe

Der prädikatenlogische Kalkül 1. Stufe besteht aus einer Menge zulässiger Zeichen und einer Menge von Formierungsregeln, die zur Erzeugung zulässiger prädikatenlogischer Formeln dienen[4]:

zulässige Zeichen:

- logische Zeichen
  - Konnektoren
    - Konjunktor: $\wedge$
    - Adjunktor : $\vee$
    - Disjunktor: $\underline{\vee}$
    - Subjunktor: $\rightarrow$
    - Bijunktor : $\leftarrow$
  - logische Operatoren
    - Negator: $\neg$
    - Quantoren
      - Allquantor: $\forall(X)$:
      - Existenzquantor: $\exists(X)$:
      - Einsquantor: $\underline{\exists}(X)$:

Fortsetzung auf der nächsten Seite ...

... Fortsetzung von der voranstehenden Seite:

- deskriptive Zeichen
  - Individuen:
    - Variablen: X
    - Konstantensymbole: Ko
  - Funktionssymbole: Fun: $st_1...st_K \rightarrow st_{K+1}$ mit $K \in \mathcal{N}_+$
  - Terme:
    - Jedes Individuum ist ein Term te.
    - Wenn Fun: $st_1...st_K \rightarrow st_{K+1}$ ein Funktionssymbol ist und wenn $te_1,...,te_K$ Terme sind, dann ist $te_{K+1} = Fun(te_1,...,te_K)$ ein Term.
    - Andere Terme sind unzulässig.
  - Prädikatssymbole:
    - $Prä(st_1,...,st_K)$ mit $K \in \mathcal{N}_+$
    - Prä()
- Hilfszeichen: "(" und ")" sowie ","

Formierungsregeln für zulässige Formeln:

- atomare Formeln
  - FR_P: Wenn $Prä(st_1,...,st_K)$ ein Prädikatssymbol ist und wenn $te_1,...,te_K$ Terme darstellen, dann ist $prä(te_1,...,te_K)$ eine Formel.
  - FR_A: Wenn Prä() ein Prädikatssymbol ist, dann ist prä() eine Formel.
- zusammengesetzte Formeln
  - FR_K: Wenn p_1 und p_2 Formeln sind und wenn & einen Konnektor darstellt, dann ist auch $p_1 \& p_2$ eine Formel.
  - FR_O: Wenn p eine Formel darstellt und wenn O ein logischer Operator ist, dann ist auch Op eine Formel.
- Alle anderen Formeln sind unzulässig.

Erläuterungen und Ergänzungen zur Definition des prädikatenlogischen Kalküls 1. Stufe:

a) Es wird vereinfachend von "dem" prädikatenlogischen Kalkül 1. Stufe gesprochen. Tatsächlich existiert eine breite Vielfalt ähnlicher, aber im Detail variierender Kalkülisierungen der Prädikatenlogik 1. Stufe[5)]. Diese Variationen unterscheiden sich aber nur hinsichtlich ihrer Begriffsbildungen und formalen Notationsweisen, ohne daß hiervon die grundsätzliche Ausdruckskraft prädikatenlogischer Kalküle tangiert wäre. Daher wird von solchen Kalkülvariationen abstrahiert.

b) Es wurde bewußt eine redundante Darstellungsweise[6)] gewählt, die mehr logische Konnektoren umfaßt, als für die Entfaltung der prädikatenlogischen Ausdruckskraft erforderlich sind. Dies erlaubt bei der Anwendung des prädikatenlogischen Kalküls kompaktere Formelnotationen.

c) Die formalen Objekte, auf die sich Quantoren bzw. Prädikatssymbole in ihren Argumenten erstrecken, dürfen im prädikatenlogischen Kalkül 1. Stufe[7)] nur Terme darstellen. Diese Terme stellen per definitionem entweder prädikatenlogische Individuen dar, oder sie sind aus diesen durch Funktionssymbole zusammengesetzt[8)]. Dadurch werden vor allem Individuenmengen und Prädikatsvariablen[9)] als Bestandteile der Argumente von Quantoren und Prädikatssymbolen ausgeschlossen. Dies bedeutet beispielsweise für einen Quantor, daß er sich stets auf *alle* Individuen aus seinem Definitionsbereich[10)] beziehen muß. Ein partieller Quantorbezug auf echte Teilmengen seines Definitionsbereichs ist dagegen ausgeschlossen. In prädikatenlogischen Kalkülen 2. Stufe können dagegen Individuenmengen und Prädikatsvariablen in den Argumenten von Quantoren und Prädikatssymbolen enthalten sein[11)]. Auf diese Weise lassen sich insbesondere charakteristische Eigenschaften von Prädikatssymbolen[12)] aus Kalkülen 1. Stufe einfach und formal präzise definieren[13)]. Trotz dieser formallogisch interessanten Aspekte kommt diese Arbeit mit einem Kalkül 1. Stufe aus[14)]. Daher wird auch vereinfachend vom prädikatenlogischen Kalkül gesprochen, ohne seine Stufenposition zu explizieren.

d) Der prädikatenlogische Kalkül ist die Syntax einer formalen Sprache. Die Sprachsyntax wird durch zwei Komponenten charakterisiert: ihr Alphabet und ihre Grammatik. Die formalen Objekte dieser Sprache sind die zulässigen prädikatenlogischen Zeichen. Die Menge aller dieser Zeichen ist das prädikatenlogische Alphabet. Das Alphabet besteht aus den beiden disjunkten, nicht-leeren Teilmengen der logischen und der deskriptiven Zeichen. Die objektsprachlichen Ausdrücke, die aus diesen Zeichen mit Hilfe der Formierungsregeln gebildet werden können, sind die prädikatenlogischen Formeln. Die prädikatenlogische Grammatik ist die Gesamtheit der Formierungsregeln[15)]. Jede Sammlung prädikatenlogischer Formeln, die als zusammenhängendes Ganzes angesehen wird, heißt ein prädikatenlogisches System[16)]. Die prädikatenlogische Objektsprache[17)] ist die Menge aller prädikatenlogischen Formeln, die in zulässiger Weise syntaktisch erzeugt werden können[18)].

e) Die logischen Zeichen werden auch als logische Konstanten oder logische Operatoren i.w.S. bezeichnet. Im letzten Fall werden der Negator und die Quantoren als Operatoren i.e.S. oder 1-stellige Operatoren präzisiert. Die Konnektoren heißen dann auch 2-stellige logische Operatoren oder Junktoren. In diesem weit gefaßten Begriffsverständnis werden die 2-stelligen Operatoren mit dem Negator zur Gruppe der aussagenlogischen Operatoren zusammengefaßt[19)]. Die Quantoren stellen dagegen die spezifisch prädikatenlogischen Operatoren dar.

f) Die All-, Existenz- und Einsquantoren[20)] stellen generalisierte Kon-, Ad- bzw. Disjunktionen dar[21)]. Die Argumente "(X):" von Quantoren enthalten jeweils genau eine Variable X, über die eine Quantifikation ausgesprochen wird[22)]. Eine Variable, die im Argument einer prädikatenlogischen Formel enthalten ist, heißt genau dann gebunden (frei), wenn dieser Formel (k)ein Quantor vorangestellt ist, dessen Argument diese Variable enthält. Formeln, die mindestens einen (keinen) Quantor enthalten, werden als quantifizierte (quantorfreie) Formeln bezeichnet.

g) Deskriptive Zeichen sind Namen im weitesten Sinne für alle formalen Objekte, die keine logischen Zeichen sind. Sie dienen dem Aufbau einer unbegrenzten Vielfalt prädikatenlogischer Ausdrücke. Die Menge aller deskriptiven Zeichen stellt eine offene, potentiell unendliche[23)], aber höchstens abzählbar-unendliche[24)] Menge dar. Diese Unendlichkeit gilt sowohl für die Teilmenge der Individuen als auch für die Teilmengen der Funktions- und Prädikatssymbole[25)]. Die Teilmengen aller Individuen und aller Prädikatssymbole werden jeweils als nicht-leer vorausgesetzt; die Menge der Funktionssymbole kann leer sein.

h) Da die Teilmenge der deskriptiven Zeichen unendlich ist, fällt auch das prädikatenlogische Alphabet unendlich groß aus.

i) Individuen eines logischen Kalküls sind deskriptive Zeichen ohne innere Struktur[26)]. Sie treten als Konstantensymbole oder Variablen auf[27)].

j) Konstantensymbole[28)] sind alle syntaktisch definierten Individuen, die im Rahmen einer prädikatenlogischen Semantik jeweils eine eindeutige Interpretation besitzen. Sie werden in der Literatur zur konventionellen Prädikatenlogik fast immer nur als Konstanten angesprochen[29)]. Strenggenommen werden Konstantensymbole aber erst im Rahmen von teilevaluierten Termen oder durch semantische Kalkülinterpretation auf jeweils genau eine symbolspezifische Konstanten abgebildet.

k) Variablen[30)] sind alle syntaktisch definierten Individuen, denen im Rahmen einer prädikatenlogischen Semantik keine eindeutige Interpretation zukommt. Sie können dort auf verschiedene Konstanten abgebildet werden. Dies schließt - als Vorgriff auf die Implementierung prädikatenlogischer Formeln durch die Programmiersprache PROLOG - auch die anonyme Variable "_"[31)] ein. Im konventionellen prädikatenlogischen Kalkül unterscheidet sie sich von anderen Variablen überhaupt nicht, sofern von ihrer PROLOG-spezifischen Notation abgesehen wird. Aber bei der späteren Formulierung und Auswertung von prädikatenlogischen Objektmodellen erweist sie sich als vorteilhaft[32)].

l) Funktionssymbole[33)] vertreten rechtseindeutige Abbildungen (Funktionen), solange deren Deklarationen noch nicht konkret bestimmt sind[34)]. Ansonsten stimmen Funktionssymbole mit Funktionen überein[35)]. Funktionen werden als semantische Konstrukte jedoch erst später eingeführt. 0-stellige Funktionssymbole fallen mit Konstantensymbolen zusammen[36)]. Konstantensymbole sind jedoch bereits als Individuen eingeführt. Daher werden 0-stellige Funktionssymbole zwecks Redundanzvermeidung nicht als eigenständige deskriptive Zeichen berücksichtigt.

m) Terme sind alle deskriptiven Zeichen, die entweder Basisterme darstellen oder aus diesen mit der Hilfe von Funktionssymbolen induktiv zusammengesetzt sind[37)]. Andere Terme werden nicht erlaubt[38)]. Bei den Basistermen handelt es sich um die prädikatenlogischen Individuen, also um Konstantensymbole und Variablen.

n) Prädikatssymbole[39)] stellen die zentralen Konstrukte des prädikatenlogischen Kalküls dar. Denn sie sind einerseits die einzigen originär definierten Zeichen, denen im Rahmen einer prädikatenlogischen Semantik Wahrheitswerte zugeordnet werden. Damit bilden sie den Angelpunkt zwischen dem objektsprachlichen Kalkül der Prädikatenlogik und seinem metasprachlichen Anwendungen in Modellierungskonzepten, innerhalb derer die Gültigkeit von realitätsbeschreibenden Formeln behauptet wird. Andererseits sind alle prädikatenlogisch definierbaren Formeln Derivate der Prädikatssymbole. Jedes Prädikatssymbol wird im Rahmen einer prädikatenlogischen Semantik auf eine Prädikatskonstante mit einer konkreten Bedeutung abgebildet[40)]. Daher läßt sich ein Prädikatssymbol als ein Platzhalter für eine Prädikatskonstante auffassen, dessen konkrete Bedeutung noch nicht festgelegt ist[41]. Prädikatsvariablen sind dagegen in der Prädikatenlogik 1. Stufe nicht vorgesehen[42].

o) K-stellige Prädikatssymbole Prä(st_1,...,st_K) mit $K \in \mathcal{N}_+$ bilden als intern strukturierte Zeichen eine Antipode[43] zu den strukturlosen Individuen. Ein Prädikatssymbol besteht aus dem Prädikatsnamen[44] "Prä" und dem K-stelligen Prädikatsargument (st_1,...,st_K). Das Prädikatssymbol Prä(st_1,...,st_K) kann verkürzt mit seinem Namen als Prädikatssymbol "Prä" angesprochen werden[45]. Jede Komponente oder Stelle[46] st_k mit $k \in \{1,...,K\}$ aus dem Prädikatsargument ist ein Platzhalter für einen Term te_k. Es wird hier vorausgesetzt, daß ein Prädikatssymbol bereits durch seinen Namen eindeutig festgelegt ist[47]. 1-stellige Prädikatssymbole lassen sich als Eigenschafts- oder Attributsymbole[48] auffassen[49]. Mehrstellige Prädikatssymbole stellen dagegen Relationssymbole[50] dar[51].

p) Zusätzlich werden auch 0-stellige Prädikatssymbole Prä() zugelassen[52]. Sie stellen den aussagenlogischen Grenzfall des Prädikatsbegriffs dar. Daher können sie auch als Aussagensymbole angesehen werden. Solche Aussagen- oder Prädikatssymbole besitzen keine innere Struktur. Ihre Argumente sind zum leeren Tupel () degeneriert. 0-stellige Prädikatssymbole bieten sich immer dann an, wenn lediglich interessiert, ob eine sachverhaltsbeschreibende Aussage zutrifft (gültig[53] ist) oder aber nicht (ungültig[54] ist). Um die formale Behandlung von 0-stelligen Prädikatssymbolen mit derjenigen von K-stelligen Prädikatssymbolen mit $K \in \mathcal{N}_+$ zu vereinheitlichen, dürfen fortan alle Prädikatssymbole als Prä(st_1,...,st_K) mit $K \in \mathcal{N}_0$ notiert werden. Dabei wird für den Grenzfall K=0 das entartete Prädikatargument zu (st_1,...,st_K) = () festgelegt

q) Falls mehrere Prädikatssymbole voneinander unterschieden werden sollen, kann dies durch eindeutig identifizierende Indices "u" geschehen: Prä_u(st_1,...,st_{Ku}). PRÄ ist die Menge aller Namen Prä_u von Prädikatssymbolen, die für ein prädikatenlogisches Formelsystem definiert sind: PRÄ = {Prä_u: u = 1,...,U} mit $U \in \mathcal{N}_+$. Da alle Prädikatssymbole durch ihre Namen eindeutig identifiziert werden, können in der Menge PRÄ auch alle Prädikatsnamen Prä_u durch die jeweils benannten Prädikatssymbole Prä_u(st_1,...,st_{Ku}) ersetzt werden: PRÄ = {Prä_u(st_1,...,st_{Ku}): u = 1,...,U}. Daher wird die Menge PRÄ allgemein als Prädikatssymbolemenge angesprochen.

r) Durch die Formierungsregeln werden alle zulässigen prädikatenlogischen Formeln induktiv definiert[55]. Die Induktionsbasis bilden die atomaren Formeln[56]. Sie werden mit Hilfe der Formierungsregeln FR_A und FR_P aus originär definierten Prädikatssymbolen abgeleitet. Falls es sich um mindestens 1-stellige Prädikatssymbole handelt, gehen in die daraus abgeleiteten Formeln zusätzlich Terme als deskriptive Zeichen ein. Darauf wird nachfolgend detaillierter zurückgekommen. Zusammengesetzte Formeln gehen aus bereits eingeführten - atomaren oder weniger komplex zusammengesetzten - Formeln hervor, indem diese mit Hilfe von logischen Zeichen kombiniert werden. Die Kombinationen werden durch die Formierungsregeln FR_K und FR_O erzeugt je nachdem, ob als logische Zeichen Konnektoren bzw. Operatoren eingesetzt werden[57]. Atomare und zusammengesetzte Formeln werden gemeinsam als prädikatenlogische Formeln bezeichnet.

s) Alle zulässigen prädikatenlogischen Formeln, die durch eine endliche Anzahl von Anwendungen der zulässigen Formierungsregeln aus den zulässigen prädikatenlogischen Zeichen erzeugt werden können, heißen auch wohlgeformte Formeln des prädikatenlogischen Kalküls[58]. In dieser Arbeit werden nur wohlgeformte Formeln berücksichtigt. Dies erzwingt die abschließende Exklusionsregel, daß keine anderen als die induktiv gebildeten atomaren oder zusammengesetzten Formeln zulässig sind[59].

t) Alle wohlgeformten Formeln werden vereinfacht als Prädikate bezeichnet und mit Kleinbuchstaben notiert[60]. Darüber hinaus wird als verkürzte Sprechweise zugelassen, auch Prädikatssymbole selbst als Prädikate zu benennen, wenn es im jeweils aktuellen Argumentationskontext entweder offensichtlich oder aber unerheblich ist, daß Prädikatssymbole gemeint sind.

u) Prädikate besitzen eine komplexe, dreifach gestufte interne Struktur. Erstens bestehen sie als zusammengesetzte Formeln aus atomaren Formeln und logischen Zeichen, d.h. Konnektoren und Operatoren. Zweitens ist jede atomare Formel aus einem Prädikatssymbol und Termen zusammengesetzt, sofern es sich nicht um den Grenzfall einer aussagenlogischen Formel handelt. Drittens läßt sich jeder Term seinerseits beliebig kompliziert aus Funktionssymbolen, Variablen und Konstantensymbolen aufbauen. Dieser Strukturreichtum eröffnet der Prädikatenlogik bereits aus syntaktischer Sicht[61] ein bemerkenswert großes Ausdruckspotential.

v) Ein Prädikat wird als eine Satzvariable[62] oder offene Formel bezeichnet, wenn sein Argument mindestens eine freie Variable umfaßt. Eine Satzvariable muß daher immer aus mindestens einem K-stelligen Prädikatssymbol mit $K \in \mathcal{N}_+$ abgeleitet sein. Ein Prädikat heißt dagegen ein Satz[63], falls es sich entweder um eine aussagenlogische Formel oder aber um eine geschlossene Formel handelt. Eine geschlossene Formel ist immer aus mindestens einem K-stelligen Prädikatssymbol mit $K \in \mathcal{N}_+$ abgeleitet, enthält aber in ihrem Argument keine freie Variable[64]. Geschlossene Formeln besitzen in erster Annäherung den Charakter von Aussagen[65]. Sie heißen auch All-, Existenz- oder Einsaussagen, wenn ihre Quantifikationen nur aus All-, Existenz- bzw. Einsquantoren bestehen.

w) Das 0-stellige Prädikatssymbol Prä() fällt gemäß der ersten Formierungsregel FR_A mit der atomaren Formel prä() weitgehend zusammen. Beide unterscheiden sich nur dadurch, daß die Namen von Prädikatssymbolen jeweils durch einen Großbuchstaben eingeleitet werden, die Namen der daraus abgeleiteten Formeln dagegen durch den entsprechenden Kleinbuchstaben. 0-stellige Formeln werden auch als aussagenlogische Formeln bezeichnet, weil sie die logische Qualität von Aussagen aus der strukturell einfacheren Aussagenlogik besitzen[66].

x) Das K-stellige Prädikatssymbol Prä($st_1,\ldots,st_K$) mit $K \in \mathcal{N}_+$ geht in eine atomare prädikatenlogische Formel prä($te_1,\ldots,te_K$) über, sobald seine Stellen st_k mit $k \in \{1,\ldots,K\}$ durch Terme te_k vollständig ersetzt werden[67]. Dies drückt die zweite Formierungsregel FR_P des prädikatenlogischen Kalküls aus.

y) Wenn ein Formelargument ($te_1,\ldots,te_K$) keine Variablen enthält, stellen alle seine Komponenten Grundterme gt_k dar. Es ist nur noch aus Konstantensymbolen und - gegebenenfalls - Funktionssymbolen aufgebaut. In diesem Fall wird von einem konstanten Formelargument ($gt_1,\ldots,gt_K$) gesprochen. Andernfalls - wenn das Formelargument mindestens eine Variable X_k enthält - liegt ein variables Formelargument ($\ldots,X_k,\ldots$)[68] vor. Die zugehörigen atomaren Formeln werden als konstante Formeln prä($gt_1,\ldots,gt_K$) bzw. als variable Formeln prä($\ldots,X_k,\ldots$) bezeichnet. Konstante Formeln werden synonym auch als Grundtermformeln angesprochen.

z) Von einer flachen Argumentstruktur wird gesprochen, wenn das Formelargument nur aus Basistermen besteht, also keine Funktionssymbole enthält. Falls es dagegen aus mindestens einem zusammengesetzten Term mit einem Funktionssymbol besteht, liegt eine tiefe Argumentstruktur vor. Entsprechend wird von flach bzw. tief strukturierten atomaren Formeln gesprochen. Wenn in einer tief strukturierten Formel die Stelle st_k des zugrundeliegenden Prädikatssymbols durch einen zusammengesetzten Term te_k mit $te_k=fun_k(\ldots)$ ersetzt ist, läßt sich die Formel als prä($\ldots,te_k,\ldots$) oder auch als prä($\ldots,fun_k(\ldots),\ldots$) notieren. Zwischen beiden Darstellungsvarianten vermittelt die Identität $te_k=fun_k(\ldots)$.

A) Die voranstehend entfalteten Variationsmöglichkeiten für die Erzeugung verschiedener atomarer Formeln aus demselben Prädikatssymbol können frei miteinander kombiniert werden. Beispielsweise erstreckt sich eine flach strukturierte, nur aus Grundtermen bestehende atomare Formel in ihrem Argument nur noch auf Konstantensymbole Ko_k. Sie wird als Formel prä(Ko_1,...,Ko_K)[69] notiert. Darüber hinaus darf dieselbe atomare Formel innerhalb eines prädikatenlogischen Systems beliebig oft angeführt werden. Dadurch kann sie am Aufbau unterschiedlicher zusammengesetzter Formeln teilnehmen.

B) Dasselbe Prädikatssymbol Prä(st_1,...,st_K) kann aufgrund der voranstehenden Erläuterungen - je nach der aktuellen Belegung seiner Argumentstellen durch Terme - in vielfach variierenden atomaren prädikatenlogischen Formeln prä(te_1,...,te_K) erscheinen. Jede dieser Erscheinungsweisen wird auch als uninterpretierte Formel[70], als Manifestation des Prädikatssymbols, als Prädikats- oder als Formelvorkommnis bezeichnet. Die Anzahl verschiedener oder auch identischer Vorkommnisse desselben Prädikatssymbols im selben prädikatenlogischen System ist grundsätzlich unbeschränkt[71]. Hierdurch wird für prädikatenlogische Formelsysteme eine bemerkenswerte Vielfalt zulässiger Prädikatsvorkommnisse eröffnet. Zugleich läßt sich jedes Prädikatssymbol Prä(st_1,...,st_K) als das abstrakte Schema aller seiner konkreten - realisierten oder auch nur potentiellen - Vorkommnisse prä(te_1,...,te_K) auffassen[72].

C) Der qualitative Aspekt verschiedenartiger Vorkommnisse desselben Prädikatssymbols Prä(st_1,...,st_K) basiert auf der induktiven Definition von Termen. Mit ihrer Hilfe ist es möglich, das Argument (te_1,...,te_K) eines Prädikatsvorkommnisses prä(te_1,...,te_K) in beliebigen Variationen zu gestalten. Dabei kann jeder involvierte Term te_k einen atomaren Term darstellen oder durch unterschiedliche Funktionssymbole aus anderen Termen zusammengesetzt werden. Hinzu kommt der quantitative Aspekt, daß Manifestationen desselben Prädikatssymbols mit identischen Argumenten innerhalb eines prädikatenlogischen Systems in beliebig vielen Formeln - also mehrfach - vorkommen dürfen.

D) Die Prädikatssymbole und die daraus formierten prädikatenlogischen Formeln, die für Synthetische Netze benötigt werden, heben sich von konventionellen Verwendungen des prädikatenlogischen Kalküls in zweierlei Hinsicht ab. Synthetische Netze schöpfen einerseits - prima facie - das Ausdrucksvermögen des prädikatenlogischen Kalküls hinsichtlich der Prädikatsstelligkeit und -zusammensetzung nur in geringem Umfang aus. Andererseits nutzen sie dieses Ausdruckspotential bei der Strukturierung von Formelargumenten konsequenter aus, als es konventionell üblich ist.

E) Das Ausdrucksvermögen der Prädikatenlogik wird im Regelfall zunächst nur benutzt, um jeder Stelle eines Synthetischen Netzes ein 1-stelliges[73] Prädikatssymbol zuzuordnen. Die eine Argumentstelle wird jeweils von einem Term eingenommen, der im Netzkontext als Marke interpretiert wird. Mehrstellige relationale Ausdrücke über diesen Marken, deren Formulierung mehrstellige Prädikatssymbole erforderte, wurden bisher im Rahmen des Petrinetz-Konzepts noch kaum verwendet[74]. Auch die Synthetischen Netze, mit deren Hilfe später die Module eines Modells für Flexible Fertigungssysteme formuliert werden, kommen überwiegend[75] mit 1-stelligen Prädikatssymbolen aus. Damit entspricht die prädikatenlogische Reichweite des Netzkonzepts in dieser Hinsicht auf den ersten Blick der monadischen, bis auf Aristoteles zurückreichenden "klassischen" Logik. Darüber hinaus werden die monadischen Prädikatssymbole der Stellen nur für die Formierung von atomaren und quantorfreien prädikatenlogischen Formeln verwendet. Weder All-, Existenz- oder Einsquantoren noch die aussagenlogischen Formeloperatoren werden verwendet, um aus den Prädikatssymbolen der Stellen zusammengesetzte Formeln zu bilden.

F) Ein Grund für die Dominanz quantorfreier atomarer monadischer Prädikate liegt darin, daß in Synthetischen Netzen oftmals tief strukturierte Prädikatsargumente verwendet werden. In das jeweils eine Prädikatsargument läßt sich ein Term einsetzen, der mit Hilfe von Funktionssymbolen in beliebig komplexer Weise aus Konstantensymbolen und Variablen zusammengesetzt sein kann. Sobald diese tiefe Argumentstruktur dadurch aufgelöst wird, daß die involvierten Konstantensymbole und Variablen auf derselben Ebene in einem Tupel nebengeordnet werden[76)], resultiert eine potentiell mehrstellige flache Argumentstruktur. Dies ist der Normalfall konventioneller Prädikat/Transition-Netze, deren Prädikatsargumente sich nicht auf Marken, sondern auf "Individuen" erstrecken. Aufgrund der schon früher geäußerten Kritik an dem hierbei fehlenden direkten Markenbezug wird auf *diese* flache Mehrstelligkeit von Prädikatsargumenten bei Synthetischen Netzen verzichtet. Zugleich wird deutlich, daß die weitgehende Beschränkung auf monadische Prädikatssymbole unerheblich ist. Denn dieselben Prädikatssymbole könnten im Prinzip durch Verflachung ihrer Argumente auch als mehrstellige Prädikatssymbole behandelt werden. Beispiele dafür werden auch in dieser Arbeit präsentiert[77)].

G) Die voranstehend angesprochene Dominanz monadischer Prädikatssymbole erstreckt sich nur auf diejenigen Prädikatssymbole, die in den Modulen eines Netzmodells explizit ausgewiesen werden, um sie den Stellen des Netzmodells zuzuordnen (Stellenprädikate[78)]). Daneben werden aber weitere Prädikatssymbole und daraus abgeleitete Formeln verwendet, um das Schaltverhalten von Transitionen näher zu bestimmen (Transitionsprädikate). Die Transitionsprädikate werden zwar zumeist nicht als prädikatenlogische Formeln, sondern als algebraische Formeln behandelt. Hierin liegt aber nur eine terminologische Nuance. Daher werden fortan Stellen- und Transitionsprädikate unter den gemeinsamen Oberbegriff der Netzprädikate subsumiert. Die Transitionsprädikate können das prädikatenlogische Ausdruckspotential vollständig ausschöpfen. Aus ein- oder auch mehrstelligen Prädikatssymbolen lassen sich mit Quantoren und aussagenlogischen Operatoren beliebig komplex zusammengesetzte Transitionsprädikate formulieren. Daher steht den Netzprädikaten im Prinzip das gesamte prädikatenlogische Ausdruckspotential offen.

H) Darüber hinaus wird später eine besondere Kategorie von Stellenprädikaten eingeführt, die als Informationsprädikate die Schnittstellen zwischen den Modulen eines Netzmodells bilden. Diese Informationsprädikate stellen zwar weiterhin quantorfreie atomare Formeln dar, beruhen aber nunmehr auf *mehr*stelligen Prädikatssymbolen.

I) Des weiteren besteht die Möglichkeit[79)], in einem Netz jede Transition mit ihren inzidenten Stellen als eine Formel in Klauselform aufzufassen. Das gesamte Netz repräsentiert dann eine beliebig komplex zusammengesetzte prädikatenlogische Formel in konjunktiver Normalform[80)]. Jede prädikatenlogische Formel läßt sich - einschließlich aller Quantoren - in eine äquivalente konjunktive Normalform transformieren. Für solche Normalformen wird später aufgezeigt, daß sie in systematischer Weise in äquivalente Netzmodelle transformiert werden können. Daher bleibt das Konzept Synthetischer Netze entgegen dem ersten Anschein keineswegs auf quantorfreie, atomare, 1-stellige Prädikate eingeschränkt. Statt dessen vermag es das syntaktische Ausdruckspotential der Prädikatenlogik 1. Stufe nahezu *vollständig* auszuschöpfen. Als einzige Ausnahme bleiben offene Formeln, in denen mindestens eine Variable durch überhaupt keinen Quantor gebunden ist, unberücksichtigt.

J) Die konsequentere Nutzung des prädikatenlogischen Kalküls betrifft vor allem die Argumentstruktur von Netzprädikaten[81)]. Im konventionellen prädikatenlogischen Kalkül werden zumeist nur flach strukturierte Prädikatsargumente konkret verwendet[82)]. Ihre Komponenten stellen strukturlose Individuen (Basisterme) dar, also nur entweder Konstantensymbole oder Variablen[83)]. Die Komponenten der Argumente von Netzprädikaten können dagegen auch zusammengesetzte Terme sein, die in beliebig komplexer Weise aus Konstantensymbolen, Variablen und Funktionssymbolen aufgebaut sind. Dadurch lassen sich Prädikate mit tief strukturierten Argumenten formulieren[84)]. Dies gilt insbesondere für die unären Stellenprädikate. Sie erstrecken sich in ihren Argumenten jeweils auf eine Marke, bei der es sich um einen beliebig komplex strukturierten Term handelt[85)]. Daher wird der übliche Individuenbezug von konventionellen, nur flach strukturierten Prädikatsargumenten zum Termbezug solcher Prädikate ausgeweitet, deren Argumente beliebig tief strukturiert sein können.

K) Darüber hinaus wird es zugelassen, daß dasselbe Prädikatssymbol mit der Hilfe von verschiedenen Funktionssymbolen zur Bildung alternativ strukturierter atomarer Formeln verwendet wird. Diese Formeln enthalten - obwohl sie aus demselben Prädikatssymbol abgeleitet wurden - unterschiedliche Argumentstrukturen. In einem solchen Fall wird auch von multipler Formelbildung gesprochen. Diese Ausnutzung des Ausdruckspotentials der Prädikatenlogik ist dem Verf. aus der einschlägigen Literatur nicht bekannt. Dort wird noch nicht einmal das Verhältnis zwischen Prädikatssymbolen und daraus formierten atomaren Formeln thematisiert[86)]. Für die Integration von Prädikatenlogik und Sortenkonzept ist es aber in dieser Arbeit erforderlich, derart multipel definierte Argumentstrukturen für ein Prädikatssymbol zu gestatten. Denn nur hierdurch wird es später möglich sein, auch solche formalen Objekte (Marken) prädikatenlogisch zu erfassen, deren Sortenstruktur mittels der Subsortentechnik auf mehrfache Weise definiert ist[87)].

L) In den meisten konventionellen prädikatenlogischen Darstellungen wird zwischen Konstantensymbolen und Konstanten einerseits ebensowenig wie zwischen Funktionssymbolen (Operationssymbolen) und Funktionen (Operationen) andererseits differenziert. Statt dessen werden dort komplex strukturierte Prädikatsargumente - wenn überhaupt - aus Konstanten, Variablen und Funktionen aufgebaut[88)]. Der Anschluß an diese weit verbreitete, aber nichtsdestoweniger formal unpräzise Vorgehensweise wird später durch die Einführung teilevaluierter Terme im Rahmen der prädikatenlogischen Semantik hergestellt.

M) Die K-stelligen Operationssymbole, die mit $K \in \mathcal{N}_+$ früher durch Signaturen definiert wurden, finden sich im prädikatenlogischen Kontext in zwei verschiedenen Erscheinungsformen wieder[89)]. Sie stellen entweder K-stellige Funktionssymbole "Fun"[90)] oder aber K-stellige Prädikatssymbole "Prä"[91)] dar. Prädikats- und Funktionssymbole verhalten sich jedoch aus zwei Gründen kategorial verschieden. Erstens ist eine formale Semantik mit Wahrheitswerten und Extensionen nur für Prädikats-, nicht aber für Funktionssymbole definiert[92)]. Zweitens dienen Funktoren zum Aufbau von tief strukturierten Prädikatsargumenten. Prädikatsargumente und Prädikatssymbole stellen zwei deutlich unterschiedene prädikatenlogische Formulierungsebenen dar. Daher dürfen Funktionssymbole nicht auf derselben Ebene wie Prädikatssymbole behandelt werden[93)].

N) 0-stellige Operationssymbole aus einer Signatur lassen sich als 0-stellige Prädikats- oder Funktionssymbole auffassen. Sie stellen jeweils Konstantensymbole dar[94)]. Da Konstantensymbole bereits originär definiert worden sind, brauchen im prädikatenlogischen Kalkül nur mindestens 1-stellige Prädikats- und Funktionssymbole explizit erfaßt zu werden.

O) Das Hilfszeichen "," wird benutzt, um in Aufzählungen die betroffenen formalen Objekte optisch voneinander zu trennen. Die Hilfszeichen "(" und ")" dienen der Einkapselung logisch enger zusammenhängender Teilformeln und zugleich der Abgrenzung von ihrer Formelnachbarschaft.

P) Sofern durch die Klammerhilfszeichen keine anderen Zusammenhangsverhältnisse ausgewiesen sind, gilt für den Zusammenhang von Teilformeln: Der Negator bindet stärker als alle Konnektoren. Kon- und Adjunktor einerseits sowie Sub- und Bijunktor andererseits binden jeweils gleich stark. Aber das erste Konnektorpaar bindet stärker als das zweite. Alle Konnektoren binden stärker als Quantoren. Es können auch abundante Klammerhilfszeichen verwendet werden, die zwar nicht notwendig sind, aber in einer Formel die voranstehend festgelegten Bindungsverhältnisse optisch verdeutlichen.

Anmerkungen zum Kapitel:

1) Prädikatenlogische Formeln werden synonym auch als Prädikate bezeichnet. Dabei wird ein weit gefaßter Prädikatsbegriff unterstellt. Prädikate i.w.S. umfassen sowohl Prädikatssymbole (Prädikate i.e.S.) als auch alle hieraus abgeleiteten Formeln des prädikatenlogischen Kalküls. Darüber hinaus rechnen hierzu auch alle Interpretationen von Prädikatssymbolen bzw. Formeln im Rahmen der prädikatenlogischen Semantik. Als Prädikate i.e.S. werden hier die Prädikatssymbole und ihre Interpretationen angesehen.

2) Eine Verallgemeinerung liegt insofern vor, als einerseits jede algebraische Formeldefinition auf der Basis von Relationen durch die prädikatenlogische Formeldefinition äquivalent ersetzt werden kann. Andererseits erlaubt die prädikatenlogische Formeldefinition die Formulierung von Sachverhalten, die über das Konzept vorgegebener algebraischer Relationen hinausweisen. Die äquivalente Substitutionsmöglichkeit besteht, weil jede formeldefinierende algebraische Relation REL_j durch eine prädikatenlogische Wahrheitsfunktion wf_j mit folgender Eigenschaft ersetzt werden kann: Sie bildet die Argumente eines Prädikats $prä_j$ genau dann auf den Wahrheitswert "gültig" ab, wenn die Argumente Elemente der Relation REL_j sind. Die Wahrheitsfunktionen werden an anderer Stelle eingeführt. Die Verallgemeinerung erstreckt sich auf den Sachverhalt zeitvariabler Prädikatsextensionen. Bei der Formeldefinition durch algebraische Relationen wird stets unterstellt, daß diese Relationen zeitinvariant gelten. Damit ist die Menge aller Argumente, für welche die algebraischen Formeln bezüglich einer Termauswertung gültig sind, als Formelextension a priori fixiert. Prädikatenlogische Formeln können dagegen so formuliert werden, daß die Mengen der Argumente, für die sie bezüglich einer Termauswertung gültig sind (Prädikatsextensionen), im Zeitablauf variieren dürfen. Auf diesen Aspekt wird später ausführlich zurückgekommen.

3) Allerdings wurden auch sie schon im Kontext von Stelle/Transition-Netzen vorgestellt. Der charakteristischen Bedeutung solcher Quantifizierungen für die Prädikatenlogik wird auch dadurch Ausdruck verliehen, daß diese Logikvariante mitunter als Quantorenlogik angesprochen wird; vgl. z.B. MENDELSON (1964), S. 45; RESCHER (1968a), S. 35; STEGMÜLLER (1969), S. 49; OPP,K. (1976), S. 99; STEGMÜLLER (1976b), S. 433; ESSER,H. (1977a), S. 38; ESSLER (1982a), S. 44; STEGMÜLLER (1983), S. 53 u. 87; STEGMÜLLER (1984b), S. 1 u. 73.

4) Die Notation der logischen Zeichen lehnt sich an DIN 5474 (1973), S. 1ff., und ESSLER (1982a), S. 35, an (sofern die Zeichen dort enthalten sind).

5) Vgl. zu solchen Darstellungen des prädikatenlogischen Kalküls GENTZEN (1936), S. 502ff.; CARNAP (1960a), S. 44ff., 7ff., 24ff., 34ff., 38ff., 71ff. u. 79ff.; MENDELSON (1964), S. 46f.; STEGMÜLLER (1968), S. 210f.; SMULLYAN (1968), S. 43; CARNAP (1968), S. 15; LUXEMBURG (1973), S. 41f.; RICHTER,M.M. (1978), S. 58f.; LOVELAND (1978), S. 10; BOOLOS (1980), S. 96ff.; POTTHOFF (1981), S. 4ff.; REITER (1984), S. 194f.; LEVI,G. (1986), S. 397; DELAHAYE (1987), S. 66ff. Im wesentlichen lehnt sich die oben gewählte Darstellungsweise an PREIß (1989), S. 13f., an.

6) Die Darstellungsweise ist redundant, weil es im Prinzip ausreicht, einen Konnektor, den Negator und einen Quantor zu verwenden; vgl. RUSSELL,B. (1921), S. 190f.; WHITEHEAD (1925), S. 6; SMULLYAN (1968), S. 14; BUCHER (1987), S. 123ff.; DELAHAYE (1987), S. 68f. Hinsichtlich der aussagenlogischen Formeloperationen i.w.S., die aus Formelkonnektionen und -negationen bestehen, kommen in Betracht:

- die Negation und die Konjunktion; vgl. MENDELSON (1964), S. 26; STEGMÜLLER (1983), S. 50; DELAHAYE (1987), S. 56; BUCHER (1987), S. 123; SOWA (1984), S. 385.
- die Negation und die Adjunktion; vgl. SHEFFER (1913), S. 487; RUSSELL,B. (1921), S. 190f.; WHITEHEAD (1925), S. 6; CARNAP (1960a), S. 79 i.V.m. S. 31; MENDELSON (1964), S. 26; CARNAP (1968), S. 82; DELAHAYE (1987), S. 56; BUCHER (1987), S. 123.
- die Negation und Subjunktion; vgl. MENDELSON (1964), S. 26, 30f. u. 47; REINFRANK (1985b), S. 27; BUCHER (1987), S. 123; DELAHAYE (1987), S. 48, 56 u. 68f.

Es ist sogar möglich, mit nur genau einem aussagenlogischen Operator auszukommen. Dies leisten aber nicht die o.a. konventionellen aussagenlogischen Formeloperationen. Statt dessen muß auf eine "exotische" Verknüpfungsoperation zurückgegriffen werden. Dabei handelt es sich insbesondere entweder um die NAND- oder aber um die NOR-Formelverknüpfung; vgl. SHEFFER (1913), S. 482ff., insbesondere S. 487f.; RUSSELL,B. (1921), S. 190; MENDELSON (1964), S. 26f.; RAMSEY,F. (1965), S. 25; SMULLYAN (1968), S. 14; DIN 44300 (1972), S. 16; RAUTENBERG (1979), S. 14; SOWA (1984), S. 385f.; DIN 44300 (1985), Teil 5, S. 8; BUCHER (1987), S. 123ff. Die NAND-Operation oder Rejektion ("SHEFFER-Strich") liefert das negierte Konjugat $\neg(FOR_i \wedge FOR_j)$, die NOR-Operation ("PEIRCE-Funktion") das negierte Adjugat $\neg(FOR_i \vee FOR_j)$ zweier Formeln. Vgl. zu Verwendungen solcher NAND- und NOR-Verknüpfungen - vor allem im (elektro)technischen Bereich - WECK (1982), S. 72; BUCHER (1987), S. 126f. NAND- und NOR-Formelverknüpfungen sind die einzigen zweistelligen logischen Operatoren, aus denen sich alle aussagenlogischen Formeloperationen ableiten lassen; vgl. MENDELSON (1964), S. 27; SMULLYAN (1968), S. 14. Darüber hinaus gibt es weitere logische Konstrukte mit der gleichen Funktion, die aber komplizierter aufgebaut sind. Dazu gehört z.B. das FREDKIN-Gatter; vgl. BARKER (1990), S. 521f.

Darüber hinaus wird in prädikatenlogischer Hinsicht nur entweder der All- oder aber der Existenzquantor benötigt - der jeweils andere Quantor läßt sich auf den einen mittels mehrfacher Negation reduzieren; vgl. MENDELSON (1964), S. 47; DELAHAYE (1987), S. 68f.; BUCHER (1987), S. 169f.
Beispielsweise ist es möglich, die Adjunktion mittels der Konjunktion und Negation zu erklären. Die Disjunktion kann aus den drei vorgenannten Operatoren gewonnen werden. Subjunktive Formelverknüpfungen lassen sich vollständig auf Negation und Adjunktion reduzieren. Die Bijunktion kann auf Konjunktion und Subjunktion zurückgeführt werden. Der All- läßt sich auf den Existenzquantor und den Negator zurückführen. Mit p_1 und p_2 als zwei beliebigen prädikatenlogischen Formeln und p(X) als Formel, die über der Variablen X definiert ist, gilt dann:

$$p_1 \vee p_2 \quad :\Leftrightarrow \quad \neg(\neg p_1 \wedge \neg p_2)$$

$$p_1 \underline{\vee} p_2 \quad :\Leftrightarrow \quad (p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2)$$

$$p_1 \rightarrow p_2 \quad :\Leftrightarrow \quad \neg p_1 \vee p_2$$

$$p_1 \leftrightarrow p_2 \quad :\Leftrightarrow \quad (p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_1)$$

$$\forall(X): p(X) \quad :\Leftrightarrow \quad \neg(\exists(X): \neg p(X))$$

7) Vgl. zur folgenden Charakterisierung der ersten prädikatenlogischen Kalkülstufe WHITEHEAD (1925), S. 51; KLEENE (1952), S. 145; CARNAP (1960a), S. 65; HERMES (1961), S. 173; STEGMÜLLER (1969), S. 708; CHANG,C.L. (1973), S. 27ff.; STEGMÜLLER (1973a), S. 54; VAUGHT (1973), S. 4; ITZINGER (1976), S. 29f.; RAPHAEL (1976), S. 135; POTTHOFF (1981), S. 4; BARR,A. (1981), S. 165; BIBEL (1982a), S. 254; DELAHAYE (1987), S. 70.
Vgl. auch die deutliche Kritik, die STEGMÜLLER (1983), S. 93, an der Begriffsbildung "1. Stufe" übt. Er hält sie für eine unglückliche Überlagerung eines formallogischen Konzepts mit der Denkweise des mathematischen Denkkonzepts der Mengentheorie. Allerdings wurde das prädikatenlogische Stufenkonzept - vornehmlich von WHITEHEAD und RUSSELL - mit der Absicht entwickelt, die *logischen* Antinomien imprädikativer Begriffsdefinitionen zu überwinden, die bereits in einer früheren Anmerkung angesprochen wurden. Die Antinomienvermeidung führte zu einer mehrstufigen Hierarchie formalsprachlicher Ausdruckstypen, die sich im prädikatenlogischen Stufenkonzept widerspiegelt. Vgl. zu dieser logischen Typentheorie WHITEHEAD (1925), S. 37ff., insbesondere S. 51ff.; FRAENKEL (1930), S. 299f. u. 302; CARNAP (1930a), S. 19f.; CARNAP (1931c), S. 96ff.; CARNAP (1960a), S. 80ff.; RAMSEY,F. (1965), S. 24ff. u. 32ff.; WANG,HA. (1965), S. 30ff.; CHURCH (1976), S. 747ff.; STEGMÜLLER (1976b), S. 437f. Vgl. zu ihrer Zwecksetzung, Antinomien zu verhindern, WHITEHEAD (1925), S. 55, 60 u. 62ff.; FRAENKEL (1930), S. 299 u. 302; CARNAP (1930a), S. 19; CARNAP (1931c), S. 98; CARNAP (1960a), S. 82; RAMSEY,F. (1965), S. 24; WANDSCHNEIDER (1974), S. 75f.; CHURCH (1976), S. 747; STEGMÜLLER (1976b), S. 437; REICHENBACH (1977), S. 335.

8) In den meisten Darstellungen der Prädikatenlogik 1. Stufe wird nur der erste Aspekt - die unmittelbare Komposition von Prädikatsargumenten aus Individuen - herausgestellt. Diese Einseitigkeit geht auf WHITEHEAD (1925), S. 51, zurück. Der mittelbare Individuenbezug, der in Termen mit Funktionssymbolen vorliegt, wird oftmals übersehen. Hierauf wird aus der Perspektive tief strukturierter Prädikatsargumente zurückgekommen.

9) Gleiches gilt für Relationsvariablen, da diese auf Prädikatsvariablen zurückgeführt werden können. Individuenmengen stellen diesbezüglich nur einen oben hervorgehobenen Sonderfall dar. Denn jede Menge läßt sich als 1-stellige Relation auffassen.
Von den hier angesprochenen Prädikatsvariablen (Relationsvariablen) deutlich zu unterscheiden sind die Prädikatssymbole, die an späterer Stelle näher erläutert werden: Eine Prädikatsvariable ist eine Variable, die durch ein beliebiges Prädikat ersetzt ("gebunden") werden kann. Das ersetzende Prädikat läßt sich aus dieser Sicht auch als eine Prädikatskonstante bezeichnen. Ein Prädikatssymbol ist dagegen ein uninterpretierter, rein syntaktisch definierter Ausdruck, aus dem sich durch Formierungsregeln atomare prädikatenlogische Formeln bilden lassen. Darauf wird später zurückgekommen. Mit den "Prädikaten", die sich als Prädikatskonstanten für eine Prädikatsvariable einsetzen lassen, sind im allgemeinen diese atomaren prädikatenlogischen Formeln gemeint. Eine Prädikatsvariable ist somit ein *Platzhalter* für eine atomare prädikatenlogische Formel. Ein Prädikatssymbol stellt hingegen ein *Schema* für die Formierung von atomaren prädikatenlogischen Formeln dar.

10) Der Definitionsbereich für einen Quantor wird unten im Kontext interpretierter Formeln eingeführt.

11) Im Prinzip lassen sich prädikatenlogische Kalküle noch höherer Stufen bilden, indem die Konstrukte der jeweils letzten Stufe auf der nächsthöheren Stufe als Bestandteile von Quantor- und Prädikatsargumenten zugelassen werden. Diese Induktion immer komplexerer prädikatenlogischer Kalküle interessiert aber hier - wie schon der Kalkül 2. Stufe - nicht weiter.

12) Gleiches gilt für die Charakteristika von Relationssymbolen, wie z.B. deren Symmetrie oder Transitivität. Denn später wird gezeigt werden, daß sich alle Relations- als Prädikatssymbole auffassen lassen.

13) Vgl. RAPHAEL (1976), S. 135; COHEN,P. (1982), S. 83, und ZELEWSKI (1986a), S. 188, Fn. 1 (mit dem Beispiel, die Eigenschaft "Zyklusfreiheit" für das 2-stellige Prädikatssymbol "Präzedenzrelation" im Kontext der Graphentheorie und Netzplantechnik als Prädikat 2. Stufe zu behandeln); LUSTI (1990), S. 259 (Definition der Identitätsrelation durch eine prädikatenlogische Formel 2. Stufe). Auch die zweistellige Besserstellungsrelation $BES_{\trianglerighteq}$, die später verwendet wird, stellt ein Prädikat 2. Stufe dar. Denn es bezieht sich auf Schaltprozesse $PRO_{0.E/e.p}$, die ihrerseits die Qualität von Prädikatsvariablen besitzen.
Einen anderen Anwendungsfall der Prädikatenlogik 2. Stufe bildet die logische Repräsentation von Kausalgesetzen. Würde sie nicht zugelassen, müßte ein modallogischer Notwendigkeits-Operator verwendet werden. Vgl. zu diesem Aspekt MCCARTHY,J. (1968), S. 412f. Dies entspricht der grundsätzlichen Möglichkeit, die Modallogik (1. Stufe) durch die Prädikatenlogik (2. Stufe) zu reformulieren; vgl. CARNAP (1960a), S. 42; VAN EMDE BOAS (1978), S. 10f.
Des weiteren lassen sich Prinzipien, die für ein prädikatenlogisches Formelsystem 1. Stufe gelten sollen, als Formeln der Prädikatenlogik 2. Stufe mit Allquantoren ausdrücken. Vgl. dazu beispielsweise die Formulierung des arithmetischen Induktionsprinzips bei BAUER,F. (1989), S. 341. Ebenso kann LEIBNIZ' Prinzip der Identität des Ununterscheidbaren als Ausdruck der Prädikatenlogik 2. Stufe formal präzisiert werden. Dabei vertreten die beiden Variablen X_1 und X_2 zwei *beliebige* Objekte. Prä(st) stellt ein *beliebiges* einstelliges Prädikatssymbol dar. Es schreibt jedem Objekt, das seine Argumentstelle substituiert, eine prädikatspezifische Eigenschaft zu. Dann kann das Identitätsprinzip ausgedrückt werden durch die äquivalenten *All*aussagen 2. Stufe:

a) $\forall X_1 \, \forall X_2\text{: } (X_1 = X_2 \leftrightarrow \neg(\exists \text{Prä(st): } (\text{prä}(X_1) \wedge \neg\text{prä}(X_2)) \vee (\text{prä}(X_2) \wedge \neg\text{prä}(X_1))))$

b) $\forall X_1 \, \forall X_2\text{: } (X_1 = X_2 \leftrightarrow (\forall \text{Prä(st): } (\text{prä}(X_1) \rightarrow \text{prä}(X_2)) \wedge (\text{prä}(X_2) \rightarrow \text{prä}(X_1))))$

Vgl. zu diesen Formalisierungen WHITEHEAD (1925), S. 57; ESSLER (1982a), S. 45.
Die Prädikatenlogik 2. Stufe läßt sich auch als die formallogische Basis ansehen, auf der im Bereich der Künstlichen Intelligenz das Konzept der semantischen Rahmen für Wissensrepräsentationen beruht; vgl. RICHTER,M.M. (1988), S. 25. Denn die Rahmenkonstrukte enthalten Variablen ("slots"), die durch Prädikate ersetzt werden können, also im Prinzip Prädikatsvariablen darstellen. Das Konzept semantischer Rahmen ("frames") wurde bereits mit weiterführenden Quellenangaben angesprochen.
Des weiteren läßt sich die Prädikatenlogik 2. Stufe auf Realtheorien anwenden, die mit Hilfe der Prädikatenlogik 1. Stufe formuliert sind und theoretische Begriffe ohne unmittelbare empirische Bedeutung enthalten. Dann können alle theoretischen Begriffe dadurch eliminiert werden, daß sie durch Prädikatsvariablen ersetzt und entsprechende Existenzquantoren über diesen neuen Variablen eingeführt werden. Mit Hilfe dieser RAMSEY-Satz-Technik resultiert eine Realtheorie, die prädikatenlogisch auf 2. Stufe so reformuliert ist, daß sie keine theoretischen Begriffe enthält. Vgl. dazu RAMSEY,F. (1965), S. 223ff., insbesondere S. 228 u. 231; STEGMÜLLER (1970a), S. 400ff., insbesondere S. 404f.; STEGMÜLLER (1986a), S. 485f.; STEGMÜLLER (1986b), S. 318f. Prädikatenlogiken 2. und höherer Stufen lassen sich ebenso dazu benutzen, um logische Antinomien zu vermeiden. Darauf wurde bereits hingewiesen. Vgl. zu weiteren Anwendungsoptionen für eine Prädikatenlogik 2. Stufe THIEL,C. (1980b), S. 626 (Reformulierung des Extensionalitätsaxioms der Mengenlehre); MARTI-OLIET (1989), S. 335 (Formulierung globaler Systemeigenschaften).

14) Es gibt im Petrinetz-Konzept zwar Anknüpfungspunkte zu höherstufigen Prädikatenlogiken. Doch die hiervon betroffenen formalsprachlichen Konstrukte lassen sich noch innerhalb der Prädikatenlogik 1. Stufe und ihrer PROLOG-Implementierung erfassen. Vgl. auch die Anmerkung bei WHITEHEAD (1925), S. 55, daß prädikatenlogische Ausdrücke höherer Stufen nur von theoretischen Interesse sind. Vgl. diesbezüglich die exemplarischen Erläuterungen zu Anwendungsfällen der Prädikatenlogik 2. Stufe in der voranstehenden Fußnote.

15) Dies gilt einschließlich der Festlegung, daß keine anderen als die hierdurch erzeugbaren prädikatenlogischen Formeln zulässig sind.

16) Die prädikatenlogischen Formeln eines solchen Systems gelten immer als konjunktiv verknüpft. Daher kann jedes prädikatenlogische System als genau ein - beliebig komplex zusammengesetztes - Konjugat aufgefaßt werden. Dies gilt allerdings nur für den Regelfall, daß das prädikatenlogische System mindestens zwei Formeln umfaßt. Im degenerierten Einformelfall ist das prädikatenlogische System mit der einen Formel identisch.

17) Die zugehörige Metasprache wird unten durch Einführung einer prädikatenlogischen Semantik konstituiert.

18) Darüber hinaus kann auch noch das prädikatenlogische Alphabet zur Sprache der Prädikatenlogik gerechnet werden; vgl. REITER (1984), S. 195; PREIß (1989), S. 15.

19) Für die aussagenlogischen Operatoren gilt folgende, am Beispiel des Negators verdeutlichte Sprachregelung: Die Ausführung der logischen Operation, die der Negat*or* symbolisiert, wird als Negat*ion* bezeichnet. Das Ergebnis der Operationsausführung heißt Negat oder negierte Formel.

20) Der Einsquantor wird in Standardnotationen des prädikatenlogischen Kalküls im allgemeinen nicht verwendet. Eine seiner seltenen expliziten Erwähnungen findet sich bei WHITEHEAD (1925), S. 30f.; KLEENE (1952), S. 199;

DIN 5474 (1973), S. 3; SHAPIRO,S.C. (1979b), S. 791 u. 793; ALLEN,J. (1984), S. 127 u. 139. Vgl. auch zur Rechtfertigung, verallgemeinerte Quantoren zur Bereicherung der Prädikatenlogik einzuführen, RICHTER,M.M. (1988), S. 29f.
Der Einsquantor wird hier aber als vereinfachte Notation für die präzise Auszeichnung des Sachverhalts benutzt, daß ein Prädikatssymbol von *genau* einem Argument erfüllt wird. Andernfalls müßte zu einer recht komplizierten Formulierung auf der Basis von Existenz- und Allquantor gegriffen werden. Denn der Einsquantor ist definiert durch:

$$\underline{\exists}(X)\text{: prä}(\ldots,X,\ldots)$$
$$:\Leftrightarrow \quad (\exists(X)\text{: prä}(\ldots,X,\ldots)) \wedge (\forall(Y)\text{: }(Y \neq X \rightarrow \neg\text{prä}(\ldots,Y,\ldots)))$$

Vgl. zu dieser Definition den Ansatz bei Whitehead (1925), S. 31 (dritte Alternative).

21) Quantoren lassen sich als abstrakte Verallgemeinerungen von gleichartig verknüpften prädikatenlogischen Formeln auffassen, die jeweils aus denselben Prädikatssymbolen formiert worden sind. Dabei werden alle Konstanten, die in den Formeln nicht identisch enthalten sind, jeweils durch eine Variable ersetzt. All-, Existenz- und Einsquantor generalisieren solche Formeln, die untereinander konjunktiv, adjunktiv bzw. disjunktiv verknüpft sind. Vgl. dazu die gleiche Ausdeutung von All- und Existenzquantoren bei LORENZEN,P. (1962), S. 20; ESSER,H. (1977a), S. 44.

22) Die vielfach übliche Schreibweise "$(X \in DB_X)$:" der Quantorargumente kann strenggenommen erst im Rahmen einer semantischen Interpretation des prädikatenlogischen Kalküls erfolgen. Denn erst dort werden den Variablen Definitionsbereiche DB_X zugeordnet.
Unter Umständen soll jedoch nicht die Interpretation einer Variablen durch ihren Definitionsbereich DB_X ausgedrückt werden, sondern die interpretationsunabhängige Zugehörigkeit der Variablen X zu einer beliebigen Menge M: "$X \in M$". Dann lassen sich mit p(X) als einer beliebigen, von der Variablen X abhängigen Formel für originär definierte All-, Existenz- und Einsquantifikationen folgende äquivalente Kurznotationen einführen:

$$\forall(X \in M)\text{: } p(X) \quad :\Leftrightarrow \quad \forall(X)\text{: } X \in M \rightarrow p(X)$$
$$\exists(X \in M)\text{: } p(X) \quad :\Leftrightarrow \quad \exists(X)\text{: } X \in M \wedge p(X)$$
$$\underline{\exists}(X \in M)\text{: } p(X) \quad :\Leftrightarrow \quad \underline{\exists}(X)\text{: } X \in M \wedge p(X)$$

Vgl. zu diesen Schreibweisen - bezüglich All- und Existenzquantifikationen - DIN 5474 (1973), S. 3; LUXEMBURG (1973), S. 42 (ohne die Kurznotation). Die derart definitorisch eingeführte Kurzschreibweise "$X \in M$" in Quantorargumenten wird in dieser Arbeit im Regelfall benutzt. Sie entspricht auch einer weiteren Darstellungsweise von All- und Existenzaussagen durch Sub- bzw. Konjugate. Hierbei drückt jeweils eine Teilformel m(X) die Zugehörigkeit der Variable X zur Menge M aus:

$$\forall(X)\text{: } m(X) \rightarrow p(X)$$
$$\exists(X)\text{: } m(X) \wedge p(X)$$

Vgl. zur letzten Notationsvariante STEGMÜLLER (1983), S. 53f.; BUCHER (1987), S. 168.

23) Die Charakterisierung "*potentiell* unendlich" drückt aus, daß in konkreten Explizierungen des prädikatenlogischen Kalküls oftmals nur endliche Mengen deskriptiver Zeichen verwendet werden. Da diese Mengen aber beliebig formuliert werden dürfen, ist die Vereinigungsmenge aller denkmöglichen konkreten Zeichenmengen unendlich. Es wird auch kurz von einer unendlichen Menge deskriptiver Zeichen gesprochen.

24) Eine Menge heißt abzählbar-unendlich, wenn sie die Mächtigkeit der Menge aller natürlichen Zahlen besitzt; vgl. STEGMÜLLER (1984b), S. 40. In dieser Arbeit werden unendliche Mengen ohne nähere Qualifizierung stets als abzählbar-unendliche Mengen mit der Kardinalzahl "Aleph-Null" unterstellt. Mengen, die - wie z.B. die überabzählbare Menge der reellen Zahlen - mächtiger als die Menge aller natürlichen Zahlen sind, spielen in dieser Arbeit keine Rolle. Vgl. zu einer näheren Darstellung der unterschiedlichen Mächtigkeiten jeweils unendlicher Mengen und ihrer transfiniten Kardinalzahlen GENTZEN (1938), S. 38f.; CARNAP (1960a), S. 150ff.; MESCHKOWSKI (1967), S. 26ff. u. 70ff.; DAUBEN (1983), S. 117 u. 119ff.; STEGMÜLLER (1984b), S. 40ff.

25) Vgl. MENDELSON (1964), S. 56f.; SMULLYAN (1968), S. 43; POTTHOFF (1981), S. 5; DELAHAYE (1987), S. 66.
Darüber hinaus läßt POTTHOFF (1981), S. 5, sogar überabzählbar-unendliche Individuenmengen zu. Infolge der früheren Anmerkungen zum logischen Konstruktivismus, zu seiner Ablehnung aktuell unendlicher Größen und zu den mathematischen Schwierigkeiten mit dem reellzahligen Kontinuum wird dieser Auffassung POTTHOFF's hier aber nicht gefolgt.

26) Vgl. zum prädikatenlogischen Individuenbegriff WHITEHEAD (1925), S. 51; CARNAP (1960a), S. 4 u. 143ff.
Auf diesen prädikatenlogischen Individuenbegriff beziehen sich im allgemeinen auch Beiträge zum Konzept der Prädikat/Transition-Netze.

27) Daher werden Konstantensymbole und Variablen auch als Individuenkonstanten bzw. -variablen bezeichnet; vgl. z.B. MENDELSON (1964), S. 46; SMULLYAN (1968), S. 43; CHANG,C.L. (1973), S. 27; OPP,K. (1976), S. 27 u. 30f.

28) Als Synonym für Konstantensymbole wird mitunter auch der Begriff "Parameter" gebraucht; vgl. z.B. SMULLYAN (1968), S. 43.
Die Notation von Konstantensymbolen lehnt sich an die Konventionen an, die früher für das Signaturkonzept eingeführt wurden. Daher werden Konstantensymbole stets durch einen Kleinbuchstaben eingeleitet oder durch genau einen Kleinbuchstaben bezeichnet.

29) Eine der seltenen Ausnahmen, die zwischen Konstantensymbolen und Konstanten deutlich differenzieren, stellt SMULLYAN (1968), S. 46f. u. 50, dar (der allerdings Konstantensymbole als Parameter anspricht).

30) Vgl. zu einer ausführlichen Diskussion des prädikatenlogischen Variablenbegriffs WHITEHEAD (1925), S. 4f.
Für die Variablennotation wird wiederum auf die Konventionen zurückgegriffen, die im Zusammenhang mit dem Signaturkonzept eingeführt wurden. Variablen beginnen daher hier immer mit einem Großbuchstaben, oder sie bestehen nur aus genau einem Großbuchstaben. Damit wird bewußt von der prädikatenlogisch üblichen Vereinbarung abgewichen, Variablen mit Kleinbuchstaben zu notieren; vgl. z.B. WHITEHEAD (1925), S. 5. Diese Devianz erfolgt mit Rücksicht auf die spätere prädikatenlogische Implementierung durch die Programmiersprache PROLOG.
Mitunter werden Variablen auch als Variablensymbole angesprochen; vgl. CHANG,C.L. (1973), S. 27; ITZINGER (1976), S. 29. Trotz der partiellen Äquivokation hinsichtlich des Begriffs "Konstantensymbole" folgt der Verf. dieser Bezeichnungsweise nicht. Denn es sind in der Prädikatenlogik keine Variablen vorgesehen, die diesen Variablensymbolen zugeordnet werden könnten. Dagegen werden die Konstantensymbole später auf Konstanten abgebildet. Daher liegt zwischen Konstantensymbolen und Variablen keine Symmetrie vor, welche die o.a. Bezeichnungsweise "Variablensymbole" sachlich rechtfertigen könnte.

31) Vgl. PROLOG (o.J.), S. 22f.; CORDES (1988), S. 18f.; KINNEBROCK (1988), S. 28 u. 32.

32) Die Formulierungsvorzüge bestehen zunächst darin, daß die anonyme Variable die explizite Einführung derjenigen konventionellen Variablen erspart, die sie ersetzt. Dies wirkt sich insbesondere dann aus, wenn zahlreiche konventionelle Variablen durch nur die eine anonyme Variable substituiert werden. Die Menge aller explizit definierten Variablen kann dadurch beträchtlich schrumpfen. Dies ist auch unter dem Aspekt der Implementierung von prädikatenlogischen Modellierungen auf Automatischen Informationsverarbeitungssystemen von Interesse, da dort jede explizite Variablendefinition - im Gegensatz zur Verwendung der anonymen Variable - spezielle Ressourcen bindet.
Darüber hinaus besitzt die anonyme Variable bei der Modellformulierung den Charakter eines Irrelevanzanzeigers, der sich vor allem bei der Formulierung von Entscheidungstabellen bewährt hat. Der Vorteil eines solchen Irrelevanzanzeigers liegt darin, dem Modellbenutzer unmittelbar vor Augen zu führen, daß die anonymisierte Variable im jeweils betrachteten Modellausschnitt keine Bedeutung besitzt. Dies unterstützt auf formaler Ebene eine Fokussierung auf die jeweils relevanten Modellaspekte, ohne daß die Vollständigkeit der Formeldefinitionen verloren ginge.
Schließlich besitzt die anonyme Variable auch den Vorzug, die Auswertung von prädikatenlogischen Modellen durch Inferenzprozeduren in der Programmiersprache PROLOG effizienter ablaufen zu lassen. Denn bei den Unifizierungsschritten des hier vorausgesetzten prädikatenlogischen Inferenzkonzepts bleibt die anonyme Variable von vornherein ausgeklammert. Dies erspart abermals erheblichen Ressourceneinsatz, der andernfalls erforderlich wäre, um in jedem Unifizierungsschritt zu versuchen, die konventionellen Variablen durch Konstanten zu binden.

33) In der prädikatenlogischen Literatur wird zumeist nur von Funktionen gesprochen. Dies führt aber zu Schwierigkeiten, weil die Funktionssymbole erst später im Rahmen der prädikatenlogischen Semantik auf Funktionen abgebildet werden. Zu den selteneren Fällen einer expliziten Bezugnahme auf Funktions*symbole* gehören z.B. CHANG,C.L. (1973), S. 27; ITZINGER (1976), S. 29; PREIß (1989), S. 14.
Funktionen werden in dieser Arbeit mit Kleinbuchstaben notiert. Dies entspricht vorherrschender prädikatenlogischer Konvention; vgl. z.B. WHITEHEAD (1925), S. 5; STEGMÜLLER (1984b), S. 38. Funktionssymbole werden dadurch unterschieden, daß sie mit einem Großbuchstaben beginnen müssen.
Alle hier betrachteten Funktionssymbole und Funktionen besitzen die Qualität von Funktionssymbolkonstanten bzw. Funktionskonstanten. Funktions(symbol)variablen werden grundsätzlich nicht betrachtet. Sie würden zu einer Prädikatenlogik 2. Stufe führen.

34) Diese Konkretisierung erfolgt erst später im Rahmen der prädikatenlogischen Semantik.

35) Daher entspricht z.B. der Funktor "Fun(...)" eines Prädikatssymbols "Fun" - bis auf die Unterscheidung zwischen Groß- und Kleinschreibung - genau dem Funktor "fun(...)" jener Funktion "fun", die in einer prädikatenlogischen Semantik dem Funktionssymbol "Fun" zugeordnet wird.

36) Vgl. LOVELAND (1978), S. 10; REINFRANK (1985b), S. 27; DELAHAYE (1987), S. 67.
0-stellige Funktions(symbol)variablen, die in einer früheren Anmerkung ausgegrenzt wurden, stimmen dagegen mit Variablen überein; vgl. REINFRANK (1985b), S. 27.

37) Vgl. MENDELSON (1964), S. 46; CHANG,C.L. (1973), S. 28f.; DIN 5474 (1973), S. 5; ITZINGER (1976), S. 29f.; LOVELAND (1978), S. 10; DELAHAYE (1987), S. 67; MURATA,TA. (1988b), S. 482; PREIß (1989), S. 14.

38) Trotz dieser Exklusionsbedingung werden durch die funktionsgestützte Terminduktion alle Terme zugelassen, die bereits im Rahmen des Signaturkonzepts als Elemente einer SIG-Termmenge eingeführt worden sind. Beispielsweise sind Grundterme auch im prädikatenlogischen Kalkül als variablenfreie Terme definiert.

39) Prädikatssymbole werden der Einfachheit halber auch als Prädikate bezeichnet, wenn aus dem Argumentationskontext ersichtlich ist, daß syntaktisch definierte Prädikatssymbole gemeint sind. Nähere Erläuterungen zu Prädikatssymbolen finden sich bei CARNAP (1960a), S. 4ff.; ERNST (1969), S. 164f.; CHARNIAK (1976a), S. 4f.; NILSSON,N. (1980a), S. 132ff.; STEGMÜLLER (1983), S. 41.
MENDELSON (1964), S. 46, benutzt den Begriff der Prädikatsbuchstaben (predicate letters). WHITEHEAD und RUSSELL bezeichnen in ihrem prädikatenlogischen Basiswerk "Principia Mathematica" Prädikatssymbole als Aussagefunktionen (propositional functions); vgl. WHITEHEAD (1925), passim, insbesondere S. 14f., 38 u. 95. Den Prädikatsbegriff führen sie auf S. 51 explizit als spezielle Bezeichnung für Aussagefunktionen ein, die der 1. Sprachstufe angehören. Genau das ist für die hier betrachtete Prädikatenlogik 1. Stufe der Fall.
Der Verf. hält den Begriff "Aussage*funktionen*" für unglücklich, weil hier der Prädikatsbegriff mit dem Funktionsbegriff vermengt wird. Besonders deutlich wird diese terminologische Konfundierung bei WHITEHEAD (1925), S. 39, wo prädikatenlogische Formeln zwecks terminologischer Vereinfachung generell mit Funktionen gleichgesetzt werden. Aus der eingangs vorgestellten Definition des prädikatenlogischen Kalküls folgt aber, daß Prädikats- und Funktionssymbole kategorial verschieden sind. Denn Funktionen sind am Aufbau der Argumente von Prädikaten beteiligt. Dagegen sind Prädikate *über* diesen Argumenten definiert. (Die Möglichkeit, Prädikate selbst in den Argumenten von Prädikaten zuzulassen, steht im Rahmen der Prädikatenlogik 1. Stufe nicht offen.) Die mangelhafte Trennung zwischen Prädikats- und Funktionsbegriff ist die Quelle mancher prädikatenlogischer Ungereimtheiten, auf die in diesem Kapitel mehrfach hingewiesen wird. Abweichender Ansicht ist lediglich OBERSCHELP (1962), S. 298, insofern, als er die Umschreibungsmöglichkeit von Funktionssymbolen durch Prädikatssymbole unterstellt.

40) Daher liegt es nahe, Prädikatssymbole als Prädikatskonstantensymbole zu bezeichnen. Dabei entspräche die Beziehung zwischen Prädikatskonstantensymbolen und Prädikatskonstanten genau der oben angedeuteten Beziehung zwischen (Individuen-)Konstantensymbolen und (Individuen-)Konstanten. Denn in beiden Fällen gehören die symbolischen Prädikats- und Individuenkonstanten zur syntaktischen Kalkülebene, während sie im Rahmen einer prädikatenlogischen Semantik durch Prädikatskonstanten bzw. Individuenkonstanten interpretiert werden. Allerdings taucht der Begriff "Prädikatskonstantensymbol" in der einschlägigen Literatur nach Kenntnis des Verf. nicht auf.

41) Vgl. OPP,K. (1976), S. 30f., der in diesem Zusammenhang von Satzformeln oder Aussageformen (S. 31) spricht.

42) Vgl. aber den Hinweis auf Prädikatsvariablen in Prädikatenlogiken höherer Stufen.

43) Gleiches gilt für Funktionssymbole, deren Deklarationen ebenfalls intern strukturiert sind.

44) Strenggenommen müßte hier von Prädikatssymbolnamen gesprochen werden. Der Einfachheit halber wird jedoch auch die Bezeichnung "Prädikatsname" zugelassen, wenn sich aus dem Kontext ersehen läßt, daß der Name eines Prädikatssymbols gemeint ist. Gleiches gilt für alle anderen hier verwandten Bezugnahmen auf Prädikatssymbole.

45) Dies folgt aus der Eindeutigkeit der Benennung aller Prädikatssymbole, die in dieser Arbeit vorausgesetzt wird. Prädikatssymbole werden hier jeweils durch einen Großbuchstaben eingeleitet. Dies ist auch oftmals im Rahmen konventioneller prädikatenlogischer Darstellungen der Fall; vgl. z.B. MENDELSON (1964), S. 46; CARNAP (1968), S. 15; OPP,K. (1976), S. 27; BUCHER (1987), S. 162; COHN (1987), S. 185.

46) Der prädikatsbezogene Stellenbegriff besitzt keine inhaltliche Überschneidung mit dem äquivoken netzbezogenen Stellenbegriff.

47) Daher ist es zulässig, darauf zu verzichten, die Stellen aus dem Argument eines Prädikatssymbols Prä(st_1,...,st_K) explizit anzuführen; vgl. OBERSCHELP (1962), S. 325. In diesem Fall kann das Prädikatssymbol verkürzt als "Prä(...)" notiert werden. In Anlehnung an die Sprachregelungen für Operationssymbole (Funktionssymbole) wird dann von einem Prädikator Prä(...) gesprochen.
Mitunter werden der Prädikatsname "Prä" oder der Prädikator Prä(...) als Funktor bezeichnet; vgl. z.B. CORDES (1988), S. 14. Ebenso werden manchmal Prädikate wie Funktionen behandelt vice versa; vgl. BARTH,G. (1987), S. 223. Der Verf. folgt diesen Auffassungen jedoch nicht, weil Prädikats- und Funktionssymbole kategorial verschieden sind.

Ein Prädikatssymbol Prä(st_1,...,st_K) wird dagegen im Rahmen der logischen Programmierung, mitunter auch seitens der Prädikatenlogik erst durch seinen Namen "Prä" *und* die Anzahl K seiner Stellen st_k eindeutig definiert; vgl. MENDELSON (1964), S. 46; CORDES (1988), S. 14. Daher wäre es grundsätzlich zulässig, die Verschiedenheit zweier Prädikatssymbole *nur* durch deren unterschiedliche Stellenanzahl auszudrücken. Demnach wären die beiden Prädikatssymbole Prä(st_1,...,$st_{K.1}$) und Prä(st_1,...,$st_{K.2}$) mit K_1 K_2 trotz identischer Prädikatsnamen "Prä" voneinander verschieden. Dann wäre die oben eingeführte Prädikator-Schreibweise Prä(...) aber nicht mehr eindeutig. Um auf die Stellenanzahl als Prädikatsidentifikator verzichten zu können, wird in dieser Arbeit daher die strengere Anforderung vereinbart, daß der Prädikatsname allein ausreicht, um ein Prädikatssymbol als solches zu identifizieren.
Erstens entspricht dies dem "intuitiven" Verständnis von Eigennamen. Beispielsweise kann die eindeutige Prädikatsidentifizierung durch paarweise verschiedene Indizierungen der Prädikatsnamen geschehen. Im o.a. Beispiel wären etwa die beiden Prädikatssymbole - unabhängig von ihren Stellenanzahlen - bereits durch ihre Namen "Prä$_1$" und "Prä$_2$" wohlunterschieden. Die Stellenanzahl K eines Prädikatssymbols Prä(st_1,...,st_K) besitzt daher nicht mehr identifizierende, sondern nur noch informative Funktion.
Zweitens ist es sogar notwendig, auf die Identifikationsfunktion der Stellenanzahl zu verzichten, wenn später die konventionelle Prädikatenlogik durch Überlagerung einer Sortenstruktur zu einer sortierten Prädikatenlogik erweitert wird. Denn im Rahmen des Signaturkonzepts wurden multiple Sortendefinitionen erlaubt. Diese Multiplizität schließt den Fall ein, daß dieselbe Sorte als Zielsorte von mehreren Operationssymbolen definiert wird, die sich durch die Anzahlen ihrer Argumentstellen voneinander unterscheiden. Vgl. dazu vor allem das Beispiel mit der mehrfach definierten Datumsangabe. Falls eine derart multipel definierte Sorte im Argument eines Prädikatssymbols vorkommt, muß für dieses Prädikatssymbol eine entsprechende Variabilität seiner Stellenanzahl zugelassen werden. Diese Variabilität der Stellenanzahl für die Prädikatssymbole - und ebenso der Funktionssymbole - aus einer sortierten Prädikatenlogik wird von OBERSCHELP (1962), S. 325f., verdeutlicht.

48) Eigenschafts- und Attributsbegriff werden in dieser Arbeit synonym verwendet. Abweichender Ansicht ist CARNAP (1960a), S. 5, der auch für Relationen den Attributbegriff verwendet (mehrstellige Attribute).

49) Falls von einem prädikatenlogischen Kalkül nur 1-stellige Prädikatssymbole zugelassen werden, liegt eine monadische Prädikatenlogik oder Klassenlogik vor. Die letztgenannte Bezeichnungsweise rührt daher, daß alle formalen Objekte, die zulässige Argumente eines monadischen Prädikatssymbols darstellen, per constructionem das prädikatspezifische Attribut besitzen. Die Zusammenfassung solcher attributgleichen Objekte wird als Klasse bezeichnet.

50) Auf die Darstellung von Relationen durch Prädikate wird zurückgekommen. Dies setzt allerdings eine präzise Definition der Bedeutung von Prädikatssymbolen voraus. Näheres dazu unter dem Aspekt prädikatenlogischer Semantiken.
Daneben können mehrstellige Prädikatssymbole auch verwendet werden, um einem Objekt (erste Prädikatsstelle) eine oder mehrere Attribute (folgende Prädikatsstellen) zuzuordnen. Die Kombinationen aus Objekten und Attributen können in einem weit gefaßten Sinn auch als - heterogen strukturierte - Relationen betrachtet werden.

51) Wenn ein prädikatenlogischer Kalkül auch mehrstellige Prädikatssymbole zuläßt, wird von einer polyadischen Prädikatenlogik oder Relationenlogik gesprochen.

52) Vgl. zur Verwendung 0-stelliger Prädikatssymbole z.B. LOVELAND (1978), S. 10; REITER (1984), S. 194; REINFRANK (1985b), S. 27; DELAHAYE (1987), S. 66 u. 68.

53) Gültigkeit und Wahrheit einer 0-stelligen prädikatenlogischen Formel werden synonym verwandet. In prädikatenlogischen Kontexten überwiegt die erste Ausdrucksweise, in aussagenlogischen die zweite.

54) Ungültigkeit und Falschheit einer 0-stelligen prädikatenlogischen Formel werden als Synonyma behandelt.

55) Vgl. z.B. PREIß (1989), S. 14.
SMULLYAN (1968), S. 43; CHANG,C.L. (1973), S. 29; ITZINGER (1976), S. 30, und POTTHOFF (1981), S. 6, sprechen zwar von einer rekursiven Definition prädikatenlogischer Formeln. Aber es wurde bereits dargelegt, daß sich jede induktive Definition ebenso als rekursiv auffassen läßt.

56) Die Atomarität prädikatenlogischer Formeln heißt keineswegs, daß sie überhaupt keine innere Struktur besäßen. Vielmehr weisen prädikatenlogische Formeln immer eine interne Struktur auf, die sich aus den Prädikatsargumenten konstituiert. Dies gilt zumindest, solange die Formeln nicht aus 0-stelligen Prädikatssymbolen abgeleitet werden. Die Atomarität prädikatenlogischer Formeln drückt lediglich aus, daß diese Formeln nicht durch Konnektoren oder Operatoren aus anderen Formeln zusammengesetzt sind.

57) Darüber hinaus können diese Kombinationen auch Hilfszeichen umfassen.

58) Vgl. zu dem Konzept, alle Formeln als wohlgeformt zu betrachten, die aus einem Kalkül induktiv hergeleitet werden können, CHURCH (1932), S. 350 u. 352; MENDELSON (1964), S. 46; LUXEMBURG (1973), S. 41.
Alle anderen Formeln sind unzulässig. Die Formelunzulässigkeit kann aus drei - unter Umständen miteinander kombinierten - Quellen resultieren. Erstens ist es möglich, daß die Formeln unzulässige Zeichen enthalten. Zweitens

lassen sich Formeln vorstellen, die auf prädikatenlogisch unzulässigen Formierungsregeln beruhen. Drittens könnten Formeln konzipiert werden, die sich nur aus der unendlich oft wiederholten Anwendung zulässiger Formierungsregeln gewinnen lassen.

59) Vgl. z.B. CHANG,C.L. (1973), S. 29; ITZINGER (1976), S. 30. Anstatt diese Exklusionsregel explizit zu formulieren, kann die Menge wohlgeformter Formeln auch so definiert werden, daß sie die *kleinste* Menge von Formeln ist, die sich durch endlich viele Anwendungen der drei Formierungsregeln induktiv erzeugen lassen; vgl. DELAHAYE (1987), S. 68.

60) Diese Notation lehnt sich an die Syntax der Programmiersprache Turbo-PROLOG an, die für die spätere Implementierung prädikatenlogischer Netzmodelle verwendet wird. Es ist zwar nur erforderlich, prädikatenlogische Formeln mit Kleinbuchstaben *beginnen* zu lassen. Aber aus formalästhetischen Erwägungen werden sie hier durchgehend mit Kleinbuchstaben dargestellt. Vgl. zur Verwendung klein notierter Formelbezeichnungen z.B. auch WHITEHEAD (1925), S. 5.

61) Auf den semantisch begründeten Ausdrucksreichtum der Prädikatenlogik wird an anderer Stelle näher eingegangen.

62) LUXEMBURG (1973), S. 41, reserviert diesen Fall für den Prädikatsbegriff.

63) Vgl. CARNAP (1960a), S. 84; LUXEMBURG (1973), S. 41.

64) Per definitionem umfassen geschlossene Formeln auch alle Grundtermformeln prä($gt_1,...,gt_K$). Denn auch solche Formeln, die überhaupt keine Variablen enthalten, besitzen keine freien Variablen. In allen anderen Formeln, die mindestens eine Variable in ihren Argumenten umfassen, müssen alle Variablen jeweils durch einen Quantor gebunden sein, damit eine geschlossene Formel vorliegt.

65) Denn nur den geschlossenen Formeln können später in einer prädikatenlogischen Semantik eindeutige Wahrheitswerte zugeordnet werden.
Trotz dieser Aussagenartigkeit stellen die prädikatenlogischen Sätze im allgemeinen keine Aussagen im Sinne der Aussagenlogik dar, sofern es sich um mindestens 1-stellige Prädikatssymbole handelt. Denn die prädikatenlogischen Sätze enthalten immer noch eine interne Argumentstruktur, die den aussagenlogischen Formeln vollständig fehlt. Dies übersieht OPP,K. (1976), S. 24, wenn er Aussagen und wahrheitsdefinite prädikatenlogische Sätze miteinander identifiziert. Nur in dem Sonderfall 0-stelliger Prädikatssymbole stellen geschlossene Formeln echte aussagenlogische Formeln dar.
Eine vollkommen anders gelagerte Differenzierung zwischen Aussagen- und Satzbegriff als die oben dargelegte findet sich bei STEGMÜLLER (1968), S. 17ff.

66) Dennoch handelt es sich bei diesen "aussagenlogischen" Formeln prä() weiterhin um prädikatenlogische Formeln, weil sie innerhalb des prädikatenlogischen Kalküls definiert sind. Insofern handelt es sich bei den Formelattribut "aussagenlogisch" nicht um ein konträres, sondern um ein spezifizierendes Attribut.

67) Entweder liegt ein Prädikatssymbol vor, in dem alle Stellen Platzhalter st_k darstellen, oder aber eine atomare prädikatenlogische Formel, in der alle Stellen durch entsprechende Terme te_k eingenommen worden sind. Hybride Argumente aus Platzhaltern und Termen werden also grundsätzlich ausgeschlossen.

68) Dabei besitzt die Notationsweise prä($...,X_k,...$) nur verdeutlichenden Charakter. Denn der zugehörige Term te_k im Formelargument ($te_1,...,te_K$) kann zusammengesetzt sein. Dann befindet sich an der k-ten Stelle des Formelarguments auf der obersten Stufe ein Funktionssymbol, das erst auf der hierarchisch tiefsten Zusammensetzungsstufe die Variable X_k einschließt: ($te_1,...,te_k,...,te_K$) mit $te_k = fun_k(...,X_k,...)$; vgl. dazu auch die Anmerkung zu tief strukturierten Formelargumenten in Erläuterung v). Von dieser Komplizierung durch mindestens eins intermediäres Funktionssymbol wird in der oben eingeführten Notation abstrahiert.

69) Hier sind die Konstantensymbole Ko_k keine Endglieder einer Zusammensetzung aus Funktoren. Vielmehr stellen sie unmittelbar definierte konstante Basisterme dar.

70) Hiervon deutlich zu unterscheiden sind die interpretierten Formeln. Sie werden später durch das Konzept prädikatenlogischer Semantiken eingeführt. Dort werden uninterpretierte Formeln prä$_j$($te_1,...,te_{Kj}$) durch Interpretationen ($ob_1,...,ob_{Kj}$) ihrer Argumente auf interpretierte Formeln prä$_j$($ob_1,...,ob_{Kj}$) abgebildet. Die hier entfaltete Terminologie für uninterpretierte Formeln gilt ebenso für diese interpretierten Formeln.

71) Die Prädikatsvorkommnisse können sich jeweils paarweise durch die Art unterscheiden, in der die Platzhalter st_k des Prädikatssymbols Prä($st_1,...,st_K$) durch Konstantensymbole, Variablen und Funktionssymbole ersetzt worden sind. Falls kein solcher Unterschied vorliegt, handelt es sich um identische atomare Formeln. Sie stellen identische Kopien desselben zugrundeliegenden Prädikatssymbols dar.

72) Aus dieser Perspektive stellt ein K-stelliges Prädikatssymbol ein abstraktes Schema Prä($st_1,...,st_K$) dar, das alle konkreten atomaren Formeln prä($te_1,...,te_K$) umfaßt, die aus ihm mittels der Formierungsregel FR_P hervorgebracht

werden können. In diesem Schema erfolgt noch keine Festlegung hinsichtlich derjenigen Terme te_k, die in die Stellen st_k des Prädikatsarguments in einer konkreten Formel eingesetzt werden.
Diese Sichtweise teilt auch die logische Programmierung - zumindest im Kontext der Programmiersprache PROLOG. Dort wird ein "Prädikat" als eine Menge von Formeln gleichen Namens und gleicher Stelligkeit aufgefaßt; vgl. z.B. CORDES (1988), S. 14 i.V.m. S. 11ff. Dabei ist mit einem Prädikat strenggenommen ein Prädikatssymbol gemeint. Auf die Prädikatsstelligkeit als Prädikatsidentifikator wurde in dieser Arbeit verzichtet.

73) 1-stellige Formeln heißen auch unäre oder monadische Formeln.

74) Vgl. dazu die Quellen, die zu Prädikat/Transition-Netzen angeführt wurden.

75) Eine Ausnahme stellen die Informationsprädikate dar, auf die in der nachfolgenden Anmerkung eingegangen wird.

76) Dann wird auch von einer "Verflachung" der ehemals tief strukturierten Prädikatsargumente gesprochen.

77) Vgl. etwa das 2-stellige Prädikatssymbol "Kunde_mit_vertrag" bei der Modellierung des Mietwagenexempels.

78) Strenggenommen müßte von stellenspezifischen quantorfreien atomaren prädikatenlogischen Formeln gesprochen werden. Zwecks terminologischer Vereinfachung wird aber die Ausdrucksweise "Stellenprädikate" gewählt. Gleiches gilt für die nachfolgenden Transitionen- und Netzprädikate.

79) Vgl. dazu die Ausführungen zur Konstruktion deklarativer Objektmodelle.

80) Eine Formel liegt in konjunktiver Normalform genau dann vor, wenn gilt: Alle Variablen sind durch Allquantoren, aber durch keinen Existenzquantor gebunden. Die Formel ist eine endliche Konjunktion von paarweise verschiedenen Klauseln oder genau eine Klausel. Eine Formel in konjunktiver Normalform entspricht daher einer nichtleeren Klauselmenge, deren Elemente implizit als konjunktiv verknüpft gelten. Eine Klausel ist eine endliche Adjunktion aus paarweise verschiedenen Literalen oder genau ein Literal. Ein Literal ist entweder eine atomare Formel oder deren Negat. Auf Klauseln und konjunktive Normalformen wird an anderer Stelle ausführlicher zurückgekommen. Vgl. zu diesen prädikatenlogischen Ausdrucksformen auch CHANG,C.L. (1973), S. 13 u. 47f.; ITZINGER (1976), S. 28; BUCHER (1987), S. 114ff.; CORDES (1988), S. 31f.; ZELEWSKI (1989c), S. 20f.; DELAHAYE (1987), S. 56f.

81) Nachfolgende Anmerkungen gelten sowohl für Stellen- als auch für Transitionenprädikate.

82) Zwar wird in prädikatenlogischen Kalkülen die induktive Termzusammensetzung mit der Hilfe von Funktionssymbolen oftmals vorgesehen. Trotzdem wird in der Regel die tiefe Strukturierungsmöglichkeit, mit der Hilfe von Funktionssymbolen zusammengesetzte Terme zu bilden, nur als theoretische Option anläßlich der allgemeinen Kalküldefinition eingeräumt. In den anschließenden praktischen Beispielen werden dann doch wieder Prädikatssymbole mit flachen Argumenten verwendet. (Darüber hinaus wird dort oftmals auf Konstanten anstelle von Konstantensymbolen und auf Funktionen anstatt von Funktionssymbolen als Termkonstituenten Bezug genommen.)

83) Besonders deutlich wird dies bei REITER (1984), S. 194. Zwar faßt er die Konstanten(symbole) und Variablen, auf die sich Prädikatsargumente erstrecken dürfen, unter dem Termbegriff zusammen. Dennoch nutzet er die funktionsgestützte Terminduktion in keiner Weise aus. DORN (1989), S. 11, spricht explizit von einer flachen Struktur der prädikatenlogischen Wissensrepräsentation. MURATA,TA. (1988b), S. 484f. u. 496, läßt klar erkennen, daß in Prädikat/Transition-Netzen derzeit nur Konstanten und Variablen als Komponenten von Prädikatsargumenten berücksichtigt werden.
Die weitgehende Vernachlässigung von prädikatenlogischen Formeln, deren Argumente auch Funktionen enthalten, geht vermutlich auf den mißverständlichen Ansatz von WHITEHEAD und RUSSELL zurück, Prädikate als Aussagefunktionen zu bezeichnen. Denn das Verbot der Prädikatenlogik 1. Stufe, in den Argumenten von Prädikaten selbst wieder Prädikate zu verwenden, verleitet dann angesichts dieser funktionsbezogenen Redeweise zu dem Fehlschluß, die Argumente von prädikatenlogischen Formeln 1. Stufe dürften *überhaupt keine* Funktionen umfassen; vgl. etwa WHITEHEAD (1925), S. 51ff. Hier liegt jedoch eine unzulässige Generalisierung vor. Denn der Ausschluß gilt strenggenommen nur für *Aussagefunktionen* im Sinne von prädikatenlogischen Formeln, nicht aber für Funktionen schlechthin.

84) Zu den seltenen Ausnahmen, die Beispiele für Prädikate mit tief strukturierten Argumenten explizit nennen, gehört GÖDEL (1931), S. 194, Fn. 58.

85) Dies gilt allerdings nur unter dem Vorbehalt, daß nicht die strukturlose Basismarke vorliegt. Näheres zum Markenaufbau an späterer Stelle.

86) Anscheinend wird unreflektiert eine einfache 1:1-Beziehung unterstellt.

87) Vgl. dazu die multiple Definition der Sorte "zeitpunkt" im später folgenden Beispiel.

88) Vgl. z.B. LOVELAND (1978), S. 10; PREIß (1989), S. 13f.

89) Dies gilt nur, wenn das Signaturkonzept - wie in dieser Arbeit - auf rein algebraische Weise definiert wird. Statt dessen ist es aber auch möglich, das Signaturkonzept direkt um prädikatenlogische Aspekte zu erweitern. Diesen Weg beschreitet z.B. HOFBAUER (1989), S. 2ff. Dort werden Prädikatssymbole von vornherein gleichberechtigt neben den Operationssymbolen eingeführt (S. 2). Dann können die Operations- ausschließlich mit Funktionssymbolen identifiziert werden, so daß diese beiden von den Prädikatssymbolen wohlunterschieden sind. Aber auch auf diese Weise wird die nachfolgend thematisierte kategoriale Verschiedenheit von Funktions- und Operationssymbolen nicht widerlegt, sondern indirekt bestätigt.

90) Die Notationen und Sprachregelungen für Operationssymbole "Op", die bereits eingeführt wurden, werden für Funktionssymbole "Fun" unverändert übernommen. Die Kleinschreibung "fun" bleibt im Interesse formaler Konsistenz der späteren Einführung von Funktionen vorbehalten, die den Operationen "op" des Signaturkonzepts entsprechen.
Demgegenüber werden Funktionssymbole in prädikatenlogischen Kontexten oftmals von Funktionen nicht unterschieden und von vornherein mit Kleinbuchstaben bezeichnet. Dies entspricht der üblichen prädikatenlogischen Notation von Funktionssymbolen; vgl. z.B. MENDELSON (1964), S. 46; CARNAP (1968), S. 15; LOVELAND (1978), S. 10; COHN (1987), S. 185.

91) Die Deutung von Operationssymbolen als Prädikatssymbole führt allerdings zu speziellen Schwierigkeiten. Sie liegen einerseits darin, daß Operationssymbole aus Argumenten *und* den Bildern ihrer Argumente bestehen, während Prädikatssymbole nur für ihre Argumente definiert sind. Diese Lücke wird später dadurch geschlossen, daß die Operationssymbole zunächst als Wahrheitsfunktionssymbole gedeutet werden, denen sich dann prädikatenlogische Prädikatssymbole eindeutig zuordnen lassen. Andererseits verleitet die gemeinsame Rückführung von Prädikats- und Funktionssymbolen auf Operationssymbole dazu, auch Prädikats- und Funktionssymbole untereinander gleichzusetzen. Dies wird an anderer Stelle näher belegt. Dennoch handelt es sich - wie nachfolgend erläutert - aus prädikatenlogischer Sicht um kategorial verschiedene Konstrukte.

92) Diese semantische Verschiedenheit äußert sich u.a. auch darin, daß bei der Deklaration von Operationssymbolen, die Prädikatssymbole darstellen, auf die Explizierung ihrer Zielsorte verzichtet wird. Denn alle diese Operationssymbole verfügen über dieselbe Zielsorte "wahr_wert" für die BOOL'schen Wahrheitswerte. Bei allen anderen Operationssymbolen, die Funktionssymbolen entsprechen, muß dagegen die Zielsorte explizit angegeben werden, da sie von Funktionssymbol zu Funktionssymbol anders ausfallen kann. Genau diese differenzierte Behandlung der Zielsorten von Prädikats- und Funktionssymbolen findet sich z.B. bei COHN (1987), S. 185.

93) Dagegen ist bei der Programmiersprache PROLOG hinsichtlich des Verhältnisses zwischen Prädikats- und Funktionssymbolen eine formale Unzulänglichkeit zu beklagen. Dort werden Prädikate und Funktionen nicht klar auseinandergehalten. So werden z.B. Prädikatoren "prä(...)" mit Funktoren "fun(...)" gleichgesetzt; vgl. dazu das PROLOG-Prädikat "functor/3" bei CORDES (1988), S. 89. Ebenso werden Prädikate in den Argumenten anderer Prädikate benutzt; vgl. CORDES (1988), S. 60, der das Prädikat "vater" im Prädikat "findall" verwendet. Da die PROLOG-Syntax aber nur die Prädikatenlogik 1. Stufe abdeckt, impliziert dies, daß dasjenige Prädikat, das im Argument eines anderen Prädikats auftaucht, als Funktion verwandt worden ist. Ähnlich führt OBERWEIS (1988a), S. 5, zunächst ein Prädikat "mietwilliger_kunde" ein, um es anschließend (S. 6) im Argument eines anderen Prädikats "marke" in der Qualität einer Funktion zu verwenden. Von KINNEBROCK (1988), S. 19, werden Prädikate und Funktionen ausdrücklich miteinander identifiziert ("Ein Prädikat ist demnach eine Funktion..."). CORDES (1988), läßt mehrfach die Verquickung prädikats- und funktionsspezifischer Formulierungen erkennen (S. 11: "Jedes Faktum ... setzt sich aus einem ... Funktor ... und Objekten ... zusammen."; S. 14: "Eine Menge von Klauseln mit demselben Funktor und gleicher Stelligkeit ..." (kursive Hervorhebung im Original hier unterlassen); S. 25: "In Prolog werden Funktoren und Prädikatsnamen in ihrer Schreibweise nicht unterschieden."). Auch werden in der Spezifizierung der PROLOG-Syntax atomare prädikatenlogische Formeln ebenso als Terme und als Funktoren deklariert; vgl. CORDES (1988), S. 185. (Es wird hier von Prädikaten und Funktionen anstelle von Prädikats- bzw. Funktionssymbolen gesprochen, weil in PROLOG-Programmen der Unterschied zwischen symbolischen Ausdrücken und ihren konkreten Manifestationen ohnehin nicht beachtet wird.) Vgl. des weiteren zur Behandlung von Prädikaten als Funktionen BIBEL (1989), S. 53f.
Diese Vermengungen von Prädikats- und Funktionsbegriff verursachen zwar keine Fehler im Rahmen der logischen Programmierung, erweisen sich aber als formale Irritationen. Sie entsprechen nicht der kategorialen Verschiedenheit von Prädikats- und Funktionssymbolen, die bereits in einer früheren Anmerkung erläutert wurde. Es ist erstaunlich, daß nur DELAHAYE (1987), S. 159, die Verquickung kategorial distinkter Funktions- und Prädikatssymbole moniert: "It should be said also that in PROLOG ... the distinction between function and predicate symbols, essential in first-order logic, is not made." Allerdings muß eingeräumt werde, daß die Quellen dieser Konfusion zwischen Prädikats- und Funktionsbegriff in der frühen prädikatenlogischen Literatur selbst liegen. Darauf wurde in bezug auf die *Principia Mathematica* von WHITEHEAD und RUSSELL bereits hingewiesen.
Dagegen besitzt der spezielle Turbo-PROLOG-Dialekt den Vorzug, Prädikate und Funktionen deutlich voneinander zu unterscheiden. Dort werden Funktoren nur als Bestandteile von Prädikatsargumenten zugelassen. Darüber hinaus werden Funktoren in der "domains"-Sektion eingeführt, während Prädikate im "predicates"-Bereich definiert wer-

den. Vgl. dazu PROLOG (o.J.), S. 38ff.; TOWNSEND,C. (1987), S. 84ff.; KINNEBROCK (1988), S. 43ff. u. 125. Diese formal saubere Trennung von Prädikaten und Funktionen ist ein weiteres Argument zugunsten der Selektion des Turbo-Dialekts für die prädikatenlogische Implementierung von Synthetischen Netzen. Darüber hinaus lassen sich die "Prädikate" im reservierten "predicates"-Bereich als Prädikatssymbole auszeichnen. Eine Ungenauigkeit hat sich allerdings in die Darstellung der Syntax des Turbo-PROLOG-Dialekts bei KINNEBROCK eingeschlichen. Denn dort werden Prädikate und Funktionen unterschiedslos definiert; vgl. KINNEBROCK (1988), S. 127 ("prädikat") vs. S. 129 ("einfacher-funktor-typ").

94) Vgl. BIBEL (1982a), S. 60.

4.2.2.2 Die semantische Dimension

Bisher wurde nur der prädikatenlogische Kalkül dargestellt. Er erschließt die rein syntaktische Dimension der Prädikatenlogik durch die Konstitution des prädikatenlogischen Formelreichtums. Das reicht als formalsprachliches Fundament für Netzmodelle aber bei weitem noch nicht aus. Vielmehr muß es möglich sein, die Gültigkeit prädikatenlogischer Formeln und deren Veränderungen im Zeitablauf auszudrücken. Daher bedarf es einer Erweiterung des prädikatenlogischen Kalküls um eine formale Semantik. Zunächst wird nur die konventionelle prädikatenlogische Semantik entfaltet, die sich auf die Darstellung und Untersuchung von Formelgültigkeiten konzentriert. Erweiterungen, die sich vor allem der Variation von Formelgültigkeiten im Zeitablauf widmen, werden später als Repräsentation von Wissen über zeitlich veränderliche Sachverhalte thematisiert.

Die prädikatenlogische Semantik wird dadurch charakterisiert, daß die aussagenlogischen Wahrheitswerte im wesentlichen[1)] übernommen und um das Konzept der Prädikatsextensionen ergänzt werden. Nachfolgend wird die konventionelle Darstellungsweise statischer Prädikatsextensionen gewählt. Hierdurch wird es einerseits möglich, bei der späteren Entfaltung von SIG-Dynamiken die Neuartigkeit ihrer zeitlich variablen Prädikatsextensionen besonders herauszustellen. Andererseits wird so der Anschluß an etablierte prädikatenlogische Semantiken gewahrt.

Definition: prädikatenlogische Semantik

Eine prädikatenlogische Semantik[2)] ordnet jedem prädikatenlogischen Formelsystem eine zulässige formale Interpretation zu. Dabei werden die syntaktischen Konstrukte des Formelsystems auf folgende semantische Interpretationen abgebildet:

- Für alle Variablen wird ein gemeinsamer, nicht-leerer Definitionsbereich DB festgelegt.
- Jedes Konstantensymbol wird auf eine Konstante aus dem Definitionsbereich DB abgebildet.
- Jeder Term wird durch einen teilevaluierten Term substituiert[3)].
- Jedes Funktionssymbol wird durch eine Funktion ersetzt[4)].
- Jedes Prädikatssymbol wird auf seine Prädikatsextension abgebildet. Die Extension eines 0-stelligen Prädikatssymbols ist der Wahrheitswert für die daraus abgeleitete aussagenlogische Formel. Die Extension eines K-stelligen Prädikatssymbols mit $K \in \mathcal{N}_+$ ist die Menge aller Formelargumente, für welche die atomaren Grundtermformeln, die im zugrundeliegenden Kalkül aus dem Prädikatssymbol formiert wurden, gültig sind.
- Jeder quantifizierten Formel aus dem prädikatenlogischen Formelsystem wird ihr Wahrheitswert zugeordnet, sofern es sich um eine geschlossene Formel handelt.

<u>Erläuterungen und Ergänzungen zur Definition einer prädikatenlogischen Semantik:</u>

a) Das prädikatenlogische Formelsystem, das zuvor auf syntaktischer Ebene mit der Hilfe des prädikatenlogischen Kalküls gebildet worden ist, kann auch eine einzelne Formel darstellen[5)]. Auf dasselbe Formelsystem läßt sich eine Vielzahl unterschiedlicher Interpretationen I_r[6)] mit $r \in \{1,...,R\}$ anwenden. Die Anzahl R aller denkmöglichen Interpretationen eines Formelsystems wird grundsätzlich offen gelassen[7)]. Es resultiert jeweils ein interpretiertes Formelsystem. Die Interpretation I_r kann als eine komplex definierte[8)] Auswertungsfunktionen aufgefaßt werden, die - in einer ersten Annäherung[9)] - einer Auswertungsfunktionen $eval_i$ aus den SIG-Termauswertungen des Signaturkonzepts entspricht[10)].

b) Der gemeinsame Definitionsbereich DB aller Variablen und Konstantensymbole ist eine Menge von atomaren formalen Objekten "ob". Sie werden als (deskriptive) Konstanten bezeichnet[11)]. Ihre Gesamtheit DB heißt auch prädikatenlogisches Universum[12)]. Jede Konstante stellt eine zulässige Variablenersetzung oder eine zulässige Interpretation für ein Konstantensymbol dar. Bei den Konstanten handelt es sich um formale Objekte 2. Stufe. Sie interpretieren die Variablen und Konstantensymbole aus dem zugrundeliegenden prädikatenlogischen Kalkül, die objektsprachliche Ausdrücke 1. Stufe bedeuten. Durch die Einführung des Definitionsbereichs DB werden die Argumente "(X):" der Quantoren in einem interpretierten Formelsystem auf "(X∈DB):" erweitert.

c) Jeder Term "te" des zugrundeliegenden Kalküls wird auf einen teilevaluierten Term "te" abgebildet[13)]. Denn nur die Konstanten- und Funktionssymbole werden durch Konstanten bzw. Funktionen ersetzt. Die Variablen werden dagegen nicht substituiert, sondern nur durch ihre Definitionsbereiche ergänzt[14)]. In dieser Hinsicht entspricht die Anwendung einer Interpretation I_r auf prädikatenlogische Formeln dem Gebrauch von Teilauswertungsfunktionen $teval_i$[15)], die früher im Rahmen des Signaturkonzepts definiert wurden. Strenggenommen handelt es sich um minimale Teilauswertungsfunktionen $xeval_i$, weil keine der Variablen aus den Formelargumenten durch Konstanten ersetzt werden[16)].

Die Terme[17)] einer prädikatenlogischen Interpretation bestehen also aus Konstanten, Variablen und Funktionen. Deshalb besteht eine gravierende formale Diskrepanz der Prädikatenlogik gegenüber dem algebraischen Signaturkonzept. Sie betrifft die kategoriale Unterscheidung zwischen Termen mit ihren Operationssymbolen, Konstantensymbolen und Variablen einerseits sowie formalen Objekten andererseits, die entweder Konstanten darstellen oder aber mit der Hilfe von Funktoren (Operatoren) aus Konstanten als atomaren formalen Objekten zusammengesetzt sind. Im Signaturkonzept werden diese beiden Kategorien der Terme und der formalen Objekte sauber auseinandergehalten, indem die ersten zur SIG-Termmenge, die zweiten zur SIG-Algebra gerechnet werden. Seitens der konventionellen Prädikatenlogik werden dagegen beide Kategorien konfundiert. Denn in den teilevaluierten Termen der Prädikatsargumente dürfen termartige Variablen und objektartige Konstanten miteinander kombiniert werden.

Lediglich bei Grundtermen wird eine solche Kategorienvermengung vermieden, weil sie als variablenfreie Terme definiert sind. Solche Grundterme werden durch eine prädikatenlogische Interpretation nur auf Konstanten oder auf Komplexe aus Konstanten und Funktoren abgebildet. Daher fällt in der Interpretation eines Grundterms der resultierende teilevaluierte Term "te" mit einem atomaren bzw. zusammengesetzten formalen Objekt "ob" zusammen. Da jedoch aus der allgemeinen Notation "te" eines Terms nicht ersichtlich ist, ob es sich um einen Grundterm handelt, wird der zugeordnete teilevaluierte Term weiterhin als "te" notiert[18)].

d) Jedem K-stelligen Funktionssymbol "Fun" mit $K \in \mathcal{N}_+$ wird eine strukturgleiche Funktion "fun" dadurch zugeordnet, daß die Deklaration des Funktionssymbols durch die Angabe einer Abbildungsvorschrift "<vorschrift>" für den Funktor fun(...) konkretisiert wird:

$$\text{fun:} \quad DB_1 \times \ldots \times DB_K \rightarrow DB_{K+1}$$
$$(te_1,\ldots,te_K) \rightarrow te_{K+1} = \text{fun}(te_1,\ldots,te_K) = \text{<vorschrift>}$$

Die Terme $te_1,\ldots,te_K$ und te_{K+1} stellen - wie voranstehend erläutert - jeweils teilevaluierte Terme dar.

e) Neben den deskriptiven Konstanten bilden die Wahrheitswerte[19)] als logische Konstanten eine zweite Kategorie formaler Objekte 2. Stufe. Sie stellen die zentralen Konstrukte jeder metasprachlichen Semantik dar[20)], obwohl sie in der o.a. Semantikdefinition nur am Rande explizit erwähnt werden. Im Fall der Prädikatenlogik spielen nur die beiden BOOL'schen Wahrheitswerte "gültig" und "ungültig" eine Rolle[21]. Sie werden in der BOOL'schen Wahrheitswertemenge BOOL = {gültig,ungültig} zusammengefaßt[22]. Die beiden Sachverhalte, daß eine objektsprachliche Formel $prä_u$ unter einer Interpretation I_r entweder gültig oder ungültig ist, wird durch die metasprachlichen Ausdrücke $I_r(prä_u)$ = gültig bzw. $I_r(prä_u)$ = ungültig dargestellt[23]. Des weiteren bezeichnen die beiden Symbole "T" und "$\perp$" die allgemeingültige (tautologische) bzw. niemals gültige (kontradiktorische) Formel[24]: $\top :\Leftrightarrow I_r(\top)$ = gültig für alle $r \in \{1,...,R\}$ und $\perp :\Leftrightarrow I_r(\perp)$ = ungültig für alle $r \in \{1,...,R\}$.

f) In der prädikatenlogischen Semantik wird jedes Prädikatssymbol aus dem prädikatenlogischen Kalkül durch die Angabe seiner Extension *vollständig* interpretiert. Dieser Sachverhalt wird als extensionale Definition des Prädikatsbegriffs bezeichnet. Die hierauf gründende extensionale Semantik der Prädikatenlogik[25] entspricht genau dem früher angesprochenen Ansatz aller formalen Konzepte, nur auf extensional definierte Konstrukte zurückzugreifen. Auf diese Weise werden alle prädikatenlogischen Semantiken, die auf intensionalen Prädikatsbedeutungen beruhen, von vornherein ausgeschlossen[26]. Ein Prädikatssymbol, dessen Bedeutung durch die Angabe seiner Extension festgelegt ist, wird als Prädikatskonstante bezeichnet.

g) Die Prädikatsextension[27] $EXT_{u.r}$, die einem K_u-stelligen Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ mit $K_u \in \mathcal{N}_+$ durch eine Interpretation I_r zugeordnet wird, ist eine Menge von konstanten K_u-Tupeln $(ob_1,...,ob_{Ku})$[28], für die gilt: Die konstante atomare Formel $prä_u(ob_1,...,ob_{Ku})$ ist unter der Interpretation I_r gültig[29]. Dabei wird vorausgesetzt, daß:

- die konstante atomare Formel $prä_u(ob_1,...,ob_{Ku})$ unter der Interpretation I_r aus einer Grundtermformel $prä_u(gt_1,...,gt_{Ku})$ durch Teilevaluierung der Terme gt_k mit $k \in \{1,...,K_u\}$ hervorgegangen ist;
- die Grundtermformel $prä_u(gt_1,...,gt_{Ku})$ durch Kombination des Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ mit den Grundtermen $gt_1,...,gt_{Ku}$ gemäß der prädikatenlogischen Formierungsregel FR_P gebildet worden ist.

Bei jeder Tupelkomponente ob_k handelt es sich um ein formales Objekt, das direkt oder indirekt aus dem Definitionsbereich DB stammt[30]: Jedes formale Objekt ob_k stellt entweder als atomares formales Objekt unmittelbar eine Konstante dar. Oder es ist mit der Hilfe von mindestens einem Funktor fun_k durch $ob_k = fun_k(...ob_l...)$ mittelbar aus Konstanten ob_l zusammengesetzt.

h) Eine uninterpretierte atomare Formel $prä_u(te_1,...,te_{Ku})$, die im zugrundeliegenden prädikatenlogischen Kalkül aus dem Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ formiert wurde, wird durch die Interpretation I_r auf eine interpretierte atomare Formel $prä_u(te_1,...,te_{Ku})$ mit teilevaluierten Termen te_k für $k \in \{1,...,K_u\}$ abgebildet[31]. Das Interpretationsergebnis - d.h. die Charakteristik der teilevaluierten Terme - hängt wesentlich davon ab, ob eine konstante Formel (Grundtermformel) oder aber eine variable Formel interpretiert wird.

i) Jede uninterpretierte atomare Grundtermformel $prä_u(gt_1,...,gt_{Ku})$ wird durch die Interpretation I_r auf eine interpretierte atomare konstante Formel $prä_u(ob_1,...,ob_{Ku})$ abgebildet[32]. Dabei werden die Grundterme gt_k unter der Interpretation I_r vollständig durch formale Objekte[33] ob_k mit $ob_k = I_r(te_k)$ für alle $k \in \{1,..,K_u\}$ substituiert.

j) Jede uninterpretierte atomare variable Formel $prä_u(...,X_k,...)$ aus dem prädikatenlogischen Kalkül wird durch die Interpretation I_r auf eine interpretierte atomare, ebenso variable Formel $prä_u(...,X_k,...)$ abgebildet. Denn durch die Interpretation I_r ist keine Substitution von Variablen durch formale Objekte als Konstanten definiert. In der interpretierten variablen Formel werden lediglich die Grundterme der uninterpretierten variablen Formel durch Konstanten und - gegebenenfalls - Funktionen ersetzt.

k) Für jede atomare Grundtermformel $prä_u(gt_1,...,gt_{Ku})$ kann entschieden werden, ob ihr Argument unter der Interpretation I_r zur Extension $EXT_{u.r}$ desjenigen Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ gehört, mit dessen Hilfe die Grundtermformel formiert wurde. Denn einer Grundtermformel wird unter der Interpretation I_r das konstante Formelargument $(ob_1,...,ob_{Ku})$ zugeordnet, dessen Zugehörigkeit zur Prädikatsextension $EXT_{u.r}$ wohldefiniert ist[34]. Für alle atomaren variablen Formeln $prä_u(...,X_k,...)$ ist es dagegen unmöglich zu entscheiden, ob ihre Argumente unter der Interpretation I_r zur Extension $EXT_{u.r}$ ihres Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ gehören. Denn es würde jeweils von der Ersetzung aller Variablen X_k eines variablen Arguments $(...,X_k,...)$ durch Konstanten ob_k ab, ob die derart transformierte Formel $prä_u(ob_1,...,ob_{Ku})$ zur Prädikatsextension $EXT_{u.r}$ gehört. Eine solche Variablensubstitution wird aber durch die Interpretation I_r nicht definiert. Daher spielen für die Extension eines Prädikatssymbols nur die uninterpretierten atomaren Grundtermformeln $prä_u(gt_1,...,gt_{Ku})$ eine Rolle.

l) Zwecks Vereinfachung der Diktion wird fortan die Bezeichnung "Prädikat $prä_u$" zugelassen, um folgende prädikatenlogischen Formeln verkürzt anzusprechen:

- uninterpretierte atomare Formeln $prä_u(te_1,...,te_{Ku})$ mit Termen te_k für alle $k \in \{1,...,K_u\}$,
- uninterpretierte atomare Grundtermformeln $prä_u(gt_1,...,gt_{Ku})$ mit Grundtermen gt_k für alle $k \in \{1,...,K_u\}$,
- uninterpretierte atomare variable Formeln $prä_u(...,X_k,...)$ mit Variablen X_k für mindestens ein $k \in \{1,...,K_u\}$,
- interpretierte atomare Formeln $prä_u(te_1,...,te_{Ku})$ mit teilevaluierten Termen te_k für alle $k \in \{1,...,K_u\}$,
- interpretierte atomare konstante Formeln $prä_u(ob_1,...,ob_{Ku})$ mit formalen Objekten ob_k für alle $k \in \{1,...,K_u\}$,
- interpretierte atomare variable Formeln $prä_u(...,X_k,...)$ mit Variablen X_k für mindestens ein $k \in \{1,...,K_u\}$.

Von einer Formel p_j wird dagegen gesprochen, wenn eine uninterpretierte oder interpretierte prädikatenlogische Formel vorliegt, die beliebig komplex aus anderen Formeln zusammengesetzt sein kann. Dies schließt als Grenzfall die vorgenannten atomaren Formeln $prä_u$ ein.

m) Betrachtet wird ein konstantes Tupel $(ob_1,...,ob_{Ku})$, das unter der Interpretation I_r ein zulässiges Argument für die uninterpretierte Grundtermformel $prä_u(gt_1,...,gt_{Ku})$ ist. Diese Formel wurde im zugrundeliegenden Kalkül aus dem Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ formiert. Falls das Tupel $(ob_1,...,ob_{Ku})$ aus der Extension $EXT_{u.r}$ des Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ stammt, dann wird die uninterpretierte Grundtermformel $prä_u(gt_1,...,gt_{Ku})$[35] unter der Interpretation I_r durch das konstante Argument $(ob_1,...,ob_{Ku})$ erfüllt[36]. Hierfür gilt semi-formal[37]:

$$I_r(gt_1,...,gt_{Ku}) = (I_r(gt_1),...,I_r(gt_{Ku})) = (ob_1,...,ob_{Ku}) \wedge (ob_1,...,ob_{Ku}) \in EXT_{u.r}$$
$$\Rightarrow \quad prä_u(gt_1,...,gt_{Ku}) \text{ "wird erfüllt durch" } (ob_1,...,ob_{Ku})$$

Die Erfüllung der uninterpretierten Grundtermformel $prä_u(gt_1,...,gt_{Ku})$ durch das Argument $(ob_1,...,ob_{Ku})$ wird äquivalent dadurch ausgedrückt, daß die interpretierte konstante Formel $prä_u(ob_1,...,ob_{Ku})$ unter der Interpretation I_r gültig[38] ist[39]. Hierfür wurde bereits oben die Notation $I_r(prä_u) = gültig$ eingeführt. Durch Erweiterung dieser metasprachlichen Notation um die Argumentstruktur der involvierten Prädikate $prä_u$ lassen sich die Erfüllung der uninterpretierten Grundtermformel $prä_u(gt_1,...,gt_{Ku})$ und die Gültigkeit der interpretierten konstanten Formeln $prä_u(ob_1,...,ob_{Ku})$ unter der Interpretation I_r durch das Argument $(ob_1,...,ob_{Ku})$ formal vollständig darstellen durch einen zweistufigen Interpretationsprozeß:

- argumentbezogene Interpretation der uninterpretierten Grundtermformel $\text{prä}_u(gt_1,...,gt_{Ku})$ durch Zuordnung eines Konstantentupels:

  $\text{prä}_u(I_r(te_1,...,te_{Ku})) = \text{prä}_u(I_r(gt_1),...,I_r(gt_{Ku})) = \text{prä}_u(ob_1,...,ob_{Ku})$

- wahrheitswertbezogene Interpretation der interpretierten konstanten Formel $\text{prä}_u(ob_1,...,ob_{Ku})$ durch Zuordnung ihres Wahrheitswerts "gültig":

  $I_r(\text{prä}_u(ob_1,...,ob_{Ku})) = \text{gültig}$

n) Die Extension $EXT_{u.r}$ eines K_u-stelligen Prädikatssymbols mit $K_u \in \mathcal{N}_+$[40] unter einer Interpretation I_r läßt sich formal präzisieren als die explizite Auflistung aller Konstantentupel, für welche die interpretierten konstanten Formeln $\text{prä}_u(ob_1,...,ob_{Ku})$ gültig sind[41]:

$$EXT_{u.r} = \{(ob_1,...,ob_{Ku}): I_r(\text{prä}_u(ob_1,...,ob_{Ku})) = \text{gültig}\}$$

Dies bedeutet zugleich, daß die syntaktisch formierten, uninterpretierten Grundtermformeln $\text{prä}_u(gt_1,...,gt_{Ku})$ von allen Argumenten $(ob_1,...,ob_{Ku})$ aus der Prädikatsextension $EXT_{u.r}$ unter der Interpretation I_r erfüllt werden. Diese Argumente werden auch als formel- oder prädikatserfüllende Argumente bezeichnet.

o) Als Grenzfall ist auch die leere Extension $EXT_{u.r} = \varnothing$ möglich. Dann wird die uninterpretierte Grundtermformel $\text{prä}_u(gt_1,...,gt_{Ku})$ unter der Interpretation I_r von überhaupt keinem Argument erfüllt; es gibt keine gültige interpretierte konstante Formel $\text{prä}_u(ob_1,...,ob_{Ku})$.

p) Um die Diktion zu vereinfachen, wird es zugelassen, die uninterpretierten Grundtermformeln $\text{prä}_u(gt_1,...,gt_{Ku})$ und die interpretierten konstanten Formeln $\text{prä}_u(ob_1,...,ob_{Ku})$ nicht mehr zu unterscheiden. Dann wird nur noch von "dem" Prädikat prä_u oder "der" Formel prä_u gesprochen. Entsprechend läßt sich von der Extension "des" Prädikats prä_u reden, das alle vorgenannten uninterpretierten und interpretierten Formeln vertritt. Dieses (Stellvertreter-)Prädikat prä_u wird durch ein Argument $(ob_1,...,ob_{Ku})$ genau dann erfüllt, wenn dieses Argument aus der Extension $EXT_{u.r}$ des zugrundeliegenden Prädikatssymbols $\text{Prä}_u(st_1,...,st_{Ku})$ stammt.

q) Falls aus dem zugrundeliegenden prädikatenlogischen Kalkül aus dem Prädikatssymbol $\text{Prä}_u(st_1,...,st_{Ku})$ nur genau eine uninterpretierte Grundtermformel $\text{prä}_u(gt_1,...,gt_{Ku})$ formiert wurde, kann die Extension $EXT_{u.r}$ des Prädikatssymbols auch vereinfacht als Extension dieser einen Formel angesehen werden. Die Prädikatsextension $EXT_{u.r}$ umfaßt dann genau alle konstanten Objekttupel $(ob_1,...,ob_{Ku})$, welche die Grundtermformel $\text{prä}_u(gt_1,...,gt_{Ku})$ unter der Interpretation I_r erfüllen. Diese formelerfüllenden Objekttupel gelten für alle identischen Vorkommnisse dieser einen atomaren Formel in einem prädikatenlogischen Formelsystem.

r) Komplizierter fällt die Prädikatsextension $EXT_{u.r}$ aus, wenn für das zugehörige Prädikatssymbol $\text{Prä}_u(st_1,...,st_{Ku})$ mehrere uninterpretierte Grundtermformeln $\text{prä}_u(gt_1,...,gt_{Ku})$ mit unterschiedlichen Argumentstrukturen gebildet wurden. Dann zerfällt die Extension EXT_j in entsprechend viele disjunkte Teilmengen, deren jede für genau eine uninterpretierte Grundtermformel $\text{prä}_u(gt_1,...,gt_{Ku})$ jene Objekttupel $(ob_1,...,ob_{Ku})$ umfaßt, für die gilt: Die Objekttupel stimmen in ihrer Struktur mit der Argumentstruktur ihrer zugehörigen Grundtermformel überein und erfüllen diese Formel unter der Interpretation I_r. Wenn von der strukturellen Unterschiedlichkeit der uninterpretierten Grundtermformeln $\text{prä}_u(gt_1,...,gt_{Ku})$ abstrahiert wird, die aus demselben Prädikatssymbol $\text{Prä}_u(st_1,...,st_{Ku})$ gebildet wurden, kann die Extension $EXT_{u.r}$ auch vereinfacht als die Extension "der" Formel prä_u bezeichnet werden.

s) Durch die Prädikatsextensionen $EXT_{u.r}$ sind die Wahrheitswerte aller interpretierten konstanten atomaren Formeln $prä_u(ob_1,...,ob_{Ku})$ in einem prädikatenlogischen Formelsystem unter der Interpretation I_r eindeutig bestimmt:

$$I_r(prä_u(ob_1,...,ob_{Ku})) = \text{gültig} \quad \Leftrightarrow \quad (ob_1,...,ob_{Ku}) \in EXT_{u.r}$$
$$I_r(prä_u(ob_1,...,ob_{Ku})) = \text{ungültig} \quad \Leftrightarrow \quad (ob_1,...,ob_{Ku}) \neq EXT_{u.r}$$

Dieser Sachverhalt wird als Wahrheitsdefinitheit[42)] der Formeln $prä_u(ob_1,...,ob_{Ku})$ bezeichnet.

t) Die Wahrheitsdefinitheit von Formeln erstreckt sich sogar noch weiter auf alle prädikatenlogischen Sätze, also auf diejenigen Formeln, die keine freien Variablen enthalten (geschlossene Formeln)[43)]. Dies ist entweder der Fall, wenn die Argumente der interpretierten Formeln keine Variablen umfassen. Oder die interpretierten Formeln enthalten nur Variablen, die von Quantoren vollständig gebunden sind. Die Wahrheitswerte solcher prädikatenlogischen Sätze ergeben sich rekursiv aus den aussagenlogischen Wahrheitswertetafeln für die Konnektoren und den Negator[44)], aus der prädikatenlogischen Wahrheitswertdefinition für Quantoren[45)] sowie aus den o.a. Wahrheitswerten der jeweils zugrundeliegenden atomaren konstanten Formeln.

u) Ein K_u-stelliges Fakt[46)] mit $K_u \in \mathcal{N}_+$ ist ein metasprachlicher Ausdruck dafür, daß eine objektsprachliche atomare K_u-stellige Grundtermformel unter der jeweils betrachteten Interpretation gültig ist[47)]: Die interpretierte Grundtermformel $prä_u(ob_1,...,ob_{Ku})$ ist bezüglich der Interpretation I_r genau dann ein Fakt, wenn ihr konstantes Argument $(ob_1,...,ob_{Ku})$ aus der Extension $EXT_{u.r}$ des Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ stammt. Ein Fakt stellt somit ein gültiges atomares Prädikatsvorkommnis $prä_u(ob_1,...,ob_{Ku})$ dar[48)]. Das Argument $(ob_1,...,ob_{Ku})$ eines Fakts heißt faktisches Objekttupel oder Fakttupel.

v) Die Extension eines K_u-stelligen Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ mit $K_u \in \mathcal{N}_+$ unter einer Interpretation I_r kann äquivalent als dessen Faktenmenge $FAK_{u.r}$ dargestellt werden:

$$\begin{aligned} FAK_{u.r} &= \{prä_u(ob_1,...,ob_{Ku}) \colon I_r(prä_u(ob_1,...,ob_{Ku})) = \text{gültig}\} \\ &= \{prä_u(ob_1,...,ob_{Ku}) \colon (ob_1,...,ob_{Ku}) \in EXT_{u.r}\} \end{aligned}$$

Extension und Faktenmenge desselben Prädikatssymbols sind zwar äquivalent, aber nicht identisch. Denn die Prädikatsextension umfaßt alle *Argumente* gültiger Prädikatsvorkommnisse. Die Faktenmenge besteht dagegen aus diesen gültigen Prädikatsvorkommnissen selbst.

w) Fakten und Faktenmengen lassen sich auf den Grenzfall 0-stelliger Prädikatssymbole und daraus abgeleiteter aussagenlogischer Formeln ausdehnen[49)]. Ein Fakt ist in diesem Fall eine aussagenlogische Formel, die unter der jeweils betrachteten Interpretation gültig ist. Für dasselbe 0-stellige Prädikatssymbol können wegen seiner Strukturlosigkeit keine verschiedenen aussagenlogischen Formeln definiert werden, die sich durch ihre Argumente voneinander unterscheiden ließen. Darüber hinaus kann die eine aussagenlogische Formel, die sich aus einem 0-stelligen Prädikatssymbol ableiten läßt, nur entweder gültig oder aber ungültig sein[50)]. Daher gilt für die Faktenmenge eines jeden 0-stelligen Prädikatssymbols $Prä_u()$ die einfache Festlegung: Die Faktenmenge ist entweder leer oder enthält genau die eine ableitbare aussagenlogische Formel $prä_u()$ als Element. Der erste (zweite) Fall $FAK_{u.r} = \emptyset$ ($FAK_{u.r} = \{prä_u()\}$) tritt genau dann ein, wenn die aussagenlogische Formel $prä_u()$ unter der Interpretation I_r ungültig (gültig ist).

x) FAK_r ist die Gesamtheit aller Fakten, die für alle Prädikatssymbole $Prä_u(st_1,...,st_{Ku})$ aus Prädikatssymbolemenge $PRÄ = \{Prä_u(st_1,...,st_{Ku}) \colon u = 1,...,U\}$ eines prädikatenlogischen Formelsystems unter einer Interpretation I_r vorliegen. Diese Faktengesamtheit läßt sich als Familie oder als Vereinigungsmenge aller Faktenmengen $FAK_{r.u}$ definieren. Da die involvierten Faktenmengen $FAK_{u.r}$ disjunkt sind[51)], verhalten sich beide Darstellungsvarianten der Faktengesamtheit äquivalent. Beide werden mit demselben Symbol "FAK_r" notiert[52)]:

$$FAK_r = (FAK_{u.r}: u=1,...,U)$$
$$\Leftrightarrow \quad FAK_r = \bigcup (u=1,...,U): FAK_{u.r}$$

Die Faktenmengen $FAK_{u.r}$, die sich jeweils auf ein Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ beziehen, werden fortan auch als prädikatspezifische Faktenmengen bezeichnet. Ihre Vereinigungsmenge FAK_r heißt dagegen eine systemspezifische Faktenmenge für die Interpretation I_r, weil sie sich auf die gesamte Prädikatssymbolemenge PRÄ eines prädikatenlogischen Formelsystems erstreckt.

y) Es kann die metasprachliche Vereinbarung getroffen werden, daß alle konstanten atomaren Formeln, die in einem Formelsystem auf objektsprachlicher Ebene explizit angeführt werden, aus metasprachlicher Perspektive Fakten darstellen. Da eine konstante atomare Formel $prä_u(ob_1,...,ob_{Ku})$ gültig ist, falls sie zur Faktenmenge $FAK_{u.r}$ ihres Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ gehört, kann unter Voraussetzung der vorgenannten Vereinbarung auf die explizite Verwendung des Wahrheitswerts "gültig" für die Auszeichnung von Fakten verzichtet werden (implizite Faktennotation). Denn dann ist jede explizit angeführte konstante atomare Formel qua Notationsvereinbarung gültig[53)]. Hiervon wird in prädikatenlogischen Abhandlungen oftmals Gebrauch gemacht. Denn auf diese Weise kommt die Beschreibung einer prädikatenlogischen Semantik mit weitgehend[54)] syntaktischen Notationsmitteln aus.

Auch für die Programmiersprache PROLOG, die später für die Implementierung prädikatenlogischer Formelsysteme herangezogen wird, ist das der Fall. Allerdings stellt diese Notationsvereinfachung aus konzeptioneller Sicht eine fragwürdige Reduktion semantischer auf syntaktische Konstrukte dar. Hierdurch wird die formalsprachliche, zur konzeptionellen Klarheit beitragende Differenzierung zwischen objekt- und metasprachlicher Ausdrucksebene aufgegeben. Statt dessen werden die beiden formalsprachlichen Ebenen auf derselben Programmiersprachenebene undifferenziert miteinander vermengt. Objektsprachliche prädikatenlogische Formeln werden mit Hilfe derselben Syntax ausgedrückt wie metasprachliche Feststellungen von Fakten, d.h. von Formelgültigkeiten. Eine explizite Semantik, in der prädikatenlogische Interpretationen und Wahrheitswerte ausgedrückt werden können, sieht die Sprache PROLOG dagegen grundsätzlich nicht vor[55)]. Wegen der Einbuße an formalsprachlichem Differenzierungsvermögen geht der Verf. mit der voranstehend skizzierten Notationsvereinfachung nur sparsam um[56)].

z) Falls jedoch die Unterscheidung zwischen objektsprachlichen Formeln und metasprachlichen Fakten bedeutsam ist[57)], zieht der Verf. einen expliziten Ausweis der Gültigkeit faktischer Formeln vor. Dies entspricht dem Postulat expliziter Modellierung. Allerdings erweisen sich der Umgang mit prädikatenlogischen Interpretationen I_r und die explizite Zuordnung des Wahrheitswertes "gültig" zu faktischen Formeln als ausgesprochen mühselig. Sie führen auch zu einer komplizierten, unübersichtlichen Notation von semantisch definierten Sachverhalten. Darüber hinaus lassen sich semantische Ausdrücke in der Programmiersprache PROLOG überhaupt nicht explizit darstellen[58)]. Doch aus den vorgenannten Schwierigkeiten besteht ein Ausweg, die Repräsentation der Gültigkeit von faktischen Formeln sowohl explizit als auch kompakt und übersichtlich zu gestalten[59)]. Darüber hinaus erlaubt er später, das Konzept der Multimengen in die Prädikatenlogik zu integrieren, das für die Repräsentation der Markierungen von Prädikat/Transition-Netzen eine entscheidenden Rolle spielt. Zu diesem Zweck werden ein besonders ausgezeichnetes, einstelliges Prädikatssymbol $Fakt_r(st)$ und eine zugehörige atomare Formel $fakt_r(te)$ eingeführt. Die atomare Formel $fakt_r(te)$ wird fortan auch als faktische Formel oder kurz als ein Fakt angesprochen.

Auf den ersten Blick bietet es sich an, das Vorliegen eines Fakts, das einen metasprachlichen Sachverhalt aus der prädikatenlogischen Semantik darstellt, mit Hilfe der rein syntaktisch definierten Formel $\text{fakt}_r(\text{te})$ in objektsprachlicher Weise darzustellen. Hierfür ließe sich als (meta-)metasprachliche Zuordnungskonvention festlegen:

$$I_r(\text{prä}_u(ob_1,...,ob_{Ku})) = \text{gültig} \quad :\Leftrightarrow \quad \text{fakt}_r(\text{prä}_u(ob_1,...,ob_{Ku}))$$

Die objektsprachliche Formel $\text{fakt}_r(\text{prä}_u(ob_1,...,ob_{Ku}))$ drückt dabei aus, daß die konstante atomare Formel $\text{prä}_u(ob_1,...,ob_{Ku})$ unter der Interpretation I_r gültig ist. Diese objektsprachliche Faktenrepräsentation durch Formeln $\text{fakt}_r(\text{prä}_u(ob_1,...,ob_{Ku}))$ würde jedoch bedeuten, daß das konventionelle Fundament der Prädikatenlogik 1. Stufe verlassen wird. Denn die Formel $\text{fakt}_r(\text{prä}_u(ob_1,...,ob_{Ku}))$ erstreckt sich in ihrem Argument auf eine prädikatenlogische Formel $\text{prä}_u(ob_1,...,ob_{Ku})$ 1. Stufe. Folglich müßte der Ausdruck $\text{fakt}_r(\text{prä}_u(ob_1,...,ob_{Ku}))$ eine Formel aus der Prädikatenlogik 2. Stufe darstellen. Dies widerspricht aber der früheren Feststellung, die Prädikatenlogik 1. Stufe reiche für die Formulierung Synthetischer Netze im wesentlichen aus. Dennoch liegen objektsprachliche Faktenrepräsentationen, die dem oben skizzierten Ansatz weitgehend gleichen, dem hier verwendeten Softwarepaket PASIPP zugrunde[60)]. Es wurde auf der prädikatenlogischen Basis der Programmiersprache PROLOG verfaßt, die ihrerseits die Prädikatenlogik 1. Stufe nicht überschreitet. Daher muß ein Weg existieren, die objektsprachliche Faktenrepräsentation innerhalb der Prädikatenlogik 1. Stufe zu realisieren. In die gleiche Richtung weisen prädikatenlogische Notationen, die im Bereich der KI-Forschung benutzt werden, um die Gültigkeit einer Formel in objektsprachlicher Weise auszudrücken[61)].

Ausgangspunkt ist die Schwäche der Programmiersprache PROLOG, zwischen Prädikaten und Funktionen nicht klar zu differenzieren[62)]. Statt dessen werden diese unterschiedlichen prädikatenlogischen Konstruktkategorien so miteinander vermengt, daß sich interpretierte atomare Formeln $\text{prä}_u(te_1,...,te_{Ku})$ genau so wie die Bilder $\text{fun}_j(te_1,...,te_{Ku})$ benutzen lassen. Dies wird ein an einem Beispiel verdeutlicht, das aus der Beschreibung der PROLOG-Implementierung des Softwarepakets PASIPP entnommen ist[63)]:

Zunächst wird als Prädikat definiert:

mietwilliger_kunde(Name,adresse(Strasse,Wohnort),Alter)

Anschließend taucht dieses "Prädikat" jedoch im Argument eines neues Prädikats auf, für das ein Fakt eingeführt wird mit:

marke(1,*mietwilliger_kunde*("Maier",adresse("Kaiserstr. 10","Karlsruhe"),40))

Es ist offensichtlich, daß die atomare Formel mietwilliger_kunde($te_{k.1},te_{k.2},te_{k.3}$) mit tief strukturierten dreistelligen Argument dazu benutzt wird, um den zweiten Term $te_{m.2}$ in der atomaren Formel marke($te_{m.1},te_{m.2}$), die wiederum ein tief strukturiertes Argument besitzt, zu ersetzen. Die Substitution $te_{m.2}$: =mietwilliger_kunde($te_{k.1},te_{k.2},te_{k.3}$) ist aber nur dann wohldefiniert, wenn der Ausdruck mietwilliger_kunde($te_{k.1},te_{k.2},te_{k.3}$) ein Bild der Funktion "mietwilliger_kunde" darstellt. Also wurde der ursprünglich als Prädikat eingeführte Ausdruck "mietwilliger_kunde" als Funktion umgedeutet: der Prädikator mietwilliger_kunde(...) wurde stillschweigend durch den Funktor mietwilliger_kunde(...) ersetzt.

Die PROLOG-Schwäche, Prädikate und Funktionen nicht klar auseinanderzuhalten, wird nunmehr ausgenutzt, um die objektsprachliche Notation von Fakten einführen zu können, ohne den Bereich der Prädikatenlogik 1. Stufe zu verlassen[64)]. Allerdings wird die kategoriale Unterscheidung zwischen Prädikaten und Funktionen zwecks formaler Präzision weiterhin aufrechterhalten. Für jedes Prädikatssymbol $\text{Prä}_u(st_1,...,st_{Ku})$ kann ein korrespondierendes Funktionssymbol

Prä_{uf}: $st_1,...,st_{Ku} \rightarrow st_{Ku+1}$ vereinbart werden[65]. Es wird als funktionale Deutung seines zugrundeliegenden Prädikatssymbols bezeichnet. Das Funktionssymbol Prä_{uf} wird unter jeder Interpretation I_r auf dieselbe Funktion prä_{uf}:$te_1,...,te_{Ku} \rightarrow te_{Ku+1}$ mit $te_{Ku+1} = \text{prä}_{uf}(te_1,...,te_{Ku})$ abgebildet. Daher korrespondiert mit jeder interpretierten atomaren Formel $\text{prä}_u(te_1,...,te_{Ku})$, die aus dem Prädikatssymbol $\text{Prä}_u(st_1,...,st_{Ku})$ gebildet wurde, das Funktionsbild $te_{Ku+1} = \text{prä}_{uf}(te_1,...,te_{Ku})$. Da dieses Funktionsbild einen konventionell definierten Term te_{Ku+1} darstellt, kann es ohne Schwierigkeiten gegen den Term "te" im Argument der Formel $\text{fakt}_r(te)$ substituiert werden. Dabei liegt der atomaren Formel fakt_r das bereits oben eingeführte Funktionssymbol $\text{Fakt}_r(st)$ zugrunde. Falls die atomare Formel $\text{prä}_u(te_1,...,te_{Ku})$ unter einer Interpretation I_r eine konstante atomare Formel $\text{prä}_u(ob_1,...,ob_{Ku})$ ist, stellt ihr korrespondierendes Funktionsbild einen konstanten Term $te_{Ku+1} = (ob_1,...,ob_{Ku})$ dar. Dann handelt es sich auch bei der Formel $\text{fakt}_r(\text{prä}_{uf}(ob_1,...,ob_{Ku}))$ um eine konstante atomare Formel. Nunmehr läßt sich die Zuordnungskonvention für die objektsprachliche Repräsentation von Fakten formal korrekt festlegen:

$$I_r(\text{prä}_u(ob_1,...,ob_{Ku})) = \text{gültig} \quad :\Leftrightarrow \quad \text{fakt}_r(\text{prä}_{uf}(ob_1,...,ob_{Ku}))$$

Da die semantisch definierten Ausdrücke der Interpretation I_r und des Wahrheitswerts "gültig" in der Programmiersprache PROLOG nicht berücksichtigt werden können, läßt sich auch die o.a. Zuordnungskonvention dort nicht formulieren. Statt dessen wird für die PROLOG-basierte Implementierung von prädikatenlogischen Formelsystemen fortan unterstellt: Jedes objektsprachliche Vorkommnis der Formel $\text{fakt}_r(\text{prä}_{uf}(ob_1,...,ob_{Ku}))$ drückt zugleich den metasprachlichen Sachverhalt aus, daß die konstante atomare Formel $\text{prä}_u(ob_1,...,ob_{Ku})$ unter der Interpretation I_r gültig ist (Interpretationsprämisse). Das oben angeführte Beispiel wird daher in dieser Arbeit ausgedrückt als:

fakt_r(1,*mietwilliger_kunde*("Maier",adresse("Kaiserstr. 10","Karlsruhe"),40))

Die Interpretationsprämisse entspricht zwar im Prinzip der problematischen Notationsvereinfachung, objekt- und metasprachliche Ausdrücke miteinander zu vermengen. Dennoch läßt sich der hier gewählte spezielle Ansatz gegenüber der oben geäußerten generellen Kritik rechtfertigen: Erstens muß *jede* Implementierung prädikatenlogischer Formelsysteme, die auf die Programmiersprache PROLOG zurückgreift, an mindestens einer Stelle metasprachliche Sachverhalte in vereinfachter objektsprachlicher Notation ausdrücken. Dies ergibt sich zwangsläufig aus dem bereits erwähnten Sachverhalt, daß explizite PROLOG-Konstrukte für die metasprachliche Formulierung semantischer Ausdrücke überhaupt nicht existieren. Zweitens wird die problematische Reduktion von metasprachlichen auf objektsprachliche Ausdrücke dadurch gelindert, daß in der speziellen objektsprachlichen Notation $\text{fakt}_r(\text{prä}_{uf}(ob_1,...,ob_{Ku}))$ durch den Formelnamen "fakt_r" der metasprachliche Charakter explizit ausgewiesen wird[66]. Daher erfolgt zwar noch eine formalsprachliche Vermengung objekt- und metasprachlicher Konstrukte. Aber bei inhaltlicher Betrachtung der verwendeten Formeln können sie weiterhin klar auseinander gehalten werden. Dies gilt hingegen nicht für die o.a. Notationsvereinfachung, Fakten als atomare Formeln $\text{prä}_u(ob_1,...,ob_{Ku})$ zu repräsentieren. Denn dem Ausdruck $\text{prä}_u(ob_1,...,ob_{Ku})$ kann per se nicht angesehen werden, ob es sich um ein objektsprachliches Prädikatsvorkommnis oder um ein metasprachliches Fakt handelt. Daher werden fortan *alle* faktischen Prädikatsvorkommnisse $\text{prä}_u(ob_1,...,ob_{Ku})$, die unter einer Interpretation I_r gültig sind, mit Hilfe der Formeln $\text{fakt}_r(\text{prä}_{uf}(ob_1,...,ob_{Ku}))$ und der funktionalen Deutungen $\text{prä}_{uf}(ob_1,...,ob_{Ku})$ dargestellt[67].

Gegenüber der oben vorgelegten, vorläufigen Zuordnungskonvention liegt zwar nur ein geringfügiger Notationsunterschied vor. Doch die voranstehenden Erläuterungen haben verdeutlicht, daß sich auf diese Weise einerseits vermeiden läßt, die Prädikatenlogik 1. Stufe zu verlassen. Andererseits wird es möglich, Fakten formal übersichtlich auf der objektsprachlichen Ebene darzustellen[68], ohne die kategoriale Verschiedenheit von Prädikaten und Funktionen außer acht zu lassen. Darüber hinaus ist die Differenzierung zwischen Prädikaten und ihren funktionalen Deutungen im allgemeinen nur zur theoretischen Klärung der formalen Sachverhalte notwendig. Bei praktischen Implementierungen von prädikatenlogischen Formelsystemen läßt sich der Differenzierungsaufwand zumeist vermeiden[69)].

A) Die objektsprachliche Faktenrepräsentation gestattet es auch, Integritätsbedingungen auf einfache Weise auszudrücken. Integritätsbedingungen dienen dazu, zwischen zulässigen und unzulässigen Faktenmengen FAK_r zu differenzieren[70)]. Jede Integritätsbedingung wird durch eine teilevaluierte prädikatenlogische Formel $ib_w(te_1,...,te_{Kw})$[71)] spezifiziert, für die gilt: Die Integritätsformel ist für jede zulässige Faktenmenge gültig und für jede unzulässige Faktenmenge ungültig[72)]. Darüber hinaus wird vorausgesetzt, daß jede Integritätsformel eine Klausel ohne freie Variablen[73)] darstellt[74)]. Unter dieser Voraussetzung ist jede Komponente einer Integritätsformel entweder eine atomare Formel $prä_u(te_1,...,te_{Ku})$ als positive Formelkomponente oder aber deren Negat $\neg prä_u(te_1,...,te_{Ku})$ als negative Formelkomponente[75)]. Falls eine Integritätsformel aus mehreren Komponenten zusammengesetzt ist, so bilden diese ein Adjugat. Eine aktuelle Faktenmenge FAK_r ist bezüglich einer Integritätsbedingung daher genau dann zulässig[76)], wenn mindestens eine der Komponenten aus der Integritätsformel für diese Faktenmenge gültig ist. Denn ein Adjugat ist genau dann gültig, wenn mindestens eine seiner Komponenten gültig ist. Für den besonders übersichtlichen Sonderfall, daß die Integritätsformel eine variablenfreie - also konstante - Klausel darstellt, folgt daraus: Eine Faktenmenge ist bezüglich dieser Integritätsformel genau dann zulässig, wenn mindestens einer von den folgenden zwei Sachverhalten zutrifft:

- Für mindestens eine positive Komponente $prä_u(ob_1,...,ob_{Ku})$ aus der Integritätsformel existiert in der Faktenmenge FAK_r ein korrespondierendes Fakt $fakt_r(prä_{uf}(ob_1,...,ob_{Ku}))$. Dann ist die positive Formelkomponente $prä_u(ob_1,...,ob_{Ku})$ unmittelbar gültig. Folglich ist auch die gesamte Integritätsformel gültig.

- Für mindestens eine negative Komponente $\neg prä_u(ob_1,...,ob_{Ku})$ aus der Integritätsformel existiert in der Faktenmenge FAK_r kein korrespondierendes Fakt $fakt_r(prä_{uf}(ob_1,...,ob_{Ku}))$. Dann ist die negative Formelkomponente $\neg prä_u(ob_1,...,ob_{Ku})$ mittelbar gültig, weil ihr Negat $prä_u(ob_1,...,ob_{Ku})$ nicht gültig ist. Folglich ist auch die gesamte Integritätsformel gültig.

Eine aktuelle Faktenmenge FAK_r ist dagegen bezüglich einer variablenfreien Integritätsformel genau dann unzulässig[77)], wenn alle Komponenten aus der Integritätsformel für diese Faktenmenge unzulässig sind. Denn ein Adjugat ist genau dann ungültig, wenn alle seine Komponenten ungültig sind. Folglich ist eine Faktenmenge bezüglich einer Integritätsformel genau dann unzulässig, wenn beide nachfolgenden Sachverhalte gemeinsam zutreffen:

- Für jede positive Komponente $prä_u(ob_1,...,ob_{Ku})$ aus der Integritätsformel existiert in der Faktenmenge FAK_r kein korrespondierendes Fakt $fakt_r(prä_{uf}(ob_1,...,ob_{Ku}))$. Dann sind alle positiven Formelkomponenten $prä_u(ob_1,...,ob_{Ku})$ unmittelbar ungültig.

- Für jede negative Komponente $\neg prä_u(ob_1,...,ob_{Ku})$ aus einer negativen Integritätsformel existiert in der Faktenmenge FAK_r ein korrespondierendes Fakt $fakt_r(prä_{uf}(ob_1,...,ob_{Ku}))$. Dann sind alle negativen Formelkomponenten $\neg prä_u(ob_1,...,ob_{Ku})$ mittelbar ungültig, weil ihre Negate $prä_u(ob_1,...,ob_{Ku})$ gültig sind.

Eine Faktenmenge ist genau dann insgesamt zulässig, wenn sie für alle Integritätsformeln von Integritätsbedingungen zulässig ist, d.h. wenn sie alle Integritätsbedingungen erfüllt[78)]. Sie ist dagegen genau dann insgesamt unzulässig, wenn sie für mindestens eine Integritätsformel einer Integritätsbedingung unzulässig ist, d.h. wenn sie mindestens eine Integritätsbedingung verletzt. Daher kann jede aktuelle Faktenmenge FAK_r eindeutig als zulässig oder unzulässig identifiziert werden. Darüber hinaus konnten die Zulässigkeit und Unzulässigkeit von Faktenmengen durch Integritätsbedingungen so definiert werden, daß in der Definition nur noch auf atomare Formeln und deren Negate sowie auf Fakten Bezug genommen wird, nicht aber mehr auf die Gültigkeit oder Ungültigkeit der Integritätsformeln. Dies verdeutlicht nochmals die Leistungsfähigkeit des

Faktenkonzepts bei der objektsprachlichen Darstellung von Wissen über die metasprachlichen Wahrheitswerte "gültig" und "ungültig"[79].

B) Der Wahrheitswert einer prädikatenlogischen Satzvariable bleibt unter einer Interpretation I_r zunächst unbestimmt. Da diese Satzvariable (offene Formel) mindestens eine freie Variable enthält, kann sie verschiedene Wahrheitswerte annehmen je nachdem, welche Konstanten für diese Variable(n) eingesetzt werden[80]. Eine Satzvariable nimmt erst dann einen definiten Wahrheitswert an, wenn jede ihrer freien Variablen durch genau eine Konstante aus dem Definitionsbereich DB substituiert wird[81]. Diese Variablensubstitutionen sind durch die Interpretation I_r einer prädikatenlogischen Semantik aber noch nicht festgelegt. Daher leistet die oben definierte Semantik nur eine partielle Interpretation des prädikatenlogischen Kalküls.

C) Die Wahrheitswerte aller Formeln aus einem prädikatenlogischen Formelsystem liegen erst dann eindeutig fest, wenn die Interpretation I_r durch eine Variablenbelegung ergänzt wird[82]. Durch eine Variablenbelegung V_c[83] werden die Vorkommnisse von freien Variablen aus Satzformeln durch jeweils eine Konstante aus dem Definitionsbereich DB der Interpretation I_r ersetzt[84]. Dieser Substitutionsprozeß wird auch als eine Variablenbindung oder Instantiierung[85] von Variablen bezeichnet. Da alle Vorkommnisse derselben Variable durch dieselbe Konstante substituiert werden, liegt eine globale Variablenbindung vor[86]. Falls in einem prädikatenlogischen Formelsystem insgesamt K verschiedene Variablen X_k mit $k \in \{1,...,K\}$ und $K \in \mathcal{N}_+$ definiert sind, fällt die Menge ZV_r zulässiger Variablenbelegungen mit dem K-fachen kartesischen Produkt über dem Definitionsbereich DB der Interpretation I_r zusammen: $ZV_r = DB_r^K$. Durch jede zulässige Variablenbelegung V_c mit $V_c \in ZV_r$ wird eine offene Formel p_j auf die geschlossene Formel $V_c(p_u)$ abgebildet. Die Kombination aus einer Interpretation I_r und einer Variablenbelegung liefert eine vollständige Interpretation des prädikatenlogischen Kalküls.

D) Die Variablenbindung wird seitens der Programmiersprache PROLOG, die hier zur Implementierung der Prädikatenlogik herangezogen wird, durch das kombinierte Unifizierungs- und Resolutionskonzept realisiert[87]. Da dieses Bindungskonzept durch die prozedurale Kontrollstruktur von PROLOG verwirklicht wird, braucht hier auf die prädikatenlogischen Ansätze zur Variablenbindung nicht näher eingegangen zu werden.

E) Bemerkenswert ist aber die enge Beziehung zwischen den prädikatenlogischen Variablenbelegungen auf der einen und den Komplementauswertungsfunktionen des Signaturkonzepts auf der anderen Seite. Diese Korrespondenz beruht darauf, daß sich jede Variablenbelegung V_c als eine - komplex definierte - Funktion aufassen läßt, welche die freien Variablen eines prädikatenlogischen Formelsystems in globaler Weise auf Konstanten abbildet[88]. Damit entspricht sie der Komplementauswertungsfunktion $keval_i$, die im Rahmen des Signaturkonzepts eingeführt wurde. Denn jede interpretierte prädikatenlogische Formel setzt - aus der Perspektive des Signaturkonzepts - voraus, daß ihre Terme bereits mit der Hilfe von minimalen Teilauswertungsfunktionen auf teilevaluierte Terme abgebildet worden sind. Daher entspricht die Substitution aller verbliebenen Variablen durch Konstanten der Anwendung einer maximalen Komplementauswertungsfunktion. Allerdings bestehen zwei Unterschiede zwischen prädikatenlogischer Variablenbelegung und signaturzugehöriger Komplementauswertungsfunktion. Einerseits berücksichtigt die Komplementauswertungsfunktion nicht die prädikatenlogische Besonderheit, daß die Variablensubstitution durch Konstanten nur auf *freie* Variablen angewendet werden darf. Andererseits läßt die Variablenbelegung - zumindest im Rahmen der konventionellen Prädikatenlogik - keine sortengerechte Variablensubstitution zu. Diese Differenzen werden jedoch später aufgelöst, wenn zu einer sortierten Prädikatenlogik und explizit notierten Variablenbindungsfunktionen übergegangen wird.

F) In der Prädikatenlogik dominieren die Formulierung von prädikatenlogischen Formeln und die Suche nach denjenigen Argumenten, für welche die interpretierten Prädikate gültig sind. Daher kann von einer prädikatenlogischen Fokussierung auf Prädikate und deren Extensionen gesprochen werden. Diese Schwerpunktsetzung erweist sich aus der Perspektive des Petrinetz-Konzepts als besonderes interessant. Denn durch die Stellen von Synthetischen Netzen werden prädikatenlogische Formeln repräsentiert, während die Netzmarkierungen die Extensionen dieser Formeln ausdrücken. Dieser Vorgriff auf die spätere Darstellung Synthetischer Netze rechtfertigt nochmals die Konzentration auf eine prädikatenlogische Objektmodellierung.

G) Darüber hinaus besteht aber auch eine konzeptionelle Diskrepanz zwischen prädikatenlogischer Semantik und Petrinetz-Konzept. Aus prädikatenlogischer Sicht werden zwar durchaus mehrere Interpretationen für dasselbe Formelsystem vorgesehen. Aber das wechselseitige Verhältnis dieser verschiedenen Interpretationen zueinander wird im allgemeinen nicht näher untersucht. Die Interpretationen werden zumeist nur als eine lockere, unstrukturierte Ansammlung von Interpretationsalternativen behandelt. Im Netzkonzept besteht dagegen zwischen den Markierungen eines Petrinetzes, die jeweils prädikatenlogischen Formelextensionen entsprechen[89)], ein wohldefinierter Zusammenhang. Er wird durch die Erreichbarkeitsrelation konstituiert. Der Überbrückung dieser Diskrepanz dienen die nachfolgenden Ausführungen.

H) Der Überleitung von unstrukturierten Interpretationsanhäufungen der Prädikatenlogik zur netzbasierten Verknüpfung von Formelextensionen durch eine Erreichbarkeitsrelation kommt in dieser Arbeit wesentliche Bedeutung zu[90)]. Auf ihr beruht die Erweiterung der konventionellen prädikatenlogischen Semantik um eine netzbasierte operationale Semantik. Ansatzpunkt dieser operationalen Semantik ist ein formales Übergangsschema, das die Transformation von Prädikatsextensionen durch Schaltakte von Netztransitionen vermittelt. Zur Herleitung dieses Schemas und zur Verdeutlichung seiner zentralen Rolle für netzbasierte Modellierungen wird zunächst die konventionelle Behandlung von Prädikatsextensionen vertieft. Im anschließenden Kapitel erfolgt eine Fortentwicklung zu zeitlich variablen Prädikatsextensionen, die sich im Rahmen der konventionellen Prädikatenlogik nur schwer darstellen lassen[91)].

I) Seitens der konventionellen Prädikatenlogik wird die Extension $EXT_{u.r}$ eines K_u-stelligen Prädikatssymbols $Prä_u$ gewöhnlich in der oben eingeführten Weise definiert. Es handelt sich um eine statische Semantik, weil eine zeitliche Variation der Prädikatsextension nicht vorgesehen ist. Der statische Charakter der konventionellen prädikatenlogischen Semantik läßt sich durch das Konzept der prädikatspezifischen Wahrheitsfunktionen besonders deutlich herausarbeiten. Die Wahrheitsfunktion[92)] einer[93)] uninterpretierten Grundtermformel[94)] $prä_u(gt_1,...,gt_{Ku})$ legt durch ihre Deklaration für alle[95)] kombinatorisch möglichen konstanten Formelargumente $(ob_1,...,ob_{Ku})$ fest, ob sie die Formel erfüllen oder nicht[96)]. Es handelt sich um eine (K_u+1)-stellige metasprachliche Funktion wf_u, die jedem objektsprachlichen Argument $(ob_1,...,ob_{Ku})$ auf einen Wahrheitswert aus der regulären[97)] BOOL'schen Menge $BOOL_2=\{gültig,ungültig\}$ der konventionellen[98)] prädikatenlogischen Wahrheitswerte abbildet. Mit DB als Definitionsbereich der prädikatenlogischen Formelinterpretation gilt z.B. für flach strukturierte Formelargumente:

$$
\begin{aligned}
wf_u\colon\quad & DB^{Ku} \rightarrow BOOL_2 \\
& (ob_1,...,ob_{Ku}) \rightarrow wf_u(ob_1,...,ob_{Ku}) \in \{gültig,ungültig\}
\end{aligned}
$$

Durch die Wahrheitsfunktion wf_u läßt sich die oben definierte Extension $EXT_{u.r}$ des Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ reformulieren zu[99)]:

$$EXT_{u.r} = \{(ob_1,...,ob_{Ku})\colon wf_u(ob_1,...,ob_{Ku}) = gültig\} \subseteq DB^{Ku}$$

Diese Reformulierung ersetzt zwar den oben verwendeten Ausdruck "$I_r(prä_u(ob_1,...,ob_{Ku}))$ = gültig" lediglich durch "$wf_u(ob_1,...,ob_{Ku})$ = gültig". Aber durch die Eliminierung des Indexes "r" innerhalb der Mengendeklaration offenbart sich der statische Charakter der Extensionsdefinition. Denn aus konventioneller prädikatenlogischer Sicht ist es unvorstellbar, daß eine Wahrheitsfunktion wf_u im Zeitablauf variiert. *Falls* die Wahrheitsfunktion a priori vollständig bekannt ist, liegt auch die hiervon abhängige Prädikatsextension $EXT_{u.r}$ von vornherein fest (statische Prädikatsextension)[100]: Der interpretationsabhängige Index "r" des Extensionssymbols "$EXT_{u.r}$" ist aus dieser Perspektive abundant. Die Extension des Prädikatssymbols $Prä_u$ wird dann durch seine Wahrheitsfunktion wf_u vollständig - d.h. vor allem auch: interpretationsunabhängig - beschrieben[101].

J) Wenn die Wahrheitsfunktion zunächst nicht vollständig bekannt ist, kann die Prädikatsextension $EXT_{u.r}$ im Zeitablauf variieren. Die Veränderung der Prädikatsextension hängt dabei nur von der Vervollständigung der Kenntnisse über die Wahrheitsfunktion ab. Das Spektrum aller möglichen Kenntnisvervollständigungen wird durch die prädikatenlogische Monotonieprämisse eingeschränkt. Diese Prämisse besitzt für die konventionelle Prädikatenlogik fundamentalen Charakter; sie spielt auch in dieser Arbeit mehrfach eine wichtige Rolle[102]. Ein Formelsystem heißt monoton[103], wenn jede Formel $prä_u(ob_1,...,ob_{Ku})$ für jedes Argument, für das sie einmal als gültig nachgewiesen wurde, auch in aller Zukunft[104] gültig bleibt. Ein Prädikatsvorkommnis $prä_u(ob_1,...,ob_{Ku})$, das als gültig erkannt wurde, bleibt also gültig unabhängig davon, welche Erkenntnisse danach über andere Prädikatsvorkommnisse noch gewonnen werden. Hieraus folgt einerseits die "Wahrheitserhaltung"[105] aller prädikatenlogisch zulässigen Schlußfolgerungen[106]: Ein Prädikatsvorkommnis bleibt gültig ("wahr") unabhängig von allen Schlußfolgerungen, die nach der Gültigkeitserkenntnis stattfinden. Andererseits müssen die Wahrheitsfunktionen in dem Ausmaß, in dem ihre Abbildungsvorschriften bereits bekannt sind, invariant gelten. Extensionsveränderungen sind allenfalls aufgrund einer monotonen Zunahme des Wissens über die Gestalt der Wahrheitsfunktion möglich. Daher kann die Prädikatsextension, die durch die Wahrheitsfunktion implizit definiert wird, im Zeitablauf lediglich "monoton" anwachsen[107].

K) Aus der Perspektive einer operationalen Logikkonzeption[108] kann es jedoch wünschenswert sein darzustellen, daß die Erfüllung einer Formel $prä_u$ durch Konstantentupel $(ob_1,...,ob_{Ku})$ zeitlich variiert. Eine solche Variabilität bildet die logische Basis der Schaltregel für Synthetische Netze. Daher muß der Ansatz zeitlich invarianter Definitionen von Prädikatsextensionen durch Wahrheitsfunktionen für die weitere Entfaltung dieses Netzkonzepts wieder aufgegeben werden. An die Stelle dieser Funktionen wird die zeitlich variable Definition der Extension eines Prädikats mit der Hilfe von Fakten treten. Diese operationale Erweiterung der konventionellen Prädikatenlogik durch variable Faktenmengen wird später ausführlicher behandelt.

L) Bei der Verwendung von Wahrheitsfunktionen werden - im Gegensatz zur früheren Erklärung der Prädikatsextension - nicht mehr alle prädikatserfüllenden Argumente $(ob_1,...,ob_{Ku})$ mit $(ob_1,...,ob_{Ku}) \in EXT_{u.r}$ aufgelistet. An die Stelle dieser expliziten Extensionsnotation tritt dann die implizite Definition der prädikatserfüllenden Argumente durch die Deklaration seiner Wahrheitsfunktion wf_u. Mit Hilfe dieser Funktion läßt sich für jedes präsentierte K_u-stellige Konstantentupel $(ob_1,...,ob_{Ku})$ entscheiden, ob es zur Extension prädikatserfüllender Argumente gehört oder nicht[109].

M) Mit Hilfe der Prädikatsextensionen lassen sich alle Relationen[110] in der Prädikatenlogik verankern[111]. Ein K_u-stelliges Relationssymbol[112] REL_u mit $K_u \in \mathcal{N}_+$ und $K_u \geq 2$ wird durch ein K_u-stelliges Prädikatssymbol $Prä_u$ in der Weise definiert, daß gilt: Die Relation REL_u ist eine Menge von konstanten K_u-Tupeln $(ob_1,...,ob_{Ku})$. Ein solches Tupel $(ob_1,...,ob_{Ku})$ gehört genau dann zur Relation REL_u, wenn das Prädikat $prä_u(ob_1,...,ob_{Ku})$ gültig ist[113]:

$$REL_u = \{(ob_1,...,ob_{Ku}): wf_u(ob_1,...,ob_{Ku}) = \text{gültig}\}$$

Dies läßt sich äquivalent darstellen als:

$$\forall(ob_1 \in DB) \ldots \forall(ob_{Ku} \in DB): (ob_1,...,ob_{Ku}) \in REL_u \leftrightarrow prä_u(ob_1,..,ob_{Ku})$$

Daraus abgeleitet ist folgende Kurznotation, in der jedes Prädikatsvorkommnis als gültig unterstellt wird:

$$REL_u = \{(ob_1,...,ob_{Ku}): prä_u(ob_1,...,ob_{Ku})\}$$

Speziell für 2-stellige Relationen wird als vereinfachte Infixnotation definiert[114]:

$$ob_1\ REL_u\ ob_2 \quad :\Leftrightarrow \quad (ob_1,ob_2) \in REL_u \quad \Leftrightarrow \quad prä_u(ob_1,ob_2)$$

Das große Ausdruckspotential der Prädikatenlogik 1. Stufe zeigt sich darin, grundsätzlich alle mehrstelligen[115] mathematischen Relationen darstellen zu können.

N) Die prädikatenlogische Relationsdefinition läßt sich dadurch erweitern, daß von den statisch definierten Wahrheitsfunktionen zu den interpretationsabhängigen Prädikatsextensionen übergegangen wird. Dann wird eine K_u-stellige Relation REL_u durch die Extension $EXT_{u.r}$ des Prädikatssymbols $Prä_u$ unter der Interpretation I_r definiert. Die Relation REL_u ist dann die Menge aller konstanten K_u-Tupel, die zur Extension $EXT_{u.r}$ des Prädikatssymbols $Prä_u$ gehören:

$$REL_u = \{(ob_1,...,ob_{Ku}): (ob_1,...,ob_{Ku}) \in EXT_{u.r}\} = EXT_{u.r}$$

In dem Ausmaß, in dem die Prädikatsextension $EXT_{u.r}$ unter verschiedenen Extensionen variiert, verändert sich auch die extensional definierte Relation REL_u[116].

O) In dieser Arbeit werden grundsätzlich alle prädikatenlogisch definierbaren Formeln zugelassen. Die Definitionsvielfalt hängt einerseits vom syntaktischen Konstruktereichtum des prädikatenlogischen Kalküls ab. Dieser wurde im voranstehenden Kapitel aufgezeigt. Andererseits wird die Definitionsvielfalt von den Möglichkeiten bestimmt, unterschiedliche Bedeutungen für die Prädikatssymbole festzulegen. Dies geschieht in Gestalt der Prädikatsextensionen. Da für die Prädikatsextensionen keinerlei Einschränkungen vorgenommen wurden, kann im Rahmen der Prädikatenlogik grundsätzlich jede Prädikatsbedeutung ausgedrückt werden. Dies gilt allerdings nur in dem Ausmaß, wie sich die Prädikatsbedeutungen durch die explizite Angabe aller Argumente definieren lassen, für welche die Prädikate jeweils gültig sind. Dies entspricht der früheren Prämisse, ausschließlich extensional definierte Konstrukte zu betrachten.

P) Es wird nicht versucht, alle in dieser Arbeit verwendeten Prädikate explizit zu definieren. Vielmehr werden Standardprädikate der Prädikatenlogik als bekannt vorausgesetzt. Hierzu gehören beispielsweise das arithmetische Identitäts-[117] und das mengentheoretische Elementprädikat[118]. Ebenso werden die relationentheoretischen Vergleichsprädikate[119] als bekannt unterstellt, die den früher eingeführten algebraischen Vergleichsformeln entsprechen. In dieser Arbeit wird unter "der" Prädikatenlogik stets ihre weit definierte Variante verstanden, die alle - voranstehend nur exemplarisch verdeutlichten - Standardprädikate umfaßt. Dies bedeutet beispielsweise, daß "die" Prädikatenlogik immer schon die arithmetische Ausdruckskraft umschließt[120].

Q) Mit Hilfe der Standardprädikate können im Bedarfsfall spezielle Prädikate durch den Definitionsoperator ":$\Leftrightarrow$"[121] eingeführt werden. In dem metasprachlichen Ausdruck:

$$\text{prä}_{\text{neu}}(te_1,...,te_{K\text{neu}}) \quad :\Leftrightarrow \quad \text{prä}_{\text{alt}}(te_1,...,te_{K\text{alt}})$$

wird die Bedeutung des neu einzuführenden Prädikats prä_{neu} (Definiendum) durch das bereits eingeführte Prädikat prä_{alt} (Definiens) in der Weise definiert, daß beide Prädikate unter allen Interpretationen jeweils dieselben Wahrheitswerte besitzen[122]. Dabei stellt das Definiendum immer eine atomare Formel dar. Für das Definiens wird dagegen jede beliebig komplex zusammengesetzte prädikatenlogische Formel zugelassen. Darüber hinaus werden im Definiens etablierte arithmetische sowie mengen- und relationentheoretische Notationen auch dann verwendet, wenn sie strenggenommen keine prädikatenlogischen Formeln darstellen. Diese Notationen lassen sich jeweils als Substitute für die ausführlichere oder ungewohntere Formulierung eines äquivalenten Prädikats auffassen[123].

R) Die prädikatenlogischen Formeln bedeuten eine Verallgemeinerung der algebraischen Formeln, die früher für die Formulierung der Restriktionenmenge RES_{SPEC} von SIG-Spezifikationen $SPEC_{SIG}$ eingeführt wurden. Denn die Prädikatsformulierung ist nicht mehr auf algebraische Relationen für die Formeldefinition eingeschränkt. Statt dessen kann das mächtigere Ausdruckspotential atomarer prädikatenlogischer Formeln ausgeschöpft werden. Hierfür wurden oben alle extensional definierbaren Prädikate zugelassen. Die Ausdruckserweiterungen erstrecken sich beispielsweise auf Formelquantifikationen[124]. Hierdurch lassen sich Prädikate definieren, die nicht vollständig auf die früher spezifizierten algebraischen Anordnungs- und Vergleichsformeln zurückgeführt werden können. Als pars pro toto werden zwei Extremierungsprädikate vorgestellt. Zugleich wird hierdurch die Möglichkeit demonstriert, aus bereits bekannten Prädikaten induktiv neue Prädikate zu definieren. Die zweistelligen Minimierungs- und Maximierungsprädikate "min" und "max" drücken jeweils die Eigenschaft eines Terms aus, in bezug auf eine vorgegebene Termmenge TM minimale bzw. maximale Elemente darzustellen[125]. Dabei wird auf die bereits eingeführten algebraischen Formeln mit den Vergleichsrelatoren "$\geq$" und "$\leq$" zurückgegriffen. Daraus lassen sich mit der Hilfe des Allquantors ableiten:

$$\begin{array}{lll} \min(te_k,TM) & :\Leftrightarrow & (te_k \in TM \wedge (\forall(te_h \in TM): te_h \geq te_k)) \\ \max(te_k,TM) & :\Leftrightarrow & (te_k \in TM \wedge (\forall(te_h \in TM): te_h \leq te_k)) \end{array}$$

Darüber hinaus können mit Hilfe der beiden Extremierungsprädikate zwei entsprechende Extremierungsfunktionen "min" und "max"[126] definiert werden. Sie liefern bei ihrer Anwendung auf eine Termmenge TM jeweils die Teilmenge aller minimalen bzw. maximalen Elemente dieser Termmenge. Mit MTM als Menge aller Termmengen gilt:

$$\begin{array}{ll} \min: & MTM \rightarrow MTM \\ & TM \rightarrow \min(TM) = \{te_k: te_k \in TM \wedge \min(te_k,TM)\} \\ \max: & MTM \rightarrow MTM \\ & TM \rightarrow \max(TM) = \{te_k: te_k \in TM \wedge \max(te_k,TM)\} \end{array}$$

S) Durch das semantische oder modelltheoretische Schlußfolgerungskonzept wird festgelegt, unter welchen Bedingungen eine Formel die logische Konsequenz einer vorgegebenen Formelmenge ist. Grob gesprochen kann eine Formel aus einer Formelmenge genau dann gefolgert werden, wenn die Formel unter allen Interpretationen gültig ist, unter denen auch die Formeln der Formelmenge gültig sind[127]. Das Konzept semantischer Schlußfolgerungen wird jedoch nicht weiter vertieft. Statt dessen wird es später - im Rahmen einer operationalen Semantik -

durch ein syntaktisches Inferenzkonzept ersetzt, das in der Kontrollstruktur der Programmiersprache PROLOG eingebettet ist. Immerhin bildet der semantische Schlußfolgerungsbegriff den theoretischen Maßstab, an dem sich jedes syntaktische Inferenzkonzept messen lassen muß[128]. Hierauf wird unter dem Aspekt der prädikatenlogischen Vollständigkeit und Korrektheit zurückgekommen.

T) Die Programmiersprache PROLOG erspart dem Modellierungsträger, sich im Detail um Techniken für die Realisierung von Variablenbindungen und Schlußfolgerungen kümmern zu müssen. Daher stellt sie aus dieser Perspektive ein herausragendes Instrument für die Implementierung prädikatenlogischer Modellierungen dar.

U) Allerdings stellt sich die Frage, ob PROLOG-Implementierungen das zuvor erläuterte prädikatenlogische Ausdruckspotential zu wahren vermögen. Denn die Formulierungsmächtigkeit der Prädikatenlogik 1. Stufe wird dadurch eingeschränkt, daß die PROLOG-Syntax als zusammengesetzte Formeln nur HORN-Klauseln[129] zuläßt[130]. Immerhin können die meisten[131] zusammengesetzten Prädikate in nicht-leere Mengen von HORN-Klauseln transformiert werden[132]. Dabei bleiben interessante Prädikatseigenschaften wie Allgemeingültigkeit, Erfüllbarkeit oder Inkonsistenz erhalten[133]. Darüber hinaus existieren bereits Ansätze, die Programmiersprache PROLOG so zu erweitern, daß sie - ohne Verlust ihres Inferenzpotentials - nicht mehr auf HORN-Klauseln beschränkt ist[134].

V) Aber selbst dann, wenn sich die Restriktionen der HORN-Klausel-Syntax der Programmiersprache PROLOG beim praktischen Umgang mit prädikatenlogischen Objektmodellen restriktiv auswirken, so liegt hierin jedoch *keine* Einschränkung für das Ausdruckspotential von Netzmodellen. Dies hat zwei Gründe. Erstens werden nicht die prädikatenlogisch formulierten Objektmodelle selbst auf PROLOG-Basis implementiert, sondern die daraus abgeleiteten Netzmodelle. Also sind die Objektmodelle nicht der Einschränkung auf HORN-Klauseln unterworfen. Zweitens werden die Netzmodelle in das Softwarepaket PASIPP so eingebettet, daß zwei scheinbar widersprüchliche Anforderungen erfüllt werden: Einerseits können die Netzmodelle jede beliebige prädikatenlogische Formel repräsentieren. Andererseits halten aber die netzrepräsentierenden PASIPP-Konstrukte auch die HORN-Klausel-Syntax von PROLOG ein. Diese prima facie paradoxe, aber hochwillkommene Eigenschaft von Netzmodellen resultiert letztlich aus dem Umstand, daß die Repräsentation prädikatenlogischer Formeln durch ein Synthetisches Netz auf einer anderen Darstellungsebene geschieht als die PROLOG-basierte Implementierung von Netzkonstrukten.

Ausgangspunkt ist ein Objektmodell, das ein System beliebiger prädikatenlogischer Formeln darstellt. *Jedes* prädikatenlogische Formelsystem läßt sich in eine logisch äquivalente konjunktive Normalform transformieren. Auf eine solche konjunktive Normalform wird ein Transformationsalgorithmus angewendet, der später detailliert vorgestellt wird. Er liefert ein - abermals logisch äquivalentes - Netzmodell. Basiskonstrukt dieser Modelltransformation sind logische Formeln vom *allgemeinen* Klauseltyp, die nicht auf die speziellen HORN-Klauseln der PROLOG-Syntax eingeschränkt sind. Jede solche allgemeine Klausel wird im Netzmodell auf die Transition eines Synthetischen Netzes abgebildet. Daher erfolgt bei der Konstruktion der Netzmodells *kein* Abstrich an der prädikatenlogischen Ausdruckskraft. Die Transitionen, die Schaltregel für ihre Übergangsoperationen und die Transitionenverknüpfungen, die durch die Netztopologie bestimmt sind, werden danach ohne Informationsverlust durch umfangreichere PROLOG-Konstrukte implementiert[135]. Folglich kann jedes - beliebig strukturierte - prädikatenlogische Formelsystem letztlich in eine logisch äquivalente PROLOG-Repräsentation übersetzt werden. Dabei wird die syntaktische Restriktion der HORN-Klauseln durch die spezielle Konstruktions- und Implementierungsweise von Netzmodellen überwunden. Denn die PROLOG-Implementierung eines Netzmodells kommt mit der Verwendung von HORN-Klauseln aus, obwohl das Netzmodell selbst beliebige prädikatenlogische Formeln repräsentieren kann.

Folglich spielt in dieser Arbeit die Einschränkung von PROLOG-Programmen auf HORN-Klauseln hinsichtlich der Ausdrucksmächtigkeit von Netzmodellen keine Rolle. Hierin liegt ein wesentlicher Beitrag des hier entfalteten Modellierungskonzepts, die Kluft zwischen möglichst ausdrucksstarken und zugleich effizient auswertbaren Modellen zu überwinden[136)]. Denn in PROLOG-basierten Netzmodellen wird die prädikatenlogische Ausdruckskraft mit der Auswertungseffizienz Automatischer Informationsverarbeitungssysteme kombiniert.

Anmerkungen zum Kapitel:

1) Es erfolgt nur eine unwesentliche taxonomische Variation. Die aussagenlogischen Wahrheitswerte "wahr" und "falsch" werden durch ihre prädikatenlogischen Pendants "gültig" bzw. "ungültig" ersetzt.

2) Vgl. zu Überblicken über prädikatenlogische Semantiken oder Interpretationen CARNAP (1960a), S. 42ff. u. 95ff.; HERMES (1961), S. 158; MENDELSON (1964), S. 49; SMULLYAN (1968), S. 46ff., insbesondere S. 48; STEGMÜLLER (1969), S. 18ff.; STEGMÜLLER (1973a), S. 54; CHANG,C.L. (1973), S. 31; DIN 5474 (1973), S. 6; ITZINGER (1976), S. 30f.; RICHTER,M.M. (1978), S. 59f. u. 63ff.; BOOLOS (1980), S. 98ff.; MCDERMOTT (1980), S. 52; POTTHOFF (1981), S. 9f.; ESSLER (1982a), S. 40ff.; STEGMÜLLER (1983), S. 56ff.; REITER (1984), S. 196; REINFRANK (1985b), S. 27f.; DELAHAYE (1987), S. 73ff.; PREIß (1989), S. 16ff.

3) Diese Ersetzung ist dem Verf. aus prädikatenlogischen Abhandlungen nicht bekannt. Auf ihr Erfordernis wird an späterer Stelle zurückgekommen.

4) Diese Abbildung findet sich in prädikatenlogischen Abhandlungen nur äußerst selten; vgl. zu den wenigen Ausnahmen etwa DELAHAYE (1987), S. 73; PREIß (1989), S. 16. Dies liegt an der mangelhaften Differenzierung zwischen Funktionssymbolen und Funktionen in prädikatenlogischen Kontexten, auf die bereits hingewiesen wurde.

5) Es wurde bereits dargelegt, daß sich jedes prädikatenlogische Formelsystem entweder als genau eine komplex strukturierte Konjugat-Formel auffassen läßt oder von vornherein aus nur einer Formel besteht. Nachfolgende Darlegungen erstrecken sich daher auch auf die Interpretation einzelner prädikatenlogischer Formeln.

6) Der Index "r" wird später in bezug auf die operationale Semantik von prädikatenlogisch fundierten Netzmodellen durch diejenige Markierung M_r erklärt, die mit jeweils einer prädikatenlogischen Interpretation I_r korrespondiert.

7) Insbesondere braucht keineswegs mit $R \in \mathcal{N}_+$ eine endliche Interpretationsanzahl unterstellt werden. Vielmehr kann z.B. ein unendlicher Definitionsbereich DB für die Variablen eines prädikatenlogischen Formelsystems (vgl. die nachfolgende Erläuterung) dazu führen, daß dieses Formelsystem unendlich viele verschiedene Interpretationen besitzt. So werden z.B. schon von GÖDEL (1930), S. 350, Fn. 4, abzählbare Individuenbereiche zugelassen. Diese bedeuten in der Terminologie dieser Arbeit abzählbar unendliche Definitionsbereiche DB.

8) Die Definitionskomplexität resultiert aus den unterschiedlichen objektsprachlichen Ausdrücken, auf die eine Interpretation I_r angewendet werden muß, sowie aus der Verschiedenartigkeit der hierbei zugeordneten formalen Objekte 2. Stufe. Vgl. zu diesen Komplexitätsdeterminanten die eingangs erfolgte Definition prädikatenlogischer Semantik durch Interpretation I_r.

9) Auf die Gemeinsamkeiten - aber vor allem auch Unterschiede - zwischen prädikatenlogischen Interpretationen und signaturbezogenen Auswertungsfunktionen wird an anderer Stelle näher eingegangen.

10) Diese Entsprechung wird besonders deutlich die Darstellung von RICHTER,M.M. (1978), S. 59f. u. 63f., der die prädikatenlogischen Interpretationen I_r von vornherein als komplexe Auswertungsfunktionen definiert, ohne das prädikatenlogische Interpretationskonzept inhaltlich zu verlassen.

11) Diese Konstanten entsprechen den formalen Objekten aus den Objektmengen OB_i einer SIG-Algebra. Oftmals wird auch von Individuen gesprochen. Diese Ausdrucksweise ist jedoch insofern unscharf, weil Individuen sowohl als Individuen-Konstanten als auch als Individuen-Variablen auftreten können. Wenn keine nähere Charakterisierung erfolgt, werden im allgemeinen Individuen mit Individuen-Konstanten gleichgesetzt.

12) Das prädikatenlogische Universum entspricht dem algebraischen Universum des Signaturkonzepts, sofern von der Sortenstruktur des letztgenannten abgesehen wird. Das prädikatenlogische Universum wird auch als Interpretations-Domäne (domain), Individuen-Universum oder Objektbereich bezeichnet.

13) Auf die Teilevaluation von Termen wird an anderer Stelle ausführlicher eingegangen.

14) Die abweichende Ansicht, durch eine prädikatenlogische Semantik werde jeder Variable eine Konstante (ein "Wert") zugeordnet, vertreten allerdings MCDERMOTT (1979), S. 566; MCDERMOTT (1980), S. 52; REINFRANK (1985b), S. 27f.

15) Im Gegensatz zum Signaturkonzept können jedoch in der Prädikatenlogik keine unterschiedlichen Sorten berücksichtigt werden. Diese Einschränkung wird später durch den Übergang zu einer sortierten Prädikatenlogik überwunden

16) Darüber hinaus sind die kanonischen Erweiterungen der minimalen Teilauswertungsfunktionen betroffen, sobald eine mehrstellige prädikatenlogische Formel interpretiert wird.

17) Wenn aus dem Argumentationszusammenhang ersichtlich ist, daß eine prädikatenlogische Interpretation thematisiert wird, lassen sich teilevaluierte Terme vereinfacht als Terme ansprechen.

18) In dieser Notationsweise liegt eine formale Unzulänglichkeit. Denn im Fall eines Grundterms werden zwei kategorial verschiedene Konstrukte ohne formale Differenzierung dargestellt: Sowohl der syntaktisch definierte Grundterm als auch das semantisch definierte formale Objekt, das aus der Teilevaluierung des Grundterms hervorgeht, werden mit "te" notiert. Allerdings sieht der Verf. auch keine Alternative, diesen Formulierungsmangel zu beheben, ohne hierdurch andere Unzulänglichkeiten in Kauf nehmen zu müssen.

19) Als alternative Formulierungen kommen "Gültigkeitsstati" oder "Bestätigungsmaße" in Betracht. Vgl. zu Übersichten über die Vielfalt denkmöglicher Bestätigungsmaße CARNAP (1959a), S. 23ff.; GÄNG (1967), S. 77; RICHTER,M.M. (1978), S. 41 i.V.m. S. 39 u. 22ff.; RAUTENBERG (1979), S. 2, 8, 105ff. u. 114ff.; ZELEWSKI (1986a), S. 98. Da in dieser Arbeit die Prädikatenlogik 1. Stufe als formallogisches Fundament ausreicht, werden hier nur die "klassischen" zweiwertigen Bestätigungsmaße berücksichtigt. Vgl. zu einer eingehenden Erörterung dieser Zweiwertigkeit der klassischen Formallogik RAUTENBERG (1979), S. 7f. Infolge der breiten Etablierung des Wahrheitswertbegriffs wird an diesem anstatt der o.a. Alternativformulierungen festgehalten.

20) Ihre tatsächliche Bedeutung wird erst aus der näheren Betrachtung der Prädikatsextensionen deutlich. Denn deren Definitionen setzen Wahrheitswerte voraus; vgl. dazu die Erläuterungen zu Prädikatsextensionen sowie zu Wahrheitsfunktionen. Darüber hinaus folgt aus der Semantik von Quantoren und der Definition geschlossener Formeln, daß jede geschlossene prädikatenlogische Formel unter jeder Interpretation genau einen wohldefinierten Wahrheitswert besitzt; vgl. DELAHAYE (1987), S. 76.

21) Im Rahmen der Prädikatenlogik werden die aussagenlogischen Wahrheitswerte "wahr" und "falsch" durch die Wahrheitswerte "gültig" bzw. "ungültig" ersetzt. Trotz dieser veränderten Terminologie liegt keine materiell bedeutsame Veränderung der Bedeutung der aussagenlogischen Wahrheitswerte vor. Daher wird auch im prädikatenlogischen Fall weiterhin von Wahrheitswerten gesprochen.
Eine differenziertere Betrachtung aussagen- und prädikatenlogischer Wahrheitswerte findet sich bei LORENZEN,P. (1987), S. 211 i.V.m. S. 155ff., 177ff. u. 210f. Sie umfaßt insgesamt 6 Varianten des Wahrheitswerts "wahr" bzw. "gültig". Konzeptionell verweist sie auf ähnliche, aber noch nicht so weit ausgebaute Differenzierungen bei CARNAP (1959b), S. 40. LORENZEN's Versionen des Wahrheitswerts "gültig" unterscheiden sich aber nicht wesentlich von ihrem BOOL'schen Pendant. Vielmehr differenzieren sie nur unterschiedliche Entstehungsmodi und Reichweiten gültiger Formeln. Sie werden daher hier nicht weiter berücksichtigt. Allerdings fließt der Reichweitenaspekt in die Unterscheidung zwischen Erfüllbarkeit, Gültigkeit und Allgemeingültigkeit prädikatenlogischer Formeln ein. Eine tiefgreifende und subtile Analyse des formalsprachlichen Wahrheitsbegriffs findet sich bei STEGMÜLLER (1968), insbesondere S. 43ff., 52ff., 197ff. u. 215ff.; vgl. zu ähnlichen Begriffsexplorationen WHITEHEAD (1925), S. 41ff.; STEGMÜLLER (1984b), S. 375ff.

22) Wenn zwischen den aussagen- und prädikatenlogischen Wahrheitswertemengen sowie späteren dreiwertigen Erweiterungen differenziert werden soll, können die präzisierenden Notationen $bool_{2.aus}$ = {wahr,falsch} und $bool_{2.prä}$ = {gültig,ungültig} verwendet werden.

23) Durch die metasprachliche Notationsweise wird der früheren Explizitheitsforderung vorläufig Rechnung getragen. Dagegen ist es bei den meisten formallogischen Darstellungen üblich, die Semantik objektsprachlicher Formelsysteme nur implizit auszudrücken. Dann werden die Gültigkeit und die Ungültigkeit einer Formel dadurch ausgedrückt, daß eben diese Formel selbst bzw. deren Negat notiert werden. Zwar wird auf diese implizite Semantik nur selten hingewiesen; vgl. zu einer der seltenen Ausnahmen ihrer ausdrücklichen Erwähnung SMULLYAN (1968), S. 20f. Doch dominiert sie weithin. Für die implizite Notation der Semantik eines Formelsystems spricht die erhebliche Vereinfachung, die aus der Vermeidung expliziter Wahrheitswerte resultiert. Diesem Vereinfachungsargument stimmt der Verf. grundsätzlich zu. Daher wird er später bei der Verwendung von Faktenmengen und bei der prädikatenlogischen Implementierung von Synthetischen Netzen durch die Programmiersprache PROLOG zur impliziten Notation von Wahrheitswerten übergehen. Dennoch hält er vorerst noch an der expliziten Darstellung von Wahrheitswerten fest, um die konzeptionellen Grundlagen der prädikatenlogischen Semantik klarer behandelt zu können. Gleiches gilt für die Diskussion dreiwertiger Logiken. Dies entspricht in beiden Fällen dem Grundsatz der kontrollierten Explizitheit, der an früherer Stelle eingeführt wurde.
Eine abweichende, aber ebenfalls explizite Notation von Wahrheitswerten findet sich bei CARNAP (1960a), S. 95, 97 u. 99; SMULLYAN (1968), S. 15; RICHTER,M.M. (1978), S. 38. Dort werden die logischen Konstanten "t", "w" oder "W" einerseits und "f" oder "F" andererseits verwendet, um den interpretationsabhängigen Wahrheitswert einer Formel auszudrücken. Dabei wird allerdings - im Gegensatz zur hier bevorzugten Darstellungsweise - die jeweils zugrundegelegte Interpretation I_r nicht offengelegt. Beispielsweise entspricht die Notation W($prä_u$) der oben eingeführten Schreibweise I_r($prä_u$) = gültig.
Ebenso kann die Gültigkeit einer Formel durch FREGEs Assertionszeichen " $\vdash$" ausgedrückt werden, das bereits in einer früheren Anmerkung vorgestellt wurde. Denn die Zustimmung zu einer Formel wird zumeist mit deren Gültigkeit (Wahrheit) identifiziert; vgl. insbesondere von RUSSELL und WHITEHEAD in WHITEHEAD (1925), S. 8ff., 17f. u. 92ff. Dennoch wird das Assertionszeichen in dieser Arbeit nicht als metasprachliche Notation für den semantischen Sachverhalt der Gültigkeit einer interpretierten Formel verwendet. Denn ein Zeichen mit nahezu der gleichen

Gestalt wird oftmals benutzt, um den syntaktischen Sachverhalt auszudrücken, daß eine Formel innerhalb eines Beweissystems abgeleitet werden kann, ohne hierbei auf semantisch definierte Formelgültigkeiten zurückgreifen zu müssen. Auch der Verf. wird später auf dieses syntaktische Ableitbarkeitssymbol zurückgreifen. Daher widmet er hier dem Assertionszeichen FREGEs keine weitere Beachtung. Die beiden verschiedenartigen Sachverhalte, daß eine Formel entweder unter einer Interpretation den Wahrheitswert "gültig" annimmt (semantischer Aspekt) oder aber in einem Beweissystem abgeleitet werden kann (syntaktischer Aspekt) findet sich dagegen z.B. bei WHITEHEAD (1925), S. 92. Dort werden die Gültigkeit (Wahrheit) einer Formel "p" und ihre Ableitbarkeit in assertorischen Ausdrücken der Form "$\vdash$p" miteinander vermengt.

24) Vgl. zur Verwendung dieser beiden Notationen, die auch als Aussagekonstanten bezeichnet werden, DIN 5474 (1973), S. 2; LORENZEN,P. (1975a), S. 59; STEGMÜLLER (1983), S. 46ff.; SOWA (1984), S. 390; LORENZEN,P. (1987), S. 56; WEDEKIND (1989c), S. 21. Das Symbol "T" für immer gültige Tautologie kann als mnemotechnische Assoziation mit dem Anfangsbuchstaben des Worts "true" oder "tautologisch" angesehen werden. Das Symbol der Kontradiktion "$\perp$" repräsentiert die logisch konträre Aussagekonstante als bildhafte Umkehrung des Tautologie-Symbols.

25) Vgl. zur charakteristischen Extensionalität der Prädikatenlogik WHITEHEAD (1925), S. 72ff.; CARNAP (1960a), S. 95f. u. 99f.; STEGMÜLLER (1983), S. 94f.; REINFRANK (1985b), S. 30 (bezüglich der Junktoren); RICHTER,M.M. (1988), S. 21 u. 23; vgl. ebenso - aber in bezug auf die Aussagenlogik - RAUTENBERG (1979), S. 7ff.

26) Dies gilt jedoch nur für die hier dargestellte formale Semantik der Prädikatenlogik. Später wird für Netzmodelle auch eine informale denotationale Semantik entwickelt. Dort werden umgangssprachliche Prädikatsbedeutungen mit intensionaler Qualität formuliert.

27) Vgl. zur Definition von Prädikatsextensionen auch WHITEHEAD (1925), S. 76; CARNAP (1960a), S. 40, 42f. u. 96f.; REITER (1984), S. 196.

28) Im Grenzfall K=1 werden die degenerierten K-Tupel "(ob_1)" vereinfacht als Konstanten "ob" mit $ob=ob_1$ notiert. Die Menge der K-stelligen Konstantentupel kann auch leer sein.

29) Damit ist noch *nicht* festgelegt, daß jede Grundtermformel $prä_u(gt_1,...,gt_{K_u})$, die unter der Interpretation I_r für das Prädikatssymbol $Prä_u(st_1,...,st_{K_u})$ gültig ist, auch in der Extension $EXT_{u,r}$ enthalten sein müßte. Beispielsweise kann es Grundtermformeln geben, die zwar *an sich* gültig sind, deren Gültigkeit aber *für einen Modellierungsträger* noch nicht bekannt ist. Konsequenzen dieser keineswegs spitzfindigen, sondern subtilen Unterscheidung werden später unter den Aspekten der geschlossenen Weltmodellierung und einer dreiwertigen Logik ausführlicher thematisiert.

30) Falls alle Argumentkomponenten unmittelbar Konstanten aus dem Definitionsbereich darstellen, muß der Prädikatsextension $EXT_{u,r}$ ein Prädikatssymbol $Prä_u(st_1,...,st_{K_u})$ mit flach strukturiertem Argument zugrundeliegen. In diesem speziellen Fall ist die Prädikatsextension eine Teilmenge aus dem K_u-stelligen kartesischen Produkt DB^{K_u} des Definitionsbereichs DB: $EXT_{u,r}$ DB^{K_u}.
Jede K_u-stellige Relation über einer Menge M mit $K_u \in \mathcal{N}_+$ kann als eine Teilmenge des K_u-fachen Kartesischen Produkts M^{K_u} über dieser Referenzmenge aufgefaßt werden. Daher läßt sich im vorgenannten Spezialfall flach strukturierter Prädikatsargumente die Prädikatsextension in äquivalenter Weise als Relation über der Menge DB^{K_u} ansehen. Im Grenzfall einer 1-stelligen "Relation" mit $K_u=1$ stellt die Prädikatsextension eine Teilmenge der Menge DB^{K_u} dar. Diese relationale Ausdeutung von Prädikatsextensionen dominiert in der prädikatenlogischen Fachliteratur; vgl. MENDELSON (1964), S. 49; SMULLYAN (1968), S. 48; STEGMÜLLER (1984b), S. 37; REINFRANK (1985b), S. 27; LEE,R. (1988a), S. 225; PREIß (1989), S. 16. Eine abweichende, funktionale Behandlung von Prädikatsextensionen folgt dagegen aus der Auffassung von STETTER (1988), S. 47, daß es sich bei Prädikaten um eine spezielle Art von Funktionen handele. Dieser Ansicht folgt der Verf. jedoch nicht. Vgl. dazu den Hinweis auf den metasprachlichen Charakter der Wahrheitsfunktionen von Prädikaten.
Die Dominanz der relationalen Definition von Prädikatsextensionen verdeutlicht, daß in konventionellen prädikatenlogischen Darstellungen zumeist nur flache Prädikatsargumente berücksichtigt werden. Denn im Falle eines tief strukturierten Prädikatsarguments kann die Prädikatsextension nicht mehr mit Hilfe der kartesischen Produktmenge DB^{K_u} ausgedrückt werden. Vielmehr liegt dann eine K_u-stellige Verschachtelung des Definitionsbereichs DB vor, welche die Argumentzusammensetzung aus Funktoren widerspiegelt. Diese Bereichsverschachtelung, die zur korrekten semantischen Erfassung von Prädikaten mit tief strukturierten Argumenten zwingend erforderlich ist, hat der Verf. aber in keinem prädikatenlogischen Standardwerk auffinden können. Dies läßt erkennen, daß der tiefe Strukturierungsansatz von Signaturkonzept und objektorientierter Systemgestaltung in dieser Arbeit tatsächlich eine Bereicherung der Prädikatenlogik erwarten läßt.

31) Zusammengesetzte Formeln werden hier nicht gesondert behandelt. Denn für sie gilt analog: Jede zusammengesetzte Formel, die im prädikatenlogischen Kalkül mit der Hilfe von Konnektoren und Operatoren formiert wurde, wird durch die Interpretation einer prädikatenlogischen Semantik auf eine zusammengesetzte Formel abgebildet, deren Argumente aus teilevaluierten Termen bestehen.

32) Im Rahmen des prädikatenlogischen Kalküls wurden uninterpretierte atomare Grundtermformeln $prä_u(gt_1,...,gt_{Ku})$ und - als Spezialfall - $prä_u(Ko_1,...,Ko_{Ku})$ eingeführt, die nur aus variablenfreien Grundtermen bzw. Konstantensymbolen bestanden. Demgegenüber liegen hier interpretierte atomare konstante Formeln $prä_u(ob_1,...,ob_{Ku})$ vor, deren Komponenten ob_k jeweils atomare oder zusammengesetzte formale Objekte darstellen. Diese formalen Objekte können letztlich immer auf Konstanten als atomare formale Objekte zurückgeführt werden. Der Einfachheit halber können uninterpretierte atomare Grundtermformeln und interpretierte atomare konstante Formeln auch als Grundtermformeln bzw. konstante Formeln angesprochen, wenn aus dem Argumentationskontext - wie z.B. aus der Formelnotation - ersichtlich ist, daß es sich jeweils um atomare Formeln handelt. Auf die kennzeichnenden Attribute "uninterpretiert" bzw. "interpretiert" kann ohnehin verzichtet werden, weil sie keinen konstitutiven, sondern nur verdeutlichenden Charakter besitzen. Denn Grundtermformeln stammen immer aus dem rein syntaktisch definierten, also noch nicht interpretierten Kalkül der Prädikatenlogik. Konstante Formeln können dagegen nur interpretierte Formeln darstellen, da die formalen Objekte in ihren konstanten Argumenten die Interpretation durch eine prädikatenlogische Semantik voraussetzen.

33) Die formalen Objekte stellen entweder unmittelbar Konstanten aus dem Definitionsbereich DB dar oder sind mittelbar durch Funktoren aus solchen Konstanten zusammengesetzt.

34) Vgl. dazu auch die Anmerkungen zur Wahrheitsdefinitheit.

35) Bereits an früherer Stelle wurde dargelegt, daß in einem prädikatenlogischen Kalkül dieselbe Formel mehrfach vorkommen darf. Daher gilt die Formelerfüllung für jede dieser Vorkommnisse derselben uninterpretierten Formel $prä_u(gt_1,...,gt_{Ku})$.

36) Die Formelerfüllung erstreckt sich also immer auf eine konstante Formel $prä_u(gt_1,...,gt_{Ku})$ aus dem zugrundeliegenden prädikatenlogischen Kalkül. Sie wird daher als syntaktisch definierte oder uninterpretierte Formel bezeichnet.

37) Vgl. MENDELSON (1964), S. 51 i.V.m. S. 50.

38) Wenn die Interpretation I_r, die der interpretierten Formel $prä_u(ob_1,...,ob_{Ku})$ zugrundeliegt, explizit ausgewiesen werden soll, wird von der I_r-Gültigkeit der Formel $prä_u$ gesprochen. Dies entspricht der späteren Definition der M_r-Aktivierung von Transitionen.

39) Die Formelgültigkeit bezieht sich daher stets auf interpretierte Formeln $prä_u(ob_1,...,ob_{Ku})$ aus der prädikatenlogischen Semantik.

40) Die Extension $EXT_{u,r}$ eines 0-stelligen Prädikatssymbols $Prä_u()$ unter einer Interpretation I_r wurde bereits eingangs als der Wahrheitswert "gültig" oder "ungültig" definiert, welcher der aussagenlogischen Formel $prä_u()$ unter dieser Interpretation zukommt. Der einfache Fall solcher 0-stelligen Prädikatssymbole und Formeln wird nachfolgend nicht weiter thematisiert.

41) Umgekehrt gilt für alle anderen konstanten Objekttupel $(ob_1,...,ob_{Ku})$, die nicht zur Extension EXT_j des Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ gehören, daß die interpretierten Formeln $prä_u(ob_1,...,ob_{Ku})$ hierfür unter der Interpretation I_r ungültig sind: $I_r(prä_u(ob_1,...,ob_{Ku}))$ = ungültig.

42) Eine Formel heißt wahrheitsdefinit, wenn ihr Wahrheitswert eindeutig entweder als "gültig" ("wahr") oder aber als "ungültig" ("falsch") bestimmt ist; vgl. WEDEKIND (1989c), S. 23.

43) Vgl. zur Wahrheitsdefinitheit prädikatenlogischer Formeln ohne freie Variablen MENDELSON (1964), S. 49.

44) Die aussagenlogischen Wahrheitswertetafeln werden auch als aussagenlogische Wahrheitsfunktionen bezeichnet. Sie gelten formelunabhängig und müssen von den später eingeführten, jeweils formelspezifischen prädikatenlogischen Wahrheitsfunktionen wf_j deutlich unterschieden werden. Die Wahrheitswertetafeln sind für die gesamte Aussagen- und Prädikatenlogik uniform definiert. Sie brauchen daher hier nicht explizit angeführt zu werden. Statt dessen wird auf die einschlägige Literatur verwiesen; vgl. z.B. WITTGENSTEIN (1921), S. 223ff.; MENDELSON (1964), S. 12ff.; RAMSEY,F. (1965), S. 6ff. u. 276; CARNAP (1968), S. 18f.; DIN 5474 (1973), S. 2; LORENZEN,P. (1975a), S. 60f.; OPP,K. (1976), S. 101ff.; ESSER,H. (1977a), S. 32ff.; RICHTER,M.M. (1978), S. 12f.; RAUTENBERG (1979), S. 8ff. u. 14; PUTNAM,H. (1982a), S. 10ff.; STEGMÜLLER (1983), S. 46f.; BUCHER (1987), S. 58ff.; GREWENDORF (1987), S. 319ff.; LORENZEN,P. (1987), S. 56f.; DELAHAYE (1987), S. 54; TIETZE,J. (1988), S. 7ff.
Vgl. zur wahrheitsfunktionalen Formulierungsalternative zu Wahrheitstafeln WHITEHEAD (1925), S. 8; MENDELSON (1964), S. 15, 17 u. 24ff.; RAMSEY,F. (1965), S. 7ff. u. 276; KÖRNER,S. (1968), S. 45ff.; SMULLYAN (1968), S. 9ff.; RICHTER,M.M. (1978), S. 13; RAUTENBERG (1979), S. 9 u. 14; STEGMÜLLER (1983), S. 45; DELAHAYE (1987), S. 53f.

45) Für die prädikatenlogischen Quantoren gelten folgende Wahrheitswertedefinitionen:

- Die geschlossene Formel (Allaussage) "$\forall(X \in DB)$: $p_u(X)$" ist genau dann gültig, wenn die konstante Formel $p_u(ob)$ für *alle* Konstanten ob mit $ob \in DB$ gültig ist.

- Die geschlossene Formel (Existenzaussage) "$\exists(X \in DB)$: $p_u(X)$" ist genau dann gültig, wenn die konstante Formel $p_u(ob)$ für *mindestens eine* Konstante ob mit ob$\in$DB gültig ist.
- Die geschlossene Formel (Einsaussage) "$\underline{\exists}(X \in DB)$: $p_u(X)$" ist genau dann gültig, wenn die konstante Formel $p_u(ob)$ für *genau eine* Konstante ob mit ob$\in$DB gültig ist.

Alle komplexer zusammengesetzten geschlossenen Formeln, die mehrere Quantoren für mehrere Variablen umfassen, können in rekursiver Weise auf die o.a. Wahrheitswertdefinitionen zurückgeführt werden. Vgl. zu den Wahrheitswertdefinitionen für All- und Existenzquantor z.B. WHITEHEAD (1925), S. 15; CARNAP (1960a), S. 34f. u. 38; MENDELSON (1964), S. 45; SMULLYAN (1968), S. 47; ESSER,H. (1977a), S. 44; GREWENDORF (1987), S. 354f.; DELAHAYE (1987), S. 74. Die Wahrheitswertdefinition für den Einsquantor wurde hier analog gebildet.

46) Zwischen dem prädikatenlogischen und dem netztheoretischen Faktenbegriff besteht kein inhaltlicher Zusammenhang. Fakten bedeuten im Kontext des Petrinetz-Konzepts "faktische" Transitionen, die niemals aktiviert sein dürfen; vgl. z.B. OBERWEIS (1988b), S. 303f. Trotz der Problematik von Äquivokationen hält der Verf. an beiden Faktenbegriffen fest, da es sich jeweils um etablierte Termini technici handelt. Der jeweils aktuelle Kontext determiniert, ob ein Fakt entweder in prädikatenlogischer oder aber in netzbezogener Weise auszulegen ist. Vgl. zu einer dritten Variante des Faktbegriffs, die in dieser Arbeit nicht weiter berücksichtigt wird, DORN (1989), S. 66ff. u. 90f.

47) Vgl. zur expliziten Kopplung von objektsprachlichen Formeln und ihrer metasprachlichen Gültigkeitsfeststellung im Faktenbegriff ZELEWSKI (1986a), S. 202f. u. 274f.; RUSPINI (1987a), S. 232f. Vgl. ebenso die Feststellung bei HARMON (1985), S. 48: "Facts take the form of logical expressions that consist of predicates and [truth]values. (Ergänzung [...] durch den Verf.).
Der Programmiersprache PROLOG liegt ein weiter gefaßter Faktenbegriff zugrunde. Er läßt im Argument eines Fakts sowohl formale Objekte (Konstanten) als auch Variablen zu; vgl. CORDES (1988), S. 11, 13 u. 185; KINNEBROCK (1988), S. 125. Der Verf. folgt dieser Auffassung nicht, weil hierdurch die Möglichkeit verloren ginge, Fakten *immer* als gültige Prädikatsvorkommnisse auszuzeichnen. Denn ein "Fakt", das mindestens eine Variable enthält, besitzt keine definiten Wahrheitswert.

48) Vereinfacht wird fortan ein Fakt auch als ein gültiges Prädikat bezeichnet. Umgekehrt kann eine prädikatenlogische Formel, deren Gültigkeit bekannt ist, als faktische Formel angesprochen werden.

49) Der Einfachheit halber werden - abgesehen von dieser Anmerkung - fortan Faktenmengen nur für K_u-stellige Prädikatssymbole und Formeln mit $K_u \in \mathcal{N}_+$ thematisiert. Der Grenzfall $K_u=0$ wird dabei jeweils implizit als analog behandelbar unterstellt. Die weitgehende Vernachlässigung dieses Grenzfalls läßt sich dadurch rechtfertigen, daß später in Netzmodellen nur noch mindestens 1-stellige Prädikatssymbole und Formeln eine Rolle spielen werden.

50) Dies gilt strenggenommen nur dann mit Sicherheit, wenn konsistente Formelsysteme betrachtet werden. Die Konsistenzprämisse wird in dieser Arbeit aber im allgemeinen als erfüllt vorausgesetzt.

51) Zwei beliebige Faktenmengen können niemals ein gemeinsames Element besitzen, weil die Faktenmengen $FAK_{\tau.u}$ - und somit auch die darin enthaltenen Fakten - jeweils für *verschiedene* Prädikatssymbole $Prä_u$ definiert sind.

52) Die Familie aller prädikatspezifischen Faktenmengen und die Vereinigungsmenge aller prädikatspezifischen Faktenmengen werden zwar unterschiedlich notiert, besitzen aber gleiche Extensionen. Infolge der früheren Prämisse, extensionsgleiche Entitäten miteinander zu identifizieren ist es hier zulässig, die Familie und die Vereinigungsmenge mit demselben Symbol "FAK_τ" zu notieren.

53) Der Faktbegriff bleibt durch seine Rückführung auf die Gültigkeit von Prädikaten - und äquivalent: auf Prädikatsextensionen - weiterhin extensional definiert. Der Übergang von der expliziten Wahrheitswertenotation auf die implizite Berücksichtigung von Wahrheitswerten durch Fakten bedeutet also keineswegs einen Rückfall in intensionale Konzeptkonstruktionen, die an früherer Stelle ausgegrenzt wurden. Statt dessen liegt eine indirekt extensionale Konstruktdefinition im Sinne des Prinzips kontrollierter Explizitheit vor.

54) Da Fakten stets konstante Formeln $prä_u(ob_1,...,ob_{K_u})$ darstellen, läßt sich die Ersetzung aller Funktions- und Konstantensymbole durch Funktionen bzw. Konstanten als semantische Interpretationsleistung in der Faktennotation nicht vermeiden.

55) Die Vermengung von objekt- und metasprachlichen Konstrukten auf derselben Programmiersprachenebene sowie die Unfähigkeit, metasprachliche Konstrukte wie Wahrheitswerte explizit zu repräsentieren, erachtet der Verf. durchaus als gravierende Mängel der Sprache PROLOG. Hinzu kommt Ungenauigkeit bei der Behandlung von Prädikaten und Funktionen, auf die bereits früher hingewiesen wurde und nachfolgend nochmals vertieft wird. Trotz dieser Mängel gilt aber wiederum das schon mehrfach bemühte "faute de mieux"-Argument: Es steht keine alternative Sprache für die Implementierung prädikatenlogischer Objektmodelle auf Automatischen Informationsverarbeitungssystemen zur Verfügung, welche die kritisierten PROLOG-Schwächen vermeidet, ohne ihre anderen, aber positiven Leistungsmerkmale einzubüßen.

56) Sie wird in dieser Arbeit nur dann benutzt, wenn der Unterscheidung zwischen objektsprachlichen Formeln und metasprachlichen Fakten kein besonderes Gewicht zukommt. Dies ist z.B. der Fall, wenn eine Menge ME_u aus formalen Objekten ob durch ein mengenspezifisches Prädikat $prä_u$ definiert wird. Hierfür gilt in ausführlicher Notation: $ME_u = \{ob: I_r(prä_u(ob)) = gültig\}$. Als vereinfachte Notation wird hierfür unter der Prämisse gültiger konstanter atomarer Formeln auch zugelassen: $ME_u = \{ob: prä_u(ob)\}$. Vgl. dazu auch die Erörterung, den Mengen- auf den Prädikatsbegriff zu reduzieren.

57) Dies ist in Netzmodellen vielfach der Fall. In dieser Arbeit wird gezeigt, wie sich alle Kenntnisse, die ein Modellierungsträger über den Zustand eines Objektmodells und deren Veränderungen besitzt, mit der Hilfe von Faktenmengen repräsentieren lassen. Faktenmengen werden daher in Netzmodellen ein zentrales Ausdrucksmittel darstellen.

58) Dies wurde schon in der vorangehenden Erläuterung aufgezeigt.

59) Die Idee für diese Repräsentationsweise hat der Verf. der Darstellung von Netzmarkierungen durch das Softwarepaket PASIPP entnommen; vgl. z.B. OBERWEIS (1988a), S. 5f.; OBERWEIS (1988b), S. 302. Allerdings wird dort der hier thematisierte Aspekt der Faktendarstellung nicht behandelt. Vielmehr geht es dort explizit nur um die prädikatenlogische Implementierung der Stellenmarkierungen von Prädikat/Transition-Netzen. Dies äußert sich darin, daß die vom Verf. eingeführte Formel fakt(te) dort nur als Formel von der Art "marke(stellenname)" oder als "marking(stellenname)" erscheint. Ein ähnlicher Ansatz zur objektsprachlichen Repräsentation von Fakten findet sich bei CORDES (1988), S. 172f.

60) Vgl. dazu die PASIPP-Konstrukte OBERWEIS (1988a), S. 6; OBERWEIS (1988b), S. 302; vgl. ebenso die spätere Notation Synthetischer Netze in einer PROLOG-orientierten Syntax.

61) Vgl. REINFRANK (1985b), S. 7; MORRIS,P. (1988), S. 385; BIBEL (1989), S. 53f.
Beispielhaft wird von REINFRANK die Notation "HOLDS(PROP(OBJS),SIT)" angeführt. Sie drückt durch das objektsprachliche Prädikat "HOLDS" aus, daß das Objekt "OBJS" in der Situation "SIT" die Eigenschaft "PROP" besitzt. Das Prädikat "HOLDS" stimmt mit der faktischen Formel $fakt_r(te)$ offensichtlich überein, wenn die Situation "SIT" mit der Interpretation I_r und die Objekteigenschaft "PROP(OBJS)" mit dem Term "te" gleichgesetzt werden.

62) Hierauf wurde bereits mehrfach hingewiesen.

63) Vgl. OBERWEIS (1988a), S. 5f. Ein analoges Beispiel liefert das Prädikat "findall(Kind,vater(kasimir,Kind), Kinderliste)" mit dem eingeschachtelten Prädikat "vater(V,P)", das von CORDES (1988), S. 60 i.V.m. S. 12f., dargestellt wird.

64) Die gleiche Vorgehensweise, die sich auf eine "Uminterpretation" von Prädikaten als Funktionen stützt, findet sich bei BIBEL (1989), S. 53f.

65) Von dieser Option wird immer dann Gebrauch gemacht, wenn das metasprachliche Wissen über gültige Vorkommnisse eines Prädikatssymbols in objektsprachlicher Weise mit Hilfe der speziellen Prädikatssymbole $Fakt_r(st)$ und Formeln $fakt_r(te)$ geschehen soll.

66) Eine ähnliche Explizierung leistet der Ausdruck "HOLDS", der bereits in einer früheren Anmerkung angesprochen wurde.

67) Davon sind im Regelfall keineswegs alle Formeln eines prädikatenlogischen Formelsystems betroffen. Vielmehr handelt es sich nur um atomare Formeln, aus denen die Repräsentation des Wissens eines Modellierungsträgers über ein prädikatenlogisch repräsentiertes Objektmodell aufgebaut wird. Bei der späteren Formulierung von Netzmodellen wird sich zeigen, daß es sich nur um diejenigen Formeln handelt, denen Prädikatssymbole zugrundeliegen, welche den Stellen eines Synthetischen Netzes zugeordnet sind. Für alle weiteren Formeln, die ebenso das Wissen über ein modelliertes Objekt ausdrücken, aber aus den atomaren Formeln zusammengesetzt sind, fallen keine Fakten an. Darüber hinaus beruht ein Netzmodell in seiner PROLOG-Implementierung noch auf weiteren prädikatenlogischen Formeln, die jedoch nicht der Objektmodellierung, sondern nur der Netzimplementierung dienen. Auch für diese Hilfsformeln werden keine Fakten erfaßt.
Es könnte an dieser Stelle die Frage auftauchen, wie mit der Zuordnung der Wahrheitswerte "ungültig" verfahren werden soll. Hierfür bieten sich grundsätzlich drei Alternativen an. Erstens können sie auch auf objektsprachlicher Ebene repräsentiert werden, indem ein zweiter ausgezeichneter Formeltyp eingeführt wird, der analog zu den Formeln $fakt_r(prä_{uf}(ob_1,...,ob_{Ku}))$ konstruiert ist, nunmehr aber auf ungültige konstante atomare Formeln $prä_u(ob_1,...,ob_{Ku})$ angewandt wird. Diese Option liegt der Einführung negativer Fakten zugrunde. Zweitens lassen sich ungültige Prädikatsvorkommnisse weiterhin in der konventionellen Weise durch prädikatenlogische Interpretationen I_r auf der metasprachlichen Ebene darstellen. Diese Alternative wäre aber inkohärent, weil sie der Intention der oben eingeführten Formeln $fakt_r(prä_{uf}(ob_1,...,ob_{Ku}))$ zuwiderliefe, die metasprachliche Notation von Wahrheitswerten durch eine objektsprachliche Notation zu ersetzen. Drittens kommt auch die radikale Möglichkeit in Betracht, den Wahrheitswert "ungültig" überhaupt nicht mehr zu repräsentieren. Obwohl diese Alternative prima facie als "Kastration" der

prädikatenlogischen Semantik und infolgedessen indiskutabel erscheinen mag, wird der Verf. genau diesen Weg später einschlagen. Dies wird an anderer Stelle ausführlich gerechtfertigt.

68) Den Vorteil, in der Sphäre der Objektsprache zu verbleiben, hebt auch BIBEL (1989), S. 53, hervor.

69) Dies beruht auf zwei Gründen. Erstens werden in PROLOG-Implementierungen Prädikate und Funktionen miteinander vermengt. Daher läßt sich vorstellen, daß von vornherein mit Formeln $fakt_r(prä_u(ob_1,...,ob_{K_u}))$ anstelle der korrekten Formulierungen $fakt_r(prä_{uf}(ob_1,...,ob_{K_u}))$ gearbeitet werden kann. Der Verf. hat dies aber nicht näher untersucht. Aber die Verwendung der Prädikatsverschachtelung "findall(Kind,vater(kasimir,Kind),Kinderliste)" bei CORDES (1988), S. 60, deutet darauf hin, daß ein solches Vorgehen möglich ist.
Zweitens kann in praxi so verfahren werden, daß alle Prädikate, deren faktischen Vorkommnisse mit der Hilfe von objektsprachlich repräsentieren Fakten behandelt werden sollen, von vornherein nicht als Prädikate, sondern als Funktionen eingeführt werden. Dann erspart sich der oben dargelegte Aufwand für die Differenzierung zwischen Prädikaten und ihren funktionalen Deutungen. Statt dessen werden dann erst das Prädikatssymbol $Fakt_r(st)$ und die atomaren Formel $fakt_r(te)$ als Prädikate behandelt. Diesen zweiten Weg habe z.B. OBERWEIS und SEIB in OBERWEIS (1988a), S. 5ff., eingeschlagen. Obwohl sie eingangs den Ausdruck "mietwilliger_kunde" explizit als Prädikat angesprochen haben, taucht dieser Ausdruck in der anschließenden Implementierung des Formelsystems *überhaupt nicht* als Prädikat auf. Vielmehr wird er dort ausschließlich als Funktionsname verwendet.
Dieser zweite Ansatz liegt den meisten praktischen Implementierungen von Prädikat/Transition-Netzen implizit zugrunde. Zwar spricht für ihn, die problematische Vermengung von Prädikaten und Funktionen praktisch zu vermeiden. Aber er leidet unter einem erheblichen theoretischen Defizit. Denn er bricht implizit mit der Basisprämisse, die seitens der Petrinetz-Theorie für alle Prädikat/Transition-Netze erhoben wird: Die Stellen dieser Netze repräsentieren grundsätzlich Prädikat(ssymbol)e. Bei der praktischen Verfahrensweise des zweiten Ansatzes wird hierin zwar verbal festgehalten; vgl. OBERWEIS (1988a), S. 5. Aber die tatsächliche prädikatenlogische Implementierung solcher Prädikat/Transition-Netze verletzt dann doch die theoretische Repräsentationsprämisse, weil die Stellen nicht mehr als Prädikatssymbole, sondern nur als Funktionssymbole aufgefaßt werden. (Darüber hinaus werden die Prädikats- und Funktionssymbole zumeist nur als Prädikate bzw. Funktionen behandelt.) Es besteht also eine fundamentale Inkohärenz zwischen theoretischer Repräsentationsprämisse und praktischer Netzimplementierung. Um diese Inkohärenz zu vermeiden, hat der Verf. oben die aufwendigere formale Differenzierung zwischen Prädikatssymbolen und ihren funktionalen Deutungen eingeführt. Erste können dann weiter den Stellen eines Netzes zugeordnet werden, während sich zweite für eine kompakte Netzimplementierung auf PROLOG-Basis heranziehen lassen.

70) Später wird jede Faktenmenge FAK_r als Repräsentation des Wissens über den aktuellen Zustand "r" eines prädikatenlogischen Objektmodells erklärt. Daher dienen Integritätsbedingungen dazu, letztlich zwischen zulässigen und unzulässigen Zuständen von Objektmodellen zu differenzieren. Daher werden solche Modellzustände auch als integre bzw. nicht-integre Modellzustände bezeichnet. (Dies entspricht der weit verbreiteten, hier aber nicht verfolgten Unterscheidung zwischen konsistenten bzw. inkonsistenten Modellzuständen.)
Hinsichtlich ihrer Modellierungsfunktion stimmen Integritätsbedingungen exakt mit den Restriktionen überein, die im Rahmen quantitativ ausgerichteter Modellierungen des Operations Research bevorzugt werden. Im Rahmen der Prädikatenlogik lassen sich aber auch quantitative Prädikate ohne Schwierigkeiten formulieren. Darauf wird an anderer Stelle anhand von arithmetischen Standardprädikaten und zwei beispielhaft angeführten Minimierungs- und Maximierungsprädikaten näher eingegangen. Daher besteht zwischen Integritätsbedingungen und Restriktionen kein grundsätzlicher Unterschied. Folglich wird in dieser Arbeit zwischen den beiden Ausdrucksarten weder formal noch inhaltlich differenziert. Lediglich werden sie in der Weise kontextbezogen eingesetzt, daß von Integritätsbedingungen in vornehmlich logisch und informationstechnisch geprägten Argumentationszusammenhängen gesprochen wird, während der Restriktionsbegriff in quantitativ bestimmten Situationen vorgezogen wird.

71) In der Integritätsformel kann auch die anonyme Variable "_" vorkommen; vgl. OBERWEIS (1988b), S. 303. Diese besondere Variable läßt sich später in der Programmiersprache PROLOG ohne Schwierigkeiten verwenden. Es wird jedoch darauf verzichtet, die anonyme Variable in der prädikatenlogischen Semantik selbst zu berücksichtigen. Denn sie stellt innerhalb der konventionellen Prädikatenlogik einen Fremdkörper dar.

72) Es wird später ausgeführt, von der expliziten Verwendung des Wahrheitswerts "ungültig" abzusehen. Dadurch verlieren die nachfolgenden Ausführungen zu Integritätsbedingungen aber nicht ihre Bedeutung. Denn es braucht lediglich jede Formulierung, die sich auf den Wahrheitswert "ungültig" einer Formel bezieht, durch eine Formulierung ersetzt zu werden, die sich auf die Unbekanntheit der Formelgültigkeit erstreckt. Beispielsweise wird dann eine Faktenmenge genau dann als unzulässig qualifiziert, wenn die Gültigkeit der Integritätsformel für diese Faktenmenge unbekannt ist.

73) Jede Variable aus dem Argument einer Integritätsformel muß daher durch einen Quantor gebunden sein. Eine variablenfreie Grundtermformel erfordert dagegen keine Quantoren. Darüber hinaus wird unterstellt, daß in einer Integritätsformel, deren Argument mindestens eine Variable enthält, entweder alle Variablen jeweils durch einen Allquantor oder aber alle Variablen jeweils durch einen Existenzquantor gebunden sind. Falls eine derart homogene Variablenquantifizierung vorliegt, läßt sich die Integritätsformel immer durch eine faktische Transition repräsentie-

ren. Vgl. dazu die später präsentierten vier Beispiele für all- und existenzquantifizierte Formeln. Dem Verf. ist dagegen keine Netzkonstruktion bekannt, mit deren Hilfe eine Integritätsformel durch eine faktische Transition repräsentiert werden kann, für die gilt: Das Argument der Integritätsformel umfaßt mindestens zwei Variablen, von denen die erste durch einen All- und die zweite durch einen Existenzquantor gebunden ist. Daher werden hier alle Integritätsformeln mit mehreren und heterogen quantifizierten Variablen ausgeschlossen.

74) Vgl. dazu die Definition des (allgemeinen) prädikatenlogischen Klauselbegriffs. In der Voraussetzung der Klauselform liegt keine Beschränkung der Universalität von Integritätsformeln. Denn jede prädikatenlogische Formel läßt sich in ihre konjunktive Normalform transformieren. Die konjunktive Normalform einer Formel ist entweder eine einzelne Klausel oder ein Konjugat aus mehreren Klauseln. Falls eine Integritätsbedingung durch eine prädikatenlogische Formel ausgedrückt wird, deren konjunktive Normalform genau eine Klausel darstellt, so wird die Integritätsbedingung durch eben diese Klausel als Integritätsformel wiedergegeben. Andernfalls - wenn die konjunktive Normalform der Integritätsbedingung ein Konjugat aus mehreren Klauseln ist - wird die Integritätsbedingung durch ein entsprechendes Konjugat aus ebenso vielen Integritätsformeln ausgedrückt. Eine Einschränkung liegt allerdings in der Voraussetzung, daß die Integritätsformel keine freien Variablen besitzen darf. Diese Anforderung ist aber erforderlich, damit für die Integritätsformel bezüglich jeder Faktenmenge festgestellt werden kann, ob die Integritätsformel entweder gültig oder aber ungültig ist.

75) Wenn eine positive oder eine negative Formelkomponente eine Variable als Term enthalten, ist er in der Integritätsformel $ib_w(te_1,...,te_{K_w})$ durch einen Quantor gebunden.

76) Derselbe Sachverhalt wird auch dadurch ausgedrückt, daß die Faktenmenge die Integritätsformel (Integritätsbedingung) erfüllt.

77) Derselbe Sachverhalt wird auch dadurch ausgedrückt, daß die Faktenmenge die Integritätsformel (Integritätsbedingung) verletzt.

78) Diese Festlegungen stellen zusammen mit den früheren Vereinbarungen für die Beziehungen zwischen Faktenmengen und Integritätsformeln sicher, daß alle integritätsrelevanten Sachverhalte mit der Hilfe von prädikatenlogischen Integritätsbedingungen ausgedrückt werden können. Denn die Gesamtheit aller integritätsrelevanten Sachverhalte läßt sich als eine Komplexformel in konjunktiver Normalform darstellen, die immer gültig sein soll. Falls ein integritätsrelevanter (Teil-)Sachverhalt zunächst durch eine prädikatenlogische Formel ausgedrückt wurde, deren Ungültigkeit gewährleistet sein sollte, so wird das Negat dieser Formel gebildet. Die negierte Formel muß dann wieder gültig sein. Daher reicht es für die Formulierung von Integritätsbedingungen grundsätzlich aus, von einer Komplexformel auszugehen, die immer gültig sein soll. Diese Komplexformel ist aufgrund ihrer konjunktiven Normalform entweder genau eine Klausel oder ein Konjugat aus mehreren Klauseln. Jede Klausel ist entweder ein Literal oder ein Adjugat aus mehreren Literalen. Jedes Literal ist entweder eine atomare Formel oder deren Negat. Die Komplexformel wird durch die Gesamtheit aller Integritätsbedingungen für ein Formelsystem repräsentiert. Jede Klausel aus der Komplexformel stellt dabei genau eine Integritätsformel dar. Dies erklärt den oben definierten Aufbau von Integritätsformeln aus atomaren Formeln, deren Negaten sowie Adjugaten. Zugleich legt die konjunktive Verknüpfung aller Klauseln die implizite Verknüpfung aller korrespondierenden Integritätsformeln aus mehreren Integritätsbedingungen als ein Konjugat fest. Dies erklärt, warum eine Faktenmenge genau dann zulässig ist, wenn alle Integritätsformeln gültig sind. Denn ein Konjugat ist nur dann gültig, wenn all seine Komponenten gültig sind.

79) Darüber hinaus ist bemerkenswert, daß auch der metasprachliche Wahrheitswert "ungültig" für Integritätsformeln auf das Faktenkonzept reduziert werden konnte, obwohl die Fakten selbst nur gültige Formelvorkommnisse darstellen. Dies bestätigt den bereits angesprochenen, später vertieften Ansatz dieser Arbeit, auf die explizite Repräsentation des Wahrheitswertes "ungültig" generell zu verzichten.

80) Vgl. MENDELSON (1964), S. 49.

81) Dabei werden alle Vorkommnisse derselben Variablen durch dieselbe Konstante ersetzt.

82) Nur für den Grenzfall, in dem ein prädikatenlogisches Formelsystem keine Satzvariablen enthält, liefert seine Interpretation bereits definite Wahrheitswerte für alle enthaltenen Formeln. Dann muß keine Variablenbelegung ergänzt werden.

83) Der Index "c" wird später im Kontext Synthetischer Netze als "Farbindex" (colour index) unterschiedlicher Schaltmodi von Transaktionen motiviert.

84) Der prädikatenlogische Formalismus zulässiger Variablensubstitutionen ist recht aufwendig; vgl. zu seiner präzisen Darstellung z.B. SMULLYAN (1968), S. 44; RICHTER,M.M. (1978), S. 60, 62f. u. 65ff.; DELAHAYE (1987), S. 71; vgl. daneben auch MENDELSON (1964), S. 50f.; STEGMÜLLER (1969), S. 35; STEGMÜLLER (1973a), S. 54; DIN 5474 (1973), S. 6; ITZINGER (1976), S. 30; POTTHOFF (1981), S. 9f.
Für das hier beabsichtigte grundsätzliche Verständnis der prädikatenlogischen Semantik reicht jedoch die informale Umschreibung aus: Eine Variablenersetzung ist für ein prädikatenlogisches Formelsystem genau dann zulässig,

wenn jedes Vorkommnis derselben Variable in allen Formeln des Systems durch dieselbe Konstante aus dem Definitionsbereich DB substituiert wird (globale Variablensubstitution). Unterschiedliche Variablen können durch verschiedene Konstanten ersetzt werden, müssen es aber nicht. Im Rahmen einer Variablenbelegung werden nur die freien Variablen auf diese Weise substituiert. Die gebundenen Variablen werden dagegen umbenannt; vgl. DELAHAYE (1987), S. 71.

85) Vgl. z.B. CORDES (1988), S. 17. Diese Bezeichnungsvariante wird im prädikatenlogischen Kontext seltener, dafür aber um so häufiger im Zusammenhang mit der objektorientierten Programmierung verwendet.

86) Vgl. dazu die nachfolgende Anmerkung, die verdeutlicht, daß durchaus auch alternative - nur lokal definierte - Variablenbindungen möglich sind.

87) Allerdings besteht hier die Besonderheit, daß die Anforderung der globalen Variablensubstitution zu einer nur noch lokalen Variablenbindung abgeschwächt wird. Nur noch diejenigen Vorkommnisse derselben Variable, die innerhalb desselben Unifizierungs- und Resolutionsprozesses gebunden werden, müssen durch dieselbe Konstante substituiert werden. Da an einem solchen Unifizierungs- und Resolutionsprozeß keineswegs alle Formeln eines Formelsystems beteiligt sein müssen, können Vorkommnisse derselben Variablen durch unterschiedliche Konstanten ersetzt werden, sofern die Variablenbindungen in unterschiedlichen Unifizierungs- und Resolutionsprozessen geschehen. Daher erfolgt die Variablensubstitution im Rahmen der logischen Programmierung auf PROLOG-Basis nur noch lokal in bezug auf den jeweils ausgeführten Unifizierungs- und Resolutionsprozeß.
Die Lokalität der Variablensubstitution ist für die spätere Implementierung Synthetischer Netze durch die Programmiersprache Turbo-PROLOG von großer Bedeutung. Denn jedes Schalten einer Transition in einem Netz wird als genau ein kombinierter Unifizierungs- und Resolutionsprozeß realisiert. Dabei werden Variablen, die in den Gewichten adjazenter Kanten und den transitionsbeschriftenden Formeln enthalten sind, durch Konstanten substituiert. Auch diese Variablenersetzung erfolgt wegen des Rückgriffs auf die PROLOG-Implementierung in lokaler Weise, und zwar hier per constructionem lokal in bezug auf den Schaltakt der betrachteten Transition. Infolge dieser lokalen, jeweils nur schaltbezogen ausgeführten Variablensubstitution können in einem Netz Vorkommnisse derselben Variablen durch unterschiedliche Konstanten ersetzt werden, sofern die Variablenbindungen in verschiedenen Schaltakten geschehen. Dies gilt unabhängig davon, ob es sich um Schaltakte derselben Transition zu verschiedenen Zeitpunkten oder um Schaltakte von unterschiedlichen Transitionen handelt.

88) Die Variablenbelegung kann daher auch als eine Variablenbindungsfunktion angesehen werden. Darauf wird an anderer Stelle ausführlich zurückgegriffen.

89) Jede Stelle repräsentiert ein Prädikatssymbol. Die Gesamtheit aller Stellen eines Netzes kann als Konjugat aller stellenspezifischen Prädikatssymbole betrachtet werden. (Dies entspricht der o.a. Auffassung, ein prädikatenlogisches Formelsystem als eine komplexe Konjugat-Formel zu betrachten.) Jede Netzmarkierung M_r entspricht einer Interpretation I_r aller stellenzugehörigen Prädikatssymbole. Daher stellt die Markierung des gesamten Netzes die Extension der stellenübergreifenden Prädikatssymbole-Konjugation dar.

90) Am Rande wird darauf hingewiesen, daß die Verknüpfung von Prädikatsextensionen, die jeweils durch Netzmarkierungen repräsentiert werden, mittels der Erreichbarkeitsrelation im Prinzip den Einstieg in einen prädikatenlogischen Kalkül höherer Stufe bedeuten. Denn die Erreichbarkeitsrelation entspricht einem Prädikat, dessen Argument sich aus interpretierten Prädikatssymbolen 1. Stufe zusammensetzt. Dieser höherstufige prädikatenlogische Ansatz wird in dieser Arbeit aber nicht weiter ausgebaut, weil er für die spätere prädikatenlogische Implementierung von Synthetischen Netzen nicht erforderlich ist. Dort läßt sich im Rahmen der Implementierungssprache PROLOG die Erreichbarkeitsrelation durch Prädikate implementieren, die keine Differenzierung unterschiedlicher prädikatenlogischer Kalkülstufen erfordern. Auf die Korrespondenz zwischen Prädikaten und Relationen wurde bereits hingewiesen.
Bemerkenswert ist hier, daß sich das relationszugehörige Erreichbarkeitsprädikat in seinem Argument nicht direkt auf Prädikatssymbole 1. Stufe erstreckt, sondern auf deren Extensionen unter einer vorgegebenen Interpretation. Diese Interpretation ist die jeweils aktuelle Netzmarkierung, die allen Stellenprädikaten ihre momentanen Extensionen zuweist. Die interpretierenden Netzmarkierungen stellen ihrerseits Abbildungen von der Menge aller Stellen eines Netzes in die Menge aller zulässigen Extensionen der zugeordneten Stellenprädikate dar. Jede Abbildung kann als rechtseindeutiger Spezialfall von Relationen aufgefaßt werden. Daher lassen sich die Netzmarkierungen - und somit auch die Prädikatsextensionen - ihrerseits als Relationen ansehen. Diese Relationen bilden aus prädikatenlogischer Perspektive Prädikate 2. Stufe. Auf diese Prädikate erstreckt sich das Argument des Erreichbarkeitsprädikats. Daher läßt sich das Erreichbarkeitsprädikat als ein Prädikat 3. Stufe auffassen. An dieser hohen Stufenzahl zeigt sich die innere Komplexität des Petrinetz-Konzepts.
Statt dessen kann aber auch von einem prädikatenlogisch formulierten Objektmodell ausgegangen werden, das durch ein Petrinetz äquivalent repräsentiert wird. Dann stellen die objektbeschreibenden Prädikate des Objektmodells Formeln 1. Stufe dar. Ihre aktuellen Extensionen werden durch Netzmarkierungen wiedergegeben. Aus diesem Blickwinkel stellt die Erreichbarkeitsrelation zwischen Netzmarkierungen nur noch ein Prädikat 2. Stufe dar, weil es sich "nur" noch auf Extensionen von Prädikaten 1. Stufe erstreckt. Diese Auffassung der Erreichbarkeitsrelation ent-

spricht dem Ansatz von LORENZEN,P. (1987), S. 124ff., einen "praktischen Modaloperator" der Erreichbarkeit von Sachverhalten einzuführen. Die Korrespondenz zwischen einem Erreichbarkeitsprädikat 2. Ordnung und einer praktischen Modallogik im Sinne LORENZEN's resultiert aus der allgemeinen logischen Beziehung, daß sich Modallogiken auf Prädikatenlogiken 2. Stufe reduzieren lassen; vgl. VAN EMDE BOAS (1978), S. 10f.
Darüber hinaus zeigt sich die Möglichkeit, das Petrinetz-Konzept modallogisch zu interpretieren. Dies ist insofern interessant, als später eine modallogische Erweiterung der Schaltregel von Synthetischen Netzen vorgenommen wird, um Schalt*notwendigkeiten* auszudrücken. Hieran zeigt sich die formallogische Reichhaltigkeit des Petrinetz-Konzepts, in mehrfacher Hinsicht über seine primäre Ausrichtung auf die Prädikatenlogik 1. Stufe hinaus zu führen: Es verweist auch auf die Modallogik - oder auf äquivalente höherstufige Prädikatenlogiken. Allerdings wird dieser konzeptionell bemerkenswerte Aspekt in der vorliegenden Ausarbeitung nicht vertieft. Infolge der Basisentscheidung, eine prädikatenlogische Implementierung von Netzmodellen mit Hilfe der Programmiersprache PROLOG vorzulegen, wird die modallogische Perspektive nicht in eine entsprechend ausformulierte modallogische Netzkonzeption umgesetzt. Denn es wird sich zeigen, daß es die PROLOG-Implementierungsumgebung zuläßt, die oben angesprochenen modallogischen Komponenten - die Erreichbarkeitsrelation und Schaltnotwendigkeiten - auch im prädikatenlogischen Kontext darzustellen. Dies zeigt zugleich, daß die prozedurale Semantik der PROLOG-Kontrollstrukturen über das Leistungspotential der Prädikatenlogik 1. Stufe hinausgeht. Andernfalls wäre es nicht möglich, mit ihrer Hilfe modal- bzw. höherstufige prädikatenlogische Konstrukte zu implementieren.

91) Auf die Schwierigkeiten, im Rahmen konventioneller prädikatenlogischer Ansätze zeitbezogenes Wissen zu repräsentieren, wurde bereits hingewiesen.

92) Vgl. CHANG,C.L. (1973), S. 28 ("A predicate is a mapping that maps a list of constants to T or F."); RICHTER,M.M. (1978), S. 59f. u. 63f.; PREIß (1989), S. 16, der zwar nicht explizit von einer Wahrheitsfunktion spricht, aber deren Definition exakt benutzt.
Allgemein ordnet die Wahrheitsfunktion wf_i jedem zulässigen K_u-stelligen Argument einer beliebigen K_u-stelligen - aussagen- oder prädikatenlogischen - Grundtermformel p_j mit $K_u \in \mathcal{N}_0$ dessen Wahrheitswert zu. Bei der Prädikatenlogik werden Argumente einer K_u-stelligen Formel p_j mit $K \in \mathcal{N}_+$ auf ihre Wahrheitswerte abgebildet. Dann umfaßt die Formelextension genau alle K_u-stelligen Argumente, welche die Formel erfüllen. Im aussagenlogischen Fall wird dagegen die Aussage p_j als 0-stellige Formel aufgefaßt. Ihr 0-stelliges Argument - das leere Wort λ - wird auf den Wahrheitswert $wf_i() \in \{wahr,falsch\}$ abgebildet. Die Formelextension ist dann mit dem jeweils zugeordneten Wahrheitswert identisch. Einen formal abweichenden Ansatz vertritt REINFRANK (1985b), S. 28. Dort bilden Wahrheitsfunktionen nicht die Argumente von prädikatenlogischen Formeln, sondern diese Formeln selbst auf Wahrheitswerte ab. Trotz der abweichenden Darstellungsweise besteht jedoch kein materieller Unterschied gegenüber der hier vertretenen Formulierung von Wahrheitsfunktionen.
Die Wahrheitsfunktionen sind - trotz der Äquivokation - deutlich von den Funktionen zu unterscheiden, die oben als Komponenten des prädikatenlogischen Kalküls eingeführt wurden. Denn die letztgenannten Funktionen stellen objektsprachliche Konstrukte dar, die rein syntaktisch definiert sind. Die hier behandelten Wahrheitsfunktionen dienen dagegen als metasprachliche Konstrukte der formalen Interpretation von syntaktisch definierten Prädikatssymbolen. Dieser metasprachliche Charakter wird auch dadurch deutlich, daß von RAMSEY,F. (1965), S. 214, Wahrheitsfunktionen als Formeln aus einer Prädikatenlogik 2. Stufe behandelt werden.
Abweichender Ansicht ist dagegen STETTER (1988), S. 47 u. 51. Erstens differenziert er nicht zwischen Prädikaten und ihren Wahrheitsfunktionen. Zweitens behandelt er die Wahrheitsfunktionen ("Prädikate") lediglich als einen Spezialfall der gewöhnlichen, objektsprachlichen Funktionen. Dies entspricht der Vermengung von Prädikats- und Funktionssymbolen seitens der logischen Programmiersprache PROLOG, die bereits kritisiert wurde.

93) Falls aus dem Prädikatssymbol $Prä_u(st_1,...,st_{K_u})$, das der Prädikatsextension $EXT_{u,r}$ zugrundeliegt, mehrere unterschiedlich strukturierte uninterpretierte Formeln $prä_u(gt_1,...,gt_{K_u})$ gebildet worden sind, existiert für jede dieser Formeln eine eigenständige Wahrheitsfunktion wf_u. Die Deklarationen dieser Wahrheitsfunktionen wf_u für dasselbe Prädikatssymbol $Prä_u$ unterscheiden sich durch die Strukturen ihrer Vorbereiche. Diese Vorbereichsstrukturen müssen jeweils mit den Strukturen der zugehörigen formelspezifischen Argumente $(gt_1,...,gt_{K_u})$ übereinstimmen. Nachfolgend wird nur der einfache Fall flach strukturierter Formelargumente explizit berücksichtigt. Dies reicht hier aus, weil später im Rahmen des prädikatenlogisch bereicherten Signaturkonzepts beliebig tief strukturierte Argumentstrukturen erfaßt werden.

94) Wahrheitsfunktionen sind nur für Grundtermformeln definiert, weil nur diese variablenfreien Prädikate wahrheitsdefinit sind.

95) Daher stellen die Wahrheitsfunktionen im Rahmen der konventionellen Prädikatenlogik vollständig definierte Funktionen dar. Später werden dagegen für zeitlich variable Prädikatsextensionen auch partielle Wahrheitsfunktionen zugelassen.

96) Das Konzept solcher Wahrheitsfunktionen lag bereits dem prädikatenlogischen Beitrag zur Definition algebraischer Formeln im Rahmen von SIG-Spezifikationen zugrunde. Dort wurde die Gültigkeit einer Formel für Grundtermtupel zurückgeführt auf die Erfüllung einer Relationsvorschrift durch diese Grundtermtupel. Die Relations-

vorschrift entspricht der o.a. Deklaration der Wahrheitsfunktion; die Grundtermtupel korrespondieren mit den o.a. Prädikatsargumenten.

97) Als regulär wird hier die BOOL'sche Menge der beiden Standard-Wahrheitswerte "gültig" und "ungültig" für die konventionelle Prädikatenlogik mit zweiwertige Semantik bezeichnet. Andere BOOL'sche Mengen werden in dieser Arbeit an anderer Stelle behandelt; vgl. beispielsweise die Ausführungen zu einer dreiwertigen Semantik.

98) Auf eine unkonventionelle Erweiterung der prädikatenlogischen Wahrheitswerte durch den dritten Wahrheitswert "unbestimmt" wird später eingegangen.

99) Aus demselben Prädikatssymbol $Prä_u$ können mehrere verschiedene uninterpretierte Formeln $prä_u(gt_1,...,gt_{Ku})$ formiert sein, die ebenso viele Wahrheitsfunktionen wf_u mit entsprechend strukturierten Vorbereichen erfordern. Daher können in der Deklaration {....} der Extensionsmenge $EXT_{u,r}$ durchaus mehrere verschieden Wahrheitsfunktionen wf_u enthalten sein. Auf eine differenzierende Indexierung wird hier der Einfachheit halber verzichtet.

100) Diese Invarianz galt auch für die früher eingeführten algebraischen Formeln aus SIG-Spezifikationen . Die Extensionen dieser Formeln sind starr, weil die formeldefinierenden Relationen jeweils fest als Mengen relationserfüllender Tupel vorgegeben sind. Die algebraischen Formeln wurden daher treffend als "Gesetze" bezeichnet. Die invarianten Formelextensionen fanden insbesondere ihren Ausdruck durch die konstanten Relationsvorschriften für die impliziten Definitionen der zugehörigen Relationen.

101) Darüber hinaus ist in der Definition der Wahrheitsfunktion wf_u auch die gesamte syntaktische Information über die Argumentstruktur des Prädikatssymbols $Prä_u$ enthalten. Daher ist es - unter der o.a. Vollständigkeitsprämisse - berechtigt, ein Prädikatssymbol mit dessen Wahrheitsfunktion gleichzusetzen. Sofern von der hier vorausgesetzten Präzisierung des allgemeinen Prädikatsbegriffs durch den Begriff des Prädikatssymbols abgesehen wird, findet sich die Identifizierung von Prädikat(ssymbol)en und ihren Wahrheitsfunktionen z.B. bei RAPHAEL (1976), S. 120: "A predicate *is* a function whose value is a truth value ... depending upon which individual it is applied to." (kursive Hervorhebung durch den Verf). Hierdurch erfolgt jedoch nicht die unzulässige Vermengung von Prädikatssymbolen und Funktion(ssymbol)en, die früher kritisiert wurde. Es wurde schon darauf hingewiesen, daß es sich bei den Wahrheitsfunktionen nicht um Funktionen im sonst üblichen objektsprachlichen Sinn, sondern um metasprachliche Konstrukte handelt.
Diese funktionale Interpretation von Prädikat(ssymbol)en erfolgt manchmal auch im Kontext der logischen Programmierung. Besonders deutlich wird dies bei KINNEBROCK (1988), S. 19 (dort werden Prädikate direkt als Funktionen ausgegeben); CORDES (1988), S. 11 u. 14 (der Prädikatsname wird als Funktor bezeichnet). Dennoch handelt es sich *dort* um unzulässige Konfundierungen von Prädikaten mit Funktionen, die unterschiedlichen Sprachebenen angehören. Denn seitens der logischen Programmierung wird zwischen objekt- und metasprachlichen Funktionen nicht differenziert. Darüber hinaus wird dort der typische BOOL'sche Nachbereich der Wahrheitsfunktionen aus Prädikatsextensionen überhaupt nicht expliziert.

102) Vgl. dazu die Ausführungen zur Nonmonotonie von operations- und kognitionsbedingten Modellveränderungen.

103) Vgl. zur logischen Monotonieeigenschaft MCDERMOTT (1979), S. 565; REITER (1980), S. 85; MCDERMOTT (1980), S. 44; MCCARTHY,J. (1980), S. 28; HABEL (1983), S. 132; BIBEL (1984), S. 155; REINFRANK (1985a), S. 92; REINFRANK (1985b), S. 3, 18f., 26 u. 28; ZELEWSKI (1986a), S. 194 u. 358f.; BEETZ (1986), S. 41; SHOHAM (1988c), S. 283; BECKSTEIN (1988a), S. 169; BREWKA (1989), S. 88.
Formal läßt sich die Monotonieeigenschaft im Rahmen konventioneller Beweissysteme als ein (meta-)metasprachlicher Folgerungszusammenhang ausdrücken. Solche Beweissysteme und die hierin definierten Formelableitungen werden an anderer Stelle näher behandelt. Ausgangspunkt sind zwei nicht-leere, nicht-identische Mengen FM_1 und FM_2, deren Elemente jeweils beliebige Formeln darstellen. Hierfür gilt bei logischer Monotonie: Wenn eine Formel p_u aus der einen Formelmenge FM_i abgeleitet werden kann, dann kann diese Formel auch immer aus der Vereinigung dieser Formelmenge FM_1 mit der anderen Formelmenge FM_2 abgeleitet werden. Daraus folgt für jedes monotone Beweissystem mit "$\vdash$" als syntaktischem Ableitungsoperator, "»" als metasprachlicher Notation für einen Folgerungszusammenhang und "&" als Schreibweise für ein metasprachliches Konjugat:

$$((FM_1 \neq FM_2) \ \& \ (FM_1 \vdash p_u)) \ » \ ((FM_1 \cup FM_2) \vdash p_u)$$

Dies umschließt auch den Fall, daß sich die beiden Formelmengen FM_1 und FM_2 einander widersprechen. Denn aus ihrer kontradiktorischen Vereinigungsmenge kann jede Formel - somit auch die Formel p_u - abgeleitet werden. Dies wurde bereits als ex falso quodlibet-Prinzip dargelegt. Da die syntaktische Ableitungsmöglichkeit in einem Beweissystem der semantischen Gültigkeit einer Formel entspricht, gilt daher ebenso: Wenn die Formel p_u bezüglich der Formelmenge FM_i gültig ist, dann ist diese Formel auch immer in der Vereinigungsmenge $FM_1 \cup FM_2$ gültig. Dies ist die o.a. Implikation der Monotonieeigenschaft.

104) Mit dieser Zukunft sind hier zukünftige Inferenzprozesse in einem formallogischen Beweissystem gemeint.

105) Die Wahrheitserhaltung wurde bereits als Charakteristikum deduktiver Logiken herausgestellt. Deduktive Logiken lassen sich mit allen Logiken identifizieren, welche die oben beschriebene Monotonieeigenschaft erfüllen. Die "klassische" Variante deduktiver Logiken stellt die Prädikatenlogik dar, die als ihren degenerierten Grenzfall die Aussagenlogik einschließt. Daher können Aussagen- und Prädikatenlogik vereinfachend mit "der" deduktiven Logik gleichgesetzt werden.

106) Prädikatenlogisch zulässige Schlußfolgerungen werden später durch den modell- und den beweistheoretischen Ansatz konkretisiert.

107) Es können nur Objekttupel hinzukommen, die zwei Bedingungen erfüllen: Erstens durfte für sie zunächst nicht bekannt sein, ob sie von der Wahrheitsfunktion auf den Wahrheitswert "gültig" abgebildet werden. Zweitens muß für sie durch Erweiterung des Wissens über die Gestalt der Wahrheitsfunktion nachträglich bekannt geworden sein, daß sie tatsächlich auf den Wahrheitswert "gültig" abgebildet werden. Danach bleiben diese Objekttupel für immer Elemente der Prädikatsextension. Ein nachträgliches Ausscheiden eines Objekttupels aus der Prädikatsextension wird in der konventionellen Prädikatenlogik durch deren Monotonieprämisse grundsätzlich ausgeschlossen.

108) Solche operationalen Logiken werden an anderer Stelle näher charakterisiert. Die formalen Operationen dieser Logikvarianten sind in einem anderen Sinne zu verstehen, als die formalen Operationen von SIG-Algebren. Jene Operationen aus SIG-Algebren erstrecken sich nur auf die objektsprachliche Ebene formaler Objekte. Statt dessen bedeuten die Operationen einer operationalen Logik Veränderungen, die an realen oder formalen Objekten durch Operationsausführungen bewirkt werden können. Hier stellen die Extensionen EXT_j von Prädikaten $prä_u$ formale Objekte dar, an denen auf metasprachlicher Ebene Veränderungen bewirkt werden sollen. Die Veränderungen resultieren in Netzen daraus, daß Schaltoperationen von Transitionen ausgeführt werden. Hierauf wird im Kontext der Schaltregeldefinition für Synthetische Netze näher zurückgekommen.

109) Für diese dichotomen Entscheidungsergebnisse gilt $wf_u(ob_1,...,ob_{K_u})$ = gültig bzw. $wf_u(ob_1,...,ob_{K_u})$ = ungültig.

110) Strenggenommen gilt die prädikatenlogische Reduzierbarkeitsbehauptung nur für rekursive Relationen. Aber es wurde bereits erläutert, daß das rekursive Definitionskonzept ohnehin die derzeit leistungsfähigste Variante der formalen Konzeptdefinitionen darstellt. Andere als rekursiv definierbare Relationen spielen weder in dieser Arbeit noch in anderen formalen Modellierungskonzepten eine Rolle.

111) Abweichender Ansicht ist BUCHER (1987). Er stützt sich dabei auf eine theoretische Unmöglichkeitsbehauptung (S. 197) und kleine Demonstrationsbeispiele (S. 199ff.). Erste wird jedoch nicht näher belegt, während letzte nicht erkennen lassen, warum sie gegen die Darstellung von Relationen durch Prädikate sprechen sollten. Auch RESCHER (1968a), S. 35, und BRINKMANN (1989), S. 11, differenzieren ohne nähere Begründung zwischen Prädikaten- und Relationenlogik.
Dagegen herrscht allgemein die - vom Verf. geteilte - Auffassung vor, innerhalb der Prädikatenlogik (1. Stufe) alle mathematischen Relationen (1. Stufe) darstellen zu können; vgl. z.B. WHITEHEAD (1925), S. 23ff., 81 u. 88, insbesondere S. 26; SCHMIDT,A. (1951), S. 187; CARNAP (1960a), S. 5f.; LUXEMBURG (1973), S. 42. Vgl. zur unmittelbaren Gleichsetzung von Relationen und Prädikaten auch MENDELSON (1964), S. 134 (Fußnote); BIBEL (1982a), S. 59; HOHENSTEIN (1986), S. 195; DELAHAYE (1987), S. 67. Bei DIN 5474 (1973), S. 5f., und POTTHOFF (1981), S. 5, wird sogar der Begriff des Relationszeichens bzw. der Relationskonstanten anstelle der Bezeichnung "Prädikatssymbol", jedoch verwendungsgleich verwendet. Vgl. des weiteren zur informalen Rückführung von Relationen auf Prädikate OPP,K. (1976), S. 31ff.; STEGMÜLLER (1984b), S. 35 u. 37f.; Ein formaler Beweis für die Möglichkeit, jede rekursive Relation in einem formalen System auf prädikatenlogischer Basis auszudrücken, findet sich bei MENDELSON (1964), S. 134 i.V.m. S. 131ff.
Selbst BUCHER (1987), S. 201ff., verwendet für seine Abhandlung der angeblich eigenständigen Relationenlogik doch wieder den prädikatenlogischen Kalkül (sic!). Der Verf. hegt den Eindruck, daß die Unterscheidung zwischen Prädikaten- und Relationenlogik auf einem historischen Fehlschluß beruht. Anscheinend wird "die" Prädikatenlogik mit ihrer frühen Ausformung gleichgesetzt, die von ARISTOTELES geprägt war: Sie stellte eine reine Klassenlogik einstelliger Prädikate ohne Berücksichtigung relationaler Sachverhalte dar. Dabei wird die spätere Verallgemeinerung des prädikatenlogischen Ausdrucksvermögens auf mehrstellige Prädikate, deren Extensionen sich jeweils als Relationen auffassen lassen, vermutlich nicht mehr wahrgenommen.

112) Relationssymbole werden in üblichen mathematischen Kontexten zumeist als Relationen bezeichnet. Die prädikatenlogische Differenzierung zwischen syntaktischer Symboldefinition und semantischer Interpretation wird im allgemeinen nicht nachvollzogen. Daher wird auch in dieser Arbeit der leichteren Verständlichkeit halber auch vereinfacht von Relationen gesprochen.

113) Jede Wahrheitsfunktion wf_u verhält sich dabei isomorph zur charakteristischen Funktion der Relation REL_u. Eine charakteristische Funktion bildet die K-stelligen Konstantentupel jeweils auf den Wert "1" oder "0" ab je nachdem, ob diese Tupel zur Menge der relationserfüllenden Konstantentupel zählen bzw. nicht dazu gehören. Die Werte "gültig" und "ungültig" der Wahrheitsfunktion entsprechen den Werten "1" bzw. "0" der charakteristischen Funktion. Vgl. zum Konzept charakteristischer Funktionen von Mengen GREWENDORF (1987), S. 341.

114) Vgl. WHITEHEAD (1925), S. 26; DIN 5474 (1973), S. 5; GENRICH (1980b), S. 519.

115) Es könnte sogar noch weiter gegangen werden, die Prädikatenlogik auch auf die Einführung aller mathematischen Mengen zu erstrecken. Denn für den Grenzfall 1-stelliger "Relationen" entspricht das Prädikatssymbol Prä_u einem Mengensymbol. Die semantische Interpretation dieses Prädikatssymbols durch seine Extension liefert dann eine extensionsgleiche konventionelle Menge M_u. Jede Menge kann in dieser Hinsicht als degenerierte, 1-stellige Relation aufgefaßt werden, für die in prädikatenlogischer Darstellungsweise gilt: $M_u = \{ob: wf_u(ob) = \text{gültig}\}$ oder kurz: $M_u = \{ob: \text{prä}_u(ob)\}$. Dem entspricht das Komprehensionsschema der naiven Mengentheorie: $ob \in M_u \Leftrightarrow wf_u(ob) = \text{gültig} \Leftrightarrow \text{prä}_u(ob)$. Vgl. zu dieser Erklärung von Mengen durch ihre mengenspezifischen Prädikate WHITEHEAD (1925), S. 23 u. 78f.; WANDSCHNEIDER (1974), S. 75 u. 82; STEGMÜLLER (1984b), S. 32.
Dennoch folgt der Verf. diesem Ansatz nicht. Denn er täuscht nur vor, daß das mathematische Mengenkonzept auf prädikatenlogischer Basis originär eingeführt werden könnte. Statt dessen wurde bereits bei der Definition des prädikatenlogischen Kalküls die Ausdrucksweise und mathematische Theorie von Mengen als bekannt vorausgesetzt. So wurden höchstens abzählbar unendliche Individuenmengen benutzt, um Konstantensymbole und Variablen einzuführen. Noch deutlicher wird die Voraussetzung des Mengenkonzepts im Rahmen der prädikatenlogischen Semantik. Dort läßt sich der zentrale Begriff der Prädikatsextension ohne Rückgriff auf den Mengenbegriff überhaupt nicht ausdrücken. Daher können im prädikatenlogischen Kontext Mengen nicht originär eingeführt, sondern allenfalls als 1-stellige Relationen präzise *re*formuliert werden. Darüber hinaus führt das o.a. naive Komprehensionsschema $ob \in M_u \Leftrightarrow \text{prä}_u(ob)$ zur Antinomie aller Mengen, die sich nicht selbst als Element enthalten; vgl. WANDSCHNEIDER (1974), S. 75, 78f. u. 82; STEGMÜLLER (1984b), S. 32f. Auch aus diesem Grund lehnt der Verf. die grundsätzliche Rückführung aller Mengen auf Prädikate ab. Statt dessen wird eine originäre Definition aller Mengen durch eine der Varianten der axiomatischen Mengentheorie vorausgesetzt; vgl. zu solchen Axiomatisierungen beispielsweise CARNAP (1960a), S. 179ff.; MESCHKOWSKI (1967), S. 178ff. u. 187ff. Sobald diese Mengen aber - außerhalb der Formalisierung dieser Arbeit - einmal axiomatisch eingeführt worden sind, können sie nachträglich analog zum o.a. Komprehensionsschema der naiven Mengentheorie mit der Hilfe eines mengenspezifischen Prädikats reformuliert werden; vgl. STEGMÜLLER (1984b), S. 33.

116) Daher ist die Relation REL_u nicht eindeutig definiert, solange kein expliziter Bezug auf die jeweils vorausgesetzte Interpretation I_r erfolgt. Strenggenommen wird durch eine Prädikatsextension $EXT_{u.r}$ eine interpretationsabhängige Relation $REL_{u.r}$ definiert. Auf diese Präzisierung wurde oben verzichtet, weil sie im Rahmen der konventionellen Prädikatenlogik unüblich ist. Im Petrinetz-Konzept wird dagegen eine interpretationsabhängige Relation verwendet, sofern seine Markierungsfunktion als Relation gedeutet wird. Denn die Markierungsfunktion M_r von Netzmodellen variiert abhängig vom jeweils aktuellen, durch den Index "r" gekennzeichneten Modellzustand. Jeder solcher Modellzustand entspricht einer Interpretation I_r des zugrundeliegenden prädikatenlogischen Objektmodells. Die Deutung der Markierungsfunktion als Relation bereitet keine Schwierigkeiten, weil sich jede Funktion als rechtseindeutige Relation auffassen läßt.

117) Das 2-stellige Identitätsprädikat $id(te_1,te_2)$ mit seiner verkürzten Infixnotation $te_1 = te_2 :\Leftrightarrow id(te_1,te_2)$ drückt die Gleichheit zweier Terme te_1 und te_2 aus. Es nimmt beim Aufbau prädikatenlogischer Systeme eine herausragende Stellung ein. Denn es erlaubt, die mathematischen Gleichungskalküle aus den Bereichen der Arithmetik, Algebra und Analysis in die Prädikatenlogik zu integrieren. Vgl. zur prädikatenlogischen Verankerung der Termgleichheit WHITEHEAD (1925), S. 18f.; WANG,HA. (1960a), S. 220; LORENZEN,P. (1962), S. 40ff.; OBERSCHELP (1962), S. 298f.; MENDELSON (1964), S. 75ff.; RICHTER,M.M. (1978), S. 75ff.; DELAHAYE (1987), S. 83.
Abweichender Ansicht ist BUCHER (1987), S. 224ff., der einen grundsätzlichen Unterschied zwischen Identität und Prädikation sieht. Der Verf. vermag der dort vorgetragenen Argumentation jedoch inhaltlich nicht zu folgen.

118) Das 2-stellige Elementprädikat $\in(ob,OB)$ mit seiner verkürzten Infixnotation $ob \in OB :\Leftrightarrow \in(ob,OB)$ ist genau dann erfüllt, wenn das formale Objekt ob aus seiner ersten Argumentstelle ein Element aus der Menge OB seiner zweiten Argumentstelle ist. Mit Hilfe dieses "Basisprädikats" wird die gesamte Mengentheorie in das Ausdruckspotential der Prädikatenlogik einbezogen. Vgl. zur prädikatenlogischen Verankerung der mengentheoretischen Elementbeziehung zwischen formalen Objekten und Mengen WHITEHEAD (1925), S. 25, 58 u. 78f.; LUXEMBURG (1973), S. 41 ("basic predicate").

119) Das vorgenannte Identitätsprädikat stellt bereits eines dieser Vergleichsprädikate dar. Eine Differenzierung zwischen Gleichheit und Identität erfolgt im hier erörteren prädikatenlogischen Kontext nicht. Denn bei extensional definierten Konstrukten, die oben vorausgesetzt wurden, lassen sich Identität und Gleichheit nicht voneinander unterscheiden.

120) Vgl. zur arithmetischen Erweiterung der Prädikatenlogik z.B. GENRICH (1987a), S. 212f.; vgl. zur Verknüpfung von Prädikatenlogik und Arithmetik ebenso PAUL,W. (1978), S. 72.
Aufgrund der arithmetischen Anreicherung der hier benutzten (Variante der) Prädikatenlogik kann von der Unentscheidbarkeit der Prädikatenlogik gesprochen werden. Denn es wurde schon darauf hingewiesen, daß die prädikatenlogische Unentscheidbarkeit nur dann gilt, wenn auch arithmetische Formeln zugelassen werden.

121) Dieser Definitionsoperator wurde bereits im Kontext von SIG-Spezifikationen formal festgelegt, dort allerdings nur auf den Sonderfall algebraischer Formeln bezogen.

122) Der Definitionsoperator ":⇔" besitzt daher den Charakter einer prädikatenlogischen Äquivalenz zwischen Definiens und Definiendum. Die Notationsweise "⇔" für Äquivalenzen wird hier zu definitorischen Zwecken um den Zusatz ":" erweitert, um die äquivalenzfremde Asymmetrie zwischen bereits bekanntem Definiens und noch nicht bekanntem Definiendum zu verdeutlichen.

123) Zwei solche Substitute wurden bereits als Infixnotationen für das relationentheoretische Identitäts- und das mengentheoretische Elementprädikat angeführt.

124) RECK (1988), S. 79, schließt solche Quantifizierungen aus, indem er für algebraische Formeln fordert, alle Variablenvorkommnisse müßten (quantor)frei formuliert sein. Diese Einschränkung wird vom Verf. aufgehoben.

125) Solche Extremwertfeststellungen erlangen später bei der Gestaltung von Netzmodellen für Maschinenbelegungen bei Flexiblen Fertigungssystemen erhebliche Bedeutung für die Modellierung von Prioritätsregeln. Darüber hinaus beruhen alle konventionellen betriebswirtschaftlichen Entscheidungsmodelle mit Zielfunktionen, die im Entscheidungsoptimum minimale oder maximale Werte annehmen sollen (Extremierungsmodelle) auf solchen Extremwertfeststellungsformeln. Für die Feststellung solcher Extremwerte kommen nur Terme in Frage, die auf einer metrischen Skala definiert sind. Andernfalls wären die metrischen Anordnungsrelationen mit den Vergleichsoperatoren "≤" oder "≥" nicht definiert. Solche metrisch skalierten Terme werden hier vorausgesetzt.

126) Trotz ihrer gleichen Namen verletzen Extremierungsprädikate und -funktionen nicht das früher eingeführte Postulat eindeutiger Prädikatsnamen. Denn die Extremierungsfunktionen stellen keine Prädikate dar. Ob im Einzelfall eine Extremierungsfunktion oder ein Extremierungsprädikat vorliegt, läßt sich eindeutig an die 1- bzw. 2-Stelligkeit des Arguments erkennen. Die Funktoren min(...) und max(...) der beiden Extremierungsfunktionen können auch als Minimierungs- bzw. Maximierungsoperatoren betrachtet werden, da algebraisch definierte Operationen und prädikatenlogisch eingeführte Funktionen einander entsprechen.

127) Semantische Schlußfolgerungen und die hiermit verknüpften semantischen Formeleigenschaften lassen sich in der prädikatenlogischen Semantik präzise definieren. Dabei notiert "»" einen (meta-)metasprachlichen Folgerungszusammenhang, der keine Schlußfolgerung *in* einer prädikatenlogischen Semantik ausdrückt, sondern eine Feststellung *über* alle prädikatenlogischen Semantiken trifft. Falls der Folgerungszusammenhang bidirektional gilt, wird dies entsprechend mit "«»" dargestellt. Analog dienen "&" und "non" als Notationen für (meta-)metasprachliche Konjugate bzw. Negate. Dann gelten folgende terminologische Vereinbarungen:

- Eine geschlossene Formel p_u ist genau dann *gültig* (*wahr*) unter einer gegebenen Interpretation I_r, wenn sie durch die Extensionen der Prädikatssymbole unmittelbar als gültige atomare Formel definiert ist oder hieraus durch Konnektoren und Operatoren als gültige Formel zusammengesetzt werden kann:

  $I_r(p_u) = \text{gültig}$.

- Eine offene Formel p_u wird durch eine Variablenbelegung V_c aus der Menge ZV_r aller Variablenbelegungen, die unter einer gegebenen Interpretation I_r zulässig sind, genau dann *erfüllt*, wenn sie durch die Variablenbelegung auf eine gültige geschlossene Formel $V_c(p_u)$ abgebildet wird:

  $I_r(V_c(p_u)) = \text{gültig}$.

- Eine offene Formel p_u ist genau dann *gültig* (*wahr*) unter einer gegebenen Interpretation I_r, wenn sie durch alle zulässigen Variablenbelegungen V_c jeweils auf eine gültige geschlossene Formel $V_c(p_u)$ abgebildet wird:

  $\forall(V_c \in ZV_r)\text{: } I_r(V_c(p_u)) = \text{gültig}$.

- Eine geschlossene Formel, die unter einer Interpretation I_r gültig ist, behält ihre Gültigkeit unabhängig davon, welche Variablenbelegungen unter der Interpretation I_r angewendet werden. Denn diese Variablenbelegungen führen nur bei den offenen Formeln zu Variablenersetzungen. Daher gilt für alle geschlossenen Formeln: $V_c(p_u) = p_u$. Daraus folgt:

  $I_r(p_u) = \text{gültig} \gg (\forall(V_c \in ZV_r)\text{: } I_r(V_c(p_u)) = I_r(p_u) = \text{gültig})$.

- Eine Interpretation I_r ist ein *Modell* der geschlossenen Formel p_u genau dann, wenn die Formel p_u unter dieser Interpretation gültig ist. Das Prädikat $MOD_g(I_r,p_u)$, das diese Modelleigenschaft einer Interpretation I_r für eine geschlossene Formel p_u ausdrückt, ist definiert durch:

  $MOD_g(I_r,p_u) \;:\Leftrightarrow\; I_r(p_u) = \text{gültig}$.

- Das Paar (I_r,V_c) aus einer Interpretation I_r und einer zulässigen Variablenbelegung V_c ist ein *Modell* der offenen Formel p_u genau dann, wenn die Formel p_u unter der Interpretation I_r von der Variablenbelegung V_c erfüllt wird. Das Prädikat $MOD_o((I_r,V_c),p_u)$, das diese Modelleigenschaft einer Interpretation I_r und einer Variablenbelegung V_c für eine offene Formel p_u ausdrückt, ist definiert durch:

  $MOD_o((I_r,V_c),p_u) \; :\Leftrightarrow \; I_r(V_c(p_u))=$gültig.

- Gegeben sei eine Formelmenge FM_q, die aus H_q Formeln p_h mit $h\in\{1,...,H_q\}$ mit $H_q\in\mathcal{N}_0$ besteht, und eine Formel p_u. Alle Formeln p_h, p_u können geschlossen oder offen sein. Eine Interpretation I_r oder ein Paar (I_r,V_c) aus einer Interpretation I_r und einer zulässigen Variablenbelegung V_c ist ein *Modell* der Formelmenge FM_q genau dann, wenn sie bzw. es zugleich ein Modell aller Formeln p_h ist, die zur Formelmenge FM_q gehören.

- Eine *semantische Schlußfolgerung* von der Formelmenge FM_q auf die Formel p_u liegt genau dann vor, wenn unter jeder Interpretation I_r für alle zulässigen Variablenbelegungen V_c gilt: Wenn alle geschlossenen und offenen Formeln p_h aus der Formelmenge FM_q unter der Interpretation I_r und der Variablenbelegung V_c gültig sind bzw. erfüllt werden, dann ist ebenso die geschlossene oder offene Formel p_u gültig bzw. erfüllt. Für die semantische Schlußfolgerung gilt also:

  $\forall(r\in\{1,...,R\})\ \forall(V_c\in ZV_r)$: $((\forall(h\in\{1,...,H_q\})$: $I_r(V_c(p_h))=$gültig$)$ » $I_r(V_c(p_u))=$gültig$)$.

- Äquivalent mit dem Voranstehenden ist die modellbezogene Definition: Eine *semantische Schlußfolgerung* von der Formelmenge FM_q auf die Formel p_u liegt genau dann vor, wenn jedes Modell der Formelmenge FM_q auch ein Modell der Formel p_u ist. Dies ist die zentrale "modelltheoretische" Definition des semantischen Folgerungsbegriffs. Sie läßt sich auch reformulieren als: Eine semantische Schlußfolgerung von der Formelmenge FM_q auf die Formel p_u liegt genau dann vor, wenn die geschlossene (offene) Formel p_u für (durch) jedes Modell der Formelmenge FM_q gültig ist (erfüllt wird).

- Die Folgerungsbeziehung zwischen der Formel p_u und der Formelmenge FM_q wird als "$FM_q \models p_u$" notiert. Die Formel p_u ist eine *logische Konsequenz* der Formelmenge FM_q genau dann, wenn "$FM_q \models p_u$" gilt. Die Formelmenge FM_q heißt dann auch die Prämissenmenge der Konklusion p_u. Die semantische Schlußfolgerung wird auch als logische oder modelltheoretische Implikation angesprochen. Die letztgenannte Bezeichnungsweise erklärt sich aus dem später eingeführten semantischen Modellbegriff.

- Eine geschlossene Formel ist unter einer Interpretation I_r genau dann *ungültig*, wenn sie unter dieser Interpretation nicht gültig ist: non$(I_r(p_u)=$gültig$)$.

- Eine offene Formel ist unter einer Interpretation I_r genau dann *ungültig*, wenn sie unter dieser Interpretation durch keine zulässige Variablenbelegung V_c erfüllt wird:

  $\forall(V_c\in ZV_r)$: non$(I_r(V_c(p_u))=$gültig$)$.

- Eine offene Formel p_u ist genau dann *unerfüllbar*, wenn sie unter allen Interpretationen I_r ungültig ist:

  $\forall(r\in\{1,...,R\})\ \forall(V_c\in ZV_r)$: non$(I_r(V_c(p_u))=$gültig$)$.

- Eine geschlossene Formel p_u ist genau dann *unerfüllbar*, wenn sie unter allen Interpretationen I_r ungültig ist:

  $\forall(r\in\{1,...,R\})$: non$(I_r(p_u)=$gültig$)$.

- Eine - geschlossene oder offene - Formel p_u wird genau dann *unerfüllbar* genannt, wenn für sie überhaupt kein Modell existiert.

- Eine geschlossene Formel p_u heißt genau dann *erfüllbar*, wenn es mindestens eine Interpretation I_r gibt, unter der die Formel gültig ist:

  $\exists(r\in\{1,...,R\})$: $I_r(p_u)=$gültig.

- Eine offene Formel p_u ist genau dann *erfüllbar*, wenn es mindestens eine Interpretation I_r und mindestens eine zulässige Variablenbelegung V_c derart gibt, daß die Formel p_u durch die Variablenbelegung V_c unter der Interpretation I_r erfüllt wird:

  $\exists(r\in\{1,...,R\})\ \exists(V_c\in ZV_r)$: $I_r(V_c(p_u))=$gültig.

- Eine - geschlossene oder offene - Formel p_u ist genau dann *erfüllbar*, wenn für sie mindestens ein Modell $MOD_g(I_r,p_u)$ bzw. $MOD_o((I_r,V_c),p_u)$ existiert.

- Eine - geschlossene oder offene - Formel p_u ist genau dann *allgemeingültig*, wenn sie unter allen Interpretationen I_r gültig ist:

  $\forall(r\in\{1,...,R\})$: $I_r(p_u)=$gültig bzw.

  $\forall(r\in\{1,...,R\})\ \forall(V_c\in ZV_r)$: $I_r(V_c(p_u))=$gültig.

- Jede allgemeingültige Formel p_u ist die logische Konsequenz der leeren Formelmenge $FM_q=\emptyset$: "$\emptyset \models p_u$" oder kurz: "$\models p_u$". Für eine allgemeingültige geschlossene oder offene Formel sind alle Interpretationen I_r bzw. alle Paare (I_r,V_c) aus Interpretationen I_r und zulässigen Variablenbelegungen V_c Modelle. Eine allgemeingültige Formel ist also unter allen Umständen gültig bzw. erfüllt, die durch eine prädikatenlogische Semantik überhaupt definiert werden.

- Allgemeingültige Formeln werden mitunter auch nur als gültige (valid) Formeln bezeichnet; vgl. z.B. SMULLYAN (1968), S. 49; BOOLOS (1980), S. 104. Dieser Gültigkeitsbegriff, der sich über *alle* Formelinterpretationen erstreckt, ist aber deutlich vom oben eingeführten Begriff der Formelgültigkeit unter einer *bestimmten* Interpretation abzugrenzen. Wenn in prädikatenlogischen Kontexten von einer Formelgültigkeit geredet wird - wie z.B. bei "Gültigkeitsproblemen" der Prädikatenlogik -, dann ist im Regelfall immer die Allgemeingültigkeit der Formel gemeint.

- Unerfüllbare Formeln heißen auch logisch falsch, kontradiktorisch oder inkonsistent. Allgemeingültige Formeln werden auch als logisch wahr (valide, gültig), tautologisch oder analytisch bezeichnet. Formeln, die weder unerfüllbar noch allgemeingültig sind, heißen synthetisch (vgl. CARNAP (1968), S. 26) oder kontingent (vgl. BUCHER (1987), S. 80). Erfüllbare Formeln schließlich werden auch als konsistent angesprochen.

- Jede erfüllbare Formel besitzt mindestens ein Modell, jede unerfüllbare Formel dagegen überhaupt kein Modell.

Vgl. zu den voranstehend erläuterten "modelltheoretischen" Aspekten semantischer Schlußfolgerungen MENDELSON (1964), S. 51 u. 54; SMULLYAN (1968), S. 49f.; STEGMÜLLER (1969), S. 49ff.; STEGMÜLLER (1973a), S. 54; CHANG,C.L. (1973), S. 11; DIN 5474 (1973), S. 6; ITZINGER (1976), S. 31ff.; RICHTER,M.M. (1978), S. 13, 38f., 60 u. 64f.; BOOLOS (1980), S. 104ff.; POTTHOFF (1981), S. 9ff.; ESSLER (1982a), S. 42ff.; STEGMÜLLER (1983), S. 79ff.; REINFRANK (1985b), S. 28f.; DELAHAYE (1987), S. 76f.; PREIß (1989), S. 17.
Kennzeichen des semantischen Folgerungsbegriffs ist seine explizite Bezugnahme auf die semantische Kategorie der Formelgültigkeit. Denn er setzt stets wahrheitsdefinite Formeln voraus. Entweder wird er nur für geschlossene Formeln definiert, die keine freien Variablen besitzen und daher immer wahrheitsdefinit bezüglich einer Interpretation sind; vgl. PREIß (1989), S. 16f. Oder es werden auch offene Formeln zugelassen, für deren freie Variablen dann aber zusätzliche Variablenbelegungen vorgesehen sind; vgl. MENDELSON (1964), S. 54; DIN 5474 (1973).

Es läßt sich beweisen, daß drei andere Ausdrucksweisen genau dann zutreffen, wenn die semantische Schlußfolgerung "$FM_q \models p_u$" für geschlossene Formel(menge)n gilt:

- Die Formeln "$FM_q \wedge (\neg p_u)$" und "$p_1 \wedge \ldots \wedge p_{Hq} \wedge (\neg p_u)$" sind jeweils unerfüllbar (inkonsistent); vgl. CHANG,C.L. (1973), S. 17; ITZINGER (1976), S. 29; DELAHAYE (1987), S. 77f.; PREIß (1989), S. 17.

- Die Formeln "$FM_q \rightarrow p_u$" und "$(p_1 \wedge \ldots \wedge p_{Hq}) \rightarrow p_u$" sind jeweils allgemeingültig; vgl. MENDELSON (1964), S. 54; CHANG,C.L. (1973), S. 16; ITZINGER (1976), S. 29; RAUTENBERG (1979), S. 40; BOOLOS (1980), S. 104f.
Die Behauptung der Allgemeingültigkeit eines der beiden vorgenannten Subjugate wird auch als semantisches oder modelltheoretisches *Theorem* bezeichnet; vgl. CHANG,C.L. (1973), S. 17; ITZINGER (1976), S. 29. Diese subjunktiv formulierten und allgemeingültigen Theoreme "$\models (FM_q \rightarrow p_u)$" oder "$\models ((p_1 \wedge \ldots \wedge p_{Hq}) \rightarrow p_u)$" stellen als "Gesetze" die zentralen formallogischen Komponenten der meisten "wissenschaftlichen" Theorien dar. Die eineindeutige Zuordnung zwischen semantischer Schlußfolgerung "$FM_q \models p_u$" einerseits und den vorgenannten Theoremen andererseits wird von CHANG,C.L. (1973), S. 16, und RAUTENBERG (1979), S. 40, als (semantisches) Deduktionstheorem bezeichnet. Das Deduktionstheorem etabliert die fundamentale Bedeutung des objektsprachlichen Subjugats für die Formulierung metasprachlicher Schlußfolgerungen. Denn mit "«»" als Notation für die eineindeutige Zuordnung zwischen zwei formalsprachlichen Ausdrücken gilt für jede semantische Schlußfolgerung aufgrund des Deduktionstheorems: $(FM_q \models p_u)$ «» $(\models (FM_q \rightarrow p_u))$.

- Alle Modelle der Formelmenge FM_q sind zugleich auch Modelle der Formel p_u:
$\forall(r\in\{1,\ldots,R\})$: $((\forall(h\in\{1,\ldots,H_q\})$: $MOD_g(I_r,p_h))$ » $MOD_g(I_r,p_u))$
$\forall(r\in\{1,\ldots,R\})\ \forall(V_c\in ZV_r)$: $((\forall(h\in\{1,\ldots,H_q\})$: $MOD_o((I_r,V_c),p_h)$ » $MOD_o((I_r,V_c),p_u)$;
vgl. PREIß (1989), S. 17.

Die (logische) Implikation "$\Rightarrow$" und die (logische) Äquivalenz "$\Leftrightarrow$" zweier Formeln p_q und p_u stellen metasprachliche Kurznotationen dar. Die Implikation "$p_q \Rightarrow p_u$" ist als die Spezialisierung der semantischen Schlußfolgerung "$FM_q \models p_u$" auf eine unäre Prämissenmenge $FM_q=\{p_q\}$ in Infixnotation definiert:

$$p_q \Rightarrow p_u \quad :\Leftrightarrow \quad \{p_q\} \models p_u$$

Die Äquivalenz "$\Leftrightarrow$" ist eine Kurznotation für die bidirektionale Implikation:

$$p_q \Leftrightarrow p_u \quad :\Leftrightarrow \quad (p_q \Rightarrow p_u \;\&\; p_u \Rightarrow p_q)$$

Unter Rückgriff auf die bereits eingeführte Erklärung der Implikation gilt ebenso:

$$p_q \Leftrightarrow p_u \quad :\Leftrightarrow \quad ((\{p_q\} \models p_u) \;\&\; (\{p_u\} \models p_q))$$

Vgl. dazu DELAHAYE (1987), S. 54. Aufgrund der o.a. Zusammenhänge zwischen semantischen Schlußfolgerungen auf der metasprachlichen Ebene einerseits und objektsprachlichen Subjugaten andererseits gilt ebenso: Implikation und Äquivalenz bedeuten die Allgemeingültigkeit der sub- bzw. bijunktiven Formelverknüpfung. Dabei wird auf die eineindeutige Zuordnung zwischen semantischer Schlußfolgerung "$FM_q \models p_u$" und allgemeingültigem Subjugat "$\models (FM_q \rightarrow p_u)$" zurückgegriffen, die durch das o.a. Deduktionstheorem ausgedrückt wurde. Zusätzlich wird die Formelmenge FM_q als einelementige Menge $\{p_q\}$ angesetzt und vereinfacht nur durch ihr Element p_q notiert. Dann gilt:

$$((p_q \Rightarrow p_u :\Leftrightarrow (\{p_q\} \models p_u)) \;\&\; ((\{p_q\} \models p_u) \text{«»} (\models (\{p_q\} \rightarrow p_u))))$$
$$\text{»} \quad (p_q \Rightarrow p_u :\Leftrightarrow (\models (p_q \rightarrow p_u)))$$

sowie:

$$((p_q \Leftrightarrow p_u \;:\Leftrightarrow\; ((\{p_q\} \models p_u) \;\&\; (\{p_u\} \models p_q))) \;\ldots$$
$$\&\; ((\{p_q\} \models p_u) \text{«»} (\models (p_q \rightarrow p_u))) \;\&\; ((\{p_u\} \models p_q) \text{«»} (\models (p_u \rightarrow p_q))))$$
$$\text{»} \quad (p_q \Leftrightarrow p_u \;:\Leftrightarrow\; ((\models (p_q \rightarrow p_u)) \;\&\; (\models (p_u \rightarrow p_q))))$$
$$\text{»} \quad (p_q \Leftrightarrow p_u \;:\Leftrightarrow\; (\models (p_q \leftrightarrow p_u)))$$

Implikation und Äquivalenz stellen also metasprachliche Ausdrücke einer formalen prädikatenlogischen Semantik dar, welche die zugrundeliegenden objektsprachlichen Sub- bzw. Bijugate als Tautologien auszeichnen. Vgl. zu diesem metasprachlichen Definitionszusammenhang SMULLYAN (1968), S. 12f.; MENDELSON (1964), S. 17; DIN 5474 (1973), S. 2; STEGMÜLLER (1983), S. 80f.; DELAHAYE (1987), S. 55.

128) Vgl. REINFRANK (1985b), S. 29.

129) HORN-Klauseln sind Klauseln, die höchstens eine nicht-negierte atomare Formel als Literal enthalten. Daher existieren drei Typen von HORN-Klauseln KL:

- Regelklauseln KL_R stellen den Standardtyp dar. Sie bestehen jeweils aus genau einer nicht-negierten atomaren Formel (einem positiven Literal) p_{Q+1} und einer nicht-leeren Menge von negierten atomaren Formeln (negativen Literalen) $\neg p_q$ mit $q \in \{1,...,Q\}$ und $Q \in \mathcal{N}_+$. Alle involvierten Literale sind miteinander adjunktiv verknüpft: $KL_R \Leftrightarrow ((\neg p_1) \vee ... \vee (\neg p_Q) \vee p_{Q+1}$. Die Formeln $\neg p_q$ lassen sich aber auch als die Antezedens-Komponenten einer Subjugatformel auffassen, deren Konklusion die Formel p_{Q+1} ist. Falls diese Perspektive eingenommen wird und eine Regelklausel mehrere Antezedens-Komponenten umfaßt, dann sind diese immer konjunktiv verknüpft. Daher gilt: $KL_R \Leftrightarrow ((p_1 \wedge ... \wedge p_Q) \rightarrow p_{Q+1})$ für $Q \geq 2$ oder $KL_R \Leftrightarrow (p_1 \rightarrow p_2)$ für $Q=1$.
- Fakten sind Klauseln KL_F, die jeweils nur genau eine nicht-negierte atomare Formel, also genau ein positives Literal p_1 umfassen. Mit $Q=0$ gilt hierfür einfach: $KL_F \Leftrightarrow p_1$.
- Zielklauseln sind Klauseln KL_Z, die jeweils keine nicht-negierte atomare Formel p_{Q+1} umfassen, aber aus einer nicht-leeren Menge von negierten atomaren Formeln (negativen Literalen) $\neg p_q$ mit $q \in \{1,...,Q\}$ und $Q \in \mathcal{N}_+$ bestehen. Falls diese Formelmenge mehrere Elemente umfaßt, gelten diese wiederum als adjunktiv verknüpft. Folglich gilt: $KL_Z \Leftrightarrow ((\neg p_1) \vee ... \vee (\neg p_Q))$ für $Q \geq 2$ oder $KL_Z \Leftrightarrow (\neg p_1)$ für $Q=1$.

Regelklauseln und Fakten werden unter den Oberbegriff der definiten Klauseln subsumiert. Zielklauseln heißen auch negative Klauseln. Bei Zielklauseln ist zu beachten, daß sie oftmals nicht als Klauseln, sondern als Anfragen an ein PROLOG-Programm thematisiert werden; vgl. z.B. CLOCKSIN (1990), S. 5ff. u. 281f. In der Frageformulierung werden jedoch die atomaren Formeln p_q mit $q \in \{1,...,Q\}$ und $Q \in \mathcal{N}_+$ aus einer Zielklausel nicht in der o.a. negierten Gestalt, sondern als positive Literale verwendet. Darüber hinaus werden bei $Q \geq 2$ diese positiven Literale in einer PROLOG-Anfrage nicht adjunktiv, sondern konjunktiv verknüpft. Trotz dieser prima facie entgegengesetzten Formulierung von Zielklauseln und PROLOG-Anfragen drücken beide Ansätze denselben Sachverhalt aus. Ihre scheinbare Widersprüchlichkeit verschwindet sofort, wenn berücksichtigt wird, daß die Programmiersprache PROLOG auf dem indirekten Inferenzkonzept der Refutation beruht. Ihm zufolge wird eine PROLOG-Anfrage dadurch beantwortet, daß das *Negat* der Anfrage zu demjenigen Formelsystem hinzugefügt wird, mit dessen Hilfe die Anfrage beantwortet werden soll. Das Negat von konjunktiv verknüpften atomaren Formeln aus einer PROLOG-Anfrage stellt immer eine adjunktive Verknüpfung der gleichen, aber negierten atomaren Formeln dar. Letztes stimmt exakt mit der oben vorgelegten Definition von Zielklauseln überein. Folglich entspricht jeder PROLOG-Anfrage *nach* deren negiertem Hinzufügen zu einem Formelsystem einer Zielklausel (vice versa).

In der voranstehenden Typisierung wurde der Übersichtlichkeit halber von Variablen abgesehen, die in den Argumenten der Formeln aus HORN-Klauseln enthalten sein können. Falls solche Variablen vorkommen, so wird in einer HORN-Klausel jede Variable durch einen Allquantor gebunden. Existenzquantoren kommen dagegen nicht vor. Dies gilt auch für Zielklauseln. Werden allerdings nicht diese Zielklauseln, sondern die ihnen entsprechenden PROLOG-Anfragen betrachtet, so sind in jenen Anfragen alle Variablen ausschließlich durch Existenzquantoren gebunden. Denn eine Zielklausel kann nur dann auf Allquantifizierungen ihrer Variablen eingeschränkt werden, wenn in der zugrundeliegenden - noch nicht negierten - PROLOG-Anfrage alle Variablen durch Existenzquantoren gebunden werden.
Aufgrund der homogenen Allquantifizierung aller Variablen in einer HORN-Klausel läßt sich z.B. jede Regelklausel mit Q-stelligem Antezedens, $Q \in \mathcal{N}_+$, $p_u(X_1,...,X_{K_u})$ als atomaren Formeln und $u \in \{1,...,Q+1\}$ formal ausdrücken als:

$$\forall(X_{1.1}) \dots \forall(X_{1.K1}) \dots \forall(X_{Q.1}) \dots \forall(X_{Q.KQ})\ \forall(X_{Q+1.1}) \dots \forall(X_{Q+1.KQ+1}): \dots$$
$$(p_1(X_{1.1},...,X_{1.K1}) \wedge \dots \wedge p_Q(X_{Q.1},...,X_{Q.KQ})) \rightarrow p_{Q+1}(X_{Q+1.1},...,X_{Q+1.KQ+1})$$

Oftmals werden die Allquantoren, die alle Variablen aus allen Formelargumenten binden, nicht explizit dargestellt. Da sie für alle HORN-Klauseln stets in derselben, invarianten Weise definiert sind, werden sie als implizit vereinbart unterstellt. Hierdurch läßt sich die aufwendige, aber wenig informative Quantorenauflistung vermeiden. Die atomaren Formeln aus dem Antezedens und - gegebenenfalls - aus der Konklusion einer HORN-Klausel werden dann in ihren Argumenten aus Termen anstelle von Variablen zusammengesetzt. Dies erlaubt im Gegensatz zu den flachen Formelargumenten, die oben aus Variablen gebildet wurden, nunmehr auch tief strukturierte Formelargumente mit ineinander verschachtelten Termen und Operatoren aufzubauen. Für diese modifizierte Notation von HORN-Klauseln gilt:

$$(p_1(te_{1.1},...,te_{1.K1}) \wedge \dots \wedge p_Q(te_{Q.1},...,te_{Q.KQ})) \rightarrow p_{Q+1}(te_{Q+1.1},...,te_{Q+1.KQ+1})$$

Vgl. zu diesem speziellen Formeltyp und seiner exklusiven PROLOG-Verwendung die Quellen, die in einer früheren Anmerkung zur Programmiersprache PROLOG angeführt wurden; z.B. CLOCKSIN (1981), S. 222f. i.V.m. S. 214ff. Vgl. daneben auch RAULEFS (1982a), S. 80; LEVI,G. (1986), S. 400 u. 407; DELAHAYE (1987), S. 141, 152 u. 156; MURATA,TA. (1988b), S. 482; KLEINE BÜNING (1988a), S. 57f.; PREIß (1989), S. 15.

130) Besonders deutlich wird diese Einschränkung der prädikatenlogischen Ausdrucksmächtigkeit von BOSSI und COCCO ausgesprochen: "In fact definite Horn clauses are a rather limited notation" (BOSSI (1989), S. 96).

131) Zwar lassen sich die meisten praktisch relevanten Prädikate als endliche HORN-Klausel-Mengen darstellen. Die Definition von HORN-Klauseln, die in der voranstehenden Anmerkung erläutert wurde, schließt aber dennoch einige wesentliche Prädikatsformulierungen aus. Dazu gehören zunächst alle negierten atomaren Formeln, die *keine* Zielklauseln, sondern "negative" Fakten darstellen; vgl. BOSSI (1989), S. 96. Auf solche "negativen" Fakten wird später noch ausführlicher eingegangen. Darüber hinaus führen Prädikate zu Schwierigkeiten, wenn sie zu Klauseln äquivalent sind, die mindestens zwei nicht-negierte atomare Formeln als Literale enthalten. Dazu gehören vor allem Subjugate, deren Konklusionen adjunktive Verknüpfungen von mindestens zwei atomaren Formeln darstellen, wie z.B. die Formel: $p_1 \vee p_2 \vee (\neg p_3) \Leftrightarrow p_3 \rightarrow (p_1 \vee p_2)$. Gleiches gilt für Subjugate, deren Konklusionen negierte Konjugate darstellen. Aber auch schon die einfache Adjunktion $p_1 \vee p_2$ läßt sich nicht direkt als HORN-Klausel-Menge formulieren. Ebensowenig ist es möglich, ein Subjugat als HORN-Klausel-Menge auszudrücken, wenn sein Antezedens aus dem Negat einer atomaren Formel besteht. Vgl. dazu sowie zu weiteren, komplexeren Beispiele für die Beschränkungen der HORN-Klausel-Repräsentation APT (1982), S. 855; SHARPE (1985), S. 220; SERGOT (1986), S. 379; DELAHAYE (1987), S. 147; POWERS (1988), S. 957f.; KLEINE BÜNING (1988a), S. 58, 60 u. 62; SCHÖNFELD,W. (1988), S. 48; RICHTER,M.M. (1988), S. 19; LIFSCHITZ (1989), S. 316f. (Formel 5).
Die problematischen Prädikate können zwar in Einzelfällen durch spezielle Umformungstechniken in HORN-Klauseln transformiert werden; vgl. DELAHAYE (1987), S. 147; LEE,R. (1988a), S. 224 (mit einer indirekten Formulierungstechnik für Adjunktionen); BOSSI (1989), S. 96 (indirekte Wiedergabe von "negativen" Fakten). Dies ist aber keineswegs immer möglich. Daher gibt es Sonderfälle, in denen sich Prädikate grundsätzlich nicht als HORN-Klauseln darstellen lassen; vgl. DELAHAYE (1987), S. 147 u. 152; RICHTER,M.M. (1988), S. 19. Allenfalls ist es möglich, durch das "negation by failure"-Prinzip Prädikate mittelbar auszudrücken, die sich nicht direkt durch HORN-Klauseln repräsentieren lassen; vgl. KOWALSKI (1987a), S. 136. An diese mittelbare Ausdrucksmöglichkeit schein auch MURATA,TA. (1988b), S. 482, zu denken, wenn er - ohne näheren Beleg - behauptet: "... any problem which can be expressed in logic can be reexpressed by means of Horn clauses." Das "negation by failure"-Prinzip, das an anderer Stelle näher dargestellt wird, führt jedoch hinsichtlich der Korrektheit von Inferenzprozeduren zu erheblichen Problemen. Daher sieht der Verf. hierin keinen akzeptablen Weg, um der Sprache PROLOG die volle Ausdruckskraft der Prädikatenlogik zu erschließen.

132) Vgl. NILSSON,N. (1980a), S. 145ff.; CLOCKSIN (1981), S. 210ff.

Der tiefere Grund für diese Transformationsmöglichkeit liegt im Konzept der konjunktiven Normalform. Denn prädikatenlogische Formeln und HORN-Klausel-Mengen werden im allgemeinen nicht unmittelbar ineinander transformiert, sondern durch das Bindeglied ihrer konjunktiven Normalformen miteinander verknüpft. Hierfür gilt:

- Jede (wohlgeformte) prädikatenlogische Formel läßt sich in ihrer konjunktiven Normalform "äquivalent" notieren, wenn die Besonderheiten der Formel-Skolemisierung beachtet werden. Vgl. zur Transformierbarkeit beliebiger Formeln in deren konjunktive Normalformen CHANG,C.L. (1973), S. 14; THIELER-MEVISSEN (1975), S. 2f.; DELAHAYE (1987), S. 57; CORDES (1988), S. 32ff.; vgl. zur Formel-Skolemisierung die Erläuterungen in der nachstehenden Anmerkung.
- Nahezu jede prädikatenlogische Formel in konjunktiver Normalform kann als eine nicht-leere Menge von HORN-Klauseln dargestellt werden, die als konjunktiv verknüpft gelten. Dies wurde in der voranstehenden Anmerkung dargelegt. (Umgekehrt ist sogar möglich, ohne Einschränkungen jede HORN-Klausel-Menge als eine Formel in konjunktiver Normalform auszudrücken.)

Folglich ist es via konjunktiver Normalform möglich, nahezu jedes Prädikat in eine HORN-Klausel-Menge zu transformieren. (Die Umkehrung gilt sogar vollständig.)

133) Allerdings gilt keine Äquivalenz im strengen prädikatenlogischen Sinne zwischen prädikatenlogischen Formeln und HORN-Klausel-Mengen. Denn es kann bei der Formeltransformation zur konjunktiven Normalform erforderlich werden, Existenzquantoren zu eliminieren. Bei diesem Reduktionsschritt - der "Formel-Skolemisierung" - müssen in schwierigen Fällen Variablen durch Funktionen über diesen Variablen (Skolem-Funktionen) ersetzt werden, deren konkrete Gestalt nicht immer bekannt ist. Daher brauchen die ursprüngliche Formel und ihre existenzquantorfreie Skolem-Normalform nicht notwendig äquivalent zu sein.
Es läßt sich aber beweisen, daß eine prädikatenlogische Formel genau dann inkonsistent (unerfüllbar) ist, wenn ihre skolemisierte Normalform inkonsistent ist; vgl. CHANG,C.L. (1973), S. 48f.; QUINE (1975b), S. 152; ITZINGER (1976), S. 34ff.; LEVI,G. (1986), S. 397; WALTHER,C. (1987), S. 120f.; CORDES (1988), S. 34; ansatzweise auch NILSSON,N. (1980a), S. 149 (allerdings in abweichender Formulierung). Die Unerfüllbarkeit prädikatenlogischer Formeln bleibt also unter Skolemisierungsoperationen äquivalent konserviert. Dies reicht für die PROLOG-Implementierung der prädikatenlogischen Beweistheorie im allgemeinen aus. Denn ihr Resolutionskonzept gehört zum Typ der Refutationsbeweise, die immer dann gelingen, wenn sich die Inkonsistenz der untersuchten Formeln ableiten läßt. Darüber ist es möglich, komplementäre Formelskolemisierungen derart zu konstruieren, daß die Allgemeingültigkeit oder die Erfüllbarkeit prädikatenlogischer Formeln äquivalent erhalten werden; vgl. MENDELSON (1964), S. 88f.; QUINE (1975b), S. 153f.; STEGMÜLLER (1984b), S. 239; DELAHAYE (1987), S. 94; CLOCKSIN (1990), S. 269.
Vgl. zur Skolem-Normalform und den zuvor skizzierten Eigenschaften ihrer Normalisierungsoperationen SKOLEM (1934), S. 156ff.; MENDELSON (1964), S. 88ff.; CHANG,C.L. (1973), S. 47ff.; RICHTER,M.M. (1978), S. 166; NILSSON,N. (1980a), S. 147; CLOCKSIN (1981), S. 211f.; STEGMÜLLER (1984b), S. 239f.; LEVI,G. (1986), S. 397; DELAHAYE (1987), S. 93ff.; WALTHER,C. (1987), S. 120ff.; CORDES (1988), S. 33; BÖHRINGER (1988), S. 14; PREIß (1989), S. 15; SPECHT,D. (1989), S. 54; CLOCKSIN (1990), S. 268f.
Das Skolemisierungsproblem entfällt nur dann, wenn aussagenlogische Formeln vorliegen oder wenn prädikatenlogische Formeln betrachtet werden, die keine Existenzquantoren besitzen. In solchen Fällen besteht sogar im strengen Sinn eine Äquivalenz zwischen prädikaten- oder aussagenlogischen Formeln einerseits und ihren konjunktiven Normalformen andererseits. In dieser Arbeit werden prädikatenlogische Formeln und ihre konjunktiven Normalformen dagegen in einem schwachen Sinn als "äquivalent" angesprochen, der auf zwei impliziten Voraussetzungen beruht: Erstens werden die o.a. Skolemisierungsoperationen implizit unterstellt. Zweitens werden nicht die objektsprachlichen Formeln bzw. ihre Normalformeln betrachtet, sondern deren metasprachlichen Formeleigenschaften der Inkonsistenz, Erfüllbarkeit und Allgemeingültigkeit.
Alle vorgenannten Komplizierungen übersieht dagegen BUCHER (1987), S. 176f., wenn er die Eliminierung von Existenzquantoren als "trivial" (S. 176) bezeichnet.

134) Vgl. POWERS (1988), S. 958ff., insbesondere S. 958 u. 964.

135) Die speziellen PROLOG-Konstrukte, die in PASIPP für die Implementierung von Netz-Transitionen verwendet werden, führen dazu, daß im PROLOG-Programm selbst nur noch HORN-Klauseln vorkommen.

136) Vgl. dazu die Anmerkungen zur Modellierungslücke, die im allgemeinen zwischen Modellierungsgüte und -effizienz klafft.

4.2.2.3 Die operationale Dimension

4.2.2.3.1 Überblick

Die operationale Dimension der Prädikatenlogik ergänzt die zuvor skizzierte prädikatenlogische Semantik um eine weitere formale Semantik. Nun steht jedoch nicht mehr der Aspekt von Formelinterpretationen im Vordergrund. An ihre Stelle treten nunmehr Formeltransformationen. Diese Formeltransformationen gehören nicht dem Bereich der prädikatenlogischen Syntax an. Denn sie betreffen nicht die syntaktische Dimension der Synthese wohlgeformter Formeln. Vielmehr setzen sie solche Formeln als Objekte der Formeltransformationen voraus[1]. Daher konstituieren die formeltransformierenden Operationen eine formale Semantik sui generis[2]. Sie erklärt die "Bedeutung" eines syntaktischen Kalküls durch die Menge aller Transformationen, die für die Formeln des Kalküls zulässig sind[3].

Grundlage einer operationalen Semantik für die Prädikatenlogik ist das Konzept der Beweissysteme. Seine Grundzüge wurden bereits an früherer Stelle kurz skizziert. Nachfolgend werden nur spezielle Aspekte thematisiert, die eine Rolle für Anwendungsmöglichkeiten und -grenzen des hier entfalteten Modellierungskonzepts Synthetischer Netze auf prädikatenlogischer Basis spielen. Dabei handelt es sich um:

- die operationale Adäquanz der Prädikatenlogik,
- die Semi-Entscheidbarkeit der Prädikatenlogik sowie
- die prozedurale Semantik der Programmiersprache PROLOG und daraus folgende Defizite von PROLOG-basierten Implementierungen von prädikatenlogischen Modellierungen.

Gemeinsames Fundament dieser drei Aspekte ist der beweistheoretische Ableitungs- oder Inferenzbegriff[4]. Eine Inferenz stellt eine rein formal definierte Operation zur Ableitung einer Formel aus einer vorgegebenen Formelmenge[5] dar[6]. Die Spezifizierung der Ausführungsvoraussetzungen und -wirkungen einer solchen Ableitungsoperation heißt Inferenzregel. Ein Inferenzkonzept ist die kohärente Gesamtheit aller Inferenzregeln, die ein in sich geschlossenes Beweissystem konstituieren.

In dieser Arbeit wird nur ein - bereits mehrfach angesprochenes - Inferenzkonzept betrachtet: die Kombination aus Unifizierungs- und Resolutionskonzept[7]. Andere Inferenzkonzepte werden seitens der Prädikatenlogik zwar angeboten[8], spielen aber hier keine Rolle[9]. Diese Eingrenzung erfolgt aus zwei Gründen, die miteinander zusammenhängen. Einerseits erfüllt das kombinierte Unifizierungs- und Resolutionskonzept die Anforderungen an ein operationales Inferenzkonzept, die im folgenden präzisiert werden. Andererseits beruht auf ihm die Programmiersprache PROLOG. Sie liegt der Implementierung von Synthetischen Netzen zugrunde, die in dieser Arbeit im Rahmen des Softwarepakets PASIPP angeregt wird. Das Unifizierungs- und Resolutionskonzept läßt sich durch drei herausragende Aspekte charakterisieren:

- Mit der Hilfe von Unifizierungsoperationen wird versucht, Variablen aus den Argumenten von prädikatenlogischen Formeln so durch teilevaluierte Terme zu ersetzen, daß die Formelargumente einheitliche Gestalt annehmen. Dies wurde bereits im Zusammenhang mit den Termauswertungen des Signaturkonzepts dargelegt.
- Die Resolution prädikatenlogischer[10] Formeln kommt mit nur einer Inferenzregel aus. Diese Resolutionsregel erstreckt sich auf die Operation der Resolventenbildung. Objekt der Regelanwendung ist eine Formel in konjunktiver Normalform, die als Klauselmenge dargestellt ist. Eine Resolvente läßt sich nur dann erzeugen, wenn die Klauselmenge mindestens zwei Klauseln umfaßt. Unter dieser Voraussetzung gilt: Falls zwei Literale in zwei unterschiedlichen Klauseln der Klauselmenge so vorkommen, daß das eine Literal das Negat des

jeweils anderen Literals ist[11], dann können beide Literale aus den Klauseln der Klauselmenge ersatzlos gestrichen werden[12]. Die derart reduzierte Klauselmenge heißt die Resolvente der ursprünglich vorgegebenen Klauselmenge. Die beiden eliminierbaren Literale werden auch als komplementäre Literale bezeichnet.

- Durch die Kombination von Unifizierungs- und Resolutionsschritten lassen sich Behauptungen über die Inkonsistenz prädikatenlogischer Formelsysteme[13] entweder bestätigen oder aber widerlegen. Zu diesem Zweck wird das Formelsystem zunächst in konjunktiver Normalform notiert. Mittels Skolemisierungs-Operationen werden alle Existenzquantoren - soweit enthalten - eliminiert. Diese konjunktive und skolemisierte Normalform wird als eine endliche Klauselmenge dargestellt. Es wird die Inferenzstrategie verfolgt, auf die Klauselmenge Unifizierungs- und Resolutionsoperationen so lange anzuwenden, bis entweder die leere Klauselmenge $\{\} = \varnothing$[14] abgeleitet ist oder aber feststeht, daß sich diese leere Klauselmenge grundsätzlich nicht ableiten läßt. Dann ist die ursprüngliche Inkonsistenzbehauptung bestätigt bzw. widerlegt. Dieses Inferenzergebnis beruht auf dem zentralen Resolutionstheorem[15], das erstmals von ROBINSON bewiesen wurde: Eine endliche, zunächst nicht-leere Klauselmenge ist *genau dann* inkonsistent (unerfüllbar), wenn aus ihr mit Hilfe von Unifizierungs- und Resolutionsoperationen die leere Klauselmenge abgeleitet werden kann[16].

Das Unifizierungs- und Resolutionskonzept stellt ein beweistheoretisch fundiertes Inferenzkonzept dar. Als solches konstituiert es eine operationale Semantik der Prädikatenlogik. Diese Semantik ist von der deklarativen Semantik, die durch das früher angesprochene modelltheoretische Schlußfolgerungskonzept gebildet wird, deutlich zu unterscheiden[17]. Beide Semantiken sind zunächst vollkommen unabhängig voneinander definiert[18]:

- Die deklarative Semantik einer Formelmenge FM_q ist die Menge aller ihrer logischen Konsequenzen p_u[19]. Sie umfaßt also alle Formeln p_u, die aufgrund des modelltheoretischen Schlußfolgerungsbegriffs für die Formelmenge FM_q definiert sind. Bezugspunkt ist die allgemeine prädikatenlogische Semantik[20].

- Die operationale Semantik einer Formelmenge FM_q ist die Menge aller ihrer Ableitungen p_u[21]. Sie besteht also aus allen Formeln p_u, die aufgrund des beweistheoretischen Ableitungsbegriffs für die Formelmenge FM_q definiert sind. Hierbei wird auf die Inferenzregeln eines speziellen Beweissystems Bezug genommen, welche die jeweils zulässigen Formelableitungen festlegen[22].

Allerdings stimmen deklarative und operationale Semantik in ihrem gemeinsamen Ziel überein, den logischen Folgerungsbegriff formal präzise festzulegen. Trotz dieser Gemeinsamkeit zeichnet sich gerade die Programmiersprache PROLOG, die dieser Arbeit zugrundeliegt, durch schwerwiegende Unterschiede zwischen ihrer operationalen und ihrer deklarativen Semantik aus[23]. Hierauf wird nachfolgend ausführlicher eingegangen.

Der bereits dargelegte semantische Schlußfolgerungsbegriff leidet unter der entscheidenden Schwäche, nicht konstruktiv zu sein. Denn *allein* auf seiner Grundlage konnte bis heute kein Verfahren konstruiert werden, mit dessen Hilfe sich für jedes präsentierte Paar aus einer Formel und einer Formelmenge entscheiden ließe, ob die erste die logische Konsequenz der zweiten ist. Die Schwierigkeit des semantischen Folgerungsbegriffs liegt darin begründet, daß er keine formalen Operationen einschließt, mit deren Hilfe sich schrittweise überprüfen ließe, ob eine Formel die logische Konsequenz einer Formelmenge ist. Ohne eine solche Zerlegung in relativ einfache Teilschritte erweist sich aber das Gesamtproblem, das Vorliegen einer logischen Konsequenz zu bestätigen oder zu verwerfen, in der Prädikatenlogik als so komplex, daß es sich im allgemeinen nicht bewältigen läßt. Hinzu kommt, daß der semantische Folgerungsbegriff an den Formelgültigkeiten anknüpft. Die zugehörigen Wahrheitswerte übersteigen als metasprachliche Ausdrücke den Bereich syntaktischer Konstrukte. Bislang konnten aber effiziente Schlußfolgerungsverfahren nur auf der Grundlage solcher Operationen entwickelt werden, die sich nicht auf die Gültigkeit von Formeln, sondern nur auf die rein syntaktisch definierte äußere Formelgestalt beziehen. Es besteht auch auf längere Sicht wenig Hoffnung, leistungsfähige Schlußfolgerungsverfahren hervorbringen zu können, die auf der Basis von Formelgültigkeiten arbeiten und damit direkt auf dem semantischen Schlußfolgerungsbegriff beruhen.

Das syntaktische Inferenzkonzept der Prädikatenlogik vermeidet dagegen die Bezugnahme auf metasprachliche Wahrheitswerte von Formeln. Es arbeitet ausschließlich mit den rein syntaktisch definierten Formelgestalten[24)]. Darüber hinaus beruht es auf kompakten Inferenzregeln. Diese Operationsvorschriften erlauben es, logische Schlußfolgerungen in eine übersichtliche Folge von Beweisschritten zu zerlegen. Daher lassen sich syntaktische Inferenzkonzepte ohne prinzipielle Schwierigkeiten in automatisch ausführbare Beweisverfahren umsetzen. Auf diese Weise ermöglichen Inferenzkonzepte die "mechanische" Realisierung von logischen Schlußfolgerungen[25)].

Das Problem syntaktischer Inferenzen besteht jedoch darin, daß infolge des fehlenden Wahrheitswertbezugs zunächst unklar bleibt, ob sie den oben eingeführten semantischen Schlußfolgerungsbegriff gleichwertig ersetzen können. Dies muß keineswegs der Fall sein[26)]. Ohne eine solche Gleichwertigkeit wäre ein syntaktisches Inferenzkonzept jedoch wenig wert. Denn nur das semantische Schlußfolgerungskonzept ist durch seine explizite Anbindung an Formelgültigkeiten unmittelbar in die prädikatenlogische Semantik und deren Wahrheitswerte eingebettet. Aus diesem Grunde wurde es an früherer Stelle als Maßstab ausgezeichnet, an dem sich jedes syntaktische Inferenzkonzept messen lassen muß. Daher wird ein logisches Folgerungskonzept genau dann als zulässig bezeichnet, wenn es bei der Untersuchung der Frage, ob eine Formel die logische Konsequenz einer Formelmenge ist, das Ergebnis einer semantischen Schlußfolgerung exakt reproduziert[27)]. Dieser Aspekt wird durch die beiden komplementären Kriterien der Vollständigkeit[28)] und Korrektheit erfaßt[29)].

Schließlich wird für jedes Inferenzkonzept noch die allgemeine Konsistenzforderung übernommen, die in dieser Arbeit für jeden kohärenten Argumentationszusammenhang vorausgesetzt wird. Sie stellt kein Spezifikum von syntaktisch definierten Inferenzkonzepten dar, sondern soll ebenso von allen semantischen Schlußfolgerungszusammenhängen erfüllt werden.

Aufgrund der voranstehenden Überlegungen werden für Inferenzkonzepte vier operationale Adäquanzpostulate aufgestellt:

- ❑ Das Inferenzkonzept muß das umfassende Problem, eine logische Folgerung auszuführen oder nachträglich zu überprüfen, in einzelne formelableitende Operationen zerlegen können. Diese Teiloperationen sollen jeweils nur auf syntaktisch definierte Formelgestalten Bezug nehmen und durch ihre Verknüpfung das untersuchte Schlußfolgerungsproblem lösen (Konstruktivität).
- ❑ Das Inferenzkonzept muß alle Schlußfolgerungen, die logisch zulässig sind, durch syntaktische Formelableitungen reproduzieren können (Vollständigkeit).
- ❑ Alle syntaktischen Formelableitungen, die durch das Inferenzkonzept ermöglicht werden, müssen logisch zulässige Schlußfolgerungen darstellen (Korrektheit). Syntaktisch erlaubte Inferenzen dürfen also niemals logisch unzulässig sein.
- ❑ Es darf nicht möglich sein, durch das Inferenzkonzept Formeln abzuleiten, die Kontradiktionen darstellen[30)] (Konsistenz)[31)].

Die Konsistenz wurde nicht in bezug auf spezielle Inferenzkonzepte, sondern für die Prädikatenlogik generell nachgewiesen. Zwar läßt sich ihre Widerspruchsfreiheit nicht mit den Mitteln der Prädikatenlogik selbst aufzeigen. Aber die prädikatenlogische Konsistenz konnte in einem umfassender definierten Konzept bewiesen werden. Sie wird daher nachfolgend nicht weiter erörtert.

Das kombinierte Unifizierungs- und Resolutionskonzept erfüllt mit seinen Operationen für die Variablenunifizierung bzw. Formelresolution das Konstruktivitätspostulat. Darüber hinaus wurde seine Vollständigkeit und Korrektheit schon vor längerer Zeit nachgewiesen[32]. Daher verfügt die Prädikatenlogik über mindestens ein operational adäquates Inferenzkonzept. Hierdurch wird die deklarative Semantik der Prädikatenlogik, die durch den modelltheoretischen Schlußfolgerungsbegriff definiert ist, durch eine operationale Semantik ergänzt. Diese operationale Semantik beruht nur noch auf dem beweistheoretischen Inferenzbegriff. Daher kommt sie mit syntaktischen definierten Formelableitungen aus[33]. Dadurch wurde es möglich, diese operationale Semantik der Prädikatenlogik auf Automatischen Informationsverarbeitungssystemen zu implementieren. Dabei wird das Unifizierungs- und Resolutionskonzept als Kontrollstruktur[34] der Programmiersprache PROLOG realisiert.

Die tatsächliche Leistungsfähigkeit, die das kombinierte Unifizierungs- und Resolutionskonzepts in seiner PROLOG-Implementierung besitzt, wird nachstehend näher untersucht. Ziel ist es, die Leistungsgrenzen auszuloten, welche das prädikatenlogische und PROLOG-basierte Fundament dieser Arbeit in seiner operationalen Dimension limitieren[35]. Um dem Vorwurf eines naiven Glaubens an die "universelle Leistungskraft" der Prädikatenlogik zu begegnen, werden diese Leistungsschranken detailliert erläutert. Dabei wird zwischen zwei Aspekten unterschieden. Erstens werden aus dem Blickwinkel von logischen Entscheidungsproblemen generelle Leistungsschranken thematisiert, die für alle prädikatenlogischen Inferenzkonzepte - und somit a fortiori auch für das Unifizierungs- und Resolutionskonzept der Programmiersprache PROLOG - gelten. Zweitens werden Einschränkungen der Inferenzfähigkeit des Unifizierungs- und Resolutionskonzepts behandelt, die aus seiner speziellen Implementierung in der Kontrollstruktur von PROLOG-Programmen resultieren. Entsprechend wird von logischen bzw. implementierungstechnischen Defiziten des Inferenzkonzepts gesprochen.

Anmerkungen zum Kapitel:

1) Dennoch werden die Inferenzregeln, welche die Gesamtheit zulässiger Formelableitungen festlegen, im allgemeinen als syntaktisch definiert bezeichnet. Diesem üblichen Sprachgebrauch hat sich der Verf. auch anläßlich der Einführung in das prädikatenlogische Fundament dieser Arbeit angeschlossen. Dabei drückt der Begriff "syntaktisch" aber nicht den Bezug auf die syntaktische Konstitution prädikatenlogischer Formeln aus. Vielmehr erstreckt er sich nur darauf, daß die Inferenzregeln vollständig formal definiert sind. Nur hinsichtlich dieser rein formalen Definitionsweise überschneiden sich Inferenzregeln mit der Formelkonstitution durch eine prädikatenlogische Syntax.

2) Diese operationale Semantik weicht von der üblichen prädikatenlogischen Semantik grundsätzlich ab. (Dies wird an anderer Stelle durch die Unterscheidung zwischen deklarativer und operationaler Semantik präzisiert.) Daher wurde die operationale Semantik hier nicht mehr zu "der" semantischen Dimension der Prädikatenlogik gerechnet, sondern als eine eigenständige logische Dimension behandelt. Dies ist aber im Sinne des nominalistischen Begriffskonzepts lediglich als eine - letztlich willkürliche - Setzung zu verstehen. Sie erfolgt mit der Absicht, den Begriff der Semantik zunächst nur in seinem üblichen prädikatenlogischen Verständnis zu verwenden. Statt dessen hätten aber ebenso semantische (i.e.S.) und operationale Dimensionen der Prädikatenlogik als Teildimensionen einer umfassender definierten semantischen Dimension (i.w.S.) festgelegt werden können.

3) Diese Kalkülbedeutung wird später durch das Paar der deklarativen und prozeduralen Semantik von Formelsystemen konkretisiert und zugleich inhaltlich differenziert.

4) Die syntaktische Inferenz bildet den Kernbegriff der Beweissysteme, die bereits kurz angesprochen wurden. Er läßt sich im prädikatenlogischen Kontext auf folgende Weise bestimmen:

- Eine Beweissystem BS = (SP,AX,IR) 1. Stufe ist ein 3-Tupel aus einer formalen Sprache SP 1. Stufe, aus einer nicht-leeren Menge AX von Axiomen und aus einer nicht-leeren Menge IR von Inferenzregeln.
- Eine Formel p_u heißt aus einer Formelmenge FM_q im Beweissystem BS genau dann *ableitbar*, falls mindestens eine endliche Sequenz von Inferenzregelanwendungen derart existiert, daß die Formel p_u durch Ausführen der Regelsequenz aus den Formeln der Formelmenge FM_q und der Axiomemenge AX erzeugt wird. Die Ableitbarkeitsbeziehung im Beweissystem BS wird durch das Symbol "$\vdash_{BS}$" notiert.
- Eine *Inferenz* der Formel p_u aus der Formelmenge FM_q liegt bezüglich des Beweissystems BS genau dann vor, wenn die Formel p_u aus der Formelmenge FM_q ableitbar ist. Dies wird mit "$FM_q \vdash_{BS} p_u$" notiert. Die abgeleitete Formel p_u heißt eine Ableitung aus der Prämissenmenge FM_q. Eine Inferenz wird auch als syntaktische, mechanische oder beweistheoretische Folgerung bezeichnet.
- Eine Formel p_u ist im Beweissystem BS genau dann *beweisbar*, wenn sie in diesem Beweissystem aus der leeren Formelmenge $FM_q = \emptyset$ und aus der Axiomemenge AX abgeleitet werden kann: "$\emptyset \vdash_{BS} p_u$" oder kurz: "$\vdash_{BS} p_u$".
- Eine beweisbare Formel heißt ein *Theorem* des Beweissystems. Theoreme eines Beweissystems entsprechen allgemeingültigen Formeln in der Weise, daß erste unabhängig von bestimmten Formeln (syntaktische Unabhängigkeit) und letzte unabhängig von bestimmten Interpretationen (semantische Unabhängigkeit) gelten. Dieser syntaktische oder beweistheoretische Theorembegriff besitzt aber nicht die Charakteristika der objektsprachlichen Subjunktion und der metasprachlichen Allgemeingültigkeit, die dem semantischen Theorembegriff zukommen.
- Ein *Beweis* ist eine endliche, aber nicht notwendig lineare Folge aus Formeln und aus Inferenzregelanwendungen, welche die Formeln miteinander verketten. Dabei ist jede Formel aus dieser Beweiskette entweder ein Axiom, oder sie läßt sich durch genau eine Inferenzregelanwendung aus der Menge derjenigen Formeln ableiten, die ihr in der Beweiskette unmittelbar voranstehen.
- Eine Formel p_u ist im Beweissystem BS genau dann *widerlegbar*, wenn ihr Negat $\neg p_u$ in diesem Beweissystem bewiesen werden kann: "$\vdash_{BS} \neg p_u$".
- Eine Formel p_u ist im Beweissystem BS genau dann *entscheidbar*, wenn sie entweder beweisbar oder aber widerlegbar ist.
- Eine Formel p_u ist im Beweissystem BS genau dann *unentscheidbar*, wenn sie nicht entschieden werden kann. Dies kann zwei Ursachen haben. Entweder ist die Formel p_u im Beweissystem BS weder beweisbar noch widerlegbar. Oder die Formel kann sowohl bewiesen als auch widerlegt werden.
- Ein Beweissystem BS heißt genau dann *deduktiv* oder *syntaktisch vollständig*, wenn in ihm jede wohlgeformte Formel p_u aus seiner Sprache SP entscheidbar ist.
- Ein Beweissystem BS heißt genau dann *syntaktisch unvollständig*, wenn seine Sprache SP mindestens eine unentscheidbare wohlgeformte Formel p_u umfaßt.

- Ein Beweissystem BS wird genau dann als *widerspruchsfrei* oder *konsistent* bezeichnet, wenn es für keine wohlgeformte Formel p_u aus seiner Sprache SP möglich ist, sowohl diese Formel p_u als auch deren Negat $\neg p_u$ abzuleiten:

  $\forall(p_u \in SP): non((\vdash_{BS} p_u) \;\&\; (\vdash_{BS} \neg p_u))$.

- Ein Beweissystem BS ist genau dann *widersprüchlich* oder *inkonsistent*, wenn seine Sprache SP mindestens eine wohlgeformte Formel p_u umfaßt, die sowohl bewiesen als auch widerlegt werden kann:

  $\forall(p_u \in SP): (\vdash_{BS} p_u) \;\&\; (\vdash_{BS} \neg p_u)$.

- Eine Formel p_u ist im Beweissystem BS genau dann *unbeweisbar*, wenn sie in diesem Beweissystem nicht bewiesen werden kann.

- Falls das Beweissystem BS mit dem Axiomensystem AX widerspruchsfrei ist, dann gilt für jede Formel p_u, die selbst ein Axiom sein kann, aber nicht braucht: Die Formel p_u ist genau dann unbeweisbar im Beweissystem BS mit dem Axiomensystem AX, wenn das modifizierte Beweissystem BS* mit dem Axiomensystem $AX^*=(AX-\{p_u\})\cup\{\neg p_u\}$ widerspruchsfrei ist.

- Ein Beweissystem BS wird genau dann als *unabhängig* bezeichnet, wenn jede Formel aus seiner Axiomemenge AX unbeweisbar ist. Für ein unabhängiges und konsistentes Beweissystem gelten also zwei Charakteristika: Erstens ist es unmöglich, eines seiner Axiome zu beweisen. Zweitens kann jedes seiner Axiome durch sein Negat ersetzt werden, ohne daß das Beweissystem inkonsistent würde.

Kennzeichen des syntaktischen Inferenzbegriffs ist, daß er auf keine semantischen Kategorien - wie etwa die Formelgültigkeit - Bezug nimmt. Dies gilt sowohl hinsichtlich formaler Semantiken mit extensional definierten semantischen Konstrukten als auch für informale Semantiken, die intensionale Bedeutungskonzepte umschließen. Die syntaktischen Inferenzregeln beziehen sich statt dessen nur auf die formalen Symbole der involvierten Formeln und deren relative Anordnung. Sie erzeugen, wenn sie auf eine Formelmenge angewendet werden, jeweils eine neue Formel.
Analog zum semantischen Deduktionstheorem existiert auch ein syntaktisches Deduktionstheorem. Es gilt allerdings nur für geschlossene Formelmengen und erfordert gewisse beweistechnische Einschränkungen; Näheres dazu bei MENDELSON (1964), S. 60f.; RICHTER,M.M. (1978), S. 41f. u. 74f.; RAUTENBERG (1979), S. 65; DELAHAYE (1987), S. 51f. Unter diesen Einschränkungen drückt das Deduktionstheorem aus, daß eine Formel p_u aus einer Formelmenge FM_q und einer Formel p_h (mit $h \neq u$ und $p_h \neq FM_q$) genau dann abgeleitet werden kann, wenn sich das Subjugat $p_h \rightarrow p_u$ aus der Formelmenge FM_q ableiten läßt:

$$(FM_q, p_h \vdash_{BS} p_u) \;«»\; (FM_q \vdash_{BS} (p_h \rightarrow p_u))$$

Auch hier offenbart sich die Bedeutung eines objektsprachlichen Subjugats als gleichwertige Notation für eine syntaktische Inferenzbeziehung. Besonders offensichtlich wird dieser Sachverhalt, wenn der Spezialfall $FM_q=\varnothing$ betrachtet wird. Dann bedeutet das syntaktische Deduktionstheorem, daß die Ableitbarkeitsbeziehung zwischen zwei geschlossenen Formeln in die Beweisbarkeit ihres Subjugats transformiert werden kann (vice versa):

$$(p_h \vdash_{BS} p_u) \;«»\; (\vdash_{BS} (p_h \rightarrow p_u))$$

Dies läßt sich von einzelnen Formeln p_q auf nicht-leere Formelmengen FM_q verallgemeinern:

$$(FM_q \vdash_{BS} p_u) \;«»\; (\vdash_{BS} (FM_q \rightarrow p_u))$$

Aufgrund der letzten Variante des syntaktischen Deduktionstheorems kann jede syntaktische Ableitbarkeit einer Formel p_u aus einer Formelmenge FM_q als Beweisbarkeit eines Subjugats aus der Ableitungsprämisse und der hieraus abgeleiteten Formel dargestellt werden. Diese Beziehung wird später bei der Erläuterung der Semi-Entscheidbarkeit genutzt.
Vgl. zu den o.a. Festlegungen für Beweissysteme und das darin eingebettete Konzept syntaktisch definierter Inferenzen KLEENE (1952), S. 124 (nur bezüglich Konsistenz); CARNAP (1960a), S. 175f.; MENDELSON (1964), S. 29ff. u. 56ff.; CARNAP (1968), S. 25ff. u. 29ff.; STEGMÜLLER (1968), S. 181ff.; SMULLYAN (1968), S. 40; STEGMÜLLER (1969), S. 48; KANITSCHEIDER (1971), S. 64; STEGMÜLLER (1973a), S. 9ff. u. 17ff.; RAPHAEL (1976), S. 113ff.; PAUL,W. (1978), S. 71 u. 76f.; RICHTER,M.M. (1978), S. 74; POTTHOFF (1981), S. 47 u. 50; STEGMÜLLER (1983), S. 86 u. 93f.; STEGMÜLLER (1984b), S. 97f.; REINFRANK (1985b), S. 18ff.; PREIß (1989), S. 18f.

5) Die Formelmenge kann auch leer sein.

6) Zumeist wird implizit angenommen, daß die abgeleitete Formel zu der vorgegebenen Formelmenge hinzugefügt wird. Dann stellt die Inferenz eine Ableitung dar, welche die vorgegebene Formelmenge in diejenige Folgemenge

überführt, die um die abgeleitete Formel erweitert ist. Diese inferenzbedingte Ableitungsbeziehung zwischen zwei Formelmengen wird vereinfacht als Formelableitung bezeichnet.

7) Vgl. zu näheren Darstellungen des Resolutionskonzepts, seiner Unifizierungskomponente sowie seinen herausragenden beweistheoretischen Merkmalen (Vollständigkeit und Korrektheit) ROBINSON,J. (1965), S. 23ff., insbesondere S. 35ff.; ERNST (1969), S. 166ff.; YATES (1970), S. 257ff. u. 263ff.; SLAGE (1972), S. 75ff., insbesondere S. 81ff.; CHANG,C.L. (1973), S. 70ff., insbesondere 80ff.; KOWALSKI (1974), S. 70ff.; HENSCHEN (1974), S. 590ff.; ROBINSON,J. (1975), S. 11ff.; MELTZER (1975), S. 23ff.; ITZINGER (1976), S. 45ff.; RAPHAEL (1976), S. 116ff. u. 123ff.; CHARNIAK (1976a), S. 6ff.; VAN EMDEN (1977), S. 266ff.; SICKEL (1977), S. 73ff.; LOVELAND (1978), S. 83ff.; RICHTER,M.M. (1978), S. 185ff.; CHANG,C.L. (1979), S. 143ff.; SHAPIRO,S.C. (1979a), S. 62ff. (mit einer algorithmischen Präzisierung); DILGER (1979), S. 268ff.; NILSSON,N. (1980a), S. 149ff. u. 161ff.; PYLYSHYN (1980), S. 451ff.; CLOCKSIN (1981), S. 218ff.; DIGRICOLI (1981), S. 539ff.; BIBEL (1982a), S. 119ff.; BIBEL (1982b), S. 103ff. u. 113ff.; APT (1982), S. 846ff.; COHEN,P. (1982), S. 86ff.; GRAHAM,N. (1983), S. 173ff.; KOWALSKI (1983a), S. 147ff.; APPELRATH (1983), S. 42ff.; BIBEL (1984), S. 150f.; STEDE (1984), S. 79ff.; BAST (1985), S. 34ff.; LAUTENBACH (1985b), S. 2ff.; LEVI,G. (1986), S. 397ff.; ZELEWSKI (1986a), S. 351ff. u. 943ff.; DELAHAYE (1987), S. 107ff.; WEIJLAND (1988), S. 710ff., insbesondere S. 716ff. (in einer modifizierten Variante); BEZEM (1988), S. 19f.; PREIß (1989), S. 19ff.; SPECHT,D. (1989), S. 53ff., sowie die zahlreichen Beiträge zum Resolutionskonzept in dem Sammelwerk SIEKMANN (1983). Vgl. des weiteren zu Resolutionskonzepten, die speziell für sortierte Prädikatenlogiken entwickelt wurden, IRANI,K. (1985), S. 1176f.; HUBER,M. (1987), S. 36ff.; WALTHER,C. (1983), S. 888ff.; WALTHER,C. (1984); WALTHER,C. (1985c), S. 7ff., insbesondere S. 32f.; WALTHER,C. (1989), S. 65ff., insbesondere S. 74ff.
Vgl. auch zur engen Verknüpfung des Resolutionskonzepts mit seinem "Schwesterkonzept" - der Unifizierung von prädikatenlogischen Formelargumenten - ROBINSON,J. (1965), S. 33; YATES (1970), S. 258; ITZINGER (1976), S. 48; CLOCKSIN (1981), S. 219; BÜRCKERT (1986), S. 186; BAST (1985), S. 34f.; LEVI,G. (1986), S. 398f.; ESTENFELD (1987), S. 67f.; SIEKMANN 1987), S. 372; DELAHAYE (1987), S. 113 u. 118; CORDES (1988), S. 37ff.; RUZICKA (1988), S. 501; PREIß (1989), S. 19.

8) Die Prädikatenlogik verfügt über ein reiches Spektrum weiterer Inferenzkonzepte. Hierzu zählen z.B.:

- die Konnektionsmethode, die maßgeblich von BIBEL ausgearbeitet wurde; vgl. DILGER (1979), S. 272ff.; BIBEL (1982b), S. 113ff., insbesondere S. 119ff. u. 133ff.; BIBEL (1984), S. 151ff.; POWERS (1988), S. 958ff.;
- die Sequenzenmethode, die auf GENTZEN (1936), S. 512ff.; GENTZEN (1938), S. 19ff., insbesondere S. 21ff., zurückgeht; vgl. auch SMULLYAN (1968), S. 109ff. u. 121ff.; STEGMÜLLER (1984b), S. 102f.;
- die Baum- oder Tableaumethode, die von SMULLYAN (1968), S. 52ff. u. 75ff. (mit Vollständigkeitsbeweis auf S. 60f.); STEGMÜLLER (1984b), S. 100ff., und SCHÖNFELD,W. (1988), S. 43ff., dargestellt wird;
- die "algebraische Vervollständigung" der logischen Programmierung, die sich auf Termersetzungssysteme (term rewriting systems) und den KNUTH/BENDIX-Vervollständigungsalgorithmus für algebraische Formelsysteme stützt; vgl. HSIANG (1983a), S. 332ff. u. 342ff.; HSIANG (1983b), S. 20ff. u. 40ff.; VARSEK (1986), S. 213; GOOS (1987a), S. 94ff.

Die Inferenzkonzepte der algebraischen Vervollständigung zeichnen sich durch zwei charakteristische Eigenschaften aus. Erstens verknüpfen sie die konventionelle Prädikatenlogik unmittelbar und explizit mit dem algebraischen Ausdrucksreichtum. Daher besitzen sie die interessante konzeptionelle Schnittstelleneigenschaft, zwischen dem algebraisch ausgerichteten Signaturkonzept einerseits und der Prädikatenlogik andererseits vermitteln zu können. Zweitens stellen Termersetzungssysteme eine besondere Ausprägung des Konzepts der Produktionsregelsystems dar, die in dieser Arbeit für die Repräsentation dynamischer Wissensbestandteile noch eine größere Rolle spielen werden. Aus beiden Gründen wäre es interessant zu untersuchen, ob es sich für die algebraisch und prädikatenlogisch fundierten Synthetischen Netze anböte, das oben zugrundegelegte kombinierte Unifizierungs- und Resolutionskonzept durch ein Inferenzkonzept mit algebraischer Vervollständigung abzulösen. Zugunsten einer solchen alternativen Inferenzbasis ließe sich z.B. anführen, daß sich das algebraisch vervollständigte Inferenzkonzept auf prädikatenlogische Formelsysteme anwenden läßt, die nicht in Klauselform vorliegen müssen; vgl. HSIANG (1983a), S. 337 u. 341. (Es wird lediglich eine Skolemisierung vorausgesetzt, so daß das Formelsystem in konjunktiver Normalform vorliegt.) Darüber hinaus weisen Anhänger der algebraischen Vervollständigung auf besondere Effizienzvorteile ihres Inferenzkonzepts gegenüber dem kombinierten Unifizierungs- und Resolutionskonzept hin; vgl. HSIANG (1983a), S. 331, 335ff. (insbesondere S. 337) u. 340.
Trotz dieser bemerkenswerten Ansätze hält der Verf. am kombinierten Unifizierungs- und Resolutionskonzept als inferentieller Basis der hier ausgearbeiteten Konzepts Synthetischer Netze fest. Dafür spricht vor allem, daß nur dieses Inferenzkonzept im Rahmen der Programmiersprache PROLOG eine ausgereifte und zugleich entwicklungsträchtige Implementierung erfahren hat. Dadurch wird das Leistungspotential der Automatischen Informationsverarbeitung erschlossen, auf das bei der Bewältigung von praktischen Inferenzaufgaben im Kontext Synthetischer Netze kaum verzichtet werden kann. Vgl. dazu insbesondere die Ausführungen zum Effizienzdefizit der Erreichbarkeitsanalyse von Synthetischen Netzen. Auch die Implementierung Synthetischer Netze mit Hilfe des Programmpakets PASIPP, das auf der Sprache PROLOG aufbaut, käme nicht mehr in Frage. Eine ähnlich leistungsfähige und benut-

zerfreundliche Softwareunterstützung, die von der algebraischen Vervollständigung ausgeht, fehlt dagegen. Schließlich stellt das Unifizierungs- und Resolutionskonzept ein bereits intensiv erforschtes Inferenzkonzept dar, während die algebraische Vervollständigung - zumindest derzeit noch - eine Außenseiterrolle spielt. Entsprechend schmal fällt die Literaturbasis aus, die sich derzeit mit dem prädikatenlogischen Inferenzvermögen der algebraischen Vervollständigung befaßt (vgl. dazu die o.a. Quellen). Deshalb sieht sich der Verf. nicht in der Lage, ihre relative Vorteilhaftigkeit gegenüber dem Unifizierungs- und Resolutionskonzept fundiert zu beurteilen.
Vgl. des weiteren zu prädikatenlogischen Inferenzkonzepten WANG,HA. (1960a), S. 223ff.; RICHTER,M.M. (1978), S. 173ff., insbesondere S. 185; STEGMÜLLER (1984b), S. 103f.

9) Daher wird das kombinierte Unifizierungs- und Resolutionskonzept fortan als "das" prädikatenlogische Inferenzkonzept behandelt.

10) Das Resolutionskonzept läßt sich ebenso - sogar in einer besonders einfachen, von Unifizierungsoperationen unabhängigen Weise - auf aussagenlogische Formeln anwenden. Hier interessiert aber nur der anspruchsvollere prädikatenlogische Fall.

11) Solange die Literale in ihren Argumenten unterschiedliche Variablen umfassen, wird versucht, sie zuvor durch Unifizierungsoperationen so aneinander anzupassen, daß sie sich wechselseitig negieren.

12) Falls sich ein Literal, das eliminiert werden soll, in einer Klausel befindet, die das Adjugat mehrerer Literale darstellt, so wird das Literal samt seines voranstehenden Adjunktors aus der Klausel gestrichen. Wenn hingegen eine Klausel aus genau einem zu eliminierenden Literal besteht, wird die gesamte Klausel aus der untersuchten Klauselmenge entfernt.

13) Es wird erstens vorausgesetzt, daß die Formelsysteme jeweils mindestens eine atomare Formel enthalten. Zweitens darf keine der atomaren Formeln die kontradiktorische Formel "$\perp$" darstellen. Dadurch wird sichergestellt, daß sich das Inferenzkonzept immer auf nicht-leere Klauselmengen bezieht (sie unten). Jedes Formelsystem kann ebenso als genau eine - beliebig komplex zusammengesetzte - Formel betrachtet werden.

14) Da jede Klauselmenge eine Formel in konjunktiver Normalform darstellt, repräsentiert auch die leere Klauselmenge $\varnothing$ eine solche Formel. Daher wird sie auch als leere Klausel bezeichnet. Darüber hinaus kann gezeigt werden, daß die leere Klauselmenge nicht irgendeine Formel darstellt, sondern jeder inkonsistenten Formel in eineindeutiger Weise entspricht. Dies folgt unmittelbar aus dem Resolutionstheorem, auf das anschließend eingegangen wird. Mit Hilfe dieses Theorems ist es zunächst möglich, jede inkonsistente Formel zunächst auf die kleinstmögliche Kontradiktion "$p_k \wedge (\neg p_k)$" zu reduzieren. Ihr entspricht die inkonsistente Klauselmenge $\{p_k, \neg p_k\}$, aus der sich die leere Klauselmenge $\{\} = \varnothing$ als Resolvente ableiten läßt. Aufgrund dieser Zusammenhänge wird die leere Klauselmenge $\{\} = \varnothing$ fortan synonym als leere Klausel $\varnothing$ oder als inkonsistente Formel $\varnothing$ angesprochen. Bei der "inkonsistenten Formel" $\varnothing$ handelt es sich allerdings nur um eine vereinfachte Diktion. Denn strenggenommen liegt keine wohldefinierte prädikatenlogische Formel, sondern nur ein mengentheoretisch definiertes Formelsurrogat vor. Dieses Surrogat vertritt die Gesamtheit aller inkonsistenten (unerfüllbaren) prädikatenlogischen Formeln, aus denen jeweils die leere Klauselmenge $\varnothing$ abgeleitet werden kann. Im Gegensatz dazu ist die kontradiktorische Formel "$\perp$" eine prädikatenlogisch wohldefinierte, atomare, nullstellige Formel.

15) Das Resolutionstheorem stellt eine Fortentwicklung des HERBRAND'schen Theorems über die Inkonsistenz (Unerfüllbarkeit) von endlichen Klauselmengen dar.
Vgl. zum Resolutionstheorem ROBINSON,J. (1965), S. 30; CHANG,C.L. (1973), S. 85f.; ROBINSON,J. (1975), S. 12; ITZINGER (1976), S. 50f. (mit einem ausführlichen Beweis); LEVI,G. (1986), S. 399; DELAHAYE (1987), S. 110ff. (mit einem ausführlichen Beweis) u. 120.
Vgl. zur Grundlegung des Resolutionstheorems durch HERBRAND's Unerfüllbarkeitstheorem ROBINSON,J. (1965), S. 26ff., insbesondere S. 27; ITZINGER (1976), S. 37ff.; DELAHAYE (1987), S. 144.
Vgl. zu einer ausführlichen Darstellung von HERBRAND's Theorem DELAHAYE (1987), S. 95ff., insbesondere S. 97f.

16) Damit konstituiert das Resolutionstheorem eine bijektive Zuordnung zwischen dem *semantischen* Inkonsistenzbegriff und dem *syntaktischen* Ableitungsbegriff für die leere Klauselmenge: Jede prädikatenlogische Formel ist genau dann unerfüllbar im semantischen Sinne, wenn aus ihr die leere Klauselmenge rein syntaktisch abgeleitet werden kann. Dabei wird auf die charakteristische Eigenschaft von Skolemisierungs-Operationen zurückgegriffen, daß eine Formel und ihre skolemisierte Normalform zwar nicht unbedingt äquivalent sein müssen, aber die Unerfüllbarkeit von Formeln invariant erhalten bleibt. Das Resolutionstheorem gewährleistet also keine Äquivalenz zwischen semantischen Schlußfolgerungen und syntaktischen Inferenzen. Aber es stellt sicher, daß semantische Inkonsistenz und syntaktische Ableitbarkeit der leeren Klauselmenge einander *eineindeutig* entsprechen.
An früherer Stelle wurde schon darauf hingewiesen, daß sich die Ergebnisse von Schlußfolgerungen invariant gegenüber der Fragestellung verhalten, ob entweder die behauptete Allgemeingültigkeit einer Formel oder aber die behauptete Inkonsistenz des Negats dieser Formel untersucht wird. Daher gilt die bijektive Zuordnung zwischen semantischer Formelinkonsistenz und syntaktischer Ableitbarkeit der leeren Klauselmenge ebenso zwischen der semantischen Allgemeingültigkeit einer Formel und der Möglichkeit, aus der Klauselmenge der negierten Formel

die leere Klauselmenge abzuleiten. Auf dieser letztgenannten eineindeutigen Entsprechung zwischen Allgemeingültigkeit und Ableitbarkeit der leeren Klauselmenge beruht die bereits konstatierte Vollständigkeit und Korrektheit des kombinierten Unifizierungs- und Resolutionskonzepts.

17) Auf das Auseinanderfallen von deklarativen Semantiken mit ihren modelltheoretischen Schlußfolgerungen einerseits und operationalen Semantiken mit ihren beweistheoretischen Inferenzen andererseits wird im übernächsten Kapitel ausführlicher eingegangen.
Allerdings fallen die deklarative und die operationale Semantik einer Formelmenge zusammen, falls der operationalen Semantik ein vollständiges und korrektes Beweissystem zugrundeliegt. Darauf wird in Kürze unter dem Aspekt der operationalen Adäquanz zurückgekommen.

18) Die Differenzierung zwischen modelltheoretischen Schlußfolgerungen und beweistheoretischen Inferenzen andererseits reflektiert die prinzipielle Dichotomie zwischen der semantischen und syntaktischen Dimension von formalsprachlichen Konzepten. Diese Dichotomie formallogischer Folgerungskonzepte wird besonders deutlich bei THIEL,C. (1980c), S. 117; SOWA (1984), S. 164; PREIß (1989), S. 16 u. 18.

19) Daher gilt für die deklarative Semantik SEM_{dek} einer Formelmenge FM_q:

$$SEM_{dek}(FM_q) = \{p_u\colon FM_q \models p_u\}$$

20) Oftmals wird die deklarative Semantik einer Formelmenge auch als Inbegriff dessen verstanden, *was* aus dieser Formelmenge gefolgert werden kann. Dabei erfolgt aber keine Festlegung, wie bei der Herleitung der Schlußfolgerungen vorzugehen ist. Vgl. z.B. ESTER (1989), S. 26.

21) Folglich gilt für die operationale Semantik SEM_{op} einer Formelmenge FM_q bezüglich eines Beweissystems BS:

$$SEM_{op}(FM_q) = \{p_u\colon FM_q \vdash_{BS} p_u\}$$

22) Aufgrund dieses Inferenzregelbezugs läßt sich die operationale Semantik einer Formelmenge als Inbegriff dessen verstehen, *wie* sich aus dieser Formelmenge durch vorgegebene Inferenzkonzepte andere Formeln ableiten lassen. Vgl. abermals z.B. ESTER (1989), S. 27.

23) Die operationale Semantik wird im Kontext der Programmiersprache PROLOG im allgemeinen als prozedurale Semantik bezeichnet. Die Begriffe operationaler und prozeduraler Semantik werden in dieser Arbeit synonym verwendet. Vgl. zur Unterscheidung zwischen operationaler (prozeduraler) und deklarativer Semantik der Programmiersprache PROLOG KOWALSKI (1979a), S. 428; CLARK,K. (1979), S. 130ff.; CLOCKSIN (1981), S. 225; RAULEFS (1982a), S. 80; RAULEFS (1982b), S. 177f.; BIBEL (1982a), S. 228ff.; SCHEFE (1982), S. 49f.; COEHO (1983), S. 39; MYLOPOULOS (1983), S. 144; KOWALSKI (1983a), S. 107; KOWALSKI (1984a), S. 1; JARKE (1984), S. 73; LEVI,G. (1986), S. 402ff. (der allerdings - abweichend von der Terminologie des Verf. - die deklarative Semantik als operationale Semantik bezeichnet); BEETZ (1986), S. 37f.; ZELEWSKI (1986a), S. 193f.; KOWALSKI (1987a), S. 129ff. u. 138; ESTENFELD (1987), S. 68; CORDES (1988), S. 45; MURATA,TA. (1988b), S. 482; ESTER (1989), S. 26ff.
Die operationale und die deklarative Semantik der Programmiersprache PROLOG sind strenggenommen nicht für diese Sprache an sich formuliert, sondern beziehen sich jeweils auf ein beliebiges Programm, das in dieser Sprache durch eine endliche Formelmenge (Klauselmenge) formuliert ist. Darüber hinaus setzt das Konzept der prozeduralen Semantik den Begriff der Kontrollstruktur voraus. Die Kontrollstruktur einer Programmiersprache bezieht sich stets auf deren Implementierung in einem Automatischen Informationsverarbeitungssystem. Sie ist der Inbegriff aller metasprachlichen Mechanismen, die seitens der Sprachimplementierung vorgehalten werden, um die objektsprachlich verfaßten Programme dieser Sprache in dem zugrundeliegenden Automatischen Informationsverarbeitungssystem tatsächlich ausführen zu können. Falls in einer Programmiersprache Entscheidungsprobleme als Programme formuliert werden, läßt sich die Kontrollstruktur dieser Programmiersprache als ein abstrakter Entscheidungsalgorithmus begreifen. Er ist für die Klasse aller Entscheidungsprobleme definiert, die sich in dieser Programmiersprache als Programme ausdrücken lassen. Die Kontrollstruktur einer prädikatenlogischen Programmiersprache legt die Gesamtheit aller Inferenzprozeduren fest, die durch Programmausführungen in dem jeweils vorausgesetzten Automatischen Informationsverarbeitungssystem prinzipiell realisiert werden können.
Die operationale Semantik der Programmiersprache PROLOG ist unter diesen Voraussetzungen die Menge aller Schlußfolgerungen, die sich aus einem PROLOG-Programm aufgrund der implementierungsabhängigen PROLOG-Kontrollstruktur durch ausführbare Inferenzprozeduren tatsächlich ziehen lassen. Dabei setzen sich die Inferenzprozeduren so aus (Teil-)Prozeduren zusammen, daß gilt: Jede konklusionsfreie Zielklausel des PROLOG-Programms wird als eine Menge von Prozeduraufrufen interpretiert. Jede definite Klausel, die ein Fakt oder eine Regelklausel darstellen kann, wird als Spezifizierung des Input/Output-Verhaltens einer auszuführenden Prozedur aufgefaßt. Die intendierte leere Klausel ist die Stopp-Operation für die Terminierung der Inferenzprozeduren. Vgl. dazu vor allem LEVI,G. (1986), S. 405; CORDES (1988), S. 45ff. Auf den Begriff der Kontrollstruktur wird an anderer Stelle zurückgekommen.

Die deklarative Semantik der Programmiersprache PROLOG umfaßt dagegen alle logischen Konsequenzen, die für die Formelmenge eines PROLOG-Programms durch den modelltheoretischen Folgerungsbegriff der Prädikatenlogik definiert sind. Diese logische Konsequenzenmenge hängt von keinen implementierungsspezifischen Besonderheiten einer PROLOG-Kontrollstruktur ab. Sie berücksichtigt daher nicht, ob ihre Elemente in einem Automatischen Informationsverarbeitungssystem durch eine Implementierung der Programmiersprache PROLOG auch als Ergebnisse von ausführbaren Inferenzprozeduren realisiert werden können.

24) Dadurch bleiben nicht nur die o.a. Formelgültigkeiten für die Ausführung prädikatenlogischer Beweise irrelevant. Auch intensionale Formelbedeutungen werden von vornherein ausgegrenzt. Dies entspricht der früher vorgetragenen Festlegung auf extensional definierte Konstrukte.

25) Vgl. REINFRANK (1985b), S. 27.

26) Es wurde bereits dargelegt, daß sich die Prädikatenlogik 2. Stufe als unvollständig erweist. Dies bedeutet, daß für sie keine syntaktischen Inferenzkonzepte angegeben werden können, die das semantische Schlußfolgerungspotential vollständig abdecken.

27) Semantische Schlußfolgerungen sind per definitionem logisch zulässig, da sie selbst zum Maßstab der logischen Zulässigkeit erhoben wurden.

28) Eine detaillierte Definition des Vollständigkeitsbegriffs erfolgt zusammen mit dem Korrektheitsbegriff in der nachfolgenden Fußnote. Der hier verwendete Vollständigkeitsbegriff ist von der syntaktischen Vollständigkeit eines Beweissystems klar zu unterscheiden. Denn hier wird die Vollständigkeit eines Beweissystems unter Einschluß der semantischen Dimension dadurch definiert, daß sich jede semantische Schlußfolgerung durch einen syntaktischen Beweis reproduzieren läßt. Daher wird auch von einem semantischen Vollständigkeitskonzept gesprochen. Die früher definierte syntaktische Vollständigkeit eines Beweissystems betrifft dagegen nur den rein syntaktischen Aspekt, jede präsentierte Formel entweder ableiten oder aber widerlegen zu können. Diese Sichtweise unterstreicht RHEINWALD (1989), S. 5, die explizit zwischen semantischer und syntaktischer (Un-)Vollständigkeit im vorgenannten Sinne differenziert.

29) Referenzpunkt für eine formal präzise Definition der (semantischen) Vollständigkeit und der Korrektheit ist ein - im einzelnen nicht näher spezifiziertes - Beweissystems BS, das sich nur auf syntaktische Inferenzen stützt. Mit Hilfe der Operatoren " $\models$" und " $\vdash_{BS}$" für die semantische Schlußfolgerung bzw. für die syntaktische Inferenz gilt dann:

- Das Beweissystem BS ist genau dann (semantisch) vollständig, wenn für jede semantische Schlußfolgerung eine syntaktische Inferenz zwischen derselben Formelmenge und Formel existiert: $(FM_q \models p_u)$ » $(FM_q \vdash_{BS} p_u)$. Daher gilt für ein vollständiges Beweissystem ebenso: Wenn eine Formel p_u aus semantischer Perspektive eine logische Konsequenz einer Formelmenge FM_q ist, dann ist diese Formel p_u auch in syntaktischer Hinsicht aus der Formelmenge FM_q ableitbar. Durch Anwenden des semantischen und syntaktischen Deduktionstheorems läßt sich ferner für jedes vollständige Beweissystem feststellen: Wenn ein Subjugat "$FM_q \rightarrow p_u$" im semantischen Sinne allgemeingültig ist, dann ist es auch im syntaktischen Verständnis beweisbar:

  $(\models (FM_q \rightarrow p_u))$ » $(\vdash_{BS} (FM_q \rightarrow p_u))$.

- Das Beweissystem BS heißt genau dann korrekt, wenn es für jede syntaktische Inferenz eine semantische Schlußfolgerung mit derselben Formelmenge und Formel gibt: $(FM_q \vdash_{BS} p_u)$ » $(FM_q \models p_u)$. Folglich trifft auf ein korrektes Beweissystem ebenso zu: Wenn eine Formel p_u in syntaktischer Hinsicht aus einer Formelmenge FM_q ableitbar ist, dann ist diese Formel p_u auch aus semantischer Perspektive eine logische Konsequenz der Formelmenge FM_q. Durch Anwenden des semantischen und syntaktischen Deduktionstheorems gilt schließlich für jedes korrekte Beweissystem: Wenn ein Subjugat "$FM_q \rightarrow p_u$" im syntaktischen Verständnis beweisbar ist, dann ist es auch im semantischen Sinne allgemeingültig:

  $(\vdash_{BS} (FM_q \rightarrow p_u))$ » $(\models (FM_q \rightarrow p_u))$.

Dabei schließt sich der Verf. den Darstellungen von RICHTER,M.M. (1978), S. 40; DELAHAYE (1987), S. 55, 57, 60ff. u. 78f.; PREIß (1989), S. 21, an. Sie bestechen durch ihre Transparenz und formale Symmetrie. Sie spiegeln genau den zentralen Aspekt wider, daß für jede formallogische Semantik mit einem vollständigen *und* korrekten Beweissystem gilt: Semantische Schlußfolgerungen und syntaktische Inferenzen stellen ergebnisgleiche - "äquivalente" - Beweiskonzepte dar, obwohl sie vollkommen unterschiedlich definiert sind.
Andere gleichwertige Definitionsansätze finden sich bei MENDELSON (1964), S. 62 u. 67 (in mißverständlicher Diktion); SMULLYAN (1968), S. 57; PAUL,W. (1978), S. 76f.; MCDERMOTT (1979), S. 566; MCDERMOTT (1980), S. 53.
Beispielsweise bedeutet die Vollständigkeit eines Beweissystems BS für den Fall $FM_q = \emptyset$ auch: Jede - im Sinne der semantischen Modelltheorie - allgemeingültige Formel ist zugleich im Beweissystem BS syntaktisch beweisbar. Analog verheißt die Korrektheit eines Beweissystems für denselben Fall: Jede syntaktisch beweisbare Formel ist zugleich eine allgemeingültige Formel im Sinne der semantischen Modelltheorie. Zusammen mit $FM_q = \emptyset$ gewährleisten daher Vollständigkeit und Korrektheit die zentrale Beziehung zwischen semantischer Modell- und syntakti-

scher Beweistheorie: Eine Formel p_u ist allgemeingültig - also ein semantisches Theorem "$\models p_u$" - *genau dann, wenn* sie beweisbar - also auch ein syntaktisches Theorem "$\vdash_{BS} p_u$" - ist; vgl. z.B. PAUL,W. (1978), S. 77f.; POTTHOFF (1981), S. 53.

Die o.a. Definitionen von DELAHAYE und PREIß fordern nicht, daß die Formelmenge FM_q als gültig vorausgesetzt sein müßte. Denn die semantische Schlußfolgerung "$FM_q \models p_u$" besagt nur: Falls die Formelmenge FM_q unter einer Interpretation I_r und bezüglich einer Variablenbelegung V_c gültig ist und wenn p_u eine logische Konsequenz der Formelmenge FM_q darstellt, dann ist auch die Formel p_u unter der Interpretation I_r bezüglich der Variablenbelegung V_c gültig. Vgl. zu dieser Auffassung auch MENDELSON (1964), S. 54; ESSLER (1982a), S. 42. Dem entspricht die syntaktische Inferenzregel des *modus ponens*, die überhaupt keine Festlegung in bezug auf Formelgültigkeiten trifft. Denn sie besagt lediglich: $FM_q, (FM_q \rightarrow p_u) \vdash_{BS} p_u$; vgl. CARNAP (1960a), S. 88; DELAHAYE (1987), S. 50 u. 72. In dem rein syntaktisch definierten Beweissystem BS werden die Wahrheitswerte der involvierten Prämissen grundsätzlich nicht berücksichtigt. Darauf weisen z.B. CARNAP (1960a), S. 85, und STEGMÜLLER (1973a), S. 7, pointiert hin. Erst wenn die semantische Schlußfolgerung "$FM_q \models p_u$" durch die *zusätzliche* Information angereichert wird, daß alle Formeln aus der Prämissenmenge FM_q unter einer Interpretation I_r bezüglich der Variablenbelegung V_c gültig sind, folgt auch, daß die Konklusion p_u für dasselbe Modell (I_r,V_c) gültig sein muß; vgl. PAUL,W. (1978), S. 77; DELAHAYE (1987), S. 55 (Proposition 5c i.V.m. 5a).

Besonders deutlich wird die Unabhängigkeit semantischer Schlußfolgerungen "$FM_q \models p_u$" und syntaktischer Inferenzen "$FM_q \vdash_{BS} p_u$" von der Gültigkeit der Formeln ihrer Prämissenmengen FM_q von BUCHER (1987), S. 17, herausgestellt: Er unterscheidet zwischen korrekten (gültigen) und wahren Schlüssen. Die vorgenannten modell- bzw. beweistheoretischen Schlüsse sind in diesem Sinne per definitionem korrekt, aber ihre Wahrheitswerte bleiben unbestimmt. Ein korrekter Schluß geht erst dann in einen wahren Schluß über, wenn auch die Formeln aus seiner Prämissenmenge FM_q als gültig bekannt sind.

Es ist allerdings oftmals üblich, Aspekte der modell- und der beweistheoretischen Implikation so miteinander zu verschränken, daß die Axiome eines Beweissystems als gültige Formeln vorausgesetzt werden; vgl. WHITEHEAD (1925), S. 90 u. 94ff.; CARNAP (1960a), S. 101. Dies liegt nahe, weil Beweise auf der Basis ungültiger Axiome wertlos sind (ex falso quodlibet). Ebenso läßt sich zugunsten gültiger Axiome wie folgt argumentieren: Die syntaktischen Theoreme p_u eines vollständigen und korrekten Beweissystems mit der Axiomemenge AX entsprechen genau den logischen Konsequenzen p_u der axiomatischen Formelmenge AX: $(AX \vdash_{BS} p_u)$ «» $(AX \models p_u)$. Aufgrund ihrer Eigenschaft, logische Konsequenzen darzustellen, sind die Theoremformeln p_u für alle Modelle (I_r,V_c) der Axiomemenge AX gültig. Folglich gilt: Wenn eine Formel p_u aus der Axiomemenge AX abgeleitet werden kann, dann ist sie für alle Interpretationen I_r und Variablenbelegungen V_c gültig, bezüglich derer auch alle Axiome aus der Menge AX gültig sind. Daher läßt sich jede Formelableitung in einem Beweissystem (syntaktischer Aspekt) implizit auf Paare (I_r,V_c) beziehen (semantischer Aspekt), die Modelle der Axiomemenge AX und der daraus abgeleiteten Formeln p_u sind. (Für alle anderen Paare (I_r,V_c), die keine Modelle der Axiomemenge darstellen, bleiben dagegen die Wahrheitswerte der Formeln p_u offen.) Unter dieser besonderen Verschränkung von syntaktischen und semantischen Folgerungsaspekten gilt: Ein Beweissystem ist genau dann vollständig (korrekt), wenn aus einer Formelmenge, deren Elemente axiomatisch als gültig vorausgesetzt werden, jede (keine) Formel durch rein syntaktisch definierte Inferenzregeln ableitet werden kann, die eine (keine) logische Konsequenz dieser axiomatischen Formelmenge im Sinne der allgemeinen prädikatenlogischen Semantik ist. Vgl. zu dieser Alternativdefinition ZELEWSKI (1986a), S. 186f.

Die Axiome eines Beweissystems als gültig anzunehmen, ist allerdings keineswegs logisch zwingend.

30) Zwei Formeln heißen kontradiktorisch, wenn jede das Negat der jeweils anderen Formel ist. Die Formelkontradiktion ist also rein syntaktisch definiert. Daher spielt für sie das semantische Schlußfolgerungskonzept keine Rolle .

31) Der Konsistenzbegriff wurde bereits oben formal präzise definiert. Ihm zufolge ist ein Beweissystem BS genau dann *konsistent*, wenn es unmöglich ist, in ihm sowohl eine Formel p_u als auch deren Negat $\neg p_u$ abzuleiten. Diese Basisanforderung an alle hier untersuchten Konzepte wurde an früherer Stelle gerechtfertigt. Mitunter wird die Konsistenzforderung auch direkt in das Korrektheitspostulat eingebettet; vgl. PAUL,W. (1978), S. 77.

32) Dies wurde bereits herausgestellt. Allerdings können Vollständigkeit und Korrektheit des Unifizierungs- und Resolutionskonzepts durch seine Implementierung seitens der Programmiersprache PROLOG verloren gehen. Vollständigkeit und Korrektheit des kombinierten Unifizierungs- und Resolutionskonzepts beruhen inhaltlich auf dem Resolutionstheorem von ROBINSON, auf das in Kürze eingegangen wird.

33) Aus Implementierungsmängeln der operationalen PROLOG-Semantik können allerdings Vollständigkeits- und Korrektheitsverletzungen resultieren.

34) Die Kontrollstrukturen der Programmiersprache PROLOG beruhen zwar immer auf dem kombinierten Unifizierungs- und Resolutionskonzept. Aber es existieren zahlreiche Varianten, die sich dadurch unterscheiden, welche speziellen Inferenzprozeduren dieses allgemeine Konzept im Einzelfall realisieren. Vgl. allgemein zu Kontrollstrukturen für die logische Programmierung z.B. KOWALSKI (1979a), S. 425 u. 428ff.; COLMERAUER (1983), S. 302ff.; LEVI,G. (1986), S. 402ff.; BEETZ (1986), S. 38f. u. 84f.; BEZEM (1988), S. 21. Diese Variationen spielen aber

in dieser Arbeit keine Rolle. Daher wird vereinfachend von "der" Kontrollstruktur der Programmiersprache PROLOG gesprochen.
Die Kontrollstruktur besteht im allgemeinen aus einem Satz starrer Regeln. Sie steuern die Reihenfolge, in welcher der abstrakte Lösungsraum eines Entscheidungsproblems durch Inferenzprozeduren nach zulässigen Lösungen durchsucht wird. Vgl. zu diesen prozeduralen Abarbeitungsregeln, die speziell für die Programmiersprache PROLOG gelten, PROLOG (o.J.), S. 54ff. (4 Basisprinzipien); CLOCKSIN (1981), S. 223f.; ROSE,T. (1986), S. 302 (dort wird auf S. 303ff. auch auf Erweiterungen der Abarbeitungsregeln durch subtileres prozedurales Metawissen eingegangen); BEETZ (1986), S. 26; TOWNSEND,C. (1987), S. 38 u. 53f.; DELAHAYE (1987), S. 156ff., insbesondere S. 157f.; FRIEDRICHS,J. (1987), S. 174ff.; CORDES (1988), S. 48.
Die Regeln bewirken in ihrer Gesamtheit eine problemunabhängig fixierte Abarbeitungssequenz vom Typ der Tiefensuche (depth first search) mit Backtracking, das in einer anderen Anmerkung näher erläutert wird. Sie konstituieren das PROLOG-spezifische Steuerungskonzept für die Ausführung von Inferenzprozeduren auf der Basis von Unifizierungs- und Resolutionskonzept. Das hierdurch repräsentierte Steuerungs- oder Kontrollwissen wird aus der Perspektive der Künstlichen Intelligenz auch als prozedurales Metawissen bezeichnet, das in der Kontrollstruktur von PROLOG-Programmen vorgehalten wird. Es überlagert das Objektwissen, das durch prädikatenlogische Formeln in PROLOG-Programmen explizit ausgedrückt wird, durch das Metawissen, mit welchen Inferenzprozeduren und in welchem Ausmaß das hierin enthaltene implizite Objektwissen erschlossen werden kann. Dadurch fundiert das Metawissen von PROLOG-Kontrollstrukturen die operationale Semantik von PROLOG-Programmen. Sie wird infolge ihrer Verwurzelung in Metawissen über zulässige Inferenzprozeduren oftmals auch als prozedurale Semantik bezeichnet. Bei BOSSI (1989), S. 96, werden die Kontrollstrukturen als "extra-logical features" gesprochen. Sie stellen eine prozedurale "Zugabe" zur deklarativen Semantik der prädikatenlogischen Formeln eines PROLOG-Programms dar.
Vgl. zur Kombination von deklarativem Objekt- und prozeduralem Metawissen durch die Programmiersprache PROLOG auch DAHL (1983), S. 106ff.; FEIGENBAUM (1983), S. 122; JARKE (1984), S. 73; WALKER,A. (1984), S. 89f.; ZELEWSKI (1986a), S. 192ff.; FRONHÖFER (1987), S. 15 (der von einer prozeduralen Kontaminierung spricht).

35) Vgl. allgemein zu Einschränkungen der Programmiersprache PROLOG gegenüber dem vollständigen und korrekten Unifizierungs- und Resolutionskonzept der Prädikatenlogik RICHTER,M.M. (1978), S. 195 (allerdings ohne direkten Bezug auf PROLOG); CLOCKSIN (1981), S. 224; APT (1982), S. 852; ELCOCK (1983a), S. 114ff., insbesondere S. 115 u. 118; WALKER,A. (1983), S. 526f.; BOBROW (1984), S. 141; KONAGAYA (1984), S. 197; LEVI,G. (1986), S. 404 u. 406f.; ZELEWSKI (1986a), S. 946f.; DELAHAYE (1987), S. 141, 152, 158 u. 162ff.; KLEINE BÜNING (1988a), S. 59f.; BEZEM (1988), S. 21f.

4.2.2.3.2 Logische Defizite

Aus rein logischer Perspektive wird die operationale Semantik der Prädikatenlogik im wesentlichen vom Terminieren ihrer Inferenzprozeduren geprägt. Diese Prozeduren dienen dazu, logische Entscheidungsprobleme zu lösen. Besondere Schwierigkeiten resultieren aus dem Umstand, daß Inferenzprozeduren und logische Entscheidungsprobleme von zwei verschiedenartigen logischen Denkansätzen ausgehen. Einerseits gehören die Inferenzprozeduren zum *beweistheoretischen* Konzept der Prädikatenlogik, das nur mit syntaktischen Ausdrucksmitteln auskommt. Logische Entscheidungsprobleme sind dagegen zunächst in semantischer Weise formuliert. Die setzen dabei den *modelltheoretischen* Ansatz der Prädikatenlogik voraus. Die Verknüpfung beider Aspekte wird im folgenden näher erläutert.

Ein logisches Entscheidungsproblem besteht in der Aufgabe zu überprüfen, ob eine Formel p_u die logische Konsequenz einer endlichen Formelmenge FM_q[1)] ist, d.h. ob "$FM_q \models p_u$" gilt[2)]. Die Formelmenge FM_q kann grundsätzlich das gesamte prädikatenlogische Ausdrucksvermögen ausschöpfen[3)]. Für die Formel p_u drohen jedoch Komplikationen, falls es sich um eine variable Formel $p_u(...,X_k,...)$ handelt, aus deren Argument mindestens eine Variable X_k mit $k \in \{1,...,K_u\}$ durch einen Allquantor gebunden ist. Diese Schwierigkeiten können immer dann auftreten, wenn die Formeln als Klauseln notiert werden, für deren Variablen generell eine implizite Allquantifizierung unterstellt wird[4)]. Diese Klauselnotation ist für das Konzept des logischen Programmierens und die daraus abgeleitete Programmiersprache PROLOG verbindlich. Sowohl das Programmierkonzept als auch die Programmiersprache stellen zentrale Komponenten des prädikatenlogischen Fundaments Synthetischer Netze dar. Daher werden die oben angedeuteten Komplikationen durch eine Restriktion für die zulässigen Formeln p_u von vornherein ausgeschlossen: Es wird vorausgesetzt, daß die Formeln p_u entweder konstante Formeln $p_u(ob_1,...,ob_{Ku})$ darstellen oder aber variable Formeln $p_u(...,X_k,...)$, deren Variablen X_k ausschließlich durch Existenzquantoren gebunden sind[5)]. Von der Erfüllung dieser Prämisse wird fortan ausgegangen[6)].

Aufgrund des semantischen Deduktionstheorems ist die Behauptung der logischen Konsequenz "$FM_q \models p_u$" äquivalent mit der Behauptung, daß das Subjugat aus der Formelmenge FM_q und der Formel p_u allgemeingültig ist, daß also "$\models (FM_q \rightarrow p_u)$" gilt[7)]. Daher läßt sich ein Entscheidungsproblem stets als die Frage nach der Allgemeingültigkeit[8)] der problemspezifischen Subjugatformel auffassen: "$\models (FM_q \rightarrow p_u)$?"[9)]. Das Entscheidungsproblem weist genau dann die *positive* Lösung "ja" auf, wenn die problemspezifische Subjugatformel tatsächlich allgemeingültig ist. Das Entscheidungsproblem besitzt dagegen die *negative* Lösung "nein" genau dann, wenn die Allgemeingültigkeit der problemspezifischen Subjugatformel zwar als Untersuchungsvorgabe behauptet wird, diese Formel in Wirklichkeit aber nicht allgemeingültig ist.

Anstatt die positive oder negative Lösung eines Entscheidungsproblems anhand der vorgenannten Kriterien direkt nachzuweisen, wird in prädikatenlogischen Kontexten zumeist indirekt argumentiert[10)]. Dabei wird zunächst das kontradiktorische Gegenteil der interessierenden Allgemeingültigkeitsbehauptung gebildet. Dann wird behauptet, daß dieses kontradiktorische Gegenteil nicht zutreffen kann. Falls es gelingt, das behauptete Nichtzutreffen des kontradiktorischen Gegenteils zu bestätigen (widerlegen), dann wird hierdurch auf indirekte Weise zugleich die ursprüngliche Allgemeingültigkeitsbehauptung bestätigt (widerlegt). Denn für jede Behauptung gilt in jeder zweiwertigen Logik[11)]: Eine Behauptung trifft genau dann zu, wenn ihr kontradiktorisches Gegenteil nicht zutrifft.

Das kontradiktorische Gegenteil der Behauptung, eine problemspezifische Subjugatformel "$FM_q \rightarrow p_u$" sei allgemeingültig, ist die Behauptung, das Negat der Formel sei erfüllbar[12)]. Das Negat der Subjugatformel "$FM_q \rightarrow p_u$" ist die Konjugatformel "$FM_q \wedge (\neg p_u)$"[13)]. Folglich handelt es sich bei dem kontradiktorischen Gegenteil der ursprünglichen Allgemeingültigkeitsbehauptung darum, die Erfüllbarkeit der Konjugatformel "$FM_q \wedge (\neg p_u)$" zu behaupten. Um die ursprüngliche Allgemeingültigkeitsbehauptung indirekt zu bestätigen oder zu widerlegen, wird daher behauptet, ihr kontradiktorisches Gegenteil treffe nicht zu, d.h. die Konjugatformel "$FM_q \wedge (\neg p_u)$" sei nicht erfüllbar. Die Unerfüllbarkeit einer Formel ist identisch mit deren Inkonsistenz. Daraus folgt schließlich: Die ursprüngliche Behauptung, die Subjugatformel "$FM_q \rightarrow p_u$" sei allgemeingültig, wird auf indirekte Weise dadurch bestätigt oder widerlegt, daß die Inkonsistenz der Konjugatformel "$FM_q \wedge (\neg p_u)$" bestätigt bzw. widerlegt wird. Also wird die zunächst interessierende Allgemeingültigkeitsbehauptung im allgemeinen auf eine Inkonsistenzbehauptung zurückgeführt.

Für die Überprüfung, ob die behauptete Allgemeingültigkeit der problemspezifischen Subjugatformel "$FM_q \rightarrow p_u$" tatsächlich zutrifft, bieten sich die prädikatenlogischen Inferenzprozeduren an[14]. Sie leisten diese Überprüfung in der voranstehend skizzierten indirekten Weise, indem sie die behauptete Inkonsistenz der zugehörigen Konjugatformel "$FM_q \wedge (\neg p_u)$" untersuchen. Zu diesem Zweck werden die zunächst semantisch formulierten Varianten des Entscheidungsproblems in syntaktischer, aber gleichwertiger Weise reformuliert. Dadurch wird von der modell- zur beweistheoretischen Betrachtungsweise übergegangen.

Ausgangspunkt des beweistheoretischen Ansatzes ist die Überprüfung, ob die problemspezifische Formel p_u aus der ebenso problemspezifischen Formelmenge FM_q mit der Hilfe von Inferenzregeln abgeleitet werden kann: "$FM_q \vdash_{BS} p_u$". Aufgrund des syntaktischen Deduktionstheorems ist dies äquivalent mit der Untersuchung, ob das Subjugat "$FM_q \rightarrow p_u$" eine beweisbare Formel darstellt[15]: "$\vdash_{BS} (FM_q \rightarrow p_u)$". Vollständigkeit und Korrektheit der prädikatenlogischen Inferenzkonzepte stellen dabei sicher, daß die syntaktisch definierte Beweisbarkeitsbehauptung bezüglich der Subjugatformel "$FM_q \rightarrow p_u$" genau dann zutrifft, wenn die semantisch definierte Allgemeingültigkeitsbehauptung für dieselbe Subjugatformel zutrifft[16]. Daher läßt sich das betrachtete Entscheidungsproblem stets als die Frage nach der Beweisbarkeit der problemspezifischen Subjugatformel auffassen: "$\vdash_{BS} (FM_q \rightarrow p_u)$?". Es besitzt eine positive (negative) Lösung genau dann, wenn die problemspezifische Subjugatformel in dem zugrundeliegenden Beweissystem tatsächlich (nicht) bewiesen werden kann[17].

Der Beweis erfolgt - wie bereits oben dargelegt wurde - indirekt. Dabei wird das Nichtzutreffen des kontradiktorischen Gegenteils der ursprünglichen Beweisbarkeitsbehauptung aufgezeigt. Das kontradiktorische Gegenteil der Beweisbarkeitsbehauptung "$\vdash_{BS} (FM_q \rightarrow p_u)$" ist die Behauptung, daß die problemspezifische Subjugatformel unbeweisbar ist. Das zugrundeliegende Beweissystem BS wird als widerspruchsfrei vorausgesetzt[18]. Unter dieser Prämisse ist die Subjugatformel "$FM_q \rightarrow p_u$" genau dann unbeweisbar, wenn sich ihr Negat - die Konjugatformel "$FM_q \wedge (\neg p_u)$" - mit dem Axiomensystem des Beweissystems widerspruchsfrei vereinbaren läßt[19]. Der indirekte Beweisansatz verfolgt jedoch das Ziel aufzuzeigen, daß dieses kontradiktorische Gegenteil der ursprünglichen Beweisbarkeitsbehauptung nicht zutrifft. Also wird behauptet, daß die Vereinigung der Konjugatformel "$FM_q \wedge (\neg p_u)$" mit dem Axiomensystem des zugrundeliegenden Beweissystems in sich widersprüchlich ist[20]. Die behauptete Inkonsistenz wird auf syntaktische Weise dadurch bestätigt, daß aus der Konjugatformel "$FM_q \wedge (\neg p_u)$" bezüglich des Beweissystems BS die inkonsistente Formel "$\emptyset$" abgeleitet wird[21]. Umgekehrt wird die Inkonsistenzbehauptung widerlegt, indem aufgezeigt wird, daß es unmöglich ist, aus der Konjugatformel im Beweissystem BS die inkonsistente Formel "$\emptyset$" abzuleiten. Folglich wird die ursprüngliche Beweisbarkeitsbehauptung für die problemspezifische Subjugatformel "$FM_q \rightarrow p_u$" dadurch auf indirekte Weise bestätigt oder widerlegt, daß die Ableitbarkeitsbehauptung "$FM_q, \neg p_u \vdash_{BS} \emptyset$"[22] der inkonsistenten Formel "$\emptyset$" bezüglich der Konjugatformel "$FM_q \wedge (\neg p_u)$" im Beweissystem BS bestätigt bzw. widerlegt wird[23].

Wegen der Gleichwertigkeit von modell- und beweistheoretischer Vorgehensweise auf der Grundlage von vollständigen und korrekten Inferenzkonzepten gilt schließlich auch: Das logische Entscheidungsproblem besitzt die positive Lösung "ja" genau dann, wenn die Ableitbarkeitsbehauptung[24] für die Konjugatformel "$FM_q \wedge (\neg p_u)$" und die inkonsistente Formel "$\emptyset$" bestätigt wird[25]. Andernfalls gilt die negative Lösung "nein" für die Widerlegung derselben Ableitbarkeitsbehauptung[26].

Abb. 28 auf der nächsten Seite faßt die semantischen und syntaktischen Formulierungsweisen für ein Entscheidungsproblem zusammen.

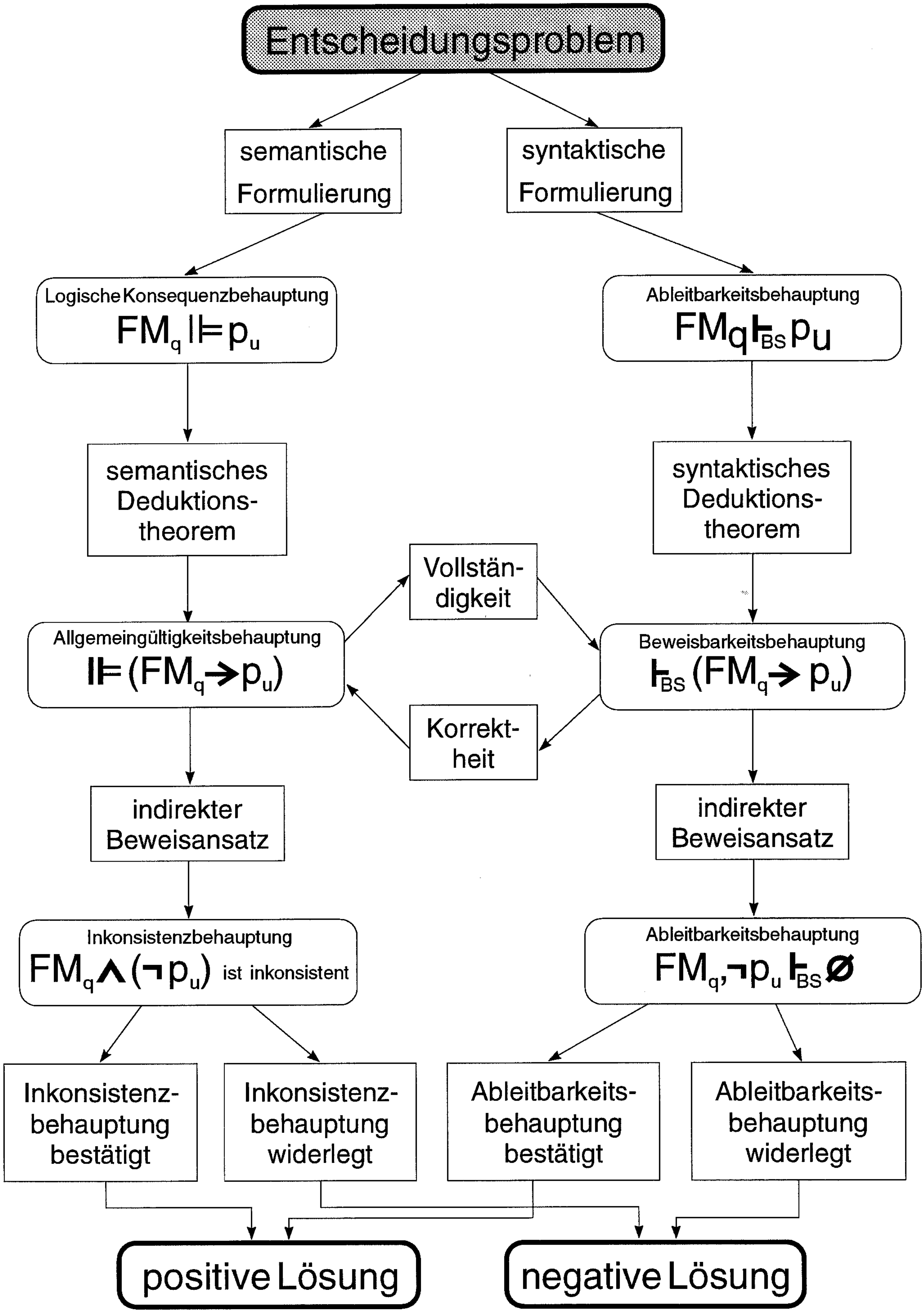

Abb. 28: Formulierungsweisen für ein Entscheidungsproblem

Ein simples Beispiel verdeutlicht die zuvor erläuterte indirekte Beweistechnik. Es ist bewußt einfach gehalten, um die Rückführung semantischer Fragestellungen auf syntaktische Untersuchungen zu veranschaulichen. Zugleich wird die Funktionsweise des kombinierten Unifizierungs- und Resolutionskonzepts demonstriert. Dieses Inferenzkonzept wurde in seinen Grundzügen bereits an früherer Stelle kurz charakterisiert: Die Unifizierung leistet eine konsistente Variablensubstitution. Die Resolutionsoperation gestattet es, Klauselmengen zu vereinfachen. Dabei werden komplementäre Literale aus unterschiedlichen Klauseln gestrichen, falls das eine Literal das Negat des jeweils anderen Literals ist. Auf der Basis des Resolutionstheorems wird die Inferenzstrategie verfolgt, aus einer zu untersuchenden Klauselmenge durch Resolventenbildung die leere Klauselmenge abzuleiten.

Ausgangspunkt ist eine Formelmenge FM_q, die aus einem Subjugat "$prä_1(X) \rightarrow prä_2(X)$" und einer konstanten atomaren Formel "$prä_1(c)$" besteht. Die Formelmenge wird als Klauselmenge notiert, in der das vorgenannte Subjugat durch seine äquivalente Klauselform ausgedrückt wird. Die Variable "X" aus den beiden atomaren Formeln "$prä_1(X)$" und "$prä_2(X)$" wird implizit durch einen Allquantor gebunden. Daher gilt:

$$FM_q \ :\Leftrightarrow\ \{\neg prä_1(X) \vee prä_2(X),\ prä_1(c)\}$$

Der Definitionsbereich der Variablen "X" umfaßt die Konstanten "a", "b" und "c". Zunächst wird behauptet, die Formel $prä_2(c)$ sei eine logische Konsequenz der Formelmenge FM_q. Es gilt also, das logische Entscheidungsproblem "$FM_q \models p_{u(1)}$" mit:

$$p_{u(1)} \ :\Leftrightarrow\ prä_2(c)$$

zu untersuchen. Aufgrund der indirekten Beweistechnik wird die Inkonsistenz der Konjugatformel "$FM_q \wedge (\neg prä_2(c))$" untersucht. Dies bereitet keine Schwierigkeiten, weil die Formel $p_{u(1)}$ keine Variablen und somit a fortiori auch keine Quantoren enthält. Mittels Unifizierungs- und Resolutionsoperationen läßt sich rasch die leere Klauselmenge $\{\} = \varnothing$ - die inkonsistente Formel "$\varnothing$" - ableiten. Zunächst wird die Konjugatformel, deren Inkonsistenz behauptet wird, als Klauselmenge dargestellt:

$$\begin{array}{ll} & FM_q \wedge (\neg prä_2(c)) \\ \Leftrightarrow & \{\neg prä_1(X) \vee prä_2(X),\ prä_1(c)\} \cup \{\neg prä_2(c)\} \\ \Leftrightarrow & \{\neg prä_1(X) \vee prä_2(X),\ prä_1(c),\ \neg prä_2(c)\} \end{array}$$

Aus der Unifizierung der Variablen "X" mit der Konstanten "c" und der Resolventenbildung für die komplementären, gemeinsam eliminierten Literale $prä_2(c)$ und $\neg prä_2(c)$ folgt eine neue Klauselmenge. Mit resolve(...) als Resolutionsoperator für die Resolventenbildung gilt hierfür:

$$resolve(\{\neg prä_1(X) \vee prä_2(X),\ prä_1(c),\ \neg prä_2(c)\}) \ = \ \{\neg prä_1(c),\ prä_1(c)\}$$

Die beiden verbliebenen Klauseln $\neg prä_1(c)$ und $prä_1(c)$ stellen unmittelbar komplementäre Literale dar, so daß als zweite Resolvente die leere Klauselmenge $\varnothing$ folgt:

$$resolve(\{\neg prä_1(c),\ prä_1(c)\}) \ = \ \{\} \ = \ \varnothing$$

Aufgrund des Resolutionstheorems ist wegen der Ableitbarkeit der leeren Klauselmenge die ursprüngliche Klauselmenge $\{\neg prä_1(X) \vee prä_2(X),\ prä_1(c),\ \neg prä_2(c)\}$ unerfüllbar. Also ist die ursprüngliche Inkonsistenzbehauptung für die Konjugatformel "$FM_q \wedge (\neg prä_2(c))$" bestätigt. Folglich besitzt das Entscheidungsproblem "$FM_q \models p_{u(1)}$" mit $p_{u(1)} :\Leftrightarrow prä_2(c)$ die positive Lösung: Die konstante Formel $prä_2(c)$ ist tatsächlich eine logische Konsequenz der Formelmenge FM_q[27].

Eine zweite Behauptung lautet, die Formel $prä_2(a)$ sei eine logische Konsequenz der Formelmenge FM_q. Analog zu den voranstehenden Ausführungen wird das logische Entscheidungsproblem "FM_q $p_{u(2)}$" mit $p_{u(2)}:\Leftrightarrow prä_2(a)$ untersucht. Es wird geprüft, ob sich aus der nunmehr relevanten Konjugatformel:

$$\begin{array}{ll} & FM_q \wedge (\neg prä_2(a)) \\ \Leftrightarrow & \{\neg prä_1(X) \vee prä_2(X), prä_1(c)\} \cup \{\neg prä_2(a)\} \\ \Leftrightarrow & \{\neg prä_1(X) \vee prä_2(X), prä_1(c), \neg prä_2(a)\} \end{array}$$

durch Unifizierungs- und Resolutionsoperationen die leere Klauselmenge $\emptyset$ ableiten läßt. Dies ist unmöglich. Da in dem hier gewählten simplen Beispiel der logische Suchraum grundsätzlich möglicher Inferenzen endlich und sehr klein ist, kann die Nichtableitbarkeit der leeren Klauselmenge mit geringem Ressourceneinsatz durch die *vollständige* Erforschung des Inferenzraums festgestellt werden. Es kommen nur zwei alternative Resolventenbildungen in Betracht, bei denen die Variable "X" entweder mit der Konstanten "c" oder aber mit der Konstanten "a" unifiziert wird. Dann lassen sich entweder die beiden komplementären Literale $\neg prä_1(c)$ und $prä_1(c)$ oder aber die beiden komplementären Literale $prä_2(a)$ und $\neg prä_2(a)$ eliminieren:

$$\begin{array}{l} resolve_1(\{\neg prä_1(X) \vee prä_2(X), prä_1(c), \neg prä_2(a)\}) = \{prä_2(c), \neg prä_2(a)\} \\ resolve_2(\{\neg prä_1(X) \vee prä_2(X), prä_1(c), \neg prä_2(a)\}) = \{\neg prä_1(a), prä_1(c)\} \end{array}$$

Wegen $c \neq a$ läßt sich keine der beiden Resolventen $\{prä_2(c), \neg prä_2(a)\}$ und $\{\neg prä_1(a), prä_1(c)\}$ weiter vereinfachen. Also ist es ausgeschlossen, die leere Klauselmenge abzuleiten. Daher ist die zunächst behauptete Inkonsistenz der Konjugatformel "$FM_q \wedge (\neg prä_2(a))$" widerlegt. Folglich besitzt das Entscheidungsproblem "$FM_q \models p_{u(2)}$" mit $p_{u(2)} :\Leftrightarrow prä_2(a)$ die negative Lösung: Die konstante Formel $prä_2(a)$ ist tatsächlich keine logische Konsequenz der Formelmenge FM_q.

Die voranstehenden Erläuterungen werden so modifiziert, daß die Formel $p_{u(3)}$ nunmehr keine atomare konstante Formel darstellt, sondern als Existenzformel eine Variable enthält: $p_{u(3)} :\Leftrightarrow \exists(X): prä_2(X)$. Wiederum wird behauptet, die Existenzformel "$\exists(X): prä_2(X)$" sei eine logische Konsequenz der Formelmenge FM_q. Es wird also das logische Entscheidungsproblem "$FM_q \models p_{u(3)}$" mit $p_{u(3)} :\Leftrightarrow \exists(X): prä_2(X)$ untersucht. In die zugehörige Konjugatformel geht das Negat der Existenzformel "$\exists(X): prä_2(X)$" als Allformel "$\forall(X): \neg prä_2(X)$" ein. Da in Formeln, die in konjunktiver Normalform notiert werden, für alle Variablen implizit Allquantoren vorausgesetzt werden, gilt für diese Konjugatformel und ihre äquivalente Klauselmenge:

$$\begin{array}{ll} & FM_q \wedge (\neg prä_2(X)) \\ \Leftrightarrow & \{\neg prä_1(X) \vee prä_2(X), prä_1(c)\} \cup \{\neg prä_2(X)\} \\ \Leftrightarrow & \{\neg prä_1(X) \vee prä_2(X), prä_1(c), \neg prä_2(X)\} \end{array}$$

Aus der Resolventenbildung für die komplementären Literale $prä_2(X)$ und $\neg prä_2(X)$ folgt:

$$resolve(\{\neg prä_1(X) \vee prä_2(X), prä_1(c), \neg prä_2(X)\}) = \{\neg prä_1(X), prä_1(c)\}$$

Durch Unifizierung der Variablen "X" mit der Konstanten "c" und Resolventenbildung für die beiden verbliebenen komplementären Literale $\neg prä_1(c)$ und $prä_1(c)$ wird die gesuchte leere Klauselmenge $\emptyset$ abgeleitet:

$$\text{resolve}(\{\neg\text{prä}_1(c), \text{prä}_1(c)\}) = \{\} = \varnothing$$

Also ist die ursprüngliche Inkonsistenzbehauptung für die Konjugatformel "$FM_q \wedge (\neg\text{prä}_2(X))$" bestätigt. Folglich besitzt das Entscheidungsproblem "$FM_q \models p_{u(3)}$" mit $p_{u(3)} :\Leftrightarrow \exists(X):\text{prä}_2(X)$ die positive Lösung: Die Existenzformel "$\exists(X):\text{prä}_2(X)$" ist tatsächlich eine logische Konsequenz der Formelmenge FM_q. Der Existenzbeweis erfolgte sogar konstruktiv. Denn aus der Unifizierung der Variablen "X" durch die Konstante "c" ergibt sich unmittelbar, daß die Existenzformel *zumindest* für die Variablenbindungsfunktion vb mit vb: $X \rightarrow c$ eine logische Konsequenz der Formelmenge FM_q darstellt. Darüber hinaus ließe sich durch die vollständige Erforschung des logischen Suchraums feststellen, daß die Unifizierung der Variablen "X" mit der Konstanten "c" die einzige Möglichkeit bildet, um die leere Klauselmenge abzuleiten. Daher gilt sogar das stark konstruktive Ergebnis: Die Existenzformel stellt *genau* für die Variablenbindungsfunktion vb mit vb: $X \rightarrow c$ eine logische Konsequenz der Formelmenge FM_q dar.

Abschließend wird eine vierte Behauptung untersucht, um zu veranschaulichen, wie aus dem Übersehen notwendiger Skolemisierungs-Operationen ein Fehlschluß resultieren kann. Es wird behauptet, daß die Allformel "$\forall(X):\text{prä}_2(X)$" eine logische Konsequenz der Formelmenge FM_q sei. Es gilt also, das logische Entscheidungsproblem "$FM_q \models p_{u(4)}$" mit $p_{u(4)} :\Leftrightarrow \forall(X):\text{prä}_2(X)$ zu untersuchen. In die zugehörige Konjugatformel geht das Negat der Allformel "$\forall(X):\text{prä}_2(X)$" als Existenzformel "$\exists(X):\neg\text{prä}_2(X)$" ein. Würde übersehen, daß Formeln in konjunktiver Normalform keine Existenzquantoren enthalten dürfen, so ließe sich die Konjugatformel und ihre äquivalente Klauselmenge - fehlerhaft - darstellen als:

$$\begin{array}{ll} & FM_q \wedge (\neg\text{prä}_2(X)) \\ \Leftrightarrow & \{\neg\text{prä}_1(X) \vee \text{prä}_2(X), \text{prä}_1(c)\} \cup \{\neg\text{prä}_2(X)\} \\ \Leftrightarrow & \{\neg\text{prä}_1(X) \vee \text{prä}_2(X), \text{prä}_1(c), \neg\text{prä}_2(X)\} \end{array}$$

Dies ist die gleiche Äquivalenz wie für das voranstehend erörterte dritte Entscheidungsproblem "$FM_q \models p_{u(3)}$". Daher würde gefolgert, die ursprüngliche Inkonsistenzbehauptung für die Konjugatformel "$FM_q \wedge (\neg\text{prä}_2(X))$" sei bestätigt. Folglich müsse das Entscheidungsproblem "$FM_q \models p_{u(4)}$" mit $p_{u(4)} :\Leftrightarrow \forall(X):\text{prä}_2(X)$ die positive Lösung besitzen: Die Allformel "$\forall(X):\text{prä}_2(X)$" sei tatsächlich eine logische Konsequenz der Formelmenge FM_q. Diese Folgerung wäre aber ein Fehlschluß. Denn mit $DB(X)=\{a,b,c\}$ als vorausgesetztem Definitionsbereich der Variablen "X" bedeutete die Allformel "$\forall(X):\text{prä}_2(X)$" u.a. auch die Behauptungen, die konstanten atomaren Formeln $\text{prä}_2(a)$ und $\text{prä}_2(b)$ seinen logische Konsequenzen der Formelmenge FM_q. Die erste dieser beiden Behauptungen wurde bereits im zweiten Entscheidungsproblem "$FM_q \models p_{u(2)}$" mit $p_{u(2)} :\Leftrightarrow \text{prä}_2(a)$ zurückgewiesen. Die zweite Behauptung ließe sich auf die gleiche Weise widerlegen. Die Ursache dieses doppelten Fehlschlusses ist offensichtlich: Die Existenzformel "$\exists(X):\neg\text{prä}_2(X)$" hätte nicht in die Konjugatformel unmittelbar übernommen werden dürfen. Zunächst wäre es erforderlich gewesen, den Existenzquantor mit Hilfe einer SKOLEM-Funktion zu eliminieren. Dann hätte die Konjugatformel in konjunktiver und skolemisierter Normalform eine andere als die o.a. fehlerhafte Gestalt angenommen. Der Fehlschluß wäre vermieden worden. Da die notwendigen Skolemisierungs-Operationen recht aufwendig ausfallen, werden sie hier nicht konkret ausgeführt.

Allerdings wurde bisher nur gezeigt, daß die Untersuchung von Entscheidungsproblemen, die zunächst in semantischer Weise das Vorliegen logischer Konsequenzen behaupten, auf rein syntaktisch definierte Inferenzen zurückgeführt werden kann. Nunmehr wird das Spektrum denkmöglicher Untersuchungsergebnisse ausdifferenziert. Dabei wird das Ziel verfolgt, den terminologischen Rahmen abzustecken, in den später das zentrale Phänomen der prädikatenlogischen Semi-Entscheidbarkeit präzise eingebettet werden kann.

Ein Entscheidungsproblem heißt lösbar, wenn sowohl seine positive als auch seine negative Lösung wohldefiniert[28] sind. Andernfalls - wenn seine positive oder negative Lösung nicht wohldefiniert sind - gilt das Entscheidungsproblem als unlösbar[29]. Unlösbare Entscheidungsprobleme spielen in dieser Arbeit grundsätzlich keine Rolle[30]. Es werden ausschließlich lösbare Entscheidungsprobleme behandelt[31]. Allerdings bedeutet die Lösbarkeit eines Entscheidungsproblems noch keineswegs, daß auch schon seine Lösung bekannt wäre. Denn die Lösbarkeit eines Problems involviert nur die Wohldefiniertheit, nicht aber die Kenntnis seiner Lösung[32]. Um diese Lösungskenntnis zu erlangen, wird das Instrument der Entscheidungsalgorithmen eingeführt.

Ein problemspezifischer Entscheidungsalgorithmus[33] ist ein Verfahren, mit dessen Hilfe sich für jedes Mitglied aus einer Klasse gleichartiger Entscheidungsprobleme in endlicher Zeit[34] feststellen läßt, ob das jeweils untersuchte - als lösbar vorausgesetzte - Entscheidungsproblem eine positive oder[35] negative Lösung besitzt. Es wird auch von einem Lösungsalgorithmus für "das" Entscheidungsproblem gesprochen[36]. Wesentlich ist, daß ein Entscheidungsalgorithmus jede tatsächliche Problemlösung erkennt, und zwar unabhängig davon, ob es sich um eine positive oder negative Lösung handelt. Dieser Aspekt wird später bei der Herausarbeitung der prädikatenlogischen Semi-Entscheidbarkeit eine herausragende Rolle spielen.

Ein lösbares Entscheidungsproblem heißt genau dann gelöst, wenn bekannt ist, *ob* hierfür ein problemspezifischer Entscheidungsalgorithmus existiert. Solange diese Kenntnis noch nicht vorliegt, gilt das lösbare Entscheidungsproblem als - vorläufig - ungelöst[37]. Die Lösung eines Entscheidungsproblems betrifft hier also nur die Existenz, nicht aber die Anwendung eines problemspezifischen Entscheidungsalgorithmus[38].

Ein gelöstes Entscheidungsproblem ist genau dann entscheidbar, wenn bekannt ist, *daß* mindestens ein problemspezifischer Entscheidungsalgorithmus existiert. Dieses Existenzwissen alleine umfaßt aber noch nicht die Kenntnis eines entsprechenden Entscheidungsalgorithmus. Mit der Hilfe von nonkonstruktiven Beweistechniken kann durchaus gezeigt werden, daß für ein Entscheidungsproblem mindestens ein Entscheidungsalgorithmus existiert, ohne jedoch einen Entscheidungsalgorithmus konkret zu kennen[39]. Solange dies der Fall ist, gilt das betrachtete Entscheidungsproblem als entscheidbar, aber - vorläufig - unentschieden. Erst wenn (mindestens) ein problemspezifischer Entscheidungsalgorithmus bekannt ist und auch auf das Entscheidungsproblem angewandt wurde, liegt ein entschiedenes Entscheidungsproblem vor. Das Entscheidungsergebnis ist die gesuchte positive oder negative Lösung des Entscheidungsproblems.

Dagegen liegt ein unentscheidbares Entscheidungsproblem vor, falls nachgewiesen wird, daß kein problemspezifischer Entscheidungsalgorithmus existieren kann. Ein solches unentscheidbares Entscheidungsproblem steht im Mittelpunkt der nachfolgenden Ausführungen zur operationalen Dimension der Prädikatenlogik 1. Stufe. Die zuvor erläuterten Eigenschaften von Entscheidungsproblemen werden dabei als bekannt vorausgesetzt. Abb. 29 auf der nächsten Seite verdeutlicht ihren Definitionszusammenhang.

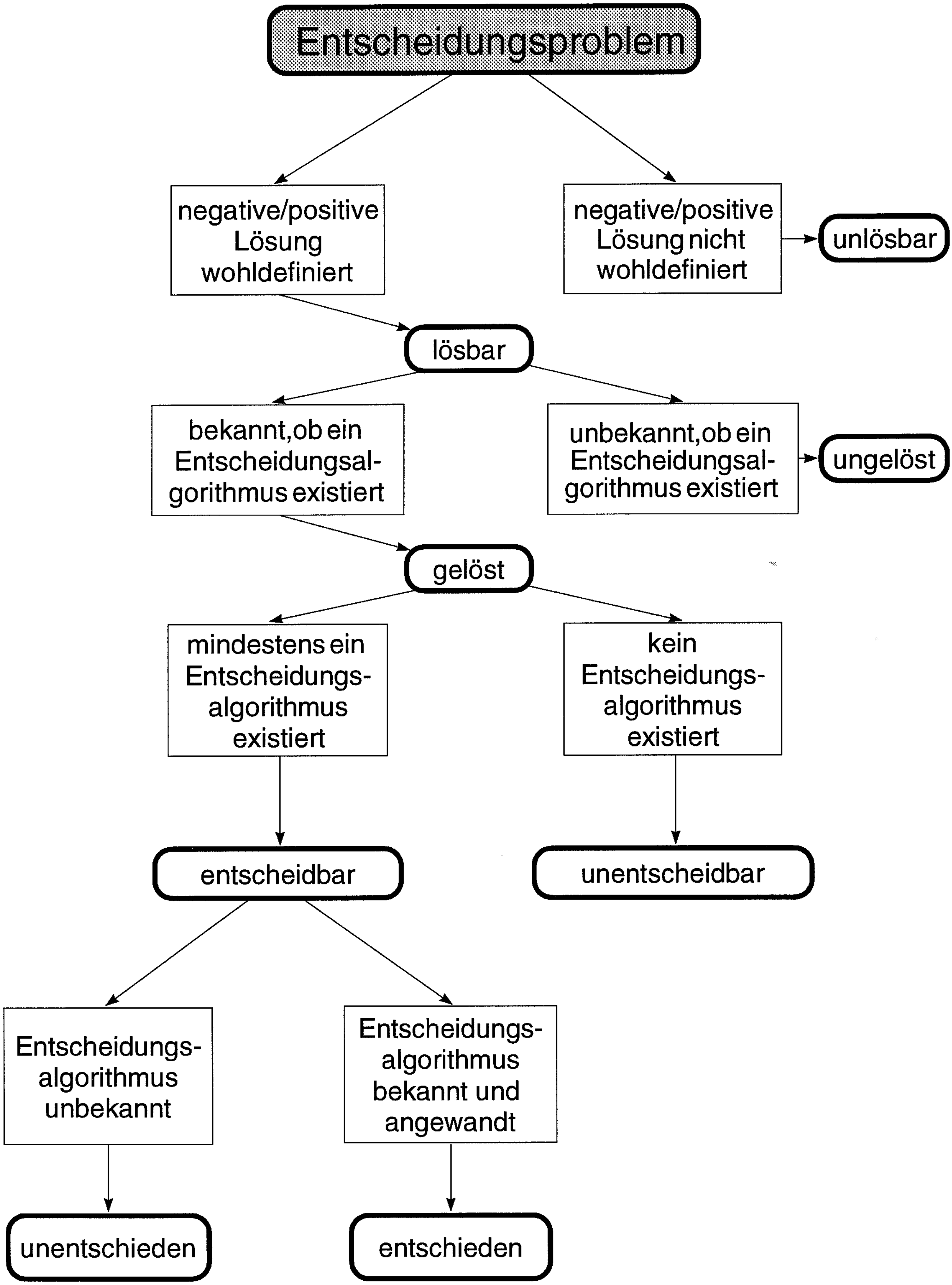

Abb. 29: Eigenschaften von Entscheidungsproblemen

Für die Lösung des oben eingeführten logischen Entscheidungsproblems sind mehrere vollständige und korrekte Inferenzkonzepte bekannt. Die Umsetzung eines dieser Inferenzkonzepte in eine konkret ausführbare Inferenzprozedur könnte einen Entscheidungsalgorithmus für das Entscheidungsproblem darstellen. Auf den ersten Blick scheint dies trivial zu sein. Dennoch werden die anschließenden Ausführungen zur operationalen Dimension der Prädikatenlogik verdeutlichen, daß grundsätzlich keine Inferenzprozedur mit der Qualität eines Entscheidungsalgorithmus im oben definierten Sinne[40] existieren kann.

Die Unentscheidbarkeit der Prädikatenlogik 1. Stufe[41] errichtet eine erste generelle Leistungsschranke für alle prädikatenlogischen Inferenzprozeduren. Sie erstreckt sich auf die Gesamtheit aller Programme, die jeweils ein Entscheidungsproblem "$\vdash_{BS} (FM_q \rightarrow p_u)$?" in einer prädikatenlogischen Programmiersprache ausdrücken. Insbesondere gilt sie auch für alle solchen PROLOG-Programme[42]. Aufgrund der prädikatenlogischen Unentscheidbarkeit kann die Ausführung eines PROLOG-Programms zu keinem wohldefinierten Abschluß gelangen, falls hierbei ein *unentscheidbares* Entscheidungsproblem "$\vdash_{BS} (FM_q \rightarrow p_u)$?" untersucht wird. Solche Entscheidungsprobleme existieren zwar[43], spielen aber für die praktische Anwendung der Prädikatenlogik auf betriebswirtschaftliche Probleme im allgemeinen keine nennenswerte Rolle.

Anders verhält es sich mit der Formulierung eines speziellen Entscheidungsproblems, das als Halteproblem für TURING-Automaten[44] bekannt geworden ist. TURING-Automaten können aufgrund der CHURCH/POST/TURING-These[45] als ein abstraktes Konzept aufgefaßt werden, mit dessen Hilfe sich beliebige Algorithmen spezifizieren lassen[46]. Aus der Unentscheidbarkeit des Halteproblems für TURING-Automaten folgt eine fundamentale Erkenntnis: Es ist unmöglich, für *jeden beliebigen* TURING-Automaten und für *jede beliebige* Eingabeinformation *a priori* zu entscheiden, ob der Automat - angesetzt auf jene Eingabeinformation - seine informationsverarbeitenden Operationen in endlicher Zeit vollständig auszuführen vermag und danach anhält.

Jedes PROLOG-Programm, das ein prädikatenlogisches Formelsystem ausdrückt, stellt zusammen mit einer PROLOG-Kontrollstruktur[47] einen Algorithmus dar[48]. Er wendet das kombinierte Unifizierungs- und Resolutionskonzept auf das implementierte Formelsystem an. Aufgrund der o.a. These von CHURCH, POST und TURING kann dieser Algorithmus durch einen TURING-Automaten dargestellt werden. Folglich läßt sich auch jedes Paar aus einem PROLOG-Programm und einer PROLOG-Kontrollstruktur als ein TURING-Automat auffassen. Die Sprache PROLOG nutzt - wie alle höheren Programmiersprachen - das Leistungsvermögen der TURING-Automaten vollständig aus[49]. Daher fällt die Gesamtheit aller PROLOG-Programme und -Kontrollstrukturen mit der Gesamtheit aller TURING-Automaten zusammen[50].

Wegen der TURING-Mächtigkeit der Programmiersprache PROLOG gilt das Halteproblem der TURING-Automaten für diese Sprache ebenso: Das Entscheidungsproblem, für *jedes beliebige* Paar aus einem PROLOG-Programm und einer PROLOG-Kontrollstruktur[51] und für *jede* Initialisierung der Programmausführung durch eine *beliebige* Eingabeinformationen *a priori* festzustellen, ob die Programmausführung nach endlicher Zeit erfolgreich abgeschlossen werden kann, ist unentscheidbar. Daher kann es grundsätzlich keinen Entscheidungsalgorithmus geben, mit dessen Hilfe sich das Terminieren des Ausführens *jedes* PROLOG-Programms[52] überprüfen ließe, ohne die Programme jeweils selbst auszuführen[53]. Deswegen besitzen alle PROLOG-Programme potentiell nicht-terminierenden[54] Charakter. Es läßt sich niemals ausschließen, daß die Ausführung eines PROLOG-Programms unendlich fortgesetzt werden kann[55], ohne jemals mit derjenigen Ausgabeinformation abzuschließen, die aufgrund der Programmspezifikation erwartet wird. In der Regel äußert sich eine solche nicht-terminierende Programmausführung als eine Endlosschleife[56]. Daher verbietet es die Eigenart des Halteproblems, für *jedes* PROLOG-Programm vor seiner tatsächlichen Ausführung zu erkennen, ob es zu einer Endlosschleife kommen wird oder nicht[57].

Die Unentscheidbarkeit des Halteproblems wirkt sich auf die Bearbeitung eines beliebigen logischen Entscheidungsproblems " $\vdash_{BS} (FM_q \rightarrow p_u)$?" gravierend aus. Denn die Implementierung des Problems auf einem Automatischen Informationsverarbeitungssystem durch die Programmiersprache PROLOG und deren Kontrollstruktur läßt sich als ein TURING-Automat auffassen[58]. Da sich das Terminieren von TURING-Automaten im allgemeinen nicht entscheiden läßt, ist es unmöglich vorherzusehen, ob die Programmausführung für jedes untersuchte Entscheidungsproblem innerhalb beliebig großer, aber endlicher Zeitschranken durch eine definite Antwort "ja" oder "nein" auf die Entscheidungsfrage " $\vdash_{BS} (FM_q \rightarrow p_u)$?" erfolgreich enden wird. Statt dessen ist immer mit der Existenz von Entscheidungsproblemen zu rechnen, bei deren Untersuchung die Programmausführung wegen beschränkt verfügbarer Ausführungsressourcen irgendwann erfolglos abgebrochen werden muß. Daher kann es keinen Entscheidungsalgorithmus geben, der für jedes logische Entscheidungsproblem dessen tatsächliche - positive oder negative - Lösung erkennen könnte.

Das gleiche Resultat läßt sich ebenso erzielen, ohne die Unentscheidbarkeit des speziellen Halteproblems von TURING-Automaten voraussetzen zu müssen. Statt dessen kann auch auf die Unentscheidbarkeit des eng verwandten[59], aber allgemein formulierten Gültigkeitsproblems der Prädikatenlogik zurückgegriffen werden. Ihr zufolge kann es prinzipiell keinen Algorithmus geben, der *alle* prädikatenlogisch darstellbaren Entscheidungsprobleme " $\vdash_{BS} (FM_q \rightarrow p_u)$?" in endlicher Zeit lösen würde[60]. Dies wurde schon früher als Charakteristikum der Prädikatenlogik herausgestellt. A fortiori kann auch für die Programmiersprache PROLOG grundsätzlich kein Entscheidungsalgorithmus formuliert werden, mit dessen Hilfe es möglich wäre, *jedes* prädikatenlogische Entscheidungsproblem in endlicher Zeit zu bewältigen[61].

Das potentielle Nichtterminieren von Verfahren für die Entscheidung prädikatenlogisch formulierter Probleme stellt einen zentralen Aspekt der operationalen Semantik der Prädikatenlogik dar[62]. Anschließend wird eine praktische Konsequenz näher beleuchtet, die aus diesem Sachverhalt für alle prädikatenlogischen Objektmodellierungen resultiert. Es handelt sich um das Phänomen der Semi-Entscheidbarkeit.

Die Semi-Entscheidbarkeit[63] der Prädikatenlogik bedeutet, daß ein Entscheidungsproblem " $\vdash_{BS} (FM_q \rightarrow p_u)$?" nur dann mit Sicherheit in endlicher Zeit gelöst werden kann, wenn die untersuchte Subjugatformel tatsächlich allgemeingültig ist. Für Formeln[64], die nicht allgemeingültig sind[65], ist ein derart terminierendes Entscheidungsverhalten dagegen grundsätzlich nicht mehr sichergestellt[66]. Das hat zur Folge, daß für das Entscheidungsproblem " $\vdash_{BS} (FM_q \rightarrow p_u)$?" *kein* Entscheidungsalgorithmus im früher definierten Sinne existiert. Nicht nur die bisher bekannten prädikatenlogischen Inferenzprozeduren, sondern alle überhaupt möglichen Inferenzprozeduren sind nicht in der Lage zu garantieren, für jede Ausprägung des Entscheidungsproblems " $\vdash_{BS} (FM_q \rightarrow p_u)$?" in endlicher Zeit sowohl dessen positive als auch dessen negative Lösung zu erkennen.

Mitunter wird die Semi-Entscheidbarkeit der Prädikatenlogik 1. Stufe auch dadurch ausgedrückt, daß auf positive und negative Verfahren für die Lösung des allgemeinen prädikatenlogischen Gültigkeitsproblems Bezug genommen wird[67]: Ein positives (negatives) Entscheidungsverfahren ist in der Lage, jede tatsächlich existierende positive (negative) Lösung des Gültigkeitsproblems als solche in endlicher Zeit zu erkennen. Falls eine untersuchte Formel tatsächlich (nicht-)allgemeingültig ist, so wird dies durch ein positives (negatives) Entscheidungsverfahren mit Sicherheit nachgewiesen[68]. Aus dieser Perspektive bedeutet die Semi-Entscheidbarkeit der Prädikatenlogik, daß für sie zwar positive, nicht aber negative Entscheidungs*verfahren* existieren[69]. Für einen Entscheidungs*algorithmus* war dagegen gefordert worden, er solle nicht nur jede positive, sondern auch jede negative Lösung erkennen können.

Es ist also bekannt, daß weder für das allgemeine prädikatenlogische Gültigkeitsproblem noch für seine spezielle Variante - das logische Entscheidungsproblem " $\vdash_{BS} (FM_q \rightarrow p_u)$?" - ein Entscheidungsalgorithmus existiert. Folglich sind Gültigkeits- und Entscheidungsproblem *unent-*

scheidbar[70]. Die Semi-Entscheidbarkeit der Prädikatenlogik führt daher zu einer weiteren Differenzierung unentscheidbarer Entscheidungsprobleme: Ein unentscheidbares Entscheidungsproblem heißt semi-entscheidbar[71], falls bekannt ist, daß hierfür mindestens ein positives (negatives), aber kein negatives (positives) Entscheidungsverfahren existiert. Dagegen wird von einem vollkommen unentscheidbaren Entscheidungsproblem gesprochen, wenn dafür nachweislich weder ein positives noch ein negatives Entscheidungsverfahren existiert. Abb. 30 auf der nächsten Seite zeigt die entsprechende Verfeinerung des Definitionszusammenhangs von Problemeigenschaften, der erstmals in Abb. 29 verdeutlicht wurde. Im folgenden spielen vollkommen unentscheidbare Entscheidungsprobleme keine Rolle[72].

Es besteht eine bemerkenswerte Diskrepanz zwischen der fundamentalen Bedeutung der Semi-Entscheidbarkeit und ihrer Würdigung bei Anwendungen der logischen Programmierung[73]. Daher wird sie nachfolgend etwas ausführlicher dargelegt[74]. Es wird eine Inferenzprozedur betrachtet, die in einem Automatischen Informationsverarbeitungssystem auf der Grundlage eines implementierten prädikatenlogischen Inferenzkonzepts ausgeführt wird:

- Die Inferenzprozedur terminiert grundsätzlich immer, falls sie auf ein logisch programmiertes Entscheidungsproblem angewendet wird, das sich tatsächlich *positiv* lösen läßt. Dabei wird die positive Problemlösung korrekt erkannt.
- Das Terminieren der gleichen Inferenzprozedur kann jedoch nicht mehr garantiert werden, falls sie auf ein Entscheidungsproblem angewendet wird, dessen tatsächliche Lösung *negativ* ist. Es besteht daher immer die Gefahr, daß sich die Prozedurausführung z.B. in einer Endlosschleife verfängt und irgendwann erfolglos abgebrochen werden muß. Aus einem solchen Ausführungsabbruch folgt keine Erkenntnis über die tatsächliche Problemlösung.

Durch diesen ambivalenten und unsicherheitsstiftenden Charakter weicht das Phänomen der Semi-Entscheidbarkeit deutlich von den früher behandelten prädikatenlogischen Charakteristika der Unentscheidbarkeit und Vollständigkeit ab[75].

Unter der Zuverlässigkeit einer Inferenzprozedur wird ihre Qualität verstanden, bei der Anwendung auf ein *beliebiges* Entscheidungsproblem die tatsächliche Problemlösung korrekt und mit Sicherheit zu erkennen. Bei Semi-Entscheidbarkeit kann es aufgrund der Existenzmöglichkeit von Entscheidungsproblemen mit negativer Problemlösung keine zuverlässige Inferenzprozedur geben. Die Prozedurzuverlässigkeit gilt nur partiell für diejenige "Hälfte" aller Entscheidungsprobleme, deren Lösung jeweils positiv ist. Die partielle Unzuverlässigkeit ist der *wesentliche Gehalt* der Semi-Entscheidbarkeit[76]. Sie hat zur Folge, daß ein prädikatenlogischer Problemlösungsversuch nur so lange zuverlässig ist, wie er innerhalb der vorgegebenen realen Ressourcenbeschränkungen für die Ausführung von Inferenzprozeduren erfolgreich terminiert. Die Korrektheit prädikatenlogischer Inferenzkonzepte garantiert dann, daß die bewiesene problemspezifische Subjugatformel auch tatsächlich allgemeingültig ist. Wenn dagegen eine Inferenzprozedur bei Erreichen der ressourcenbedingten Ausführungsgrenzen abgebrochen werden muß, bleibt die Lösung des Entscheidungsproblems indeterminiert. Denn der erfolglose Abbruch einer Inferenzprozedur läßt sich stets auf zwei Weisen interpretieren[77]: Es könnte eine positive Problemlösung jenseits der Ressourcenbeschränkungen liegen. Ebenso wäre aber auch eine unendliche Prozedurfortsetzung deswegen möglich, weil die tatsächliche, aber unbekannte Lösung des untersuchten Entscheidungsproblems negativ ist. Was tatsächlich der Fall ist, kann nicht entschieden werden.

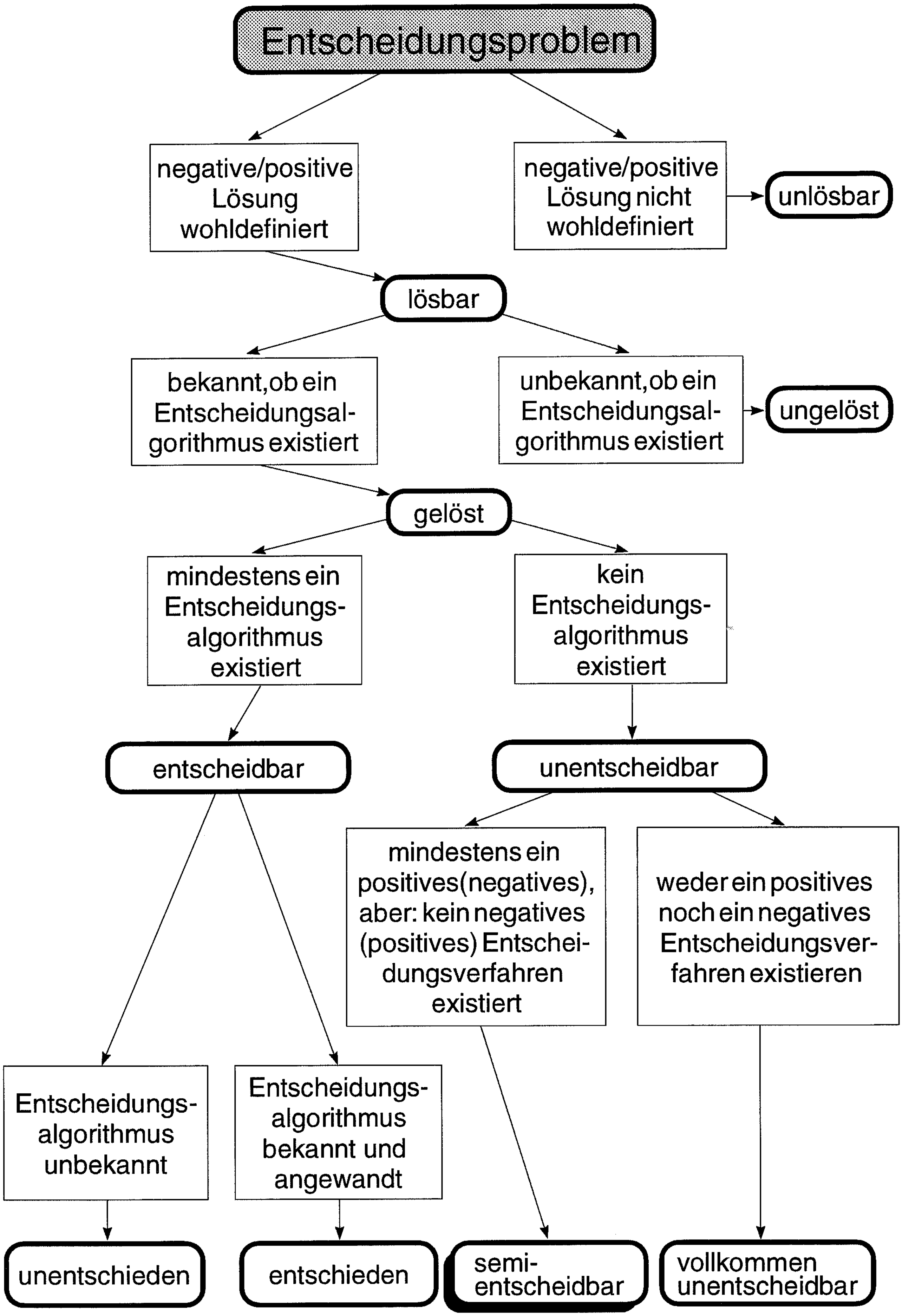

Abb. 30: Eigenschaften von Entscheidungsproblemen (verfeinert)

Für ein PROLOG-Programm besteht also immer dann die Gefahr von ambivalenten Endlosschleifen, wenn die Allgemeingültigkeit einer Formel behauptet wird, die tatsächlich nicht allgemeingültig ist. Leider ist bei der Formulierung eines Entscheidungsproblems "$\models (FM_q \rightarrow p_u)$?" die Allgemeingültigkeit der untersuchten Formel "$FM_q \rightarrow p_u$" in der Regel nicht bekannt, sondern fraglich[78]. Daher läßt sich grundsätzlich nicht ausschließen, die Allgemeingültigkeit einer nicht-allgemeingültigen Formel zu behaupten. Folglich läßt sich das Phänomen der prädikatenlogischen Semi-Entscheidbarkeit grundsätzlich nicht vermeiden. Dies gilt - im Gegensatz zur oben angesprochenen Unentscheidbarkeit - nicht nur für unentscheidbare, sondern auch für entscheidbare Entscheidungsprobleme. Denn ein PROLOG-Programm kann sich aufgrund der partiellen Unzuverlässigkeit aller prädikatenlogischen Inferenzkonzepte auch dann in einer Endlosschleife verfangen, wenn für das untersuchte Entscheidungsproblem "an sich" entweder eine positive oder negative Lösung existiert, aber die korrekte Lösung a priori unbekannt ist.

Wenn betriebswirtschaftlich interessante Sachverhalte als prädikatenlogische Entscheidungsprobleme modelliert werden[79], resultieren im allgemeinen entscheidbare Entscheidungsprobleme, deren korrekte Lösung während der Problemformulierung unbekannt ist. Daher besitzt die prädikatenlogische Semi-Entscheidbarkeit auch für praktische Modellierungsaufgaben - und somit auch für die Modellierung Flexibler Fertigungssysteme - fundamentale Bedeutung.

Allerdings läßt sich ein Ausweg aus dieser konzeptionellen Leistungsschranke vorstellen. Als Quelle der Semi-Entscheidbarkeit wurde das fehlende Wissen identifiziert, ob die Subjugatformel "$FM_q \rightarrow p_u$", deren Allgemeingültigkeit in einem Entscheidungsproblem ursprünglich behauptet wird, tatsächlich allgemeingültig ist. Diese Wissenslücke läßt sich unter günstigen Umständen[80] dadurch überbrücken, daß zwei kontradiktorische Entscheidungsprobleme untersucht werden[81]. Das erste Entscheidungsproblem ist mit dem ursprünglich vorgegebenen Problem identisch. Im hierzu kontradiktorischen zweiten Entscheidungsproblem wird dagegen die Erfüllbarkeit der negierten Subjugatformel, also die Erfüllbarkeit des Konjugats "$FM_q \wedge (\neg p_u)$" behauptet. Dies entspricht der Behauptung, das ursprüngliche Entscheidungsproblem besitze eine negative Lösung. Die Erfüllbarkeit der Konjugatformel läßt sich dadurch beweisen, daß die Unmöglichkeit aufgezeigt wird, aus ihr die kontradiktorische Formel "$\varnothing$" abzuleiten.

Wenn der logische Inferenzraum[82] eines Entscheidungsproblems endlich ist, kann diese Unmöglichkeit durch vollständige Untersuchung des Inferenzraums in endlicher Zeit festgestellt werden. Daher läßt sich die Semi-Entscheidbarkeit bei endlichen Inferenzräumen dadurch überwinden, daß zwei kontradiktorische Entscheidungsprobleme formuliert und parallel bearbeitet werden[83]: Falls die Subjugatformel "$FM_q \rightarrow p_u$" aus dem ersten Entscheidungsproblem tatsächlich allgemeingültig ist, terminiert ein hierauf angewandtes prädikatenlogisches Inferenzkonzept mit dem Beweis der Subjugatformel "$FM_q \rightarrow p_u$". Die "an sich" existierende positive Lösung des ursprünglichen Entscheidungsproblems wird in endlicher Zeit als solche erkannt. Falls die Subjugatformel aus dem ersten Entscheidungsproblem tatsächlich nicht allgemeingültig ist, muß die Konjugatformel "$FM_q \wedge (\neg p_u)$" des kontradiktorischen zweiten Entscheidungsproblems erfüllbar sein. Dann terminiert auch hier eine prädikatenlogische Inferenzprozedur, nachdem sie den Inferenzraum vollständig durchsucht hat und hierbei aus der Konjugatformel "$FM_q \wedge (\neg p_u)$" keine kontradiktorische Formel "$\varnothing$" abzuleiten vermochte[84]. Folglich wird die "an sich" existierende negative Lösung des ursprünglichen Entscheidungsproblems wiederum als solche in endlicher Zeit erkannt.

Leider gilt diese Überwindung der Semi-Entscheidbarkeit nur unter der einschränkenden Voraussetzung, daß der logische Inferenzraum endlich ist. Dies kann aber für Entscheidungsprobleme nicht garantiert werden[85]. Da es bei der Formulierung eines Entscheidungsproblems im allgemeinen unbekannt ist, ob sein Inferenzraum endlich ist[86], läßt sich auch nicht erkennen, ob die Formulierung zweier kontradiktorischer Entscheidungsprobleme das Terminieren der Inferenzprozeduren garantiert. Folglich kann die prädikatenlogische Semi-Entscheidbarkeit auch auf diesem Wege nicht generell überwunden werden[87]. Daher drohen aufgrund der Semi-Entscheidbarkeit für PROLOG-Programme auch dann noch Endlosschleifen, wenn die Programme "an sich" entscheidbare Entscheidungsprobleme ausdrücken.

Abschließend könnte der Einwand[88] erhoben werden, das Phänomen der Semi-Entscheidbarkeit gelte "nur" für logische Entscheidungsprobleme. Die dort untersuchten logischen Konsequenzen "$FM_q \models p_u$" und äquivalenten Allgemeingültigkeitsbehauptungen "$\models (FM_q \rightarrow p_u)$" könnten zwar von theoretischem Interesse sein. Für den praktischen Umgang mit prädikatenlogisch formulierten Objektmodellen blieben solche abstrakten Problemstellungen jedoch unerheblich. Denn in der Modellierungspraxis seien nur Untersuchungen *einzelner* Behauptungen von Interesse. *Logische* Probleme und *allgemeingültige* Formeln wären irrelevant.

Ein solcher Einwand übersieht jedoch die fundamentale Bedeutung, die das Phänomen der Semi-Entscheidbarkeit *auch für praktische* Entscheidungsprobleme besitzt. Die Einlassung, es interessiere nur die Gültigkeit *einzelner* Behauptungen, verkennt nämlich zwei wesentliche Sachverhalte: die Bedeutung des Hintergrundwissens und die Qualität der prädikatenlogischen Allgemeingültigkeit.

Die Behauptung der Gültigkeit einer einzelnen Formel (Referenzformel) p_u kann strenggenommen nur entweder dogmatisch gesetzt oder aber auf die Gültigkeit anderer Behauptungen zurückgeführt werden[89]. Der erste Fall interessiert hier nicht weiter. Denn Formeln, deren Gültigkeit a priori supponiert wird, brauchen nicht mehr hinsichtlich ihrer behaupteten Gültigkeit untersucht zu werden. Im zweiten Fall erlangen aber diejenigen Formeln Gewicht, die bereits als gültig vorausgesetzt werden müssen, um die Gültigkeit der Referenzformel untersuchen zu können. Jene Formeln und ihre präsupponierte Gültigkeit bilden das Hintergrundwissen, das die Gültigkeitsuntersuchung der Referenzformel stützt[90]. Dieses untersuchungsstützende Hintergrundwissen wurde bei der Erörterung der prädikatenlogischen Semi-Entscheidbarkeit als Formelmenge FM_q[91] offengelegt[92].

Das *logische* Entscheidungsproblem "$FM_q \models p_u$" befaßt sich daher mit der Aufgabe, die behauptete Gültigkeit der Formel p_u vor dem Hintergrund zu untersuchen, daß alle Formeln aus der Menge FM_q als gültig akzeptiert werden. Dies bedeutet nichts anderes als das *praktische* Entscheidungsproblem, die behauptete Gültigkeit der einzelnen Referenzformel p_u zu untersuchen. Allerdings wird bei der logischen Reformulierung des praktischen Entscheidungsproblems das präsupponierte Hintergrundwissen offengelegt, das immer notwendig ist, um Gültigkeitsbehauptungen untersuchen zu können. Daher handelt es sich bei den oben thematisierten logischen Entscheidungsproblemen keineswegs um praxisferne Konstrukte. Vielmehr werden die Fragestellungen der Modellierungspraxis nur so formuliert, daß die impliziten Präsuppositionen der Modellierungspraxis expliziert werden. Hierin sieht der Verf. einen wesentlichen Beitrag, durch prädikatenlogisch fundierte Objektmodellierungen die Behandlung praktischer Modellierungsprobleme zu verbessern. Maßstäbe der Modellierungsgüte sind dabei die früher gerechtfertigten Postulate der möglichst weitgehenden Prämissenoffenlegung und der kontrollierten Explizitheit.

Es wurde aufgezeigt, daß jedes praktische Entscheidungsproblem, bei dem es gilt, die behauptete Gültigkeit einer einzelnen Formel p_u zu untersuchen, durch die oben eingeführten logischen Entscheidungsprobleme "$FM_q \models p_u$" korrekt - und sogar verbessert - wiedergegeben wird. Daher ist das Phänomen der prädikatenlogischen Semi-Entscheidbarkeit der Prädikatenlogik 1. Stufe auch für alle praktischen Entscheidungsprobleme relevant, in denen "nur" die Gültigkeit einzelner Formeln erörtert wird.

Dennoch könnte auf dem Einwand insistiert werden, in logischen Entscheidungsproblemen werde die *Allgemein*gültigkeit von Subjugatformeln "$FM_q \rightarrow p_u$" untersucht. Dies involviere einen kategorischen Geltungsanspruch, der sich nicht mit praktischen Entscheidungsproblemen vereinbaren lasse, in denen nur die Gültigkeit einzelner Formeln p_u erörtert würde. Selbst wenn das Hintergrundwissen durch eine Formelmenge FM_q berücksichtigt werde, so involviere dies immer noch nicht die Beschäftigung mit allgemeingültigen Formeln.

Aber auch dieser Einwand läßt sich nicht aufrechterhalten. Denn er übersieht, daß die Behauptung der Allgemeingültigkeit der Subjugatformel "$FM_q \rightarrow p_u$" logisch gleichwertig ist mit der Behauptung "$FM_q \models p_u$"[93]. Die letztgenannte Behauptung liegt aber - wie zuvor dargelegt wurde - auch jedem praktischen Entscheidungsproblem zugrunde, in dem anscheinend "nur" die Gültigkeit einer einzelnen Referenzformel p_u behauptet wird. Die Allgemeingültigkeit der Subjugatformel "$FM_q \rightarrow p_u$" geht über die zu untersuchende Gültigkeit der Formel p_u nur in dem Ausmaß hinaus, daß das präsupponierte Hintergrundwissen der Formelmenge FM_q explizit berücksichtigt wird. Daher involviert jede Gültigkeitsbehauptung zwingend eine korrespondierende Allgemeingültigkeitsbehauptung, sobald der Aspekt des Hintergrundwissens einbezogen wird[94].

Die Allgemeingültigkeitsbehauptung "$\models (FM_q \rightarrow p_u)$" besteht allerdings keineswegs darin, die zu untersuchende Formel p_u oder das vorausgesetzte Hintergrundwissen der Formelmenge FM_q in kategorischer Weise als allgemeingültig zu unterstellen. Statt dessen wird nur behauptet: In allen Fällen, in denen die Formelmenge FM_q des Hintergrundwissens gültig ist, muß auch die betrachtete Formel p_u gültig sein[95]. Die Allgemeingültigkeitsbehauptung "$\models (FM_q \rightarrow p_u)$", die als angeblich *kategorische* Behauptung stigmatisiert wurde, schrumpft somit auf die Qualität einer *hypothetischen* Gültigkeitsbehauptung zusammen. Sie drückt lediglich aus: *Falls* die Gültigkeit der Formelmenge FM_q aus dem Hintergrundwissen akzeptiert wird, dann muß - so wird behauptet - auch die Referenzformel p_u gültig sein. *Ob* die Formelmenge FM_q des Hintergrundwissens tatsächlich gültig ist oder als gültig akzeptiert wird, darüber sagt die Allgemeingültigkeitsbehauptung "$\models (FM_q \rightarrow p_u)$" nichts aus. Damit besitzt die Allgemeingültigkeitsbehauptung keineswegs kategorischen Charakter. Statt dessen ist sie bemerkenswert vorsichtig formuliert. Denn die behauptete Gültigkeit der Referenzformel p_u wird stets an die Akzeptanz des jeweils vorausgesetzten und offengelegten Hintergrundwissens gebunden. Eine solche Verankerung von Gültigkeitsbehauptungen in gemeinsam akzeptiertem Hintergrundwissen muß aber bei jedem praktischen Entscheidungsproblem anerkannt werden, sofern auf dogmatische Gültigkeitssetzungen verzichtet wird. Damit sind die o.a. Einwendungen gegen die praktische Relevanz der prädikatenlogischen Semi-Entscheidbarkeit widerlegt.

Die Bedeutung der Semi-Entscheidbarkeit für beliebige praktische Entscheidungsprobleme läßt sich wie folgt zusammenfassen. Ausgangspunkt ist die Behauptung, auf ein prädikatenlogisch repräsentiertes Modellierungsobjekt treffe ein Sachverhalt zu. Dieser Sachverhalt wird als eine Formel p_u ausgedrückt, dessen Gültigkeit behauptet wird. Die Gültigkeitsbehauptung erfolgt aber nicht dogmatisch, sondern vor dem Hintergrund des Wissens, das im prädikatenlogischen Objektmodell abgebildet wird. Dieses Hintergrundwissen bildet die behauptungsstützende Formelmenge FM_q. Die ursprüngliche Sachverhaltsbehauptung bedeutet also, die Allgemeingültigkeit der Subjugatsformel "$FM_q \rightarrow p_u$" zu behaupten. Für das zugehörige, zunächst semantisch formulierte Entscheidungsproblem "$\models (FM_q \rightarrow p_u)$?" lassen sich grundsätzlich drei Fälle unterscheiden:

- ❑ *Falls* die zu untersuchende Subjugatsformel tatsächlich allgemeingültig ist, so läßt sich dies mit der Hilfe jedes vollständigen prädikatenlogischen Inferenzkonzepts als Beweisbarkeit derselben Subjugatsformel in endlicher Zeit nachweisen.
- ❑ Wenn mittels eines Inferenzkonzepts eine prädikatenlogische Subjugatsformel als beweisbar erkannt worden ist und dieses Inferenzkonzept korrekt ist, dann muß dieselbe Subjugatsformel auch allgemeingültig sein.
- ❑ *Falls* die zu untersuchende Subjugatsformel tatsächlich nicht allgemeingültig ist, dann existiert grundsätzlich kein prädikatenlogisches Inferenzkonzept, das garantieren könnte, die Allgemeingültigkeitsbehauptung in endlicher Zeit zurückzuweisen. In einem solchen Fall *kann* sich der Beweisversuch in einer nicht-terminierenden Endlosschleife verfangen.

Ein naives Vertrauen in das Erkenntnispotential der Prädikatenlogik liegt vor, sofern nur die beiden ersten Fälle berücksichtigt werden. Dann kann aus der - zutreffenden - Existenz vollständiger und korrekter Inferenzkonzepte der Fehlschluß gezogen werden, jede Allgemeingültigkeitsbehauptung lasse sich in endlicher Zeit korrekt entscheiden. Der dritte Fall klärt jedoch darüber auf, daß selbst vollständige und korrekte Inferenzkonzepte das Terminieren von Behauptungsuntersuchungen nicht sicherzustellen vermögen. Vollständigkeit und Korrektheit gewährleisten daher keine Entscheidbarkeit, sondern nur die Semi-Entscheidbarkeit von Entscheidungsproblemen[96).

Anmerkungen zum Kapitel:

1) In Beweissystemen hat sich die Bezeichnungsweise der Formelmenge etabliert. Strenggenommen handelt es sich aber um ein Formelsystem, weil die Elemente der Formelmenge - die Formeln - nicht isoliert nebeneinander stehen, sondern systematisch miteinander verknüpft sind: Alle Formeln der Formelmenge gelten implizit als konjunktiv verknüpft, sofern die Formelmenge mindestens zwei Elemente umfaßt. Daher wird die Formelmenge in dieser Arbeit auch synonym als Formelsystem angesprochen. Letztgenanntes wurde bereits im Kontext der konventionellen Prädikatenlogik eingeführt. Darüber hinaus wird die Formelmenge stets als ein konsistentes Formelsystem vorausgesetzt. Andernfalls könnte aus einem inkonsistenten Formelsystem jede beliebige Formel abgeleitet werden (ex falso quodlibet). Schließlich wird unterstellt, daß die Formelmenge keine logischen Antinomien enthält.

2) Die Formel p_u und die Formelmenge FM_q werden auch als problemspezifische Formel bzw. Formelmenge bezeichnet.

3) Allerdings wird fortan der Übersichtlichkeit halber unterstellt, daß die Formelmenge immer schon als Formelsystem in konjunktiver Normalform vorliegt. Dann sind alle Variablen der darin enthaltenen Formeln jeweils durch einen Allquantor gebunden. Dies bedeutet jedoch keine Einschränkung der prädikatenlogischen Ausdrucksmächtigkeit. Denn alle Formeln, deren Variablen zunächst noch durch mindestens einen Existenzquantor gebunden worden sind, lassen sich stets mittels der Skolemisierungsoperation in Formeln transformieren, die nur noch Allquantoren besitzen.

4) Bei dieser Klauselnotation kann ein Fehlschluß auftreten, wenn die implizite Allquantifizierung aller Variablen aus klauselartigen Formeln übersehen wird. Ausgangspunkt sind eine beliebig formulierte, endliche Formelmenge FM_q und eine variable Formel $p_u(...,X_k,...)$ mit $k \in \{1,...,K_u\}$. Die Formel enthält mindestens einen Allquantor, der hier die Variable X_k bindet. Beispielsweise kann es sich bei der Formel $p_u(...,X_k,...)$ um ein einstelliges atomares Prädikat $prä(X_1)$ handeln, dem ein Allquantor vorangestellt ist. Hierfür gilt mit $k=K_u=1$: $p_u(X_1) :\Leftrightarrow \forall(X_1){:}\,prä(X_1)$. Von diesem einfachen Fall wird fortan der Übersichtlichkeit halber ausgegangen.
Nun wird die Behauptung untersucht, die Allformel "$\forall(X_1){:}\,prä(X_1)$" sei eine logische Konsequenz der Formelmenge FM_q: "$FM_q \models (\forall(X_1){:}\,prä(X_1))$". Aufgrund der nachfolgend erläuterten indirekten Beweistechnik wird diese logische Konsequenzbehauptung nicht direkt untersucht. Statt dessen wird die Behauptung analysiert, das Konjugat aus der Formelmenge FM_q und der negierten Allformel sei inkonsistent. Als problematisch erweist sich die formale Darstellung dieser Konjugatformel, wenn eine Klauselnotation vorausgesetzt wird. Zwar läßt sich die Konjugatformel prima facie als "$FM_q \wedge (\neg prä(X_1))$" aufstellen. Aber genau diese Formulierung ist fehlerhaft. Denn die implizite Allquantifizierung aller Variablen aus klauselartigen Formeln bedeutet für das hier betrachtete Beispiel, daß die Konjugatformel bei vollständiger Explizierung "$FM_q \wedge(\forall(X_1){:}\,\neg prä(X_1))$" lauten müßte. Die zweite Komponente der Konjugatformel - die Allformel "$\forall(X_1){:}\,\neg prä(X_1)$" stellt keineswegs das intendierte Negat der ursprünglichen Allformel "$\forall(X_1){:}\,prä(X_1)$" dar. Das korrekte Negat wäre statt dessen die Existenzformel "$\exists(X_1){:}\,\neg prä(X_1)$". Diese Existenzformel kann aber nicht unmittelbar herangezogen werden, um eine zu untersuchende Konjugatformel "$FM_q \wedge (\exists(X_1){:}\,\neg prä(X_1))$" zu bilden. Denn die Konjugatformel stellt aufgrund ihres Existenzquantors keine Klausel dar. Statt dessen müssen auf die Konjugatformel zunächst die Skolemisierungs-Operationen angewendet werden, die schon in einer früheren Anmerkung angesprochen wurden. Sie erlauben es, durch die Einführung spezieller SKOLEM-Funktionen die Konjugatformel so zu modifizieren, daß sie keinen Existenzquantor mehr enthält. Erst die derart "skolemisierte" Konjugatformel stellt eine Klausel dar, deren Inkonsistenz näher untersucht werden kann. Vgl. dazu auch den Hinweis auf die erforderliche Formel-Skolemisierung im Zusammenhang mit dem Resolutionstheorem.
Die voranstehenden Erläuterungen zeigen, daß es durchaus möglich ist, Formeln $p_u(...,X_k,...)$ auch dann noch korrekt zu behandeln, wenn mindestens eine ihrer Variablen X_k durch einen Allquantor gebunden ist. Aber der korrekte Umgang mit solchen Formeln bereitet wegen der erforderlichen Skolemisierungs-Operationen erheblichen Aufwand. Darüber hinaus bergen sie das Risiko in sich, zu Fehlschlüssen zu führen, falls die notwendige Formel-Skolemisierung übersehen wird. Ein Beispiel dafür wird später präsentiert. Aus beiden Gründen zieht der Verf. die Einschränkung vor, derart komplikationsbeladene Formeln $p_u(...,X_k,...)$ mit mindestens einer allquantifizierten Variablen X_k aus der Untersuchung logischer Konsequenzen der Gestalt "$FM_q \models p_u$" auszuschließen.

5) Dies entspricht auch dem Ansatz der logischen Programmierung. Dort wird bei der Formulierung einer Anfrage "p_u" untersucht, ob die konstante Formel $p_u(ob_1,...,ob_{K_u})$ oder die variable Formel $p_u(...,X_k,...)$, deren Variablen ausschließlich durch Existenzquantoren gebunden sind, die logische Konsequenz einer Formelmenge FM_q darstellen. Im ersten Fall einer konstanten Formel entstehen wegen ihrer Variablenfreiheit ohnehin keine Probleme. Im zweiten Fall einer variablen Formel werden ihre Existenzquantoren durch das Negieren der Anfrageformel, das im Rahmen der indirekten Inferenztechnik erfolgt, alle Existenz- in Allquantoren transformiert. Dies wurde bereits in einer früheren Anmerkung dargelegt, um die Entsprechung zwischen existenzquantifizierten PROLOG-Anfragen und allquantifizierten Zielklauseln aus PROLOG-Programmen zu erhellen. Für die nunmehr negierte, nur noch Allquantoren enthaltende variable Formel $\neg p_u(...,X_k,...)$ verschwinden die Probleme, die an früherer Stelle skizziert wurden. Falls es gelingt, aus dem Konjugat von Formelmenge FM_q und negierter Formel $\neg p_u(...,X_k,...)$ einen Widerspruch in

Gestalt der leeren Klausel abzuleiten, ist auf indirekte Weise bewiesen, daß die fragliche, existenzquantifizierte Formel $p_u(...,X_k,...)$ tatsächlich eine logische Konsequenz der Formelmenge FM_q darstellt. Es wird dann auch davon gesprochen, daß die Anfrageformel $p_u(...,X_k,...)$ vor dem Hintergrund der Formelmenge FM_q erfüllbar sei. Allerdings bezieht sich dieses Erfüllbarkeitsurteil nur auf die Eigenschaft der variablen Formel $p_u(...,X_k,...)$, eine Existenzanfrage zu formulieren. Denn die positive Beantwortung dieser Anfrage durch den vorgenannten indirekten Beweis drückt aus: Für jede Interpretation, unter der die Formelmenge FM_q insgesamt eine gültige Komplexformel darstellt, existiert mindestens eine Variablenbelegung "X_k", welche die Anfrageformel $p_u(...,X_k,...)$ erfüllt. Es wird also nicht nur die *Erfüllbarkeit* der existenzquantifizierten Anfrageformel $p_u(...,X_k,...)$ aufgezeigt, sondern zugleich auch die *Allgemeingültigkeit* dieser Anfrageerfüllung bezüglich aller Interpretationen, unter denen die Formelmenge FM_q gültig ist. Die anschließenden Ausführungen konzentrieren sich auf den letztgenannten Allgemeingültigkeitsaspekt.

6) Dadurch wird lediglich sichergestellt, daß die Schwierigkeiten niemals eintreten können, die in einer der voranstehenden Anmerkungen dargelegt wurden. Diese Prämisse bedeutet jedoch keine echte Einschränkung der universellen Gültigkeit der nachfolgenden Ausführungen. Denn die Prämisse könnte durchaus aufgehoben werden. Dann müßten aber - wie zuvor dargelegt wurde - die aufwendigen Skolemisierungs-Operationen für alle Formeln $p_u(...,X_k,...)$ berücksichtigt werden, die mindestens eine allquantifizierte Variable X_k enthalten.

7) Der Ausdruck "$\models (FM_q \rightarrow p_u)$" wurde als modelltheoretisches oder semantisches Theorem bezeichnet. Daher kann das Entscheidungsproblem ebenso als die Aufgabe formuliert werden zu überprüfen, ob das Subjugat "$FM_q \rightarrow p_u$" ein solches Theorem darstellt.

8) Daher stellen die nachfolgend behandelten Entscheidungsprobleme spezielle Ausprägungen des allgemeinen Gültigkeitsproblems der Prädikatenlogik dar. Folglich können die dort präsentierten Resultate - insbesondere die Unentscheidbarkeit des Gültigkeitsproblems - anschließend auf die hier untersuchten Entscheidungsprobleme übertragen werden.

9) Diese Konstruktion eines Entscheidungsproblems schließt auch den konträren Fall ein, in dem die Inkonsistenz einer problemspezifischen Formel p_u in bezug auf eine Formelmenge FM_q behauptet wird. Eine solche Inkonsistenz wird in einer prädikatenlogischen Semantik dadurch präzisiert, daß das Konjugat "$FM_q \wedge p_u$" unerfüllbar ist. Um diese Unerfüllbarkeitsbehauptung zu prüfen, wird die problemspezifische Formel p_u in ihr Negat $\neg p_u$ transformiert. Da die Formelmenge FM_q stets als konsistent vorausgesetzt wird, gilt: Die problemspezifische Formel p_u ist genau dann inkonsistent bezüglich FM_q, wenn ihr Negat $\neg p_u$ eine logische Konsequenz der Formelmenge FM_q ist, d.h. wenn "$FM_q \models \neg p_u$" gilt. Es trifft auch die äquivalente Formulierung zu: Eine Inkonsistenz liegt genau dann vor, wenn die Subjugatformel "$FM_q \rightarrow (\neg p_u)$" allgemeingültig ist, d.h. falls "$\models (FM_q \rightarrow (\neg p_u))$" zutrifft. Die Subjugatformel $FM_q \rightarrow (\neg p_u)$ läßt sich hinsichtlich ihrer Allgemeingültigkeit in der gleichen Weise prüfen wie die Allgemeingültigkeit der Subjugatformel "$FM_q \rightarrow p_u$" bei der o.a. konventionellen Formulierung eines logischen Entscheidungsproblems. Folglich läßt sich der Inkonsistenznachweis immer auf einen Allgemeingültigkeitsnachweis zurückführen; q.e.d. Vgl. zu dieser Gleichwertigkeit von Allgemeingültigkeit und Inkonsistenz für logische Entscheidungsprobleme z.B. HABEL (1983), S. 122.

10) Dies liegt daran, daß die prädikatenlogischen Inferenzprozeduren in der Regel indirekte Beweistechniken darstellen. Dies gilt insbesondere auch für Prozeduren, die das kombinierte Unifizierungs- und Resolutionskonzept verwirklichen, das im Zentrum dieser Arbeit steht. Auf diese syntaktisch definierten Inferenzprozeduren, die dem beweistheoretischen Ansatz der Prädikatenlogik zugehören, wird nachfolgend zurückgekommen. Zunächst wird nur der semantische Aspekt der modelltheoretischen Variante der Prädikatenlogik beleuchtet.

11) Die hier zugrundeliegende Prädikatenlogik gehört zur Klasse der zweiwertigen Logiken.

12) Dies bedeutet zu behaupten, daß mindestens ein Paar aus einer zulässigen Interpretation I_r und einer zulässigen Variablenbelegung V_e existiert, das die Negatformel erfüllt. Die Negatformel besitzt also mindestens ein Modell (I_r,V_e).

13) Das Negat der problemspezifischen Subjugatformel "$FM_q \rightarrow p_u$" ergibt sich als syntaktische Umformulierung nach DE MORGAN:

$$\neg(FM_q \rightarrow p_u) \;\Leftrightarrow\; \neg(\neg FM_q \vee p_u) \;\Leftrightarrow\; FM_q \wedge (\neg p_u).$$

Vgl. zur DE MORGAN-Äquivalenz CARNAP (1960a), S. 31 u. 33; ESSER,H. (1977a), S. 37a, Punkt 11; WECK (1982), S. 72; BUCHER (1987), S. 108.

14) Vgl. dazu den Hinweis, daß die konkrete Ausführung von syntaktisch definierten Inferenzprozeduren wesentlich einfacher ist, als die Allgemeingültigkeit von Formeln mit den semantischen Mitteln des modelltheoretischen Ansatzes zu beweisen. Anschließend wird stets ein vollständiges und korrektes Beweissystem vorausgesetzt, das die Grundlage der jeweils angewandten Inferenzprozeduren bildet. Welche Gestalt dieses Beweissystem im einzelnen annimmt, interessiert hier jedoch nicht weiter.

15) Da jede beweisbare Formel innerhalb eines Beweissystems ein beweistheoretisches oder syntaktisches Theorem darstellt, läßt sich das zu untersuchende Entscheidungsproblem auch auf folgende Art reformulieren: Es gilt zu entscheiden, ob es sich bei der präsentierten Subjugatformel um ein solches Theorem handelt; vgl. z.B. LOVELAND (1978), S. 30; DELAHAYE (1987), S. 34. Vgl. dazu auch die komplementäre semantische Problemformulierung durch Bezugnahme auf ein modelltheoretisches Theorem.

16) Infolge Vollständigkeit ist jede allgemeingültige Formel beweisbar. Die Korrektheit der Inferenzkonzepte stellt umgekehrt sicher, daß jede syntaktisch bewiesene Formel auch allgemeingültig ist.

17) Die syntaktische Bestätigung, daß die Beweisbarkeitsbehauptung "$\vdash_{BS} (FM_q \rightarrow p_u)$" tatsächlich zutrifft, bedeutet aus semantischer Perspektive, daß die problemspezifische Subjugatformel "$FM_q \rightarrow p_u$" allgemeingültig ist. Folglich ist auch die Allgemeingültigkeitsbehauptung dieser Subjugatformel "$FM_q \rightarrow p_u$" bestätigt. Dieser Sachverhalt wird vereinfacht auch dadurch ausgedrückt, die Allgemeingültigkeitsbehauptung der Subjugatformel sei "bewiesen" worden. Strenggenommen ist der Beweisbegriff aber nicht für die semantische Sphäre von Formelgültigkeiten, sondern nur für syntaktische Beweissysteme definiert.

18) Da die Widerspruchsfreiheit der Prädikatenlogik nachgewiesen wurde, können stets widerspruchsfreie prädikatenlogische Beweissysteme aufgestellt werden. Inkonsistente Beweissysteme werden in dieser Arbeit grundsätzlich ausgeschlossen. Vgl. dazu die entsprechende Ausgrenzung inkonsistenter Argumentationsgrundlagen.

19) Vgl. dazu die Definition der Unbeweisbarkeit. Aus ihr folgt: Die Subjugatformel "$FM_q \rightarrow p_u$" ist in einem widerspruchsfreien Beweissystem BS mit dem Axiomensystem AX genau dann unbeweisbar, wenn das modifizierte Beweissystem BS* mit dem Axiomensystem $AX^* = (AX - \{FM_q \rightarrow p_u\}) \cup \{\neg(FM_q \rightarrow p_u)\}$ widerspruchsfrei ist. Wegen $(\neg(FM_q \rightarrow p_u)) \Leftrightarrow (FM_q \wedge (\neg p_u))$ gilt auch: Die Subjugatformel "$FM_q \rightarrow p_u$" ist genau dann unbeweisbar, wenn das Beweissystem BS* mit dem Axiomensystem $AX^* = (AX - \{FM_q \rightarrow p_u\}) \cup \{FM_q \wedge (\neg p_u)\}$ widerspruchsfrei ist. Aus der hier stets unterstellten Voraussetzung, daß das ursprüngliche Beweissystem BS die Subjugatformel "$FM_q \rightarrow p_u$" nicht als Axiom enthält, folgt: Die Subjugatformel "$FM_q \rightarrow p_u$" ist genau dann unbeweisbar, wenn sich aus dem erweiterten Beweissystem BS* mit dem Axiomensystem $AX^* = AX \cup \{FM_q \wedge (\neg p_u)\}$ kein Widerspruch ableiten läßt.

20) Dies entspricht in semantischer Hinsicht genau der o.a. Behauptung, daß die Konjugatformel "$FM_q \wedge (\neg p_u)$" inkonsistent sei. Daher wird auch bei der syntaktischen Betrachtung von Inferenzprozeduren die vereinfachte Sprechweise zugelassen, von einer Inkonsistenzbehauptung bezüglich der Konjugatformel "$FM_q \wedge (\neg p_u)$" zu reden. Es handelt sich dabei jedoch insofern um eine Vereinfachung, weil aus syntaktischer Sicht niemals die Konjugatformel allein, sondern stets ihre Vereinigung mit dem Axiomensystem des jeweils zugrundeliegenden Beweissystems untersucht wird.

21) Die Ableitbarkeit der inkonsistenten Formel "$\varnothing$" aus der Konjugatformel "$FM_q \wedge (\neg p_u)$" im Beweissystem BS entspricht der Möglichkeit, im modifizierten Beweissystem BS*, dessen Axiomensystem um die Konjugatformel "$FM_q \wedge (\neg p_u)$" erweitert wurde, die inkonsistente Formel zu beweisen: "$\vdash_{BS^*} \varnothing$". Die Beweisbarkeit der inkonsistenten Formel bedeutet wiederum, daß im modifizierten Beweissystem für eine beliebige Formel p_k sowohl diese Formel selbst als auch deren Negat - die Formel $\neg p_k$ - bewiesen werden kann. Diese Möglichkeit, sowohl eine Formel als auch deren Negat zu beweisen, definiert aber ein in sich widersprüchliches Beweissystem. Vgl. zu diesen Zusammenhängen die Definitionen beweistheoretischer Konstrukte.

22) In beweistheoretischen Argumentationen ist es üblich, die konjunktiv verknüpften Prämissen einer Ableitung links neben dem Ableitungsoperator "$\vdash_{BS}$" nicht als Konjugat zu notieren, sondern aufzuzählen. Dies wurde bereits erwähnt. Dieser vorherrschenden Notationsweise folgt die o.a. Darstellung der Ableitbarkeitsbehauptung.

23) Die syntaktische Formulierungsweise hat den Vorzug, daß die behauptete Ableitbarkeit der inkonsistenten Formel "$\varnothing$" rein formalsprachlich ausgedrückt werden kann. Die entsprechende semantisch formulierte Inkonsistenzbehauptung erfordert dagegen eine semi-formalsprachliche Darstellungsweise, die das natürlichsprachliche Attribut "inkonsistent" enthält. Denn im modelltheoretischen Konzept der Prädikatenlogik ist zwar eine formalsprachliche Notation für die Allgemeingültigkeit von Formeln vorgesehen, nicht aber für deren Inkonsistenz.

24) Die zu untersuchende Behauptung, die inkonsistente Formel "$\varnothing$" lasse sich aus der Konjugatformel "$FM_q \wedge (\neg p_u)$" ableiten, wird fortan auch kurz als Inkonsistenzbehauptung für die Konjugatformel "$FM_q \wedge (\neg p_u)$" angesprochen.

25) Der syntaktischen Bestätigung der Ableitbarkeitsbehauptung "$FM_q, \neg p_u \vdash_{BS} \varnothing$" entspricht aus semantischem Blickwinkel der Sachverhalt, daß die Konjugatformel "$FM_q \wedge (\neg p_u)$" durch überhaupt kein Paar (I_r, V_c) aus einer Interpretation I_r und einer Variablenbelegung V_c erfüllt wird. Diese Inkonsistenz der Formel "$FM_q \wedge (\neg p_u)$", welche das Negat der problemspezifischen Subjugatformel "$FM_q \rightarrow p_u$" darstellt, bestätigt die ursprünglich behauptete Allgemeingültigkeit dieser Subjugatformel. Daher wird durch die Bestätigung der Ableitbarkeitsbehauptung "$FM_q, \neg p_u \vdash_{BS} \varnothing$" zugleich die Allgemeingültigkeitsbehauptung der Subjugatformel "$FM_q \rightarrow p_u$" bestätigt. Da die Bestätigung der Ableitbarkeitsbehauptung zugleich auf indirekte Weise die zugrundeliegende Beweisbarkeits-

behauptung für die Subjugatformel "$FM_q \rightarrow p_u$" bestätigt, wird ebenso davon gesprochen, die Allgemeingültigkeitsbehauptung der Subjugatformel "$FM_q \rightarrow p_u$" sei "bewiesen" worden.

26) Die syntaktische Zurückweisung der Ableitbarkeitsbehauptung "$FM_q, \neg p_u \vdash_{BS} \varnothing$" bedeutet aus semantischer Perspektive, daß für die Konjugatformel "$FM_q \wedge (\neg p_u)$" mindestens ein Modell (I_r, V_c) aus einer Interpretation I_r und einer Variablenbelegung V_c existiert. Diese Erfüllbarkeit der Formel "$FM_q \wedge (\neg p_u)$", die das Negat der problemspezifischen Subjugatformel "$FM_q \rightarrow p_u$" ist, steht im Widerspruch zur behaupteten Allgemeingültigkeit der Subjugatformel. Daher wird durch die Widerlegung der Ableitbarkeitsbehauptung "$FM_q, \neg p_u \vdash_{BS} \varnothing$" zugleich die Allgemeingültigkeitsbehauptung der Subjugatformel "$FM_q \rightarrow p_u$" zurückgewiesen. Für diesen Sachverhalt wird auch die vereinfachte Formulierung zugelassen, die Allgemeingültigkeitsbehauptung der Subjugatformel sei widerlegt worden. Strenggenommen ist der Widerlegungsbegriff aber nicht für die semantische Sphäre von Formelgültigkeiten, sondern nur für syntaktische Beweissysteme definiert.

27) Dies hätte sich auch mit der Inferenzregel des "modus ponens" unmittelbar nachweisen lassen. Aus dem Subjugat "$prä_1(X) \rightarrow prä_2(X)$" und der konstanten atomaren Formel "$prä_1(c)$" folgt unmittelbar die konstante atomare Formel $prä_2(c)$. Dies interessiert hier jedoch nicht weiter.

28) Die Wohldefiniertheit einer Problemlösung zu definieren, stellt ein schwieriges metasprachliches Unterfangen dar. Seine Ausführung wird hier nur grob skizziert, da das Erkenntnisziel dieser Arbeit nicht in einer vollständig ausgearbeiteten Terminologie für Entscheidungsprobleme liegt.
Das Konzept der Wohldefiniertheit setzt die früheren Ausführungen zum MÜNCHHAUSEN-Trilemma voraus. Daraus folgt, daß kein Begriff in einem unbedingten Sinne wohldefiniert sein kann. Statt dessen gilt die Wohldefiniertheit eines Begriffs immer nur unter der einschränkenden Bedingung, daß bereits eine Sammlung anderer Begriffe als wohldefiniert akzeptiert werden. Sie gehören zum "Hintergrundwissen", das in jeder Argumentation immer schon präsupponiert wird. Hinzu kommt eine Gruppe von Begriffsverknüpfungen, deren Zulässigkeit ebenfalls akzeptiert sein muß. Zwecks klarer Unterscheidung von den früher erwähnten undefinierten Basisbegriffen werden die hier vorausgesetzten wohldefinierten Begriffe als Grundbegriffe bezeichnet. Ein Grundbegriff heißt genau dann wohldefiniert, wenn seine Extension eine entscheidbare Menge ist. Andernfalls heißt der Grundbegriff entweder schlechtdefiniert, wenn er in irgendeiner anderen Weise definiert ist, oder aber undefiniert, wenn er überhaupt nicht definiert ist. Undefinierte Grund- fallen mit den zuvor erwähnten undefinierten Basisbegriffen zusammen.
Eine Begriffsextension ist die Menge aller Entitäten, die unter einen Begriff fallen. Diese Menge muß keineswegs endlich sein. Wohldefiniertheit eines Grundbegriffs setzt also nicht voraus, daß es möglich sein müßte, die Menge aller Entitäten, die den Grundbegriff erfüllen, durch endliche Auflistung ihrer Elemente vollständig zu explizieren. Statt dessen wird nur die schwächere Anforderung erhoben, daß sich für jede präsentierte Entität eindeutig entscheiden läßt, ob sie zur Extension des Grundbegriffs gehört oder nicht. Hierfür ist ein endliches Entscheidungsverfahren notwendig. Auch dieses Entscheidungsverfahren gehört zum präsupponierten Hintergrundwissen.
Mit Hilfe der voranstehenden Festlegungen läßt sich die o.a. Wohldefiniertheit der positiven und negativen Lösungen von Entscheidungsproblemen verdeutlichen. Es werden vorausgesetzt als wohldefinierte Grundbegriffe mit entscheidbaren Begriffsextensionen: die prädikatenlogische Formel, das Formelsystem, die Interpretation eines Formelsystems, die Variablenbelegung für ein Formelsystem und die Gültigkeit einer Formel in bezug auf das Paar aus einer Interpretation und einer Variablenbelegung. Falls Zweifel daran bestehen, wird auf die früheren Definitionen dieser Begriffe verwiesen, die anläßlich der prädikatenlogischen Grundlegung dieser Arbeit erfolgten. Dadurch wird die Voraussetzung wohldefinierter Grundbegriffe aber nur eine Stufe weiter hinausgeschoben. Des weiteren werden die prädikatenlogischen Operatoren als zulässige Begriffsverknüpfungen vorausgesetzt. Dann läßt sich zeigen: Aus der Wohldefiniertheit der Gültigkeit einer Formel bezüglich einer Interpretation und einer Variablenbelegung, aus der Wohldefiniertheit eines Formelsystems und aus den Definitionen der prädikatenlogischen Operatoren, die am Aufbau eines Formelsystems teilhaben, folgt die Wohldefiniertheit der Gültigkeit eines Formelsystems bezüglich einer Interpretation und einer Variablenbelegung. Wegen der unterstellten Akzeptanz prädikatenlogischer Operatoren ist dann auch die problemspezifische Subjugatformel eines Entscheidungsproblems wohldefiniert. Wenn die Gesamtheit aller zulässigen prädikatenlogischen Interpretationen und Variablenbelegungen feststeht, dann ist auch die Gültigkeit der Subjugatformel bezüglich aller zulässigen Paare aus einer Interpretation und einer Variablenbelegung wohldefiniert. Also ist auch die Allgemeingültigkeit der problemspezifischen Subjugatformel wohldefiniert. Dann sind die positive und die negative Lösung eines Entscheidungsproblems dadurch wohldefiniert, daß die Definition der Allgemeingültigkeit seiner problemspezifischen Subjugatformel erfüllt bzw. verletzt wird.
Die voranstehende Charakterisierung der Wohldefiniertheit eines Begriffs verhält sich kohärent zur früher gerechtfertigten Bevorzugung extensionaler Konzepte. Denn die Wohldefiniertheit wird nur an die Begriffsextension, nicht aber an alternative intensionale Begriffsvorstellungen geknüpft. Darüber hinaus wird das Prinzip kontrollierter Explizitheit unterstrichen, indem auf die vollständige Explizierung der Begriffsextension verzichtet wird. Statt dessen reicht es aus, ein Entscheidungsverfahren offenzulegen, mit dessen Hilfe sich die Extensionszugehörigkeit jeder Entität überprüfen läßt. Diese indirekte Festlegung der Begriffsextension durch ein begriffsspezifisches Entscheidungsverfahren ist nicht nur kompakter, sondern aus leistungsfähiger als eine vollständige Explizierung der Begriffsextension. Denn potentiell unendliche Extensionen lassen sich nur mit Hilfe eines solchen Entscheidungsverfahrens beherrschen (sofern die Extensionen rekursiv entscheidbare Mengen darstellen).

Diese Leistungsausweitung ist hier wesentlich. Denn die oben exemplarisch vorausgesetzte Wohldefiniertheit des Begriffs "prädikatenlogische Formel" läßt sich nicht auf eine vollständige Auflistung der Begriffsextension zurückführen. Dies liegt an dem induktiven Charakter des früher vorgestellten prädikatenlogischen Kalküls. Seine Zeichen und Formierungsregeln gestatten es, eine potentiell unendliche Vielfalt prädikatenlogischer Formeln zu erzeugen. Dennoch kann anhand der Kalküldefinition für jede präsentierte Entität entschieden werden, ob es sich um eine prädikatenlogische Formel im Sinne des Kalküls - eine wohlgeformte Formel - handelt. Die Korrespondenz zwischen den Begriffsbildungen der *Wohl*definiertheit und der *wohl*geformten Formeln ist hier keineswegs zufällig, sondern Ausdruck des Bemühens terminologischer Kohärenz.
Abschließend wird versucht, einem Mißverständnis vorzubeugen. Es könnte der Einwand erhoben werden, die später aufgezeigte Unentscheidbarkeit mancher Entscheidungsprobleme widerspreche der Wohldefiniertheit der Problemlösungen. Denn die Unentscheidbarkeit beruhe auf dem Fehlen eines Entscheidungsalgorithmus für das Feststellen positiver und negativer Problemlösungen. Da diese Problemlösungen qua Voraussetzung wohldefiniert sind, müsse es doch möglich sein, aus den zugrundeliegenden Entscheidungsverfahren für die Begriffsextensionen ein Verfahren abzuleiten, das auch das Entscheidungsproblem löse. Folglich könne es überhaupt keine unentscheidbaren Probleme geben. Diese Vorhaltung läßt sich jedoch nicht aufrechterhalten. Sie unterliegt einer unzulässigen Vermengung zweier unterschiedlicher Argumentationsebenen: Auf der Ebene eines Entscheidungsproblems kann tatsächlich der Fall eintreten, daß hierfür kein Entscheidungsalgorithmus existiert. Dies wird später dargelegt. Auf einer tieferen Ebene wird das Entscheidungsproblem aus wohldefinierten Grundbegriffen und zulässigen Begriffsverknüpfungen sukzessiv aufgebaut. Nur für jene wohldefinierten Grundbegriffe werden Entscheidungsverfahren vorausgesetzt. Der oben skizzierte Einwand beruht auf der Präsupposition, daß sich die Entscheidbarkeit der Extensionen wohldefinierter Grundbegriffe *notwendig* auf jedes Entscheidungsproblem übertragen müsse, das mit ihrer Hilfe definiert wird. Diese Unterstellung erweist sich jedoch als Irrtum. Denn eine Folge der prädikatenlogischen Unentscheidbarkeit besteht darin, daß die Entscheidbarkeit von wohldefinierten Begriffen verloren gehen kann, wenn sie zu problemformulierenden Begriffskomplexen verknüpft werden.
Bemerkenswert ist allerdings die rekursive Struktur des Unterfangens, Eigenschaften von Entscheidungsproblemen zu definieren. Sie klingt in der voranstehenden Erläuterung bereits als hierarchisch verschachtelte Bezugnahme auf Entscheidungsverfahren an: Auf der oberen Definitionsebene werden Entscheidungsalgorithmen betrachtet, von deren Existenz die Problemeigenschaft der Unentscheidbarkeit abhängt. Dabei werden lösbare Entscheidungsprobleme vorausgesetzt, die sich durch die Wohldefiniertheit ihrer Lösungen auszeichnen. Auf einer tieferen Definitionsebene wird die Wohldefiniertheit von Begriffen ihrerseits auf Entscheidungsverfahren zurückgeführt. Sie dienen dazu, um die Zugehörigkeit von Entitäten zur Begriffsextension zu überprüfen. Entscheidungsverfahren für die Lösung eines Entscheidungsproblems setzen also andere Entscheidungsverfahren voraus, die bereits sicherstellen, daß das vorliegende Entscheidungsproblem überhaupt lösbar ist. Diese zweistufige Hierarchie von Entscheidungsverfahren läßt sich beliebig vertiefen, indem die Extensionszugehörigkeit von Entitäten ihrerseits als ein eigenständiges Entscheidungsproblem konzeptualisiert wird. Damit dieses neue Entscheidungsproblem lösbar ist, müssen wiederum tieferliegende Entscheidungsverfahren dafür Sorge tragen, daß die Lösungen des Zugehörigkeitsproblems wohldefiniert sind usw. ad infinitum.

29) Die Unlösbarkeit eines Entscheidungsproblems kommt diesem "an sich", d.h. ohne Bezugnahme auf die Existenz eines Entscheidungsalgorithmus zu. Bei der nachfolgend erklärten Unentscheidbarkeit eines Entscheidungsproblems liegt dagegen ein "an sich" lösbares Entscheidungsproblem vor, für das jedoch ein entsprechender Entscheidungsalgorithmus fehlt.
Einen abweichenden Begriff der Unlösbarkeit von Problemen vertritt z.B. BACHEM (1980), S. 812f. u. 840f. Er setzt - unausgesprochen - schon immer die Wohldefiniertheit der Problemlösungen voraus. Darüber hinaus betrachtet er nicht Entscheidungsprobleme, die eine Antwort "ja" oder "nein" erfordern, sondern Probleme, bei denen es gilt, die Elemente einer Lösungsmenge zu ermitteln. Auf dieser impliziten Grundlage definiert er ein Problem als unlösbar, wenn seine wohldefinierte Lösungsmenge bekannt, aber leer ist oder wenn sich seine wohldefinierte Lösungsmenge überhaupt nicht ermitteln läßt. Beide Fälle werden in der Terminologie dieser Arbeit jedoch zu den lösbaren Problemen gerechnet. Leere Lösungsmengen gehören zu den entscheidbaren Problemen. Unbekannte Lösungsmengen können sowohl auf - vorläufig - ungelösten als auch auf gelösten, aber unentscheidbaren Entscheidungsproblemen beruhen. Die Problemeigenschaften der (Un-)Entscheidbarkeit und (Un-)Gelöstheit werden im folgenden näher erklärt.

30) Gleichwohl lassen sich unlösbare Entscheidungsprobleme vorstellen. Dazu brauchen die Probleme lediglich so formuliert zu werden, daß ihre Lösungen schlechtdefiniert sind. Dies ist der Fall, wenn die Problemformulierung auf Grundbegriffen aufbaut, die nicht als wohldefiniert akzeptiert werden. Ein Beispiel dafür wurde schon früher im argumentationstheoretischen Kontext angedeutet: Das Entscheidungsproblem, ob eine Rechtfertigung "plausibel" sei, erweist sich als unlösbar. Denn es besitzt keine wohldefinierte positive und negative Lösung. Die Schlechtdefiniertheit der Problemlösungen wird durch den Plausibilitätsbegriff verursacht, der weder selbst wohldefiniert ist noch auf wohldefinierte Grundbegriffe zurückgeführt werden kann.

31) Daher kann fortan auf das Attribut "lösbar" zur Kennzeichnung von Entscheidungsproblemen verzichtet werden.

32) Ein anschauliches Beispiel für Entscheidungsprobleme mit wohldefinierten, aber dennoch unbekannten Lösungen läßt sich in Anlehnung an BACHEM (1980), S. 815f., aufstellen: Es gilt für eine beliebige natürliche Zahl "n" zu entscheiden, ob eine Funktion "fgo" den Wert "1" annimmt. Dabei ist die Funktion definiert durch die Abbildungsvorschrift "fgo(n) = 1, falls Gott existiert" und "fgo(n) = 0, falls Gott nicht existiert". Die Lösungen dieses Entscheidungsproblems sind wohldefinierte Funktionswerte. Denn das Entscheidungsproblem ist genau dann positiv (negativ) gelöst, wenn "fgo(n) = 1" (nicht) gilt. Die involvierten Grundbegriffe der natürlichen Zahl, der Funktion und der Gleichheit zwischen einem Funktionswert und einer natürlichen Zahl sind allesamt wohldefiniert. Dennoch besteht keine Aussicht darauf, jemals entscheiden zu können, ob der Funktionswert "1" zutrifft oder nicht. Dies gilt zumindest so lange, wie das erforderliche Wissen über die "Existenz Gottes" nicht zur Verfügung steht. Eine ähnlich konstruierte Funktion, die sich allerdings auf FERMAT's "Theorem" bezieht, findet sich bei MACHOVER (1983), S. 9.
Ein anderes, mathematisch "seriöseres" Entscheidungsproblem mit wohldefinierten, aber dennoch unbekannten Lösungen (für $n \geq 5$) beruht auf dem Automatentyp des "fleißigen Bibers" (busy beaver). Ein "fleißiger Biber" ist ein TURING-Automat - im folgenden kurz: ein bb-Automat (für: busy beaver-Automat) - mit n internen Zuständen (oder Anweisungen) und $n \in \mathcal{N}_+$. Er wird auf ein Band als linearen Informationsspeicher angesetzt, das zunächst nur Nullen enthält, nach endlich vielen Berechnungsschritten garantiert anhält und dann mindestens so viele Einsen auf das Band geschrieben hat wie jeder andere TURING-Automat mit derselben Zustandsanzahl (Anweisungsanzahl). Die RADO-Funktion gibt die Anzahl von Einsen an, die von einem solchen bb-Automaten mit n Zuständen (Anweisungen) maximal erzeugt werden können. Es konnte bewiesen werden, daß die RADO-Funktion für *beliebige* natürliche Zahlen "n" prinzipiell nicht berechnet werden kann. Es handelt sich um eine *nicht*-rekursive Funktion, deren Funktionswerte "$\Sigma(n)$" - für ein jeweils hinreichend groß gewähltes "n" - schneller als jede überhaupt (TURING-)berechenbare Funktion anwachsen. Bis heute sind die Werte der RADO-Funktion nur für $n \in \{1,2,3,4\}$ bekannt. Folglich sind die Lösungen für Entscheidungsprobleme der Art "$\Sigma(n) = k_n$?" für jeweils *fest vorgegebene* natürliche Zahlen "n" und "k_n" zwar wohldefiniert: nämlich als "ja" ("nein"), falls "$\Sigma(n) = k_n$" tatsächlich (nicht) zutrifft. Aber sie sind dennoch für $n \geq 5$ unbekannt, weil die Werte "$\Sigma(n)$" der RADO-Funktion unbekannt sind. (Dabei wird davon abgesehen, daß für n=5 und n=6 bereits untere bzw. obere Schranken der RADO-Funktion ermittelt werden konnten. Daher lassen sich die vorgenannten Entscheidungsprobleme für solche "k_n", die unter bzw. über diesen Schranken liegen, bereits mit einem "nein" lösen.) Entscheidungsprobleme für bb-Automaten in der Form "$\Sigma(n) = k_n$?" sind also lösbar für festliegende Problemparameter "n" und "k_n", ohne daß ihre Lösungen für beliebige "n" mit $n \geq 5$ heute schon bekannt wären. Näheres zur unberechenbaren RADO-Funktion und zu "fleißigen Bibern" findet sich bei RADO (1962), S. 878ff.; BRAUER (1968), S. 45ff.; BOOLOS (1980), S. 27 u. 34ff.; LUDEWIG (1983), S. 4ff.; DEWDNEY (1984), S. 8ff.; HOPCROFT (1984), S. 45ff.; ZELEWSKI (1986a), S. 37 u. 89ff.; BRAUER (1990a), S. 62 u. 64ff.

33) Ein Algorithmus wird in dieser Arbeit in einer groben Annäherung als ein Verfahren verstanden, für das gilt: Das Verfahren ist für eine nicht-leere Menge zulässiger Eingabeinformationen (Inputs) definiert. Falls das Verfahren auf eine zulässige Eingabeinformation angewendet wird, erzeugt es mit endlichem Ressourcenverbrauch eine eindeutig bestimmte Ausgabeinformation (Output). Diejenige Funktion, die jeden zulässigen Input auf den jeweils zugehörigen Output abbildet, heißt die Spezifikation des Algorithmus. Das algorithmusspezifische Verfahren wird auch als Prozedur angesprochen. Vgl. zu verfeinerten Algorithmusdefinitionen, die für das Verständnis dieser Arbeit jedoch nicht erforderlich sind, BACHEM (1980), S. 814f.; ZELEWSKI (1986a), S. 280ff., und die dort angeführten Quellen.

34) Strenggenommen muß von endlichem Ressourceneinsatz geredet werden (vgl. die voranstehende Anmerkung). Denn unterschiedliche Ressourcen - von deren die Zeit für die Ausführung von Inferenzprozeduren nur eine ist - können bei der Bewältigung von Entscheidungsproblemen gegenseitig substituiert werden. Vor allem kommen als weitere Ressourcenmaßstäbe auch das Speichervolumen eines informationsverarbeitenden Automaten und die Anzahl seiner unabhängig arbeitenden Prozessoren in Frage. Auf solche differenzierten Ressourcenbetrachtungen wird hier der Übersichtlichkeit halber verzichtet; es wird vereinfachend nur die Ressource "Zeit" betrachtet. Vgl. dagegen zu anspruchsvolleren Betrachtungen der speichervolumenbezogenen Komplexität von Algorithmen MEYER,A. (1972), S. 125ff.; COOK,S. (1973), S. 29ff.; BÖHLING (1974), S. 204ff.; HOPCROFT (1974), S. 624; CARDOZA (1976), S. 52; JONES,N. (1977), S. 280; PAUL,W. (1978), S. 93; GAREY (1979), S. 170ff.; LENSTRA (1979), S. 124f.; LENSTRA (1982), S. 206f.; COOK,S. (1983), S. 404; GOLDBERG (1984a), S. 46; ZELEWSKI (1989a), S. 20, 28f., 60 u. 91ff.

35) Es wird hier bewußt das einschließliche "oder" verwendet, das der formalsprachlichen Adjunktion entspricht. Dadurch wird zugelassen, daß mehrfache Anwendungen des gleichen Entscheidungsalgorithmus durchaus zu dem Ergebnis führen dürfen, das untersuchte Entscheidungsproblem besitze sowohl eine positive als auch eine negative Lösung. Dann ist das untersuchte Entscheidungsproblem zwar lösbar, weil seine Lösungen wohldefiniert sind. Aber das Problem ist in sich widersprüchlich formuliert (inkonsistent). Aus einer inkonsistenten Problemformulierung läßt sich jedes Ergebnis ableiten, also auch sowohl dessen positive als auch dessen negative Lösung.

36) Dabei liegt eine generische Ausdrucksweise zugrunde. Sie verallgemeinert die algorithmusspezifische Klasse gleichartiger Entscheidungsprobleme zur Vorstellung eines - nunmehr abstrakter formulierten - Entscheidungs-

problems. Alle Mitglieder der algorithmusspezifischen Problemklasse heißen dann Ausprägungen dieses Entscheidungsproblems. In dieser Arbeit werden beide Ausdrucksweisen zugelassen, sofern sie nicht miteinander vermengt werden. Auf einer abstrakteren Formulierungsebene wird von einem Entscheidungsproblem und der Klasse seiner Ausprägungen gesprochen. Die gleichen Sachverhalte lassen sich auf einer konkreteren Formulierungsebene äquivalent in der Weise ausdrücken, daß gleichartige Entscheidungsprobleme zu einer gemeinsamen Problemklasse zusammengefaßt werden. Im ersten Fall handelt es sich um ein *abstraktes* Entscheidungsproblem mit mehreren konkreten Ausprägungen. Im zweiten Fall liegen dagegen mehrere *konkrete* Entscheidungsprobleme vor, die zu einer Klasse gleichartiger Probleme abstrahiert werden. Daher stellt ein abstrakt formuliertes Entscheidungsproblem eine Klasse von ähnlichen konkreten Entscheidungsproblemen dar. Jedes konkrete Entscheidungsproblem ist eine Ausprägung des gemeinsam zugrundeliegenden abstrakten Entscheidungsproblems. Vgl. dazu auch ZELEWSKI (1989a), S. 11.
Beispielsweise handelt es sich bei dem oben eingeführten logischen Entscheidungsproblem um ein abstraktes Problem. Denn die Formulierung seiner problemspezifischen Subjugatformel läßt eine Vielzahl von verschiedenen Ausprägungen zu. Sie unterscheiden sich dadurch voneinander, daß sie in jeweils anderer Weise die Formelbestandteile "Formelsystem FM_q" und "Formel p_u" konkretisieren. Alle Ausprägungen erweisen sich jedoch insofern als gleichartig, als sie die gleiche Struktur der problemspezifischen Subjugatformel aufweisen. Sie sind daher zulässige Eingabeinformationen für einen Entscheidungsalgorithmus, der sich auf jede Subjugatformel der Gestalt "$FM_q \rightarrow p_u$" anwenden läßt.

37) Ein Entscheidungsproblem des Petrinetz-Konzepts, das vorerst noch ungelöst ist, wird an späterer Stelle vorgestellt. Es handelt sich um das Lebendigkeitsproblem für Netze, deren Kantengewichte nicht auf das Einheitsgewicht eingeschränkt sind.

38) Daher müssen zwei Lösungsbegriffe, die verschiedenen Sprachebenen angehören, deutlich unterschieden werden: Die Lösung eines Entscheidungsproblems betrifft nur die *Existenz* eines problemspezifischen Entscheidungsalgorithmus. Das Wissen, daß entweder mindestens ein oder aber kein solcher Algorithmus existiert, "löst" nur ein metasprachliches *Existenzproblem.* Die *Anwendung* eines solchen Entscheidungsalgorithmus auf ein konkretes Entscheidungsproblem "löst" dagegen durch seine Ausgabeinformation das jeweils untersuchte, nunmehr aber objektsprachlich formulierte *Entscheidungsproblem.*

39) Vgl. dazu die kritische Erörterung nonkonstruktiver Beweistechniken.

40) Es wird jedoch später gezeigt, daß die Inferenzprozeduren einen abgeschwächten Begriff prädikatenlogischer Entscheidungsalgorithmen erfüllen (können). Dies wird zur Herausbildung des Begriffs semi-entscheidbarer Probleme führen.

41) Unter der Prädikatenlogik wird hier - wie überall in dieser Arbeit - stets die erweiterte Prädikatenlogik verstanden, die auch alle arithmetischen Prädikate umfaßt. Darauf wurde bereits explizit hingewiesen, insbesondere in bezug auf das arithmetische Identitätsprädikat. Daher gelten die nachfolgenden Ausführungen für die Unentscheidbarkeit der arithmetisch erweiterten Prädikatenlogik. Vgl. dazu auch den klarstellenden Hinweis zur prädikatenlogischen Unentscheidbarkeit.

42) Jedes prädikatenlogische Entscheidungsproblem läßt sich durch eine endliche Formel in konjunktiver Normalform ausdrücken. Denn oben wurden Entscheidungsprobleme als Allgemeingültigkeitsbehauptungen für Subjugate formuliert, die jeweils aus einer endlichen Formelmenge als Prämisse und einer einzelnen Formel als Konklusion bestehen. Die Transformation einer solchen Subjugatformel in ihre behauptungsäquivalente konjunktive Normalform liefert das allgemeine Schema, das in einer früheren Anmerkung schon vorgestellt wurde. Jede Formel in konjunktiver Normalform kann als eine endliche Klauselmenge notiert werden. Sofern in dieser Klauselmenge nur HORN-Klauseln vorkommen, läßt sie sich unmittelbar als ein Programm der Programmiersprache PROLOG darstellen. Von diesem Fall wird fortan ausgegangen, wenn PROLOG-Programme als Repräsentationen von prädikatenlogischen Entscheidungsproblemen thematisiert werden. Später wird gezeigt, wie sich mit der Hilfe von Netzmodellen auch beliebige Allgemeine Klauseln, die keine HORN-Klauseln sind, dennoch in ein PROLOG-Programm aus HORN-Klauseln übersetzen lassen. Daher gilt die nachfolgende Argumentation allgemein für *alle* prädikatenlogischen Entscheidungsprobleme. Denn sie lassen sich immer - unmittelbar oder mittelbar - durch PROLOG-Programme ausdrücken.

43) Auf den ersten Blick mag die Behauptung unentscheidbarer Entscheidungsprobleme paradox erscheinen. Tatsächlich läßt sich jedoch in der Prädikatenlogik eine Vielzahl unentscheidbarer Entscheidungsprobleme formulieren. Ein herausragender Repräsentant dieser Problemklasse ist das prädikatenlogische Gültigkeitsproblem, das bereits behandelt wurde. Auch die Unentscheidbarkeit von HILBERT's 10. Problem wurde schon angesprochen. Des weiteren erweist sich das Entscheidungsproblem der Form "$\Sigma(n)=k_n$?" für Automaten vom Typ des "fleißigen Bibers", die in einer früheren Anmerkung vorgestellt wurden, für *beliebige* natürliche Zahlen "n" und "k" als unentscheidbar. Es wurde bereits dargelegt, daß die RADO-Funktion "Σ" für diese Automaten und beliebige natürliche Zahlen "n" grundsätzlich nicht berechnet werden kann. Daher ist es unmöglich, für jedes "n" zu entscheiden, ob ihr

Funktionswert "$\Sigma(n)$" mit einer jeweils vorgegebenen natürlichen Zahl "k_n" übereinstimmt oder nicht. Auf ein letztes unentscheidbares Problem - das Halteproblem der TURING-Automaten - wird nachfolgend näher eingegangen.
Vgl. darüber hinaus zu weiteren unentscheidbaren Entscheidungsproblemen KLEENE (1952), S. 382ff.; HERMES (1952), S. 188f.; HERMES (1961), S. 262f.; LORENZEN,P. (1962), S. 140ff.; MENDELSON (1964), S. 259f.; ROGERS,H. (1967), S. 24ff., 32ff. u. 95f.; CARNAP (1968), S. 93ff.; PUTNAM,H. (1973), S. 71ff.; GAREY (1979), S. 12; BACHEM (1980), S. 840f.; BUCHER (1987), S. 252; RICHTER,M.M. (1988), S. 18; ZELEWSKI (1989c), S. 13 u. 16f.; DAUCHET (1988), S. 262ff.; HOWELL (1988a), S. 351.
Sogar im Rahmen des Petrinetz-Konzepts existieren unentscheidbare Probleme. Dazu gehören die Unentscheidbarkeit des:
- Problems, ob zwei Stelle/Transition-Netze mit gleichen Stellenmengen auch die gleichen Erreichbarkeitsmengen besitzen; vgl. COOPRIDER (1976), S. 24.
- Erreichbarkeitsproblems für Stelle/Transition-Netze, die zur Zeitnetzvariante von MERLIN erweitert sind; vgl. JONES,N. (1977), S. 296 (Beweis auf S. 296f.).
- Lebendigkeitsproblems für Höhere Netze, in denen Inhibitorkanten oder unendliche Mengen zulässiger Schaltfarben verwendet werden; vgl. VALETTE (1979b), S. 156; VIDAL-NAQUET (1982b), S. 8 (Inhibitorkanten und Schaltfarben von Netzen werden in dieser Arbeit an anderer Stelle näher erläutert).
- Beschränktheitsproblems in zwei Sonderfällen. Der erste bezieht sich auf die o.a. Stelle/Transition-Zeitnetze in der Variante von MERLIN; vgl. abermals JONES,N. (1977), S. 296(f). Der zweite Sonderfall erstreckt sich auf Höhere Netze mit Inhibitorkanten; vgl. VALETTE (1979b), S. 156.
- Problems, ob zwei Prädikat/Transition-Netze semantisch äquivalent sind; vgl. GENRICH (1988b), S. 246f. (unter semantischer Netzäquivalenz wird von GENRICH im wesentlichen die Möglichkeit verstanden, zwei Netze durch eine endliche Anzahl von Operationen wechselseitig so ineinander zu transformieren, daß sie sich in dieselben elementaren Netzsysteme entfalten lassen; vgl. S. 242 i.V.m. S. 238 u. 235).

Die Unentscheidbarkeit der beiden erstgenannten Probleme besitzt erhebliche Bedeutung für die formale Semantik von Netzen. Denn aus ihnen folgt, daß sich die semantische Kategorie der Äquivalenz von Netzen nicht generell entscheiden läßt. Für das zweite Problem ist dies offensichtlich. Hinsichtlich des ersten Problems folgt dies aus der früheren Festlegung, die Netzäquivalenz auf die Isomorphie ihrer Erreichbarkeitsgraphen zurückzuführen. Vgl. zu weiteren unentscheidbaren Problemen für Petrinetze BURKHARD (1982a), S. 92 u. 95 (bei Anwendung einer speziellen "fairen" Schaltregel).

44) Den Aufbau und die Funktionsweise von TURING-Automaten hat der Verf. an anderer Stelle ausführlich dargestellt; vgl. ZELEWSKI (1989c), S. 19ff. Vgl. darüber hinaus - als historischen Ursprung - TURING (1937a), S. 231ff. u. 241ff., sowie HERMES (1937), S. 114ff., insbesondere S. 118ff.; HERMES (1952), S. 184ff.; KLEENE (1952), S. 356ff.; PETER (1957), S. 202ff.; WANG,HA. (1957), S. 85ff.; DAVIS,M. (1958), S. 3ff.; SHEPHERDSON (1963), S. 233ff.; MENDELSON (1964), S. 229ff.; HARTMANIS (1965), S. 286ff.; FISCHER,P. (1965), S. 570ff.; PUTNAM,H. (1966), S. 140ff.; ROGERS,H. (1967), S. 13ff.; BRAUER (1968), S. 12ff.; MINSKY (1971), S. 160ff.; PUTNAM,H. (1973), S. 69ff.; BÖHLING (1974), S. 8ff.; HERSCHEL (1974), S. 67ff.; OTTMANN (1975a), S. 5ff.; BAUR (1976), S. 11ff.; SAVAGE (1976), S. 174ff.; MEHLHORN (1977), S. 190ff.; ZERVOS (1977), S. 285ff.; HERMES (1978), S. 18ff., 33ff. u. 203ff.; PAUL,W. (1978), S. 25ff. u. 62ff.; GAREY (1979), S. 23ff.; BOOLOS (1980), S. 19ff.; BRUCKER (1981), S. 148ff. u. 153ff.; WEIZENBAUM (1982), S. 80ff., insbesondere S. 88ff.; HOPCROFT (1984), S. 34ff.
Das Halteproblem für TURING-Automaten besteht in der Aufgabe, für jeden beliebigen TURING-Automaten und für jede beliebige Eingabeinformation zu entscheiden, ob dieser Automat - angewendet auf jene Eingabeinformation - nach endlich vielen internen informationsverarbeitenden Operationen in einem wohldefinierten internen Zustand anhält (terminiert) oder nicht. Dabei soll über das Terminieren des Automaten "a priori" entschieden werden, d.h. bevor er auf eine beliebige Eingabeinformation tatsächlich angewendet wird. Eine solche a priori-Untersuchung des Halteproblems von TURING-Automaten kann also nur auf die allgemeine Spezifizierung der Funktionsweise solcher Automaten und die Definition grundsätzlich zulässiger Eingabeinformationen Bezug nehmen. Unter diesen abstrakten Voraussetzungen wurde gezeigt: Das Halteproblem für TURING-Automaten ist unentscheidbar. Daher kann es kein Entscheidungsverfahren - keinen "Super"-Automaten - geben, mit dessen Hilfe sich das Terminieren oder Nichtanhalten eines *beliebigen* TURING-Automaten für eine *beliebige* Eingabeinformation *a priori* entscheiden ließe.
Vgl. zum Halteproblem für TURING-Automaten und zu seiner Unentscheidbarkeit TURING (1937a), S. 246ff., insbesondere S. 248 u. 262; VON NEUMANN/BURKS (1966), S. 52f. u. 124f.; ROGERS,H. (1967), S. 24ff.; BRAUER (1968), S. 49ff.; MINSKY (1971), S. 194ff.; SAVAGE (1976), S. 185ff.; COHORS-FRESENBORG (1977), S. 44f.; HERMES (1978), S. 144ff.; PAUL,W. (1978), S. 59f.; BOOLOS (1980), S. 49f.; HOPCROFT (1984), S. 45; ZELEWSKI (1986a), S. 937ff.; BINMORE (1987), S. 207; ZELEWSKI (1989c), S. 14 u. 16f.; BRAUER (1990a), S. 66.

45) Die CHURCH/POST/TURING-These besagt, daß alle algorithmisch oder "effektiv" berechenbaren Funktionen durch TURING-Automaten berechnet werden können. Oftmals wird dieser Sachverhalt in mittelbarer Weise durch die These ausgedrückt, daß alle algorithmisch oder "effektiv" berechenbaren Funktionen als rekursive Funktionen formuliert werden können. Für rekursive Funktionen ist aber nachgewiesen, daß sie grundsätzlich immer von TURING-Automaten berechnet werden können. Vgl. zu dieser - unmittelbaren oder mittelbaren - Verknüpfung zwischen algorithmischer oder "effektiver" Berechenbarkeit einerseits und TURING-Automaten andererseits durch die

CHURCH/POST/TURING-These CHURCH (1936a), S. 356ff.; TURING (1937a), S. 255; POST,E. (1944), S. 285; KLEENE (1952), S. 300, 317ff., 332, 356 u. 376; PETER (1957), S. 202ff.; SHEPHERDSON (1963), S. 217; MENDELSON (1964), S. 155f.; PUTNAM,H. (1966), S. 162; ROGERS,H. (1967), S. 20; BRAUER (1968), S. 5f.; MINSKY (1971), S. 149; STEGMÜLLER (1973a), S. 46f. u. 71; PUTNAM,H. (1973), S. 69; SCHNORR (1974), S. 20 u. 54; COHORS-FRESENBORG (1977), S. 44; HERMES (1978), S. 31; PAUL,W. (1978), S. 53f.; BACHEM (1980), S. 815; PETERSON,J. (1981), S. 201f.; WEIZENBAUM (1982), S. 94f.; HOPCROFT (1984), S. 39 u. 44; BINMORE (1987), S. 205; ZELEWSKI (1989a), S. 30ff. (einschließlich der Erörterung von Einwänden gegen die o.a. These in Fn. 91 auf S. 30), insbesondere S. 30 u. 37f.; BRAUER (1990a), S. 63 u. 67f., der auf S. 68 eine Ausweitung zu einer CHURCH/POST/TURING/GANDY-These andeutet.

46) Aufgrund der CHURCH/POST/TURING-These lassen sich alle algorithmisch berechenbaren Funktionen durch TURING-Automaten berechnen. Dies wurde in der voranstehenden Anmerkung dargelegt. Jeder Algorithmus läßt sich durch eine partielle Funktion spezifizieren. Sie bildet jede Information, die eine zulässige Eingabeinformation des Algorithmus darstellt, auf diejenige Information ab, die bei der Ausführung des Algorithmus als Ausgabeinformation erzeugt würde. Daher ist es möglich, jeden Algorithmus zunächst durch eine algorithmusspezifische Funktion zu spezifizieren und dann die Berechnung dieser Funktion für jede zulässige Eingabeinformation durch einen TURING-Automaten ausführen zu lassen. Wenn vom vermittelnden Glied der algorithmusspezifischen Funktion abstrahiert wird, gilt vereinfacht: Jeder Algorithmus kann durch einen TURING-Automaten so spezifiziert werden, daß der TURING-Automat die Algorithmusausführung leistet.

47) PROLOG-Kontrollstrukturen bestimmen die Art, in der die prädikatenlogischen Formeln eines PROLOG-Programms bei der Programmausführung durch eine kontrollstrukturspezifische Realisierung des Unifizierungs- und Resolutionskonzepts ausgewertet werden.

48) Dies hat KOWALSKI - der Nestor des logischen Programmierens - in der plastischen Formulierung "Algorithm = Logic + Control" zusammengefaßt. Sie findet sich im Original bei KOWALSKI (1979a), S. 424. Dabei ist mit "Logic" ein logisches Programm in Klauselform - z.B. ein PROLOG-Programm - gemeint, während "Control" die Kontrollstruktur der jeweils verwendeten Programmiersprache bezeichnet. Vgl. zu weiteren Erörterungen der o.a. "KOWALSKI-Gleichung" KOWALSKI (1979a), S. 425, 430 u. 432.

49) Dieser Sachverhalt wird auch kurz als TURING-Mächtigkeit der jeweils betrachteten Programmiersprache bezeichnet.

50) Dies bedeutet konkret: Jedes Paar aus einem PROLOG-Programm und einer PROLOG-Kontrollstruktur kann als ein TURING-Automat dargestellt werden - und jeder TURING-Automat läßt sich als eine Kombination eines PROLOG-Programms mit einer PROLOG-Kontrollstruktur formulieren.

51) Gleiches gilt auch für alle anderen Programmiersprachen. Denn TURING-Automaten stellen die abstrakte Spezifizierung von Algorithmen dar, die in *beliebigen* Programmiersprachen formuliert sein können. Daher handelt es sich bei der hier behandelten Schwierigkeit keineswegs um eine PROLOG-spezifische Thematik. In diesem Sinne folgert auch BRAUER (1990a), S. 66, aus der Unentscheidbarkeit des Halteproblems von TURING-Automaten: "Das Betriebssystem eines Computers ist nicht in der Lage zu entscheiden, ob ... ein Programm ... jemals hält ...".

52) Strenggenommen sind hier immer die kombinierten PROLOG-Programme und -Kontrollstrukturen gemeint. Der Einfachheit halber werden nachfolgend nur noch die PROLOG-Programme explizit angesprochen.

53) Existierte dagegen ein solcher omnipotenter Entscheidungsalgorithmus, dann könnte mit seiner Hilfe jedes lösbare Entscheidungsproblem in endlicher Zeit positiv oder negativ gelöst werden. Dies würde aber auch einen Entscheidungsalgorithmus für das Gültigkeitsproblem der Prädikatenlogik liefern. Dies verletzte jedoch deren nachgewiesene Unentscheidbarkeit.

54) Der Begriff des Nichtterminierens ist ein theoretischer Grenzbegriff, der in der Realität niemals erfüllt werden kann. Denn real existieren keine nicht-terminierenden, d.h. unendlich lange ausgeführten Programme. Vielmehr müssen alle realen Programmausführungen infolge von Ressourcenbeschränkungen nach endlicher Zeit abgebrochen werden, wenn sie nicht von selbst erfolgreich geendet sind. Daher meint der Begriff nicht-terminierender Programmausführung nur die *Fiktion* einer theoretisch unbegrenzten Ausführungsfortsetzung, die nur dann gültig *wäre*, falls von allen realen Ressourcenbeschränkungen abstrahiert werden *könnte*.

55) Vgl. dazu den einschränkenden Hinweis auf den fiktiven, alle realen Ressourcenbeschränkungen vernachlässigenden Charakter der Vorstellung nicht-terminierender Programme.

56) Dies muß jedoch nicht unbedingt der Fall sein. Zwar besteht ein PROLOG-Programm - wie jedes andere real implementierbare Programm der Automatischen Informationsverarbeitung - aus endlich vielen Programmanweisungen. Aber eine unendliche Programmausführung kann aus einer endlichen Anweisungsmenge auch als aperiodische Anweisungsfolge erzeugt werden. Ein Analogon ist die Notation irrationaler Zahlen als aperiodische, unendliche Dezimalzahlen, die aus einem endlichen Vorrat von nur zehn Dezimalziffern komponiert werden. Allerdings kommen bei der Programmiersprache PROLOG solche aperiodisch-unendlichen Anweisungsfolgen nicht in Betracht.

Denn ihre starre Kontrollstruktur führt bei unendlichen Programmausführungen stets zu festen Anweisungszyklen. Daher werden fortan unendliche Programmausführungen mit Endlosschleifen identifiziert. Darüber hinaus wird davon abgesehen, daß sich Endlosschleifen bei realen Programmausführungen niemals realisieren lassen, weil sie unendlich große Ressourcen voraussetzen, die in Wirklichkeit nicht zur Verfügung stehen. Darüber hinaus ist anzumerken, daß Endlosschleifen bei der Programmausführung nicht identisch sein müssen mit Zyklen von Erreichbarkeitsgraphen für Petrinetze. Denn solche Zyklen wiederholen ständig dieselbe, unveränderte Folge von Netzmarkierungen. In Endlosschleifen können hingegen die gleichen Programmstatements in rekursiver Weise auf ihre eigenen Ergebnisse angewendet werden. Dann werden zwar immer wieder die gleichen Statements ausgeführt; aber die Programmzustände wiederholen sich nicht in zyklischer Weise. Dies entspricht in Netzmodellen der Wiederholung des Schaltens gleicher Transitionen, die jedoch immer wieder zu neuen Netzmarkierungen führen. Dann resultiert ein Erreichbarkeitsgraph, der keinen Markierungszyklus aufweisen muß, statt dessen aber unendlich anwächst.

57) Das verhindert allerdings nicht, das Terminieren von *einzelnen* - aber eben nicht allen - PROLOG-Programmen positiv oder negativ entscheiden zu können.

58) Das wurde bereits kurz zuvor für beliebige PROLOG-Programme und -Kontrollstrukturen dargelegt. Es mag hier für den speziellen Fall des logischen Entscheidungsproblems "$\vdash_{BS}(FM_q \rightarrow p_u)$?" auf den ersten Blick Schwierigkeiten bereiten, den relevanten TURING-Automaten und seine Eingabeinformationen zu identifizieren. Sie lassen sich jedoch schnell beseitigen: Das interessierende PROLOG-Programm besteht hier aus der Formelmenge FM_q. Die PROLOG-Kontrollstruktur realisiert das kombinierte Unifizierungs- und Resolutionskonzept, das die vorausgesetzte indirekte Beweistechnik ermöglicht. Das Paar aus PROLOG-Programm und -Kontrollstruktur ist ein Algorithmus, der es für jede Eingabeinformation "p_u" zu untersuchen gestattet, ob sich die Formel p_u aus der Formelmenge FM_q ableiten läßt. Es handelt sich allerdings um keinen Entscheidungsalgorithmus, da sein Terminieren nicht für jede denkmögliche Formel p_u gesichert ist. Um Komplikationen zu vermeiden, die aus der Klauselnotation von PROLOG-Programmen resultieren können, wird vorausgesetzt, daß es sich bei den Eingabeinformationen um keine variablen Formeln $p_u(...,X_k,...)$ handeln darf, in deren Argumenten mindestens eine Variable X_k durch einen Allquantor gebunden wird. Den Normal bilden daher konstante Formeln $p_u(ob_1,...,ob_{K_u})$ als Eingabeinformationen. Allenfalls kommen noch variable Formeln $p_u(...,X_k,...)$ in Betracht, deren Variablen X_k ausschließlich durch Existenzquantoren gebunden sind. Unter diesen Voraussetzungen stellt der Algorithmus, der aus dem PROLOG-Programm "FM_q" und der zugehörigen PROLOG-Kontrollstruktur gebildet wird, einen TURING-Automaten dar, der für jede Eingabeinformation "p_u" zu untersuchen vermag, ob sich die Formel "p_u" aus der Formelmenge FM_q ableiten läßt.

59) Die Unentscheidbarkeit des Gültigkeitsproblems der Prädikatenlogik und die Unentscheidbarkeit des Halteproblems für TURING-Automaten lassen sich unmittelbar miteinander verknüpfen. Denn die erstgenannte prädikatenlogische Unentscheidbarkeit wird oftmals dadurch nachgewiesen, daß TURING-Automaten für die Untersuchung des Gültigkeitsproblems konstruiert und ihr nicht-terminierendes Verhalten aufgezeigt werden; vgl. dazu die Quellen, die zur Unentscheidbarkeit des Gültigkeitsproblems der Prädikatenlogik angeführt wurden, z.B. BOOLOS (1980), S. 112ff.

60) Die hier vorgetragene Argumentation knüpft an die algorithmische Unentscheidbarkeitsvariante an, die erstmals von CHURCH und TURING nachgewiesen wurde. Denn es wird auf Algorithmen Bezug genommen, die später dazu dienen, die Funktionsweise von TURING-Automaten oder von Konstrollstrukturen einer Programmiersprache zu determinieren. Ebenso hätte aber auch auf die dort skizzierte GÖDEL'sche Unentscheidbarkeits-Erkenntnis zurückgegriffen werden können, die sich auf einzelne Formeln anstatt Algorithmen bezieht. Denn in jedem arithmetisch erweiterten prädikatenlogischen Kalkül muß sich mindestens ein Entscheidungsproblem formulieren lassen, dessen spezifische Subjugatformel "$FM_q \rightarrow p_u$" weder bewiesen noch widerlegt werden kann. Folglich kann kein prädikatenlogisches Inferenzkonzept - und auch keine Kontrollstruktur einer prädikatenlogischen Programmiersprache existieren, daß das Entscheidungsproblem "$FM_q \rightarrow p_u$?" in endlicher Zeit zu bewältigen vermag.

61) Vgl. ROBINSON,J. (1965), S. 30 u. 37; ITZINGER (1976), S. 32f.; RAPHAEL (1976), S. 122; RICHTER,M.M. (1978), S. 195; HABEL (1983), S. 123.

62) Es existieren Ansätze, im Rahmen von dynamischen und temporalen Logiken zu untersuchen, ob die Lösungsverfahren für einzelne, konkret vorliegende Problemformulierungen terminieren, ohne die Verfahren selbst auszuführen. Diese Versuche sind jedoch bis heute über den Status der theoretischen Grundlagenforschung noch kaum hinausgewachsen. Sie können daher im folgenden unberücksichtigt bleiben, wenn die praktischen Konsequenzen des potentiellen Nichtterminierens für prädikatenlogische Objektmodellierungen erörtert werden. Vgl. zu solchen Studien der Verfahrensterminierung die Quellen, die zu dynamischen und temporalen Logikkalkülen angeführt wurden.

63) Vgl. zur Semi-Entscheidbarkeit der Prädikatenlogik im allgemeinen oder des kombinierten Unifizierungs- und Resolutionskonzepts im besonderen WANG,HA. (1960a), S. 224; HENSCHEN (1974), S. 591; BOOLOS (1980), S. 112 u. 142f.; HABEL (1983), S. 121f.; ELCOCK (1983a), S. 114; NIELSEN,M. (1984b); ZELEWSKI (1986a), S. 943ff.;

DELAHAYE (1987), S. 33f. (mit einer besonders transparenten und kompakten Zusammenfassung der Semi-Entscheidbarkeit), 44f., 126 u. 163; REIMER (1989), S. 86; ZELEWSKI (1991b), S. 239f.
Ohne nähere Erläuterung seiner Grundlagen und Konsequenzen findet sich der Begriff der Semi-Entscheidbarkeit auch bei REINFRANK (1985b), S. 42f. u. 109; BREWKA (1989), S. 95.

64) Der Einfachheit halber werden die problemspezifischen Subjugatformeln aus Entscheidungsproblemen "$FM_q \rightarrow p_u$?" auch nur als Formeln bezeichnet.

65) Diesen Fall, in dem die Allgemeingültigkeit einer tatsächlich nicht-allgemeingültigen Formel behauptet wird, deckt die (semantische) Vollständigkeit der Prädikatenlogik überhaupt nicht ab. Denn ihre Definition setzt bereits von vornherein Formeln voraus, die tatsächlich allgemeingültig sind. Daher erstreckt sich die Semi-Entscheidbarkeit der Prädikatenlogik in ihrem hier interessierenden Aspekt der Nichtterminierung nur auf jenen Fall, der von der prädikatenlogischen Vollständigkeit ignoriert wird. Diese Vereinbarkeit von prädikatenlogischer Vollständigkeit und Semi-Entscheidbarkeit wird auch bei HABEL (1983), S. 121f., besonders deutlich.

66) Vgl. zum *nicht-terminierenden* Verhalten *aller* Entscheidungsalgorithmen für das prädikatenlogische Gültigkeitsproblem WANG,HA. (1960a), S. 222; HENSCHEN (1974), S. 591; ITZINGER (1976), S. 33; RAPHAEL (1976), S. 122; RICHTER,M.M. (1978), S. 195; BOOLOS (1980), S. 112ff. u. 143; DELAHAYE (1987), S. 34; vgl. auch APPELRATH (1983), S. 41 u. 76, REIMER (1989), S. 86; ZELEWSKI (1991b), S. 239.
Es handelt sich um eine notwendige Eigenart *aller* prädikatenlogischen Inferenzkonzepte, also a fortiori auch aller PROLOG-Kontrollstrukturen.

67) Vgl. zu dieser alternativen Formulierung der Semi-Entscheidbarkeit BOOLOS (1980), S. 121 u. 142f.

68) Der Nachweis der Allgemeingültigkeit einer Formel durch ein positives Entscheidungsverfahren geschieht entweder direkt oder aber indirekt. Im zweiten Fall wird die Inkonsistenz des Negats der untersuchten Formel nachgewiesen. Auf die komplementäre Beziehung zwischen diesen beiden Nachweisansätzen wurde bereits hingewiesen. Ein negatives Entscheidungsverfahren würde dagegen - falls es existierte - die behauptete Allgemeingültigkeit einer Formel zurückweisen, indem es die Erfüllbarkeit des Negats dieser Formel nachweist. Sowohl positive als auch negative Entscheidungsverfahren zeichnen sich also dadurch aus, daß sie jeweils nur metasprachlich definierte Formeleigenschaften *nachweisen*: die Allgemeingültigkeit oder Inkonsistenz bzw. die Erfüllbarkeit. Keines dieser Verfahren vermag eine dieser Formeleigenschaften für *alle* prädikatenlogischen Formeln mit Sicherheit *zurückzuweisen*. Die Zurückweisung einer Formeleigenschaft kann nur dadurch gesichert erfolgen, daß die jeweils kontradiktorische Formeleigenschaft nachgewiesen wird. Die Allgemeingültigkeit einer beliebigen Formel läßt sich also nur dadurch zurückweisen, daß die Erfüllbarkeit des Negats dieser Formel nachgewiesen wird.

69) Dies bedeutet, daß zwar positive Entscheidungsverfahren existieren, mit denen die behauptete Allgemeingültigkeit oder Inkonsistenz von beliebigen Formeln nachgewiesen sowie die behauptete Erfüllbarkeit von Formeln zurückgewiesen werden kann. Aber es fehlen negative Entscheidungsverfahren, die es gestatten würden, die behauptete Allgemeingültigkeit oder Inkonsistenz von beliebigen Formeln zurückzuweisen bzw. die behauptete Erfüllbarkeit von Formeln nachzuweisen. Diese prädikatenlogische Asymmetrie zugunsten positiver Entscheidungsverfahren wird besonders deutlich von BOOLOS (1980), S. 121 u. 143, dargelegt.

70) Hinsichtlich des allgemeinen Gültigkeitsproblems wurde dies schon in einer früheren Anmerkung belegt. Da bei der Formulierung des logischen Entscheidungsproblems das Ausdrucksvermögen der Prädikatenlogik in keiner Weise eingeschränkt wird, trifft die Unentscheidbarkeit des Gültigkeitsproblems ebenso auf das Entscheidungsproblem zu.

71) Da semi-entscheidbare Entscheidungsprobleme einen Unterfall der unentscheidbaren Entscheidungsprobleme darstellen, läge es nahe, statt dessen von semi-*un*entscheidbaren Entscheidungsproblemen zu sprechen. Der Begriff der Semi-Entscheidbarkeit hat sich aber - sofern das zugrundeliegende Phänomen überhaupt berücksichtigt wird - durchgesetzt. Daher wird diese Begriffsbildung vom Verf. übernommen. Der Sachverhalt, daß die Semi-Entscheidbarkeit einen Unterfall der *Un*entscheidbarkeit von Problemen darstellt, wird besonders deutlich bei REIMER (1989), S. 86 ("Nichtentscheidbarkeit").

72) Allerdings ist der Hinweis von REINFRANK (1985b), S. 42f., bemerkenswert, daß sich nonmonotone Logiken - im Gegensatz zur monotonen Prädikatenlogik - nicht mehr semi-entscheidbar verhalten. Solche nonmonotonen Logiken werden an anderer Stelle für die Repräsentation temporalen Wissens herangezogen. Sie bilden einen Rahmen, in den sich die logische Qualität der Schaltregeln aus dem Petrinetz-Konzept einordnen läßt. Daher wäre prima facie zu erwarten, daß sich Entscheidungsprobleme, die mit der Hilfe von Netzen modelliert werden, noch nicht einmal semi-entscheidbar, sondern vollkommen unentscheidbar verhalten. Dies ist jedoch nicht der Fall. Denn es wird in dieser Arbeit aufgezeigt, daß das Petrinetz-Konzept die Prädikatenlogik im wesentlichen nicht überschreitet. Dies wurde erstens anläßlich der objektsprachlichen Repräsentation von Fakten demonstriert. Zweitens wird darauf an späterer Stelle zurückgekommen. Dort wird dargelegt, daß die Erreichbarkeit von Netzmarkierungen, die auf Anwendungen von Schaltregeln basiert, das prädikatenlogische Ausdrucksvermögen nur geringfügig übersteigt. Die zusätzlichen Aspekte dynamischer Logiken, die an dieser Stelle in das Petrinetz-Konzept einfließen, reichen nicht

aus, um die prädikatenlogische Semi-Entscheidbarkeit für das Petrinetz-Konzept aufzuheben. Denn die später angeführten Netztheoreme von MURATA/ZHANG und LAUTENBACH weisen nach: Prädikatenlogische Entscheidungsprobleme lassen sich anhand ihrer Netzmodelle so entscheiden, daß die netzbasierten Entscheidungsergebnisse mit den rein prädikatenlogisch ermittelten Entscheidungsergebnissen identisch sind. Folglich können sich die schaltregelbedingten Erweiterungen der Prädikatenlogik nicht so auswirken, daß im Petrinetz-Konzept die Semi-Entscheidbarkeit der Prädikatenlogik verloren geht.

73) Zu den seltenen Ausnahmen, die zwar nicht das Phänomen der Semi-Entscheidbarkeit selbst ansprechen, aber immerhin die praktische Bedeutung der zugrundeliegenden prädikatenlogischen Unentscheidbarkeit klar herausstellen, gehört MERTENS II (1990), S. 19: "Die Problematik der ... Prädikatenlogik liegt in ihrer Nichtentscheidbarkeit, so daß man bei einem laufenden Beweisprogramm niemals sicher quantifizieren kann, ob eine Formel nicht beweisbar ist oder ob der Beweis nur noch nicht gefunden wurde. Ein solches Beweisprogramm ist nicht in der Lage, selbständig zu terminieren, wenn der Beweis einer Formel nicht möglich ist; diese Tatsache führt zu einer gewissen Unsicherheit bezüglich des Status eines Ableitungsprozesses ...".
Bemerkenswert sind auch die Ausführungen von BINMORE (1987), S. 204ff., und BINMORE (1988), S. 12f. Zwar bezieht er sich weder auf die logische Programmierung, noch benutzt er den Begriff der Semi-Entscheidbarkeit. Doch stürzt er sich inhaltlich auf das Phänomen der Semi-Entscheidbarkeit, um im spieltheoretischen Kontext die Begrenzung praktischer Rationalität durch (prädikaten-)logische Sachverhalte herauszuarbeiten. Interessant hieran ist, daß die hier thematisierte operationale Dimension der Prädikatenlogik mit den Konzepten praktischer und prozeduraler Rationalität, die an früherer Stelle angedeutet wurden, kohärent verknüpft werden. Vgl. des weiteren die Erläuterungen, welche den Einfluß der Semi-Entscheidbarkeit auf betriebswirtschaftliche Modellierungsaufgaben beleuchten.

74) Die präzise Herausarbeitung der Semi-Entscheidbarkeit erfordert größeren formalen Aufwand. Eine analoge, aber wesentlich einfachere Darstellung findet sich bei ZELEWSKI (1986a), S. 950ff. Dort wurde vor allem von wesentlichen Aspekten der prädikatenlogischen Differenzierung zwischen modelltheoretischen Schlußfolgerungen und beweistheoretischen Inferenzen abstrahiert.
Ausgangspunkt ist ein konsistentes Formelsystem. Seine Formelmenge FM_q ist aus den Formeln $p_1,...,p_{Hq}$ mit $H_q \in \mathcal{N}_+$ und $H_q \geq 2$ konjunktiv zusammengesetzt: $FM_q \Leftrightarrow p_1 \wedge ... \wedge p_{Hq}$. (Die nachfolgende Argumentation läßt sich auch auf die Sonderfälle $H_q=1$ und $H_q=0$ ausdehnen, wäre dann aber komplizierter zu formulieren.) Ein Entscheidungsproblem wird hinsichtlich dieses Formelsystems durch die Formel p_u spezifiziert. Ohne Beschränkung der Allgemeingültigkeit wird unterstellt, daß die problemspezifische Formel p_u in der Formelmenge FM_q nicht enthalten ist: $p_u \neq FM_q$. (Andernfalls würde das Formelsystem mit der Formelmenge FM_q-$\{p_u\}$ betrachtet.) Alle involvierten Formeln p_h mit $h \in \{1,...,H_q\}$ und p_u werden als geschlossene Formeln unterstellt, so daß von Komplizierungen durch die Festlegung zulässiger Variablenbelegungen und -unifizierungen abgesehen werden kann. Unter diesen Annahmen wird behauptet, die Formel p_u sei eine logische Konsequenz der Formelmenge FM_q: $FM_q \models p_u$. Dies entspricht der Behauptung, die Subjugatformel "$FM_q \rightarrow p_u$" sei allgemeingültig. Das Entscheidungsproblem "$\models (FM_q \rightarrow p_u)$?" besteht dann in der Aufgabe festzustellen, ob die behauptete Allgemeingültigkeit des problemspezifischen Subjugats "$FM_q \rightarrow p_u$" tatsächlich zutrifft (positive Lösung) oder nicht (negative Lösung).
Als Beweissystem wird das kombinierte Unifizierungs- und Resolutionskonzept UR unterstellt, das die beweistheoretischen Qualitäten der Konsistenz, Vollständigkeit und Korrektheit erfüllt. (Es könnte aber durch jedes andere konsistente, vollständige und korrekte prädikatenlogische Beweissystem BS ersetzt werden.) Infolge seiner Vollständigkeit und Korrektheit trifft die Allgemeingültigkeitsbehauptung "$\models (FM_q \rightarrow p_u)$" genau dann zu, wenn die Beweisbarkeitsbehauptung "$\vdash_{UR} (FM_q \rightarrow p_u)$" zutrifft. Eine Behauptung heißt genau dann verifiziert (falsifiziert), wenn ihr Zutreffen mit beweistheoretischen Formeltransformationen nachgewiesen bzw. widerlegt worden ist. Alle nachfolgenden Überlegungen über Inferenzen im Beweissystem UR erstrecken sich direkt nur noch auf diese Beweisbarkeitsbehauptung, gelten aber indirekt vermittels Vollständigkeit und Korrektheit ebenso für die originäre Allgemeingültigkeitsbehauptung.
Mit dem metasprachlichen Symbol "«»" werden (meta-)metasprachliche Folgerungsbeziehungen ausgedrückt, die innerhalb des Beweissystems UR in beiden Richtungen gelten. Sie erlauben Transformationen einer Beweisbarkeitsbehauptung, gegenüber denen das Ergebnis der Behauptungsuntersuchung invariant ist. Die Konsistenz des Beweissystems verbietet u.a., daß die Erfüllbarkeit und die Inkonsistenz einer Formel zugleich zutreffen können. Schließlich wird das tertium non datur-Prinzip vorausgesetzt. Ihm zufolge gilt u.a. für alle Formeln: Wenn eine Formel inkonsistent (beweisbar) ist, dann muß ihr Negat beweisbar (inkonsistent) sein, ohne daß das Negat selbst konstruktiv bewiesen worden wäre. Dies ist das beweistheoretische Analogon zur modelltheoretischen Komplementarität zwischen der Allgemeingültigkeit einer Formel und der Inkonsistenz ihres Negats. Sie wurde bereits im Kontext des allgemeinen prädikatenlogischen Gültigkeitsproblems benutzt.
Die Beweisbarkeitsbehauptung "$\vdash_{UR} (FM_q \rightarrow p_u)$" wird zunächst in die Beweisbarkeitsbehauptung ihrer adjunktiven Normalform transformiert:

$$\vdash_{UR} (FM_q \rightarrow p_u) \quad «» \quad \vdash_{UR} ((p_1 \wedge ... \wedge p_{Hq}) \rightarrow p_u)$$

$$\vdash_{UR} ((p_1 \wedge ... \wedge p_{Hq}) \rightarrow p_u) \quad «» \quad \vdash_{UR} ((\neg p_1) \vee ... \vee (\neg p_{Hq}) \vee p_u)$$

Die erste Transformationszeile resultiert aus der früher dargelegten implizit konjunktiven Formelverknüpfung in der Formelmenge FM_q. Die zweite Transformationszeile benutzt die definitorische Reduktion von Subjugaten auf Adjugate und Negate; vgl. CARNAP (1960a), S. 8; BUCHER (1987), S. 106. Durch transitives Komprimieren der Transformationsschritte ergibt sich:

$$\vdash_{UR} (FM_q \rightarrow p_u) \quad \text{«»} \quad \vdash_{UR} ((\neg p_1) \vee \ldots \vee (\neg p_{Hq}) \vee p_u)$$

Aufgrund der indirekten Beweistechnik des Unifizierungs- und Resolutionskonzepts UR wird nicht versucht, die Beweisbarkeitsbehauptung "$\vdash_{UR} ((\neg p_1) \vee \ldots \vee (\neg p_{Hq}) \vee p_u)$" zu verifizieren. Statt dessen wird die Falsifizierung des kontradiktorischen Gegenteils dieser Behauptung angestrebt. Das kontradiktorische Gegenteil der Beweisbarkeitsbehauptung ist die Behauptung, daß die ursprüngliche Behauptung "$(\neg p_1) \vee \ldots \vee (\neg p_{Hq}) \vee p_u$" unbeweisbar ist. Diese Unbeweisbarkeitsbehauptung bedeutet - wie bereits ausgeführt wurde -, daß aus dem Negat der ursprünglichen Behauptung kein Widerspruch abgeleitet werden kann. Für dieses Negat gilt nach DE MORGAN die Äquivalenz: $\neg ((\neg p_1) \vee \ldots \vee (\neg p_{Hq}) \vee p_u) \Leftrightarrow (p_1 \wedge \ldots \wedge p_{Hq} \wedge (\neg p_u))$. Es wird also zunächst - hypothetisch - die Widerspruchsfreiheit der Formel $(p_1 \wedge \ldots \wedge p_{Hq} \wedge (\neg p_u))$ behauptet.
Die Formel "$p_1 \wedge \ldots \wedge p_{Hq} \wedge (\neg p_u)$" ist die konjunktive Normalform für das kontradiktorische Gegenteil der zu beweisenden, in adjunktive Normalform transformierten Formel "$(\neg p_1) \vee \ldots \vee (\neg p_{Hq}) \vee p_u$". Diese beiden zentralen Formeln werden fortan als negative bzw. positive Entscheidungsformeln bezeichnet. Infolge der Bidirektionalität der Formeltransformationen des Beweissystems - sowie wegen seiner Vollständigkeit und Korrektheit - gilt für diese beiden ausgezeichneten Formeln:

- Die positive Entscheidungsformel "$(\neg p_1) \vee \ldots \vee (\neg p_{Hq}) \vee p_u$" ist genau dann beweisbar, wenn die Beweisbarkeitsbehauptung "$\vdash_{UR} (FM_q \rightarrow p_u)$" - und somit auch die Allgemeingültigkeitsbehauptung "$\models (FM_q \rightarrow p_u)$" - tatsächlich zutreffen. Die Beweisbarkeit der positiven Entscheidungsformel und die positive Lösung des Entscheidungsproblems korrespondieren also wechselseitig miteinander.

- Die negative Entscheidungsformel "$p_1 \wedge \ldots \wedge p_{Hq} \wedge (\neg p_u)$" ist genau dann widerspruchsfrei, wenn die Beweisbarkeitsbehauptung "$\vdash_{UR}(FM_q \rightarrow p_u)$" - und somit auch die Allgemeingültigkeitsbehauptung "$\models (FM_q \rightarrow p_u)$" - tatsächlich nicht zutreffen. Die Widerspruchsfreiheit der negativen Entscheidungsformel und die negative Lösung des Entscheidungsproblems korrespondieren also wechselseitig miteinander.

- Da die Beweisbarkeitsbehauptung der positiven und die Widerspruchsfreiheitsbehauptung der negativen Entscheidungsformel kontradiktorische Gegenteile darstellen, muß immer genau eine dieser beiden Behauptungen zutreffen. Denn infolge des prädikatenlogischen tertium non datur-Prinzips muß einerseits mindestens einer von zwei kontradiktorischen Ausdrücken zutreffen. Andererseits kann aufgrund der prädikatenlogischen Widerspruchsfreiheit von zwei kontradiktorischen Ausdrücken immer nur höchstens einer von beiden abgeleitet werden. Daher impliziert das Zutreffen einer dieser Behauptungen das Nichtzutreffen der jeweils anderen Behauptung.

Die hypothetische Widerspruchsfreiheitsbehauptung des indirekten Beweisansatzes für die negative Entscheidungsformel "$p_1 \wedge \ldots \wedge p_{Hq} \wedge (\neg p_u)$" ist genau dann falsifiziert, wenn sich die Inkonsistenz dieser Entscheidungsformel ableiten läßt. Daher wird versucht, die hypothetische Widerspruchsfreiheitsbehauptung dadurch zu widerlegen, daß aus der negativen Entscheidungsformel die kontradiktorische Formel "$\emptyset$" abgeleitet wird. Im Falle des Unifizierungs- und Resolutionskonzepts gilt: Die negative Entscheidungsformel ist genau dann inkonsistent, wenn aus ihr die leere Klausel abgeleitet werden kann. Die Ableitungsschritte (Inferenzen) lassen sich als Suchoperationen in einem abstrakten logischen Inferenzraum auffassen, der alle kombinatorisch zulässigen Inferenzen umfaßt. Dieser Inferenzraum kann auch für Entscheidungsprobleme, die nur mit endlich vielen Formeln $p_1,\ldots,p_{Hq},p_u$ definiert worden sind, unendlich groß sein. Als "richtiger" Sektor des Inferenzraums wird derjenige - nicht notwendig zusammenhängende - Teilraum bezeichnet, in dem zulässige Inferenzen auf die Inkonsistenz der negativen Entscheidungsformel stoßen (sofern diese tatsächlich inkonsistent ist). Für das Unifizierungs- und Resolutionskonzept ist dieser Teilraum durch alle Inferenzen definiert, deren Anwendungsresultat jeweils die leere Klausel ist.
Angenommen, der Falsifizierungsversuch gelingt im Inferenzraum: Dann läßt sich das tertium non datur-Prinzip anwenden, weil die negative Entscheidungsformel per constructionem das Negat ihres positiven Pendants ist. Daher folgt aus der Widersprüchlichkeit der negativen Entscheidungsformel "$p_1 \wedge \ldots \wedge p_{Hq} \wedge (\neg p_u)$" die ursprünglich behauptete Beweisbarkeit der positiven Entscheidungsformel "$(\neg p_1) \vee \ldots \vee (\neg p_{Hq}) \vee p_u$". Folglich sind die Beweisbarkeitsbehauptung "$\vdash_{UR} (FM_q \rightarrow p_u)$" verifiziert und die ursprüngliche Allgemeingültigkeitsbehauptung "$\models (FM_q \rightarrow p_u)$" gefolgert worden; q.e.d. Falls dagegen der Falsifizierungsversuch erfolglos abgebrochen werden muß, ist die Lösung des Entscheidungsproblems unbekannt.
Es läßt sich jedoch *zusätzlich* die Entscheidungsregel des "negation by failure"-Prinzips heranziehen: Ihr zufolge wird ein Prozedurabbruch als Verifizierung der hypothetischen Widerspruchsfreiheitsbehauptung für die negative Entscheidungsformel festgesetzt. Dies entspricht der Erfüllbarkeit der Konjugatformel "$FM_q \wedge (\neg p_u)$", also der *negativen* Lösung des ursprünglichen Entscheidungsproblems. Vgl. zu diesem "negation by failure"-Prinzip, das oftmals auch als "negation as failure"-Prinzip angesprochen wird, KOWALSKI (1978), S. 95ff.; REITER (1978b), S. 56

u. 60; CLARK,K. (1978), S. 293ff., insbesondere S. 294; BULLERS (1980), S. 355; APT (1982), S. 855f. u. 858; AIDA (1983), S. 90; BIBEL (1984), S. 158f.; REINFRANK (1985b), S. 37; SHARPE (1985), S. 225; SERGOT (1986), S. 379; ZELEWSKI (1986a), S. 274; KOWALSKI (1987a), S. 136 u. 138ff.; BARTH,G. (1987), S. 220; KLEINE BÜNING (1988a), S. 59; WEIJLAND (1988), S. 721ff.; BEZEM (1988), S. 26; POWERS (1988), S. 958; WEDEKIND (1989c), S. 24 u. 26f.; ZELL,A. (1989), S. 99; KLEINHANS (1989), S. 88; LIFSCHITZ (1989), S. 315ff., insbesondere S. 317; REIMER (1989), S. 86 (dort nur indirekt als pragmatische Lösung der Nichtentscheidbarkeit thematisiert); ZELEWSKI (1991b), S. 244f. Auf dieses Prinzip wird an späterer Stelle noch einmal zurückgekommen.

Konstitutiv für die Semi-Entscheidbarkeit ist nun folgende Fallunterscheidung für die Überprüfung der Beweisbarkeitsbehauptung "$\vdash_{UR} (FM_q \rightarrow p_u)$" durch einen indirekten Beweisversuch. Es müssen zwei Fälle unterschieden werden:

Fall "P": Das Entscheidungsproblem "$\models (FM_q \rightarrow p_u)$?" besitzt die positive Lösung, d.h. die problemspezifische Subjugatformel "$FM_q \rightarrow p_u$" ist tatsächlich allgemeingültig. Daher muß die positive Entscheidungsformel "$(\neg p_1) \vee ... \vee (\neg p_{Hq}) \vee p_u$" beweisbar sein. Dies ist jedoch zu Beginn eines Beweisversuchs unbekannt. Daher verläuft die indirekte Inferenzprozedur in folgenden gedanklichen Schritten SP_m:

SP_1: Aufstellen der hypothetischen Widerspruchsfreiheitsbehauptung für die negative Entscheidungsformel "$p_1 \wedge ... \wedge p_{Hq} \wedge (\neg p_u)$".

SP_2: Es wird versucht, die hypothetische Widerspruchsfreiheitsbehauptung dadurch zu falsifizieren, daß die negative Entscheidungsformel als inkonsistent aufgezeigt wird.

SP_3: Da die positive Entscheidungsformel qua Voraussetzung beweisbar ist, kann die Widerspruchsfreiheitsbehauptung tatsächlich nicht zutreffen. Folglich existiert im Inferenzraum die kontradiktorische Formel "$\varnothing$".

SP_4: Die Inferenzprozedur kann grundsätzlich in endlicher Zeit die Inkonsistenz der hypothetisch unterstellten negativen Entscheidungsformel durch Ableitung der kontradiktorischen Formel "$\varnothing$" feststellen.

SP_5: Falls die verfügbaren Ressourcen für die Prozedurausführung ausreichen und die Kontrollstruktur der Inferenzprozedur den "richtigen" Sektor des Inferenzraums untersucht, wird die hypothetische Widerspruchsfreiheitsbehauptung tatsächlich falsifiziert.

SP_6: Wenn die hypothetische Widerspruchsfreiheitsbehauptung tatsächlich falsifiziert wird, folgt daraus die Verifizierung der ursprünglichen Beweisbarkeitsbehauptung "$\vdash_{UR} (FM_q \rightarrow p_u)$". Daher trifft die Allgemeingültigkeitsbehauptung "$\models (FM_q \rightarrow p_u)$" zu. Die tatsächliche positive Lösung des Entscheidungsproblems "$\models (FM_q \rightarrow p_u)$?" wird korrekt erkannt; q.e.d.

SP_7: Falls die verfügbaren Ressourcen für die Prozedurausführung nicht ausreichen oder falls die Kontrollstruktur der Inferenzprozedur den "richtigen" Sektor des Inferenzraums immer wieder verfehlt, wird der Beweisversuch erfolglos abgebrochen. Die tatsächliche positive Lösung des Entscheidungsproblems "$\models (FM_q \rightarrow p_u)$?" bleibt aufgrund praktischer Ausführungsmängel unerkannt.

SP_8: Falls der Schritt SP_7 durch die Entscheidungsregel des "negation by failure"-Prinzips ergänzt wird, gilt die hypothetische Widerspruchsfreiheitsbehauptung der negativen Entscheidungsformel aus Schritt SP_1 als verifiziert. Daraus "folgen" die Falsifizierung der kontradiktorischen Beweisbarkeitsbehauptung "$\vdash_{UR} (FM_q \rightarrow p_u)$" und der Allgemeingültigkeitsbehauptung "$\models (FM_q \rightarrow p_u)$". Für das Entscheidungsproblem "$\models (FM_q \rightarrow p_u)$?" wird fehlerhaft die negative Lösung ausgewiesen.

Fall "N": Das Entscheidungsproblem "$\models (FM_q \rightarrow p_u)$?" besitzt die negative Lösung, d.h. die Subjugatformel "$FM_q \rightarrow p_u$" ist tatsächlich nicht allgemeingültig. Daher müssen die Konjugatformel "$FM_q \wedge (\neg p_u)$" erfüllbar und die korrespondierende negative Entscheidungsformel "$p_1 \wedge ... \wedge p_{Hq} \wedge (\neg p_u)$" widerspruchsfrei sein. Wie schon im Fall "P" steht dieses Lösungswissen zu Beginn eines Beweisversuchs nicht zur Verfügung. Daher nimmt eine Inferenzprozedur nunmehr folgende gedankliche Verlaufsform mit Schritten SN_n an:

SN_1: Aufstellen der hypothetischen Widerspruchsfreiheitsbehauptung für die negative Entscheidungsformel "$p_1 \wedge ... \wedge p_{Hq} \wedge (\neg p_u)$" (identisch mit SP_1).

SN_2: Es wird versucht, die hypothetische Widerspruchsfreiheitsbehauptung dadurch zu falsifizieren, daß die negative Entscheidungsformel als inkonsistent aufgezeigt wird (identisch mit SP_2).

SN_3: Da die negative Entscheidungsformel qua Voraussetzung widerspruchsfrei und das Beweissystem UR konsistent ist, kann die kontradiktorische Formel "$\varnothing$" im Inferenzraum nicht existieren.

SN_{4a}: Obwohl die kontradiktorische Formel "$\varnothing$" nicht existiert, wird sie dennoch von der Inferenzprozedur gesucht, da per constructionem ein Falsifizierungsversuch unternommen wird.

SN_{4b}: Falls der Inferenzraum endlich ist, kann die Inferenzprozedur diesen Raum grundsätzlich vollständig durchsuchen und am Ende feststellen, daß die negative Entscheidungsformel tatsächlich nicht inkonsistent, sondern erfüllbar ist.

SN_5: Wenn ein endlicher Inferenzraum vorliegt und die verfügbaren Ressourcen für seine vollständige Untersuchung ausreichen, dann wird die hypothetische Widerspruchsfreiheitsbehauptung - entgegen der Intention des indirekten Beweisansatzes - tatsächlich verifiziert.

SN_6: Wenn die hypothetische Widerspruchsfreiheitsbehauptung tatsächlich verifiziert wird, folgt daraus die Falsifizierung der ursprünglichen Beweisbarkeitsbehauptung " $\vdash_{UR} (FM_q \rightarrow p_u)$". Daher trifft die Allgemeingültigkeitsbehauptung " $\models (FM_q \rightarrow p_u)$" nicht zu. Die tatsächliche negative Lösung des Entscheidungsproblems " $\models (FM_q \rightarrow p_u)$?" wird korrekt erkannt; q.e.d.

SN_{7a}: Falls der Inferenzraum zwar endlich ist, aber die verfügbaren Ressourcen für seine vollständige Durchsuchung nicht ausreichen oder die Kontrollstruktur der Inferenzprozedur den "richtigen" Sektor des Inferenzraums immer wieder verfehlt, wird der Beweisversuch erfolglos abgebrochen. Die negative Lösung des Entscheidungsproblems bleibt aufgrund praktischer Ausführungsmängel unerkannt.

SN_{7b}: Falls der Inferenzraum unendlich ist, kann er von der Inferenzprozedur grundsätzlich nicht in endlicher Zeit vollständig durchsucht werden. Da die Inferenzprozedur nicht terminiert, muß sie in der Realität - d.h. bei beschränkten Ausführungsressourcen - immer erfolglos abgebrochen werden. Folglich läßt sich die tatsächliche Erfüllbarkeit der negativen Entscheidungsformel grundsätzlich nicht erkennen. Die negative Lösung des Entscheidungsproblems bleibt aufgrund theoretischer Ausführungsmängel unerkannt.

SN_8: Falls die Schritte SN_{7a} oder SN_{7b} durch die Entscheidungsregel des "negation by failure"-Prinzips ergänzt werden, gilt jeweils die hypothetische Widerspruchsfreiheitsbehauptung der negativen Entscheidungsformel aus Schritt SN_1 als verifiziert. Daraus folgen die Falsifizierung der Beweisbarkeitsbehauptung " $\vdash_{UR} (FM_q \rightarrow p_u)$" und der Allgemeingültigkeitsbehauptung " $\models (FM_q \rightarrow p_u)$". Für das Entscheidungsproblem " $\models (FM_q \rightarrow p_u)$?" wird die negative Lösung korrekt ausgewiesen.

Aus der Fallunterscheidung ergeben sich folgende Charakteristika der Semi-Entscheidbarkeit:

- Wenn von realen Ressourcenbeschränkungen und Kontrollstrukturen abstrahiert wird, führt die Inferenzprozedur garantiert zur korrekten Erkenntnis der Lösung eines Entscheidungsproblems, falls dieses Entscheidungsproblem tatsächlich die positive Lösung besitzt.

- Wenn von realen Ressourcenbeschränkungen und Kontrollstrukturen abstrahiert wird, vermag die Inferenzprozedur nicht mehr sicherzustellen, zur korrekten Erkenntnis der Lösung eines Entscheidungsproblems zu führen, falls dieses Entscheidungsproblem tatsächlich die negative Lösung besitzt. Unendliche Suchräume verhindern die Erkenntnis der korrekten Problemlösung durch Nicht-Terminieren der Prozedurausführung.

- Trotz Abstraktion von realen Ressourcenbeschränkungen und Kontrollstrukturen führt die Inferenzprozedur aufgrund der voranstehenden Asymmetrie nur dann mit Sicherheit zur korrekten Erkenntnis der tatsächlichen Problemlösung, falls das Entscheidungsproblem tatsächlich positiv lösbar ist. Daher ist die Inferenzprozedur für den allgemeinen Fall *beliebiger* Entscheidungsprobleme unzuverlässig. Grund der Unzuverlässigkeit ist hier das Nichterkennen der tatsächlichen Problemlösung bei unendlichen Suchräumen für Entscheidungsprobleme mit negativen Lösungen (theoretische Unzuverlässigkeit).

- Falls reale Ressourcenbeschränkungen oder Kontrollstrukturen berücksichtigt werden, wird die Unzuverlässigkeit von Inferenzprozeduren um einen zweiten Aspekt ergänzt. Das Erkennen der tatsächlichen Problemlösung läßt sich nunmehr weder für Entscheidungsprobleme mit negativen noch für Entscheidungsprobleme mit positiven Lösungen garantieren (ressourcen- und kontrollbedingte Unzuverlässigkeit). Die Inferenzprozedur zeigt nur dann die tatsächliche Problemlösung korrekt an, wenn die Prozedurausführung innerhalb ihrer realen Ausführungsumgebung erfolgreich endet. Aus einem erfolglosen Prozedurabbruch erfolgt überhaupt keine Erkenntnis über die tatsächliche Problemlösung.

- Sofern die Entscheidungsregel des negation by failure-Prinzips für den Fall erfolgloser Prozedurausführungen hinzugefügt wird, resultiert ein dritter Aspekt der Unzuverlässigkeit. Die regelinduzierte Lösungsfestsetzung ist genau dann korrekt (potentiell fehlerhaft), wenn das untersuchte Entscheidungsproblem negativ (positiv) lösbar ist. Da jedoch die tatsächliche Problemlösung des untersuchten Problems - unabhängig von der ausgeführten und abgebrochenen Inferenzprozedur - im allgemeinen unbekannt ist, kann nicht festgestellt werden, ob die Regelanwendung korrekte Ergebnisse liefert (regelbedingte Unzuverlässigkeit). Vgl. zur potentiellen Fehlerhaftigkeit des negation by failure-Prinzips auch WESTPHAL (1986), S. 236; LIFSCHITZ (1989), S. 317.

Das negation by failure-Prinzip läßt sich als ein Bestandteil der Kontrollstruktur einer Inferenzprozedur auffassen. Daher wird die o.a. kontrollbedingte Unzuverlässigkeit (i.e.S.) mit der regelbedingten Unzuverlässigkeit zur kontrollbedingten Unzuverlässigkeit i.w.S. zusammengefaßt. Kontrollbedingte Unzuverlässigkeit i.w.S. und ressourcenbedingte Unzuverlässigkeit bilden zusammen die praktische Unzuverlässigkeit. Alle Aspekte der praktischen und theoretischen Unzuverlässigkeit zusammen unterstreichen die herausragende Bedeutung, die der Semi-Entscheidbarkeit von Beweissystemen und ihren zugrundeliegenden Inferenzkonzepten zukommt. Da die Semi-Entscheidbarkeit allen prädikatenlogischen Inferenzkonzepten zukommt, limitieren die skizzierten Unzuverlässigkeitsaspekte grundsätzlich alle theoretischen und praktischen Anwendungen der Prädikatenlogik. Dies gilt a fortiori auch für die prädikatenlogische Implementierung von Synthetischen Netzen in dieser Arbeit. Dies unterstreicht den eingangs geäußerten Vorbehalt gegenüber einem naiven Glauben an die "universelle prädikatenlogische Leistungsfähigkeit".

75) Um Mißverständnisse zu vermeiden, werden die charakteristischen Unterschiede von Vollständigkeit (und Korrektheit), Unentscheidbarkeit und Semi-Entscheidbarkeit der Prädikatenlogik noch einmal hervorgehoben:

- Die (semantische) *Vollständigkeit* der Prädikatenlogik drückt aus, daß mindestens ein Beweissystem existiert, in dem *jede allgemeingültige* Formel mit syntaktischen Inferenzen aus den Axiomen des Beweissystems abgeleitet und somit bewiesen werden kann. Dies garantiert noch nicht die Existenz eines Entscheidungsalgorithmus, der das Gültigkeitsproblem der Prädikatenlogik für *jede* prädikatenlogische Formel lösen könnte. Denn die prädikatenlogische Vollständigkeit garantiert nur die Existenz eines Entscheidungsalgorithmus, der alle allgemeingültigen Formeln als solche zu erkennen vermag. Wie ein solcher Algorithmus reagiert, wenn er auf eine nicht-allgemeingültige Formel angewendet wird, bleibt im Rahmen der prädikatenlogischen Vollständigkeit vollkommen offen.

- Die *Unentscheidbarkeit* der Prädikatenlogik besagt, daß es keinen universellen, finiten und syntaktisch definierten Algorithmus gibt, mit dessen Hilfe sich die Allgemeingültigkeit aller zulässigen prädikatenlogischen Formeln in einem beliebigen Beweissystem entscheiden läßt. Es existiert kein Beweissystem, in dem *jede* Formel mit syntaktischen Inferenzen entweder bewiesen oder aber widerlegt werden kann (syntaktische Unvollständigkeit).

- Die *Semi-Entscheidbarkeit* der Prädikatenlogik faßt die Unentscheidbarkeit der Prädikatenlogik mit der Existenz von semantisch vollständigen und korrekten Beweissystemen zusammen. Die Semi-Entscheidbarkeit betrifft die Aufgabe, die behauptete Allgemeingültigkeit einer beliebigen prädikatenlogischen Formel in einem semantisch vollständigen und korrekten Beweissystem durch syntaktische Inferenzen zu entscheiden, ohne die Formelgültigkeit a priori zu kennen. Dann gilt:
  - *Falls* die Formel tatsächlich allgemeingültig ist, kann sie infolge semantischer Vollständigkeit des Beweissystems mit endlichem Ressourceneinsatz bewiesen und infolge Korrektheit des Beweissystems als allgemeingültig erkannt werden. Entscheidungsprozesse für Entscheidungsprobleme, die tatsächlich eine positive Lösung besitzen, terminieren daher mit Sicherheit.
  - *Falls* die Formel tatsächlich nicht allgemeingültig ist, folgt aus der prädikatenlogischen Unentscheidbarkeit, daß die Inferenzen im Beweissystem unendlich fortschreiten können (aber nicht müssen). Entscheidungsprozesse für Entscheidungsprobleme, die tatsächlich eine negative Lösung besitzen, brauchen nicht zu terminieren. Aus dem erfolglosen Inferenzabbruch nach Erreichen einer beliebig großen, aber endlichen Ressourceneinsatzschranke folgt kein sicheres Wissen, ob die untersuchte Formel allgemeingültig ist oder nicht.

76) Mit dieser partiellen Unzuverlässigkeit korrespondiert das CHURCH/ROSSER-Konzept für die Überprüfung, ob Input/Output-Systeme (Transitionssysteme) korrekt funktionieren. Dabei wird die Funktionsweise eines solchen Systems durch drei Prädikate beschrieben, die das Ein-/Ausgabeverhalten des Systems festlegen. Das Prädikat $ZUL(ein(M_0))$ muß von jeder zulässigen Eingabeinformation "ein" erfüllt werden, mit der die Systemoperationen im Ausgangszustand M_0 der Systems gestartet werden. Das Prädikat $ERR(M_0,M_e)$ drückt aus, daß das Input/Output-System nach endlicher Zeit einen wohldefinierten Endzustand M_e erreicht, nachdem es im Ausgangszustand M_0 gestartet wurde. Das Prädikat $SPE(ein(M_0),aus(M_e))$ spezifiziert für jede zulässige Eingabeinformation "$ein(M_0)$" diejenige Ausgabeinformation "$aus(M_e)$", die vom Input/Output-System erzeugt werden soll. Die zeitlogischen Operatoren $G_0(...)$ und $F_0(...)$ drücken aus, daß eine prädikatenlogischen Formel für alle bzw. mindestens einen derjenigen Systemzustände gültig ist, die auf den Ausgangszustand M_0 des Input/Output-Systems folgen. Auf der Grundlage dieser Vereinbarungen läßt sich die Korrektheit der Systemfunktionsweise in zwei Bestandteile aufspalten:

- Das System funktioniert genau dann *partiell korrekt*, wenn für sein Ein-/Ausgabeverhalten gilt:

  $$ZUL(ein(M_0)) \quad\Rightarrow\quad G_0(ERR(M_0,M_e) \rightarrow SPE(ein(M_0),aus(M_e)))$$

  Falls das System im Ausgangszustand M_0 auf eine zulässige Eingabeinformation angewendet wird, dann gilt für alle zukünftigen Systemzustände: *Wenn* das System nach endlicher Operationszeit einen wohldefinierten Endzustand M_e erreicht, dann erfüllt dessen Ausgabeinformation "aus" zusammen mit der zulässigen Eingabeinformation "ein" das Spezifikationsprädikat $SPE(ein(M_0),aus(M_e))$. *Ob* der ein solcher Endzustand jemals erreicht wird, das bleibt jedoch offen.

- Das System *terminiert* genau dann, wenn es bei der Anwendung auf jede zulässige Eingabeinformation jeweils in endlicher Zeit mindestens einen wohldefinierten Endzustand M_e erreicht:

  $$ZUL(ein(M_0)) \quad\Rightarrow\quad F_0(ERR(M_0,M_e))$$

Ein Input/Output-System heißt genau dann *total korrekt*, wenn es sowohl partiell korrekt ist als auch terminiert. In diesem Fall gilt sogar:

$$ZUL(ein(M_0)) \quad\Rightarrow\quad F_0(ERR(M_0,M_e) \wedge SPE(ein(M_0),aus(M_e)))$$

Der zentrale Aspekt der prädikatenlogischen Semi-Entscheidbarkeit besteht darin, daß Input/Output-Systeme, die prädikatenlogische Entscheidungsprobleme bewältigen sollen, im allgemeinen nur partiell, aber nicht total korrekt sind. Denn ihr Terminieren läßt sich nicht generell sicherstellen.
Vgl. zu den oben vorgestellten Input/Output/Systemen einschließlich ihrer Korrektheits- und Terminierensdefinitionen ROSEN (1975), S. 126f.; ROSEN (1976), S. 183 u. 199; PNUELI (1977), S. 48; PNUELI (1979), S. 10ff. (als Hauptquelle für die oben skizzierten Zusammenhänge) u. 21; MANNA (1979), S. 392 u. 396; GABBAY (1980), S. 164f. Bei PNUELI (1979), S. 9f., und GABBAY (1980), S. 163, finden sich vor allem auch die oben verwendeten beiden zeitlogischen Operatoren. Ähnliche Operatoren stellt ABRAHAMSON (1979), S. 26, vor.
Vgl. darüber hinaus zu dem zugrundeliegenden CHURCH/ROSSER-Konzept für die Korrektheitsanalysen für Input/ Output- oder Transitionssysteme SETHI (1974), S. 673ff.; ROSEN (1975), S. 126ff.; ROSEN (1976), S. 184 u. 200ff.

77) Vgl. zu dieser Ambivalenz erfolglos abgebrochener Problemlösungsversuche HABEL (1983), S. 122; ZELEWSKI (1986a), S. 950ff.

78) Im allgemeinen ist es für ein Entscheidungsproblem nicht bekannt, ob seine Lösung positiv oder negativ ist, d.h. ob seine problemspezifische Subjugatformel tatsächlich allgemeingültig ist. Denn dies gilt es ja erst zu untersuchen. Daher stellte es eine petitio principii dar, schon während der Formulierung eines Entscheidungsproblems das Wissen über seine Lösung vorauszusetzen. Eine solche Prämisse erscheint nur für den Sonderfall plausibel, in dem Inferenzkonzepte auf Entscheidungsprobleme mit bereits bekannter Problemlösung angewendet werden. Dies kann z.B. mit der Absicht erfolgen, die Leistungsfähigkeit eines Inferenzkonzepts und seiner Implementierung in einem Automatischen Informationsverarbeitungssystem zu untersuchen. Von diesem Sonderfall wird aber hier abgesehen, weil er für die Modellierung Flexibler Fertigungssysteme keine Relevanz besitzt.

79) Dabei wird unterstellt, daß die Modellierungen jeweils die Formulierung eines Entscheidungsmodells umfassen. Die Ermittlung mindestens einer Modellösung entspricht dann der Lösung eines Entscheidungsproblems. Die Rückführung betriebswirtschaftlicher Extremierungsmodelle auf prädikatenlogische Entscheidungsprobleme ist dabei grundsätzlich immer möglich; vgl. ZELEWSKI (1989a), S. 5ff.

80) Diese günstigen Umstände werden unten dadurch charakterisiert, daß sie die Endlichkeit des logischen Inferenzraums garantieren.

81) Dies rechtfertigt die Bezugnahme QUINE's auf komplementäre Entscheidungsverfahren für den Nachweis der Allgemeingültigkeit bzw. Inkonsistenz.

82) Der logische Inferenzraum eines Beweissystems ist die Gesamtheit aller Formeln, die aus seinen Axiomen und den - gegebenenfalls hinzugefügten - Formelmengen durch Anwenden von Inferenzregeln abgeleitet werden können.

83) Die Idee einer parallelen Lösungssuche für zwei Entscheidungsprobleme, die sich wechselseitig ergänzen, findet sich bereits bei SIKLOSSY (1974), S. 814ff. Allerdings wird diese Idee dort nicht zur Überwindung der prädikatenlogischen Semi-Entscheidbarkeit, sondern nur zur Effizienzerhöhung von Inferenzprozeduren benutzt; vgl. dazu auch die Erläuterungen bei ZELEWSKI (1986a), S. 952, Fn. 1.

84) Denn die vollständige, aber erfolglose Suche nach der kontradiktorischen Formel "$\varnothing$" in einem endlichen Inferenzraum korrespondiert mit der Widerspruchsfreiheit der untersuchten Konjugatformel "$FM_q \wedge (\neg p_u)$". Diese syntaktische Widerspruchsfreiheit entspricht wiederum der semantischen Erfüllbarkeit derselben Konjugatformel.

85) Vgl. zur Existenzmöglichkeit unendlich großer Inferenzräume für prädikatenlogische Beweissysteme SMULLYAN (1968), S. 48; CLARK,K. (1979), S. 131; APT (1982), S. 852; LENAT (1984), S. 181; REINFRANK (1985b), S. 21 (unendlich viele Folgerungen aus einer endlichen Axiomen- oder Prämissenmenge); DE KLEER (1986b), S. 176; ZELEWSKI (1986a), S. 945f.; BEZEM (1988), S. 21; PIQUE (1988), S. 7.
Einen Sonderfall stellen die "mysteriösen" Formeln dar, auf die SMULLYAN (1968), S. 63, und STEGMÜLLER (1983), S. 89, hinweisen. Diese prädikatenlogischen Formeln sind gültig, können aber durch keine *endlichen* Inferenzprozesse als gültig nachgewiesen werden. Die Formeln sind nur über unendlichen Argumentemengen gültig, die sich durch keinen endlichen Inferenzprozeß überprüfen lassen. SMULLYAN (1968), S. 63, und STEGMÜLLER (1983), S. 89, geben hierfür konkrete Beispiele.

86) Dies läßt sich durch Verweis auf eine petitio principii begründen, falls nicht vom Sonderfall von Entscheidungsproblemen mit a priori bekannter Problemlösung ausgegangen wird.

87) Dies äußert sich in der Unmöglichkeit, auf *jede* prädikatenlogische Formel sowohl ein positives als auch ein negatives Entscheidungsverfahren derart parallel anzuwenden, daß eines von ihnen nach endlicher Zeit mit dem Nachweis endet, die untersuchte Formel sei allgemeingültig bzw. nicht allgemeingültig. Dies folgt aus der bereits dargelegten Nichtexistenz negativer Entscheidungsverfahren für das allgemeine Gültigkeitsproblem der Prädikatenlogik. Vgl. auch zu der Unmöglichkeit der o.a. Konstruktion zweier dualer Entscheidungsformeln für *beliebige* prädikatenlogische Formeln BOOLOS (1980), S. 121 u. 143.

88) Ein anderer Einwand wäre vorstellbar, der alle hier vorgetragenen Argumente radikal in Zweifel zieht: Sie könnten auf praktische Entscheidungsprobleme nicht angewendet werden, weil die "harte" Prädikatenlogik den Erfordernissen der betriebswirtschaftlichen Argumentationspraxis nicht angemessen sei. Diese Vorhaltung wird hier aus zwei Gründen nicht weiterverfolgt. Erstens wurde als eine Basisprämisse dieser Arbeit vorausgesetzt, Modellierungen Flexibler Fertigungssysteme auf eine prädikatenlogische Beschreibung der Modellierungsobjekte zu stützen. Damit erfolgte bereits eine Entscheidung zugunsten der "harten" prädikatenlogischen Folgerungszusammenhänge. Verweise auf eine - angeblich - abweichende Argumentationspraxis bleiben daher hier irrelevant. Zweitens hat der Verf. an früherer Stelle seine Skepsis gegenüber "weichen" Argumentationskriterien vorgetragen, sofern sie im *Begründungs*zusammenhang angewendet werden sollen. Bei der Überprüfung von Gültigkeitsbehauptungen, die sich auf prädikatenlogisch fundierte Objektmodelle erstrecken, handelt es sich um einen solchen Begründungszusammenhang. Daher ist es hier unerheblich, daß etwa bei der Rechtfertigung von Prämissensetzungen durchaus "weiche" Argumentationskriterien akzeptiert werden können - oder sogar müssen.

89) Dies ist die prädikatenlogische Reformulierung des MÜNCHHAUSEN-Trilemmas. Da dies bereits ausführlich behandelt wurde, verzichtet der Verf. hier, es im prädikatenlogischen Kontext nochmals zu rechtfertigen. Allerdings ist darauf hinzuweisen, daß das Trilemma hier zu einem Dilemma verkürzt wurde. Denn prädikatenlogische Untersuchungen von Gültigkeitsbehauptungen beruhen stets auf syntaktischen Beweissystemen. In solchen Beweissystemen sind aber nur endliche Beweise zulässig, die von dogmatisch vorausgesetzten Axiomen ausgehen und in den zu beweisenden Formeln enden. Zirkuläre "Beweis"führungen sind in solchen prädikatenlogischen Beweissystemen überhaupt nicht definiert. Daher entfällt der dritte Aspekt des MÜNCHHAUSEN-Trilemmas, der sich auf solche Argumentationszirkel bezieht.

90) Strenggenommen handelt es sich nur um einen Teil des stützenden Hintergrundwissens. Hinzu kommt als Bestandteil des Hintergrundwissens ein Beweissystem mit seinen Axiomen und Inferenzregeln. Erst dieser weitere Wissensbestandteil legt in operationaler Weise fest, was es bedeuten soll, die behauptete Gültigkeit einer Formel sei "nachgewiesen" oder "zurückgewiesen" worden. Diese inferentielle Komponente des Hintergrundwissens wird im folgenden nicht weiter explizit erwähnt, sondern stets implizit vorausgesetzt. Hierfür spricht, daß in Untersuchungen von Gültigkeitsbehauptungen die jeweils zugrundegelegten Beweissysteme nicht strittig sind.

91) Die Formeln des untersuchungsstützenden Hintergrundwissens entsprechen in TOULMIN's Argumentationstheorie den Stützungsargumenten (backings).

92) Dies entspricht dem früher erhobenen Postulat der möglichst weitgehenden Prämissenoffenlegung. Zugleich wird verdeutlicht, daß die Darlegung der Prämissen unvollständig ist. Denn die inferentielle Komponente des Hintergrundwissens wird in der Formelmenge FM_q nicht offengelegt. Allerdings wird der Sachverhalt, daß im Hintergrundwissen ein bestimmtes Beweissystem BS mit seinen Axiomen und Inferenzregeln vorausgesetzt wird, durch das Ableitungssymbol "$\vdash_{BS}$" angezeigt. Sobald die zu untersuchende, semantisch formulierte Allgemeingültigkeitsbehauptung "$\models (FM_q \rightarrow p_u)$" in die gleichwertige syntaktische Beweisbarkeitsbehauptung "$\vdash_{BS}(FM_q \rightarrow p_u)$" transformiert worden ist, wird die Prämisse eines bestimmten Beweissystems BS offensichtlich. Dadurch wird zwar das jeweils benutzte Beweissystem BS noch nicht inhaltlich beschrieben. Aber immerhin wird offengelegt, daß die Argumentationen auch von der Präsupposition eines solchen Beweissystems abhängen. Dies genügt dem Grundsatz der kontrollierten Explizitheit. Ihm zufolge wird zwar die Prämisse, *ein* Beweissystem vorauszusetzen, expliziert. Da es aber im allgemeinen für betriebswirtschaftliche Argumentationen unerheblich ist, *welches* prädikatenlogische Beweissystem konkret benutzt wird, kann auf dessen explizite Spezifizierung verzichtet werden.

93) Dies wurde bei der Darlegung der prädikatenlogischen Semi-Entscheidbarkeit bereits aufgezeigt.

94) Umgekehrt gilt: Wer in Anspruch nimmt, die Gültigkeit einer einzelnen Formel behaupten zu können, ohne zugleich eine - implizite oder explizite - Allgemeingültigkeitsbehauptung vertreten zu müssen, der muß jedes Hintergrundwissen negieren. Dies ist entweder wirklichkeitsfremd, oder es beruht auf einer dogmatischen Voraussetzung der Gültigkeit der zu untersuchenden Formel. Erstes grenzt der Verf. im Interesse eines realitätsnahen Modellierungskonzepts aus. Zweites wurde schon an früherer Stelle ausgeschlossen. Folglich bleibt jede Argumentation im Kontext von Entscheidungsproblemen - insbesonderen auch jede praxisbezogene Argumentation - darauf verwiesen, zumindest implizit Allgemeingültigkeitsbehauptungen vertreten zu müssen.

95) Diese umgangssprachliche Formulierung umschreibt lediglich die prädikatenlogische Definition einer semantischen Schlußfolgerung: Die betrachtete Referenzformel ist gültig für alle Modelle (d.h. für alle Paare aus Interpretationen I_r und Variablenbelegungen V_c), unter denen das Hintergrundwissen der Formelmenge FM_q gültig ist.

96) In diesem Sinne identifiziert BREWKA (1989), S. 95, die Semi-Entscheidbarkeit eines logischen Konzepts mit seiner Vollständigkeit und Korrektheit.

4.2.2.3.3 Implementierungstechnische Defizite

Eine letzte Quelle für Leistungsschranken der logischen Programmierung beruht nicht mehr auf generellen Charakteristika aller prädikatenlogischen Inferenzkonzepte. Statt dessen gründet sie in der speziellen Art, in der das kombinierte Unifizierungs- und Resolutionskonzept durch die[1] Kontrollstruktur der Programmiersprache PROLOG implementiert ist. Denn die Kontrollstrukturen aller etablierten PROLOG-Dialekte erweisen sich als mangelhaft.

Die implementierungstechnischen Defizite führen dazu, daß die operationale Semantik der PROLOG-Kontrollstruktur eine echte Teilmenge der deklarativen Semantik von prädikatenlogischen Formelsystemen ist[2]. Daher läßt es sich nicht ausschließen, daß tatsächlich existierende Resolutions- oder Unifizierungsschritte von der PROLOG-Kontrollstruktur übersehen werden[3]. Falls die hierdurch unterlassenen Inferenzprozeduren zur korrekten Erkenntnis einer tatsächlich existierenden Problemlösung geführt hätten, besteht die Gefahr, daß diese Problemlösung aus implementierungstechnischen Gründen nicht zuverlässig erkannt wird. Hierfür kommen im wesentlichen drei Gründe in Betracht.

Erstens werden im allgemeinen obere Schranken für den maximal zulässigen Ressourcenverzehr einer Inferenzprozedur eingerichtet, bei deren Erreichen die Prozedurausführung automatisch abgebrochen wird. Daher ist es möglich, daß die tatsächlich existierende positive Lösung eines Entscheidungsproblems "$\models (FM_q \rightarrow p_u)$?" nicht erkannt, sondern der Lösungsversuch erfolglos abgebrochen wird. Daher kann nicht mehr sichergestellt werden, daß die tatsächliche Allgemeingültigkeit der Subjugatformel "$FM_q \rightarrow p_u$" als deren Beweisbarkeit erkannt wird. Folglich erweist sich das implementierte Inferenzkonzept als unvollständig. Diese ressourcenbedingte Unvollständigkeit besteht für jede konkret vorgegebene Ressourcenschranke: Stets lassen sich Entscheidungsprobleme formulieren, die zwar an sich eine positive Lösung besitzen, deren Lösungsnachweis im Inferenzraum aber mehr als die verfügbaren Ressourcen benötigen würde[4].

Zweitens kann die spezielle Kontrollstruktur der Programmiersprache PROLOG dazu führen, daß die Inferenzprozedur bei Entscheidungsproblemen mit unendlichen Inferenzräumen in der "falschen" Sektion des Inferenzraums sucht[5]. Dann ist es möglich, daß sich die Lösungssuche in einem unendlichen Teilraum verfängt[6], obwohl das untersuchte Entscheidungsproblem tatsächlich eine positive Lösung besitzt[7]. In einem solchen Fall muß die Inferenzprozedur nicht nur bei einer bestimmten vorgegebenen, sondern bei *jeder beliebigen* endlichen Ressourcenschranke erfolglos abgebrochen werden. Da auch hierbei eine tatsächlich existierende positive Problemlösung nicht erkannt wird, liegt erneut eine Verletzung der prädikatenlogischen Vollständigkeit vor. Zugleich besteht hierin ein dritter Grund für das Auftreten von Endlosschleifen bei der Ausführung von PROLOG-Programmen. Insgesamt kommen daher für die Existenzmöglichkeit von Endlosschleifen mit nicht-terminierenden Programmausführungen drei Quellen in Betracht[8]: die prädikatenlogische Unentscheidbarkeit und Semi-Entscheidbarkeit sowie Implementierungsdefizite der PROLOG-Kontrollstruktur[9].

Drittens wird im allgemeinen jeder erfolglose Prozedurabbruch durch das "negation by failure"-Prinzip als eine "erfolgreiche" Problemlösung gedeutet: Die PROLOG-Kontrollstruktur unterstellt als Lösungssurrogat[10], daß die Lösung eines Entscheidungsproblems negativ ist, falls sein Lösungsversuch scheitert. Der Prozedurabbruch kann aber auch bei einem Entscheidungsproblem mit positiver Lösung eintreten, weil die vorgegebenen Ressourcenschranken erreicht worden sind. Daher kann das "negation by failure"-Prinzip zum Ausweis falscher Problemlösungen führen[11]. Folglich verletzt das derart implementierte Inferenzkonzept auch die prädikatenlogische Korrektheit.

Die voranstehend aufgezeigten implementierungstechnischen Mängel der gegenwärtig dominierenden PROLOG-Kontrollstrukturen zeigen, daß die theoretische Inferenzmächtigkeit der Prädikatenlogik - ihre operationale Adäquanz - seitens der Programmiersprache PROLOG nicht umfassend realisiert wird. Statt dessen läßt sich die theoretische Vollständigkeit und Korrektheit prädikatenlogischer Beweissysteme in praktischen Systemimplementierungen auf PROLOG-Basis nicht mehr garantieren[12)]. Diese implementierungsbedingten Defizite treten als Spezifika der Programmiersprache PROLOG neben die weiter oben behandelten allgemeinen prädikatenlogischen Inferenzdefizite der Unentscheidbarkeit und Semi-Entscheidbarkeit.

Die logischen und die implementierungstechnischen Einschränkungen der prädikatenlogischen Inferenzmächtigkeit lassen sich wie folgt zusammenfassen[13)]:

- Ineffektivität: Falls ein unentscheidbares Entscheidungsproblem untersucht wird (dessen Unentscheidbarkeit dem Untersuchenden aber nicht bekannt ist), dann kann die Suche nach der Problemlösung grundsätzlich nicht terminieren. Der Lösungsversuch muß nach endlichem Ressourceneinsatz erfolglos abgebrochen werden. Aus dem gescheiterten Lösungsversuch kann aber kein gesichertes Wissen über den Lösungscharakter des untersuchten Entscheidungsproblems gewonnen werden. Es bleibt unbekannt, ob es sich entweder um ein entscheidbares oder aber um ein unentscheidbares Entscheidungsproblem handelt. Diese Ergebnisoffenheit wird als Ineffektivität der Lösungssuche bezeichnet.
- Unvollständigkeit: Wenn in einem Entscheidungsproblem die Allgemeingültigkeit einer Subjugatformel behauptet wird, die tatsächlich nicht allgemeingültig ist, dann ist es möglich, daß die Suche nach der Problemlösung nicht terminiert. Falls die Lösungssuche nicht terminiert, muß sie nach endlichem Ressourceneinsatz erfolglos abgebrochen werden. Gleiches gilt für ein Entscheidungsproblem, dessen Allgemeingültigkeitsbehauptung zwar an sich zutrifft, aber innerhalb der real verfügbaren Ressourcen von der implementierten Kontrollstruktur nicht erkannt wird. In beiden Fällen wird die tatsächlich existierende negative bzw. positive Lösung des Entscheidungsproblems nicht erkannt.
- Inkorrektheit: Um die Ergebnisoffenheit gescheiterter Lösungsversuche zu beseitigen, wird zumeist das Lösungssurrogat des "negation by failure"-Prinzips eingeführt. Daher kann das Lösungssurrogat durchaus einen Fehlschluß bewirken. Dies ist genau dann der Fall, wenn ein Entscheidungsproblem tatsächlich eine positive Lösung besitzt, die jedoch innerhalb der realen Ressourcenbeschränkungen nicht gefunden wurde. Dann verleitet das Lösungssurrogat zu dem Fehlschluß, das Entscheidungsproblem besäße die negative Lösung.

Hinzu kommen Variationen der Ergebniszuverlässigkeit und der Ausführungseffizienz von Inferenzprozeduren, die sensitiv auf die jeweils gewählte Formulierung eines Entscheidungsproblems reagieren können[14)].

Eine ergebnisbezogene Formulierungssensitivität kann vor allem dann eintreten, wenn PROLOG-Programme zwecks Begrenzung von Inferenzprozeduren mit Hilfe des speziellen Kontrolloperators "cut"[15)] formuliert werden. Dann wird die Lösungssuche nach dem Auffinden einer ersten Problemlösung automatisch abgebrochen. Falls für ein untersuchtes Entscheidungsproblem mehrere Lösungen existieren, hängt es von der jeweils willkürlichen Anordnung der Formeln in einem PROLOG-Programm ab, welche Problemlösung tatsächlich erkannt wird. Die Ergebnissensitivität gilt ebenfalls, wenn Ressourcenbeschränkungen existieren, nach deren Erreichen ein Lösungsversuch beendet werden muß. Dann kann die willkürliche Formelanordnung im zugrundeliegenden PROLOG-Programm abermals das Inferenzergebnis beeinflussen. Diesmal hängt von ihr ab, ob eine tatsächlich existierende Problemlösung überhaupt erkannt wird. Denn die Länge erfolgreicher Inferenzprozeduren kann von der jeweils gewählten Formelanordnung beeinflußt werden[16)]. Falls einige dieser Prozeduren die Ressourcenbeschränkungen übersteigen, andere dagegen nicht, kann die willkürlich gewählte Formelanordnung darüber entscheiden, ob der jeweils unternommene Inferenzprozeß erfolgreich endet oder aber erfolglos abgebrochen werden muß. Im ersten Fall würde die tatsächlich existierende Problemlösung erkannt, im zweiten aber nicht. Aus der Perspektive der deklarativen PROLOG-Semantik müßte dagegen die Art der Formelanordnung für die Erkenntnis der positiven oder negativen Lösbarkeit eines Entscheidungsproblems im Prinzip irrelevant sein[17].

Hinzu kommen kann eine weitere ergebnisbezogene Formulierungssensitivität, falls drei Voraussetzungen erfüllt sind. Erstens muß es sich um entscheidbare Entscheidungsprobleme handeln. Zweitens muß der Inferenzraum der Problemformulierung endlich sein. Drittens müssen die real verfügbaren Ressourcen ausreichen, um den endlichen Inferenzraum vollständig zu durchsuchen. Dann hängen die Lösungsversuche für ein Entscheidungsproblem hinsichtlich ihres Ergebnisses von der zufälligen Entscheidung ab, das Problem durch die Allgemeingültigkeitsbehauptung einer Subjugatformel "$FM_q \rightarrow p_u$" oder aber durch die Erfüllbarkeitsbehauptung ihres Negats - der Konjugatsformel "$FM_q \wedge (\neg p_u)$" - zu formulieren. Genau eine von diesen beiden kontradiktorischen Formulierungen führt mit Sicherheit zur Erkenntnis der korrekten Problemlösung. Es ist aber im Zeitpunkt der Problemformulierung unbekannt, welche der beiden alternativen problemspezifizierenden Formeln zur korrekten Problemlösung führt. Daher bleibt es dem Zufall oder dem - rational nicht belegbaren - "Instinkt" des Problemformulierers überlassen, mit welcher Formelvariante der Versuch einer Problemlösung gestartet wird. Je nachdem, welche Variante zugrundegelegt wird, kann das Lösungsverfahren erfolgreich terminieren oder aber einen erfolglosen Abbruch infolge realer Ressourcenbeschränkungen erzwingen.

Eine effizienzbezogene Formulierungssensitivität resultiert dagegen oftmals aus dem Umstand, daß die Anzahl der Inferenzschritte, die von einer Kontrollstruktur zur korrekten Erkenntnis der Lösung eines Entscheidungsproblems ausgeführt werden, erheblich mit der Anordnung der problembeschreibenden Formeln in einem PROLOG-Programm variieren kann. Dies resultiert aus der Tatsache, daß in der Kontrollstruktur von PROLOG-Programmen ein Backtracking-Konzept mit fester Tiefensuche implementiert ist. Es hängt von der zufälligen Anordnung der Formeln in einer PROLOG-basierten Problembeschreibung ab, wie rasch diese Suchstrategie zu einer erfolgreichen Problemlösung gelangt, falls die Lösungssuche überhaupt terminiert.

Darüber hinaus kann die Effizienz der Problemlösung auch dann von der Problemformulierung abhängen, wenn das bisher unterstellte einphasige Inferenzkonzept durch eine bedingt zweiphasige Lösungssuche ersetzt wird[18]. Dabei stimmt die erste Lösungsphase mit dem konventionellen Inferenzkonzept überein, für ein Entscheidungsproblem "$\models (FM_q \rightarrow p_u)$?" die Allgemeingültigkeitsbehauptung der Subjugatformel "$FM_q \rightarrow p_u$" zu untersuchen. Falls dieser Lösungsversuch erfolglos abgebrochen werden muß, wird in einer zweiten Phase die kontradiktorische Erfüllbarkeitsbehauptung der Konjugatformel "$FM_q \wedge (\neg p_u)$" getestet. Dann ist es möglich - aber bei realen Ressourcenbeschränkungen keineswegs notwendig, im zweiten Lösungsversuch die korrekte Problemlösung erfolgreich zu entdecken. Dies ist immer dann der Fall, wenn erstens das untersuchte Entscheidungsproblem tatsächlich eine negative Lösung besitzt und wenn zweitens die positive Lösung des kontradiktorischen Entscheidungsproblems mit der Negatformel innerhalb der real beschränkten Ressourcen tatsächlich erkannt wird. Wäre dagegen diese kontradiktorische Problemformulierung zufällig oder instinktiv von vornherein ausgewählt worden, hätte sich in diesem Spezialfall der zusätzliche Ressourceneinsatz für den zweiten Lösungsversuch einsparen lassen. Daher kann auch die Effizienz von Problemlösungen sensitiv auf die Problemformulierung reagieren.

Den logischen und implementierungstechnischen Leistungsschranken, denen PROLOG-basierte Inferenzprozeduren in prädikatenlogisch formulierten Objektmodellen unterworfen sind, stehen aber auch gravierende Vorzüge gegenüber[19].

Erstens gilt das "faute de mieux"-Argument, daß ein besserstellendes Modellierungs- und Inferenzkonzept bislang unbekannt ist. Denn einerseits kann für Objektmodellierungen, welche die prädikatenlogische Ausdruckskraft ausnutzen, grundsätzlich kein Inferenzkonzept existieren, das die logischen Inferenzdefizite nicht aufweist. Dies folgt aus dem Umstand, daß die Defizite der Unentscheidbarkeit und Semi-Entscheidbarkeit für *alle* Inferenzkonzepte der uneingeschränkten Prädikatenlogik zutreffen. Konzeptionelle Inferenzverbesserungen ließen sich also nur zu Lasten der Modellierungsmächtigkeit erzielen, indem auf das volle Ausdruckspotential der Prädikatenlogik verzichtet wird. Dies widerspricht aber der früher dargelegten Modellierungsprämisse, vom Primat der Formulierungsmächtigkeit auszugehen. Andererseits steht bislang keine Implementierung eines prädikatenlogischen Inferenzkonzepts zur Verfügung, das die Implementierungsmängel der PROLOG-Kontrollstruktur vermeidet, ohne deutlichen Effizienzeinbußen bei der Ausführung von Inferenzprozeduren ausgesetzt zu sein[20)].

Zweitens besitzen PROLOG-basierte Inferenzprozeduren für die Realisierung von Termauswertungen die beiden bemerkenswerten Eigenschaften der schwachen und starken Konstruktivität. Ihr Ausgangspunkt ist das Entscheidungsproblem des Signaturkonzepts. Es besteht in der Aufgabe festzustellen, ob für die Terme in einer uninterpretierten Formel mindestens eine Auswertungsfunktion existiert, die diese Terme auf formale Objekte aus einer zugeordneten SIG-Algebra abbildet. Diesem Existenzproblem ist das Konstruktionsproblem zugeordnet, die konkrete Gestalt dieser Auswertungsfunktion - oder zumindest das Ergebnis ihrer Anwendung auf die vorliegenden Terme - konkret zu bestimmen, falls das Existenzproblem mindestens eine positive Lösung besitzt[21)].

Falls das Existenzproblem eine positive Lösung besitzt, so garantiert dies im allgemeinen noch nicht die positive Lösung des zugeordneten Konstruktionsproblems. Das kombinierte Unifizierungs- und Resolutionskonzept besitzt jedoch eine grundsätzlich konstruktive Natur. Jede positive Lösung des Existenzproblems enthält zugleich eine positive Lösung des schwachen Konstruktionsproblems: Wenn durch die Anwendung dieses Inferenzkonzepts eine positive Lösung des Existenzproblems gelingt, so kann aus der Beweisführung zugleich mindestens eine Ersetzung aller Terme aus der jeweils untersuchten Formel durch konkret bestimmte Auswertung formale Objekte extrahiert werden. Diese Termersetzung determiniert zwar nicht notwendig eine bestimmte Auswertungsfunktion, aber es legt die Auswertungsergebnisse aller Auswertungsfunktionen fest, welche die erkannte positive Lösung des Existenzproblems hervorbringen können. Daher wird durch das prädikatenlogische Inferenzkonzept auf jeden Fall das schwache Konstruktionsproblem gelöst. Diese immanente Konstruktivität bleibt auch in den Konzeptimplementierungen durch die Programmiersprache PROLOG erhalten[22)]. Im Signaturkonzept war dagegen noch nicht einmal eine solche schwache Lösung des Existenzproblems garantiert.

Darüber hinaus ermöglicht die Kontrollstruktur der Programmiersprache PROLOG sogar die starke Lösung des Konstruktionsproblems: Falls ein Existenzproblem überhaupt mindestens eine positive Lösung besitzt, so ist es - zumindest theoretisch[23)] - möglich, für *alle* positiven Lösungen die zugehörigen Termersetzungen durch formale Objekte zu ermitteln[24)]. Die Ursache dieser starken Konstruktivität liegt im Backtracking-Konzept[25)] der PROLOG-Kontrollstruktur[26].

Anmerkungen zum Kapitel:

1) Der Einfachheit halber wird anschließend von "der" Kontrollstruktur der Programmiersprache PROLOG gesprochen. Tatsächlich existiert aber eine Vielzahl verschiedener Kontrollstrukturen, die jeweils in verschiedenen PROLOG-Dialekten implementiert wurden. Von deren Modifizierungen wird hier abstrahiert, sofern sie für die grundsätzliche Betrachtung der Defizite von PROLOG-Implementierungen keine besondere Bedeutung besitzen.

2) Sofern eine PROLOG-Implementierung korrekt ist, also keine logisch fehlerhaften "Beweise" hervorzubringen vermag, ist die operationale Semantik immer eine - echte oder unechte - Teilmenge der deklarativen Semantik. Tatsächlich stellt die operationale PROLOG-Semantik aber sogar eine echte Teilmenge der deklarativen prädikatenlogischen Semantik dar. Denn die nachfolgend erläuterten Implementierungsdefizite führen dazu, daß seitens der PROLOG-Kontrollstruktur einige der prädikatenlogisch zulässigen Inferenzprozeduren ausgeschlossen werden. Daher trifft das mitunter behauptete Zusammenfallen von operationaler und deklarativer PROLOG-Semantik unter realen Bedingungen im allgemeinen nicht zu. Vgl. zu solchen Äquivalenzbehauptungen KOWALSKI (1974), S. 572 (mittelbar); CLARK,K. (1979), S. 131; ZELEWSKI (1986a), S. 193f. (distanziert).

3) Vgl. zu solchen Schwierigkeiten aller real implementierten PROLOG-Kontrollstrukturen CLOCKSIN (1981), S. 223f.; ELCOCK (1983a), S. 115; WALKER,A. (1983), S. 526f.; BOBROW (1984), S. 141; LEVI,G. (1986), S. 404 u. 406f.; WESTPHAL (1986), S. 236; ESTER (1989), S. 38; vgl. auch DELAHAYE (1987), S. 135, 137, 141 u. 165f., der mehrere Implementierungsvarianten für das Resolutionskonzept erörtert, die jeweils zulässige Inferenzprozeduren unterbinden und dadurch zu eingeschränkten operationalen Semantiken führen können.

4) Denn das terminierende Verhalten für Entscheidungsprobleme mit positiver Lösung garantiert nur den Lösungsausweis nach endlich vielen Lösungsschritten. Wie groß diese endliche Schrittanzahl ist, ist für *beliebige* Entscheidungsprobleme ex ante unbestimmt. Daher werden für jede reale Begrenzung der Ausführungsressourcen "pathologische" Entscheidungsprobleme existieren: Sie zeichnen sich dadurch aus, daß Lösungsprozeduren für sie infolge positiver Problemlösbarkeit grundsätzlich terminieren müssen, jedoch innerhalb der vorgegebenen Ressourcenbeschränkungen tatsächlich nicht zum erfolgreichen Abschluß geführt werden können. Dann besteht wieder das bereits oben dargelegte Problem, daß sich erfolglose Prozedurabbrüche nicht eindeutig interpretieren lassen. Solche pathologischen Entscheidungsprobleme sind daher theoretisch entscheidbar, aber *praktisch unentscheidbar*. Es könnte zwar bezweifelt werden, daß für *jede* obere Ressourcenschranke Entscheidungsprobleme existieren, deren Lösungsversuche innerhalb dieser Grenzziehung nicht terminieren können. Dieser Einwand läßt sich aber widerlegen. Denn es existieren *theoretisch unentscheidbare* Probleme, die durch keinen - noch so großen - endlichen Ressourceneinsatz bewältigt werden können. Im Gegensatz zu den o.a. Entscheidungsproblemen mit entweder positiver oder negativer Lösung besitzen diese unentscheidbaren Probleme im Endlichen überhaupt keine Lösung. Das bekannteste unentscheidbare Entscheidungsproblem stellt das Halteproblem für TURING-Automaten dar. Auf solche unentscheidbaren Probleme wurde bereits hingewiesen.

5) Daher kann eine implementierte PROLOG-Kontrollstruktur nur dann vollständig und korrekt sein, wenn die Inferenzräume der jeweils untersuchten Entscheidungsprobleme endlich sind; vgl. DELAHAYE (1987), S. 141 u. 144. Die Vollständigkeitsverletzung bei unendlichen Inferenzräumen wird nachfolgend explizit erläutert. Der Korrektheitsverstoß bei unendlichen Inferenzräumen ergibt sich mittelbar, wenn der erfolglose Abbruch einer Inferenzprozedur durch das "negation by failure"-Prinzip überlagert wird. Darauf wird später noch näher eingegangen.

6) Vgl. LEVI,G. (1986), S. 404; DELAHAYE (1987), S. 135 u. 165f.

7) Daher ist es möglich, daß der Inferenzprozeß einer PROLOG-Programmausführung immer wieder mit erfolglosen Resolutions- und Unifizierungsschritten an einem tatsächlich existierenden erfolgreichen Beweis für die Subjugatformel eines Entscheidungsproblems vorbeiführt; vgl. ZELEWSKI (1986a), S. 946f.

8) Vgl. zur Existenzmöglichkeit solcher Endlosschleifen bei der Ausführung von PROLOG-Programmen oder - genereller - bei Implementierungen des Unifizierungs- und Resolutionskonzepts ROBINSON,J. (1965), S. 37; RICHTER,M.M. (1978), S. 195; CLOCKSIN (1981), S. 224; AIDA (1983), S. 90; WALKER,A. (1983), S. 527; DELAHAYE (1987), S. 34 (i.V.m. S. 33), 126, 135 u. 162ff.; KOWALSKI (1987a), S. 138; KLEINE BÜNING (1988a), S. 59; CORDES (1988), S. 46ff. (implizit); ESTER (1989), S. 39.
Das Unifizierungs- und Resolutionskonzept ist in seinen vollständigen und korrekten Reinformen so angelegt, daß bei der Konzeptanwendung entweder Endlosschleifen niemals auftreten können (z.B. bei der Breitensuche) oder daß Mechanismen (wie etwa der "occur-check") vorgehalten werden, durch die sich solche Schleifen erkennen und wieder verlassen lassen. Implementierungen dieses Inferenzkonzepts können sich dagegen in nicht-terminierenden Endlosschleifen verfangen, wenn sie erstens nicht auf den inhärent endlosschleifenfreien Konzeptformen beruhen und zweitens nicht alle erforderlichen Mechanismen zum Erkennen und Verlassen von Endlosschleifen realisieren. Vgl. dazu die o.a. Quellen; vgl. darin insbesondere die Erörterung des fehlenden "occur-checks".

9) Dabei besitzen die drei Ursachen nicht-terminierender Programmausführungen unterschiedliche praktische Relevanz. Da es sich bei der Unentscheidbarkeit und der Semi-Entscheidbarkeit um zwei allgemeine Eigenschaften der

Prädikatenlogik handelt, gelten sie für jedes prädikatenlogisch fundierte Beweissystem. Folglich treffen sie auch auf das kombinierte Unifizierungs- und Resolutionskonzept zu. A fortiori wirken sie sich auf die Kontrollstruktur *jedes* PROLOG-Dialekts aus. Sie können daher von PROLOG-Implementierungen prädikatenlogischer Modellierungskonzepte grundsätzlich nicht vermieden werden. Aber die Unentscheidbarkeit betrifft nur artifizielle Problemkonstruktionen ohne praktische Bedeutung; vgl. dazu die Anmerkung zu unentscheidbaren Problemen des GÖDEL-Typs. Die Semi-Entscheidbarkeit kann dagegen auch für praktische Entscheidungsprobleme im Rahmen konkreter Modellierungen von Realitätsausschnitten eine Rolle spielen. Denn das semi-entscheidbare Problem, die Allgemeingültigkeit einer prädikatenlogischen Formel zu entscheiden, kann sich in vielfachen Modellierungskontexten stellen. Die dritte Ursache nicht-terminierender Programmausführungen, die Implementierungsdefizite prozeduraler Kontrollstrukturen, besitzen sogar universelle praktische Relevanz in dem Sinne, daß sie sich - unabhängig vom jeweils untersuchten Problemtyp - jederzeit auswirken können.

10) Das Lösungssurrogat stellt *keine* logisch zulässige Schlußfolgerung dar, da laut Voraussetzung der prädikatenlogische Inferenzprozeß bei der Lösungssuche erfolglos eingestellt wurde.

11) Dies wurde in einer früheren Anmerkung ausführlicher dargelegt. Allerdings ist es wichtig zu betonen, daß solche Fehlschlüsse kein Defizit der Prädikatenlogik selbst darstellen, sondern erst aus der immanenten Inkorrektheit des Lösungssurrogats folgen. Die Prädikatenlogik selbst ist dagegen - wie schon oben ausgeführt wurde - als korrekt nachgewiesen. Einem Mißverständnis unterliegt daher z.B. ELCOCK (1983a), S. 114, mit seiner Behauptung, das (Unifizierungs- und) Resolutionskonzept verletze die Korrektheit der Prädikatenlogik. Denn auch dieses Konzept ist als vollständig und korrekt nachgewiesen. Statt dessen wird das Resolutionskonzept gewöhnlich durch das "negation by failure"-Prinzip überlagert. Erst diese Superposition verursacht den festgestellten Korrektheitsverstoß.

12) Diese Einschränkung des Inferenzpotentials der Prädikatenlogik 1. Stufe muß jedoch in Kauf genommen werden. Denn derzeit stehen keine alternativen Implementierungen der Prädikatenlogik zur Verfügung, die ähnlich ausgereift wie die Programmiersprache PROLOG sind und zugleich deren Inferenzmängel überwinden. Darauf wird später im Sinne des schon früher erwähnten "faute de mieux"-Kriteriums praktischer Rationalität zurückgekommen.

13) Vgl. dazu auch ansatzweise ZELEWSKI (1986a), S. 948f.

14) Vgl. zur Sensitivität der Inferenzergebnisse und der Inferenzeffizienz gegenüber der Formulierung von Entscheidungsproblemen in PROLOG-Programmen APT (1982), S. 852; AIDA (1983), S. 87; SHAPIRO,E. (1983), S. 26; BEETZ (1986), S. 26; VARSEK (1986), S. 216; ZELEWSKI (1986a), S. 933; DELAHAYE (1987), S. 162 u. 164f.; CORDES (1988), S. 48; RICHTER,M.M. (1988), S. 20; KLEINE BÜNING (1988a), S. 59; BEZEM (1988), S. 21.

15) Vgl. PROLOG (o.J.), S. 58ff. u. 149; TOWNSEND,C. (1987), S. 75ff. u. 80f.; KINNEBROCK (1988), S. 60f.; CORDES (1988), S. 66ff. u. 76f.

16) Vgl. dazu die Anmerkung zur formulierungsabhängigen Ausführungseffizienz von PROLOG-Programmen aufgrund ihrer starren Tiefensuche nach dem Backtracking-Konzept.

17) Damit ist gemeint, daß das Inferenzergebnis für alle äquivalenten Formulierungen desselben Entscheidungsproblems durch die zugrundeliegende Beweistheorie eindeutig bestimmt ist. Dies bedeutet unter anderem, daß die Formelanordnung innerhalb eines PROLOG-Programms aus der Perspektive der zugrundeliegenden prädikatenlogischen Beweistheorie zufällig und irrelevant ist. Denn alle Formeln eines solchen Programms sind implizit konjunktiv verknüpft. Der Wahrheitswert eines Konjugats hängt aber infolge Kommutativität der logischen Konjunktion nicht von der Reihenfolge der Konjugatskomponenten ab.

18) Vgl. ZELEWSKI (1986a), S. 949 u. 952. Die hier angewandte Inferenzidee lag bereits der parallelen Lösungssuche zugrunde, die hinsichtlich der Überwindung der Semi-Entscheidbarkeit erörtert wurde. Hier wird lediglich die parallele Ausführung zweier paralleler Inferenzprozeduren in deren konsekutive Ausführung transformiert.

19) Darüber hinaus müßte die Praxisrelevanz der oben thematisierten Leistungsschranken anhand umfangreicher empirischer Tests mit implementierten prädikatenlogischen Objektmodellen überprüft werden. Eine solche intensive empirische Ausrichtung entspricht jedoch nicht den grundsätzlichen Intentionen dieser Arbeit, die eingangs dargelegt wurden. Schließlich handelt es sich zum Teil um Mängel, die nicht durch die prädikatenlogische Basis der hier gewählten Modellierungskonzeption verursacht werden. Vielmehr beruhen sie auf Defiziten der speziellen PROLOG-gestützten Implementierung des prädikatenlogischen Unifizierungs- und Resolutionskonzepts. Solche informationstechnischen Implementierungsaspekte und deren Überwindungschancen liegen ebenfalls außerhalb des Erkenntnisinteresses dieser Arbeit.

20) Dies wird mittelbar durch eine Äußerung von BOSSI (1989), S. 96, unterstrichen. Ihr zufolge zielen die speziellen Kontrollstrukturen der Programmiersprache PROLOG darauf ab, eine effiziente Sprachimplementierung zu verwirklichen. Aus dem Umkehrschluß folgt: Wird auf die problematischen Kontrollstrukturen verzichtet, so drohen gravierende Effizienzmängel bei der Implementierung.

21) Existenz- und Konstruktionsprobleme sowie ihre Bedeutung für das Signaturkonzept wurden an früherer Stelle ausführlich behandelt.

22) Vgl. zur Konstruktivität von Existenzbeweisen, die in der Programmiersprache PROLOG ausgeführt werden, LEVI,G. (1986), S. 399f. u. 407; ZELEWSKI (1986a), S. 194. Näheres zu speziellen Techniken, im Rahmen des Unifizierungs- und Resolutionskonzepts konstruktive Existenzbeweise (und Fragebeantwortungen) zu verwirklichen, findet sich bei ITZINGER (1976), S. 86ff.; RAPHAEL (1976), S. 132ff.; DILGER (1979), S. 277f.; WALKER,A. (1983), S. 526ff.; BEETZ (1986), S. 25; DELAHAYE (1987), S. 144:
Darüber hinaus läßt sich zeigen, daß alle prädikatenlogischen Beweise, die bei der Ausführung von PROLOG-Programmen realisiert werden können, genau den konstruktiven Beweisanforderungen entsprechen, die seitens der intuitionistisch-konstruktivistischen Logikkonzeption gefordert werden. Dies liegt letztlich an der Einschränkung der PROLOG-Syntax auf HORN-Klauseln. Vgl. zu dieser eineindeutigen Korrespondenz zwischen PROLOG-Beweisen und Konstruktivismus RICHTER,M.M. (1988), S. 22f.; vgl. auch die Anmerkung zur Konstruktivität des Unifizierungskonzepts. KOWALSKI (1978), S. 87, charakterisiert die Konstruktivität der Sprache PROLOG dadurch, daß er sie als kombinierte Repräsentations- und Abfragesprache hervorhebt. Ihre Eigenschaft als Abfragesprache ermöglicht es, Existenzfragen konstruktiv zu beantworten, indem die existierenden Objekte konkret aufgelistet werden.

23) Diese Einschränkung reflektiert die oben angesprochenen praktischen Limitierungen, denen PROLOG-Implementierungen des kombinierten Unifizierungs- und Resolutionskonzepts unterliegen.

24) Vgl. zu dieser vollständigen Auflistungsmöglichkeit aller formalen Objekte, die einem präsentierten Term aus einer vorgegebenen Objektmenge zugeordnet werden können, PROLOG (o.J.), S. 24 (u. 57f.); TOWNSEND,C. (1987), S. 99f.; DELAHAYE (1987), S. 158 u. 163; KINNEBROCK (1988), S. 62.

25) Backtracking stellt ein allgemeines Lösungskonzept für die vollständige Lösungssuche in Lösungsräumen dar, die in der Gestalt von Lösungsbäumen strukturiert sind. Beim Backtracking wird entlang des Lösungsbaums zurückgeschritten ("back-tracking"), sobald erkannt wird, daß eine von vier Abbruchsbedingungen erfüllt ist:
- Erstens kann festgestellt werden, daß sich auf dem bisher untersuchten Lösungspfad grundsätzlich keine zulässige Lösung mehr finden läßt.
- Zweitens ist es möglich einzusehen, daß auf diesem Lösungspfad mit Sicherheit keine bessere Lösung als eine bereits gefundene zulässige Lösung erhalten werden kann. (Diese Option erscheint im Kontext der logischen Programmierung zunächst irrelevant, weil dort wegen der zweiwertigen Logik nur zwischen zulässigen und unzulässigen Lösungen unterschieden werden kann, nicht aber zwischen verschieden guten zulässigen Lösungen. Sie gestattet aber, die nachfolgende dritte Abbruchsbedingung als Derivat der zwei vorgenannten aufzufassen.)
- Eine dritte Abbruchsbedingung ist erfüllt, wenn die Lösungssuche am Ende eines Lösungspfads im Lösungsbaum angelangt ist. Die beiden voranstehenden Abbruchsbedingungen schließen diese Bedingung ein. Entweder wurde auf dem Lösungspfad bisher noch keine zulässige Lösung entdeckt. Dann steht auch in seinem Endpunkt fest, daß keine zulässige Lösung mehr gefunden werden kann (erste Abbruchsbedingung). Denn in einem Endpunkt des Lösungsbaums kann die Lösungssuche auf demselben Pfad per definitionem nicht fortgesetzt werden. Oder es wurde auf dem Lösungspfad bereits mindestens eine zulässige Lösung gefunden. In diesem Fall ist in seinem Endpunkt bekannt, daß sich auf diesem Pfad keine besserstellende Lösung finden läßt (zweite Abbruchsbedingung). Auch hier ist eine Fortsetzung der Lösungssuche auf dem aktuellen Pfad per definitionem unmöglich.
- Viertens kann die Lösungssuche auf einem Lösungspfad abgebrochen werden, weil eine formale Restriktion hinsichtlich der Suchtiefe erreicht wurde, ohne daß eine der drei vorgenannten Abbruchsbedingungen erfüllt werden konnte.

In jedem der vier voranstehend skizzierten Fälle wird auf dem Lösungspfad zu dem jeweils nächsten voranliegenden Knoten zurückgeschritten, von dem ein alternativer, noch nicht untersuchter Lösungspfad abzweigt. Dieser Lösungspfad wird in der gleichen Weise untersucht wie der zuvor analysierte. Die Lösungssuche wird solange fortgesetzt, bis entweder erkannt wird, daß alle existierenden Lösungspfade des Lösungsbaums abgebrochen oder aber zu Ende durchsucht worden sind (vollständige Baumabarbeitung). Oder die Lösungssuche muß erfolglos eingestellt werden, weil ein vorgegebener maximal zulässiger Ressourceneinsatz für die Lösungssuche überschritten worden ist (ressourcenbedingter Abarbeitungsabbruch).
Lösungsalgorithmen, die auf dem Backtracking-Konzept beruhen, sind im Operations Research vor allem als beschränkte Enumerationsalgorithmen vom branch and bound-Typ bekannt. Vgl. zum Konzept des Backtracking REITMAN (1964), S. 307f.; BITNER (1975), S. 651ff.; NILSSON,N. (1980a), S. 24f. u. 55ff.; KORNFELD (1981a), S. 575; COHEN,P. (1982), S. 526ff.; LAURENT (1984), S. 154ff.; ZELEWSKI (1986a), S. 257 u. 356; DORN (1989), S. 135f.
Strenggenommen handelt es sich beim hier angesprochenen Backtracking-Konzept nur um die vorherrschende Variante aus einem facettenreichen Spektrum verwandter Backtracking-Konzepte. Sie wird mitunter auch präzisierend als chronologisches Backtracking bezeichnet, weil zum jeweils *nächsten* voranliegenden Knoten mit einem noch nicht untersuchten alternativen Lösungspfad zurückgeschritten wird; vgl. z.B. DE KLEER (1986a), S. 135f., und

DORN (1989), S. 135. Diese Art des Zurückschreitens ist jedoch keineswegs notwendig. Eine andere Variante wird später als problemsensitives Backtracking vorgestellt.

26) Vgl. zur Anwendung des Backtracking-Konzepts im Rahmen der logischen Programmierung PROLOG (o.J.), S. 23f. u. 56ff.; CLARK,K. (1979), S. 133f.; PLÜMER (1986), S. 137ff.; DE KLEER (1986a), S. 136; ESTENFELD (1987), S. 68f.; TOWNSEND,C. (1987), S. 52f.; DELAHAYE (1987), S. 141; CORDES (1988), S. 22f. (u. 47f.); KINNEBROCK (1988), S. 58f. u. 62ff.; DRESSLER (1989b), S. 14.

4.2.3 Erweiterungen der konventionellen Prädikatenlogik

4.2.3.1 Repräsentation temporalen Wissens

Als zentrale Modellierungsprämisse dieser Arbeit wurde vorausgesetzt, daß sich alle Aspekte der Koordinierung Flexibler Fertigungssysteme durch prädikatenlogische Objektmodelle ausdrücken lassen. Ein wesentliches Modellierungsziel besteht darin, diese Objektmodelle ohne Informationsverlust in Netzmodelle zu transformieren, die mit der Hilfe von Synthetischen Netzen formuliert werden.

Zwischen den intendierten Netzmodellen einerseits und den vorausgesetzten prädikatenlogischen Objektmodellen andererseits klafft jedoch vorerst noch eine erhebliche Modellierungslücke. Denn Netzmodelle zeichnen sich u.a. dadurch aus, Veränderungen der Modellierungsobjekte und die kausalen Mechanismen, die diesen Veränderungen zugrundeliegen, unmittelbar abbilden zu können[1)]. Demgegenüber bereitet es im Rahmen der Prädikatenlogik Schwierigkeiten, Wissen über zeitlich variable Sachverhalte auszudrücken[2)]. Hierdurch wird die Darstellungsmächtigkeit prädikatenlogischer Objektmodelle erheblich beeinträchtigt. Dies äußert sich beispielsweise in dem schon früher angesprochenen prädikatenlogischen Monotonieprinzip: Ihm zufolge bleibt eine prädikatenlogische Formel für immer gültig, sobald die einmal als gültig aufgewiesen werden konnte. Daraus resultiert eine problematische Starrheit prädikatenlogischer Objektmodelle.

Zwar lassen sich im Rahmen der konventionellen Prädikatenlogik statische Wissensinhalte ohne Schwierigkeiten repräsentieren. Hierzu gehören einerseits Informationen über die jeweils aktuellen Objektzustände. Andererseits umfaßt das statische Wissen zeitlich invariante Restriktionen, die von allen zulässigen Objektzuständen erfüllt werden müssen. Sie werden in der Regel als Integritätsbedingungen thematisiert.

Aber es erweist sich als umständlich, in konventionellen prädikatenlogischen Objektmodellen Prozesse darzustellen, in deren Verlauf sich die Zustände der Modellierungsobjekte verändern[3)]. Das zugehörige Objektwissen wird hier als prozessuales oder kinetisches Wissen bezeichnet. Noch schwieriger ist es, auf rein prädikatenlogische Weise diejenigen kausalen Mechanismen darzustellen, welche die Übergänge zwischen den Objektzuständen bewirken[4)]. Das Wissen über solche Kausalmechanismen heißt fortan dynamisches Wissen. Beiden vorgenannten Wissenskategorien gemeinsam ist ihr Bezug auf Veränderungen des jeweils modellierten Objekts. Sie drücken zwei verschiedene Aspekte derjenigen Operationen aus, die in einem Objektmodell abgebildet werden. Das prozessuale Wissen beschreibt die Wirkungen *tatsächlicher* Operations*ausführungen* durch die Zustandsveränderungen des modellierten Objekts. Das dynamische Wissen repräsentiert dagegen die Gesamtheit aller *möglichen* Zustandsveränderungen, indem es die Ausführungsvoraussetzungen und -wirkungen aller zulässigen Operationen darstellt. Darüber hinaus können prozessuales und dynamisches Wissen als flache bzw. tiefe Variante des operationsbezogenen Objektwissens aufgefaßt werden[5)]. Sie werden daher unter dem Begriff des operationalen Wissens zusammengefaßt.

Zeitliche Veränderungen von Wissen über ein modelliertes Objekt erstrecken sich aber nicht nur auf das operationale Wissen. Vielmehr können sie auch noch auf einer übergeordneten Ebene dadurch eintreten, daß das Wissen über ein Modellierungsobjekt durch Lern- und Vergessensprozesse auf der Seite des Modellierungsträgers zu- bzw. abnimmt[6)]. Solche Veränderungen des Objektwissens können ihrerseits als Metawissen über einen Modellierungsprozeß repräsentiert werden. Es wird als epistemisches Wissen[7)] angesprochen. Zusammen mit dem operationalen Wissen wird es unter den Oberbegriff des temporalen oder zeitbezogenen Wissens subsumiert.

Zentrales Anliegen der anschließenden Erörterungen ist es, das temporale Wissen in prädikatenlogische Objektmodellierungen einzubinden. Hierdurch wird die o.a. Modellierungslüche zwischen Objekt- und Netzmodellen geschlossen. Angelpunkt der prädikatenlogischen Repräsentation temporalen Wissens ist das Konzept der Fakten. Ein Fakt wurde bereits als konstante atomare Formel[8] $prä_u(ob_1,...,ob_{Ku})$[9] eingeführt, die unter einer gegebenen Interpretation I_r gültig ist[10]. Die prädikatspezifische Faktenmenge $FAK_{u.r}$ umfaßt alle Grundtermformeln $prä_u(gt_1,...,gt_{Ku})$ des Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$, die unter der aktuellen Interpretation I_r gültige konstante Formeln $prä_u(ob_1,...,ob_{Ku})$ darstellen[11]. Die modellspezifische Faktenmenge FAK_r ist die Vereinigungsmenge aller prädikatspezifischen Faktenmengen[12], die sich über alle modellkonstituierenden atomaren Grundtermformeln erstreckt. Die Gesamtheit des Objektwissens, das über ein Modellierungsobjekt aktuell zur Verfügung steht, wird durch die modellspezifische Faktenmenge und durch die logische Verknüpfung[13] der involvierten Fakten im prädikatenlogischen Objektmodell repräsentiert[14].

Die Repräsentation von Wissen über zeitlich variable Sachverhalte knüpft grundsätzlich an Variationen der prädikatspezifischen Faktenmengen an. Solche Veränderungen der prädikatspezifischen Faktenmengen schlagen sich notwendig als entsprechende Variationen der modellspezifischen Faktenmenge nieder[15]. Der Einfachheit halber konzentrieren sich die anschließenden Ausführungen auf die modellspezifische Faktenmenge[16]. Prozessuales Wissen wird dadurch ausgedrückt, daß nicht mehr gültige atomare Formeln aus dieser Faktenmenge eliminiert und gültig werdende Formeln in die Faktenmenge aufgenommen werden. Dynamisches Wissen wird durch Übergangsoperationen formuliert, welche das Ungültig- bzw. Gültigwerden von Fakten determinieren. Epistemisches Wissen über Lernvorgänge findet darin seinen Ausdruck, daß der Informationsgehalt der modellspezifischen Faktenmenge im Zeitablauf zunimmt[17].

Die zeitliche Veränderung der prädikatspezifischen Faktenmengen $FAK_{u.r}$ setzt voraus, daß die zugrundeliegenden Interpretationen I_r der Formeln eines prädikatenlogischen Objektmodells entsprechend variiert werden. Durch diese Interpretationen werden auch die Extensionen $EXT_{u.r}$ aller Prädikatssymbole $Prä_u(st_1,...,st_{Ku})$ festgelegt. Daher kann temporales Wissen nicht nur durch Veränderungen der Faktenmengen $FAK_{u.r}$, sondern ebenso durch zeitlich variable Extensionen $EXT_{u.r}$ der Prädikatssymbole eines Objektmodells repräsentiert werden. Da beide Darstellungsweisen durch die gemeinsam zugrundeliegenden Interpretationen I_r eindeutig determiniert werden, verhalten sie sich zueinander äquivalent. Folglich läßt sich temporales Wissen sowohl fakten- als auch extensionsbezogen behandeln. Beide Alternativen werden nachfolgend gleichberechtigt verwendet[18].

Der Index "r" von Faktenmengen $FAK_{u.r}$ und von Prädikatsextensionen $EXT_{u.r}$ mit $r \in \mathcal{N}_0$ verweist jeweils auf denjenigen Zeitabschnitt[19], innerhalb dessen sich das prädikatenlogisch repräsentierte Wissen über ein modelliertes Objekt nicht verändert. Ein solcher Zeitabschnitt wird als Zustand "r" des Objektmodells bezeichnet[20]. Dieser Modellzustand korrespondiert mit derjenigen Interpretation I_r des prädikatenlogischen Objektmodells, die den prädikatspezifischen Faktenmengen $FAK_{u.r}$ und den Prädikatsextensionen $EXT_{u.r}$ zugrundeliegt. Das gesamte Wissen, das im Zustand "r" über ein prädikatenlogisches Objektmodell vorliegt, wird durch die modellspezifische aktuelle Faktenmenge FAK_r repräsentiert. Sie wird daher auch als faktische Repräsentation des Modellzustands "r" bezeichnet.

Der Zustand eines prädikatenlogischen Objektmodells ist umfassender definiert als der Zustand des jeweils modellierten Objekts. Denn der Modellzustand hängt nicht nur vom Modellierungsobjekt selbst, sondern auch vom Modellierungsträger ab. Diese zweifache Dependenz läßt sich anhand möglicher Übergänge zwischen Modellzuständen verdeutlichen. Ein Übergang zwischen zwei benachbarten Modellzuständen wird jeweils dadurch ausgelöst, daß der Modellierungsträger eine modellierungsrelevante Veränderung des betrachteten Modellierungsobjekts wahrgenommen hat. Eine solche Wahrnehmungsveränderung kann einerseits auf tatsächlichen oder hypothetischen Veränderungen des Modellierungsobjekts beruhen[21]. Andererseits läßt sie sich ebenso auf kognitive Prozesse des Modellierungsträgers zurückführen, die das modellierte Objekt als verändert erscheinen lassen, obwohl es tatsächlich keine Veränderung erfahren hat.

Tatsächliche Veränderungen des modellierten Objekts heißen operationsbedingt, weil sie durch prozessuales Wissen über Operationsausführungen im Objektmodell repräsentiert werden. Ebenso werden die hypothetischen Objektveränderungen, die durch das dynamische Wissen über grundsätzlich zulässige Operationen im Objektmodell festgelegt sind, zu den operationsbedingten Veränderungen eines Objektmodells gerechnet. Von kognitionsbedingten Modellveränderungen wird dagegen gesprochen, wenn Variationen eines Objektmodells ausschließlich aus Lernprozessen des Modellierungsträgers resultieren[22]. Sie werden durch epistemisches Wissen ausgedrückt. Je nachdem, auf welche Weise der Übergang zwischen zwei benachbarten Modellzuständen verursacht wird, liegt eine operations- oder eine kognitionsbedingte Modellveränderung vor. Darüber hinaus können auch beide Veränderungsmodi miteinander kombiniert auftreten[23].

Operations- und kognitionsbedingte Modellveränderungen beim Übergang zwischen je zwei benachbarten Modellzuständen "r" und "f"[24] schlagen sich in einem prädikatenlogischen Objektmodell als Variationen der zugehörigen Faktenmengen $FAK_{u.r}$ bzw. $FAK_{u.f}$ nieder. Gleiches gilt für die korrespondierenden Veränderungen der Prädikatsextensionen $EXT_{u.r}$ bzw. $EXT_{u.f}$. Diese Modifizierungen von Faktenmengen bzw. Prädikatsextensionen[25] stellen die Basiskonzeption dar, mit deren Hilfe sich auch in prädikatenlogischen Objektmodellen Wissen über zeitlich variable Sachverhalte repräsentieren läßt. Die Repräsentationsmöglichkeit temporalen Wissens impliziert zwei Konsequenzen, die nicht nur aus der engeren prädikatenlogischen Perspektive Interesse erwecken, sondern sich auch auf generelle modell- und netztheoretische Aspekte erstrecken. Sie werden anhand der logischen Monotonieproblematik und in bezug auf Lernprozesse erörtert.

Die erste Konsequenz betrifft die Monotonieprämisse der konventionellen Prädikatenlogik. Da die Fakten- und Extensionsvariationen keinen generellen Einschränkungen unterliegt, kann nicht ausgeschlossen werden, daß vormals gültige Prädikatsvorkommnisse ungültig werden[26]. Diese spezielle Variante der zeitlichen Variabilität von Faktenmengen und Prädikatsextensionen bedeutet, daß die prädikatenlogische Monotonieprämisse nicht mehr eingehalten werden kann. Dieser Monotonieverlust stellt jedoch keinen grundsätzlichen Modellierungsnachteil dar. Vielmehr öffnet er das hier entfaltete Modellierungskonzept in logischer Hinsicht für das - im Vergleich zur konventionellen Prädikatenlogik - bisher weit weniger beachtete Feld der nonmonotonen Logiken[27].

An dieser Stelle interessiert nur das Ausdruckspotential, das durch den Verzicht auf die prädikatenlogische Monotonieprämisse erschlossen wird[28]. Es zeichnet sich dadurch aus, daß innerhalb einer nonmonotonen Logik auch Ereignisse, Aktionen, Prozesse und ähnliche dynamische Konzepte behandelt werden können, welche die Wahrheitswerte von Prädikaten im Zeitablauf verändern[29]. Erst hierdurch wird die konventionelle Prädikatenlogik so modifiziert, daß sich mit ihrer Hilfe auch dynamische Systemstrukturen und daraus folgende Systemprozesse innerhalb eines logisch ausgerichteten Objektmodells adäquat repräsentieren lassen[30]. Auf diese Weise können einerseits in prädikatenlogischen Objektmodellen die Ursachen und Wirkungen derjenigen Operationen dargestellt werden, auf denen alle operationsbedingten Modellveränderungen beruhen. Andererseits wird eine Brücke zum Petrinetz-Konzept geschlagen, das in seinen Transitionen und Markierungsfolgen von vornherein über dynamische bzw. prozessuale Komponenten verfügt. Ein Festhalten an der Monotonieprämisse der konventionellen Prädikatenlogik würde hingegen bedeuten, alle vorgenannten Aspekte temporalen Wissens nicht auf logischer Ebene bewältigen zu können.

Die zweite Konsequenz, die aus der Repräsentationsmöglichkeit temporalen Wissens folgt, betrifft die Chance, in prädikatenlogischen Objektmodellen und daraus abgeleiteten Netzmodellen Lernprozesse der Modellierungsträger zu erfassen. Allerdings wird diese Perspektive nachfolgend nur als eine Option vorgestellt, die grundsätzlich realisiert werden könnte, aber in dieser

Arbeit nicht konkret umgesetzt wird. Denn die Repräsentation solcher Lernprozesse könnte zwar aus modellierungstheoretischem Blickwinkel von großem Interesse sein. Doch erscheint dem Verf. der erhebliche Aufwand, der dafür erforderlich wäre, für die Problemstellung der hier vorgelegten Arbeit als nicht gerechtfertigt. Dennoch wird die Darstellungsmöglichkeit von Lernprozessen skizziert, um die Ausdrucksmächtigkeit des hier entfalteten Modellierungskonzepts zu demonstrieren. Zugleich wird unter dem Aspekt der Konzeptfruchtbarkeit die Option einer späteren Konzeptfortentwicklung aufgezeigt, ohne sie jetzt schon zu verwirklichen.

Zur Erfassung von Lernprozessen wird die frühere Definition eines prädikatenlogischen Fakts in wissensbezogener Weise nuanciert[31]. Nunmehr gilt als ein Fakt jede interpretierte atomare Grundtermformel $prä_u(ob_1,...,ob_{Ku})$, die unter einer gegebenen Interpretation I_r als gültig *bekannt* ist. Der Wahrheitswert eines Prädikatsvorkommnisses $prä_u(ob_1,...,ob_{Ku})$, das nicht zur aktuellen Faktenmenge $FAK_{u.r}$ des Prädikats $Prä_u$ gehört, ist daher zunächst unterbestimmt. Denn wegen $prä_u(ob_1,...,ob_{Ku})$ $FAK_{u.r}$ steht nur fest, daß es sich bei dem Prädikatsvorkommnis $prä_u(ob_1,...,ob_{Ku})$ um kein Fakt handeln kann. Also besteht strenggenommen nur Nichtwissen: Es ist *unbekannt*, ob das Prädikatsvorkommnis gültig ist. Gleiches gilt bezüglich der Ungültigkeit des Prädikatsvorkommnisses. Aus einem solchen doppelten Nichtwissen läßt sich aber ohne Zusatzannahmen kein Wissen über den tatsächlichen Wahrheitswert des Prädikatsvorkommnisses erschließen. Falls das Wissen über das modellierte Objekte unvollständig ist, kann auf der Basis dieses Wissens nicht entschieden werden, ob die Grundtermformel $prä_u(gt_1,...,gt_{Ku})$ für das Konstantentupel $(ob_1,...,ob_{Ku})$ gültig oder ungültig ist.

Beispielsweise läßt sich vorstellen, daß das Konstantentupel $(ob_1,...,ob_{Ku})$ das Prädikat $prä_u$ zwar erfüllt[32], dies aber bisher noch nicht nachgewiesen wurde[33]. Hieran wird deutlich, daß die Faktenmenge $FAK_{u.r}$ eines Prädikats $Prä_u$ keineswegs alle Prädikatsvorkommnisse $prä_u(ob_1,...,ob_{Ku})$ enthalten muß, die im Sinne einer deklarativen Semantik gültig sind[34]. Vielmehr umfaßt die Faktenmenge nur diejenigen Fakten, deren Gültigkeit vom Modellierungsträger auch tatsächlich gewußt wird[35]. Dies hebt eine wesentliche Charakteristik aller zeitlich variablen Faktenmengen hervor. Sie betrifft die fundamentale epistemologische Differenzierung zwischen explizitem und implizitem Wissen[36]. Das explizite Wissen über die Faktenmenge $FAK_{u.r}$ eines Prädikats $Prä_u$ umfaßt hier[37] die Menge aller Prädikatsvorkommnisse $prä_u(ob_1,...,ob_{Ku})$, deren aktuelle Gültigkeit unmittelbar bekannt ist. Das implizite Wissen über dasselbe Prädikat $Prä_u$ stellt dagegen die Menge aller Prädikatsvorkommnisse $prä_u(ob_1,...,ob_{Ku})$ dar, die aus der explizit gewußten Faktenmenge durch die Anwendung zulässiger[38] Inferenzregeln grundsätzlich abgeleitet werden können[39]. Da die Inferenzregeln gültigkeitserhaltenden Charakter besitzen und die explizit bekannten Fakten per definitionem gültige Prädikatsvorkommnisse sind, müssen auch alle implizit gewußten Prädikatsvorkommnisse gültig sein. Sofern diese gültigen Prädikatsvorkommnisse noch nicht in der explizit gewußten Faktenmenge $FAK_{u.r}$ enthalten sind, stellen sie Protofakten dar. Diese Protofakten werden zu Fakten, sobald ihre Gültigkeit durch tatsächliches Anwenden der erforderlichen Inferenzregeln explizit bekannt wird.

Durch die zeitliche Variabilität der Faktenmenge $FAK_{u.r}$ eines Prädikats $Prä_u$ kann ausgedrückt werden, wie sich das explizit verfügbare Wissen über gültige Prädikatsvorkommnisse durch inferentielle Auswertung des impliziten Wissens im Zeitablauf vermehrt. Durch diese Explizierung des impliziten Wissens wird die prädikatspezifische Faktenmenge $FAK_{u.r}$ sukzessiv vergrößert[40]. Hierbei werden Prädikatsvorkommnisse, deren Wahrheitswerte vor Ausführen der Inferenzen noch unbekannt waren, als faktische Prädikatsvorkommnisse aufgedeckt, deren Gültigkeit nunmehr bekannt ist. Daher liegt bei der Wissensexplizierung ein typischer Lernprozeß vor. Dieser Lernprozeß läßt sich durch das monotone Anwachsen der Faktenmenge $FAK_{u.r}$ im Zeitablauf modellieren.

Da beim inferenzbedingten Lernen die Monotonieprämisse der konventionellen Prädikatenlogik nicht verletzt wird, betrifft es nicht die oben thematisierten operationsbedingten Veränderungen von Objektmodellen. Statt dessen handelt es sich beim Lernen durch Wissensexplizierung um kognitionsbedingte Veränderungen von Objektmodellen. Denn das explizite Wissen eines Modellierungsträgers über ein modelliertes Objekt wird im Zeitablauf dadurch größer, daß implizites Wissen, das im Objektmodell enthalten ist, inferentiell erschlossen wird. Hierdurch wächst der explizit verfügbare Informationsgehalt des Objektmodells. Dabei wird der Informationsgehalt IG des Objektmodells durch die Anzahl derjenigen interpretierten atomaren Grundtermformeln[41] gemessen, deren Gültigkeit explizit bekannt ist. Dieser Informationsgehalt ist die Kardinalität der modellspezifischen Faktenmenge FAK_r im aktuellen Modellzustand "r": $IG=\#(FAK_r)$.

Interessant an den kognitionsbedingten Modellveränderungen ist nicht die Tatsache der Wissenserschließung an sich. Sie gilt für alle inferentiellen Auswertungen logischer Objektmodelle. Bemerkenswert ist vielmehr, daß durch die Variabilität der prädikat- und der modellspezifischen Faktenmengen der inferenzbedingte Lernprozeß selbst explizit dokumentiert werden kann. Noch bedeutsamer ist allerdings die Möglichkeit, die Konzepte variabler Faktenmengen und kognitionsbedingter Modellveränderungen durch den Übergang zu einer dreiwertigen Logik zu verfeinern. Dadurch können einige gravierende Schwierigkeiten beseitigt werden, unter denen konventionelle prädikatenlogische Objektmodellierungen leiden.

Ansatzpunkt dieser Schwierigkeiten ist das "negation by failure"-Prinzip der logischen Programmierung, das schon an früherer Stelle erläutert wurde[42]. Es stellt eine *epistemische Konvention* dar, mit deren Hilfe zumeist das *Nichtwissen* bezüglich der Wahrheitswerte solcher Prädikatsvorkommnisse $prä_u(ob_1,...,ob_{Ku})$ beseitigt wird, die in einem prädikatenlogischen Objektmodell nicht zur aktuellen Faktenmenge $FAK_{u.r}$ des Prädikats $Prä_u$ gehören. Aufgrund des "negation by failure"-Prinzips wird ein Prädikat $Prä_u$ im aktuellen Modellzustand "r" für alle diejenigen Konstantentupel $(ob_1,...,ob_{Ku})$ als ungültig unterstellt, auf die zwei Bedingungen zutreffen[43]: Erstens dürfen die Konstantentupel keine Argumente von explizit bekannten Fakten darstellen. Zweitens dürfen sie nicht zu Prädikatsvorkommnissen gehören, deren Gültigkeit aus dem Objektmodell inferentiell erschlossen werden kann.

Diese Konvention wird seitens des Petrinetz-Konzepts im Regelfall sogar dahingehend verschärft, daß ein Prädikatsvorkommnis $prä_u(ob_1,...,ob_{Ku})$ bereits dann als ungültig betrachtet wird, wenn es nicht zur aktuellen Faktenmenge $FAK_{u.r}$ des Prädikats $Prä_u$ gehört. Diese Konvention wird als Explizitheitsprämisse des Petrinetz-Konzepts bezeichnet.

Das gesamte Wissen, das sich über den Zustand "r" eines modellierten Objekts durch objektbeschreibende Prädikatsvorkommnisse und deren Wahrheitswerte ausdrücken läßt, wird aufgrund dieser Konvention durch die modellspezifische Faktenmenge FAK_r vollständig repräsentiert. Denn die Prädikatsvorkommnisse, die hier stets als interpretierte atomare Grundtermformeln gemeint sind, können im Rahmen einer klassischen zweiwertigen Logik[44] nur gültig oder ungültig sein. Prädikatsvorkommnisse, welche die aktuelle Faktenmenge umfaßt, sind per definitionem gültig. Alle anderen Prädikatsvorkommnisse, die in der Faktenmenge nicht enthalten sind, sind per conventionem ungültig. Eine dritte Möglichkeit existiert nicht.

Bei der Explizitheitsprämisse handelt es sich um eine erhebliche Vereinfachung der praktischen Handhabung von Netzmodellen. Erstens werden aufgrund des "negation by failure"-Prinzips alle Prädikatsvorkommnisse, die weder direkt als gültig ausgewiesen sind noch indirekt als gültig abgeleitet werden können[45], von vornherein als ungültig unterstellt. Damit werden alle weiterführenden Überlegungen, welcher Gültigkeitswert solchen Prädikatsvorkommnissen zukommen könnte, vermieden. Zweitens werden durch die Explizitheitsprämisse sogar alle indirekten Ableitungsmöglichkeiten der Prädikatsgültigkeit ausgeschlossen. Nur genau diejenigen Prädikatsvorkommnisse werden als gültig anerkannt, die in der aktuellen Netzmarkierung als Fakten explizit ausgewiesen werden[46]. Damit umfaßt die aktuelle Faktenmenge FAK_r nicht nur diejenigen Prädikatsvorkommnisse, deren Gültigkeit explizit bekannt ist. Vielmehr wird unterstellt, daß sie alle überhaupt gültigen Prädikatsvorkommnisse enthält. Die Denkmöglichkeit, die Gültigkeit von Prädikatsvorkommnissen erst nachträglich durch Inferenzprozesse zu erschließen, wird qua Prämisse ausgeblendet. Diese Nichtbeachtung indirekter Ableitungsmöglichkeiten spielt *hier* keine wesentliche Rolle, weil nur die *Objektrepräsentation* interessiert. Der Aspekt der Inferenzmöglichkeit bleibt dagegen in *diesem* repräsentationsbezogenen Kontext irrelevant[47]. Daher wird die Explizitheitsprämisse fortan nur unter dem Aspekt ihres ersten Vereinfachungsschritts, des "negation by failure"-Prinzips, näher beleuchtet.

Das "negation by failure"-Prinzip erweist sich aus epistemologischer Perspektive als eine rigide, potentiell fehlerhafte[48] Festlegung[49]. Sie entspricht[50] jedoch der allgemein üblichen[51] modelltheoretischen Prämisse einer geschlossenen Weltmodellierung[52]. Diese Prämisse unterstellt eine Modellierungsweise, bei der für alle modellierungsrelevanten Aspekte des abgebildeten Realitätsausschnitts[53] gilt: Entweder werden sie im Objektmodell explizit dargestellt. Oder sie lassen sich zumindest aus dem Wissen, das im Objektmodell implizit enthalten ist, inferentiell erschließen[54]. Ein derart konzipiertes Objektmodell wird hier als extensional[55] definiertes Objektmodell mit kontrollierter Explizitheit[56] bezeichnet. Die Kohärenz zur früher vorgetragenen, u.a. in WITTGENSTEIN's Frühwerk besonders hervorgehobenen[57] Präferenz extensionaler Konzeptkonstruktionen ist offensichtlich. Allerdings wird die Präzision und Transparenz extensionaler Konzeptkonstruktionen durch die epistemologische Rigidität und potentielle Fehlerhaftigkeit des negation by failure-Prinzips erkauft. Daher werden im folgenden Modellierungsalternativen erörtert, welche dieses Prinzip vermeiden.

Es läßt sich die alternative Festlegung vorstellen, die Wahrheitswerte aller interpretierten atomaren Grundtermformeln $prä_u(ob_1,...,ob_{Ku})$, die weder selbst noch als ihre Negate zur aktuellen Faktenmenge $FAK_{u.r}$ des Prädikats $Prä_u$ gehören, zunächst[58] als unbestimmt zu behandeln. Dies entspricht einer offenen Weltmodellierung[59]. erfordert den Übergang zu einer dreiwertigen Logik, deren formale Semantik durch die erweiterte BOOL'sche Menge $BOOL_3 = \{gültig, ungültig, unbestimmt\}$ charakterisiert wird[60]. Hierdurch wird das tertium non datur-Prinzip[61] der konventionellen zweiwertigen Logik außer Kraft gesetzt[62]. Denn ein Prädikatsvorkommnis $prä_u(ob_1,...,ob_{Ku})$ erhält vorläufig den dritten Wahrheitswert "unbestimmt", solange es weder als gültig noch als ungültig bekannt ist[63]. Die beiden konventionellen Wahrheitswerte "gültig" und "ungültig" werden demgegenüber unter den Oberbegriff der bestimmten oder definiten Wahrheitswerte subsumiert.

Die Verwendung eines dritten Wahrheitswerts mit der Bedeutung "unbestimmt" reflektiert Ansätze der formallogischen Forschung, den Kalkül der Prädikatenlogik um eine trichotome Semantik zu bereichern[64]. Insbesondere BLAU[65] und KÖRNER[66] haben zwei subtilere[67] Varianten vorgelegt. In ihnen wird die formale Semantik eines indefiniten Wahrheitswerts "unbestimmt"[68] präzise ausgearbeitet. Er tritt neben die beiden definiten Wahrheitswerte "gültig" und "ungültig" aus konventionellen zweiwertigen Logiken. Zwar haben BLAU und KÖRNER nicht deutlich auf die Konzeption Bezug genommen, den Wahrheitswert "unbestimmt" so lange zu verwenden, wie einer der beiden bestimmten Wahrheitswerte "gültig" und "ungültig" noch nicht feststeht[69]. Doch ließen sich ihre Ansätze nach Einschätzung des Verf. im Sinne dieser Konzeption reformulieren[70]. Insbesondere die dreiwertige Logik KÖRNER's bietet sich hierfür an[71]. Denn sie sieht explizit die Möglichkeit vor, den Wahrheitswert einer Formel, der "provisorisch" als unbestimmt festgelegt worden ist, nachträglich in einen Wahrheitswert der konventionellen BOOL'schen Menge {gültig,ungültig} umzuwandeln[72].

Bei einer dreiwertigen logischen Basis müssen alle Prädikatsvorkommnisse $prä_u(ob_1,...,ob_{Ku})$, deren Gültigkeit oder Ungültigkeit bekannt ist, explizit repräsentiert werden[73]. Dies kann einerseits dadurch geschehen, daß als Fakten auch die Negate von interpretierten atomaren Grundtermformeln zugelassen werden[74]. Dann werden ungültige Prädikatsvorkommnisse $prä_u(ob_1,...,ob_{Ku})$ als Fakten $\neg prä_u(ob_1,...,ob_{Ku})$ in der Faktenmenge $FAK_{u.r}$ explizit aufgeführt. Der Begriff der Faktenmenge bleibt hierbei gegenüber dem o.a. konventionellen Modellierungskonzept unverändert. Andererseits ist es ebenso möglich, die bisher eingeführten Faktenmengen $FAK_{u.r}$ jeweils in zwei disjunkte Teilmengen $FAK_{u.r}^+$ und $FAK_{u.r}^-$ aufzuspalten. Die erste umfaßt alle Prädikatsvorkommnisse $prä_u(ob_1,...,ob_{Ku})$ mit bekannter Gültigkeit ("positive" Fakten). Die zweite erstreckt sich dagegen auf alle Prädikatsvorkommnisse $prä_u(ob_1,...,ob_{Ku})$ mit bekannter

Ungültigkeit ("negative" Fakten)[75]. Der Verf. präferiert die Alternative zweier disjunkter Faktenmengen. Denn nur sie erfüllt die modelltheoretische Anforderung der Unmittelbarkeit. Ein ungültiges Prädikatsvorkommnis wird in der Gestalt eines negativen Fakts unmittelbar als ein solches ausgewiesen. Bei der alternativen Option, weiterhin nur eine konventionell definierte Faktenmenge zu verwenden, würden dagegen ungültige Prädikatsvorkommnisse durch die Gültigkeit ihrer Negate nur mittelbar dargestellt.

Für den unkonventionellen Ansatz einer dreiwertigen Logik sprechen zunächst mehrere Gründe[76]. Erstens wird die Ausdruckskraft der konventionellen, zweiwertigen Prädikatenlogik um den dritten Wahrheitswert erweitert. Dies ist aus der Perspektive der Modellierungsmächtigkeit für das hier zu entfaltende Petrinetz-Konzept grundsätzlich von Interesse. Zweitens werden die früher dargelegten Schwierigkeiten vermieden, zu denen das "negation by failure"-Prinzip der logischen Programmierung führen kann. Denn aus dem Fehlschlag eines Inferenzversuchs, die Gültigkeit einer Formel abzuleiten, wird nicht mehr der - potentiell fehlerhafte - Schluß gezogen, das Negat der untersuchten Formel sei gültig. Vielmehr wird im Rahmen einer dreiwertigen Logik der logisch korrekte Schluß gezogen, der untersuchten Formel den Wahrheitswert "unbestimmt" zuzuordnen. Drittens existieren seitens der Petrinetz-Theorie bereits Konzepte, die Ungültigkeit eines Prädikatsvorkommnisses explizit durch Konstrukte zu repräsentieren, die sich als "negative" Fakten deuten lassen[77].

Der wesentliche - und vierte - Grund liegt aber in der Möglichkeit, nunmehr auch solche Lernprozesse explizit und formal dokumentieren zu können, die sich während der Gestaltung von Objektmodellen als empirischer Wissenserwerb über den jeweils modellierten Realitätsausschnitt abspielen. Hiermit bleiben kognitive Modellveränderungen nicht auf den oben erläuterten Spezialfall des inferenzbedingten Lernens beschränkt. Statt dessen läßt sich jetzt auch jedes Lernen durch empirische Wissensakquisition in das Modellierungskonzept einbeziehen. Dieses akquisitionsbedingte Lernen spielt bei der praktischen Modellierung betrieblicher Realprobleme bei weitem eine größere Rolle als das inferenzbedingte Lernen. Demgegenüber berücksichtigen konventionelle Modellierungskonzepte im allgemeinen keine Lernprozesse, die sich während der Objektmodellierung durch Wissensakquisition einstellen[78]. Sie bleiben im Regelfall darauf beschränkt, das jeweils vorhandene Wissen über einen relevanten Realitätsausschnitt in einem Objektmodell abzubilden. Akquisitionsbedingte Veränderungen des aktuellen Wissensbestands über das Modellierungsobjekt werden dagegen zumeist nicht näher konzeptualisiert.

Der Akt einer Wissensakquisition - d.h. der empirische Erwerb neuen Wissens über ein modelliertes Objekt - läßt sich im Rahmen einer dreiwertigen Logik dadurch repräsentieren, daß vom Wahrheitswert "unbestimmt" für ein objektbeschreibendes Prädikatsvorkommnis $prä_u(ob_1,...,ob_{Ku})$ zu einem der bestimmten Wahrheitswerte "gültig" oder "ungültig" übergegangen wird[79]. Dadurch wächst der Umfang der Faktenmenge $FAK_{u.r}$ oder eines ihrer beiden Derivate $FAK_{u.r}^+$ und $FAK_{u.r}^-$. Zugleich steigt hierdurch der Informationsgehalt IG des prädikatenlogischen Objektmodells, der wiederum durch die Kardinalität der modellspezifischen Faktenmenge(n) definiert wird: $IG = \#(FAK_r) = \#(FAK_r^+ \cup FAK_r^-)$.

Trotz der voranstehend aufgezeigten Vorzüge wird dem Modellierungskonzept dieser Arbeit keine dreiwertige Logik zugrundelegt. Hierfür spricht einerseits, daß die vertraute zweiwertige Semantik der Prädikatenlogik nicht verlassen zu werden braucht. Dadurch wird die Kompatibilität zu dem ausgereiften syntaktischen und semantischen Apparat der klassischen Logik - insbesondere auch zu allen hieraus gewonnenen formallogischen Erkenntnissen - gewahrt[80]. Andererseits scheinen dem Verf. die netztheoretischen Ansätze für die Erfassung "negativer" Fakten bisher noch zu wenig ausgereift, um auf ihnen bereits das Konzept Synthetischer Netze aufbauen zu können. Dies entspricht der allgemeinen Einschätzung von WEDEKIND, daß die Informatik insgesamt noch über keine praxistauglichen Konzepte verfügt, die auf dreiwertigen Logiken beruhen[81].

Schließlich lassen sich die oben erörterten Aspekte einer dreiwertigen Logik nicht in die prädikatenlogische Implementierung von Objektmodellen auf PROLOG-Basis integrieren[82]. Die vielfachen anderen Vorzüge dieser Implementierungsumgebung, die bereits dargelegt wurden, haben jedoch schon zu der Basisentscheidung geführt, dem Modellierungskonzept dieser Arbeit ein PROLOG-Fundament zugrundezulegen. Die - unbestritten interessanten - Aspekte einer dreiwertigen Logik rechtfertigen es nach Einschätzung des Verf. nicht, diese Basisentscheidung wieder aufzuheben. Insbesondere ginge das gesamte Implementierungsinstrumentarium verloren, das durch die Programmiersprache PROLOG und das Softwarepaket PASIPP zur Verfügung steht. Dreiwertige Logiken weisen dagegen noch nicht einmal Ansätze zu ihrer Implementierung auf. Damit bleiben sie zwar theoretisch vielversprechend, aber für praktische Modellierungszwecke untauglich.

Da an einer klassischen Logik mit ihrer zweiwertigen Semantik festgehalten wird, stellt sich die Frage, ob dennoch die oben thematisierten Schwierigkeiten des "negation by failure"-Prinzips und der geschlossenen Weltmodellierung vermieden und Aspekte operations- oder kognitionsbedingter Modellveränderungen erfaßt werden können. Dies ist tatsächlich der Fall. Zu diesem Zweck ist es allerdings erforderlich, die konventionell definierten Wahrheitsfunktionen der Prädikatenlogik als partielle Funktionen umzuformulieren und auf die Repräsentation von Wissen von Formelungültigkeiten zu verzichten. Der Verf. legt diesen Ansatz seinem Modellierungskonzept zugrunde. Die Verwendung partiell definierter Wahrheitsfunktionen und die Beschränkung auf gültige Prädikatsvorkommnisse bewertet er als eine nur geringfügige Komplizierung bzw. Einbuße an Darstellungskraft. Sie erscheinen angesichts der erheblichen Vorzüge gerechtfertigt, die darin liegen, auf die epistemisch fragwürdigen Prinzipien des "negation by failure" und der geschlossenen Weltmodellierung zu verzichten, aber dennoch Wissen über zeitlich variable Sachverhalte darstellen zu können.

Die Reformulierung der Wahrheitsfunktionen wf_u erstreckt sich auf zwei zusammenhängende Aspekte. Erstens wird ihre Bildmenge von der konventionellen BOOL'schen Menge $BOOL_2 = \{gültig, ungültig\}$ auf die reduzierte Menge $BOOL_1 = \{gültig\}$ eingeschränkt. Zweitens wird ihre Abbildungsvorschrift als eine partiell definierte Vorschrift aufgefaßt. Sie ordnet nicht mehr allen konstanten Objekttupeln $(ob_1,...,ob_{Ku})$ eines Prädikatssymbols $Prä_u$ jeweils einen Wahrheitswert zu. Statt dessen bildet sie nur noch diejenigen Objekttupel auf ihren Wahrheitswert ab, die als gültig *bekannt* sind[83]. Somit gilt für die derart modifizierten, *partiellen* Wahrheitsfunktionen pwf_u:

$$\begin{aligned} pwf_u: \quad & DB^{Ku} \rightarrow \{gültig\} \\ & (ob_1,...,ob_{Ku}) \rightarrow_{part} pwf_u(ob_1,...,ob_{Ku}) = gültig \end{aligned}$$

Für alle Objekttupel $(ob_1,...,ob_{Ku})$, die Argumente eines Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ sein könnten, denen die Wahrheitsfunktion wf_u aber keinen Wahrheitswert "gültig" zuordnet, bleibt der Wahrheitswert unbekannt[84]. Dies bedeutet zunächst, daß der Wahrheitswert "ungültig" weder explizit noch implizit erfaßt wird. Dies steht im Gegensatz zum "negation by failure"-Prinzip und zum Prinzip geschlossener Weltmodellierung: Aus dem Umstand, daß einem Objekttupel $(ob_1,...,ob_{Ku})$ der Wahrheitswert "gültig" weder direkt zugeordnet ist noch indirekt aus anderen Formelgültigkeiten hergeleitet werden kann, läßt sich nicht mehr die Ungültigkeit des Prädikatsvorkommnisses $prä_u(ob_1,...,ob_{Ku})$ folgern. Statt dessen drückt dieser Sachverhalt nur das aus, was er unmittelbar repräsentiert: Die Gültigkeit des Prädikatsvorkommnisses $prä_u(ob_1,...,ob_{Ku})$ ist unbekannt. Auf jede Überlagerung durch eine epistemische Konvention, wie dieses Nichtwissen als ein Wissen über den "tatsächlichen" Wahrheitswert des Prädikatsvorkommnisses interpretiert werden solle, wird bewußt verzichtet. Dadurch werden die epistemischen Komplikationen der beiden o.a. Prinzipien - und aller anderen "wissenschaffenden" Konventionen - von vornherein unterbunden. Es resultiert ein "epistemisch bereinigtes" Modellierungskonzept.

Darüber hinaus wird die zweiwertige Semantik der konventionellen Prädikatenlogik in besonderer Weise interpretiert. Die Semantik bleibt zwar zweiwertig, aber der konventionelle Wahrheitswert "ungültig" wird durch die Unbekanntheit eines Wahrheitswerts ersetzt[85]. Dadurch liegen nicht mehr zwei gleichberechtigte Wahrheitswerte vor. Vielmehr wird nur noch ein wohlbestimmter Wahrheitswert "gültig" benutzt. Sofern dieser Wahrheitswert unter einer Interpretation einer prädikatenlogischen Formel nicht zugeordnet werden kann, erhält diese Formel keinen anderen wohlbestimmten Wahrheitswert. Statt dessen bleibt ihr Wahrheitswert unbekannt. Diese Beschränkung auf nur noch einen wohlbestimmten Wahrheitswert drücken die partiell definierten Wahrheitsfunktionen mit ihrer einelementigen Bildmenge BOOL_1 aus.

Die "konzeptionellen Kosten" des epistemisch bereinigten Ansatzes bestehen vor allem darin, Wissen über ungültige Prädikatsvorkommnisse nicht mehr repräsentieren zu können. Denn es fehlt der wohlbestimmte Wahrheitswert "ungültig". Dies spielt allerdings für praktische Modellierungsaufgaben keine wesentliche Rolle. Denn jedes Prädikat, das in einem prädikatenlogischen Objektmodell zur Beschreibung eines modellierungsrelevanten Realitätsausschnitts beiträgt, läßt sich so formulieren, daß nur die Gültigkeit seiner Vorkommnisse Bedeutung erlangt[86]. Diese Formulierungsweise prädikatenlogischer Objektmodelle, die sich ausschließlich an Formelgültigkeiten ausrichtet, wird in dieser Arbeit vorausgesetzt (Gültigkeitsprämisse). Folglich läßt sich das *gesamte* Wissen, daß ein Modellierungsträger über einen modellierten Realitätsausschnitt besitzt, durch eine Sammlung gültiger Prädikatsvorkommnisse in einem Objektmodell abbilden[87]. Dies ist die schon früher eingeführte Faktenmenge prädikatenlogischer Objektmodelle.

Die Faktenmenge mit ihren gültigen Prädikatsvorkommnissen wird fortan anstelle der Wahrheitsfunktionen benutzt, um das Wissen eines Modellierungsträgers in einem Objektmodell auszudrücken. Hierdurch treten an die Stelle der statisch definierten partiellen Wahrheitsfunktionen pwf_u für die Prädikatssymbole $Prä_u$ die entsprechenden, aber zeitvariabel definierten prädikatspezifischen Faktenmengen $FAK_{u.r}$. Alle Konzepte für die Berücksichtigung operations- und kognitionsbedingter Modellveränderungen, die oben mit der Hilfe dieser Faktenmengen $FAK_{u.r}$ eingeführt wurden, werden übernommen. Dies gilt hier allerdings nur unter folgenden Einschränkungen: Es wird in Faktenmengen nur die bekannte Gültigkeit von Prädikatsvorkommnissen repräsentiert[88]. Veränderungen von Faktenmengen bedeuten nur, daß entweder Prädikatsvorkommnisse, deren Wahrheitswerte vormals als gültig bekannt waren, nun nicht mehr als gültig bekannt sind oder daß Prädikatsvorkommnisse, deren Wahrheitswerte vormals unbekannt waren, nunmehr als gültig bekannt sind. Dies folgt aus dem Verzicht auf den Wahrheitswert "ungültig". Aufgrund der Gültigkeitsprämisse können aber auch Prädikatsvorkommnisse, die den Wahrheitswert "ungültig" neu erhalten oder einbüßen würden, durch entsprechend formulierte komplementäre Prädikatsvorkommnisse erfaßt werden, die den Wahrheitswert "gültig" verlieren bzw. hinzugewinnen. Daher lassen sich alle operations- und kognitionsbedingten Veränderungen eines prädikatenlogischen Objektmodells durch die variablen Faktenmengen $FAK_{u.r}$ der modellkonstituierenden Prädikatssymbole $Prä_u$ vollständig erfassen.

Beispielsweise läßt sich der Wissenszuwachs, der durch inferentielles Erschließen des Wissens, das in einem Objektmodell implizit enthalten ist, explizit repräsentieren. Denn die Unbekanntheit des Wahrheitswertes aller Prädikatsvorkommnisse, die nicht explizit als gültig bekannt sind, hält die Möglichkeit offen, daß ihre Gültigkeit noch nachträglich durch Inferenzprozeduren erkannt wird, die auf das prädikatenlogische Formelsystem des Objektmodells angewendet werden. Dabei werden Prädikatsvorkommnisse, die wegen ihres unbekannten Wahrheitswertes zunächst nicht zur aktuellen Faktenmenge FAK_r des Objektmodells gehörten, nach dem inferenzbedingten Wissenszuwachs in die neue Faktenmenge FAK_f aufgenommen, sofern sie als gültig erkannt werden konnten.

Aufgrund der voranstehenden Erläuterungen ergibt sich folgender Rahmen für die prädikatenlogische Objektmodellierung: Ausgangspunkt ist eine Beschreibung des modellierungsrelevanten Realitätsausschnitts durch ein prädikatenlogisches Formelsystem. Dieses Formelsystem ist das prädikatenlogische Objektmodell.

Die statische Struktur des Objektmodells wird durch die Gesamtheit aller atomaren Prädikatsvorkommnisse, aus denen das Formelsystem zusammengesetzt ist, und die logischen Verknüpfungen zwischen diesen Prädikatsvorkommnissen konstituiert. Das Wissen über den jeweils aktuellen Zustand "r" des Objektmodells drückt eine Interpretation I_r des Formelsystems aus. Diese Interpretation bildet alle atomaren Grundtermformeln, die im aktuellen Modellzustand als gültig bekannt sind, auf diesen Wahrheitswert ab. Die Wahrheitswerte aller anderen atomaren Grundtermformeln bleiben unbekannt. Die Menge aller gültigen interpretierten atomaren Grundtermformeln ist die modellspezifische Faktenmenge FAK_r. Diese modellspezifische Faktenmenge repräsentiert das gesamte Wissen des Modellierungsträgers über den jeweils aktuellen Modellzustand, das sich durch Formelgültigkeiten darstellen läßt. Aufgrund der Gültigkeitsprämisse werden alle Wissenskomponenten, die sich als Formelungültigkeiten niederschlagen würden, mit der Hilfe komplementärer, jeweils gültiger Formeln repräsentiert.

Die dynamische Struktur des Objektmodells wird durch Übergangsoperationen ausgedrückt, die den jeweils aktuellen Modellzustand "r" in einen Folgezustand M_f transformieren können. Wird eine Übergangsoperation tatsächlich ausgeführt, so wird die Faktenmenge FAK_r operationsbedingt durch die Faktenmenge FAK_f ersetzt. Zustandsverändernde Ausführungen von Übergangsoperationen werden daher im prädikatenlogischen Objektmodell grundsätzlich als Veränderungen der zustandsrepräsentierenden Faktenmengen behandelt. Aus der kinetischen Modellierungsperspektive stellt jede solche Veränderung der Faktenmenge ein Ereignis dar. Ein Prozeß läuft im Objektmodell als intermittierende Abfolge von zustandsrepräsentierenden Faktenmengen und zustandsverändernden Operationsausführungen ab. Eine ausgezeichnete Faktenmenge - die Ausgangsfaktenmenge FAK_0 - repräsentiert den Ausgangszustand M_0 des Objektmodells. Zusätzlich kann zwischen zulässigen und unzulässigen Modellzuständen mit der Hilfe von Integritätsbedingungen differenziert werden. Sie definieren jeweils Anforderungen, die von allen modellspezifischen Faktenmengen FAK_r, die zulässige Modellzustände "r" repräsentieren sollen, erfüllt werden müssen.

Durch zeitvariable Faktenmengen lassen sich sowohl das aktuelle Vorhandensein von Wissen über ein modelliertes Objekt als auch dessen Veränderung im Zeitablauf darstellen. Aus dieser Perspektive legen sie - in Abhängigkeit von der jeweils betroffenen Wissenskategorie[89)] - das Fundament einer "kinetischen", "dynamischen" oder "epistemischen" Semantik für prädikatenlogische Objektmodelle. Da Faktenmengen den Extensionen von prädikatenlogischen Formeln entsprechen, erfolgt zugleich eine "Dynamisierung" des extensionalen, nur statisch definierten Prädikatsbegriffs der konventionellen Prädikatenlogik. Beide Aspekte werden in dieser Arbeit aber nicht weiter vertieft. Statt dessen werden zeitvariable Faktenmengen später benutzt, um für prädikatenlogische Objektmodelle eine operationale Semantik auf der Basis von Übergangsoperationen einführen zu können. Diese bilden die formale Grundlage für die Explizierung der Schaltregel von Synthetischen Netzen.

Anmerkungen zum Kapitel:

1) Dieses Urteil wird hier nur als Behauptung vorgetragen. Es wird jedoch später, nachdem das netzgestützte Modellierungskonzept detailliert entfaltet wurde, ausführlicher gerechtfertigt.
Die Veränderungsfreundlichkeit von Modellierungen auf der Basis von Petrinetzen trug ursprünglich dazu bei, deren Eignung für die Koordinierung Flexibler Fertigungssysteme näher zu untersuchen. Denn sowohl der Flexibilitätsbegriff als auch die Instabilität von realen Produktionssystemen begründen einen Bedarf nach Modellierungskonzepten, die sich besonders für die Repräsentation von Veränderungen im Wissen über ein Modellierungsobjekt eignen.

2) Auf diese Schwierigkeiten wurde bereits ausführlich hingewiesen.

3) Eine solche mühevolle Repräsentationsweise besteht z.B. darin, die Argumente der prädikatenlogischen Formeln mit Zeitindizes zu versehen.

4) Die logische Repräsentation des Wissens über solche übergangsverursachenden Mechanismen erfolgt im allgemeinen nicht mehr im Rahmen der konventionellen Prädikatenlogik. Vielmehr wird zu diesem Zweck auf Erweiterungen dieser Logik zurückgegriffen, die vor allem als Aktionslogik, als dynamische Logik oder als Handlungslogik thematisiert werden. Vgl. zu diesen Fortentwicklungen der Prädikatenlogik, die bis in den Bereich der Modallogik hineinreichen können, VON WRIGHT (1966), S. 121ff.; GENRICH (1973b), S. 2ff., insbesondere S. 6ff.; LENK (1976), S. 271ff.; GENRICH (1979b), S. 78ff.; HAREL (1979); VON WRIGHT (1980), S. 21ff.; BRENNENSTUHL (1980), S. 35ff.; VON KUTSCHERA (1980), S. 67ff.; MOORE,R. (1980), S. 53ff., bei dem der modallogische Erweiterungsaspekt besonders deutlich wird; APPELT (1980), S. 131ff.; GRABOWSKI,J. (1980c), S. 53ff. u. 59ff.; ROSENSCHEIN (1981), S. 331ff.; BIBEL (1986), S. 115ff.; ZELEWSKI (1986a), S. 197; BIBEL (1989), S. 49ff. Vgl. auch die ausführliche Thematisierung von "Zeitstrukturen" bei OBERWEIS (1990a), S. 24ff. Der Autor argumentiert zwar nicht ausdrücklich auf logischer Basis, doch lassen sich seine Überlegungen in diesen Kontext problemlos einordnen.
Eine besonders enge Verbindung zwischen dem Petrinetz-Konzept einerseits und den oben angeführten Logikkonzepten besteht in den Ausführungen von BIBEL. Sein aktionslogisches Konzept der Linearen Beweise konzentriert sich darauf, einen Plan zur Lösung eines vorgegebenen Problems durch einen prädikatenlogischen Theorembeweis zu erzeugen. Dabei wird bewiesen, daß eine Zielsituation (Problemlösung) von einer Ausgangssituation (Problembeschreibung) aus durch eine endliche Menge von Transformationsregeln erreicht werden kann; vgl. BIBEL (1989), S. 50f. Jede dieser Transformationsregeln entspricht einer Aktion, die im Petrinetz-Konzept als eine Transition modelliert werden kann. Darüber hinaus liegt eine offensichtliche Übereinstimmung mit dem problemtheoretischen Strukturierungsparadigma vor, das an früherer Stelle beschrieben wurde.

5) Prozessuales Wissen wird hier als flach bezeichnet, weil es nur die Zustandsveränderungen der modellierten Objekte beschreibt, die sich an der "Objektoberfläche" direkt beobachten lassen. Da dynamisches Wissen dagegen die "tieferliegenden" Kausalmechanismen repräsentiert, die das äußerlich beobachtete Objektverhalten erklären (sollen), wird es als tiefes Wissen angesprochen.

6) Fortan werden Lernprozesse bevorzugt thematisiert, weil sie als Erkenntnisgewinn von besonderem Interesse sind. Die nachfolgenden Ausführungen zu Wissensveränderungen gelten jedoch analog ebenso für Vergessensprozesse, auch wenn diese nicht explizit angesprochen werden.

7) In der Bezeichnung "epistemisches Wissen" könnte ein Pleonasmus gesehen werden. Tatsächlich ist dies jedoch nicht der Fall. Denn der Wissensbegriff bezieht sich auf die Ebene des *Meta*wissens, auf der lern- oder vergessensbedingte Veränderungen von Wissensinhalten repräsentiert werden. Das Attribut "epistemisch" referiert dagegen auf das *Objekt*wissen, das jenen Veränderungen unterliegt.

8) Zwecks Vereinfachung der Diktion werden diese interpretierten atomaren Grundtermformeln fortan auch in undifferenzierter Weise als Prädikatsvorkommnisse angesprochen.

9) Das Argument $(gt_1,...,gt_{K_u})$ der Grundtermformel $prä_u(gt_1,...,gt_{K_u})$ wird unter der Interpretation I_r auf das Konstantentupel $(ob_1,...,ob_{K_u})$ abgebildet. Daher nimmt die interpretierte Grundtermformel die Gestalt $prä_u(ob_1,...,ob_{K_u})$ an.

10) Eine solche Formel wird fortan auch als faktische Formel bezeichnet.

11) Diese Faktenmenge wird fortan auch vereinfacht als Faktenmenge des Prädikats $Prä_u$ angesprochen. Dabei wird auf die Vereinbarung zurückgegriffen, Prädikatssymbole unter den Prädikatsbegriff zu subsumieren.

12) Es könnte der Einwand erhoben werden, daß die Vereinigung von prädikatspezifischen Faktenmengen zu einem Informationsverlust führen kann. Denn durch die mengentheoretische Vereinigungsoperation gehen alle Informationen über mehrfache Vorkommnisse derselben Elemente in verschiedenen Mengen verloren. Dieser Sachverhalt spielt hier aber keine Rolle, weil bereits früher vereinbart wurde, daß nur paarweise verschiedene Prädikatsnamen verwendet werden. Daher sind die prädikatspezifischen Faktenmengen notwendig disjunkt. Folglich kann ihre Vereinigung niemals einen Informationsverlust bewirken.

13) Die logische Verknüpfung der Fakten geschieht durch die zusammengesetzten prädikatenlogischen Formeln, deren Bestandteile die atomaren faktischen Formeln $prä_u(ob_1,...,ob_{K_u})$ sind. Verknüpfungskonstituierend wirken hierbei die aussagenlogischen Konnektoren.

14) Der Einfachheit halber wird fortan auch davon gesprochen, daß die modellspezifische Faktenmenge FAK_r das aktuelle Objektwissen repräsentiert. Dabei wird implizit stets der Wissensbeitrag inbegriffen, der aus der logischen Faktenverknüpfung resultiert.

15) Dies folgt unmittelbar aus der Disjunktheit der prädikatspezifischen Faktenmengen, auf die bereits hingewiesen wurde, und aus deren Vereinigung zur modellspezifischen Faktenmenge.

16) Wenn die Faktenmenge durch kein Attribut näher spezifiziert wird, ist daher stets die modellspezifische Faktenmenge gemeint.

17) Der Informationsgehalt der Faktenmenge kann hier noch nicht präzise bestimmt werden. Er wird aber später durch die Differenzierung zwischen bekannten und unbekannten Wahrheitswerten für interpretierte Grundtermformeln konkretisiert.

18) Es wäre sehr aufwendig, stets beide Formulierungsweisen anzugeben. Darüber hinaus würde hierdurch infolge der Formulierungsäquivalenz kein Informationsgewinn erzielt. Daher wird jeweils nur eine der zwei Formulierungsalternativen explizit ausgeführt. Es wird diejenige Alternative gewählt, welche die einfachere und übersichtlichere Argumentation erlaubt. Es erfolgt bewußt keine Festlegung auf nur eine Formulierungsweise. Denn durch eine solche Einschränkung gingen Darstellungsreichtum und -flexibilität verloren, die durch die Äquivalenz von fakten- und extensionsbezogener Wissensrepräsentation eröffnet werden.

19) Hinsichtlich der Ausdehnung dieses Zeitabschnitts wird hier keine Festlegung getroffen. Fällt er punktförmig aus, erfolgen Veränderungen des Objektmodells zeitkontinuierlich. Wenn sich dagegen der Zeitabschnitt über ein endliches, von Null verschiedenes Zeitintervall erstreckt, liegt eine zeitdiskrete Objektmodellierung vor. Im Rahmen des Petrinetz-Konzepts wird aber stets der zweite Fall unterstellt.

20) Später wird dieser Modellzustand mit der Markierung M_r eines Netzmodells identifiziert. Unter Vorgriff auf diesen Sachverhalt wird bereits hier der Modellzustand durch das Symbol "r" vertreten.

21) Dabei wird unterstellt, daß die Objektveränderungen erstens vom Modellierungsträger als solche erkannt werden und zweitens ein modellierungsrelevantes Ausmaß erreichen. Falls die erste Voraussetzung verletzt wird, liegen unerkannte Objektveränderungen vor. Sie können in einem Modellierungskonzept grundsätzlich nicht erfaßt werden. Wenn dagegen erkannte, aber irrelevante Objektveränderungen erfolgen, impliziert das Irrelevanzurteil, solche Variationen nicht zu berücksichtigen.

22) Ebenso kommen Vergessensprozesse in Betracht. Sie werden hier jedoch nicht weiter explizit behandelt.

23) Der Übersichtlichkeit halber werden die beiden Veränderungsmodi hier voneinander deutlich unterschieden und nachfolgend getrennt behandelt. Dies stellt aber eine rein analytische Separation dar. Aus empirischer Perspektive besteht dagegen zwischen den beiden Veränderungsmodi ein enger, allerdings asymmetrischer Zusammenhang. Denn jede tatsächliche oder potentielle Objektveränderung kann in einem Modellierungsprozeß nur dadurch Relevanz erlangen, daß sie vom Modellierungsträger als solche auch erkannt wird. Daher ist eine operationsbedingte Modellveränderung immer mit einer kognitionsbedingten Modellveränderung verknüpft. Die Umkehrung gilt jedoch nicht. Denn die Objektwahrnehmung seitens des Modellierungsträgers kann sich verändern, ohne daß hiermit eine operationsbedingte Veränderung des modellierten Objekts verbunden sein müßte. Solche operationsunabhängigen Wahrnehmungsveränderungen geschehen vor allem dadurch, daß der Modellierungsträger durch Verfeinerung oder Vergröberung seiner Modellierungsperspektive die Art seiner Objektperzeption variiert.

24) Zwei Modellzustände heißen benachbart, wenn sie unmittelbar, d.h. ohne Dazwischenliegen eines dritten Modellzustands aufeinander folgen. Von zwei benachbarten Modellzuständen wird der im Zeitablauf frühere als Referenzzustand bezeichnet und mit dem Index "r" notiert. Der unmittelbar folgende Modellzustand heißt dagegen Folgezustand und wird durch den Index "f" angesprochen. Entsprechend heißen die zugehörigen Faktenmengen und Prädikatsextensionen Referenzfaktenmenge $FAK_{u,r}$ und Referenzextension $EXT_{u,r}$ bzw. Folgefaktenmenge $FAK_{u,f}$ und Folgeextension $EXT_{u,f}$. Als Regelfall wird mit $r \neq f$ unterstellt, daß Referenz- und Folgezustand voneinander verschieden sind. Der Grenzfall $r=f$ wird jedoch aus systematischen Gründen ebenso zugelassen. Hierdurch wird es möglich, den Veränderungsbegriff für Wissensinhalte so weit zu fassen, daß er auch die Wissenskonstanz umschließt.

25) Der Einfachheit halber werden diese Veränderungen fortan auch als Fakten- bzw. Extensionsvariationen angesprochen.

26) Dies kann einerseits durch Operationen bewirkt werden, die das modellierte Objekt so verändern, daß ein objektbeschreibendes, im Referenzzustand des Objektmodells gültiges Prädikatsvorkommnis im Folgezustand un-

gültig wird. Ebenso kommen kognitive Lernprozesse in Betracht, durch die ein Modellierungsträger zu der Einsicht gelangt, ein vormals für gültig gehaltenes Prädikatsvorkommnis müsse in einer realitätsgerechten Objektmodellierung als ungültig ausgewiesen werden.

27) Vgl. zu Konzepten für nonmonotone Logiken ZELEWSKI (1986a), S. 359ff., und die dort angeführten Quellen; RICHTER,M.M. (1988), S. 31f.

28) Die speziellen Inferenzmöglichkeiten und -schwierigkeiten nonmonotoner Logiken spielen dagegen bei der Anpassungsplanung im Rahmen des Opportunistischen Koordinierungskonzepts eine Rolle.

29) Wegen der nicht mehr gesicherten logischen Monotonie kann ein Prädikatsvorkommnis, das einmal als gültig erkannt worden ist, durch weitere Erkenntnisse nachträglich ungültig werden. Daher besitzen nonmonotone Logiken nicht mehr die Eigenschaft der Wahrheitserhaltung, die für alle deduktiven Logiken charakteristisch war. Dieser Verlust der Wahrheitserhaltung beim Übergang von "klassischen" (monotonen) zu nonmonotonen Logiken wird auch von REINFRANK (1985b), S. 29, herausgestellt.

30) Die derart modifizierte konventionelle Prädikatenlogik wird fortan auch kurz als Prädikatenlogik angesprochen.

31) Die Nuance besteht lediglich darin, die kategorische Formulierung "... gültig ist" durch die epistemische Reformulierung "... als gültig bekannt ist" zu substituieren. Hieran knüpfen sich aber beachtliche Konsequenzen, wie z.B. der - nachfolgend skizzierte - Übergang zu einer dreiwertigen Logik.

32) Dabei wird implizit die Perspektive eines imaginären Betrachters unterstellt, der über dem prädikatenlogisch formulierten Objektmodell steht. Er verfügt über das externe, im unvollständigen Wissen der Faktenmenge FAK_r des Objektmodells nicht explizit enthaltene Wissen, daß das Prädikatsvorkommnis $prä_u(ob_1,...,ob_{K_u})$ tatsächlich gültig ist.

33) Daher ist es eine Grundintention des hier bereits mehrfach angesprochenen Konzepts der logischen Programmierung, derart unvollständiges Wissen über die Wahrheitswerte von Prädikatsvorkommnissen $prä_u(ob_1,...,ob_{K_u})$ dadurch zu beseitigen, daß versucht wird, *alle* Konstantentupel $(ob_1,...,ob_{K_u})$ abzuleiten, für welche die jeweils betrachtete Grundtermformel $prä_u(gt_1,...,gt_{K_u})$ gültig ist. Vgl. dazu CORDES (1988), S. 17f. u. 65; KINNEBROCK (1988), S. 22f., 58 u. 62ff.

34) Ausgangspunkt ist eine Menge von Prädikatsvorkommnissen (Fakten), deren Gültigkeit von Anfang an vorausgesetzt wird. Dabei ist es unbeachtlich, ob ihre Gültigkeit tatsächlich bekannt oder nur hypothetisch unterstellt wird. Die deklarative Semantik dieser ersten Prädikatsvorkommnissemenge ist eine zweite Menge von Prädikatsvorkommnissen. Sie umfaßt alle Prädikatsvorkommnisse, die auf der Grundlage des modelltheoretischen Schlußfolgerungsbegriffs logische Konsequenzen der erstgenannten Menge von Prädikatsvorkommnissen darstellen. Der modelltheoretische Schlußfolgerungsbegriffs allein sagt noch nichts über die Gültigkeit der gefolgerten logischen Konsequenzen aus. Es legt nur fest, daß in allen prädikatenlogischen "Modellen", in denen die Prädikatsvorkommnisse der erstgenannten Menge gültig sind, auch deren logischen Konsequenzen gültig sind. Nun wurde aber eingangs für alle Prädikatsvorkommnisse der ersten Menge vorausgesetzt, *daß* sie gültig sind. Erst daraus folgt, daß auch alle Prädikatsvorkommnisse aus der zweiten Menge, die als logische Konsequenzen der ersten Prädikatsvorkommnissemenge ausgezeichnet wurden, ebenso gültig sein müssen. Auf alle Prädikatsvorkommnisse, die zur zweiten, aber *nicht* zur ersten Menge gehören, trifft zu: Ihre Gültigkeit ist nicht von Anfang an bekannt, sondern steht erst dann fest, nachdem es gelungen ist, sie als logische Konsequenzen von Prädikatsvorkommnissen aus der ersten Menge zu folgern. Auf genau diese Prädikatsvorkommnisse, deren Gültigkeit zunächst unbekannt ist, die sich aber nachträglich als gültig folgern lassen, beziehen sich die oben vorgetragenen Ausführungen. Vgl. dazu auch den Begriff der impliziten Fakten bei ESTER (1989), S. 27.

35) Es handelt sich um diejenigen Prädikatsvorkommnisse, deren Gültigkeit in der voranstehenden Anmerkung von Anfang an vorausgesetzt wurde. Vgl. auch ESTER (1989), S. 27, zum Begriff expliziter Fakten, die in Wissensbasen von vornherein als solche vorgehalten werden - und nicht erst abgeleitet werden müssen.

36) Vgl. zur Bedeutung dieser Differenzierung ZELEWSKI (1986a), S. 171 u. 342ff.; WINTER,RO. (1991), S. 213 (dort aber anhand der Differenzierung zwischen extensionalen und intensionalen Informationen thematisiert); ESTER (1989), S. 27.

37) Diese Einschränkung reflektiert zunächst, daß nur Wissen über die Wahrheitswerte von Prädikaten thematisiert wird. Wissen generell könnte sich auch auf vielfältige andere Sachverhalte erstrecken. Darüber hinaus erfolgt eine Einschränkung auf Wissen über den Wahrheitswert "gültig". Diese Restriktion wird später aufgehoben, indem auch explizites Wissen über den Wahrheitswert "ungültig" erfaßt wird.

38) Der Umfang des impliziten Wissens hängt also nicht nur vom jeweils vorgegebenen expliziten Wissen ab, sondern auch von der Inferenzregelmenge, die als logisch zulässig anerkannt wird. Hierüber können durchaus Meinungsunterschiede bestehen, wie z.B. im Hinblick auf die Zulässigkeit indirekter Beweise durch Annahme und Inkonsistenznachweis des kontradiktorischen Gegenteils; vgl. dazu die Anmerkungen zu nicht-konstruktiven Exi-

stenzbeweisen. Von solchen Problemen der Zulässigkeit von Inferenzregeln wird hier aber der Einfachheit halber abgesehen, indem eine bereits akzeptierte Inferenzregelmenge unterstellt wird. Auch das "negation by failure"-Prinzip, das nachfolgend angesprochen wird, läßt sich als eine Inferenzregel auffassen. Ihre Berechtigung ist durchaus umstritten; vgl. dazu die Problematisierungen des "negation by failure"-Prinzips.

39) Das implizite Wissen bezüglich eines vorgegebenen expliziten Wissens läßt sich daher als die deklarative Semantik des expliziten Wissens auffassen. Zugleich stellt das implizite Wissen die Menge aller logischen Schlußfolgerungen (Implikationen) aus dem vorgegebenen expliziten Wissen dar; vgl. dazu die Anmerkungen zur modelltheoretischen Variante der prädikatenlogischen Semantik. Da jedes Wissen sich selbst impliziert, ist das explizite Wissen stets eine Teilmenge seines zugehörigen impliziten Wissens.

40) Da die Inferenzregeln - zumindest im Rahmen des hier vorausgesetzten Paradigmas der deduktiven Logik - gültigkeitserhaltend wirken, ist es ausgeschlossen, durch Anwenden von Inferenzregeln das explizite Wissen über gültige Prädikatsvorkommnisse zu vermindern. Dies entspricht der zuvor angesprochenen Monotonieprämisse der konventionellen Prädikatenlogik. Das inferentielle Explizieren von implizitem Wissen verläßt also nicht diesen konventionellen prädikatenlogischen Rahmen.

41) Sie werden im hier entfalteten Argumentationskontext der Einfachheit halber als Prädikatsvorkommnisse angesprochen.

42) Das "negation by failure"-Prinzip konstatiert nicht unmittelbar die Ungültigkeit von Prädikatsvorkommnissen. Statt dessen wird ein Prädikatsvorkommnis $prä_u(ob_1,...,ob_{K_u})$ erst dann als ungültig festgesetzt, wenn zwei Bedingungen erfüllt sind. Erstens darf das Prädikatsvorkommnis $prä_u(ob_1,...,ob_{K_u})$ nicht explizit als Fakt vorliegen, dessen Gültigkeit per definitionem bekannt wäre. Zweitens muß ein Versuch fehlgeschlagen sein ("failure"), die Gültigkeit des Prädikatsvorkommnisses aus dem prädikatenlogischen Objektmodell durch Anwenden von Inferenzregeln herzuleiten. Falls diese beiden Voraussetzungen zutreffen, wird die Gültigkeit des Negats ("negation") des untersuchten Prädikatsvorkommnisses unterstellt. Aus diesem Blickwinkel stellt das "negation by failure"-Prinzip eine Inferenzregel sui generis dar. Denn es wird die Gültigkeit einer prädikatenlogischen Formel "erschlossen". Unter Voraussetzung des tertium non datur-Prinzips ist diese Negatgültigkeit mit der Ungültigkeit des betrachteten Prädikatsvorkommnisses äquivalent.
Das Inferenzergebnis einer Anwendung des "negation by failure"-Prinzips läßt sich grundsätzlich auf zwei Weisen darstellen: Es wird entweder die Gültigkeit des negierten Prädikatsvorkommnisses $\neg prä_u(ob_1,...,ob_{K_u})$ oder aber die Ungültigkeit des Prädikatsvorkommnisses $prä_u(ob_1,...,ob_{K_u})$ repräsentiert. Die letztgenannte Möglichkeit scheidet bei konventionellen prädikatenlogischen Objektmodellen aus. Denn das Objektwissen kann nur durch Fakten repräsentiert werden, die per definitionem gültige Prädikatsvorkommnisse darstellen. Daher muß die erste Alternative realisiert werden, die Ungültigkeit des Prädikatsvorkommnisses $prä_u(ob_1,...,ob_{K_u})$ mittelbar durch die Gültigkeit seines Negats $\neg prä_u(ob_1,...,ob_{K_u})$ zu repräsentieren. Dieses Negat wird als neu erschlossenes Fakt in die prädikatspezifische Faktenmenge $FAK_{u.r}$ aufgenommen.
Es könnte der Einwand erfolgen, daß die explizite Repräsentation von Negaten in prädikatenlogisch basierten Informationsverarbeitungssystemen oftmals technisch unmöglich sei. Dies trifft tatsächlich für zahlreiche frühe Varianten solcher Informationsverarbeitungssysteme zu. Dies gilt insbesondere auch dann, wenn sie mit Hilfe der Programmiersprache PROLOG implementiert wurden. Denn die HORN-Klauseln von PROLOG lassen es nicht zu, isolierte Negate oder Negate in den Prämissen oder Konklusionen von Subjugaten darzustellen. Vgl. zu diesen Implementierungsdefiziten bei Negaten JANAS (1979), S. 431 i.V.m. S. 430; AIDA (1983), S. 87f.; SHARPE (1985), S. 225; SERGOT (1986), S. 379; ZELEWSKI (1986a), S. 273f.; MURATA,TA. (1988b), S. 496. Dieser Negateinwand braucht jedoch heute keine Rolle mehr zu spielen. Denn es liegen neuere Implementierungen prädikatenlogischer Informationsverarbeitungssysteme vor, welche die explizite Darstellung von Negaten zulassen; vgl. AIDA (1983), S. 88f. Auch im Rahmen theoretischer Inferenzkonzepte wurde die Verwendung von explizit repräsentierten Negaten bereits behandelt; vgl. WATERMAN (1970), S. 126ff.; SIKLOSSY (1975), S. 44f.; KRAMOSIL (1975), S. 53ff. (die dort verwendete Nicht-Ableitbarkeit einer Formel kann als die Gültigkeit ihres Negats interpretiert werden; näher und differenzierter wird darauf bei CLARK,K. (1978), S. 297, eingegangen); SHAPIRO,S.C. (1979b), S. 796; ZELEWSKI (1986a), S. 275ff.

43) Es könnte der Einwand erhoben werden, daß in Netzmodellen das "negation by failure"-Prinzip sogar noch verschärft würde. Anlaß zu dieser Kritik wäre die spätere Einführung von Inhibitorkanten. Sie führen dazu, daß eine adjazente Transition nur dann aktiviert sein kann, wenn die Ursprungsstelle der Inhibitorkante unmarkiert ist, d.h. wenn ihre Faktenmenge leer ist. Dieser Sachverhalt könnte zum Anlaß genommen werden, der Netzmodellierung zu unterstellen, die Ungültigkeit von Prädikatsvorkommnissen mit leeren Faktenmengen zu identifizieren. Falls dieser Einschätzung zugestimmt würde, läge tatsächlich eine Verschärfung des "negation by failure"-Prinzips vor. Denn ein Prädikatsvorkommnis $prä_u(ob_1,...,ob_{K_u})$ würde auf diese Weise notwendig als ungültig erklärt, sobald es nicht zur aktuellen Faktenmenge $FAK_{u.r}$ seines Prädikatssymbols $Prä_u$ gehört. Dadurch würde die zweite Komponente des "negation by failure"-Prinzips nicht mehr beachtet, welche die Möglichkeit zuläßt, die Gültigkeit eines Prädikatsvorkommnisses indirekt aus anderen Formelgültigkeiten abzuleiten.

Tatsächlich trifft die voranstehend skizzierte Unterstellung jedoch nicht zu. Sie beruht vielmehr auf einer weder notwendigen noch fruchtbaren Interpretation der prädikatenlogischen Qualität von Inhibitorkanten. Denn Inhibitorkanten bedeuten zunächst nur, daß eine notwendige Voraussetzung für die Aktivierung einer Transition die Nichtmarkierung der betroffenen Eingangsstelle ist. Wie diese Nichtmarkierung interpretiert wird, bleibt vorerst offen. Daher braucht sie keineswegs als Ungültigkeit aller Prädikatsvorkommnisse für das Prädikatssymbol der betrachteten Eingangsstelle angenommen werden. Statt dessen läßt sich die Nichtmarkierung einer Stelle ebenso in dem Sinne auslegen, daß keine gültigen Vorkommnisse des stellenspezifischen Prädikatssymbols unter der aktuellen Netzmarkierung bekannt sind. Diese Interpretation wird vom Verf. gewählt, weil sie sich kohärent in die oben thematisierte Behandlung von temporärem Nichtwissen und Lernprozessen einfügt. Die Festsetzung, die Nichtmarkierung einer Stelle als Unbekanntheit von gültigen Prädikatsvorkommnissen zu deuten, drückt sowohl das aktuelle Nichtwissen über aktuell gültige Prädikatsvorkommnisse als auch die Möglichkeit aus, durch Lernprozesse zukünftig doch noch gültige Prädikatsvorkommnisse kennenzulernen.

44) Dieser Rahmen wird durch Übergang zu einer dreiwertigen Logik verlassen.

45) Die Möglichkeit, die Gültigkeit einer Formel in einem Beweissystem durch Anwenden von Inferenzregeln indirekt nachzuweisen, wird mitunter übersehen; vgl. zu einer solchen Vernachlässigung indirekter Gültigkeitsnachweise z.B. KLEINHANS (1989), S. 88.

46) Dies stimmt mit der Feststellung von RUSPINI (1987a), S. 232f., überein, jede Formel als ungültig zu betrachten, sobald sie in einem Modell nicht explizit repräsentiert wird: "explicit absence of a representation for a statement is equivalent to its falsehood" (S. 232). Auch KLEINHANS (1989), S. 88, legt fest: "Alle nicht explizit gespeicherten Aussagen sind falsch". Die gleiche Ansicht vertritt NAKANO,R. (1983), S. 557 (in bezug auf geschlossene Weltmodellierungen). Eine etwas verschwommene Formulierung von SERGOT (1986), S. 379, kann ebenfalls in der Weise ausgelegt werden, daß aus der Nichtrepräsentation einer Formel deren Ungültigkeit gefolgert werden solle: "Anything which is not known is assumed to be false." (kursive Hervorhebungen im Original hier unterlassen).

47) Statt dessen kann aber durchaus auch der Standpunkt bezogen werden, Petrinetze als netztheoretische Darstellung von Inferenzmöglichkeiten zu begreifen. Dabei wird von Petrinetzen ausgegangen, die jeweils eine logische Problembeschreibung repräsentieren. Dann umfaßt die Erreichbarkeitsrelation des Petrinetzes genau alle zulässigen Inferenzen, die in jener Problemrepräsentation auf Netzbasis ausgeführt werden können. Vgl. zu dieser inferenzbezogenen Anwendung von Petrinetzen LIU,N. (1987), S. 121ff.; SHIH (1987), S. 48ff.; SAHRAOUI (1987), S. 1161ff.; MURATA,TA. (1987a), S. 381ff.; MURATA,TA. (1988a), S. 73ff.; ZELEWSKI (1989c), S. 17ff., insbesondere S. 25ff.

48) Diese Rigidität bedeutet eine Fehlerquelle, die durch kein prädikatenlogisches Inferenzkonzept unter realen Realisierungsbedingungen - d.h. bei beschränkten Ressourcen für die Ausführung von Inferenzprozeduren - beseitigt werden kann. Die Fehlerquelle resultiert aus der Inkorrektheit des "negation by failure"-Prinzips bei Ressourcenbeschränkungen. Sie läßt sich kurz dadurch charakterisieren, daß beim ressourcenbedingten Abbrechen einer Inferenzprozedur immer die Gültigkeit des Negats eines untersuchten Prädikats angenommen wird, obwohl das Prädikat in Wirklichkeit auch gültig - also sein Negat ungültig - sein kann. Dies konnte aber innerhalb der realen Ressourcenbeschränkungen nicht erkannt werden.

49) Vgl. dazu auch die Kritik an den epistemischen Defiziten des "negation by failure"-Prinzips bei REINFRANK (1985b), S. 37; ZELEWSKI (1986a), S. 274 u. 948f.; WEDEKIND (1989c), S. 26f. Aufgrund solcher Kritik wird der indirekten Erschließung der Ungültigkeit einer Formel mittels des "negation by failure"-Prinzips eine geringere epistemische Qualität zuerkannt als der Ableitung einer Formelgültigkeit; vgl. WALKER,A. (1983), S. 526f.; ähnlich auch REITER (1978b), S. 56ff.
Bei REINFRANK (1985b), S. 37, findet sich ein ebenso einfaches wie überzeugendes Beispiel für die epistemische Problematik des "negation by failure"-Prinzips: Es werden zwei Formeln $prä_1$ und $prä_2$ betrachtet, von denen nur bekannt ist, daß ihr Adjugat $prä_1 \vee prä_2$ gültig ist. Es wird versucht, aus diesem Adjugat die Gültigkeit zunächst der Formel $prä_1$ und danach der Formel $prä_2$ abzuleiten. Beide nacheinander ausgeführten Ableitungsversuche müssen scheitern, da ein entsprechende Schlußfolgerung logisch unzulässig wäre. Aufgrund des "negation by failure"-Prinzips werden jedoch aus dem zweimaligen Scheitern die zusätzlichen Erkenntnisse "gefolgert", sowohl das Negat $\neg prä_1$ als auch das Negat $\neg prä_2$ seien gültig. Aus der Gültigkeit zweier Formeln folgt immer die Gültigkeit ihres Konjugats. Also muß auch die Formel $(\neg prä_1) \wedge (\neg prä_2)$ gültig sein. Dieses Konjugat ist aufgrund der DE MORGAN-Regeln äquivalent mit dem negierten Adjugat $\neg(prä_1 \vee prä_2)$. Die Gültigkeit des negierten Adjugats $\neg(prä_1 \vee prä_2)$ und die eingangs vorausgesetzte Gültigkeit des Adjugats $prä_1 \vee prä_2$ stellen aber eine logische Kontradiktion dar. Folglich hat die Anwendung des "negation by failure"-Prinzips zu einer "Folgerung" verleitet, die in einen logischen Widerspruch mündet. Die Widerspruchsfreiheit wurde jedoch als eine Basisprämisse für alle Argumentationsformen vorausgesetzt. Die Verletzung dieser Basisprämisse läßt den epistemischen Defekt des "negation by failure"-Prinzips besonders deutlich hervortreten.

50) Das Entsprechungsverhältnis zwischen "negation by failure"-Prinzip und geschlossener Weltmodellierung besitzt die Qualität einer wechselseitigen Korrespondenz, aber nicht einer Identität. Denn das "negation by failure"-

Prinzip gehört zur Sphäre der Epistemologie, während die geschlossene Weltmodellierung ein ontologisches Prinzip darstellt. Das "negation by failure"-Prinzip erstreckt sich auf (gescheiterte) *Inferenzprozesse*. Die geschlossene Weltmodellierung betrifft dagegen die (Nicht-)Existenz von Sachverhalten, die in einem Objektmodell weder direkt noch indirekt enthalten sind.
Die Differenz zwischen epistemologischer und ontologischer Perspektive läßt sich auch in folgender Weise ausdrücken: Das "negation by failure"-Prinzip beruht auf einer *unvollständigen Lösungssuche* im abstrakten Lösungsraum eines Inferenzproblems. Die geschlossene Weltmodellierung bezieht sich dagegen auf eine *unvollständige Weltabbildung* durch das Modell eines Realitätsausschnitts. Die Korrespondenz zwischen der epistemologischen und der ontologischen Perspektive kommt dadurch zustande, daß jeder direkt modellierte Sachverhalt epistemologisch unmittelbar gegeben ist und jeder indirekt modellierte Sachverhalt durch Inferenzprozeduren in einem prädikatenlogisch formulierten Modell erschlossen werden kann. Folglich entsprechen bei einer geschlossenen Weltmodellierung alle Sachverhalte, die in einem Modell weder direkt noch indirekt enthalten sind, genau denjenigen Sachverhalten, die weder unmittelbar gegeben noch durch Inferenzen ableitbar sind.

51) Die Prämisse der geschlossenen Weltmodellierung wird meistens nicht explizit erwähnt. Dennoch liegt sie allen konventionellen Modellierungskonzepten implizit zugrunde. Dies bestätigt Ruspini (1987a), S. 232: "facts ... are ... first order predicate calculus expressions ... *implicitly* assumed to be true descriptors of the state of the real world" (kursive Hervorhebung durch den Verf.). Hierdurch wird die früher vorgetragene allgemeine Einschätzung unterstrichen, daß eine vollständige Prämissenoffenlegung zwar in der Modellierungstheorie angestrebt werden mag, aber in der Modellierungspraxis kaum jemals konsequent verwirklicht wird. Vgl. darüber hinaus zum Vorherrschen geschlossener Weltmodellierungen Nakano,R. (1983), S. 557 ("The closed world assumption is appropriate in most applications ...").

52) Vgl. zur Prämisse geschlossener Weltmodellierungen (closed world assumption) Reiter (1978b), S. 56 u. 59ff.; Clark,K. (1978), S. 294; Apt (1982), S. 858; Nakano,R. (1983), S. 557; Wilkins (1984), S. 275; Hertzberg (1985), S. 13f.; Reinfrank (1985b), S. 36f.; Sergot (1986), S. 379; Zelewski (1986a), S. 274; Kowalski (1987a), S. 136; Ruspini (1987a), S. 232f.; Richter,M.M. (1988), S. 20; Fidelak (1988b), S. 15; Kleinhans (1989), S. 87f.; Moerkotte (1990), S. 21.

53) Dieser Realitäts*ausschnitt* wird aus modelltheoretischem Blickwinkel oftmals - in nicht ernsthaft gemeinter Übertreibung - als "Welt" thematisiert.

54) Das Erschließen von impliziten Wissensbestandteilen wird mitunter nicht zur geschlossenen Weltmodellierung gerechnet. Statt dessen wird die Ansicht vertreten, daß bei einer geschlossenen Weltmodellierung jede atomare Formel, die nicht explizit als Fakt ausgewiesen wird, ungültig ist. Vgl. dazu beispielsweise Nakano,R. (1983), S. 557. Diese Auffassung ist jedoch unnötig eng. Denn sie übersieht, daß z.B. im Rahmen deduktiver Datenbanksysteme die Gültigkeit atomarer Formeln auch dann noch mit der Hilfe von Inferenzen erschlossen werden kann, wenn diese Formeln in den Datenbanken nicht unmittelbar als explizite Fakten ausgewiesen sind. Allerdings lag diese enge Auslegung bereits der Explizitheitsprämisse zugrunde, die kurz zuvor erläutert wurde.

55) Die Extensionalität bezieht sich auf die Prädikatsextensionen $EXT_{u,r}$ aller Prädikate $Prä_u$ eines prädikatenlogischen Objektmodells. Mit diesen Prädikatsextensionen sind die Faktenmengen $FAK_{u,r}$ äquivalent. Diese Faktenmengen enthalten alle Prädikatsvorkommnisse $prä_u(ob_1,...,ob_{K_u})$, deren Gültigkeit unter der Interpretation I_r bekannt ist. Daher korrespondiert die Extensionalität der Modellierungsweise qua Prädikatsextensionen mit dem explizit repräsentierten Wissen über gültige Prädikatsvorkommnisse.

56) Das Wissen, das ein Objektmodell über die Gültigkeit von Prädikatsvorkommnissen enthält, ist über die expliziten Faktenmengen einerseits und das implizite Modellwissen andererseits verteilt. In der Aufteilung zwischen explizitem und implizitem Wissensanteil liegt das Ausmaß der Explizitheit der Wissensrepräsentation. Erfolgt diese Aufteilung unter bewußter Abwägung zwischen dem Unmittelbarkeitsvorzug expliziter und dem Kompaktheitsvorteil impliziter Wissensrepräsentation, so wird von einer kontrollierten Explizitheit gesprochen.

57) Wittgenstein (1921), S. 217f., Punkt 4.116, hat in seinem Frühwerk diese extensionale Einstellung markant in dem Motto zusammengefaßt: "Alles was überhaupt gedacht werden kann, kann *klar* gedacht werden. Alles was sich aussprechen läßt, läßt sich *klar aussprechen.*" (kursive Hervorhebungen durch den Verf.). Wittgenstein's Adverb "klar" wird vom Verf. durch die Unmittelbarkeit der expliziten Wissensrepräsentation und durch die Stringenz des inferentiellen Erschließens impliziter Wissensbestandteile interpretiert.

58) Durch inferentielles Erforschen der logischen Konsequenzen einer Faktenmenge kann nachträglich die Gültigkeit oder Ungültigkeit von Formeln erkannt werden, die in der Faktenmenge selbst nicht enthalten sind.

59) Offene Weltmodellierungen zeichnen sich dadurch aus, daß der Gültigkeitsstatus einer interpretierten atomaren Grundtermformel so lange unbestimmt ("unbekannt") bleibt, wie gilt: Die Grundtermformel ist einerseits in der Faktenmenge nicht unmittelbar als gültige atomare Formel explizit ausgewiesen. Andererseits ist es auch (noch) nicht gelungen, entweder die Gültigkeit oder aber die Ungültigkeit der Grundtermformel dadurch zu erschließen, daß sie selbst bzw. ihr Negat als eine logische Konsequenz der Faktenmenge nachgewiesen wurde. Vgl. zum Prinzip

der offenen Weltmodellierung (open world assumption) REITER (1978b), S. 55f. u. 61; KOWALSKI (1978), S. 95ff.; WALKER,A. (1983), S. 526; NAKANO,R. (1983), S. 557; ZELEWSKI (1986a), S. 274.
Für offene Weltmodellierungen existieren durchaus theoretische Konzepte. Sie zeichnen sich dadurch aus, daß die Objektmodelle von vornherein im Bewußtsein ihrer unvollständigen Weltabbildung entworfen werden. Vgl. dazu MYLOPOULOS (1983), S. 150f. Dabei erstreckt sich die Unvollständigkeit nicht nur auf die modellierungsirrelevanten Weltaspekte. Vielmehr betrifft sie vor allem gerade jene Komponenten, die zum relevanten Weltausschnitt gehören, deren realen Ausprägungen im Modellierungszeitpunkt aber noch nicht bekannt sind. Für solche relevanten, aber hinsichtlich ihrer realen Ausprägung vorläufig unbekannten Modellkomponenten werden spezielle Konzepte vorgehalten. Sie gestatten es, mit solchen Komponenten sowohl während der Unbekanntheit ihrer realen Ausprägungen als auch nach deren Bekanntwerden in systematischer Weise umzugehen. Zu diesen Konzepten für offene Weltmodellierungen gehören z.B. das "default reasoning" und die "assumption based truth maintenance"-Systeme. Sie werden später im Rahmen des opportunistischen Steuerungskonzepts angesprochen. Vgl. auch WEDEKIND (1989c), S. 25f. Dort wird die Entwicklung von dialog- und transaktionsorientierten Datenbanksystemen für die betriebliche Informationsverarbeitung erörtert. Dabei wird das "negation by failure"-Prinzip auf der Basis einer konstruktivistisch-intuitionistischen Logik verworfen. Statt dessen wird eine offene Weltmodellierung in Datenbanksystemen angeregt, die sich u.a. auf das oben angesprochene Konzept der "assumption based truth maintenance"-Systeme stützt.

60) Es ließe sich darüber streiten, ob der Begriff BOOL'scher Mengen auf konventionelle zweiwertige Logiken mit den Wahrheitswertemengen $BOOL_{2.aus}$ = {wahr,falsch} oder $BOOL_{2.prä}$ = {gültig,ungültig} beschränkt ist. Der Verf. faßt diesen Begriff so weit auf, daß er sich auf die beiden vorgenannten Wahrheitswertemengen und alle unmittelbaren Derivate bezieht. Dies entspricht der weit gefaßten aussagenlogischen Semantik von RICHTER,M.M. (1978), S. 41 i.V.m. S. 39. Er läßt als Wahrheitswertemenge eine beliebige Trägermenge zu, sofern sie nur die Spezifikation BOOL'scher Algebren erfüllt. Die BOOL'schen Algebren werden bei RICHTER,M.M. (1978), S. 22ff., ausführlich und präzise definiert.

61) Das tertium non datur-Prinzip drückt die Wahrheitsdefinitheit aller Formeln bezüglich einer zweiwertigen Logik aus: Jede Formel ist entweder gültig oder ungültig - ein aliud existiert nicht; vgl. HEYTING (1931), S. 114; CARNAP (1960a), S. 26 (L8-1a u. L8-1b); ESSER,H. (1977a), S. 37a; RAUTENBERG (1979), S. 8; GETHMANN (1980b), S. 36; MITTELSTAEDT (1983), S. 27f.; VON WRIGHT (1986), S. 3f.; WEDEKIND (1989c), S. 22.

62) Dies entspricht der Kritik des logischen Konstruktivismus, es sei unzulässig, in Beweisen auf dieses Prinzip zurückzugreifen. Vgl. zu dieser Ablehnung des tertium non datur-Prinzips FRAENKEL (1930), S. 291f.; HEYTING (1931), S. 107; ACKERMANN,W. (1957), S. 11; MENDELSON (1964), S. 4f.; HEYTING (1968a), S. 313; KÖRNER,S. (1968), S. 144 u. 158; MYHILL (1968), S. 326; STEGMÜLLER (1976b), S. 438ff.; MITTELSTAEDT (1983), S. 34f. u. 42f.; FITTING (1987), S. 400f.; WEDEKIND (1989c), S. 23f. u. 27; vgl. zu weiteren, aber nicht konstruktivistisch motivierten Logikkonzeptionen, die ohne das tertium non datur-Prinzip auskommen, VON WRIGHT (1986), S. 5ff.; BACK,R. (1986), S. 15ff.
Vgl. darüber hinaus die allgemeinen Erläuterungen zum logischen Konstruktivismus und die dort angeführten Quellen. Die konstruktivistische Kritik korrespondiert mit Erkenntnissen der Realwissenschaften, daß im Bereich der Quantentheorie das tertium non datur-Prinzip ebenfalls nicht aufrechterhalten werden kann; vgl. VON WEIZSÄCKER (1985), S. 322 u. 325.

63) Allerdings wird am tertium non datur-Prinzip innerhalb des Bereichs *bekannter* Formelgültigkeiten festgehalten: Wenn bekannt ist, daß ein Prädikatsvorkommnis $prä_u(ob_1,...,ob_{K_u})$ gültig (ungültig) ist, dann folgt daraus stets die Ungültigkeit (Gültigkeit) der Negation $\neg prä_u(ob_1,...,ob_{K_u})$.
Grundsätzlich besteht die Alternative, den Wahrheitswert des Negats $\neg prä_u(ob_1,...,ob_{K_u})$ als unbestimmt zu qualifizieren, wenn die Ungültigkeit des Prädikats $prä_u(ob_1,...,ob_{K_u})$ bekannt ist. Dies ließe die Möglichkeit zu, daß bei späterem Informationsgewinn die Ungültigkeit sowohl des Prädikats $prä_u(ob_1,...,ob_{K_u})$ als auch seines Negats $\neg prä_u(ob_1,...,ob_{K_u})$ erkannt wird. Dies bedeutet strenggenommen noch keine Inkonsistenz. Denn ein logischer Widerspruch läge erst dann vor, wenn zugleich die Gültigkeit eines Prädikats und die Gültigkeit seines Negats behauptet wird. Hier werden aber nicht deren simultanen Gültigkeiten, sondern nur deren simultanen Ungültigkeiten zugelassen. Da das tertium non datur-Prinzip außer Kraft gesetzt ist, läßt sich aus Formelungültigkeiten aber nicht die Gültigkeit der jeweils negierten Formeln ableiten. Daher wird die Konsistenzbedingung, daß niemals eine Formel und ihr Negat zugleich gültig sein dürfen, durch die simultane Ungültigkeit des Prädikats $prä_u(ob_1,...,ob_{K_u})$ und seines Negats $\neg prä_u(ob_1,...,ob_{K_u})$ keineswegs verletzt.
Dies entspricht der Einstellung von Anhängern der konstruktivistisch-intuitionistischen Logik, daß der Nachweis der Ungültigkeit einer Formel noch keineswegs einen überzeugenden Beweis für die Gültigkeit der negierten Formel liefere, weil der Gültigkeitsbeweis nicht effektiv konstruiert worden sei. Darauf wurde bereits im Kontext der indirekten, nicht-konstruktiven Beweistechnik hingewiesen. Aus dieser Perspektive bleibt der Wahrheitswert eines Negats $\neg prä_u(ob_1,...,ob_{K_u})$ solange unbekannt, wie nicht diese Formel selbst mittels eines konstruktiven Verfahrens als gültig oder ungültig nachgewiesen wird. Dieser Wahrheitswert "unbekannt" verändert sich auch dann nicht, wenn das Prädikat $prä_u(ob_1,...,ob_{K_u})$ als ungültig erkannt wird. Erst wenn sich jenes Prädikat als gültig herausstellt

und zusätzlich die Konsistenz des gesamten Formelsystems vorausgesetzt wird, akzeptiert auch die konstruktivistisch-intuitionistische Auffassung, das Negat $\neg prä_u(ob_1,...,ob_{K_u})$ als gültig festzusetzen. Andernfalls würde nämlich die simultane Gültigkeit des Prädikats $prä_u(ob_1,...,ob_{K_u})$ und seines Negats $\neg prä_u(ob_1,...,ob_{K_u})$ in Kauf genommen. Diese Inkonsistenz lehnt auch der logische Konstruktivismus ab.
Dennoch klammert der Verf. die voranstehend skizzierte Denkalternative aus seinen weiteren Überlegungen aus. Denn bei dem hier interessierenden Modellierungskonzept spielen die Ansichten des logischen Konstruktivismus über die Zulässigkeit von *Beweistechniken* keine Rolle. Vgl. HEYTING (1931), S. 114, zur Absicht des logischen Konstruktivismus, seine Kritik am tertium non datur-Prinzip auf *reine* Beweisführungskontexte zu beziehen. Vielmehr geht es in dieser Arbeit nur um die prädikatenlogische *Repräsentation* von Wissen über die jeweils modellierten Objekte. Aus dieser Repräsentationsperspektive bleibt aber die epistemische Frage, welche *positive* denotationale Bedeutung die gleichzeitige Ungültigkeit eines Prädikats und seines Negats besitzen könnte, vollkommen im dunkeln. Die mögliche quantentheoretische Ausdeutung, die sich bei VON WEIZSÄCKER (1985), S. 325, für den subatomaren Mikrokosmos findet, kommt für den betriebswirtschaftlich interessanten Bereich des Mesokosmos nicht in Betracht.
Es könnte allerdings eingewendet werden, daß die - vom Verf. geteilte - konstruktivistische Kritik an indirekten Beweisführungen auf der Basis des tertium non datur-Prinzips dazu führe, das Resolutionskonzept nicht mehr anwenden zu dürfen. Denn dieses Inferenzkonzept basiert auf dem indirekten Refutationsansatz. Damit werde aber eine wesentliche Grundlage für die Termauswertung in Synthetischen Netzen zerstört. Denn solche Auswertungen wurden auf das kombinierte Unifizierungs- und Resolutionskonzept zurückgeführt. Dieser Einwand trifft jedoch nicht zu. Vielmehr kann das Resolutionskonzept mit den strengen Beweistechniken des logischen Konstruktivismus so reformuliert werden, daß seine Vollständigkeit und Korrektheit erhalten bleiben, ohne auf das tertium non datur-Prinzip zurückgreifen zu müssen; vgl. FITTING (1987), S. 402ff. Da diese Resolutionsmodifizierung jedoch komplizierter als die konventionellen Resolutionsvarianten ausfällt, wird es hier nur angeführt, um die Verträglichkeit des Resolutionskonzepts mit dem logischen Konstruktivismus zu demonstrieren. Bei der praktischen Anwendung des Resolutionskonzepts wird im Interesse der Ausführungseffizienz weiterhin auf seine etablierten PROLOG-Implementierungen zurückgegriffen.

64) Vgl. zu Hinweisen auf solche dreiwertigen Prädikatenlogiken - über die beiden nachfolgend genannten Autoren - hinaus z.B. STEGMÜLLER (1970a), S. 274f.; COOMANN (1983), S. 115; BUNGE (1985a), S. 62; STEGMÜLLER (1986a), S. 185ff.; vgl. des weiteren RAUTENBERG (1979), S. 105ff., zu einer dreiwertigen Aussagenlogik.

65) Vgl. BLAU (1973), S. 20ff.; BLAU (1985), S. 370ff., insbesondere S. 377ff., hinsichtlich der dort u.a. untersuchten Logik "L3"; STEGMÜLLER (1986a), S. 186ff.

66) Vgl. KÖRNER,S. (1970), S. 56ff., insbesondere S. 60ff.; CLEAVE (1987), S. 137ff.

67) Eine exemplarische Verdeutlichung der Subtilität erfolgt an anderer Stelle hinsichtlich des Wahrheitswertes von Konjugaten aus komplementären Prädikaten.

68) Bei BLAU und KÖRNER dominiert zwar die Bezeichnung "neutral" für den dritten Wahrheitswert. Doch findet sich vereinzelt auch die synonyme Verwendung von "unbestimmt"; vgl. z.B. KÖRNER,S. (1970), S. 60. Die gleiche Wahrheitswertinterpretation der Unbestimmtheit präsentieren RAUTENBERG (1979), S. 105; BUNGE (1985a), S. 62. Im folgenden wird diese Interpretation vorausgesetzt. Dies gilt auch dann, wenn im Original eine andere Bezeichnung für den dritten Wahrheitswert gewählt wurde.

69) BLAU deutet den Wahrheitswert "unbestimmt" entweder als Vagheit des Wissens über die Zugehörigkeit von Individuen zur Extension eines Prädikats. Oder ein Prädikat wird als unbestimmt qualifiziert, weil es kategorial beschränkt ist, d.h. eine verletzte Präsupposition ausdrückt. Vgl. dazu vor allem BLAU (1985), S. 374f.
Eine Präsuppositionsverletzung würde z.B. die Behauptung bedeuten: "Die Programmiersprache PROLOG ist wegen der Entscheidbarkeit der Prädikatenlogik 1. Stufe für die Lösung von Entscheidungsproblemen grundsätzlich ohne Einschränkungen geeignet." Denn die unterstellte ("präsupponierte") prädikatenlogische Entscheidbarkeit existiert tatsächlich nicht. Näheres zu solchen Präsuppositionen und ihren Verletzungen findet sich bei GETHMANN (1980b), S. 28; WAHLSTER (1981), S. 23ff.; MORIK (1982), S. 123ff.; WAHLSTER (1982), S. 253f.; BLAU (1985), S. 74f.; ZELEWSKI (1986a), S. 473; GREWENDORF (1987), S. 421ff.
KÖRNER verwendet seinen Wahrheitswert "unbestimmt" ähnlich wie in BLAUs erster Variante der Vagheit. KÖRNER benutzt den dritten Wahrheitswert, um inexakte (einstellige) Prädikate zu formulieren, die jeweils inexakte Objektklassen definieren. Zwischen Vagheit und Inexaktheit besteht nur eine terminologische Nuancierung.

70) Diese Reformulierung wäre allerdings keinesfalls trivial. Vielmehr erforderte sie die Berücksichtigung spezieller logischer Probleme, die erst daraus resultieren, daß auch die Unbekanntheit von definiten Wahrheitswerten als eigener - indefiniter - Wahrheitswert eingeführt wird. Beispielsweise kann sowohl der Wahrheitswert eines Prädikats als auch der Wahrheitswert seines Negats unbestimmt sein. In einer "naiven" dreiwertigen Logik würde dann auch der Wahrheitswert des Konjugats dieser beiden komplementären Prädikate als unbestimmt festgesetzt. Eine inhaltliche Betrachtung zeigt jedoch sofort, daß das Konjugat eine Kontradiktion darstellt, die notwendig den definiten Wahrheitswert "ungültig" besitzt. Auf diese Schwierigkeit hat bereits BLAU (1985), S. 376, hingewiesen.

71) BLAU (1985), S. 376, wendet sich explizit gegen die Interpretation "unbekannt" für den dritten Wahrheitswert. Allerdings hält der Verf. die dort vorgetragenen Bedenken nicht für stichhaltig. Denn sie gelten nur für die "naiven" dreiwertigen Logiken, die bereits in einer früheren Anmerkung angesprochen wurden. Wenn die speziellen logischen Folgen der o.a. Wahrheitswertinterpretation berücksichtigt werden, stehen ihr nach Einschätzung des Verf. keine grundsätzlichen logischen Schwierigkeiten im Wege.

72) Vgl. KÖRNER,S. (1970), S. 65f. u. 68.

73) Für die Repräsentation des dritten Wahrheitswerts "unbekannt" besteht dagegen ein Freiheitsgrad. Entweder wird er ebenso explizit dargestellt. Oder er wird auf indirekte Weise durch die Konvention repräsentiert, daß alle Formeln, deren Wahrheitswerte nicht explizit als gültig oder ungültig ausgewiesen sind, den Wahrheitswert "unbekannt" erhalten. Letztes entspricht einer Modifizierung des tertium non datur-Prinzips der zweiwertigen Logik zu einem quartum non datur-Prinzip der dreiwertigen Logik. Der Verf. sieht einerseits keinen Anreiz, auf eine mehr als dreiwertige Logik überzugehen, die ein Verwerfen des quartum non datur-Prinzips erfordern würde. Andererseits bedeutet die implizite Repräsentation des Wahrheitswerts "unbekannt" eine erhebliche Vereinfachung der expliziten Darstellung, da anstelle von drei nur zwei Wahrheitswerte gehandhabt werden müssen. Daher unterstellt er fortan die implizite Darstellungsalternative. In der gleichen Weise legt sich WALKER,A. (1983), S. 526 fest ("a fact which is not listed is considered to be unknown").

74) Diese Vorgehensweise ist im Rahmen der logischen Programmierung zwar unüblich, kann aber grundsätzlich mit ihr konsistent vereinbart werden.

75) Dieser weit gefaßte Faktenbegriff, der auch ungültige Prädikatsvorkommnisse als Fakten zuläßt, deckt sich mit den entsprechend allgemein gehaltenen Faktendefinitionen von HARMON (1985), S. 48: "Facts take the form of logical expressions that consist of predicates and [truth] values." (Ergänzung [...] durch den Verf.), und RUSPINI (1987a), S. 233 ("Facts ... are truth-qualified propositions ..."). WALKER,A. (1983), S. 526, spricht sogar explizit von "both positive and negative facts".
Der Begriff der "negativen" Fakten wird in dieser Arbeit weit aufgefaßt. Im hier diskutierten Kontext bezieht er sich nur auf die Ungültigkeit von atomaren Prädikatsvorkommnissen. Die Ungültigkeit eines solchen Prädikatsvorkommnisses $prä_u(ob_1,...,ob_{K_u})$ verhält sich aber äquivalent dazu, daß das Negat des Prädikatsvorkommnisses - die Formel $\neg prä_u(ob_1,...,ob_{K_u})$ - gültig ist. Folglich kann unter einem "negativen" Fakt ebenso die Gültigkeit eines negierten atomaren Prädikatsvorkommnisses $\neg prä_u(ob_1,...,ob_{K_u})$ verstanden werden. Das zuletzt genannte Verständnis "negativer" Fakten lag z.B. der Erläuterung zu HORN-Klauseln zugrunde. Allerdings wurde dort der metasprachliche Aspekt der Formelgültigkeit nicht besonders hervorgehoben.

76) Sie gelten unabhängig davon, ob mit konventionellen Faktenmengen $FAK_{u.r}$ oder aber mit disjunkten Faktenmengen $FAK_{u.r}^+$ und $FAK_{u.r}^-$ gearbeitet wird.

77) LIU und DILLON haben angeregt, zwei disjunkte Markenkategorien zu verwenden; vgl. LIU,N. (1987), S. 121ff., insbesondere S. 122; ZELEWSKI (1989c), S. 79f. Die eine Markenkategorie dient im konventionellen Sinne dazu, die Gültigkeit von Prädikaten durch entsprechende Markierung von prädikatspezifischen Stellen abzubilden. Marken der zweiten Markenkategorie repräsentieren dagegen auf denselben Stellen jeweils die Ungültigkeit der zugehörigen Prädikate. Zwar bleiben LIU und DILLON noch weitgehend in einer aussagenlogischen Darstellungsweise verhaftet, doch läßt sich ihr Konzept grundsätzlich auf prädikatenlogische Kontexte übertragen. Ein anderes Konzept der Petrinetz-Theorie, das sich zwecks Abbildung positiver und negativer Fakten näher untersuchen ließe, erstreckt sich auf die Arbeiten über bipolare Synchronisationssysteme. Auch dort werden zwei unterschiedliche Markenkategorien verwendet.

78) Gleiches gilt für Vergessensprozesse. Sie wurden jedoch schon für diese Ausarbeitung als bedeutungslos ausgeklammert.

79) Auch der komplementäre Aspekt des Wissensverlusts durch Vergessensprozesse läßt sich repräsentieren. Dies geschieht durch den Übergang von einem der bestimmten Wahrheitswerte "gültig" oder "ungültig" zum Wahrheitswert "unbestimmt" für dasselbe Prädikatsvorkommnis $prä_u(ob_1,...,ob_{K_u})$ und entsprechende Umfangsverringerung der prädikatspezifischen Faktenmenge $FAK_{u.r}$.

80) Vgl. auch BUNGE (1985a), S. 62 u. 73ff., der diesen Kompatibilitätsvorteil - neben anderen Argumenten zugunsten der klassischen zweiwertigen Logik - anführt.

81) Vgl. WEDEKIND (1989c), S. 23f.

82) Solche Desintegrationsprobleme nicht-klassischer Logikkonzeptionen klingen auch bei BUNGE (1985a), S. 75, an.

83) Vgl. dazu die entsprechende Interpretation der unmarkierten Eingangsstellen von Inhibitorkanten.

84) Dies ist nicht mit dem Wahrheitswert "unbestimmt" der dreiwertigen Logik zu verwechseln. Denn in der hier verwendeten zweiwertigen Semantik bedeutet die Unbekanntheit des Wahrheitswertes nur, daß nicht feststeht, ob das Prädikatssymbol $Prä_u$ für das betrachtete Objekttupel $(ob_1,...,ob_{K_u})$ gültig ist. Diese Unbekanntheit der Prädikatsgültigkeit läßt aber im Sinne einer dreiwertigen Logik beide Alternativen zu: sowohl den Wahrheitswert "ungültig" als auch den Wahrheitswert "unbestimmt" für das Prädikatsvorkommnis $prä_u(ob_1,...,ob_{K_u})$.

85) Daher könnte zu vollständig definierten Wahrheitsfunktionen mit einer modifizierten BOOL'schen Menge $BOOL_{2*}$ = {gültig,unbekannt} zurückgekehrt werden. Sie wären mit den oben eingeführten partiell definierten Wahrheitsfunktionen gleichwertig. Welche Darstellungsalternative gewählt wird, ist letztlich Ausdruck einer willkürlichen Formalisierungsprämisse. Der Verf. bevorzugt die Verwendung partiell definierter Wahrheitsfunktionen, weil sie die unterschiedliche epistemische Qualität zwischen dem Wahrheitswert "gültig" und der Unbekanntheit eines Wahrheitswertes verdeutlicht: Beide Aspekte werden in den partiell definierten Wahrheitsfunktionen verschieden behandelt, indem nur der Wahrheitswert "gültig" in ihren Bildern zugelassen wird. In vollständig definierten Wahrheitsfunktionen würden dagegen beide Aspekte als Wahrheitswerte "gültig" und "unbekannt" formal gleichwertig behandelt. Dies widerspräche aber der qualitativen Unterschiedlichkeit der beiden Aspekte: Der Wahrheitswert "gültig" drückt wohlbestimmtes metasprachliches Wissen aus. Die Unbekanntheit eines Wahrheitswerts repräsentiert dagegen metasprachliches Nichtwissen darüber, ob einer der zwei wohlbestimmten Wahrheitswerte "gültig" oder "ungültig" vorliegt.

86) Falls zunächst die Ungültigkeit eines Prädikatsvorkommnisses betrachtet worden sein sollte, wird statt dessen ein Prädikatsvorkommnis mit kontradiktorischer Bedeutung gebildet. Dessen Gültigkeit ist mit der Ungültigkeit des ursprünglichen Prädikatsvorkommnisses äquivalent. Wenn für ein Prädikatsvorkommnis sowohl dessen Gültigkeit als auch dessen Ungültigkeit Interesse erlangen kann, so wird dieses Prädikatsvorkommnis durch zwei komplementäre Vorkommnisse ersetzt: Das erste stimmt mit dem ursprünglichen Prädikatsvorkommnis überein und wird nur hinsichtlich seiner Gültigkeit betrachtet. Das zweite Prädikatsvorkommnis wird wiederum als kontradiktorisches Gegenteil des ursprünglichen Prädikatsvorkommnisses angesetzt, so daß die ursprünglich interessierende Ungültigkeit abermals in die relevante Gültigkeit des neu gebildeten Prädikatsvorkommnisses transformiert wird. Mit Hilfe dieser Umformulierungen brauchen nur noch gültige Prädikatsvorkommnisse berücksichtigt zu werden. Diese Vorgehensweise wird an einem Beispiel verdeutlicht: Für das Prädikatsvorkommnis "Maschine MA ist betriebsbereit" kann bei der Modellierung von Maschinenbelegungen sowohl dessen Gültigkeit als auch dessen Ungültigkeit Bedeutung erlangen. Daher wird es durch die zwei komplementären Prädikatsvorkommnisse "Maschine MA ist betriebsbereit" und "Maschine MA ist ausgefallen" ersetzt. Da beide Prädikatsvorkommnisse kontradiktorische Bedeutungen besitzen, gilt: Anstatt bei einem Maschinenausfall (einer Maschineninstandsetzung) die Ungültigkeit des ersten (zweiten) Prädikatsvorkommnisses zu betrachten, wird die Gültigkeit des (ersten) zweiten Prädikatsvorkommnisses berücksichtigt. Folglich ist es unnötig, ungültige Prädikatsvorkommnisse zu benutzen, wenn auf die oben definierten Prädikatsformulierungen zurückgegriffen wird. Vgl. zu einer analogen Konstruktion von Prädikaten mit kontradiktorischer Bedeutung BOSSI (1989), S. 96 (für dominierte und nicht-dominierte Vektoren).

87) Daher bedeutet die Beschränkung auf den nur einen wohlbestimmten Wahrheitswert "gültig" in Verbindung mit der Gültigkeitsprämisse also auch keinen Verlust an prädikatenlogischer Ausdrucksmächtigkeit. Lediglich der Aufwand für die Umformulierung von Prädikatsvorkommnissen und die Modellaufblähung durch zusätzliche komplementäre Prädikatsvorkommnisse, die beide in einer früheren Anmerkung skizziert wurden, stellen echte konzeptionelle Kosten des hier vorgestellten, epistemisch bereinigten Modellierungsansatzes dar.

88) Damit scheiden negative Fakten für die Repräsentation ungültiger Prädikatsvorkommnisse aus.

89) Vgl. dazu die Differenzierung zwischen kinetischem, dynamischem und epistemischen Wissen.

4.2.3.2 Überlagerung einer Sortenstruktur

4.2.3.2.1 Überblick

Es besteht eine weitreichende - allerdings auch nicht vollständige Korrespondenz zwischen Signaturkonzept und Prädikatenlogik. Sie wird in den nachfolgenden zwei Kapiteln näher erläutert. Diese Korrespondenz bildet eine bemerkenswerte abstrakte Schnittstelle zwischen zwei - zunächst voneinander unabhängig definierten - formalen Konzepten. Sie bietet Gelegenheit, Prädikatenlogik und Signaturkonzept durch wechselseitige Zuordnung ihrer typischen Konstrukte miteinander zu kombinieren. Es liegt nahe, daß diese Konzeptintegration zu einer gegenseitigen Befruchtung beider Teilkonzepte führt und insgesamt ein besonders reichhaltiges formales Konzept hervorbringt.

Der wesentliche Befruchtungseffekt geht von dem Umstand aus, daß Prädikatenlogik und Signaturkonzept unterschiedliche Schwerpunkte bei der syntaktischen Konstruktion formalsprachlicher Ausdrücke und ihrer semantischen Interpretation setzen. Aus der Sicht der Prädikatenlogik dominieren die syntaktische Formulierung von beliebig komplex strukturierten Formeln und die semantische Untersuchung, welche dieser Formeln unter einer Interpretation gültig sind. Daher stehen Prädikate, deren Extensionen und die entsprechenden Faktenmengen im Vordergrund des Interesses. Im Rahmen des Signaturkonzepts ragt dagegen kein bestimmter Ausdruckstyp hervor. Formeln können zwar als algebraische Formeln von SIG-Spezifikationen gebildet werden. Doch stellen sie nur eine formalsprachliche Ausdruckskategorie neben vielen anderen dar. Statt dessen liegt der Schwerpunkt des Signaturkonzepts auf der komplexen Ausdifferenzierung des algebraischen Universums durch eine überlagerte Sortenstruktur.

Die unterschiedlichen Foki von Prädikatenlogik und Signaturkonzept werden in einer sortierten Prädikatenlogik miteinander kombiniert. Das Integrationsresultat zeichnet sich durch hohe potentielle Formelkomplexität, Berücksichtigung von Formelgültigkeiten sowie sortenbezogene Strukturierung der Formelargumente aus. Hierdurch wird einerseits die Prädikatenlogik um strukturell reichhaltigere Formulierungsmöglichkeiten für die Formelargumente erweitert. Dadurch werden übersichtlichere und zugleich kompaktere Formulierungen von prädikatenlogischen Objektmodellen ermöglicht. Ein Beispiel weiter unten wird dies verdeutlichen.

Darüber hinaus lassen sich im Rahmen einer sortierten Prädikatenlogik Inferenzen im allgemeinen effizienter ausführen als in konventionellen prädikatenlogischen Beweissystemen[1]. Denn die sortengerechte Variablenbindung erlaubt es, den logischen Suchraum grundsätzlich möglicher Variablensubstitutionen drastisch zu reduzieren. Anstatt Variablen durch Konstanten aus dem unsortierten prädikatenlogischen Universum ersetzen zu können, kommen in einer sortierten Prädikatenlogik nur noch Konstanten aus den jeweils sortenspezifischen Teilbereichen für die Variablenbindung in Betracht.

Dies wirkt sich insbesondere auch auf das Inferenzkonzept der logischen Programmiersprache PROLOG aus, die in dieser Arbeit für die Implementierung von Netzmodellen herangezogen wird. Denn bei jedem Unifizierungsschritt brauchen nur die voranstehend erläuterten sortengerechten Variablenbindungen untersucht zu werden. Auf diese Weise steuert das Signaturkonzept durch seine Sortenstruktur, die den Argumenten prädikatenlogischer Formeln überlagert wird, einen bemerkenswerten Beitrag zur Erhöhung der prädikatenlogischen Inferenzeffizienz bei. Der hier ausgewählte Sprachvariante Turbo-PROLOG gehört zu den wenigen PROLOG-Dialekten, die eine solche Sortierung ihrer Formeln zulassen. Diese Eigenschaft stellte einen wesentlichen Grund dar, sie für die Implementierung Synthetischer Netze auszuwählen.

Bisher wurden nur die Befruchtungen der Prädikatenlogik durch das Signaturkonzept herausgestellt. Die konzeptionelle Bereicherung gilt aber auch in der umgekehrten Richtung. So ist es mit Hilfe des prädikatenlogischen Unifizierungs- und Resolutionskonzepts möglich, das Existenzproblem von Auswertungsfunktionen für Signaturen konstruktiv zu bewältigen. Dies wurde schon früher herausgestellt. Darüber hinaus bietet sich die prädikatenlogische Programmiersprache PROLOG an, um die Lösungen für das vorgenannte Existenzproblem automatisch zu ermitteln. Diesem zweifachen Transfer aus dem Bereich der Prädikatenlogik in die Domäne des Signaturkonzepts kommt wesentliche Bedeutung zu. Denn für die Bestimmung geeigneter Auswertungsfunktionen wurde innerhalb des Signaturkonzepts bisher kein eigenständiges Lösungskonzept entwickelt. Des weiteren läßt sich das Signaturkonzept, das zunächst auf algebraische Konstrukte fixiert ist, so erweitern, daß es eine komfortable und zugleich kompakte Repräsentation prädikatenlogischer Formeln erlaubt. Dies führt später zu einem prädikatenlogisch erweiterten Signaturkonzept.

Anmerkungen zum Kapitel:

1) Darauf wurde schon eingangs anläßlich der Rechtfertigung des prädikatenlogischen Ansatzes hingewiesen.

4.2.3.2.2 Sortierte Prädikatenlogik

Die konventionelle Prädikatenlogik wird zu einer strukturell reichhaltigeren sortierten Prädikatenlogik[1)] erweitert, indem sie mit dem Signaturkonzept kombiniert wird[2)]. Dadurch wird den konventionellen prädikatenlogischen Formeln eine Sortenstruktur überlagert[3)]. Allerdings erfolgt auf diese Weise keine grundsätzliche Ausweitung des prädikatenlogischen Ausdruckspotentials[4)]. Statt dessen wird durch die Strukturanreicherung "nur" der Formulierungskomfort vergrößert[5)]. Der Formulierungskomfort erstreckt sich insbesondere darauf, das algebraische Ausdruckspotential in die Prädikatenlogik auf übersichtliche und "natürliche" Weise einzubinden[6)]. Dies betrifft vor allem die algebraisch definierten Operationen und Restriktionen. Hinzu kommen weitere Vorzüge von sortierten Ansätzen, die sich vor allem auf technische Aspekte des Umgangs mit Formelsystemen beziehen[7)]. Sie spielen im folgenden keine nennenswerte Rolle.

Die Sortenstruktur einer sortierten Prädikatenlogik beruht auf der Sortenmenge $SO=\{sort_i, ..., sort_I\}$ einer zugrundeliegenden Signatur $SIG=(SO,OP)$ mit $I \in \mathcal{N}_+$ und $I \geq 2$. Diese Sortenmenge wird benutzt, um in einer prädikatenlogischen Semantik den vormals unstrukturierten Definitionsbereich DB für Variablen und Konstantensymbole in sortenspezifische Teilbereiche OB_i mit $i \in \{1,...,I\}$ auszudifferenzieren und gleichzeitig hierarchisch zu strukturieren. Das Ergebnis der Bereichsdifferenzierung und -strukturierung wird fortan als prädikatenlogische Objektmenge OB notiert. Sie ist das sortierte prädikatenlogische Universum. Es umfaßt alle formalen Objekte, aus denen die Argumente von interpretierten konstanten Formeln $prä_u(ob_1,...,ob_{Ku})$ aufgebaut werden können[8)].

Die sortenspezifischen Teilbereiche OB_i stellen jeweils Mengen formaler Objekte "ob" dar. Diese Objekte brauchen keineswegs mehr - wie noch die Elemente aus dem konventionell definierten Definitionsbereich DB - atomare formale Objekte (Individuen) zu sein. Vielmehr kann es sich nun auch um beliebig komplex zusammengesetzte formale Objekte handeln. Dies folgt aus der Möglichkeit, Sorten als Zielsorten von Funktionssymbolen derivativ zu definieren[9)]. Die Objektmenge, die einer solchen Zielsorte zugeordnet ist, umfaßt dann per definitionem zusammengesetzte Objekte. Die sortenspezifischen Objektmengen müssen im Prinzip nicht disjunkt definiert sein[10)], sondern können gemeinsame Objekte besitzen. Die Vereinigungsmenge $OB = \cup(i \in \{1,...,I\})\colon OB_i$ aller sortenspezifischen Objektmengen OB_i ist das prädikatenlogische Universum. Es umfaßt alle formalen Objekte, die für ein prädikatenlogisches Formelsystem mit der Prädikatssymbolemenge PRÄ definiert sind. Die Sortenstruktur, die diesem prädikatenlogischen Universum OB durch die Sortierung ihrer Objektmengen überlagert ist, stellt die Mengenfamilie $OBF=(OB_i\colon i=1,...,I)$ dar[11)].

Mit Hilfe der sortenspezifischen Mengen formaler Objekte läßt sich die Menge KAF_u aller konstanten atomaren Formeln definieren, die sich aus einem Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ formieren lassen. Dies bereitet im Gegensatz zur konventionellen Prädikatenlogik nunmehr keine besonderen Schwierigkeiten, weil die formalen Objekte "ob" sowohl atomar als auch beliebig komplex zusammengesetzt sein dürfen. Jeder Stelle st_k des sortierten Arguments $(st_1,...,st_{Ku})$ eines Prädikatssymbols wird eine Menge $OB_{i(k)}$ formaler Objekte aus dem prädikatenlogischen Universum OB mit $k \in \{1,..,K_u\}$ sortengerecht zugeordnet. Dann gilt für die Menge KAF_u aller konstanten atomaren Formeln $prä_u(ob_1,...,ob_{Ku})$ eines Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$:

$$KAF_u = \{prä_u(ob_1,...,ob_{Ku})\colon ob_1 \in OB_{i(1)} \wedge ... \wedge ob_{Ku} \in OB_{i(Ku)}\}$$

Die Menge KAF_u wird auch kurz als konstante Prädikatemenge für das Prädikatssymbol $Prä_u$ bezeichnet.

Entsprechend läßt sich die Menge TAF_u aller atomaren Formeln $prä_u(te_1,...,te_{Ku})$ einführen, die für ein Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ mit Argumenten aus teilevaluierten Termen te_k für $k \in \{1,..,K_u\}$ gebildet werden können[12]. Mit $TTM_{i(k)}(VA_{i(k)})$ als Menge aller teilevaluierten Terme der Sorte $sort_i$ gilt:

$$TAF_u = \{prä_u(te_1,...,te_{Ku}): te_1 \in TTM_{i(1)}(VA_{i(1)}) \wedge ... \wedge te_{Ku} \in TTM_{i(Ku)}(VA_{i(Ku)})\}$$

Die Menge TAF_u heißt auch teilevaluierte Prädikatemenge für das Prädikatssymbol $Prä_u$. Ihre Elemente $prä_u(te_1,...,te_{Ku})$ werden kurz als teilevaluierte atomare Formeln angesprochen.

Die Faktenmenge $FAK_{u.r}$ eines Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ unter einer Interpretation I_r kann mit Hilfe einer beliebigen Teilmenge $TKAF_u$[13] der konstanten Prädikatemenge KAF_u dargestellt werden[14]:

$$FAK_{u.r} = \{fakt_r(prä_u(ob_1,...,ob_{Ku})): \exists(TKAF_u \in pot(KAF_u)): prä_u(ob_1,...,ob_{Ku}) \in TKAF_u\}$$

Ein potentielles Fakt ist jeder formalsprachliche Ausdruck $fakt(prä_u(ob_1,...,ob_{Ku}))$, der unter jeder denkmöglichen Interpretationen I_r tatsächlich ein Fakt $fakt_r(prä_u(ob_1,...,ob_{Ku}))$ darstellen könnte. Potentielles und tatsächliches Fakt unterscheiden sich formal also nur durch das Fehlen bzw. Vorhandensein des interpretationsbezogenen Indexes "r". Für die potentielle Faktenmenge $PFAK_u$ eines Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ gilt:

$$PFAK_u = \{fakt(prä_u(ob_1,...,ob_{Ku})): prä_u(ob_1,...,ob_{Ku}) \in KAF_u\}$$

Die potentielle Faktenmenge $PFAK_u$ fällt mit der Menge KAF_u aller konstanten atomaren Vorkommnisse des Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ bis auf die Faktenkennzeichnung "fakt(...)" zusammen. Daher läßt sich die potentielle Faktenmenge auch mit Hilfe eines neu eingeführten Faktoperators "fak" aus der Menge KAF_u ableiten. Der Operator wird so definiert, daß er bei seiner Applikation auf eine beliebige Menge ME für jedes Element x_h aus dieser Menge den Ausdruck "$fak(x_h)$" liefert gemäß:

$$fak: \quad ME = \{x_h: h=1,...,H\} \rightarrow fak(ME) = \{fak(x_h): h=1,,,.H\}$$

Dann gilt für die potentielle Faktenmenge vereinfacht:

$$PFAK_u = fak(KAF_u)$$

Wenn für ein prädikatenlogisches Formelsystem insgesamt U verschiedene Prädikatssymbole $Prä_u$ mit $u \in \{1,...,U\}$ definiert sind, dann ist die Vereinigungsmenge $PFAK = \cup (u=1,...,U): PFAK_u$[15] die potentielle Faktenmenge des gesamten Formelsystems[16].

Darüber hinaus läßt sich der Faktoperator zu einem indizierten Faktoperator "fak_r" erweitern, der jedes Element x_h aus einer Menge ME auf das indizierte faktische Element "$fak_r(x_h)$" mit beliebigem Index "r" abbildet. Daher kann jede Faktenmenge nunmehr ebenso notiert werden als:

$$FAK_{u.r} = fak_r(TKAF_u) \quad \text{mit:} \quad TKAF_u \in pot(KAF_u)$$

Die kompakte Darstellungsweise von potentiellen und tatsächlichen Faktenmengen mit Hilfe des Faktoperators und seiner indizierten Erweiterung wird sich auch später als vorteilhaft erweisen[17].

Die interne Strukturierung des Universums prädikatenlogischer Objekte, über denen in einer prädikatenlogischen Semantik Variablen und Konstantensymbole definiert sind, wirkt sich auch auf die Notationen prädikatenlogischer Konstrukte im zugrundeliegenden syntaktischen Kalkül aus. Für eine Variable X, die zur Sorte $sort_i$ gehört, nehmen die Argumente ihrer Quantoren in einem sortierten und interpretierten prädikatenlogischen Kalkül die Gestalt "$(X \in OB_i)$:" an[18]. Die Deklarationen der Funktions- und Prädikatssymbole werden nicht mehr als Fun_j:$st_1 ... st_{Kj} \rightarrow st_{Kj+1}$ bzw. $Prä_u(st_1,...,st_{Ku})$ notiert. Statt dessen werden ihre Stellen st_k durch die jeweils zugehörigen Sorten $sort_{i(j.k)}$ bzw. $sort_{i(u.k)}$ spezifiziert[19]:

$$Fun_j\text{: } sort_{1(j.1)} ... sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)}$$
$$Prä_u(sort_{1(u.1)},...,sort_{i(u.Ku)})$$

Unter Rückgriff auf das früher erörterte Konzept der Wahrheitsfunktionen[20] kann die Notation eines sortierten Prädikatssymbols $Prä_u$ auch so zu einem sortierten Wahrheitsfunktionssymbol ergänzt werden, daß das Prädikatssymbol in seiner speziellen Darstellungsweise durch das Wahrheitsfunktionssymbol von der üblichen Deklaration sortierter Funktionssymbole nicht mehr abweicht. Hierzu dient die Sorte "wahr_wert", die speziell für die Wahrheitsfunktionssymbole von Prädikatssymbolen $Prä_u$ definiert ist. Ihr zugehöriger Definitionsbereich $OB_{wahr\text{-}wert}$ ist mit der bereits eingeführten BOOL'schen Objektmenge $BOOL_2 = OB_{wahr\text{-}wert} = \{gültig, ungültig\}$ zulässiger Wahrheitswerte identisch. Allerdings wird hier bereits auf die partiell definierten Wahrheitsfunktionen pwf_u mit ihrer reduzierten BOOL'schen Objektmenge $BOOL_1 = OB_{wahr\text{-}wert} = \{gültig\}$ zurückgegriffen, die im vorigen Kapitel eingeführt wurde. Unter diesen Voraussetzungen gilt für Funktions-, Prädikats- und Wahrheitsfunktionssymbole in einer sortierten Prädikatenlogik:

$$Fun_j\text{: } sort_{1(j.1)} ... sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)}$$
$$Prä_u(sort_{1(u.1)},...,sort_{i(u.Ku)})$$
$$Pwf_u\text{: } sort_{1(u.1)} ... sort_{i(u.Ku)} \rightarrow wahr\_wert$$

Diese zweite Notationsweise verdeutlicht, daß Funktions- und Wahrheitsfunktionssymbole in einer sortierten Prädikatenlogik aus der Perspektive des Signaturkonzepts strukturell übereinstimmende Varianten von Operationssymbolen darstellen. Die Argumente von Funktionssymbolen und von prädikatszugehörigen Wahrheitsfunktionssymbolen sind jeweils beliebige K_j- bzw. K_u-stellige Worte aus einer positiven Sortensprache SOS_{SIG} der jeweils zugrundeliegenden Signatur SIG. Allerdings besteht ein Unterschied hinsichtlich der Zielsorte: Die Zielsorte "wahr_wert" ist für Wahrheitsfunktionssymbole reserviert; sie können aber auch keine andere Zielsorte besitzen. Funktionssymbole dürfen dagegen beliebige Zielsorten aufweisen; ausgenommen jedoch die prädikatspezifische Zielsorte "wahr_wert". In dieser unterschiedlichen Zielsortendefinition manifestiert sich die kategoriale Verschiedenheit von Funktions- und Prädikatssymbolen, die schon früher betont wurde. Die Zielsorte "wahr_wert" ist aber nicht unmittelbar für Prädikatssymbole, sondern nur für ihre Wahrheitsfunktionssymbole definiert. Dies verdeutlicht, daß die Prädikatssymbole nicht unmittelbar als Operationssymbole im Signaturkonzept verändert sind, sondern nur über den Umweg ihrer Wahrheitsfunktionssymbole. Hierauf wird später zurückgekommen.

Die Art und die Anordnung der Sorten, die das Argument eines Funktions- oder Prädikatssymbols konstituieren, werden als dessen Sortenstruktur bezeichnet[21]. Entsprechend wird von sortierten Funktions- und Prädikatssymbolen gesprochen[22]. Die Sortenstruktur von Funktions- und Prädikatssymbolen kann mehrdeutig sein. Hierbei wird auf die multipel definierten Sorten der zugrundeliegenden Signatur SIG zurückgegriffen. Liegt eine solche multiple Sortendefinition für mindestens eine Sorte $sort_{i(j.k)}$ oder $sort_{i(u.k)}$ aus dem Funktions- bzw. Prädikatsargument vor, so kann die entsprechende k-te Stelle in der zugehörigen Funktion oder prädikatenlogischen Formel auf mindestens zwei alternative Weisen aus Konstanten, Variablen und Funktoren zusammengesetzt werden. Dies setzt jedoch eine Interpretation der Funktions- bzw. Prädikatssymbole voraus.

Bei ihrer Interpretation werden die syntaktisch definierten sortierten Funktions- und Prädikatssymbole Fun_j bzw. $Prä_u$ sowie die sortierten Wahrheitsfunktionssymbole Pwf_u mit der Sortenstruktur $(OB_1,...,OB_I)$ aus einer prädikatenlogischen Semantik kombiniert. Dadurch werden den Funktions- und Prädikatssymbolen entsprechend sortierte, nunmehr semantisch definierte Funktionen fun_j bzw. Prädikate $prä_u$[23] zugeordnet. Analog werden die Wahrheitsfunktionssymbole Pwf_u durch sortierte Wahrheitsfunktionen pwf_u interpretiert. Dabei erstrecken sich die Funktionen fun_j, die Prädikate $prä_u$ und die partiellen Wahrheitsfunktionen pwf_u als semantische Konstrukte auf teilevaluierte Terme te_k[24]. Diese Terme stammen jeweils aus den sortenspezifischen Teilbereichen OB_i des prädikatenlogischen Universums OB mit $i \in \{1,...,I\}$. Folglich gilt:

a) in bezug auf Funktionssymbole Fun_j und Funktionen fun_j:

Fun_j: $sort_{1(j.1)} \dots sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)}$

fun_j: $OB_{1(j.1)} \times \dots \times OB_{i(j.Kj)} \rightarrow OB_{i(j.Kj+1)}$

$(te_1,...,te_{Kj}) \rightarrow fun_j(te_1,...,te_{Kj}) = te_{Kj+1}$

b) in bezug auf Prädikatssymbole $Prä_u$ und Prädikate $prä_u$ einschließlich der korrespondieren Wahrheitsfunktionssymbole Pwf_u bzw. Wahrheitsfunktionen pwf_u:

$Prä_u(sort_{1(u.1)} \dots sort_{i(u.Kj)})$

Pwf_u: $sort_{1(u.1)} \dots sort_{i(u.Kj)} \rightarrow wahr\_wert$

$prä_u(te_1,...,te_{Kj})$

pwf_u: $OB_{1(u.1)} \times \dots \times OB_{i(u.Kj)} \rightarrow OB_{wahr\_wert} = BOOL_1$

$(te_1,...,te_{Kj}) \rightarrow pwf_u(te_1,...,te_{Kj}) = te_{Kj+1}$

Die voranstehenden Zusammenfassungen syntaktischer und semantischer Definitionen von Funktions- und Prädikatssymbolen bzw. Funktionen und Prädikaten werden fortan vereinfacht als sortierte Funktions- und Prädikatsdefinitionen angesprochen. Falls die Ausprägung eines Terms te_k im Argument einer sortierten Funktion oder eines sortierten Prädikats nicht näher spezifiziert werden soll, kann die anonyme Variable "_" der Programmiersprache PROLOG als Irrelevanzanzeiger benutzt werden.

Sortierte Funktionsdefinitionen lösen zwei formale Probleme auf, die bei unsortierten Funktionsdefinitionen vorherrschen. Erstens läßt sich bei einer unsortierten Funktion fun_j: $ME_1 \times \dots \times ME_{Kj} \rightarrow ME_{Kj+1}$ die Bezugnahme auf unterschiedliche Mengen ME_k mit $k \in \{1,...,K_j\}$ in der Funktionsdeklaration mit der konventionellen Prädikatenlogik nicht vereinbaren. Denn es gibt dort nur ein undifferenziertes prädikatenlogisches Universum DB. In einer sortierten Prädikatenlogik können die Mengen M_k jedoch ohne Schwierigkeiten als Mengen $OB_{i(k)}$ formaler Objekte aus der Sortenstruktur $(OB_1,...,OB_I)$ des Universums OB gewonnen werden. Zweitens bleibt in konventionellen Funktionsdefinitionen fun_j: $(x_1,...,x_{Kj}) \rightarrow x_{Kj+1}$ der formale Charakter der Komponenten x_k mit $k \in \{1,...,K_{j+1}\}$ aus dem Argument und dem Bild der Funktion im Unklaren. Zumeist werden sie als Variablen behandelt, oftmals aber auch als Konstanten angesehen. Es irritiert aber, mit demselben Symbol x_k unterschiedliche formale Kategorien zu bezeichnen. Darüber hinaus bleibt zweifelhaft, ob solche konventionellen Funktionsdefinitionen tief strukturierte Funktionsargumente zulassen. Zumindest verbietet der Variablen- und Konstantenbegriff strenggenommen eine tiefe Strukturierung. Diese Schwierigkeiten werden von vornherein vermieden, wenn die Komponenten x_k in einer sortierten Prädikatenlogik als teilevaluierte Terme $te_{i(k)}$ festgelegt werden. Denn diese Terme sind formal präzise als Variablen, Konstanten oder als beliebig komplexe Zusammensetzungen aus Variablen, Konstanten und Funktoren definiert. Daher werden in dieser Arbeit die o.a. sortierten Funktionsdefinitionen benutzt[25)].

Mit Hilfe sortierter Funktionsdefinitionen lassen sich auch beliebig komplexe formale Objekte "ob" definieren, denen eine Sortenstruktur überlagert ist. Grundlage ist ein sortiertes K-stelliges Funktionssymbol "Obj" mit $K \in \mathcal{N}_+$ und:

$$\text{Obj:} \quad sort_{1(1)} \ldots sort_{i(Kj)} \rightarrow sort_{i(K+1)}$$

Ein sortiertes Objekt "ob" wird dann als Bild der zugehörigen sortierten Funktion "obj" für ein konstantes Argument $(ob_1,\ldots,ob_K)$ eingeführt:

$$\begin{aligned} \text{obj:} \quad & OB_{1(1)} \times \ldots \times OB_{i(K)} \rightarrow OB_{i(K+1)} \\ & (ob_1,\ldots,ob_K) \rightarrow obj(ob_1,\ldots,ob_K) = ob_{K+1} = ob \end{aligned}$$

Die Überlagerung einer Sortenstruktur über den Kalkül der konventionellen Prädikatenlogik 1. Stufe führt zum Kalkül der sortierten Prädikatenlogik 1. Stufe. Für ihn resultiert durch Zusammenfassen der früheren konventionellen Kalküldefinition mit den oben eingeführten Konstrukten:

Definition:
Kalkül der sortierten Prädikatenlogik 1. Stufe

zulässige Zeichen:

- logische Zeichen
  - Konnektoren
    - Konjunktor: $\wedge$
    - Adjunktor : $\vee$
    - Disjunktor: $\underline{\vee}$
    - Subjunktor: $\rightarrow$
    - Bijunktor : $\leftarrow$
  - logische Operatoren
    - Negator: $\neg$
    - Quantoren
      - Allquantor: $\forall(X)$:
      - Existenzquantor: $\exists(X)$:
      - Einsquantor: $\underline{\exists}(X)$:

Fortsetzung auf der nächsten Seite ...

... Fortsetzung von der voranstehenden Seite:

- deskriptive Zeichen
  - Sorten: $sort_i$ mit $i \in \{1,...,I\}$ und $I \in \mathcal{N}_+$
  - Individuen:
    - Variablen: $X \in VA_i$ für $i \in \{1,...,I\}$
    - Konstantensymbole: Ko: $\rightarrow sort_i$ für $i \in \{1,...,I\}$
  - Funktionssymbole: Fun: $sort_{i(1)} \ldots sort_{i(k)} \rightarrow sort_{K+1}$ mit $K \in \mathcal{N}_+$
  - Terme:
    - Jede Variable $X \in VA_i$ ist ein Term te der Sorte $sort_i$ aus der sortenspezifischen Termmenge $TTM_i(VA_i)$.
    - Wenn Ko: $\rightarrow sort_i$ ein Konstantensymbol ist, dann ist te mit te=ko() ein Term der Sorte $sort_i$ aus der sortenspezifischen Termmenge $TTM_i(VA_i)$.
    - Wenn Fun: $sort_{i(1)} \ldots sort_{i(K)} \rightarrow sort_{K+1}$ ein Funktionssymbol ist und wenn $te_1,...,te_K$ Terme aus den sortenspezifischen Termmengen $TTM_{i(1)}(VA_{i(1)}),...,TTM_{i(K)}(VA_{i(K)})$ sind, dann ist te_{K+1} mit $te_{K+1}=fun(te_1,...,te_K)$ ein Term der Sorte $sort_{i(K+1)}$ aus der sortenspezifischen Termmenge $TTM_{i(K+1)}(VA_{i(K+1)})$.
    - Andere Terme sind unzulässig.
  - Prädikatssymbole:
    - $Prä(sort_{i(1)},...,sort_{i(k)})$ mit $K \in \mathcal{N}_+$
    - Prä()
- Hilfszeichen: "(" und ")" sowie ","

Formierungsregeln für zulässige Formeln:

- atomare Formeln
  - FR_P: Wenn Prä($sort_{i(1)}$,...,$sort_{i(K)}$) ein Prädikatssymbol ist und wenn te_1,...,te_K Terme darstellen, dann ist prä(te_1,...,te_K) eine Formel.
  - FR_A: Wenn Prä() ein Prädikatssymbol ist, dann ist prä() eine Formel.
- zusammengesetzte Formeln
  - FR_K: Wenn p_1 und p_2 Formeln sind und wenn & einen Konnektor darstellt, dann ist auch $p_1 \& p_2$ eine Formel.
  - FR_O: Wenn p eine Formel darstellt und wenn $\mathcal{O}$ ein logischer Operator ist, dann ist auch $\mathcal{O}$p eine Formel.
- Alle anderen Formeln sind unzulässig.

In der voranstehend skizzierten Konzeption einer sortierten Prädikatenlogik wird eine bemerkenswerte strukturelle Übereinstimmung zwischen Signaturkonzept und Prädikatenlogik ausgenutzt. Die Parallelen zwischen diesen beiden formalen Konzepten erstrecken sich auf mehrere Ebenen: Eine Signatur SIG wird zunächst durch eine SIG-Algebra interpretiert. Dabei werden den Sorten aus der Signatur Objektmengen OB_i zugeordnet, deren Vereinigung das algebraische Universum OB ist. Dies entspricht[26] der konventionellen Interpretation eines prädikatenlogischen Kalküls, bei der alle Variablen des Kalküls auf ihren gemeinsamen Definitionsbereich DB abgebildet werden. Darüber hinaus wird im Signaturkonzept eine SIG-Termmenge durch eine SIG-Termauswertung u.a.[27] dadurch interpretiert, daß Operations- und Konstantensymbole durch Operationen bzw. Konstanten ersetzt werden. Hiermit korrespondiert in einer prädikatenlogischen Kalkülinterpretation die Abbildung der Funktionssymbole auf Funktionen, der Konstantensymbole auf Konstanten sowie der Prädikatssymbole auf deren Extensionen. Aufgrund dieser Parallelen entspricht die semantische Dimension der Prädikatenlogik einerseits der Kombination aus einer SIG-Algebra und einer SIG-Termauswertung andererseits. Erste interpretiert Formeln aus dem prädikatenlogischen Kalkül - letze interpretiert eine Signatur SIG, die um eine SIG-Termmenge erweitert ist.

Infolge der skizzierten strukturellen Übereinstimmung zwischen Signaturkonzept und Prädikatenlogik konnten die Sorten einer Signatur SIG herangezogen werden, um dem ehemals homogenen prädikatenlogischen Definitionsbereich DB eine Sortenstruktur $OBF = (OB_i\text{: } 1 = 1,\ldots,I)$ zu überlagern. Daraus resultierte das nunmehr sortierte prädikatenlogische Universum OB, das mit dem algebraischen Universum OB der zugrundegelegten Signatur SIG zusammenfällt.

Darüber hinaus lassen sich auch die Operationssymbole einer Signatur in korrespondierende prädikatenlogische Konstrukte übersetzen. Denn 0-stellige Operationssymbole stellen prädikatenlogische Konstantensymbole dar. K_j-stelligen Operationssymbolen mit $K_j \in \mathcal{N}_+$ entsprechen in einem prädikatenlogischen Kalkül entweder Funktions- oder aber Wahrheitsfunktionssymbole.

Schließlich korrespondieren die sortenspezifischen Objektmengen OB_i aus dem Universum OB einer SIG-Algebra mit den sortenspezifischen Teilbereichen OB_i aus dem extensionsgleichen Universum OB der prädikatenlogischen Interpretation.

Allerdings bestehen auch charakteristische Unterschiede zwischen Signaturkonzept und Prädikatenlogik. Eine erste Diskrepanz betrifft die Behandlung von Variablen. Im Signaturkonzept bedeutet eine SIG-Termauswertung die vollständige Abbildung aller Termkomponenten - der Variablen, Konstanten- und Operationssymbole - auf Konstanten und Operationen. Die prädikatenlogische Semantik stimmt hiermit nur im Hinblick auf Konstanten- und Operationssymbole (Funktionssymbole) überein. Dagegen werden die Variablen in Prädikatssymbolen und in den Termen der Funktionssymbole nicht durch Konstanten ersetzt. Denn in prädikatenlogischen Interpretationen wird den Variablen nur ein Definitionsbereich, nicht aber eine Konstante aus diesem Bereich zugeordnet. Daher bestehen die Argumente von interpretierten prädikatenlogischen Formeln stets aus teilevaluierten Termen[28)], die immer noch Variablen enthalten können[29)].

Falls die prädikatenlogischen Formeln noch Variablen umfassen, unterscheiden sie sich deutlich von den algebraischen Ausdrücken, die in einer SIG-Algebra nur aus variablenfreien Kombinationen von Konstanten und Operatoren bestehen. Daher korrespondieren die prädikatenlogischen Interpretationen nicht vollständig, sondern nur partiell mit den algebraischen SIG-Termauswertungen. Erst wenn neben den "reinen" Auswertungsfunktionen $eval_i$ auch die Teilfunktionen $teval_i$ für teilevaluierte Terme zugelassen werden, läßt sich eine vollständige Entsprechung zwischen dem - derart erweiterten - Signaturkonzept und der Prädikatenlogik herstellen.

Ein zweiter wesentlicher Unterschied zwischen prädikatenlogischen Interpretationen einerseits sowie SIG-Algebren und -Termauswertungen andererseits erstreckt sich auf die Objektmengen, die der Interpretation von Variablen dienen[30)]. In der konventionellen Prädikatenlogik handelt es sich beim Definitionsbereich DB immer um eine unstrukturierte Menge prinzipiell gleichartiger und atomarer formaler Objekte (Konstanten). Im Signaturkonzept wird dagegen dem algebraischen Universum OB eine Sortenstruktur aufgeprägt: Entsprechend zur Menge der Sorten $sort_i$ aus der zugrundeliegenden Signatur SIG ist das Universum formaler Objekte in die Objektmengen OB_i zerlegt. Auf diese Weise wird die Menge aller formalen Objekte, die als Konstanten zur Interpretation von Variablen herangezogen werden können, intern strukturiert. Diese formalen Objekte können auch beliebig komplex zusammengesetzt sein. Eine analoge Segmentierung ihres Definitionsbereichs DB kennt die konventionelle Prädikatenlogik grundsätzlich nicht. Erst wenn sie durch eine Sortenstruktur in der oben erläuterten Weise überlagert wird, erhält die - nunmehr sortierte - Prädikatenlogik denselben Strukturreichtum für die formalen Objekte, die zur Interpretation von Variablen dienen.

Drittens unterscheiden sich Prädikatenlogik und Signaturkonzept auch im Hinblick auf Art und Umfang der Definition von logischen Formeln. Nur die erste gestattet transparente, einfache und zugleich strukturell reichhaltige Formeldefinitionen. Grundlage dieses Formulierungskomforts ist die systematische Herleitung prädikatenlogischer Formeln aus Prädikatssymbolen und Termen. Auf diesem Fundament entfaltet sich das beträchtliche Ausdruckspotential der prädikatenlogischen Formeln. Es wurde durch die Konzipierung von sortierten Prädikatsdefinitionen nochmals erweitert. Das konventionelle Signaturkonzept läßt dagegen noch nicht einmal die formal korrekte Definition von Prädikatssymbolen zu. Es kann diese Prädikatssymbole durch seine Operationssymbole nicht unmittelbar abbilden[31)]. Denn die Operationssymbole besitzen stets eine Zielsorte, die für Prädikatssymbole grundsätzlich nicht definiert ist. Daher mußten oben die Wahrheitsfunktionssymbole eingeführt werden, um Prädikatssymbole im Signaturkonzepts mittelbar erfassen zu können. Diese konzeptionelle "Krücke" vermag aber nicht zu befriedigen, weil Wahrheitsfunktionssymbole sowohl im Signaturkonzept als in der Prädikatenlogik selten benutzte und nur schwer zu handhabende Konstrukte darstellen[32)]. Auch die breite Palette prädikatenlogischer Formeln, die sich aus Prädikatssymbolen herleiten lassen, findet im Signaturkonzept nur einen bescheidenen Abglanz. Er manifestiert sich in der Gestalt algebraischer Formeln für die Restriktionsformulierung in SIG-Spezifikation. Ihr Ausdruckspotential bleibt jedoch auf algebraische Relationen eingeschränkt, die darüber hinaus als bekannt vorausgesetzt werden müssen. Die algebraischen Formeln besitzen keine unmittelbare Rückführung auf Basiskonstrukte des Signaturkonzepts.

Um die vorgenannten Mängel des konventionellen Signaturkonzepts zu beheben, wird es im anschließenden Kapitel in prädikatenlogischer Richtung erweitert. Dabei werden Prädikatssymbole und daraus abgeleitete prädikatenlogische Formeln direkt in ein prädikatenlogisch modifiziertes Signaturkonzept eingebunden. Es resultieren schließlich algebraisch-prädikatenlogische Spezifikationen, die ein signaturbezogenes Pendant zur sortierten Prädikatenlogik darstellen.

Anmerkungen zum Kapitel:

1) Vgl. zu sortierten, mehrsortigen (many sorted) oder getypten (typed) Prädikatenlogiken SCHMIDT,A. (1938), S. 485ff., der eine sortierte Prädikatenlogik erstmals konsequent ausformulierte; SCHMIDT,A. (1951), S. 187ff., insbesondere S. 191ff.; WANG,HA. (1952), S. 105ff.; CARNAP (1960a), S. 83 u. 191f.; OBERSCHELP (1962), S. 297ff.; REITER (1978b), S. 57ff.; BIBEL (1982b), S. 132; SCHURZ (1983a), S. 356ff.; OHSUGA (1983), S. 304ff.; ALLEN,J. (1984), S. 128ff.; COHN (1987), S. 185ff.; SMOLKA (1988a), S. 183ff., insbesondere S. 187ff., der sich sogar auf eine sortierte Logik für die logische Programmierung bezieht; DORN (1988b), S. 205ff.; KIRCHNER (1988), S. 287ff. (mit explizitem Bezug auf das Signaturkonzept auf S. 289f.; SCHMITT,P. (1988), S. 79ff.; VON LUCK (1989), S. 278ff.; WALTHER,C. (1989), S. 65ff.; BOLLINGER (1989), S. 203ff.; OHLBACH (1989), S. 33ff.; DORN (1989), S. 13.

2) Es läßt sich auch die alternative Vorgehensweise vorstellen, das Signaturkonzept um das Ausdrucksvermögen der Prädikatenlogik zu bereichern. Dann wird die Signatur SIG=(SO,OP) um die Menge PRÄ der Prädikatssymbole zu einer prädikatenlogischen Signatur PSIG=(SO,OP,PRÄ) erweitert. Dieser Ansatz findet sich z.B. bei HOFBAUER (1989), S. 2ff. Beide Vorgehensweisen sind im Prinzip gleichwertig. Der Verf. wählt zunächst die Prädikatenlogik als Basiskonzept aus, weil sie in engerem Zusammenhang mit der logischen Programmiersprache PROLOG steht. Die Wichtigkeit dieser Sprache für die Implementierung von Netzmodellen mit Hilfe der Automatischen Informationsverarbeitung wurde schon mehrfach herausgestellt. Im anschließenden Kapitel wird er die komplementäre prädikatenlogische Signatur erläutern. Sie besitzt den Vorzug, später eine kompaktere formale Spezifikation von Synthetischen Netzen zuzulassen. Sortierte Prädikatenlogik und prädikatenlogische Signatur werden in dieser Arbeit so formuliert, daß sie sich gegenseitig austauschen lassen. Daher kann jeweils diejenige konzeptionelle Basis bevorzugt werden, die im jeweils aktuellen Argumentationskontext am geeignetsten erscheint.

3) Formeln, denen eine Sortenstruktur überlagert ist, werden auch kurz als sortierte Formeln bezeichnet. Entsprechend wird von sortierten Formelmengen gesprochen.

4) Jede Formel, die sich in einer sortierten Prädikatenlogik ausdrücken läßt, kann prinzipiell in einer unsortierten Prädikatenlogik äquivalent - wenn auch aufwendiger - reformuliert werden. Beispielsweise wird eine Allaussage betrachtet, die allen formalen Objekten ob der Sorte $sort_i$ eine Eigenschaft zuschreibt, die durch das einstellige sortierte Prädikatssymbol $Prä(sort_i)$ ausgedrückt wird. Die gleiche Allaussage läßt sich äquivalent als ein Subjugat aus zwei einstelligen, aber unsortierten Prädikatssymbolen $Sor_i(st_1)$ und $Prä(st_1)$ formulieren. Das zweite Prädikatssymbol drückt wiederum die Objekteigenschaft aus. Das erste Prädikatssymbol gibt dagegen die Sortenzugehörigkeit wieder, indem die Sortenzugehörigkeit wie eine zweite Objekteigenschaft behandelt wird. Daher gilt folgende Äquivalenz zwischen sortierter und unsortierter Allaussage:

$$\forall(ob \in OB_i): prä(ob)$$

$$\Leftrightarrow \quad \forall(ob \in OB): (sor_i(ob) \rightarrow prä(ob))$$

Äquivalenz bedeutet dabei, daß die Formel aus der sortierten Prädikatenlogik in eine Formel aus der unsortierten Prädikatenlogik so transformiert werden kann, daß gilt: Die erste Formel ist im sortierten Bereich genau dann beweisbar, wenn sich die zweite Formel im unsortierten Kontext beweisen läßt. Daher sind unsortierte und sortierte Erscheinungsformen der Prädikatenlogik 1. Stufe im Prinzip gleichwertig. Vgl. dazu SCHMIDT,A. (1938), S. 486 u. 493ff.; SCHMIDT,A. (1951), S. 193ff.; OBERSCHELP (1962), S. 298; SCHURZ (1983a), S. 357.
Umgekehrt bedeutet die Gleichwertigkeit von sortierter und unsortierter Prädikatenlogik, daß der größere Strukturreichtum einer sortierten Prädikatenlogik keine Einbuße ihrer Inferenzmächtigkeit bewirkt. Dies ist angesichts der früher erläuterten gegenläufigen Tendenz, die in formalen Logiken zwischen deren Ausdruckspotential und Inferenzmächtigkeit besteht, keineswegs selbstverständlich. Darüber hinaus hat z.B. OBERSCHELP (1962), S. 309ff., unabhängig von der voranstehenden Gleichwertigkeitsargumentation nachgewiesen, daß auch die sortierte Prädikatenlogik vollständig und korrekt ist.

5) In der sortierten Prädikatenlogik wird vom implizit einsortigen Ansatz der unsortierten, konventionellen Prädikatenlogik zu einem explizit mehrsortigen Ansatz übergegangen. Hierdurch wird die Reichhaltigkeit der internen Strukturierung prädikatenlogischer Formelargumente grundsätzlich vergrößert. Dies gestattet, Formeln innerhalb einer sortierten Prädikatenlogik erheblich einfacher und kompakter zu formulieren als ihrem unsortierten konventionellen Pendant; vgl. OBERSCHELP (1962), S. 297ff.; DORN (1989), S. 13.

6) Dessen ungeachtet hätten algebraische Ausdrucksmöglichkeiten aber auch auf andere - nicht sortierte - Weise zum Kalkül der konventionellen Prädikatenlogik hinzugefügt werden können.

7) Dies betrifft vor allem zwei Aspekte. Erstens lassen sich Inferenzen in sortierten Formelmengen oftmals schneller ausführen, als es - ceteris paribus - im unsortierten Fall möglich gewesen wäre. Denn die Inferenzoperationen können auf die jeweils sortengerechten Formelargumente beschränkt werden. Vgl. zu solchen Beschleunigungseffekten bei der Verwendung sortierter Formelmengen ESTER (1989), S. 127. Zweitens kann das Einhalten von Integritäts-

bedingungen dadurch unterstützt werden, daß Restriktionen für den Umgang mit Sorten aufgestellt werden. Vgl. dazu die Erläuterung "typ-inhärenter Konsistenzbedingungen" bei ESTER (1989), S. 31f., 34ff. u. 127.

8) Dabei wird vorausgesetzt, daß alle formalen Objekte aus dem prädikatenlogischen Definitionsbereich DB auch im neu geschaffenen prädikatenlogischen Universum OB enthalten sind: $DB \subseteq OB$. Die Umkehrung braucht allerdings nicht zu gelten. Denn das Universum OB kann nunmehr auch zusammengesetzte formale Objekte enthalten, die im Definitionsbereich DB der konventionellen Prädikatenlogik noch nicht vorgesehen waren. Darauf wird im folgenden Abschnitt eingegangen.

9) Vgl. dazu die analoge Zielsortendefinition für Operationssymbole im Kontext von SIG-Algebren. Die dort angesprochenen Operationssymbole stimmen mit den hier thematisierten Funktionssymbolen praktisch überein. Es gilt nur die eine Ausnahme, daß 0-stellige Operationssymbole zugelassen werden, nicht jedoch 0-stellige Funktionssymbole. Vgl. darüber hinaus die Erläuterung zu den algebraischen Objektmengen OB_i, deren Elemente sowohl atomare als auch beliebig komplex zusammengesetzte Objekte sein können.

10) Den Regelfall stellen aber disjunkte Teilbereiche dar.

11) Vgl. dazu die analogen Anmerkungen zur Strukturierung des algebraischen Universums OB_{SIG}.

12) Sie werden später für die Definition von Übergangsoperationen einer operationalen Semantik benötigt.

13) Der Potenzmengenoperator "pot" läßt auch die leere Teilmenge $TKAV_u = \emptyset$ zu. Dies berücksichtigt, daß die Faktenmenge $FAK_{u,r}$ auch leer sein kann. Da $prä_u(ob_1,...,ob_{K_u}) \in TKAF_u$ für $TKAF_u = \emptyset$ von keiner konstanten atomaren Formel $prä_u(ob_1,...,ob_{K_u})$ erfüllt werden kann, gilt für diesen degenerierten Fall: $FAK_{u,r} = \emptyset$.

14) Dabei wird von den Komplikationen abstrahiert, die aus der syntaktischen Notation des semantischen Faktenbegriffs resultieren. Dennoch kann von ihnen hier unter Vorgriff auf die Erläuterungen des nächsten Kapitels abgesehen werden. Denn dort wird die sortierte Prädikatenlogik als algebraisch-prädikatenlogische Spezifikation derart reformuliert, daß sich die Faktennotation "$fakt_r(...)$" in rein objektsprachlicher Weise ohne formale Schwierigkeiten bewältigen läßt.

15) Ebenso hätte die Familie $PFAK = (PFAK_u: u = 1,...,U)$ definiert werden. Wegen der Disjunktheit der potentiellen Faktenmengen ist sie mit der oben festgelegten Vereinigungsmenge äquivalent. Vgl. dazu die analogen Ausführungen zur systemspezifischen Faktenmenge FAK_r.

16) Sie wird als systemspezifische potentielle Faktenmenge bezeichnet.

17) Vgl. dazu die kompakte Darstellungsmöglichkeit von Faktenmultimengen und die Verdeutlichung von Symmetrien zwischen Konstrukten bei der Definition Synthetischer Netze.

18) Im Kontext einer Signatur SIG gehört die betrachtete Variable X innerhalb einer SIG-Termfamilie $TMF_{SIG}(VAF)$ zur sortenspezifischen Variablenmenge VA_i. Dieser Variablenmenge VA_i ist durch die zugehörige SIG-Algebra eine Menge OB_i zulässiger formaler Objekte (Konstanten) zugeordnet. Daher wird dort eine Variable X mit $X \in VA_i$ bei einer zulässigen Variablenbindung durch ein formales Objekt ob mit $ob \in OB_i$ substituiert. Da die sortenspezifischen Teilbereiche OB_i in einer sortierten Prädikatenlogik mit den sortenspezifischen Objektmengen OB_i des Signaturkonzepts zusammenfallen, ist das prädikatenlogische Quantorargument "$X \in OB_i$:" mit der signaturbezogenen Notation "$X \in VA_i$, $ob \in OB_i$ und X substituiert durch ob" gleichwertig.

19) Vgl. OBERSCHELP (1962), S. 299f.; COHN (1987), S. 185 (in leicht modifizierter Notation).

20) Semantisch definierte Wahrheitsfunktionen werden an den hier thematisierten syntaktischen Definitionsbereich dadurch angepaßt, daß zu Wahrheitsfunktionssymbolen übergegangen wird. Dies entspricht genau der gewöhnlichen prädikatenlogischen Korrespondenz zwischen semantisch definierten Funktionen und syntaktisch definierten Funktionssymbolen.

21) Die Sorten des Arguments eines Funktions- oder Prädikatssymbols können, müssen aber nicht verschieden sein. Im Grenzfall der Identität aller Sorten aus dem Argument liegt ein unsortiertes Funktions- bzw. Prädikatssymbol wie bei der konventionellen Prädikatenlogik vor.

22) Auch alle formalsprachlichen Konstrukte, die aus sortierten Funktions- oder Prädikatssymbolen abgeleitet werden, heißen sortierte Konstrukte.

23) Diese Prädikate stellen interpretierte atomare prädikatenlogische Formeln dar.

24) In den teilevaluierten Termen sind alle Konstanten- und Funktionssymbole aus dem zugrundeliegenden prädikatenlogischen Kalkül durch Konstanten bzw. Funktionen aus der zugeordneten Kalkülinterpretation ersetzt. Die teilevaluierten Terme können aber weiterhin Variablen enthalten. Daher sind die Argumente von Funktionen und Prädikaten nicht notwendig formale Objekte, wie sie im Signaturkonzept für die Argumente von Operationen definiert wurden.

25) Dies gilt auch dann, wenn keine explizite Bezugnahme auf eine sortierte Prädikatenlogik erfolgt. Darüber hinaus verzichtet der Verf. darauf, alle früher erfolgten Funktionsdefinitionen in sortierter Weise zu reformulieren. Für sie wird der Einfachheit halber unterstellt, sie jederzeit in eine sortierte Definition transformieren zu können.

26) Allerdings liegt keine vollständige Analogie vor. Denn die Sortenstruktur des algebraischen Universums OB, die durch seinen Aufbau aus den sortenspezifischen Objektmengen OB_i konstituiert wird, findet im unstrukturierten Definitionsbereich DB der konventionellen prädikatenlogischen Semantik keine Parallele.

27) Daneben werden aber auch Variablen durch Konstantensymbole substituiert. Dieser Aspekt wird an anderer Stelle noch speziell erörtert.

28) Der teilevaluierte Charakter von interpretierten Formeln der Prädikatenlogik wird besonders deutlich bei SMULLYAN (1968), S. 46f. Dort wird ausdrücklich herausgestellt, daß die Formelkomponenten nur Variablen oder Konstanten, nicht aber Konstantensymbole ("Parameter") sein dürfen.

29) Falls es sich bei den zugrundeliegenden uninterpretierten Formeln um Grundtermformel handelt, liegt ein Grenzfall vor. Dann stellen die Argumente der interpretierten Formeln jeweils konstante Tupel $(ob_1,...,ob_{Kj})$ dar. Daher *können* die interpretierten Formeln zwar noch Variablen enthalten, *müssen* es aber *nicht* unter allen Umständen.

30) Konstantensymbole werden in beiden Konzepten übereinstimmend jeweils durch atomare formale Objekte (Konstanten) interpretiert.

31) Formeln können vom Signaturkonzept nur in der bescheidenen Variante algebraischer Formeln in den Restriktionen von SIG-Spezifikationen berücksichtigt werden. Dort wird aber das prädikatenlogische Formulierungspotential bei weitem nicht ausgenutzt. Darüber hinaus werden syntaktische Formeldefinitionen und semantische Formelinterpretationen als solche überhaupt nicht unterschieden.

32) Daher hat der Verf. sie oben nur hilfsweise benutzt, um den theoretischen Zusammenhang zwischen Signaturkonzept und Prädikatenlogik erläutern zu können.

4.2.3.2.3 Algebraisch-prädikatenlogische Spezifikationen

Die algebraische Fundierung Synthetischer Netze durch das Signaturkonzept erreichte mit der Definition von SIG-Spezifikationen einen vorläufigen Abschluß. Weiterführende algebraische Konstrukte wurden im Kontext des Petrinetz-Konzepts - abgesehen von ihrer ohnehin noch seltenen Verwendung - bisher nicht vorgelegt[1)]. Das Signaturkonzept erweist sich jedoch als so fruchtbar, daß es eine deutlichere[2)] Einbindung prädikatenlogischer Konzepte gestattet. Zu diesem Zweck werden algebraisch-prädikatenlogische Spezifikationen eingeführt. Sie bereichern das Signaturkonzept um eine direkte Darstellung von Prädikatssymbolen.

Die früher vorgestellte Notation konventioneller Signaturen SIG durch strukturierte Listen wird um einen dritten Abschnitt "Präs" erweitert. Er dient der Definition von Prädikatssymbolen $\text{Prä}_u(st_1,...,st_{Ku})$ mit $u \in \{1,...,K_u\}$ und $K_u \in \mathcal{N}_+$. 0-stellige Prädikatssymbole bleiben grundsätzlich ausgeklammert[3)]. Die Prädikats- unterscheiden sich von den Operationssymbolen der Sektion "Ops" lediglich durch die fehlende Zielsorte. Mit PRÄ wird die Menge aller Prädikatssymbole bezeichnet, die in der Sektion "Präs" eingeführt werden. Diese Prädikatssymbole werden in der Menge PRÄ durch ihre eindeutig identifizierenden Prädikatsnamen Prä_u vertreten. Die Sortenmenge SO und die Menge OP aller Operationssymbole werden aus einer konventionell definierten Signatur übernommen. Das 3-Tupel PSIG=(SO,OP,PRÄ) stellt eine prädikatenlogisch erweiterte - oder kurz: prädikatenlogische - Signatur PSIG dar[4)].

Entsprechend wird die zugehörige Algebra zu einer prädikatenlogischen PSIG-Algebra erweitert. Zu diesem Zweck wird ein dritter Abschnitt "präs" für die Definition konstanter atomarer Formeln $\text{prä}_u(ob_1,...,ob_{Ku})$ ergänzt. Jede Formel $\text{prä}_u(ob_1,...,ob_{Ku})$ ist genau einem Prädikatssymbol $\text{Prä}_u(st_1,...,st_{Ku})$ aus der Sektion "Präs" der Signatur PSIG zugeordnet. Sie wird auch kurz als Prädikat bezeichnet. Die Deklaration eines solchen Prädikats besteht zunächst aus der Angabe der formalen Objektmengen $OB_{i(k)}$ für alle seine Argumente $k \in \{1,...,K_u\}$. Die Objektmengendeklaration wird als prä_u: $OB_{i(1)}$ x...x $OB_{i(Ku)}$ notiert. Die Menge KAF_u aller konstanten atomaren Formeln $\text{prä}_u(ob_1,...,ob_{Ku})$, die aufgrund dieser Objektmengendeklaration aus dem Prädikatssymbol $\text{Prä}_u(st_1,...,st_{Ku})$ gebildet werden können, ergibt sich wie bei der sortierten Prädikatenlogik im voranstehenden Kapitel zu:

$$KAF_u = \{\text{prä}_u(ob_1,...,ob_{Ku}) \colon ob_1 \in OB_{i(1)} \wedge ... \wedge ob_{Ku} \in OB_{i(Ku)}\}$$

Die Objektmengendeklaration eines Prädikats prä_u weicht von der Sektion "ops" einer konventionellen SIG-Algebra nur dadurch ab, daß keine Nachbereiche wie für Operationen existieren. Dadurch entfällt die Möglichkeit, eine Operationsvorschrift zu definieren. Statt dessen wird die formelspezifische Faktenmenge $FAK_{u.0}$ angegeben. Sie bezieht sich auf einen ausgezeichneten Grundzustand "r=0", in dem die PSIG-Algebra definiert wird[5)]. Die Faktenmenge kann ihre Extension durch Anwenden von Operationen aus der Sektion "ops" im Zeitablauf ändern. Dies reflektiert die früheren Erläuterungen zur Repräsentation von Wissen über zeitlich veränderliche Sachverhalte durch variable Faktenmengen[6)]. Die Faktenmenge $FAK_{u.0}$ ist endlich, kann dabei aber auch leer sein. Die Gesamtheit aus der Objektmengendeklaration und der Faktenmenge für eine konstante atomare Formel $\text{prä}_u(ob_1,...,ob_{Ku})$ heißt auch kurz die Deklaration des Prädikats prä_u.

Die Faktenmenge $FAK_{u.0}$ umfaßt alle konstanten atomaren Vorkommnisse der Formel prä_u, die im ausgezeichneten Grundzustand als Fakten gültig sind. Für diesen metasprachlichen Sachverhalt wird die früher erläuterte objektsprachliche Notation $\text{fakt}_0(\text{prä}_u(ob_1,...,ob_{Ku}))$ verwendet. Die prädikatenlogischen Komplikationen dieser Notationsweise entfallen hier[7)] aus drei Gründen. Erstens fehlt dem Signaturkonzept generell die komplikationsverursachende Differenzierung zwischen unterschiedlichen Konzeptstufen. Im Gegensatz zur Prädikatenlogik, für die hierarchisch gegliederte Konzeptvarianten 1., 2. und höherer Stufen definiert sind, handelt es sich grundsätzlich um ein flaches formalsprachliches Konzept. Zweitens wird der Ausdruck "fakt_0" nicht als eigenständiges Prädikatssymbol eingeführt, sondern als spezieller Notator verwendet. Damit wird aus prädikatenlogischer Perspektive sein metasprachlicher Charakter bewahrt, ohne das flache Signaturkonzept verlassen zu müssen. Drittens wird später eine PROLOG-Implementierung prädikatenlogischer Signaturen und Algebren im Kontext Synthetischer Netze vorgelegt, die sicherstellt, daß die faktischen Formelvorkommnisse $\text{fakt}_0(\text{prä}_u(\text{ob}_1,\ldots,\text{ob}_{Ku}))$ korrekt als gültige objektsprachliche Prädikatsvorkommnisse verarbeitet werden.

Jede Faktenmenge $\text{FAK}_{u.0}$ für den ausgezeichneten Grundzustand "r=0" läßt sich mit Hilfe einer Teilmenge TKAF_u der oben angeführten Menge konstanter atomarer Formeln des Prädikatssymbols $\text{Prä}_u(\text{st}_1,\ldots,\text{st}_{Ku})$ definieren. Dies geschieht analog zu den bereits eingeführten Faktenmengen $\text{FAK}_{u.r}$ der sortierten Prädikatenlogik[8)]:

$$\text{FAK}_{u.0} = \{\text{fakt}_0(\text{prä}_u(\text{ob}_1,\ldots,\text{ob}_{Ku})): \exists(\text{TKAF}_u \in \text{pot}(\text{KAF}_u)): \text{prä}_u(\text{ob}_1,\ldots,\text{ob}_{Ku}) \in \text{TKAF}_u\}$$

$$= \text{fak}_0(\text{TKAF}_u)$$

$$= \{\text{fakt}_0(\text{prä}_u(\text{ob}_1,\ldots,\text{ob}_{Ku}))::\ \text{ob}_1 \in \text{OB}_{i(1)} \wedge \ldots \wedge \text{ob}_{Ku} \in \text{OB}_{i(Ku)}\}$$

Die Menge PFAK_u aller potentiellen Fakten wird analog zu ihrer früheren Definition im Rahmen der sortierten Logik festgelegt:

$$\text{PFAK}_u = \{\text{fakt}(\text{prä}_u(\text{ob}_1,\ldots,\text{ob}_{Ku})): \text{prä}_u(\text{ob}_1,\ldots,\text{ob}_{Ku}) \in \text{KAF}_u\}$$

$$= \text{fak}(\text{KAF}_u)$$

Falls ein potentielles Fakt $\text{fakt}(\text{prä}_u(\text{ob}_1,\ldots,\text{ob}_{Ku}))$ im Grundzustand mit dem identifizierenden Index "0" tatsächlich ein Fakt darstellt, wird es als $\text{fakt}_0(\text{prä}_u(\text{ob}_1,\ldots,\text{ob}_{Ku}))$ notiert.

Für die potentielle Faktenmenge PFAK aller Prädikate prä_u, die in der Sektion "<u>prä</u>" deklariert wurden, gilt: $\text{PFAK} = \cup(u \in \{1,\ldots,U\}): \text{PFAK}_u$. Sie konstituiert das Alphabet einer abstrakten Faktensprache FAS mit FAS PFAK*. Jeder Buchstabe des Alphabets ist ein potentielles Fakt $\text{fakt}(\text{prä}_u(\text{ob}_1,\ldots,\text{ob}_{Ku}))$ mit $\text{Prä}_u \in \text{PRÄ}$. Jedes K-stellige Wort der Faktensprache stellt ein K-Tupel aus potentiellen Fakten dar. Wenn ein Wort mehrere potentielle Fakten enthält ($K \geq 2$), dann brauchen diese Fakten keineswegs verschieden zu sein[9)]. Falls die potentiellen Fakten als Listen notiert werden, gilt für die Faktensprache[10)]:

$$\text{PFAK}^0 = \{[\,]\}$$

$$\text{PFAK}^1 = \text{PFAK}$$

$$\text{PFAK}^K = \text{PFAK}\ x_1 \ldots x_{K-1}\ \text{PFAK} \quad \text{für alle } K \in \mathcal{N}_+ \text{ mit } K \geq 2$$

$$\text{PFAK}^+ = \cup(K \in \mathcal{N}_+): \text{PFAK}^K$$

$$\text{PFAK}^* = \text{PFAK}^+ \cup \text{PFAK}^0$$

$$\text{FAS} \subseteq \text{PFAK}^*$$

Die Familien OBF und OPF aller Objektmengen bzw. Operationen, die den Sorten bzw. Operationssymbolen aus der zugrundeliegenden Signatur PSIG zugeordnet sind, werden aus der konventionellen Definition von SIG-Algebren analog übernommen. FAK_0 ist die Familie ($\text{FAK}_{u.0}$: $u=1,\ldots U$) aller Faktenmengen $\text{FAK}_{u.0}$, die in der Sektion "<u>präs</u>" für jedes Prädikatssymbol Prä_u aus der Menge PRÄ spezifiziert werden. Das geordnete 3-Tupel $A_{\text{PSIG}} = (\text{OBF},\text{OPF},\text{FAK}_0)$ stellt die prädikatenlogische PSIG-Algebra dar, die der Signatur PSIG zugeordnet ist.

PSIG-Termfamilien und PSIG-Termauswertungen unterscheiden sich von konventionell definierten SIG-Termfamilien bzw. SIG-Termauswertungen auf den ersten Blick überhaupt nicht. Denn die betroffenen Terme sind sowohl für Operationssymbole und Operationen als auch für Prädikatssymbole und atomare konstante Formeln in derselben Weise definiert. Daher wird hier die induktive Formierung von Termen aus den Operationssymbolen der zugrundeliegenden Signatur PSIG und einer Variablenfamilie VAF nicht mehr behandelt. Die Variablen, aus denen die Familie VAF aufgebaut ist, werden in einer eigenständigen Sektion "Vars" durch sortenspezifische Variablenmengen $VA_i=\{X_{i.1},...,X_{i.Vi}\}$ mit $V_i \in \mathcal{N}_+$ explizit eingeführt[11]. Die konventionell definierte Termfamilie $TMF_{PSIG}(VAF)$ gilt als implizit vereinbart. Ebenfalls als bekannt vorausgesetzt werden die rekursiv definierte Familie $TA_{PSIG}=(eval_i: i=1,...,I)$ aller sortenspezifischen Auswertungsfunktionen $eval_i$ einer PSIG-Termauswertung. Gleiches gilt für deren Aufspaltungen in Teilauswertungsfunktionen $teval_i$ und Komplementauswertungsfunktionen $keval_i$ sowie die zugehörigen kanonischen Erweiterungen. Falls keine abweichenden Festlegungen erfolgen, werden jeweils minimale Teilauswertungsfunktionen und maximale Komplementauswertungsfunktionen unterstellt.

Es wurde mehrfach herausgestellt, daß in der Prädikatenlogik im allgemeinen nicht mit den rein syntaktisch definierten Termen, sondern mit teilevaluierten Termen aus einer prädikatenlogischen Semantik gearbeitet wird. Sie bestehen aus Konstanten, Variablen und Operatoren. Die teilevaluierten Terme gehen aus konventionell definierten Termen der Termfamilie $TMF_{PSIG}(VAF)$ dadurch hervor, daß Teilauswertungsfunktionen auf die Terme te sortengerecht angewendet werden: Wenn Term te ein Term der Sorte $sort_i$ aus der sortenspezifischen Termmenge $TM_i(VA_i)$ ist, die Menge OB_i formaler Objekte für die Sorte $sort_i$ in der PSIG-Algebra eingeführt wurde, $TTM_i(VA_i)$ die Menge aller teilevaluierten Terme aus der Sorte $sort_i$ ist und $teval_i$:$TM_i(VA_i) \rightarrow TTM_i(VA_i)$ eine sortenspezifische Teilauswertungsfunktion darstellt, dann ist $te'=teval_i(te)$ ein teilevaluierter Term. Dies wurde bereits an früherer Stelle ausführlich erläutert. Da als Regelfall minimale Teilauswertungsfunktionen vorausgesetzt werden, bleiben in den teilevaluierten Termen zunächst alle Variablen erhalten. Die Familie aller Mengen aus teilevaluierten Termen, die sich aus der zugrundeliegenden Signatur PSIG, ihrer Algebra A_{PSIG}, ihrer Termfamilie $TMF_{PSIG}(VAF)$ und ihrer Termauswertung TA_{PSIG} bilden lassen, ist die PSIG-Termfamilie $TTMF(VAF)=(TTM_i(VA_i): i=1,...,I)$ mit $VAF=(VA_i: i=1,...,I)$[12]. In der Sektion "terms" erfolgt die induktive Definition dieser Termfamilie[13].

Für den Umgang mit teilevaluierten Termen, die aus der Anwendung von (minimalen) Teilauswertungsfunktionen $teval_i$ hervorgegangen sind, reichen in einer prädikatenlogischen Signatur (maximale) Komplementauswertungsfunktionen $keval_i$ aus. Jede Komplementauswertungsfunktion $keval_i$: $TTM_i(VA_i) \rightarrow OB_i$ bildet per definitionem jeden teilevaluierten Term te' der Sorte $sort_i$ auf ein sortengleiches formales Objekt "ob" aus der Menge OM_i ab. Dabei werden auch alle Variablen, die der teilevaluierte Term te' enthielt, im formalen Objekt "ob" sortengerecht auf Konstanten abgebildet. Die formale Handhabung der Komplementauswertungsfunktionen $keval_i$ erweist sich aber als recht aufwendig, weil sie die Verknüpfung mit den jeweils zugrundeliegenden Teilauswertungsfunktionen erfordert. Darüber hinaus wird in prädikatenlogischen Formelsystemen eine solche Teilauswertung nicht explizit vollzogen, sondern als Teilaspekt ihrer Semantik implizit vorausgesetzt. Beides wird hier dadurch berücksichtigt, daß die algebraischen Komplementauswertungsfunktionen in das prädikatenlogische Konzept der Variablenbelegungen integriert werden.

Die prädikatenlogischen Variablenbelegungen V_c werden mit Hilfe der Komplementauswertungsfunktionen $keval_i$ um den Aspekt der Sortentreue bereichert[14]. Zu diesem Zweck wird jede Variablenbelegung V_c als eine sortentreue Variablenbindungsfunktion vb_c ausgestaltet. Die Variablenbindungsfunktion vb_c ist die Zusammenfassung von K_c sortenspezifischen Komplementauswertungsfunktionen $keval_{i(k)}$ mit $k \in \{1,...,K_c\}$ und $K_c \in \mathcal{N}_+$. Sie entspricht somit formal den früher vorgestellten operationssymbolspezifischen kanonischen Erweiterungen $keval_j$ von Komplementauswertungsfunktionen auf Termtupel, ist aber nicht auf die K_J-stelligen Operationssymbole Op_j beschränkt. Statt dessen kann eine Variablenbindungsfunktion vb_c auf jedes K_c-stellige Termtupel $(te_1,...,te_{Kc})$ angewendet werden, das aus dem kartesischen Produkt jener sortenspezifischen Mengen teilevaluierter Terme stammt, das den Definitionsbereich der Variablenbindungsfunktion vb_c bildet[15]:

$$vb_c: \quad TTM_{i(1)}(VA_{i(1)}) \times ... \times TTM_{i(Kc)}(VA_{i(Kc)}) \rightarrow OB_{i(1)} \times ... \times OB_{i(Kc)}$$
$$(te_1,...,te_{Kc}) \rightarrow vb_c(te_1,...,te_{Kc}) = (ob_1,...,ob_{Kc})$$

mit:

$$\forall(k \in \{1,...,K_c\}): ob_k = keval_{i(k)}(te_k)$$

VBM ist die Menge aller sortentreuen Variablenbindungsfunktionen vb_c[16], die für teilevaluierte Terme aus der Termfamilie TTMF(VAF) mit der Hilfe von Komplementauswertungsfunktionen $keval_i$ definiert werden können. Sie wird wie die Termfamilie TTMF(VAF) in der Sektion "terms" aufgeführt. Es wird vorausgesetzt, daß immer mindestens eine Variablenbindungsfunktion vb_c definiert ist. Daher gilt für die Anzahl C aller definierten Variablenbindungsfunktionen auf jeden Fall $C \geq 1$. Diese Prämisse ist bei nicht-leeren Objektmengen OB_i immer erfüllt. Darüber läßt die Einschränkung auf endliche Objektmengen OB_i jeweils nur endlich viele verschiedene Variablenbindungsfunktionen vb_c in Systemen aus jeweils endlich vielen Formeln zu[17]. Folglich genügt die Bindungsfunktionenanzahl stets der Bedingung $C \in \mathcal{N}_+$. Die Menge VBM aller denkmöglichen Variablenbindungsfunktionen ist daher in dieser Arbeit stets eine endliche Menge.

Gegenüber SIG-Spezifikationen entfällt in der prädikatenlogischen Erweiterung des Signaturkonzepts die Beschränkung auf algebraische Formeln. Statt dessen werden für die Restriktionsformulierung alle prädikatenlogisch definierbaren Formeln zugelassen. Sie brauchen nicht explizit definiert zu werden, sofern es sich um Standardformeln handelt. Sie gelten als implizit vereinbart und werden in der Menge $FOR_{standard}$ aller Standardformeln zusammengefaßt[18]. Nur extraordinäre prädikatenlogische Formeln werden explizit definiert. Die Anzahl Z aller explizit spezifizierten Formeln kann daher auch Null betragen; folglich gilt $Z \in \mathcal{N}_0$. Darüber hinaus werden die Restriktionen nicht mehr auf die Operations- und Prädikatssymbole der zugrundeliegenden Signatur PSIG bezogen, sondern auf die Operationen und Prädikatsdeklarationen der zugeordneten PSIG-Algebra. Der Grund hierfür liegt im Übergang von den algebraisch definierten Termen aus der SIG-Termfamilie $TMF_{SIG}(VAF)$ zu den prädikatenlogisch bevorzugten teilevaluierten Termen aus der PSIG-Termfamilie $TTMF_{PSIG}(VAF)$. Dadurch gehen die algebraischen Restriktionsformeln, deren Argumente in SIG-Spezifikationen aus Termen bestanden[19], in prädikatenlogische Restriktionsformeln über, deren Argumente aus teilevaluierten Termen te_k zusammengesetzt sind[20]. Diese Veränderung des Formelcharakters schlägt sich in der Notation der Restriktionsfestlegung nieder: Die Sektion "Res" aus SIG-Spezifikationen wird nunmehr als "res" notiert. Die Menge aller Restriktionsformeln wird aber weiterhin mit RES bezeichnet. Sie umfaßt alle explizit spezifizierten Formeln sowie alle implizit vereinbarten Standardprädikate aus der Menge $FOR_{standard}$:

$$RES = \begin{cases} FOR_{standard}; & \text{für } Z=0 \\ \{for_z(te_1,...,te_{Kz}): z=1,...,Z\} \cup FOR_{standard}; & \text{für } Z \in \mathcal{N}_+ \end{cases}$$

Die Gesamtheit aller voranstehend erläuterten prädikatenlogischen Erweiterungen des Signaturkonzepts stellt eine algebraisch-prädikatenlogische Spezifikation $SPEC_{PSIG}$ dar[21]. Im Gegensatz zur früheren engen Definition von SIG-Spezifikationen erstreckt sie sich also nicht nur auf die Kombination einer Signatur SIG=(SO,OP) mit der Restriktionsmenge RES. Vielmehr wird die PSIG-Spezifikation so weit ausgelegt, daß sie auch die oben erläuterten Konstrukte der PSIG-Algebra, der PSIG-Termfamilie sowie der PSIG-Variablenbindung umfaßt. Dieses umgreifende

Spezifikationsverständnis wird in der nachfolgenden Definition zusammengefaßt. Es liegt dieser Arbeit fortan zugrunde.

Definition: algebraisch-prädikatenlogische Spezifikation

Eine algebraisch-prädikatenlogische PSIG-Spezifikation ist ein geordnetes 9-Tupel $SPEC_{PSIG}$= (SO,OP,PRÄ;OBF,OPF,FAK_0;TTMF(VAF),VBM,RES) mit zugrundeliegender prädikatenlogischer Signatur PSIG=(SO,OP,PRÄ) genau dann, wenn gilt:

- SO ist eine endliche, nicht-leere Menge $\{sort_i: i=1,...,I\}$ aus Sorten $sort_i$ mit $I \in \mathcal{N}_+$.
- OP ist eine endliche, nicht-leere Menge $\{Op_j: j=1,...,J\}$ aus K_j-stelligen Operationssymbolen Op_j mit $K_j \in \mathcal{N}_0$ für alle $j \in \{1,...,J\}$ und $J \in \mathcal{N}_+$.
- PRÄ ist eine endliche, nicht-leere Menge $\{Prä_u: u=1,...,U\}$ aus K_u-stelligen Prädikatssymbolen $Prä_u$ mit $K_u \in \mathcal{N}_+$ für alle $u \in \{1,...,U\}$ und $U \in \mathcal{N}_+$.
- Zwischen der Sorten-, Operationssymbol- und Prädikatssymbolemenge bestehen die Beziehungen $OP \subseteq (SO^* \times SO)$ und $PRÄ \subseteq SO^+$.
- OBF ist eine Familie $(OB_i: i=1,...,I)$ aus Mengen formaler Objekte. Jeder Sorte $sort_i$ aus der zugrundeliegenden Signatur PSIG ist genau eine sortenspezifische Objektmenge OB_i zugeordnet.
- OPF ist eine Familie $(op_j: j=1,...,J)$ aus Operationen op_j. Jedem Operationssymbol Op_j aus der zugrundeliegenden Signatur PSIG ist genau eine Operation op_j zugeordnet.
- FAK_0 ist eine Familie $(FAK_{u.0}: u=1,...U)$ aus Faktenmengen $FAK_{u.0}$. Jedem Prädikatssymbol $Prä_u$ aus der zugrundeliegenden Signatur PSIG ist genau eine Faktenmenge $FAK_{u.0}$ zugeordnet.
- TTMF(VAF) ist die Familie aller sortenspezifischen Mengen aus teilevaluierten Termen, die aus den Operationssymbolen und Operationen zusammen mit einer exogen vorgegebenen Variablenfamilie VAF durch Anwenden von beliebigen Teilauswertungsfunktionen gebildet werden können.
- VBM ist die Menge $\{vb_c: c=1,...,C\}$ aller sortentreuen Variablenbindungsfunktionen vb_c mit $C \in \mathcal{N}_+$, die teilevaluierte Terme auf formale Objekte aus den Mengen der Familie OBF abbilden.
- RES ist eine Menge $\{for_z: z=1,...,Z\}$ von interpretierten mehrstelligen prädikatenlogischen Restriktionsformeln for_z mit $Z \in \mathcal{N}_0$. Ihre Argumente bestehen aus teilevaluierten Termen.

Die formalen Konstrukte einer PSIG-Spezifikation werden jeweils nach folgendem Schema festgelegt:

<<u>Spezifikationsname</u>> =

<u>sorts</u>: $sort_i$

...

$sort_I$

$SO = \{sort_i: i=1,...,I\}$

<u>Ops</u>: Op_1: $sort_{i(1.1)} \ldots sort_{i(1.K1)} \rightarrow sort_{i(1.K1+1)}$ 22)

...

Op_J: $sort_{i(J.1)} \ldots sort_{i(J.KJ)} \rightarrow sort_{i(J.KJ+1)}$

$OP = \{Op_j: j=1,...,J\}$

<u>Präs</u>: $Prä_1(sort_{i(1.1)} \ldots sort_{i(1.K1)})$

...

$Prä_U(sort_{i(U.1)} \ldots sort_{i(U.KU)})$

$PRÄ = \{Prä_u: u=1,...,U\}$

<u>OBs</u>: OB_i

...

OB_I

$OBF = (OB_i: i=1,...,I)$

<u>ops</u>: op_1: $OB_{i(1.1)} \times \ldots \times OB_{i(1.K1)} \rightarrow OB_{i(1.K1+1)}$

$(ob_i,...,ob_{K1}) \rightarrow op_1(ob_i,...,ob_{K1}) = ob_{K1+1}$ 23)

...

op_J: $OB_{i(J.1)} \times \ldots \times OB_{i(J.KJ)} \rightarrow OB_{i(J.KJ+1)}$

$(ob_i,...,ob_{KJ}) \rightarrow op_J(ob_i,...,ob_{KJ}) = ob_{KJ+1}$

$OPF = (op_j: j=1,...,J)$

<u>präs</u>: $prä_1$: $OB_{i(1.1)} \times \ldots \times OB_{i(1.K1)}$

$prä_1(ob_i,...,ob_{KU})$

$FAK_{1.0} = \{fakt_0(prä_1(ob_1,...,ob_{K1})):: ob_1 \in OB_{i(1.1)} \wedge \ldots \wedge ob_{K1} \in OB_{i(1.K1)}\}$

...

$prä_U$: $OB_{i(U.1)} \times \ldots \times OB_{i(U.KU)}$

$prä_U(ob_i,...,ob_{KU})$

$FAK_{U.0} = \{fakt_0(prä_U(ob_1,...,ob_{KU})):: ob_1 \in OB_{i(U.1)} \wedge \ldots \wedge ob_{K1} \in OB_{i(U.KU)}\}$

$FAK_0 = (FAK_{u.0}: u=1,...,U)$

Vars: $VA_1 = \{X_{i.1},...,X_{i.V1}\}$

...

$VA_I = \{X_{I.1},...,X_{i.VI}\}$

$VAF = (VA_i\!: i=1,...,I)$

terms: $KB_1 = \{te: \exists(op_j \in OPF): op_j: \rightarrow OB_1 \wedge op_j() = te\}$

$TTM_1(VA_1)$ minimal mit:

$$
\begin{aligned}
& KB_1 = \emptyset \rightarrow OB_1 \subseteq TTM_1(VA_1) \\
\wedge\ & KB_1 \neq \emptyset \rightarrow KB_1 \subseteq TTM_1(VA_1) \\
\wedge\ & VA_1 \subseteq TTM_1(VA_1) \\
\wedge\ & (\forall(op_j \in OPF): \\
& \quad (op_j\!: OB_{i(j.1)} \times ... \times OB_{i(j.KJ)} \rightarrow OB_{i(j.Kj+1)} \\
& \quad \wedge\ (\forall(k \in \{1,...,K_j\}): te_k \in TTM_{i(j.k)}(VA_{i(j.k)})) \\
& \quad \wedge\ \iota(j.K_j+1) = 1 \ \wedge\ op_j(te_1,...,te_{KJ}) = te_{KJ+1}) \\
& \rightarrow te_{KJ+1} \in TTM_1(VA_1))
\end{aligned}
$$

...

$KB_I = \{te: \exists(op_j \in OPF): op_j: \rightarrow OB_I \wedge op_j() = te\}$

$TTM_I(VA_I)$ minimal mit:

$$
\begin{aligned}
& KB_I = \emptyset \rightarrow OB_I \subseteq TTM_I(VA_I) \\
\wedge\ & KB_I \neq \emptyset \rightarrow KB_I \subseteq TTM_I(VA_I) \\
\wedge\ & VA_I \subseteq TTM_I(VA_I) \\
\wedge\ & (\forall(op_j \in OPF): \\
& \quad (op_j\!: OB_{i(j.1)} \times ... \times OB_{i(j.KJ)} \rightarrow OB_{i(j.Kj+1)} \\
& \quad \wedge\ (\forall(k \in \{1,...,K_j\}): te_k \in TTM_{i(j.k)}(VA_{i(j.k)})) \\
& \quad \wedge\ \iota(j.K_j+1) = I \ \wedge\ op_j(te_1,...,te_{KJ}) = te_{KJ+1}) \\
& \rightarrow te_{KJ+1} \in TTM_I(VA_I))
\end{aligned}
$$

$TTMF(VAF) = (TTM_i(VA_i)\!: i=1,...,I)$

$VBM = \{vb_c\!: c=1,...,C \wedge C \in \mathcal{N}_+ \wedge (\forall(c \in \{1,...,C\}): K_c \in \mathcal{N}_+ \wedge ...$

$\quad vb_c\!: TTM_{i(1)}(VA_{i(1)}) \times ... \times TTM_{i(Kc)}(VA_{i(Kc)}) \rightarrow OB_{i(1)} \times ... \times OB_{i(Kc)})\}$

res: a) entweder für Z=0:

$RES = FOR_{standard}$

b) oder aber für $Z \in \mathcal{N}_+$:

$for_1(te_1,...,te_{K1})$

...

$for_z(te_1,...,te_{Kz})$

mit:

$$\forall(z \in \{1,\dots,Z\}): K_z \in \mathcal{N}_+ \wedge (\forall(k \in \{1,\dots,K_z\}: te_k \in \mathrm{TTMF(VAF)})$$

$$\mathrm{RES} = \{for_z(te_1,\dots,te_{Kz}): z = 1,\dots,Z\} \cup \mathrm{FOR}_{standard}$$

Die ausgesprochen kompakte und strukturell übersichtliche Darstellungsweise von algebraisch-prädikatenlogischen Spezifikationen rechtfertigt nochmals, das - nunmehr prädikatenlogisch erweiterte - Signaturkonzept als Basis für Synthetische Netze auszuzeichnen. Insbesondere ist hervorzuheben, daß die prädikatenlogischen Schwierigkeiten bei der Faktenrepräsentation im Signaturkonzept nicht wirksam werden. Darüber hinaus umfassen PSIG-Spezifikationen bereits den gesamten syntaktischen *und* den semantischen Darstellungsaspekt der Prädikatenlogik. Ein Vergleich von Signatur und Algebra mit dem früher dargestellten, rein syntaktisch definierten Kalkül der sortierten Prädikatenlogik läßt den Kompaktheitsvorteil deutlich zu Trage treten.

Anmerkungen zum Kapitel:

1) Vgl. dazu den Abschluß signaturbasierter Petrinetz-Darstellungen mit SIG-Spezifikationen bei KRÄMER (1987a), S. 277ff.; RECK (1988), S. 48ff. i.V.m. S. 81ff.; REISIG (1989a), S. 6f. i.V.m. S. 10ff., insbesondere S. 12.

2) Prädikatenlogische Ansätze enthält das Signatur-Konzept bereits in den algebraischen Formeln von SIG-Spezifikationen. Dort bleibt der prädikatenlogische Beitrag jedoch auf die Qualität eines Hilfskonzepts für die Formelexpression beschränkt.

3) Dies mag auf den ersten Blick erstaunen, weil früher für konventionelle und sortierte Prädikatenlogik auch 0-stellige Prädikatssymbole zugelassen wurden. Dennoch ist der o.a. und nachfolgend stets vorausgesetzte Ausschluß 0-stelliger Prädikatssymbole gerechtfertigt. Denn es wird später gezeigt werden, daß in Netzmodellen alle 0-stelligen aussagenlogischen Formeln durch 1-stellige Formeln substituiert werden. Daher werden in den anschließenden Ausführungen, die ausschließlich der Entfaltung von Netzmodellen dienen, 0-stellige Formeln grundsätzlich nicht mehr berücksichtigt.

4) Das charakteristische 3-Tupel (SO,OP,PRÄ) der prädikatenlogisch erweiterten Signatur findet sich auch bei HOFBAUER (1989), S. 2ff. Allerdings hebt er den prädikatenlogischen Charakter gegenüber dem Sortierungsaspekt deutlich hervor.

5) Die Faktenmenge kann ihre Extension durch Anwenden von Operationen aus der Sektion "<u>ops</u>" auf die Argumente der atomaren Formel $\text{prä}_u(ob_1,...,ob_{Ku})$ im Zeitablauf ändern.

6) Allerdings wird die tatsächliche Veränderung der Faktenmengen nicht innerhalb des Signaturkonzepts dargestellt, das weiterhin statisch definiert bleibt. Vielmehr werden die kinetischen und dynamischen Aspekte erst später durch Synthetische Netze formal abgebildet.

7) Daher kann auch in der Notation $\text{fakt}_0(\text{prä}_u(ob_1,...,ob_{Ku}))$ auf den differenzierenden Index "f" verzichtet werden, der eingeführt wurde, um auf die funktionale Deutung einer prädikatenlogischen Formel zu verweisen.

8) Von den beiden nachfolgenden Definitionsvarianten der Faktenmenge $FAK_{u.0}$ präferiert der Verf. die erste, weil sie sich im Rahmen der konventionellen Mengennotation bewegt. Der konventionelle Mengenbildungsoperator ":" bedeutet in der ersten Variante $FAK_{u.0}=\{\text{fakt}_0(\text{prä}_u(ob_1,...,ob_{Ku})): \exists(TKAF_{u.0}\in \text{pot}(KAF_u)): \text{prä}_u(ob_1,..., ob_{Ku})\in TKAF_u\}$, daß *alle* konstanten atomaren Formeln, welche die Bedingung $\text{prä}_u(ob_1,...,ob_{Ku})\in TKAF_u$ erfüllen, in der Menge $FAK_{u.0}$ als Fakten $\text{fakt}_0(\text{prä}_u(ob_1,...,ob_{Ku}))$ enthalten sind. Dies entspricht dem konventionellen Mengenbildungsoperator. In der zweiten Definitionsvariante $FAK_{u.0}=\{\text{fakt}_0(\text{prä}_u(ob_1,...,ob_{Ku}))::\ ob_1\in OB_{i(1)}\wedge ... \wedge ob_{Ku}\in OB_{i(Ku)}\}$ bedeutet dagegen der modifizierte Mengenbildungsoperator "::" nur, daß *manche* der formalen Objekte $ob_1,...,ob_{Ku}$, welche die Bedingung $ob_1\in OB_{i(1)}\wedge ... \wedge ob_{Ku}\in OB_{i(Ku)}$ erfüllen, die Argumente von Fakten $\text{fakt}_0(\text{prä}_u(ob_1,...,ob_{Ku}))$ aus der Menge $FAK_{u.0}$ bilden. Auf diese beiden unterschiedlichen Operatordefinitionen wurde bereits in einer früheren Anmerkung hingewiesen. Trotz der unkonventionellen Modifizierung des Mengenbildungsoperators wird in dieser Arbeit die zweite Definitionsvariante zugelassen. Dafür sprechen zwei Gründe. Erstens läßt sich die Faktenmenge $FAK_{u.0}$ kompakter definieren. Zweitens wird es weiter unten möglich, auf diese Weise PSIG-Spezifikationen zu formulieren, ohne die Menge KAF_u und ihre Teilmengen $TKAF_{u.0}$ einführen zu müssen. Auch das vereinfacht die Notation.

9) Dies läßt später die Definition multipler Fakten und entsprechender Multimengen zu.

10) Vgl. dazu die analoge Definition einer Sortensprache.

11) Es wurde bereits dargelegt, daß es sich bei Variablen um Konstrukte sui generis handelt, die von den Konstanten-, Operations- und Prädikatssymbolen nicht überdeckt werden.

12) Die Termmengenfamilie TTMF(VAF) wird später in das Definitionstupel von algebraisch-prädikatenlogischen Spezifikationen PSIG explizit aufgenommen. Daher kann - im Gegensatz zur früher rein algebraisch definierten Termmengenfamilie $TMF_{SIG}(VAF)$ - hier der kennzeichnende Index "PSIG" entfallen. Die teilevaluierten Terme aus der Termmengenfamilie heißen auch einfach "Terme", wenn aus dem Argumentationskontext ihre Ableitung aus prädikatenlogisch erweiterten Signaturen PSIG ersichtlich ist.

13) Die Definition der Sektion "terms" wurde schon eingeführt. Sie bedarf daher hier keiner näheren Erläuterung. Die frühere Sektionsdefinition wird hier lediglich um die Variablenbindungsfunktionen vb_c bereichert. Darauf wird anschließend eingegangen.

14) Damit wird die frühere Andeutung, prädikatenlogische Variablenbelegungen würden den Komplementauswertungsfunktionen des Signaturkonzepts entsprechen, präzisiert und vertieft.

15) Der Definitionsbereich einer Variablenbindungsfunktion kann aus beliebigen Mitgliedern der teilevaluierten Termfamilie TTMF(VAF) aufgebaut werden. Er darf auch aus genau einer sortenspezifischen Menge $TTM_i(VA_i)$ teilevaluierter Terme bestehen (degeneriertes kartesisches Produkt). Dann bewirkt die Variablenbindungsfunktion

nur, daß jeweils einer einzelnen Variablen aus der Menge VA_i ein sortengleiches formales Objekt zugeordnet wird. Ebenso ist es möglich, daß der Definitionsbereich einer Variablenbindungsfunktion das kartesische Produkt *aller* Mitglieder aus der Termfamilie TTMF(VAF) ist. Im letzten Fall führt die Variablenbindungsfunktion vb_c dazu, daß *alle* Variablen eines prädikatenlogischen Formelsystems durch formale Objekte sortengerecht substituiert werden. Dies wird im allgemeinen unterstellt, wenn eine Variablenbindungsfunktion vb_c als sortentreue Erweiterung einer konventionellen prädikatenlogischen Variablenbelegung V_c betrachtet wird. Statt dessen können aber Variablenbindungsfunktionen auch auf Ausschnitte eines Formelsystems angewendet werden, wie z.B. auf einzelne ausgesuchte Formeln oder sogar nur auf einzelne Variablen. Diese flexible Festlegung gestattet es, später Variablenbindungen vorzunehmen, die jeweils nur lokal zu den Variablen einer einzelnen zu schaltenden Transition gehören.

16) Die Menge VBM wird daher auch als PSIG-Variablenbindung bezeichnet.

17) Zwei Variablenbindungsfunktionen heißen genau dann verschiedenartig, wenn sie sich nicht auf die gleichen Variablenmengen erstrecken. In einem endlichen Formelsystem können nur endlich viele Variablen vorkommen. Daher können auch nur endlich viele verschiedenartige Variablenbindungsfunktionen definiert werden. Denn bei einer endlichen Variablenanzahl ist auch die Anzahl der unterschiedlichen Variablenmengen, durch die sich verschiedenartige Variablenbindungsfunktionen unterscheiden können, endlich.
Zwei gleichartige Variablenbindungsfunktionen sind aber immer noch verschieden, wenn sie sich zwar auf die gleichen Variablenmengen erstrecken, aber mindestens eine Variable durch jeweils ein anderes formales Objekt binden. Bei endlichen Mengen OB_i formaler Objekte, die für diese sortentreue Variablenbindung vorgegeben sind, können aber nur endlich viele verschiedene formale Objekte für dieselbe Variable eingesetzt werden. Darüber hinaus führt die Endlichkeit der Variablenanzahl im Formelsystem dazu, daß die gleichartigen Variablenbindungsfunktionen jeweils nur über endliche Variablenmengen erstrecken können. Die doppelte Endlichkeit der Variablenmengen und der für jede Variable einsetzbaren formalen Objekte impliziert, daß nur endlich viele gleichartige, aber verschiedene Variablenbindungsfunktionen existieren können.

18) Für die implizite Vereinbarung von Prädikaten wird jeweils eine Implementierungsumgebung für prädikatenlogische Formelsysteme vorausgesetzt. In dieser Arbeit handelt es sich um die Programmiersprache Turbo-PROLOG und das Softwarepaket PASIPP. Alle dort definierten Prädikate brauchen nicht mehr explizit eingeführt werden. Sie bilden die wichtigste Teilmenge der Menge aller Standardformeln dieser Arbeit. Vgl. dazu die Definitionen von Turbo-PROLOG-Prädikaten in PROLOG (o.J.), S. 179ff.; KINNEBROCK (1988), S. 131ff. Hinzu kommen als Standardformeln solche prädikatenlogischen Formeln, die an dieser Arbeit bereits explizit eingeführt worden sind. Dazu gehören inbesondere die Vergleichs- und Anordnungsprädikate, die Identitäts- und Elementprädikate sowie die Minimierungs- und Maximierungsprädikate. Da sich alle Relationen als mehrstellige Prädikate auffassen lassen, werden auch die konventionellen arithmetischen Relationen zu den Standardformeln gerechnet. Zum großen Teil sind sie auch schon in den vorgenannten Vergleichsprädikaten enthalten.

19) Daher stellten die algebraischen Restriktionsformeln aus prädikatenlogischer Sicht rein syntaktisch definierte, d.h. uninterpretierte Formeln dar.

20) Folglich handelt es sich bei den prädikatenlogischen Restriktionsformeln um teilinterpretierte Formeln.

21) Sie wird fortan auch kurz als PSIG-Spezifikation angesprochen.

22) Darin sind nullstellige Operationssymbole (Konstantensymbole) $Op_j: \rightarrow sort_{i(j)}$ eingeschlossen. Für diesen Zweck wird vereinbart, daß in jedem Operationssymbol Op_j mit $Op_j: sort_{i(j.1)} \dots sort_{i(j.Kj)} \rightarrow sort_{i(j.Kj+1)}$ für $K_j=0$ alle Argumentsorten $sort_{i(j.1)},\dots,sort_{i(j.Kj)}$ ersatzlos gestrichen werden. Zusätzlich wird die Notation der Zielsorte $sort_{i(j.Kj+1)}$ zu $sort_{i(j)}$ vereinfacht.

23) Darin sind nullstellige Operationen (Konstanten) $op_j: \rightarrow OB_{i(j)}$ mit $op_j()=ob$ eingeschlossen. Für diesen Zweck wird vereinbart, daß in jeder Operation op_j mit $op_j: OB_{i(j.1)} \dots OB_{i(j.Kj)} \rightarrow OB_{i(j.Kj+1)}$ mit $\rightarrow op_j(ob_i,\dots,ob_{Ki})=ob_{Ki+1}$ für $K_j=0$ alle Objektmengen $OB_{i(j.1)},\dots,OB_{i(j.Kj)}$ ersatzlos gestrichen werden. Zusätzlich wird die Notation der Objektmenge $OB_{i(j.Kj+1)}$ zu $OB_{i(j)}$ vereinfacht. Darüber hinaus wird die Abbildungsvorschrift auf den Ausdruck $\rightarrow op_j()=ob$ reduziert.

4.2.3.2.4 Exemplarische Verdeutlichung

Abschließend wird anhand eines Beispiels[1] der strukturelle Formulierungsreichtum verdeutlicht, der aus der Überlagerung einer Sortenstruktur über konventionell definierte Prädikate hervorgeht. Es illustriert sowohl das Konzept der sortierten Prädikatenlogik als auch die prädikatenlogische Erweiterung des Signaturkonzepts. Ziel ist es, Objekte darzustellen, die jeweils durch Objekteigenschaften - die Attribute - vollständig definiert sind. Dabei wird zweistufig vorgegangen. Zunächst wird im Rahmen einer konventionellen, unsortierten Prädikatenlogik eine präzise Objektrepräsentation dadurch gewonnen, daß zwischen Attributnamen, Attributausprägungen und Definitionsbereichen zulässiger Attributausprägungen differenziert wird. Anschließend wird diese klare, aber auch relativ aufwendige Objektrepräsentation durch die Überlagerung einer Sortenstruktur wesentlich kompakter gestaltet, ohne hierbei an formaler Präzision einzubüßen.

Jedes Objekt, das durch Q Attribute mit $Q \in \mathcal{N}_+$ charakterisiert ist[2], wird als Element einer Objektklasse aufgefaßt. Diese Klasse ist als die Gesamtheit aller Objekte mit derselben Attributstruktur festgelegt[3]. Die klassenspezifische Attributstruktur besteht aus einer Q-stelligen Funktion "obj". Sie bildet das Q-fache kartesische Produkt über der Menge "SYMBOL" aller symbolischen Ausdrücke auf eben diese Menge ab. Jede Stelle aus dem Argument der Funktion "obj" vertritt ein Bild einer 1-stelligen attributspezifischen Funktion att_q. Jede attributspezifische Funktion att_q stellt ein Attribut mit dem Namen att_q dar. Es bildet den attributspezifischen Definitionsbereich DB_q zulässiger Attributausprägungen at_q auf die abstrakte Menge "SYMBOL" ab. Jedes Objekt, das die Attributstruktur der voranstehend definierten Objektklasse erfüllt, ist dann ein Bild der klassenspezifischen Funktion "obj". Ein solches Objekt besitzt eine tiefe, zweifach hierarchisch gegliederte interne Struktur. Auf der ersten Stufe ist es aus den Namen att_q seiner Q Attribute zusammengesetzt. Auf der zweiten Stufe besteht jedes seiner Attribute seinerseits aus genau einer zulässigen Attributausprägung. Allgemein gilt hierfür[4]:

$$\begin{array}{ll} \text{obj:} & \text{SYMBOL } x_1 \ldots x_Q \text{ SYMBOL} \rightarrow \text{SYMBOL} \\ & (att_1(at_1),\ldots,att_Q(at_Q)) \rightarrow obj(att_1(at_1),\ldots,att_Q(at_Q)) \end{array}$$

mit:

$$\begin{array}{ll} att_1: & DB_1 \rightarrow \text{SYMBOL} \\ & at_1 \rightarrow att_1(at_1) \\ \ldots & \\ att_Q: & DB_Q \rightarrow \text{SYMBOL} \\ & at_Q \rightarrow att_Q(at_Q) \end{array}$$

Die zuvor allgemein dargelegte Konstruktionsweise mag auf den ersten Blick kompliziert und artifiziell erscheinen. Aber sie führt bei genauerer Analyse zu einer transparenten Objektdarstellung[5]. Dabei hilft sie, unklare[6] oder aufwendigere[7] Alternativkonstruktionen auf der Basis von flach strukturierten Attributtupeln zu vermeiden. Z.B. wird die Repräsentation eines konkreten Auftrags betrachtet, der zur Objektklasse "Auftrag" gehört. Die Attributstruktur dieser Objektklasse wird durch die Funktion "auftrag"[8] so festgelegt, daß jeder Auftrag durch die Attribute mit den Namen "kunde", "produktart", "produktanzahl" und "liefertermin" und deren Ausprägungen eindeutig bestimmt wird. Der hier betrachtete konkrete Auftrag besitzt als Attributausprägungen die Konstanten "Meyer", "getriebe", "14" bzw. "22.10.89". Er wird als ein tief strukturiertes, aus Konstanten und Funktionen zusammengesetztes Objekt notiert, das ein Bild der klassenspezifischen Funktion "auftrag" ist:

auftrag(kunde("Meyer"),produktart(getriebe),produktanzahl(14),liefertermin("22.10.89"))

In dieser Darstellungsweise kann die Zusammensetzung des Objekts "auftrag", das zur Objektklasse "Auftrag" gehört, aus seinen vier Attributen und deren Ausprägungen unmittelbar abgelesen werden. Ohne die explizit angeführten Namen der jeweils involvierten Attribute hätte sich aus der flach strukturierten Objektrepräsentation auftrag("Meyer",getriebe,14,"22.10.89") z.B. hinsichtlich der Konstante "14" kaum erkennen lassen, daß es sich um die Produktanzahl handelt.

Die voranstehende Konstruktion der Attributstrukturen und Attributausprägungen von Objekten ist zwar präzise, aber immer noch recht aufwendig. Denn die Beschränkung auf eine unsortierte Logik erfordert, zwecks Darstellungsklarheit die Namen der involvierten Attribute als Operatoren $att_q(...)$ der attributspezifischen Funktionen zu explizieren. Diesen Darstellungsaufwand reduziert der Übergang zu einer sortierten Logik. Dabei wird die signaturspezifische Korrespondenz zwischen einer Sorte $sort_i$ und der Menge OB_i aller formalen Objekte, die zu dieser Sorte gehören, auf die isomorphe Differenzierung zwischen einem Attribut mit dem Namen att_q und seinen Ausprägungen at_q übertragen. Daher läßt sich jedes Attribut att_q in einer sortierten Prädikatenlogik genau so wie eine Sorte $sort_i$ konzeptualisieren. Der Definitionsbereich DB_q aller zulässigen Attributausprägungen stimmt dann mit der sortenspezifischen Objektmenge OB_i überein. Jede einzelne Attributausprägung at_q stellt ein formales Objekt aus dieser Menge dar. Die Objektklasse, die durch ihre Attributstruktur eindeutig festgelegt ist, bildet eine eigenständige, derivativ definierte Sorte "objekt". Objektklassen, Objekte, Attributstrukturen und Attributausprägungen werden dann im Rahmen einer sortierten Prädikatenlogik allgemein dargestellt als:

<u>sorts</u>: att_1
...
att_Q
objekt

<u>Ops</u>: Obj: $att_1 \ldots att_Q \rightarrow$ objekt

<u>OBs</u>: DB_1 = SYMBOL
...
DB_Q = SYMBOL
DB_{objekt} = SYMBOL

<u>ops</u>: obj: $DB_1 \times \ldots \times DB_Q \rightarrow DB_{objekt}$
$(at_1,\ldots,at_Q) \rightarrow obj(at_1,\ldots,at_Q)$

Ein einzelnes Objekt mit Q Attributen wird nunmehr durch das Tupel $obj(at_1,\ldots,at_Q)$ repräsentiert. Es weist als Bild der Operation (Funktion) "obj" nur noch eine Hierarchiestufe auf. Die Attributzugehörigkeit der Attributausprägungen ist aber weiterhin eindeutig definiert, weil jede Komponente at_q aus dem Attributtupel $(at_1,\ldots,at_Q)$ aufgrund der Signaturdefinition mit genau einer attributspezifischen Sorte att_q korrespondiert. Beispielsweise wird der konkrete Auftrag im o.a. Beispiel nur noch notiert als:

auftrag("Meyer",getriebe,14,"22.10.89")

Dabei wird zur Präzisionswahrung auf folgende Definition der Sortenstruktur für das Operationssymbol "Obj" und die daraus abgeleiteten sortierten Konstrukte zurückgegriffen:

<u>sorts</u>: kunde
produktart
produktanzahl
liefertermin
objekt

<u>Ops</u>: Auftrag: kunde produktart produktanzahl liefertermin → objekt

<u>OBs</u>: DB_{kunde} = STRING
$DB_{produktart}$ = SYMBOL
$DB_{produktanzahl}$ = INTEGER
$DB_{liefertermin}$ = STRING
DB_{objekt} = SYMBOL

<u>ops</u>: auftrag: $DB_{kunde} \times DB_{produktart} \times DB_{produktanzahl} \times DB_{liefertermin} \rightarrow DB_{objekt}$
$(at_1,...,at_4) \rightarrow auftrag(at_1,...,at_4)$

Bei konkreten Bezugnahmen auf den Auftrag im Argument einer prädikatenlogischen Formel wird nur noch das sortierte Tupel auftrag("Meyer",getriebe,14,"22.10.89") verwendet. Infolge der überlagerten Sortenstruktur ist diese Notation ebenso präzise wie die oben benutzte mit komplexer zusammengesetzten Attributtupeln, fällt aber deutlich kompakter als jene aus. In dieser Arbeit wird bei Synthetischen Netzen auf diese letzte, sortierte Darstellungsweise intensiv zurückgegriffen.

Darüber hinaus läßt sich anhand des oben vorgelegten Beispiels die Wirkung multipler Sortendefinitionen illustrieren9). Zu diesem Zweck wird die vormals eindeutig definierte Sorte "liefertermin" mit Hilfe mehrerer Subsorten als Sorte "termin" so reformuliert, daß sich der Liefertermin sowohl durch eine absolute Datumsangabe als auch durch die relative Angabe des nächsten Liefertags10) ausdrücken läßt. Dabei wird die Termindefinition gegenüber dem voranstehenden Beispiel erheblich verfeinert. Hierfür resultiert als modifizierte Definition der Sortenstruktur für das Operationssymbol "Obj" und die daraus abgeleiteten sortierten Größen:

<u>sorts</u>:	kunde	
	produktart	
	produktanzahl	
	termin	
	objekt	
	tag	
	monat	
	jahr	
	absoluter_zeitpunkt	
	wochentag	
	relativer_zeitpunkt	
<u>Ops</u>:	Auftrag:	Kunde produktart produktanzahl termin $\rightarrow$ objekt
	$TT_1, ..., TT_{31}$:	$\rightarrow$ tag
	$MM_1, ..., MM_{12}$:	$\rightarrow$ monat
	$JJ_0, ..., JJ_{99}$:	$\rightarrow$ jahr
	Datum:	tag monat jahr $\rightarrow$ absoluter_zeitpunkt
	Abs_liefertag:	absoluter_zeitpunkt $\rightarrow$ termin
	$WW_1, ..., WW_7$:	$\rightarrow$ wochentag
	Wöchentlich:	wochentag $\rightarrow$ relativer_zeitpunkt
	Rel_liefertag:	relativer_zeitpunkt $\rightarrow$ termin
<u>OBs</u>:	DB_{kunde} = STRING	
	$DB_{produktart}$ = SYMBOL	
	$DB_{produktanzahl}$ = INTEGER	
	DB_{termin} = STRING	
	DB_{objekt} = SYMBOL	
	DB_{tag} = STRING	
	DB_{monat} = STRING	
	DB_{jahr} = STRING	
	$DB_{absoluter\_zeitpunkt}$ = STRING	
	$DB_{wochentag}$ = STRING	
	$DB_{relativer\_zeitpunkt}$ = STRING	
<u>ops</u>:	auftrag:	$DB_{kunde} \times DB_{produktart} \times DB_{produktanzahl} \times DB_{termin} \rightarrow DB_{objekt}$
		$at_1,...,at_4) \rightarrow$ auftrag$(at_1,...,at_4)$
	"01", ..., "31":	$\rightarrow DB_{tag}$
	"01", ..., "12":	$\rightarrow DB_{monat}$
	"00", ..., "99":	$\rightarrow DB_{jahr}$

datum:	$DB_{tag} \times DB_{monat} \times DB_{jahr} \rightarrow DB_{absoluter\_zeitpunkt}$ (tt,mm,jj) $\rightarrow$ datum(tt,mm,jj) = "tt.mm.jj"
abs_liefertag:	$DB_{absoluter\_zeitpunkt} \rightarrow DB_{termin}$ datum(tt,mm,jj) $\rightarrow$ abs_liefertag(datum(tt,mm,jj)) = datum(tt,mm,jj)
"Montag", ..., "Sonntag":	$\rightarrow DB_{wochentag}$
wöchentlich:	$DB_{wochentag} \rightarrow DB_{relativer\_zeitpunkt}$ ww $\rightarrow$ wöchentlich(ww) = "jeden ww"
rel_liefertag:	$DB_{relativer\_zeitpunkt} \rightarrow DB_{termin}$ wöchentlich(ww) $\rightarrow$ rel_liefertag(wöchentlich(ww)) = wöchentlich(ww)

Auf der Basis dieser Sortenstruktur sind dann beispielsweise folgende sortierte Objekte als Aufträge definiert[11)]:

auftrag("Meyer",getriebe_G23,1,"jeden Dienstag")
auftrag("Schumacher",kolben_K03,18,"07.08.90")
auftrag("Gebhardt",getriebe_G15,5,abs_liefertag("01.01.90"))
auftrag("Franz",bolzen_B7,9,rel_liefertag(wöchentlich("Montag")))
auftrag(_,bolzen_B11,_,datum("01","01","90"))
auftrag(Kunde,getriebe_G23,_,wöchentlich("Montag"))
auftrag("Schulze",Produktart,Produktanzahl,Termin)

Dabei wurde besonderes Gewicht auf Variationen der Lieferterminangabe gelegt. Es wurden noch nicht einmal alle Darstellungsvarianten ausgeschöpft. Dies verdeutlicht den beachtlichen prädikatenlogischen Ausdrucksreichtum.

Anmerkungen zum Kapitel:

1) Später wird ein sortierter prädikatenlogischer Ansatz für die Markensignaturen von Synthetischen Netzen ausführlicher dargestellt.

2) Im Regelfall gilt Q≥2. Dann wird auch von Objekten mit Multiattributstrukturen gesprochen. Die nachfolgenden Konstruktionen decken aber ebenso den Grenzfall Q=1 von Objekten mit Monoattributstrukturen ab, die jeweils nur genau ein Attribut besitzen.

3) Die Objektklasse und ihre Objekte entsprechen dem Entity-Typ bzw. den Entities relationaler Datenbanksysteme. Die dort skizzierte Entsprechung zwischen prädikatenlogischer Wissensrepräsentation und relationalen Datenbanksystemen beruht letztlich auf der hier vorgestellten Konstruktionsweise von Attributen, Objekten und Objektklassen.

4) In ähnlicher Weise sind die "Tupelobjekte" (tuple objects) von BANCILHON (1986), S. 53, konzipiert.

5) Vgl. zu einer solchen transparenten Darstellungsweise konkreter formaler Objekte mit zahlreichen Attributen die Repräsentation von Aufträgen und ihren auftragsspezifischen "Daten" bei OBERWEIS (1987b), S. 21; vgl. auch die spätere systematische Konstruktion von Attributmarken.

6) Die Darstellung ist unklar, wenn nur die Attributausprägungen explizit repräsentiert werden und einem Objekt durch ein mehrstelliges Objekttupel mehrere Attribute zugeschrieben werden. Dann läßt sich - ohne weitere Informationen über die Bedeutungen der einzelnen Tupelstellen - aus der Tupelnotation nicht mehr unmittelbar ersehen, welche Tupelstelle die Ausprägung welches Attributs bedeutet; vgl. LEE,R. (1988a), S. 223. Wenn z.B. im flach strukturierten Attributtupel $(at_1,...,at_q)$ die Komponenten at_q durch konkret bestimmte Attributausprägungen ersetzt werden, läßt sich die Attributzugehörigkeit dieser Ausprägungen oftmals nicht mehr unmittelbar erkennen. Vgl. dazu auch das nachfolgende Beispiel.

7) Die Unklarheiten, die in der voranstehenden Fußnote verdeutlicht wurden, lassen sich dadurch vermeiden, daß 2-Tupel (att_q,at_q) aus dem Attributnamen und der Attributausprägung für jedes Attribut benutzt werden; vgl. LEE,R. (1985), S. 63; ZELEWSKI (1986a), S. 189, Fn. 1. Dies bereitet aber durch seine Notationsverdopplung erheblichen Darstellungsaufwand. Außerdem leitet dies bei mehreren Attributen je Objekt bereits zu tief strukturierten Tupeln über. Denn die 2-Tupel lassen sich dann jeweils als Subtupel der argumentdarstellenden Objekttupel auffassen: $((att_1,at_1),...,(att_Q,at_Q))$.

8) Die Kleinschreibung der Funktionen und der symbolischen Konstanten resultiert aus den früher festgelegten Notationskonventionen. Aus demselben Grund werden Variablen durch Großbuchstaben eingeleitet. Eine Konstante, die mit einem Großbuchstaben beginnen soll, muß als Zeichenfolge aus der Menge "STRING" definiert werden, damit sie nicht mit einer Variablen verwechselt werden kann. Diese Technik wird nachfolgend auf das Kunden-Attribut angewendet. Formale Objekte aus der Menge "STRING" werden nunmehr in Anführungszeichen eingeschlossen. Dies nimmt die spätere Notationskonvention der Implementierung auf PROLOG-Basis vorweg.

9) In dem Beispiel, das zur Explizierung multipler Sortendefinitionen für die formalen Objekte von SIG-Algebren diente, wurde auch der hier vereinfacht notierte Umgang mit Strings in einer PROLOG-Implementierung skizziert.

10) Dabei wird unterstellt, daß die Lieferungen im wöchentlichen Rhythmus erfolgen.

11) Die ersten vier Auftragsobjekte können Argumente von Fakten darstellen, weil sie nur aus Konstanten und Funktoren bestehen. Beispielsweise lassen sich in einem Formelsystem das einstellige Prädikatssymbol Lieferung(st) und die zugehörige atomare Formel lieferung(te) definieren. Falls die Faktenmenge des Prädikatssymbols Lieferung(st) unter einer Interpretation die konstante Formel lieferung(auftrag("Meyer",getriebe_G23,1, "jeden Dienstag")) enthält, so gilt: Der Kunde "Meyer" ist jeden Dienstag mit einem Produkt der Art "getriebe_G23" zu beliefern. Das fünfte Objekt mit seinen zwei anonymen Variablen kann mit allen Formelargumenten unifiziert werden, welche die Lieferung einer beliebigen Anzahl von Produkten der Art "bolzen_B11" - notiert als Symbol, nicht als Zeichenfolge - am 01.01.1990 bedeuten. Das sechste Objekt läßt sich benutzen, um in einem interpretierten prädikatenlogischen Formelsystem nach denjenigen Fakten zu suchen, welche die Lieferung jeweils einer beliebigen Anzahl der Produktart "getriebe_G23" an beliebige Kunden am jeweils nächstliegenden Montag ausdrücken. Die jeweils zu liefernde Getriebeanzahl kann allerdings wegen der anonymen Variable nicht konkret erschlossen werden. Aber die zu beliefernden Kunden lassen sich als alle Konstanten der Sorte "kunde" identifizieren, die während der Faktensuche im Formelsystem zu erfolgreichen Bindungen der Variable "Kunde" geführt haben. Das letzte Objekt gestattet es, aus den Fakten des Formelsystems alle Lieferungen zu erschließen, die für den Kunden "Schulze" bestimmt sind. Dabei kann die Variable "Termin" als Angabe des Liefertermins jeweils durch einen relativen oder auch absoluten Zeitpunkt bestimmt sein. Z.B. könnte das Formelsystem u.a. die Fakten lieferung("Schulze", getriebe_G23,1,"jeden Dienstag") und lieferung("Schulze",kolben_K03,18,"21.03.89") enthalten. Das erste Fakt liefert einen relativ, das zweite einen absolut bestimmten Liefertermin. Dies verdeutlicht, daß für dieselbe Auftragskomponente der Sorte "termin" durch multiple Sortendefinition Fakten mit verschieden strukturierten Argumenten formuliert werden können.

4.2.3.3 Multimengen

Das Konzept der prädikatenlogischen Fakten hat sich als ebenso kompaktes wie leistungsfähiges formales Instrument erwiesen, um Wissen über Formelgültigkeiten und deren Veränderungen im Zeitablauf darzustellen. Allerdings unterliegen Fakten noch einer Einschränkung, die jeder zweiwertigen Logik immanent ist, aber dem arithmetischen Charakter der Marken von Petrinetzen zuwiderläuft. Die Diskrepanz zwischen logischem Fakten- und arithmetischem Markenkonzept liegt darin begründet, daß alle Fakten wechselseitig wohlunterschieden sind[1)], während Marken in beliebig vielen identischen Kopien existieren können[2)]. Da die identischen Kopien derselben Marken qualitativ nicht voneinander unterschieden, wohl aber gezählt werden können, läßt sich von einer Quantifizierungslücke des prädikatenlogischen Faktenkonzepts sprechen[3)].

Die Diskrepanz zwischen dem Faktenkonzept einerseits, das zunächst rein qualitativ definiert ist, und dem Markenkonzept andererseits, das auch den quantitativen Aspekt identischer Markenkopien einschließt[4)], besitzt erhebliche konzeptionelle Bedeutung. Denn in allen Petrinetzen vom Typ der Prädikat/Transition-Netze - und somit auch in Synthetischen Netzen - entsprechen Netzmarkierungen jeweils einer Faktenmenge. Infolge dieser Korrespondenz muß die Quantifizierungslücke zwischen Fakten- und Markenkonzept geschlossen werden. Dies läßt sich grundsätzlich auf zwei Weisen vorstellen. Erstens läßt sich der quantitative Aspekt des Markenkonzepts dadurch eliminieren, daß sich auf derselben Stelle eines Netzes niemals mehr als eine Kopie derselben Marke unter derselben Markierung befinden darf. Dieser Ansatz liegt den Prädikat/Ereignis-Netzen zugrunde. Hierdurch wird jedoch die Ausdrucksmächtigkeit des Markenkonzepts um seinen quantitativen Aspekt *reduziert*. Dies widerspricht dem Postulat, Modellierungskonzepte mit einer möglichst großen Ausdrucksmächtigkeit auszustatten. Daher wählt der Verf. die zweite Alternative, die Quantifizierungslücke durch *Erweiterung* des Faktenkonzepts um eine quantitative Komponente zu schließen. Dies geschieht mit der Hilfe von Multimengen.

Bei der prädikatenlogischen Erweiterung von Stelle/Transition-Netzen zu Synthetischen Netzen repräsentiert - grob gesprochen - jede Stelle s_m ein Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ und jede Marke auf einer Stelle s_m ein konstantes Argument $(ob_1,...,ob_{Ku})$, für welches die konstante atomare Formel $prä_u(ob_1,...,ob_{Ku})$ gültig ist. Die aktuelle Markierung $M_r(s_m)$ der Stelle s_m im Netz entspricht daher der Menge aller gültigen atomaren Formelvorkommnisse des Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ unter derjenigen Interpretation I_r, die mit der Netzmarkierung M_r korrespondiert. Da gültige atomare konstante Formeln $prä_u(ob_1,...,ob_{Ku})$ unter einer Interpretation I_r Fakten $fakt_r(prä_u(ob_1,...,ob_{Ku}))$[5)] darstellen, drückt die Markierung $M_r(s_m)$ der Stelle s_m die Faktenmenge $FAK_{u.r}$ des Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ aus. Folglich würden Mehrfachmarkierungen derselben Stelle mit mehreren identischen Kopien derselben Marke bedeuten, daß dieselbe konstante atomare Formel $prä_u(ob_1,...,ob_{Ku})$ unter derselben Interpretation I_r durch *mehrere unterschiedslose* Argumente $(ob_1,...,ob_{Ku})$ erfüllt wird. Dieser Sachverhalt läßt sich in synonymer Weise auch dadurch ausdrücken, daß die gültige Formel $prä_u(ob_1,...,ob_{Ku})$ und das Fakt $fakt_r(prä_u(ob_1,...,ob_{Ku}))$ unter der Interpretation I_r multipel definiert sind. Daher erfolgt eine entsprechende "Vervielfachung der Prädikatsgültigkeit"[6)].

Die potentielle Multiplizität von gültigen Formeln und Fakten kann zwar im bislang behandelten prädikatenlogischen Rahmen nicht gerechtfertigt werden[7)]. Doch wird dies im Rahmen einer denotationalen Semantik möglich, die formale Objekte durch reale Referenzobjekte interpretiert. Dann kann es "sinnvoll"[8)] sein, demselben formalen Objekt mehrere reale Referenzobjekte zuzuordnen. In einem solchen Fall unterscheiden sich die Referenzobjekte zwar real, doch entspricht ihnen in der jeweils betrachteten Formalsprache nur genau ein formales Objekt[9)]. Es kann dennoch von Interesse sein, in dieser Formalsprache Ausdrücke zu formulieren, die sich *explizit* auf *mehrere* reale Referenzen dieses einen formalen Objekts erstrecken. Dies erfordert eine formalsprachliche Syntax, die Ausdrücke zuläßt, in denen dasselbe formale Objekt *mehr-*

fach vorkommen darf. Die multiplen, untereinander identischen Vorkommnisse desselben formalen Objekts können gezählt, aber nicht voneinander unterschieden werden. Jedes Vorkommnis eines solchen formalen Objekts wird als dessen Kopie bezeichnet[10]. Genau diese abstrakte formale Eigenschaft, in mehreren identischen Kopien vorliegen zu dürfen, erfüllen die Marken des Petrinetz-Konzepts[11]. Daher können Netzmarken grundsätzlich herangezogen werden, um die oben skizzierte multiple Erfüllung desselben Prädikats durch identische Argumente $(ob_1,...,ob_K)$ als Mehrfachmarkierung der prädikatsrepräsentierenden Stelle ausdrücken zu können.

Allerdings stellt sich die Frage nach dem "Sinn" multipler Prädikatsgültigkeiten, die aus der Kombination einer denotationalen Semantik mit Mehrfachreferenzen für formale Objekte resultieren. Eine mögliche, im Rahmen dieser Arbeit einzig relevante Antwort liegt in der modelltheoretischen Sinnkonstitution. Es sollen prädikatenlogisch formulierte Modell(baustein)e für reale Objekte aus dem Realitätsbereich Flexibler Fertigungssysteme konstruiert werden. Eine identische "Weltverdopplung" würde dem modelltheoretischen Kerngedanken widersprechen, *vereinfachte* - aber dennoch leistungsfähige - Abbilder realer Objekte zu konstruieren. Aus diesem Vereinfachungsansatz folgt zwingend, daß in einem prädikatenlogischen, somit formalsprachlichen Objektmodell[12] ein formales Objekt jeweils eine Klasse ähnlicher, aber real verschiedener Objekte repräsentiert, von deren realen Unterschieden im formalen Objektmodell abstrahiert wird. Solange eine solche Klasse nicht zur leeren Klasse oder zur unären Klasse degeneriert ist, besitzt das klassenrepräsentierende formale Objekt eine reale Mehrfachreferenz. Dies ist in der Modellierungspraxis der Normalfall.

Beispielsweise kann in einem Modell, das die Werkzeugversorgung eines Flexiblen Fertigungssystems abbildet, das eine formale Objekt "Bohrkopf_<Spezifikation>" auf eine Vielzahl von realen Bohrköpfen verweisen, welche alle dieselbe Bohrkopfspezifikation erfüllen. Ein Prädikat, das die Verfügbarkeit eines Bohrkopfes mit dieser Spezifikation ausdrückt (Verfügbarkeitsprädikat), ist zwar für das formale Objekt "Bohrkopf_<Spezifikation>" gültig. Aber aus dieser Prädikatsgültigkeit, die den Charakter einer qualitativen Information besitzt, kann nicht erschlossen werden, *wie viele* dieser Objekte real zur Verfügung stehen. Eine Möglichkeit[13], diese zusätzliche quantitative Information auch im prädikatenlogischen Modell zu erfassen, besteht im Rückgriff auf die oben eingeführten identischen Kopien formaler Objekte. Dann wird jeder Bohrkopf, der im modellierten Werkzeugsystem mit der erforderlichen Spezifikation real vorhanden ist, im Modell des Werkzeugsystems durch genau eine Kopie des einen formalen Objekts "Bohrkopf_<Spezifikation>" abgebildet. Bei entsprechender Anpassung des Faktenbegriffs[14] läßt sich in einer prädikatenlogischen Modellformulierung aus der aktuellen Faktenmenge des Verfügbarkeitsprädikats ablesen, wie viele Kopien des formalen Objekts "Bohrkopf_<Spezifikaion>" dieses Prädikat als dessen Argument jeweils erfüllen. Damit steht auch im prädikatenlogisch formulierten Objektmodell die quantitative Information über Werkzeuganzahlen bereit, ohne den modelltheoretischen Kerngedanken aufzugeben, die spezifikationsgleichen, aber real verschiedenen Werkzeuge im Modell auf genau ein formales Objekt abzubilden.

Das voranstehende Beispiel verdeutlicht die erhebliche Bedeutung, die dem - zunächst artifiziell und unverständlich anmutenden - Konzept multipler Prädikatsgültigkeiten aus modelltheoretischer Sicht grundsätzlich zukommt. Einerseits stellt die Anknüpfung an die Prädikatenlogik ein Ausdruckspotential für die formalsprachliche Formulierung von *qualitativen* Sachverhalten bereit, das von keinem anderen formalsprachlichen und betriebswirtschaftlich etablierten Modellierungskonzept erreicht - geschweige denn übertroffen wird. Andererseits bindet das Konzept, multiple Erfüllungen desselben Prädikats durch mehrere identische Kopien desselben formalen Objekts zuzulassen, die Repräsentation *quantitativer* Aspekte in die ansonsten qualitativ ausgerichtete Prädikatenlogik ein.

Darüber hinaus entspricht die formalsprachliche Zulässigkeit mehrerer Kopien desselben formalen Objekts genau jener Mehrfachreferenz dieses formalen Objekts, die im Rahmen einer denotationalen Semantik durch die Zuordnung mehrerer realer Objekte erfolgt. In dieser Entsprechung liegt ein wesentlicher Fortschritt von formalsprachlichen Modellierungskonzepten[15]. Denn eine denotationale Semantik stellt stets eine Bezugnahme auf *außersprachliche*[16] Referenzobjekte dar, die *innerhalb* eines formalsprachlichen Konzepts nicht mehr erklärt werden kann. Daher werden denotationale Semantiken für formalsprachliche Konzepte im allgemeinen als natürlichsprachliche Erläuterungen von Konzeptkomponenten gestaltet. Auf diese natürlichsprachlichen Konzeptergänzungen können per constructionem diejenigen Instrumente, die innerhalb eines Konzepts als formalsprachlich definierte Operationen auf formalen Objekten Geltung besitzen, nicht mehr angewendet werden. Die Zulässigkeit mehrerer Kopien desselben formalen Objekts *innerhalb* eines formalsprachlichen Konzepts macht es überflüssig, auf denotationale Semantiken zurückgreifen zu müssen, um Mehrfachreferenzen ausdrücken zu können. Damit verschwindet auch das Problem, daß sich formalsprachlich definierte Instrumente bei denotational formulierten Mehrfachreferenzen nicht mehr anwenden lassen. Statt dessen lassen sich nunmehr Bezüge auf mehrere reale Objekte formalsprachlich explizieren, obwohl diese realen Objekte unter Abstraktion von ihren irrelevanten Unterschieden durch dasselbe formale Objekt modelliert worden sind. Diese explizite formalsprachliche Mehrfachreferenz leisten die mehrfach vorhandenen Kopien des einen formalen Objekts.

Aufgrund der voranstehenden Erläuterungen wird die Konstruktion formaler Modellierungskonzepte wesentlich bereichert, wenn es möglich ist:

- ❑ demselben formalen Objekt innerhalb des Konzeptformalismus eine beliebige Anzahl identischer Kopien explizit zuzuordnen (allgemeine modelltheoretische Perspektive);
- ❑ die Erfüllung desselben Prädikats durch mehrere identische Argumente als multiple Prädikatsgültigkeit in der Faktenmenge des Prädikats explizit auszudrücken (spezielle prädikatenlogische Perspektive).

Beide Perspektiven werden im Rahmen des Petrinetz-Konzepts durch Marken erfüllt, die einerseits als Kopien desselben formalen Objekts und andererseits als Argumente aus der aktuellen Extension eines Prädikats aufgefaßt werden. Hierin sieht der Verf. einen zentralen Ansatzpunkt seiner Ausführungen zum Petrinetz-Konzept. Aus modelltheoretischem Blickwinkel stellen Marken *das ausschlaggebende Charakteristikum* des Petrinetz-Konzepts dar, das es von anderen formalen Modellierungskonzepten unterscheidet. In prädikatenlogischer Hinsicht läßt sich mit Marken eine Erweiterung des Faktenbegriffs einführen, die das - ohnehin schon beträchtliche - Ausdruckspotential der Prädikatenlogik noch weiter ausdehnt.

Bisher wurde nur gerechtfertigt, warum auf der objektsprachlichen Ebene identische Kopien jeweils eines formalen Objekts zugelassen und auf der metasprachlichen Ebene multiple Prädikatsgültigkeiten eingeführt werden sollen. Nunmehr gilt es, den formalen Apparat bereitzustellen, um diese Aspekte in das Konzept Synthetischer Netze aufnehmen zu können. Ansatzpunkt hierfür ist der konventionelle Mengenbegriff. Mit seiner Hilfe wurde die Faktenmenge $FAK_{u.r}$ eines Prädikatssymbols $Prä_u$ als die Menge aller konstanten atomaren Prädikatsvorkommnisse $prä_u(ob_1,...,ob_{Ku})$ definiert, die unter einer Interpretation I_r gültig sind. Sobald es gelingt, innerhalb derselben prädikatspezifischen Faktenmenge mehrere identische Kopien desselben Prädikatsvorkommnisses $prä_u(ob_1,...,ob_{Ku})$ explizit formal darstellen und benutzen zu können, liegt zugleich eine multiple Prädikatsgültigkeit vor. Denn dann sind mehrere identische Prädikatsvorkommnisse $prä_u(ob_1,...,ob_K)$ aktuell gültig. Eine solche Darstellung ist auf konventioneller mengentheoretischer Basis nicht möglich, weil sich alle Elemente $prä_u(ob_1,...,ob_K)$ aus einer Fakten*menge* jeweils paarweise unterscheiden müssen. Diese Einschränkung wird nachfolgend durch das Konzept der *Multimengen* überwunden. Multimengen sind formale Konstrukte, die mengentheoretische und arithmetische Aspekte integrieren. Sie dienen der späteren Reformulierung der Faktenmenge von prädikatenlogischen Formeln.

Definition: Multimengen

Eine Multimenge[17] $mult_D$ ist eine rechtseindeutige Abbildung einer beliebigen Menge D auf die Menge der natürlichen Zahlen $\mathcal{N}_0$:

$$\begin{aligned} mult_D\colon\ & D \rightarrow \mathcal{N}_0 \\ & d \rightarrow mult_D(d) = mu_d \end{aligned}$$

<u>Erläuterungen und Ergänzungen zur Multimengen-Definition:</u>

a) Die rechtseindeutige Abbildung, welche die Multimenge $mult_D$ definiert, heißt Multimengenfunktion[18]. Ihre Funktionswerte mu_d werden als Multiplizitäten der jeweils abgebildeten Elemente d bezeichnet.

b) Der Vorbereich D der Multimengenfunktion heißt Trägermenge oder Domäne der Multimenge. Die Trägermenge wird hier stets als nicht-leere und endliche[19] Menge vorausgesetzt.

c) Für dieselbe Trägermenge D können beliebig viele verschiedene Multimengen $mult_D$ definiert werden, die sich jeweils durch ihre Abbildungsvorschriften unterscheiden. Die Menge aller Multimengen $mult_D$, die über der Trägermenge D grundsätzlich definiert werden könnten, wird mit $MULT_D$ bezeichnet. Diese Menge ist immer nicht-leer, sie kann unendlich groß sein.

d) Die leere Multimenge "$\varnothing_D$" ist für jede Trägermenge D dadurch definiert, daß jedem ihrer Elemente der Funktionswert Null zugeordnet wird:

$$\begin{aligned} \varnothing_D\colon\ & D \rightarrow \{0\} \\ & d \rightarrow \varnothing_D(d) = 0 \end{aligned}$$

Falls die Trägermenge "D" und der Bezug auf eine Multimenge aus dem Kontext ersichtlich sind, wird als vereinfachte Notation der leeren Multimenge auch das indexfreie Symbol "$\varnothing$" zugelassen.

e) Die originäre Definition einer Multimenge als Multimengenfunktion $mult_D$ läßt sich in äquivalenter Weise als eine konventionelle Menge $Mult_D$ reformulieren[20]. Diese Menge umfaßt alle geordneten 2-Tupel (mu_d,d), welche die Abbildungsvorschrift der Multimengenfunktion $mult_D$ erfüllen:

$$Mult_D = \{(mu_d,d)\colon d \in D \wedge mu_d = mult_D(d)\} \subseteq (\mathcal{N}_0 \times D)$$

In diesem Fall wird von einer binären Mengennotation gesprochen. Jedes Element (mu_d,d) aus der konventionell definierten Menge $Mult_D$ wird auch als ein Multielement bezeichnet, weil es mu_d Vorkommnisse desselben Elements "d" repräsentiert. Die Komponente mu_d heißt die Multiplizität des Elements "d". Falls die leere Multimenge $Mult_D = \varnothing$ vorliegt, enthält die Multimenge in binärer Mengennotation ausschließlich 2-Tupel (mu_d,d) mit $mu_d=0$ für alle $d \in D$. Für diesen degenerierten Fall gilt: $Mult_D = \{(0,d)\colon d \in D\} = \varnothing_D$. Als Kurznotation wird hierfür ebenso $Mult_D = \varnothing$ zugelassen.

f) Von einer binären Mengennotation mit positiven Multiplizitäten wird gesprochen, falls nur die Paare (mu_d,d) mit $mu_d \in \mathcal{N}_+$ in der Menge $Mult_D$ explizit angeführt werden. Die 2-Tupel aus der binären Mengennotation einer Multimenge werden dann in ihrer zweiten Komponente im allgemeinen nicht mehr alle Elemente "d" aus der Trägermenge D enthalten. Es entfallen in der expliziten Mengenschreibweise diejenigen Elemente "d", deren Multiplizitäten $mu_d=0$ erfüllen. Für jedes Element "d" aus der Trägermenge D, für das kein Paar (mu_d,d) explizit aufgelistet wird, gilt aber als implizit vereinbarte Multiplizität weiterhin $mu_d=0$. Daher bleibt die o.a. Definition von Multimengen in ihrer binären Mengennotation grundsätzlich unverändert, obwohl nur positive Multiplizitäten expliziert werden. Es wird nur eine kompaktere Darstellungsweise eingeführt, nicht aber die Multimengendefinition selbst variiert. Daher bleibt auch bei Notation mit positiven Multiplizitäten als degenerierter Fall die leere Multimenge $Mult_D=\emptyset$ weiterhin definiert.

g) Die Menge MUL_D *aller* Multimengen $Mult_D$, die sich über der Trägermenge D in binärer Mengennotation definieren lassen, kann aus dem kartesischen Produkt der natürlichen Zahlen mit der Trägermenge D gewonnen werden[21]. Dabei ist jedoch die Asymmetrie zwischen der ersten und der zweiten Komponente der 2-Tupel (mu_d,d) in jeder Multimenge zu beachten: Jedes Element "d" muß innerhalb derselben Multimenge in einem 2-Tupel genau einmal als zweite Komponente vorkommen. Dieselbe Multiplizität kann aber mehreren verschiedenen Elementen zukommen, so daß gelten darf: $d_1 \neq d_2$ und $mu_{d1}=mu_{d2}$. Dagegen sind mehrere 2-Tupel, die in ihrer zweiten Komponente "d" übereinstimmen, in derselben Multimenge verboten. Diese Anforderungen werden berücksichtigt durch:

$$MUL_D = \{Mult_D\colon Mult_D \subseteq (\mathcal{N}_0 \times D) \wedge (\forall(d \in D)\ \underline{\exists}(mu_d \in \mathcal{N}_0)\colon (mu_d,d) \in Mult_D)\}$$

Zwecks Notationsvereinfachung wird der Multimengenoperator "mul" eingeführt: Er erzeugt aus einer beliebigen Trägermenge D die Menge MUL_D aller Multimengen, die über der Trägermenge D in binärer Mengennotation definiert werden können.

$$\text{mul:}\quad D \rightarrow mul(D) = MUL_D$$

mit:

$$mul(D) = \{Mult_D\colon Mult_D \subseteq (\mathcal{N}_0 \times D) \wedge (\forall(d \in D)\ \underline{\exists}(mu_d \in \mathcal{N}_0)\colon (mu_d,d) \in Mult_D)\}$$

In beiden voranstehenden Fällen wird unterstellt, daß in jeder Multimenge $Mult_D$ nur die Paare (mu_d,d) mit positiven Multiplizitäten mu_d explizit angeführt werden.

h) Aus der binären Mengennotation leiten sich zwei weitere äquivalente Schreibweisen für Multimengen ab[22]. Diese Notationen können nicht mehr als konventionelle Mengen aufgefaßt werden.

i) Die Listennotation[23] einer Multimenge $mult_D$ ist jedes ungeordnete Tupel $mult_D=[...]$[24], das folgende Eigenschaften erfüllt:

- Der leeren Multimenge $mult_D=\emptyset_D$ ist die leere Liste $mult_D=[]$ zugeordnet.
- Jedes Paar $(mu_d,d) \in Mult_D$ wird in der Liste $mult_D=[...]$ genau dann explizit berücksichtigt, wenn $mu_d \in \mathcal{N}_+$ gilt. Paare $(mu_d,d) \in Mult_D$ mit $mu_d=0$ werden dagegen nicht aufgelistet.
- Das Paar $(mu_d,d) \in Mult_D$ mit $mu_d \in \mathcal{N}_+$ kann entweder dadurch explizit berücksichtigt werden, daß die Komponente "d" in der Liste $mult_D=[...]$ genau mu_d mal vorkommt (aufzählende Listennotation). Oder das Tupel enthält das Konstrukt "$mu_d \cdot d$" genau einmal (multiplikative Listennotation).

- Bei multiplikativer Listennotation kann das Konstrukt "$mu_d \cdot d$" zu "$mu_d d$" vereinfacht werden, falls "d" ein beliebiges, aber nicht-numerisches Symbol darstellt. Ebenso läßt sich das Konstrukt "$mu_d \cdot d$" zum Ausdruck "d" vereinfachen, wenn $mu_d = 1$ gilt.

j) Jede Multimenge $mult_D$, die als aufzählende Liste notiert ist, stellt ein K-stelliges Wort aus dem freien Monoid D* über der Trägermenge D mit $K \in \mathcal{N}_0$ dar. Daher gilt für die Menge $MULT_D$ aller listennotierten Multimengen $mult_D$, die über der Trägermenge zulässig sind: $MULT_D = D^*$. Da alle Notationsweisen für Multimengen äquivalent sind, wird diese Definition aller Multimengen mit der Trägermenge D unabhängig von der jeweils gewählten Notationsart verwendet.

k) Die Darstellungsweise der Multimenge $mult_D$ ist in ihrer Listennotation mehrdeutig. Erstens lassen sich die Tupelkomponenten in beliebigen Reihenfolgen anordnen. Es herrscht Mehrdeutigkeit aufgrund von Permutationen. Zweitens kann dasselbe Element (mu_d, d) aus der Menge $Mult_D$ entweder als mu_d-faches Vorkommnis der Komponente "d" im Tupel $mult_D$ ausgedrückt werden oder aber als einfaches Vorkommnis des Konstrukts "$mu_d \cdot d$". Dies konstituiert eine Mehrdeutigkeit infolge Multiplizität. Schließlich existiert Mehrdeutigkeit aufgrund unterschiedlicher Ausschöpfung der o.a. Vereinfachungsoptionen. Diese Mehrdeutigkeiten wirken sich aber auf die Operationen und Relationen nicht aus, die später für Multimengen eingeführt werden. Die Mehrdeutigkeiten erweisen sich daher als irrelevant.

l) Die formale Summennotation[25)] einer Multimenge stützt sich auf das Konzept "symbolischer" oder "formaler Summen"[26)]. Die Darstellung einer Multimenge $mult_D$ als formale Summe erfüllt folgende Eigenschaften:

- Der leeren Multimenge $mult_D = \emptyset_D$ ist die Nullsumme $mult_D = 0$ zugeordnet.
- Jedes Paar $(mu_d, d) \in Mult_D$ wird in der formalen Summe $mult_D$ genau dann explizit berücksichtigt, wenn $mu_d \in \mathcal{N}_+$ gilt. Paare $(mu_d, d) \in Mult_D$ mit $mu_d = 0$ werden dagegen nicht dargestellt.
- Das Paar $(mu_d, d) \in Mult_D$ mit $mu_d \in \mathcal{N}_+$ wird als formales Produkt "$mu_d \cdot d$" repräsentiert.
- Wenn die Menge $Mult_D$ mindestens zwei Paare (mu_{d1}, d_1) und (mu_{d2}, d_2) mit $mu_{d1}, mu_{d2} \in \mathcal{N}_+$ umfaßt, dann werden die paarrepräsentierenden formalen Produkte "$mu_{d1} \cdot d_1$" bzw. "$mu_{d2} \cdot d_2$" jeweils zu formalen Summen "$mu_{d1} \cdot d_1 + mu_{d2} \cdot d_2$" verknüpft.
- Das formale Produkt "$mu_d \cdot d$" kann zu "$mu_d d$" vereinfacht werden, falls "d" ein beliebiges, aber nicht-numerisches Symbol darstellt. Ebenso läßt sich das formale Produkt "$mu_d \cdot d$" zum Ausdruck "d" vereinfachen, wenn $mu_d = 1$ gilt.

Für jede formale Summe aus einer Trägermenge D ist auch die allgemeine Sigmanotation "$\Sigma(d \in D)$: $mu_d \cdot d$" zulässig.

m) Formale Summen besitzen gegenüber der Listenschreibweise einerseits den Vorteil, die notationelle Mehrdeutigkeit bei Multiplizität zu vermeiden[27)]. Andererseits leiden sie unter dem Nachteil, daß sich Listen leichter mit Hilfe der Automatischen Informationsverarbeitung implementieren lassen. Die Benutzung von binären Mengen hat gegenüber den beiden vorgenannten Notationsweisen der Vorzug, mit dem konventionellen Mengenbegriff auszukommen. Allerdings fällt ihre Formulierung auch relativ aufwendig aus. Da alle drei Darstellungsformen von Multimengen äquivalent sind, wird in dieser Arbeit jeweils auf diejenige Variante zurückgegriffen, die im aktuellen Anwendungskontext am günstigsten erscheint. Dies bedeutet, daß beispielsweise in arithmetischen Kontexten die Notation formaler Summen bevorzugt wird.

n) Zwei Multimengen $mult_{D1}$ und $mult_{D2}$ heißen disjunkt, wenn ihre Trägermengen D_1 und D_2 disjunkt sind.

o) Für disjunkte Multimengen ist die Vereinigungsmenge als eine Multimenge definiert, für die gilt:

$$\text{mult}_{D1} \cup \text{mult}_{D2}\text{: } (D_1 \cup D_2) \rightarrow \mathcal{N}_0$$
$$d \rightarrow \text{mult}_{D1} \cup \text{mult}_{D2}(d) = \text{mu}_d$$

mit:

$$\text{mult}_{D1} \cup \text{mult}_{D2}(d) = \begin{cases} \mu\text{ult}_{D1}(d); \text{ sofern } d \in D_1 \\ \mu\text{ult}_{D2}(d); \text{ sofern } d \in D_2 \end{cases}$$

oder in äquivalenter Darstellung:

$$\text{mult}_{D1} \cup \mu\text{ult}_{D2}$$
$$:\Leftrightarrow \quad \text{Mult}_{D1} \cup \text{Mult}_{D2} = \{(\text{mu}_d, d)\text{: } d \in (D_1 \cup D_2) \wedge \ldots$$
$$(d \in D_1 \rightarrow \text{mu}_d = \text{mult}_{D1}(d)) \wedge (d \in D_2 \rightarrow \text{mu}_d = \text{mult}_{D2}(d))\}$$

p) Eine Multimenge mult_{D1} ist eine Teilmenge einer anderen Multimenge mult_{D2}, wenn die Trägermenge D_1 der ersten Multimenge eine Teilmenge der Trägermenge D_2 der zweiten Multimenge ist und wenn jedes Element aus der ersten Trägermenge in der ersten Multimenge höchstens so oft enthalten ist, wie es in der zweiten Multimenge vorkommt:

$$\text{mult}_{D1} \subseteq \text{mult}_{D2}$$
$$:\Leftrightarrow \quad D_1 \subseteq D_2 \wedge (\forall(d \in D_1)\text{: } \text{mu}_{d.1} \leq \text{mu}_{d.2})$$

q) Für die Verwendung von Multimengen werden formale Operationen und Relationen definiert[28]. Sie knüpfen am natürlichzahligen Charakter des Nachbereichs der Multimengenfunktion mult_D an. Hierdurch wird es möglich, den formalen Umgang mit Multimengen auf einfache arithmetische Operationen und Relationen zurückzuführen[29]. Dabei wird vorausgesetzt, daß sich die involvierten Multimengen jeweils auf dieselbe Trägermenge D erstrecken; andernfalls sind die Operationen und Relationen nicht definiert. Dann gilt für beliebige Multimengen $\text{mult}_{D.q} = \Sigma(d \in D)\text{: } \text{mu}_{d.q} \bullet d$ mit $q \in \{0,1,2\}$ und für jeden Skalar $c \in \mathcal{N}_+$ in der Notationsweise formaler Summen:

- für jeden Relator "≡" aus der Menge $\{\leq,=,\geq\}$:

  $\text{mult}_{D.1} \equiv \text{mult}_{D.2} \;:\Leftrightarrow\; \forall(d \in D)\text{: } \text{mu}_{d.1} \equiv \text{mu}_{d.2}$

- für den Relator "≠":

  $\text{mult}_{D.1} \neq \text{mult}_{D.2} \;:\Leftrightarrow\; \exists(d \in D)\text{: } \text{mu}_{d.1} \neq \text{mu}_{d.2}$

- für die Addition:

  $$\text{mult}_{D.1} + \text{mult}_{D.2}\text{: } D \rightarrow \mathcal{N}_0$$
  $$d \rightarrow \text{mult}_{D.1} + \text{mult}_{D.2}(d) = \Sigma(d \in D)\text{: } (\text{mu}_{d.1} + \text{mu}_{d.2}) \bullet d$$

- für die Subtraktion[30]:

  $$\text{mult}_{D.1} - \text{mult}_{D.2}\text{: } D \rightarrow \mathcal{N}_0$$
  $$d \rightarrow \text{mult}_{D.1} - \text{mult}_{D.2}(d) = \Sigma(d \in D)\text{: } (\text{mu}_{d.1} - \text{mu}_{d.2}) \bullet d;$$
  falls $\text{mult}_{D.1} \geq \text{mult}_{D.2}$

- für die Multiplikation:

$$\text{mult}_{D.1}\bullet\text{mult}_{D.2}\colon\ D \rightarrow \mathcal{N}_0$$
$$d \rightarrow \text{mult}_{D.1}\bullet\text{mult}_{D.2}(d) = \Sigma(d\in D)\colon (\text{mu}_{d.1}\bullet\text{mu}_{d.2})\bullet d$$

$$c\bullet\text{mult}_{D.0}\colon\ D \rightarrow \mathcal{N}_0$$
$$d \rightarrow c\bullet\text{mult}_{D.0}(d) = \Sigma(d\in D)\colon (c\bullet\text{mu}_{d.0})\bullet d$$

Die Additions- und Multiplikationsoperation für jeweils zwei Multimengen lassen sich auf beliebig, aber höchstens endlich viele Multimengen erweitern[31]. Dann können der kompakte Summen- und Produktoperator "Σ" bzw. "Π" auf Multimengen genau so angewendet werden wie auf die Addition bzw. Multiplikation reeller Zahlen.

r) Für eine endliche Familie aus Multimengen $\text{Mult}_{D.w}$ in binärer Mengennotation, die mit $w\in\{1,\ldots,W\}$ und $W\in\mathcal{N}_+$ über derselben Trägermenge D definiert sind, bestimmt der Maximierungsoperator "max" eine neue Multimenge $\text{Mult}_{D.max}$, die für jedes Element $d\in D$ die jeweils höchste Multiplizität aus den zugrundeliegenden Multimengen $\text{Mult}_{D.w}$ ausweist:

$$\begin{aligned}\text{Mult}_{D.max} &= \max(\text{Mult}_{D.w}\colon w=1,\ldots,W)\\ &= \{(\text{mu}_{d.max},d)\colon d\in D \wedge \ldots\\ &\quad (\exists\,(w^*\in\{1,\ldots,W\})\colon \text{mu}_{d.max}=\text{mu}_{d.w^*} \wedge \forall(w\in\{1,\ldots,W\})\colon \text{mu}_{d.w^*}\geq\text{mu}_{d.w})\}\end{aligned}$$

Der Maximierungsoperator läßt sich als ein Operator für die Bildung der Vereinigungsmenge einer Familie von Multimengen auffassen, die alle dieselbe Trägermenge D besitzen.

s) Für jede Multimenge mult_D gibt die Zählfunktion #[32] an, wie viele Vorkommnisse von Elementen aus ihrer Trägermenge D die Multimenge insgesamt enthält:

$$\#\colon\quad \text{MULT}_D \rightarrow \mathcal{N}_0$$
$$\text{mult}_D \rightarrow \#(\text{mult}_D) = \Sigma(d\in D)\colon \text{mu}_d$$

<u>Beispiele zur Multimengen-Definition:</u>

Für die Trägermenge $D=\{d_1,d_2,d_3,d_4,d_5\}$ wird die Multimenge mult_D originär definiert durch die Multimengenfunktion:

$$\begin{aligned}\text{mult}_D\colon\ D &\rightarrow \mathcal{N}_0\\ d_1 &\rightarrow \text{mult}_D(d_1) = 2\\ d_2 &\rightarrow \text{mult}_D(d_2) = 0\\ d_3 &\rightarrow \text{mult}_D(d_3) = 1\\ d_4 &\rightarrow \text{mult}_D(d_4) = 3\\ d_5 &\rightarrow \text{mult}_D(d_5) = 0\end{aligned}$$

Hierfür sind folgende, bei weitem nicht alle Notationsvarianten ausschöpfenden[33] äquivalenten Notationen möglich:

- binäre Menge: $\text{Mult}_D = \{(2,d_1),(1,d_3),(3,d_4)\}$;
- aufzählende Liste: $\text{mult}_D = [d_1,d_1,d_3,d_4,d_4,d_4]$;
- permutierte aufzählende Liste: $\text{mult}_D = [d_4,d_1,d_3,d_4,d_1,d_4]$;

- multiplikative Liste: $mult_D = [2d_1,d_3,3d_4]$;
- formale Summe (unvereinfacht): $mult_D = 2 \bullet d_1 + 0 \bullet d_2 + 1 \bullet d_3 + 3 \bullet d_4 + 0 \bullet d_5$;
- formale Summe (maximal vereinfacht): $mult_D = 2d_1 + d_3 + 3d_4$;
- Elementeanzahl: $\#(mult_D) = 2+0+1+3+0 = 6$.

Ursprünglich wurden Faktenmengen als konventionell definierte Mengen eingeführt. Dies trifft sowohl auf die Faktenmengen $FAK_{u.r}$ zu, die sich im Rahmen der prädikatenlogischen Semantik jeweils auf eine Interpretation I_r bezogen, als auch auf die Faktenmengen $FAK_{u.0}$, die für den ausgezeichneten Grundzustand "r=0" der algebraisch-prädikatenlogischen Spezifikation $SPEC_{PSIG}$ eines Objektmodells galten. Fortan handelt es sich bei allen Betrachtungen von Faktenansammlungen um Faktenmultimengen. Sie werden - wenn nicht ausdrücklich anders vereinbart - stets in ihrer binären Mengennotation mit positiven Multiplizitäten verwendet. Daher lassen sie sich auch vereinfacht weiterhin als Faktenmengen ansprechen[34].

Fakten wurden als konstante atomare Formeln $prä_u(ob_1,...,ob_{Ku})$ eingeführt, die jeweils aus einem Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ formiert sind und die bezüglich einer Interpretation I_r oder im Hinblick auf einen Modellzustand "r" als gültig bekannt sind. Die syntaktische Trägermenge einer Faktenmultimenge $FAK_{u.r}$ ist daher die Menge KAF_u aller verschiedenen[35] konstanten atomaren Formeln, die aus den Mengen $OB_{i(k)}$ formaler Objekte für die Argumentstellen des Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ gebildet werden können. Sie wurde schon früher als konstante Prädikatemenge für das Prädikatssymbol $Prä_u$ eingeführt. Für alle - immer noch rein syntaktisch definierten - Multimengenfunktionen $mult_{KAFu.r}$ über dieser Prädikatemenge KAF_u gilt mit $r \in \mathcal{N}_+$:

$$\begin{array}{ll} mult_{KAFu.r}: & KAF_u \rightarrow \mathcal{N}_0 \\ & prä_u(ob_1,...,ob_{Ku}) \rightarrow mult_{KAFu.r}(prä_u(ob_1,...,ob_{Ku})) = mu_{u.r} \end{array}$$

Daraus folgt in der binären Mengennotation von Multimengen:

$$\begin{array}{rl} Mult_{KAFu.r} = & \{(mu_{u.r},prä_u(ob_1,...,ob_{Ku})): \wedge\ mu_{u.r} \in \mathcal{N}_0 \wedge prä_u(ob_1,...,ob_{Ku})) \in KAF_u\ ... \\ & mu_{u.r} = mult_{KAFu.r}(prä_u(ob_1,...,ob_{Ku}))\} \subseteq (\mathcal{N}_0 \times KAF_u) \end{array}$$

Jedes Element einer solchen Multimenge ist ein geordnetes 2-Tupel $(mu_{u.r},prä_u(ob_1,...,ob_{Ku}))$. Seine zweite Komponente stellt eine konstante atomare Formel für das Prädikatssymbol $Prä_u$ dar. Seine erste Komponente gibt als Multiplizität an, wie viele identische Vorkommnisse dieser Formel existieren. Es werden nur diejenigen Elemente $(mu_{u.r},prä_u(ob_1,...,ob_{Ku}))$ explizit angeführt, die mit $mu_{u.r} \in \mathcal{N}_+$ positive Multiplizitäten besitzen[36]. Die $mu_{u.r}$ Vorkommnisse derselben konstanten atomaren Formel $prä_u(ob_1,...,ob_{Ku})$ werden auch als Kopien des Prädikats $prä_u$ bezeichnet. Je nach konkreter Ausgestaltung der zugrundeliegenden Multimengenfunktion $mult_{KAFu.r}$ resultieren andere Multiplizitäten $mu_{u.r}$ und somit auch andere Multimengen $Mult_{KAFu.r}$. Der oben eingeführte Multimengenoperator "mul" wird um den Teilindex "r" für die jeweils interessierende Faktenmenge $FAK_{u.r}$ zum indizierten Multimengenoperator "mul_r" erweitert[37]. Für die Menge $MKAF_{u.r}$ aller unterschiedlichen Multimengen $Mult_{KAFu.r}$, die sich über der Trägermenge KAF_u in binärer Mengennotation definieren lassen, gilt dann:

$$\mathrm{MKAF}_{u.r} = \mathrm{mul}_r(\mathrm{KAF}_u)$$

mit:

$$\mathrm{mul}_r(\mathrm{KAF}_u) = \{\mathrm{Mult}_{\mathrm{KAFu.r}}: \mathrm{Mult}_{\mathrm{KAFu.r}} \subseteq (\mathcal{N}_0 \times \mathrm{KAF}_u) \wedge \ldots$$
$$(\forall(\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}}) \in \mathrm{KAF}_u)\ \exists(\mathrm{mu}_{u.r} \in \mathcal{N}_0): \ldots$$
$$(\mathrm{mu}_{u.r},\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}})) \in \mathrm{Mult}_{\mathrm{KAFu.r}})\}$$

Der semantische Aspekt eines Fakts - die bekannte Gültigkeit der Formel $\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}})$ unter einer Interpretation I_r oder in einem Modellzustand "r" - wurde früher durch die Notation $\mathrm{fakt}_r(\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}}))$ eingeführt und erläutert. Da die gültige Formel $\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}})$ aus dem Blickwinkel von Multimengen in mehrfachen Kopien mit der Multiplizität $\mathrm{mu}_{u.r}$ vorkommen kann, wird die Notation eines Fakts auf das geordnete 2-Tupel $\mathrm{fakt}_r(\mathrm{mu}_{u.r},\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}}))$ erweitert. Folglich wird die Fakten(multi)menge $\mathrm{FAK}_{u.r}$ für ein Prädikatssymbol $\mathrm{Prä}_u$ in einem Modellzustand "r"[38] definiert durch eine Multimenge $\mathrm{Mult}_{\mathrm{KAFu.r}}$ über der Trägermenge KAF_u sowie durch die Beziehung:

$$\mathrm{FAK}_{u.r} = \{\mathrm{fakt}_r(\mathrm{mu}_{u.r},\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}})): \exists(\mathrm{Mult}_{\mathrm{KAFu.r}} \in \mathrm{MKAF}_{u.r}): \ldots$$
$$(\mathrm{mu}_{u.r},\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}})) \in \mathrm{Mult}_{\mathrm{KAFu.r}}\}$$

Jedes Element $\mathrm{fakt}_r(\mathrm{mu}_{u.r},\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}}))$ aus der Faktenmultimenge $\mathrm{FAK}_{u.r}$, das eine Multiplizität $\mathrm{mu}_{u.r} \geq 2$ besitzt, wird als Multifakt bezeichnet. Denn *dasselbe* Prädikatsvorkommnis $\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}})$ ist dann als faktische Formel mit der Multiplizität $\mathrm{mu}_{u.r}$ *mehrfach* gültig und entsprechend $\mathrm{mu}_{u.r}$-fach in der Faktenmultimenge $\mathrm{FAK}_{u.r}$ enthalten. Jedes Multifakt repräsentiert in formal präzisierter Weise die eingangs thematisierte multiple Formelgültigkeit.

Die potentielle Faktenmenge PFAK_u des Prädikatssymbols $\mathrm{Prä}_u$ ergibt sich entsprechend zu ihrer früheren Definition auf der Basis konventioneller Mengen nunmehr in bezug auf Multimengen bei binärer Mengennotation zu:

$$\mathrm{PFAK}_u = \{\mathrm{fakt}(\mathrm{mu}_u,\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}})): \mathrm{mu}_u \in \mathcal{N}_0 \wedge \mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}}) \in \mathrm{KAF}_u\}$$

Eine kompaktere Notation läßt sich erzielen, wenn auf den bereits eingeführten Faktoperator "fak" zurückgegriffen wird. Dann gilt für die potentielle Faktenmenge des Prädikatssymbols $\mathrm{Prä}_u$:

$$\mathrm{PFAK}_u = \mathrm{fak}((\mathcal{N}_0 \times \mathrm{KAF}_u)$$

Der indizierte Faktoperator "fak_r" erweitert den Faktoperator "fak": Jedes potentielle Fakt fakt(...), das durch den Operator "fak" erzeugt würde, wird durch den Operator "fak_r" als Fakt $\mathrm{fakt}_r(\ldots)$ unter der Interpretation I_r generiert. Jede zulässige Faktenmenge $\mathrm{FAK}_{u.r}$ aus $\mathrm{mu}_{u.r}$-fach gültigen konstanten atomaren Formeln $\mathrm{prä}_u(\mathrm{ob}_1,\ldots,\mathrm{ob}_{\mathrm{Ku}})$ entsteht daher aus einer Anwendung des indizierten Faktoperators "fak_r" auf eine Multimenge $\mathrm{Mult}_{\mathrm{KAFu.r}}$, die über der Mange KAF_u konstanter atomarer Formeln für das Prädikatssymbol $\mathrm{Prä}_u$ definiert ist[39]:

$$\mathrm{FAK}_{u.r} = \mathrm{fak}_r(\mathrm{Mult}_{\mathrm{KAFu.r}})$$

Analog zum indizierten Faktoperator "fak_r" läßt sich ein indizierter Potenzmengenoperator "pot_r" einführen. Er bildet jede Menge auf die Menge aller ihrer Teilmengen ab und ergänzt dabei deren Elemente mit dem Index "r". Dann kann jede zulässige Faktenmenge $FAK_{u.r}$ ebenso dargestellt werden als:

$$FAK_{u.r} \in pot_r(PFAK_u)$$

Der indizierte Faktoperator "fak_r" wird so erweitert, daß er sich auch auf Multimengenfamilien anwenden läßt. Wenn MUL_D diejenige Menge darstellt, die alle Multimengen $Mult_{D.k}$ über derselben Trägermenge D in binärer Mengennotation mit $k \in \{1,...,K\}$ und $K \in \mathcal{N}_+$ umfaßt, so gilt:

$$fak_r: \quad MUL_D = \{Mult_{D.k}: k=1,...,K\} \rightarrow fak_r(MUL_D) = \{fak_r(Mult_{D.k}): k=1,...,K\}$$

Dann gilt für jede Faktenmenge:

$$FAK_{u.r} \in (fak_r(mul_r(KAF_u)))$$

Wenn ein prädikatenlogisches Formelsystem die Prädikatssymbolemenge $PRÄ=\{Prä_u: u=1, ...,U\}$ besitzt, wird die Gesamtheit aller seiner Fakten unter einer Interpretation I_r als Familie $FAK_r=(FAK_{u.r}: u=1,...,U)$ aller prädikatspezifischen Faktenmengen $FAK_{u.r}$ festgelegt. Im Gegensatz zur früheren analogen Definition der Faktengesamtheit bezieht sie sich jetzt nicht mehr auf konventionelle Faktenmengen für die einzelnen Prädikatssymbole, sondern auf deren Faktenmultimengen in binärer Mengennotation. Da die prädikatspezifischen Faktenmengen disjunkt sind[40)], kann die Faktengesamtheit ebenso als Vereinigungsmenge aller prädikatspezifischen Faktenmengen dargestellt werden. Für diese systemspezifische Faktenmenge[41)] FAK_r gilt:

$$\begin{aligned} FAK_r &= \cup(u=1,...,U): FAK_{u.r} \\ &= \{fakt_r(mu_{u.r},prä_u(ob_1,...,ob_{Ku})): \forall(u \in \{1,...,U\})\ \exists(Mult_{KAFu.r} \in MKAF_u): ... \\ &\qquad (mu_{u.r},prä_u(ob_1,...,ob_{Ku})) \in Mult_{KAFu.r}\} \end{aligned}$$

Analog zu prädikatspezifischen Fakten(multi)mengen $FAK_{u.r}$, die durch Multimengen $Mult_{KAFu.r}$ über konstanten Prädikatemengen KAF_u definiert wurden, lassen sich für jedes Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ auch Multimengen $MTA_{u.v}$[42)] aus teilevaluierten atomaren Formeln $prä_u(te_1,...,te_{Ku})$ einführen[43)]. Jede Multimenge $MTA_{u.v}$ ist eine Multimenge $Mult_{TAFu.v}$ über der Trägermenge TAF_u in binärer Mengennotation mit $v \in \mathcal{N}_+$: $MTA_{u.v}=Mult_{TAFu.i.}$ Die Trägermenge TAF_u wurde bereits als teilevaluierte Prädikatemenge definiert. Jedes Element aus einer Multimenge $MTA_{u.v}$ stellt ein geordnetes 2-Tupel $(mu_{u.v},prä_u(te_1,...,te_{Ku}))$ dar. Seine zweite Komponente $prä_u(te_1,...,te_{Ku})$ ist eine atomare Formel mit teilevaluierten Termen te_k für $k \in \{1,...,K_u\}$. Die erste Tupelkomponente $mu_{u.v}$ gibt die Anzahl der Vorkommnisse dieser Formel in der Multimenge $MTA_{u.v}$ an. Es gilt abermals die Vereinbarung, nur 2-Tupel mit positiven Multiplizitäten explizit aufzulisten. Mit $MTAF_u$ wird die Menge aller unterschiedlichen Multimengen $MTA_{u.v}=Mult_{TAFu.v}$ bezeichnet, die sich über der Trägermenge TAF_u bilden lassen. Dann gilt analog zu den oben entfalteten Konstrukten, nunmehr allerdings ohne Berücksichtigung von Fakten:

$$\mathrm{mult}_{\mathrm{TAFu.v}}: \quad \mathrm{TAF}_u \rightarrow \mathcal{N}_0$$
$$\mathrm{prä}_u(\mathrm{te}_1,\ldots,\mathrm{te}_{Ku}) \rightarrow \mathrm{mult}_{\mathrm{TAFu.v}}(\mathrm{prä}_u(\mathrm{te}_1,\ldots,\mathrm{te}_{Ku})) = \mathrm{mu}_{u.v}$$

$\mathrm{MTA}_{u.v} = \mathrm{Mult}_{\mathrm{TAFu.v}}$

mit:

$$\mathrm{Mult}_{\mathrm{TAFu.v}} = \{(\mathrm{mu}_{u.v},\mathrm{prä}_u(\mathrm{te}_1,\ldots,\mathrm{te}_{Ku})): \mathrm{mu}_{u.v} \in \mathcal{N}_0 \wedge \mathrm{prä}_u(\mathrm{te}_1,\ldots,\mathrm{te}_{Ku})) \in \mathrm{TAF}_u \wedge \ldots$$
$$\mathrm{mu}_{u.v} = \mathrm{mult}_{\mathrm{TAFu.v}}(\mathrm{prä}_u(\mathrm{te}_1,\ldots,\mathrm{te}_{Ku}))\} \subseteq (\mathcal{N}_0 \times \mathrm{TAF}_u)$$

$\mathrm{MTAF}_u = \mathrm{mul}_v(\mathrm{TAF}_u)$

mit:

$$\mathrm{mul}_v(\mathrm{TAF}_u) = \{\mathrm{Mult}_{\mathrm{TAFu.v}}: \mathrm{Mult}_{\mathrm{TAFu.v}} \subseteq (\mathcal{N}_0 \times \mathrm{TAF}_u) \wedge \ldots$$
$$(\forall(\mathrm{prä}_u(\mathrm{te}_1,\ldots,\mathrm{te}_{Ku}) \in \mathrm{TAF}_u)\ \underline{\exists}(\mathrm{mu}_{u.v} \in \mathcal{N}_0): \ldots$$
$$(\mathrm{mu}_{u.v},\mathrm{prä}_u(\mathrm{te}_1,\ldots,\mathrm{te}_{Ku})) \in \mathrm{Mult}_{\mathrm{TAFu.v}})\}$$

Anmerkungen zum Kapitel:

1) In jeder konventionell definierten Faktenmenge FAK_r wird jede konstante atomare Formel $prä_u(ob_1,...,ob_{K_u})$ *höchstens einmal* als Fakt angeführt. Denn das potentielle Fakt $prä_u(ob_1,...,ob_{K_u})$ ist in einer Faktenmenge FAK_r nur dann - und zwar genau einmal - als tatsächliches Fakt $fakt_r(prä_u(ob_1,...,ob_{K_u}))$ enthalten, wenn die Formel $prä_u(ob_1,...,ob_{K_u})$ unter der jeweils betrachteten Interpretation I_r gültig ist. Daher kann dasselbe Fakt in einer Faktenmenge niemals mehrfach vorkommen. Alle Fakten $fakt_r(prä_u(ob_1,...,ob_{K_u}))$ aus derselben Faktenmenge sind deshalb paarweise voneinander unterschieden. Dies mußte bereits bei der Definition von Fakten*mengen* implizit präsupponiert werden, da der konventionelle Mengenbegriff ohnehin nur für paarweise wohlunterschiedene Elemente definiert ist.

2) Denn auf derselben Stelle eines Netzes können sich unter derselben Netzmarkierung beliebig viele Kopien derselben Marke befinden. Dies wird besonders deutlich bei Stelle/Transition-Netzen, für die ohnehin nur eine Marke definiert ist. Die Anzahl der Kopien dieser einen unstrukturierten Marke kann aber jeden beliebigen Werte annehmen.

3) Da Fakten mit Prädikatsextensionen gleichwertig sind, trifft dieselbe Quantifizierungslücke auch auf jene Extensionen zu. Sie gilt nicht spezifisch für Fakten, sondern für die prädikatenlogische Semantik schlechthin. Die prädikatenlogische Syntax leidet dagegen nicht unter einem analogen Defizit, weil die syntaktische Formulierung quantitativer - z.B. arithmetischer - Prädikate keine Schwierigkeiten bereitet.

4) Unabhängig von diesem quantitativen Aspekt besitzt das Markenkonzept aber auch einen qualitativen Aspekt, sofern strukturierte Marken zugelassen werden. Denn diese Marken können sich durch unterschiedliche interne Strukturen auch qualitativ voneinander unterscheiden. Solche strukturierten Marken werden für Synthetische Netze später ausführlich behandelt.

5) Anknüpfungspunkt dieser Darstellungsweise ist die Einführung von Fakten $fakt_0(prä_u(ob_1,...,ob_{K_u}))$ durch eine PSIG-Algebra für eine prädikatenlogische Signatur PSIG. Daher wirken sich die früher behandelten prädikatenlogischen Komplikationen der Faktenrepräsentation hier nicht mehr aus.

6) Obwohl der prädikatenlogische Gültigkeitsbegriff sich nicht "vervielfachen" läßt, können multiple Prädikatsgültigkeiten dennoch sinnvoll erklärt werden.

7) Die Prädikatenlogik kennt grundsätzlich nur die Dichotomie zwischen entweder Gültigkeit oder aber Ungültigkeit einer konstanten Formel, nicht jedoch die Kardinalität multipel definierter gültiger Formeln. Diese konventionelle prädikatenlogische Sichtweise liegt der Einschränkung von REISIG (1989a), S. 36, zugrunde, nur dann von Prädikat/Transition-Netzen zu sprechen, wenn sich auf jeder Stelle von jeder Marke höchstens eine Kopie befinden darf. In diesem Fall repräsentieren die Stellen eines Prädikat/Transition-Netzes Prädikatssymbole mit zeitlich variablen, aber konventionell definierten Faktenmengen. In ihnen darf jedes potentielle Fakt nur höchstens einmal tatsächlich enthalten sein. Dies entspricht den oben erwähnten Prädikat/Ereignis-Netzen.
Allerdings lassen sich multiple Prädikatsgültigkeiten als ein Reflex auffassen, der die Erweiterung der Prädikatenlogik um arithmetische Ausdrucksmöglichkeiten widerspiegelt. Denn auf der einen Seite können die Markierungen von Stellen, denen Prädikatssymbole zugeordnet sind, die *Gültigkeit* atomarer prädikatenlogischer Formeln ausdrücken. Auf der anderen Seite werden die *Anzahlen* von Markenkopien, die sich auf den Stellen eines Netzes befinden, durch die Anwendung einer Schaltregel beeinflußt. Ihre Schaltwirkung kann in den arithmetischen Kategorien der Addition und Subtraktion von Markenkopien ausgedrückt werden. Die Kombination beider Aspekte involviert die Möglichkeit multipler Prädikatsgültigkeiten.

8) Der hierbei implizit hergestellte Sinnbezug wird weiter unten offengelegt.

9) Vgl. REISIG (1989a), S. 8.

10) Das gilt auch dann, wenn ein formales Objekt in einem Ausdruck nur einmal vorkommt. Auch dieses eine Objektvorkommnis stellt aus formalsprachlicher Perspektive eine Objektkopie dar. Durch diese prima facie eigentümliche Festlegung wird erreicht, daß der formalsprachliche Umgang mit solchen Objekten und ihren Kopien erheblich vereinfacht wird. Die Sprachsyntax kann dann allgemeingültig formuliert werden, weil sie unabhängig davon ist, wie groß die aktuelle und kontingente Anzahl der Objektvorkommnisse in formalsprachlichen Ausdrücken ist. Vgl. zum Kopie-Begriff für formale Objekte z.B. REISIG (1989a), S. 8.

11) Strenggenommen liegen die bisher eingeführten Marken von Stelle/Transition-Netzen nicht in mehreren Kopien vor. Vielmehr *sind* diese Marken die identischen Kopien von genau einem formalen Objekt, das bisher noch nicht explizit eingeführt worden ist. Dies wird später nachgeholt, wenn eine markenbezogene Signatur als Basis für Synthetische Netze mit strukturierten vorgestellt wird. Dort wird *die* Basismarke als dasjenige formale Objekt identifiziert, dessen Kopien die (unstrukturierten) Marken von Stelle/Transition-Netzen sind.

12) Der Begriff "Objektmodell" bezieht sich jeweils auf das komplexe reale Erkenntnisobjekt, d.h. auf den jeweils modellierungsrelevanten Realitätsausschnitt. Dieses eine Objekt umgreift seinerseits eine Vielzahl realer Teil-

objekte, die - infolge der Rekursivität des Objektbegriffs - selbst wieder Objekte darstellen. Diese letztgenannten Objekte sind die hier angesprochenen realen Objekte, die durch formale Objekte modelliert werden.

13) Eine andere Möglichkeit bestünde darin, nicht mit identischen Kopien von formalen Objekten zu arbeiten, sondern jedes Prädikat, das für ein formales Objekt einen qualitativen Sachverhalt ausdrückt, um die quantitative Angabe zu erweitern, für wie viele reale Objekte im modellierten Realitätsausschnitt dieser Sachverhalt zutrifft. Der Verf. erachtet es grundsätzlich als eine reizvolle Aufgabe, diese alternative Modellierungsweise mit dem hier gewählten Konzept der identischen Kopien desselben formalen Objekts zu vergleichen und eventuell sogar noch nach weiteren Modellierungsalternativen für dieselbe Problemstellung zu suchen. Doch das Studium dieser Alternativen liegt außerhalb des Erkenntnisrahmens dieser Arbeit.

14) Vgl. dazu die Erweiterung von Fakten zu Multifakten.

15) Daneben läßt sich noch ein Aspekt anführen, der nicht auf formalsprachliche Modellierungskonzepte beschränkt ist, sondern für Modellierungskonzepte schlechthin gilt. Der modelltheoretische Kerngedanke besteht - wie bereits oben ausgeführt wurde - darin, vereinfachte Abbilder von Realitätsausschnitten dadurch zu konstruieren, daß ähnliche, aber real verschiedene Objekte durch Klassenbildung im Modell als genau ein Objekt behandelt werden. (Dabei wird nicht mehr vorausgesetzt, daß dieses eine Objekt im Modell ein formales Objekt sein müßte.) Jede Modellkonstruktion kann daher bei abstrakter Betrachtungsweise als eine Abbildung vom n:1-Typ aufgefaßt werden mit $n \in \mathcal{N}_+$. Genau diese modelltheoretische n:1-Beziehung zwischen modelliertem Realitätsausschnitt und Modell spiegelt sich innerhalb des Modells in der Zulässigkeit von n Kopien für jeweils genau 1 Objekt wider. Diese Widerspiegelung besitzt zwar keine unmittelbare Konsequenz für die praktische Anwendung von Modellierungskonzepten. Doch kann ihr aus der Perspektive einer "(modell)theoretischen Ästhetik" ein Eigenwert zuerkannt werden. Eine Vertiefung solcher ästhetisierender Betrachtungen liegt jedoch außerhalb des Erkenntnisrahmens dieser Arbeit.

16) Bezugspunkt für die Unterscheidung zwischen außer- und Innersprachlichen Konzepten ist hier die Sprache des jeweils betrachteten formalen Modellierungskonzepts. Dessen ungeachtet sind auch die "außer"sprachlichen Aspekte selbst wieder sprachlich formuliert, allerdings in einer (natürlichen) Sprache, die hier nicht von Interesse ist.

17) Vgl. zu solchen Multimengen ("bags", "multisets") GOSTELOW (1971), S. 67ff.; CERF (1972), S. 21ff.; PETERSON,J. (1973a), S. 222f.; PETERSON,J. (1976), S. 22f.; CRESPI-REGHIZZI (1976a), S. 131f.; CRESPI-REGHIZZI (1976b), S. 176; ZERVOS (1977), S. 282ff.; RIEDEMANN (1979), S. 4ff.; PETERSON,J. (1981), S. 237ff.; JENSEN (1982a), S. 4; GOLTZ,U. (1983c), S. 12; REISIG (1986a), S. 150f.; JENSEN (1987a), S. 254f.; GENRICH (1988b), S. 232; RECK (1988), S. 80f.; REISIG (1989a), S. 1 u. 8ff., der Multimengen sogar in den Kontext sortierter Algebren einbettet; MARTI-OLIET (1989), S. 316.

18) Obwohl Multimengen durch ihre Multimengen*funktionen* originär als Abbildungen definiert sind, läßt sich dennoch ihre etablierte Bezeichnung als Multi*mengen* rechtfertigen. Denn es ist einerseits möglich, Multimengen als konventionelle Mengen von 2-Tupeln darzustellen. Darauf wird noch zurückgekommen. Andererseits lassen sich konventionelle Mengen als Grenzfälle von Multimengen auffassen; vgl. dazu auch PETERSON,J. (1976), S. 23; PETERSON,J. (1981), S. 238. Jede konventionelle Teilmenge M aus einer beliebigen Obermenge D kann dabei einer mengenspezifischen Multimenge $mult_{D.M}$ mit der Trägermenge D und dem speziellen binären Nachbereich $\{0,1\}$ wie folgt eineindeutig zugeordnet werden:

$$mult_{D.M}: D \rightarrow \{0,1\}$$

$$d \rightarrow mult_{D.M}(d) = \begin{cases} 1; \text{ sofern } d \in M \\ 0; \text{ sofern } d \neq M \end{cases}$$

Diese eineindeutige Zuordnung zwischen konventioneller Menge M und Multimengenfunktion $mult_{D.M}$ läßt sich komprimiert ausdrücken als die Äquivalenz: $d \in M \Leftrightarrow mult_{D.M}(d)=1$. Wenn für die Multimenge $mult_{D.M}$ die unten angeführte Listennotation gewählt wird und dabei die dort vereinbarten notationellen Vereinfachungen ausgeschöpft werden (Fortlassen der Koeffizienten $mu_d=1$), dann fallen die Notationen der Menge M und der Multimenge $mult_{D.M}$ zusammen, sofern von dem marginalen Notationsunterschied zwischen listenbegrenzenden Klammern "[" und "]" einerseits sowie Mengenklammern "{" bzw. "}" andererseits abgesehen wird. Denn dann gilt für eine beliebige endliche Menge M mit insgesamt N Elementen d_n aus der Trägermenge D:

$$M=\{d_1,...,d_N\} \Leftrightarrow mult_{D.M}=[d_1,...,d_N].$$

Darüber hinaus läßt sich anhand dieser Grenzfallbetrachtung in formaler Weise präzisieren, worin die Erweiterung von Multimengen gegenüber konventionellen Mengen besteht: Multimengen $mult_D$ erweitern den mengenspezifischen binären Nachbereich $\{0,1\}$ auf den allgemeinen natürlichzahligen Nachbereich $\mathcal{N}_0$.

Schließlich besitzt die Multimengenfunktion $mult_{D,M}$ für Mengen M die Qualität einer charakteristischen Funktion $\#_D$. Die charakteristische Funktion $\#_D$ einer Menge M nimmt für jedes Element d aus einer Trägermenge D genau dann den Wert "1" ("0") an, wenn dieses Element zugleich auch zur Menge M gehört: $\#_D(d)=mult_D(d)=1 \Leftrightarrow d \in M$ und $\#_D(d)=mult_D(d)=0 \Leftrightarrow d \notin M$. Vgl. zu solchen charakteristischen Funktionen ROGERS,H. (1967), S. XVIf.; PETERSON,J. (1981), S. 238.

19) Multimengen können auch über abzählbar unendlichen Trägermengen definiert werden; doch ist der infinite Fall für die nachfolgenden Ausführungen irrelevant.

20) Dieser Ansatz wird später für die Definition von Faktenmultimengen verwendet.

21) Es ist nicht erforderlich, hierbei auf eine bestimmte Multimengenfunktionen $mult_D$ Bezug zu nehmen.

22) Vgl. zu weiteren Möglichkeiten der - inhaltlich gleichwertigen - Repräsentation von Multimengen CRESPI-REGHIZZI (1976a), S. 131f.; CRESPI-REGHIZZI (1976b), S. 176; ZERVOS (1977), S. 282f.; RIEDEMANN (1979), S. 4. Hierbei ist besonders die Parikhfunktion pf hervorzuheben. Vgl. zu dieser Funktion CRESPI-REGHIZZI (1976a), S. 131; CRESPI-REGHIZZI (1976b), S. 176; LANDWEBER (1978), S. 354; MAYR,E. (1980a), S. 8. Die Parikhfunktion beruht auf einer zweistelligen, tupelbezogenen Erweiterung $\#_{D,2}$ der einstelligen und mengenbezogenen charakteristischen Funktion $\#_D$, die bereits in einer früheren Anmerkung eingeführt wurde. Die erweiterte charakteristische Funktion $\#_{D,2}$ bildet jedes Element aus der Trägermenge D und ein beliebiges Tupel "tup" auf die Anzahl (Multiplizität) ab, in der dieses Element in jenem Tupel "tup" vorkommt. Dabei wird die Trägermenge D stets als endliche Menge mit insgesamt N_D Elementen d_n vorausgesetzt. Die Parikhfunktion pf_D ist auf dieser Grundlage für jedes Tupel "tup" definiert, das ein Element aus dem freien Monoid über der Trägermenge D darstellt. Sie bildet dieses Tupel auf ein N_D-stelliges Tupel ab (Bildtupel). Die Komponenten des Bildtupels und die Elemente d_n der Trägermenge D mit $n \in \{1,...,N_D\}$ sind einander eineindeutig zugeordnet. Jede Komponente des Bildtupels gibt die Anzahl der Vorkommnisse (Kopien) des zugehörigen Elements d_n im Tupel "tup" an. Also gilt für beliebige Trägermengen D:

$$\begin{array}{ll} pf_D: & D^* \rightarrow \mathcal{N}_0^{ND} \\ & tup \rightarrow pf_D(tup) = (\#_{D,2}(d_n,tup): n \in \{1,...,N_D\}) \end{array}$$

Mit Hilfe der Parikhfunktion läßt sich eine Multimenge $mult_D$, die in der unten erläuterten Listennotation als ein Tupel "tup" mit $tup=mult_D$ vorliegt, ebenso darstellen als das Bild $pf_D(tup)$ der Parikhfunktion für dieses Tupel "tup". Allerdings setzt dies eine Vereinbarung darüber voraus, welche Stellen des Bildtupels $pf_D(tup)$ jeweils welches Element d_n aus der Trägermenge D repräsentieren. Diese Vereinbarung läßt sich durch die nachfolgend angeführten Notationen von Multimengen als Listen oder formale Summen vermeiden. Daher werden Parikhfunktionen in dieser Arbeit nicht weiter berücksichtigt.
Besonders einfach ist die Darstellung von Multimengen in der Programmiersprache SMALLTALK: dort wird ein spezieller Datentyp "bag" für die unmittelbare extensionale Spezifizierung von Multimengen vorgesehen; vgl. O.V. (1981b), S. 47; BAUMEISTER,H. (1987), S. 247; SCHMITZ,L. (1988b), S. 56 u. 58. Leider existiert diese Option für die hier zugrundegelegte Programmiersprache PROLOG (noch) nicht.

23) Die Listennotation dient vor allem der Implementierung von Multimengen durch Automatische Informationsverarbeitungssysteme; vgl. dazu auch die nachfolgende Anmerkung.

24) Der Klammertyp "[","]" entspricht der Listennotation in der Programmiersprache PROLOG. Vgl. zur Definition und Verwendung von PROLOG-Listen PROLOG (o.J.), S. 45ff.; KINNEBROCK (1988), S. 51ff.; CORDES (1988), S. 103ff.

25) Die formale Summennotation wird später zur Beschriftung der Kanten von Synthetischen Netzen verwendet.

26) Vgl. PAUL,W. (1978), S. 72; REISIG (1986a), S. 150f.; JENSEN (1987a), S. 254; GENRICH (1987a), S. 214f.

27) Aufgrund der Kommutativität formaler Summen bleibt die Mehrdeutigkeit aufgrund von Permutationen der Komponenten erhalten. Die Mehrdeutigkeit infolge unterschiedlichen Ausschöpfens von notationellen Vereinfachungsoptionen besteht ebenso.

28) Vgl. zu diesen "Rechenregeln" für Multimengen die Quellen, die bereits zu Multimengen im allgemeinen angegeben wurden.

29) Daher werden die konventionellen Operatoren und Relatoren für die arithmetischen Grundoperationen und -relationen in unveränderter Notation auf die hieraus abgeleiteten Operationen bzw. Relationen von Multimengen übertragen. Beispielsweise wird der Infixoperator "+" für die arithmetische Grundoperation der Addition auch für die "Addition" von Multimengen benutzt.

30) Die Subtraktion zweier Multimengen stellt nur eine partielle Funktion dar, weil sie nur dann mit natürlichzahligen Funktionswerten definiert ist, wenn der Minuend mindestens so groß wie der Subtrahend ausfällt.

31) Die Erweiterung geschieht in trivialer Weise durch sukzessives Addieren bzw. Multiplizieren von je zwei Multimengen. Dabei stellt jedes Additions- bzw. Multiplikationsergebnis eine Multimenge dar, die ihrerseits in den jeweils nächsten Sukzessionsschritt eingeht.

32) Die Zählfunktion # bildet jede Menge ME auf die Anzahl ihrer Elemente ab. Sie wurde bereits als Kardinalität von Mengen eingeführt. Sie läßt sich aber ebenso aus der Multimengenfunktion herleiten. Zu diesem Zweck wird für jede endliche konventionelle Menge ME jeweils eine Multimenge $mult_{D.ME}$ betrachtet, die zwei Bedingungen gerecht wird. Erstens besitzt die Multimenge $mult_{D.ME}$ eine beliebige Obermenge D der Menge ME als Trägermenge. Zweitens erfüllt die Multimenge $mult_{D.ME}$ die Äquivalenzen $d \in ME \Leftrightarrow mult_{D.ME} = 1$ und $d\ ME \Leftrightarrow mult_{D.ME} = 0$. Dann definiert die Multimenge $mult_{D.ME}$ die Zählfunktion # für die endliche Menge ME durch folgende Funktionsvorschrift: $\#(ME) = \Sigma(d \in D):mult_D(d)$.

33) Die Varianten gehen vor allem aus der Permutation von Listenelementen und der Kommutation formaler Summanden hervor. Sie beziehen sich daneben auch auf die bereits vereinbarten Notationsvereinfachungen für Listen und formale Summen.

34) Denn in ihrer binären Notation stellen Faktenmultimengen tatsächlich konventionell definierte Mengen aus 2-Tupeln - den unten näher erläuterten Multifakten - dar. Die früher eingeführten konventionellen Faktenmengen sind hierin als Grenzfall mit der Multiplizität $mu_{u.r} = 1$ enthalten. Denn jedes Fakt $fakt_r(prä_u(ob_1,...,ob_{Ku}))$ aus einer konventionell definierten Faktenmenge läßt sich als Element $(1,fakt_r(prä_u(ob_1,...,ob_{Ku})))$ einer Faktenmultimenge reformulieren. Die hierbei vorausgesetzte spezielle Faktendarstellung im Kontext von Multimengen wird anschließend näher ausgeführt.

35) Dieser Zusatz ist insofern wichtig, als das Multimengenkonzept nunmehr auch erlaubt, von mehreren, aber *identischen* Vorkommnissen desselben Prädikatssymbols zu reden.

36) Diese Einschränkung auf Notationen mit positiven Multiplizitäten wird fortan nicht mehr ständig wiederholt. Sie gilt als implizit vereinbart, sofern nicht ausdrücklich andere Festlegungen erfolgen.

37) Vgl. zur Erweiterung von Operatoren um eine Teilindex-Generierung die Anmerkung zum indizierten Operator "fak_r".

38) Der Einfachheit halber wird fortan nur noch auf Zustände "r" eines prädikatenlogischen Objektmodells explizit Bezug genommen. Sie entsprechen implizit jeweils einer Interpretation I_r des Objektmodells.

39) Dies schließt auch die leere Faktenmenge $FAK_{u.r} = \emptyset$ mit $FAK_{u.r} = \{(0,prä_u(ob_1,...,ob_{Ku}):prä_u(ob_1,...,ob_{Ku}) \in KAF_u\}$ ein. Denn die Multimengen $Mult_D$ wurden oben so eingeführt, daß sie auch den degenerierten Fall der leeren Multimenge mit (mu_d,d) und $mu_d = 0$ für alle $d \in D$ umfassen.

40) Die Trägermengen KAF_1 und KAF_2 konstanter atomarer Vorkommnisse für die Faktenmengen $FAK_{1.r}$ bzw. $FAK_{2.r}$ zweier beliebiger, aber verschiedener Prädikatssymbole $Prä_1$ bzw. $Prä_2$ sind wegen der unterschiedlichen Indizierung immer disjunkt. Multimengen über disjunkten Trägermengen sind ebenfalls disjunkt. A fortiori müssen auch die Faktenmengen $FAK_{u.r}$ für die Prädikatssymbole $Prä_u$ aus der Prädikatssymbolemenge PRÄ eines prädikatenlogischen Formelsystems stets paarweise disjunkt sein.

41) Es gilt abermals, daß die Familie aller prädikatspezifischen Faktenmengen wegen deren Disjunktheit ebenso als Vereinigungsmenge aller Faktenmengen dargestellt werden kann. Hinzu kommt jetzt die o.a. Vereinbarung, Multimengen vereinfacht als Mengen anzusprechen, wenn sie in binärer Mengennotation behandelt werden. Daher läßt sich die Familie aller prädikatspezifischen Faktenmultimengen ebenso als Vereinigungsmenge aller disjunkten Faktenmultimengen behandeln. Das Resultat ist wiederum eine Faktenmultimenge, die wegen ihrer binären Mengennotation als systemspezifische Faktenmenge angesprochen werden darf.

42) Der zweite Teilindex "v" wird später durch Bezugnahme auf Transaktionen tr_v erklärt.

43) Sie werden später für die Definition einer operationalen Semantik durch Übergangsoperationen benötigt.

4.2.3.4 Ein Übergangsschema

Es wird ein Übergangsschema eingeführt, um bei der späteren Entfaltung Synthetischer Netze eine formale Explizierungslücke zu schließen[1]. Sie betrifft die Definition der Schaltregel von Petrinetzen. Zwar stellt die Schaltregel eines der herausragenden Kennzeichen des Petrinetz-Konzepts dar. Dennoch wird sie in formalen Netzdefinitionen im allgemeinen nicht angeführt. Auf dieses erstaunliche Definitionsdefizit wurde bereits bei der Erläuterung der Stelle/Transition-Netze hingewiesen. Auch im Rahmen der formal wesentlich anspruchsvolleren Konstruktion von Prädikat/Transition-Netzen wird die Schaltregel selbst nicht in der Netzdefinition aufgeführt[2]. Das gilt sogar für diejenigen Ansätze, die Netze vom Typ der Prädikat/Transition-Netze auf der Basis des Signaturkonzepts eingeführt haben[3].

Das Übergangsschema dient der formalen Darstellung und Ausführung von Übergangsoperationen tr_v[4]. Diese Übergangsoperationen bilden das Fundament einer operationalen Semantik für prädikatenlogische Objektmodelle[5]. Die operationale Semantik legt fest, wie sich das Wissen, das ein Modellierungsträger über Modellzustände besitzt, bei Operationsausführungen verändert.

Das Wissen eines Modellierungsträgers über den jeweils aktuellen Zustand "r" des Objektmodells drückt eine Fakten(multi)menge[6] FAK_r aus[7]. Sie wird stets in ihrer binären Mengennotation mit positiven Multiplizitäten[8] dargestellt. Ein einzelnes Fakt nimmt daher die Form eines Multifakts $fakt_r(mu_{u.r}, prä_u(ob_1,...,ob_{Ku}))$ mit $mu_{u.r} \in \mathcal{N}_+$ an. Dabei bedeutet $prä_u(ob_1,...,ob_{Ku})$ eine konstante interpretierte atomare Formel, die als gültig bekannt ist und daher aus der aktuellen Extension $EXT_{u.r}$ des zugrundeliegenden Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ stammt. Die Multiplizität $mu_{u.r}$ der Prädikatsgültigkeit gibt an, wie viele identische Vorkommnisse (Kopien) der Formel $prä_u(ob_1,...,ob_{Ku})$ für das Prädikatssymbol $Prä_u$ im aktuellen Modellzustand "r" als gültig bekannt sind[9].

Der aktuelle Modellzustand "r", auf den eine Übergangsoperation angewendet wird, wird als Referenzzustand bezeichnet. Der Modellzustand, der durch das Ausführen einer Übergangsoperation hervorgebracht wird, heißt ein Folgezustand "f". Das Wissen über diesen Folgezustand wird mittels der Faktenmenge FAK_f repräsentiert. Die involvierten Faktenmengen FAK_r und FAK_f werden entsprechend als Referenz- und Folgefaktenmenge angesprochen. Jede Übergangsoperation repräsentiert explizites (meta-)metasprachliches[10] Wissen über die Möglichkeit, eine vorliegende Referenzfaktenmenge FAK_r in eine Folgefaktenmenge FAK_f[11] zu transformieren[12]. Die allgemeine Form, die von allen zulässigen Übergangsoperationen tr_v erfüllt wird, stellt ein Übergangsschema ÜS dar.

Der Grundzustand[13] des Objektmodells mit "r=0" wird durch genau eine Ausgangsfaktenmenge FAK_0 eindeutig ausgezeichnet. Sie läßt sich durch die Menge aller Fakten $fakt_0(mu_{u.0}, prä_u(ob_1,...,ob_{Ku}))$ festlegen, die in den Prädikatsdeklarationen einer algebraisch-prädikatenlogischen Spezifikation eingeführt oder später auf Faktenmultimengen zurückgeführt werden. Die Ausgangsfaktenmenge darf auch die leere Faktenmenge $FAK_0=\varnothing$ sein. Sie drückt dann aus, daß im Ausgangszustand des Objektmodells überhaupt kein atomares Prädikatsvorkommnis als gültig bekannt ist[14].

Um auszudrücken, wie Operationsausführungen das Wissen eines Modellierungsträgers über Modellzustände verändern, wird eine Reihe von Vereinbarungen für eine beliebige Übergangsoperation tr_v vorausgesetzt:

❑ PRÄ ist die nicht-leere, aber endliche Menge aller Prädikatssymbole $Prä_u(st_1,...,st_{Ku})$, die für ein prädikatenlogisches Formelsystem definiert sind. IPRÄ mit $\#(IPRÄ) \in \mathcal{N}_+$ ist die Indexmenge aller Prädikatssymbole, die zur Menge PRÄ gehören.

- Der Vorbereich $VB_v=\{Prä_u(st_1,...,st_{Ku}): u\in IVB_v\}$ ist eine endliche Menge von Prädikatssymbolen $Prä_u$ mit der Indexmenge IVB_v.
- Der Nachbereich $NB_v=\{Prä_u(st_1,...,st_{Ku}): u\in INB_v\}$ ist eine endliche Menge von Prädikatssymbolen $Prä_u$ mit der Indexmenge INB_v.
- Der Informationsbereich $IB_v=\{Prä_u(st_1,...,st_{Ku}): u\in IIB_v\}$ ist eine endliche Menge von Prädikatssymbolen $Prä_u$ mit der Indexmenge IIB_v.
- Die Vereinigung $EB_v=VB_u\cup IB_u$ von Vor- und Informationsbereich heißt Einflußbereich. Die Indexmenge IEB_v aller zum Einflußbereich gehörigen Prädikatssymbole ist $IEB_v=IVB_v\cup IIB_v$.
- Die Vereinigung $WB_v=VB_u\cup NB_u$ von Vor- und Nachbereich heißt Wirkungsbereich. Die Indexmenge IWB_v aller zum Wirkungsbereich gehörigen Prädikatssymbole ist $IWB_v=IVB_v\cup INB_v$.
- Die Vereinigung $OB_v=WB_v\cup IB_v$ von Wirkungs- und Informationsbereich heißt Operationsbereich. Die Indexmenge IOB_v aller zum Operationsbereich gehörigen Prädikatssymbole ist $IOB_v=IWB_v\cup IIB_v$.
- Vor- und Nachbereich einerseits und Informationsbereich andererseits sind disjunkt: $VB_v\cap IB_v=\emptyset$ und $NB_v\cap IB_v=\emptyset$. Vor- und Nachbereich brauchen untereinander nicht disjunkt zu sein.
- Der Irrelevanzbereich IB_v ist die Restmenge, die von der Menge PRÄ aller Prädikatssymbole eines prädikatenlogischen Formelsystems verbleibt, nachdem sie um den Operationsbereich OB_v der Übergangsoperation tr_v vermindert worden ist: $IB_v=PRÄ\text{-}OB_v$.
- $FAK_{u.r}$ und $FAK_{u.f}$ sind die Faktenmultimengen für jedes Prädikatssymbol $Prä_u$, das im Operationsbereich OB_v der Übergangsoperation tr_v enthalten ist, vor bzw. nach Ausführung dieser Operation. Jedes Fakt wird als $fakt_r(mu_{u.r},prä_u(ob_1,...,ob_{Ku}))$ bzw. $fakt_f(mu_{u.f},prä_u(ob_1,...,ob_{Ku}))$ notiert.
- Für jedes Prädikatssymbol $Prä_u$ aus dem Vorbereich VB_v, dem Informationsbereich IB_v und dem Nachbereich NB_v existiert jeweils genau eine endliche Multimenge $MTAV_{u.v}$, $MTAI_{u.v}$ bzw. $MTAN_{u.v}$ teilevaluierter atomarer Formeln $prä_u(te_1,...,te_{Ku})$ in binärer Mengennotation mit positiven Multiplizitäten $mu_{u.i.}$ Jedes Element aus den Multimengen besitzt die Gestalt eines geordneten 2-Tupels $(mu_{u.v},prä_u(te_1,...,te_{Ku}))$.
- Mit Hilfe der Multimengen $MTAV_{u.v}$, $MTAI_{u.v}$ und $MTAN_{u.v}$ sind folgende Formelmultimengen und deren Familien für den Vor-, Informations-, Nach-, Einfluß-, Wirkungs- und Operationsbereich definiert:

  $MTAE_{u.v}\in\{MTAV_{u.v},MTAI_{u.v}\}$;
  $MTAW_{u.v}\in\{MTAV_{u.v},MTAN_{u.v}\}$;
  $MTAO_{u.v}\in\{MTAV_{u.v},MTAI_{u.v},MTAN_{u.v}\}$;
  $MTA_{u.v}\in\{MTAV_{u.v},MTAI_{u.v},MTAN_{u.v},MTAE_{u.v},MTAW_{u.v},MTAO_{u.v}\}$;
  $MTAV_v=(MTAV_{u.v}:u\in IVB_v)$;
  $MTAI_v=(MTAI_{u.v}:u\in IIB_v)$;
  $MTAN_v=(MTAN_{u.v}:u\in INB_v)$;
  $MTAE_v=(MTAE_{u.v}:u\in IEB_v)$;
  $MTAW_v=(MTAW_{u.v}:u\in IWB_v)$;
  $MTAO_v=(MTAO_{u.v}:u\in IOB_v)$;
  $MTA_v\in\{MTAV_v,MTAI_v,MTAN_v,MTAE_v,MTAW_v,MTAO_v\}$.
- $VA(MTA_{u.v})$ ist die Menge[15)] aller Variablen X, die in den teilevaluierten Termen te_k aus den Argumenten von atomaren Prädikatsvorkommnissen $prä_u(te_1,...,te_{Ku})$ aus den Multimengen $MTA_{u.v}$ enthalten sind.

- vb_c ist eine beliebige Variablenbindungsfunktion aus der Menge $VBM = \{vb_c: c = 1,...,C\}$ aller Variablenbindungsfunktionen, die sich für eine algebraisch-prädikatenlogische Spezifikation PSIG mit $C \in \mathcal{N}_+$ definieren lassen.
- $vb_c(prä_u(te_1,...,te_{Ku})) = prä_u(ob_1,...,ob_{Ku})$ ist das konstante atomare Formelvorkommnis, das aus der Bindung aller Variablen X im atomaren Prädikatsvorkommnis $prä_u(te_1,...,te_{Ku})$ durch die Variablenbindungsfunktion vb_c hervorgeht.
- $vb_c(MTA_{u.v})$ ist die Multimenge aller konstanten atomaren Formelvorkommnisse $prä_u(ob_1,...,ob_{Ku})$, die aus der Anwendung der Variablenbindungsfunktion vb_c auf das Formelvorkommnis $prä_u(te_1,...,te_{Ku})$ aus der Multimenge $MTA_{u.v}$ hervorgeht:

  $vb_c(MTA_{u.v}) = \{(mu_{u.v}, prä_u(ob_1,...,ob_{Ku})): ...$

  $prä_u(ob_1,...,ob_{Ku}) = vb_c(prä_u(te_1,...,te_{Ku})) \wedge (mu_{u.v}, prä_u(te_1,...,te_{Ku})) \in MTA_{u.v}\}.$
- $vb_c(MTA_v)$ ist die Familie aller Multimengen $vb_c(MTA_{u.v})$ aus konstanten atomaren Formelvorkommnissen, die aus den Anwendungen der Variablenbindungsfunktion vb_c auf die Formelvorkommnisse $prä_u(te_1,...,te_{Ku})$ aus allen Multimengen $MTA_{u.v}$ desselben Transaktionsbereichs hervorgehen:

  $vb_c(MTAV_v) = (vb_c(MTAV_{u.v}): u \in IVB_v);$
  $vb_c(MTAI_v) = (vb_c(MTAI_{u.v}): u \in IIB_v);$
  $vb_c(MTAN_v) = (vb_c(MTAN_{u.v}): u \in INB_v);$
  $vb_c(MTAE_v) = (vb_c(MTAE_{u.v}): u \in IEB_v);$
  $vb_c(MTAW_v) = (vb_c(MTAW_{u.v}): u \in IWB_v);$
  $vb_c(MTAO_v) = (vb_c(MTAO_{u.v}): u \in IOB_v).$
- RES_v ist eine endliche Menge prädikatenlogischer Formeln, die eine Teilmenge der Menge RES aller Restriktionsformeln aus einer algebraisch-prädikatenlogischen Spezifikation darstellt.

Unter diesen Voraussetzungen wird ein Übergangsschema ÜS definiert durch eine komplex strukturierte, bedingte Anweisung. Sie determiniert die Ausführung einer beliebigen Übergangsoperation tr_v eindeutig bis auf die Auswahl einer bestimmten, aber im Übergangsschema selbst noch nicht festgelegten sortentreuen Variablenbindungsfunktion vb_c. Die bedingte Anweisung des Übergangsschemas wird in Pseudocode notiert[16).

Definition: Übergangsschema

```
IF      prätest(FAK_u.r) = gültig für alle u∈IIB_v
|
THEN    IF      FAK_u.r ≥ vb_c(MTAE_u.v)  für alle u∈IEB_v
                AND
                haupttest(vb_c(MTAE_u.v)) = gültig  für alle u∈IEB_v
                AND
                vb_c(X_k) = vb_c(bestimme_k(X_k.1,...,X_k.Hk))
                mit:    ∀(h∈{1,...,H_k}) ∃(u∈IEB_v): X_k.h∈VA(MTAE_u.v)
                        für alle X_k∈VA(MTAN_u.v)
        |
        THEN    DO      Übergangsprozedur
                |
                        DO      Faktenentfernung
                        |       FAK_u.- := FAK_u.r - vb_c(MTAV_u.v)  für alle u∈IVB_v
                        |       FAK_u.- := FAK_u.r  für alle u∈(IPRÄ-IVB_v)
                        ENDDO   Faktenentfernung

                        DO      Faktenergänzung
                        |       FAK_u.+ := FAK_u.- + vb_c(MTAN_u.v)  für alle u∈INB_v
                        |       FAK_u.+ := FAK_u.-  für alle u∈(IPRÄ-INB_v)
                        ENDDO   Faktenergänzung
                |
                ENDDO   Übergangsprozedur

                IF      posttest(FAK_u.+) = gültig  für alle u∈IWB_v
                |
                THEN    DO      Übergangsakt
                        |       FAK_u.f := FAK_u.+  für alle u∈IPRÄ
                        |       übergang(vb_c(MTAO_v)) := gültig
                        ENDDO   Übergangsakt
                |
                ELSE    UNDO    Übergangsprozedur
                        |       FAK_u.f := FAK_u.r für alle u∈IPRÄ
                        |       übergang(vb_c(MTAO_v)) := ungültig
                        EUNDO   Übergangsprozedur
                |
                ENDIF
        ENDIF
ENDIF
```

Jede Übergangsoperation tr_v, die das Übergangsschema ÜS erfüllt, wird notiert als ein 7-Tupel, für das gilt:

$$
\begin{aligned}
tr_v = \quad & (\, VB(tr_v) = \{Prä_u(st_1,...,st_{Ku}) : u \in IVB_v\}, \\
& MTAV_v = \{MTAV_{u.v} : u \in IVB_v\}, \\
& IB(tr_v) = \{Prä_u(st_1,...,st_{Ku}) : u \in IIB_v\}, \\
& MTAI_v = \{MTAI_{u.v} : u \in IIB_v\}, \\
& NB(tr_v) = \{Prä_u(st_1,...,st_{Ku}) : u \in INB_v\}, \\
& MTAN_v = \{MTAN_{u.v} : u \in INB_v\}, \\
& RES_v \in \{\emptyset, \{for_{z(v.h)} : \underline{\exists}(H_v \in \mathcal{N}_+)\ \forall(h \in \{1,...,H_v\}) : for_{z(v.h)} \in RES\}\}\,)
\end{aligned}
$$

<u>Erläuterungen und Ergänzungen zur Übergangsschema-Definition:</u>

a) Die einzelnen Komponenten des Übergangsschemas ÜS lassen sich wie folgt charakterisieren:

- Für jedes Prädikatssymbol $Prä_u$ aus dem Informationsbereich IB_v spezifiziert das Prätestprädikat "prätest($FAK_{u.r}$)" eine Testbedingung, die von der Referenzfaktenmenge $FAK_{u.r}$ des Prädikatssymbols erfüllt werden muß. Die Prätestbedingung wird untersucht, *bevor* überhaupt damit begonnen wird, für die Übergangsoperation eine zulässige Variablenbindungsfunktion zu ermitteln. Falls die Prätestbedingung verletzt ist, wird die Ausführung der Übergangsoperation abgebrochen. Als implizite Voreinstellung[17)] jedes nicht explizit angeführten Prätestprädikats dient das immer gültige Prätestprädikat "T". Es stellt sicher, daß die Operationsausführung fortgesetzt wird, solange keine (andere) Prätestbedingung für das zugehörige Prädikatssymbol explizit definiert worden ist und solange keine definierten Prätestbedingungen für andere Prädikatssymbole verletzt werden.
- Die Ungleichungen $FAK_{u.r} \geq vb_c(MTAE_{u.v})$ für alle Prädikatssymbole $Prä_u$ aus dem Einflußbereich EB_v repräsentieren Inklusionstests. Sie beziehen sich jeweils auf den Vergleich zweier Multimengen aus konstanten atomaren Formelvorkommnissen[18)] und überprüfen hierbei, ob die zweite in der ersten Multimenge enthalten ist. Falls alle Ungleichungen erfüllt sind, ist gewährleistet, daß die Argumente der Fakten aus jeder Faktenmultimenge $FAK_{u.r}$ ausreichen, um für die Formelvorkommnisse aus der jeweils korrespondierenden Multimenge $MTAE_{u.v}$ unter der Variablenbindungsfunktion vb_c eingesetzt zu werden. Andernfalls wird die Ausführung der Übergangsoperation abgebrochen.
- Die Haupttestprädikate "haupttest($vb_c(MTAE_{u.v})$)" legen für jedes Prädikatssymbol $Prä_u$ aus dem Einflußbereich EB_v eine Testbedingung fest, welche die Multimenge $MTAE_{u.v}$ seiner atomaren Vorkommnisse unter der Variablenbindungsfunktion vb_c erfüllen muß. Sie besitzen die Qualität einer Restriktion, weil sie die zulässigen Variablenbindungsfunktionen vb_c aus der Menge VBM auf solche einschränken, welche alle Haupttestbedingungen erfüllen. Als implizite Voreinstellung für jedes nicht explizit definierte Haupttestprädikat dient wiederum das immer gültige Prätestprädikat "T".
- Die Bestimmungsfunktionen "bestimme$_k(X_{k.1},...,X_{k.Hk})$" können sich auf beliebige Variablen $X_{k.h}$ aus den Mengen $VA(MTAE_{u.v})$ mit $h \in \{1,...,H_k\}$ und $H_k \in \mathcal{N}_+$ erstrecken[19)]. Jede der Variablenmengen $VA(MTAE_{u.v})$ umfaßt alle Variablen, die in den Argumenten der Formelvorkommnisse aus der Multimenge $MTAE_{u.v}$ eines Prädikatssymbols $Prä_u$ aus dem Einflußbereich EB_v der Übergangsoperation tr_v enthalten sind. Die Variable X_k stammt aus der Variablenmenge $VA(MTAN_{u.v})$, die in analoger Weise alle Variablen enthält, die in Argumenten der Formelvorkommnisse aus den Multimengen $MTAN_{u.v}$ eines Prädikatssymbols $Prä_u$ aus dem Nachbereich NB_v der Übergangsoperation tr_v vorkommen. Die Bestimmungs-

gleichungen "$vb_c(X_k) = vb_c(bestimme_k(X_{k.1},...,X_{k.Hk}))$"[20] drücken die Restriktion aus, daß die Variable X_k aus dem Nachbereich der Übergangsoperation tr_v unter der jeweils ausgewählten Variablenbindungsfunktion vb_c denjenigen Wert annehmen muß, den die Bestimmungsfunktion "$bestimme_k$" unter derselben Variablenbindungsfunktion für alle ihre Variablen $X_{k.h}$ aus dem Einflußbereich derselben Übergangsoperation besitzt.

- Die "Übergangsprozedur" ist eine Komplexanweisung[21], die ausgeführt wird, um für jedes Prädikatssymbol $Prä_u$ aus der Menge PRÄ die Faktenmenge $FAK_{u.r}$, die vor der Ausführung der Übergangsoperation vorlag, in die vorläufige Folgefaktenmenge $FAK_{+.u}$ nach der Operationsausführung zu transformieren. Sie stellt den prozeduralen Kern jeder Übergangsoperation tr_v dar. Zugleich handelt es sich um *die* zentrale Konstituente jener operationalen Logik, die früher auf der Basis von Übergangsoperationen konzipiert wurde. Die Übergangsprozedur wird nur - und dann auch nur vorläufig - ausgeführt, wenn die drei vorgenannten Prä-, Inklusions- und Haupttests positiv ausgefallen sind.
- Die Anweisung[22] "Faktenentfernung" - oder kurz: Entfernungsanweisung - berechnet für jedes Prädikatssymbol $Prä_u$ aus dem Vorbereich VB_v der Übergangsoperation die Differenz aus seiner alten Faktenmenge $FAK_{u.r}$, die vor der Operationsausführung vorlag, und aus der Multimenge $vb_c(MTAV_{u.v})$ der konstanten Formelvorkommnisse unter der Variablenbindung vb_c. Das Berechnungsergebnis wird der intermediären Faktenmenge $FAK_{u.-}$ für das jeweils betrachtete Prädikatssymbol zugeordnet. Die Faktenmengen aller anderen Prädikatssymbole des Objektmodells bleiben unverändert.
- Die Anweisung "Faktenergänzung" - oder kurz: Ergänzungsanweisung - berechnet für jedes Prädikatssymbol $Prä_u$ aus dem Nachbereich NB_v der Übergangsoperation die Summe aus seiner intermediären Faktenmenge $FAK_{u.-}$, die nach der Ausführung der Entfernungsanweisung vorgelegen hat, und aus der Multimenge $vb_c(MTAN_{u.v})$ der konstanten Formelvorkommnisse unter der Variablenbindung vb_c. Das Berechnungsergebnis wird der vorläufigen Folgefaktenmenge $FAK_{u.+}$ für das jeweils betrachtete Prädikatssymbol zugeordnet. Die Faktenmengen aller anderen Prädikatssymbole des Objektmodells bleiben unverändert.
- Für jedes Prädikatssymbol $Prä_u$ aus dem Wirkungsbereich spezifizieren die Posttestprädikate "$posttest(FAK_{u.+})$" eine Testbedingung. Sie muß von der Faktenmenge $FAK_{u.+}$ *nach* der vorläufigen Ausführung der Übergangsoperation erfüllt werden, damit keine unzulässige Operationsausführung resultieren kann. Als implizite Voreinstellung für jedes nicht explizit definierte Posttestprädikat dient das immer gültige Posttestprädikat "T".
- Sofern alle Posttestbedingungen erfüllt werden, wird die komplexe Anweisung "Übergangsakt" ausgeführt. Dabei wird allen Prädikatssymbolen $Prä_u$ aus der Menge PRÄ des Objektmodells die zunächst vorläufige Folgefaktenmenge $FAK_{u.+}$ endgültig als Folgefaktenmenge $FAK_{u.f}$ zugeordnet. Zugleich wird das Übergangsprädikat "$übergang(vb_c(MTAO_v))$" als gültig ausgewiesen. Das Übergangsprädikat beschreibt, wie die jeweils ausgewählte Variablenbindungsfunktion vb_c die Operationsausführung durch eindeutige Festlegung der Variablen in allen atomaren Formelvorkommnissen der Multimengen $MTAV_{u.v}$, $MTAI_{u.v}$ und $MTAN_{u.v}$ aus dem Operationsbereich OP_v der Übergangsoperation tr_v geprägt ("gefärbt") hat. Die Variablenbindungsfunktion vb_c ist in diesem Fall zulässig.
- Falls sich mindestens eine der Posttestbedingungen verletzt ist, widerruft die universelle Anweisung "UNDO" die vorläufige Operationsausführung[23]. Dabei wird die vorläufig ausgeführte Übergangsprozedur zurückgesetzt, indem die alten Faktenmengen $FAK_{u.r}$, die vor Ausführen der Entfernungs- und Ergänzungsanweisungen vorlagen, für alle Prädikatssymbole $Prä_u$ aus der Menge PRÄ wiederhergestellt werden. Zugleich wird das Übergangsprädikat "$übergang(vb_c(MTAO_v))$" für die zunächst ausgewählte Variablenbindungsfunktion vb_c als ungültig ausgewiesen. Die Variablenbindungsfunktion vb_c ist in diesem Fall unzulässig.

Für die Formulierung der vorgenannten Prädikate und Anweisungen wird jede prädikatenlogisch ausdrückbare Formel zugelassen.

b) Die Prä-, Haupt- und Posttestbedingungen der Übergangsoperation sowie deren Bestimmungsgleichungen werden mit Hilfe der endlichen Formelmenge RES_v formuliert[24)]. Die Formelmenge RES_v besitzt genau jene Formeln, die zur Formulierung der vorgenannten Bedingungen und Bestimmungsgleichungen erforderlich sind. Sie ist eine Teilmenge der Menge RES aller Restriktionsformeln, die im Rahmen von algebraisch-prädikatenlogischen Spezifikationen eingeführt wurde. Falls keine Restriktionsformeln für die Übergangsoperation benötigt werden, gilt $RES_v = \varnothing$ als implizite Voreinstellung.

c) Die Gesamtheiten aller Prätestprädikate, Inklusionsungleichungen, Haupttestprädikate und Posttestprädikate werden als Prä-, Inklusions-, Haupt- bzw. Posttest bezeichnet. Prädikatssymbole, die zum Vor-, Informations-, Nach-, Einfluß-, Wirkungs- oder Operationsbereich einer Übergangsoperation gehören, heißen deren Eingangs-, Informations-, Ausgangs-, Einfluß-, Wirkungs- bzw. Operationsprädikate. Entsprechend werden die teilevaluierten Formeln, die den vorgenannten Prädikatssymbolen $Prä_u$ in den Multimengen $MTAV_{u.v}$, $MTAI_{u.v}$, $MTAN_{u.v}$, $MTAE_{u.v}$, $MTAW_{u.v}$ und $MTAO_{u.v}$ jeweils eindeutig zugeordnet sind, als Eingangs-, Informations-, Ausgangs-, Einfluß-, Wirkungs- bzw. Operationsformeln angesprochen.

d) Der Operationsbereich einer Übergangsoperation umfaßt alle Prädikatssymbole, welche die Operationsausführung beeinflussen oder durch die Operationsausführung beeinflußt werden. Zur ersten Kategorie gehören alle Prädikatssymbole, die Eingangs- oder Informationsprädikate aus dem Einflußbereich der Übergangsoperation darstellen. Die zweite Kategorie bilden alle Prädikatssymbole, die als Eingangs- oder Ausgangsprädikate zum Wirkungsbereich der Übergangsoperation zählen. Einfluß- und Wirkungsbereich können, müssen aber nicht disjunkt sein. Gleiches gilt a fortiori für die hierin eingeschlossen Vor- bzw. Nachbereiche der Übergangsoperation. Vor- und Nachbereich können auch zusammenfallen. Einfluß- und Wirkungsbereich dürfen dagegen nur dann übereinstimmen, wenn der Informationsbereich leer ist[25)].

e) KAF_u ist die Menge aller konstanten atomaren Formeln $prä_u(ob_1,...,ob_{Ku})$, die für ein Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ aus der Prädikatssymbolemenge PRÄ formiert werden können. $MKAF_{u.r}$ bezeichnet die Menge aller Multimengen $Mult_{KAFu.u}$, die sich über der Trägermenge KAF_u in binärer Mengennotation definieren lassen. $FAK_{u.x}$ ist eine beliebige Faktenmultimenge[26] mit $x \in \{r,+,-\}$. $MTA_{u.v}$ vertritt eine beliebige Multimenge aus atomaren Prädikatsvorkommnissen $prä_u(te_1,...,te_{Ku})$ mit $MTA_{u.v} \in \{MTAV_{u.v}, MTAI_{u.v}, MTAN_{u.v}, MTAE_{u.v}, MTAW_{u.v}, MTAO_{u.v}\}$. Für beide Multimengen wird eine binäre Mengennotation mit positiven Multiplizitäten $mu_{u.x}$ und $mu_{u.v}$ vorausgesetzt. Unter diesen Voraussetzungen sind die relationalen und operationalen Verknüpfungen[27)] zwischen einer Faktenmultimenge $FAK_{u.x}$ und einer Multimenge konstanter atomarer Prädikatsvorkommnisse $vb_c(MTA_{u.v})$ festgelegt durch:

$$FAK_{u.x} \geq vb_c(MTA_{u.v})$$

$$:\Leftrightarrow \quad \forall(mu_{u.x} \in \mathcal{N}_+)\ \forall(mu_{u.v} \in \mathcal{N}_+)\ \forall(prä_u(ob_1,...,ob_{Ku}) \in KAF_u): ...$$

$$(mu_{u.v}, prä_u(ob_1,...,ob_{Ku})) \in vb_c(MTA_{u.v})$$

$$\rightarrow (fakt_x(mu_{u.x}, prä_u(ob_1,...,ob_{Ku})) \in FAK_{u.x} \wedge mu_{u.x} \geq mu_{u.v})$$

$$FAK_{u.x} + vb_c(MTA_{u.v})$$
$$= \{fakt_+(mu_{u.+},prä_u(ob_1,...,ob_{Ku})): \exists(Mult_{KAFu.r} \in MKAF_{u.r}): ... (mu_{u.+},prä_u(ob_1,...,ob_{Ku})) \in Mult_{KAFu.r}\}$$

mit:

$$mu_{u.+} = \begin{cases} mu_{u.x} + mu_{u.v}; & \text{sofern } fakt_b(mu_{u.x},prä_u(ob_1,...,ob_{Ku})) \in FAK_{u.x} \\ & \wedge\ (mu_{u.v},prä_u(ob_1,...,ob_{Ku})) \in vb_c(MTA_{u.v}) \\ mu_{u.x}; & \text{sofern } fakt_x(mu_{u.x},prä_u(ob_1,...,ob_{Ku})) \in FAK_{u.x} \\ & \wedge\ (mu_{u.v},prä_u(ob_1,...,ob_{Ku})) \neq vb_c(MTA_{u.v}) \\ mu_{u.v}; & \text{sofern } fakt_x(mu_{u.x},prä_u(ob_1,...,ob_{Ku})) \neq FAK_{u.x} \\ & \wedge\ (mu_{u.v},prä_u(ob_1,...,ob_{Ku})) \in vb_c(MTA_{u.v}) \end{cases}$$

$$FAK_{u.x} - vb_c(MTA_{u.v})$$
$$= \{fakt_-(mu_{u.-},prä_u(ob_1,...,ob_{Ku})): \exists(Mult_{KAFu.r} \in MKAF_{u.r}): ... (mu_{u.-},prä_u(ob_1,...,ob_{Ku})) \in Mult_{KAFu.r}\};$$
$$\text{falls } FAK_{u.x} \geq vb_c(MTA_{u.v})$$

mit[28]:

$$mu_{u.-} = \begin{cases} mu_{u.x} - mu_{u.v}; & \text{sofern } fakt_x(mu_{u.x},prä_u(ob_1,...,ob_{Ku})) \in FAK_{u.x} \\ & \wedge\ (mu_{u.v},prä_u(ob_1,...,ob_{Ku})) \in vb_c(MTA_{u.v}) \\ mu_{u.x}; & \text{sofern } fakt_x(mu_{u.x},prä_u(ob_1,...,ob_{Ku})) \in FAK_{u.x} \\ & \wedge\ (mu_{u.v},prä_u(ob_1,...,ob_{Ku})) \neq vb_c(MTA_{u.v}) \end{cases}$$

Wegen $mu_{u.-} \in \mathcal{N}_+$ fallen alle "Fakten" $fakt_-(mu_{u.-},prä_u(ob_1,...,ob_{Ku}))$ mit $mu_{u.-} = mu_{u.x} - mu_{u.v} = 0$ aus der Faktenmenge der Differenz $FAK_{u.x} - vb_c(MTA_{u.v})$ heraus. Dies gewährleistet die Integrität des Faktenbegriffs. Denn eine konstante atomare Formel $prä_u(ob_1,...,ob_{Ku})$, für die wegen $mu_{u.-}=0$ überhaupt kein gültiges Formelvorkommnis bekannt ist, stellt per definitionem kein Fakt dar.

f) Die Variablenbindungsfunktion vb_c[29] braucht - trotz ihrer eminenten Rolle bei der Formulierung des Übergangsschemas - nicht im Detail betrachtet zu werden. Denn einerseits wurde sie früher im Kontext von algebraisch-prädikatenlogischen Spezifikationen als sortentreue, kanonisch erweiterte Komplementfunktion von Teilauswertungsfunktionen präzise eingeführt. Andererseits wird sie später in der PROLOG-gestützten Implementierung von Synthetischen Netzen durch das Software-Paket PASIPP automatisch realisiert. Dabei liefert das kombinierte Unifizierungs- und Resolutionskonzept der Programmiersprache PROLOG im Prinzip alle[30] sortentreuen Variablenbindungen, die in einem Übergangsschema ÜS zulässige Substitutionen der Variablen X aus der Variablenmengen $VA(MTA_{u.v})$ darstellen[31]. Daher brauchen die Variablenbindungsfunktionen überhaupt nicht mehr ermittelt zu werden. Folglich wird ihre konkrete Ermittlung hier nicht weiter vertieft. Dennoch waren die Variablenbindungsfunktionen erforderlich, um das theoretische, algebraisch und prädikatenlogisch ausgerichtete Fundament der später verwendeten PROLOG-Implementierung offenzulegen[32].

g) Ausgangsvariablen (Einflußvariablen) einer Übergangsoperation tr_v sind alle Variablen X einer Sorte $sort_i$, die in den teilevaluierten atomaren Formeln $prä_u(te_1,...,te_{Ku})$ aus Multimengen $MTAN_{u.v}$ ($MTAE_{u.v}$) für Prädikatssymbole $Prä_u$ des Nachbereichs NB_v (Einflußbereichs EB_v) der

Übergangsoperation tr_v vorkommen. Die Ausgangsvariablen können zunächst durch alle formalen Objekte aus ihren sortengerechten Definitionsbereichen OB_i gebunden werden. Dies führt im allgemeinen zu einer großen Vielfalt zulässiger Variablenbindungsfunktionen vb_c. Es wird hier eine zusätzliche Konzeptprämisse vereinbart, um diese hohe Varietät von Übergangsoperationen einzuschränken[33]. Jede Ausgangsvariable muß auf genau eine der beiden nachfolgenden Weisen an Einflußvariablen anschließen (Anschlußprämisse):

- ❑ Die Ausgangsvariable ist mit genau einer Einflußvariable identisch[34].
- ❑ Die Ausgangsvariable wird vermittels der Bestimmungsgleichung des Übergangsschemas durch eine endliche Anzahl von Eingangsvariablen unter einer gegebenen Variablenbindungsfunktion eindeutig determiniert.

Aufgrund der Anschlußprämisse sind die Werte aller Ausgangsvariablen eindeutig determiniert, sobald eine Variablenbindungsfunktion alle Einflußvariablen gebunden hat. Daher werden Variablenbindungsfunktionen fortan nur noch für die Einflußvariablen einer Übergangsoperation definiert. Die Variablenbindungen aller Ausgangsvariablen ergeben sich daraus mittelbar mit Hilfe der Anschlußprämisse.

h) Die Anwendung einer Übergangsoperation heißt erfolgreich[35], wenn ihre Übergangsprozedur vorläufig ausgeführt werden konnte und auch nicht nachträglich widerrufen werden mußte. Dadurch wird ein zulässiger Zustand des Objektmodells mit der Faktenmenge FAK_r in einen wiederum zulässigen Folgezustand mit der Faktenmenge FAK_f transformiert. Die erfolgreiche Prozedurausführung wird durch das Übergangsprädikat übergang($vb_c(MTAO_v)$) charakterisiert. Es weist die jeweils ausgewählte, zulässige Variablenbindungsfunktion vb_c aus. Zusätzlich gibt es für die Familie der Multimengen $MTAV_{u.v}$, $MTAI_{u.v}$ und $MTAN_{u.v}$ für alle Prädikatssymbole $Prä_u$ aus dem Operationsbereich OB_v der Übergangsoperation tr_v an, wie sie unter Anwendung der Variablenbindungsfunktion vb_c aus konstanten Formelvorkommnissen $prä_u(ob_1,...,ob_{Ku})$ zusammengesetzt wird[36].

i) Eine erfolglose Anwendung der Übergangsoperation[37] liegt dagegen immer dann vor, wenn die Übergangsprozedur entweder überhaupt nicht ausgeführt oder nachträglich widerrufen worden ist. Erstes geschieht, wenn mindestens einer der Prä-, Inklusions- oder Haupttests nicht erfüllt werden konnte. Zweites tritt im Fall mindestens eines unerfüllten Posttests ein. Ein erfolgloser Ausführungsversuch bedeutet, daß schließlich dieselbe zulässige Referenzfaktenmenge FAK_r vorliegt wie vor der Anwendung der Übergangsoperation.

j) Wenn eine Übergangsoperation tr_v mit Hilfe einer zulässigen Variablenbindungsfunktion vb_c erfolgreich ausgeführt werden kann und hierbei die Referenzfaktenmenge FAK_r in die Folgefaktenmenge FAK_f transformieren würde, so wird dies notiert als[38]:

$$FAK_r\,[tr_v,vb_c\rangle\,FAK_f$$

Das Paar (tr_v,vb_c) heißt eine Ausführung der Übergangsoperation tr_v im Modus vb_c[39]. Unter Berücksichtigung der involvierten Referenz- und Folgefaktenmenge FAK_r bzw. FAK_f wird auch von einem Schaltakt $sa_{r.v.c.f}$ gesprochen[40]. Er läßt sich mit der Notation $FAK_r[tr_v,vb_c\rangle FAK_f$ identifizieren. Der Index "c" des Übergangsmodus vb_c kann - unter Vorgriff auf die spätere Einbettung von Übergangsoperationen in das Petrinetz-Konzept[41] - als "Farbe" von Operationsausführung und Schaltakt bezeichnet werden.

k) Durch den Schaltakt $sa_{r.v.c.f}$ ist eindeutig festgelegt, welche Folgefaktenmenge FAK_f aus einer vorgelegten Referenzfaktenmenge FAK_r resultiert. Der Übergang zwischen zwei unmittelbar aufeinander folgenden Faktenmengen muß nicht unbedingt mit einer Veränderung der betroffenen Faktenmengen einhergehen. Daher wird die identische Transformation $FAK_r[tr_v,vb_c\rangle FAK_f$

mit $FAK_f = FAK_r$ als Übergangsoperation für $f \neq r$ grundsätzlich zugelassen. Dies ermöglicht später, durch das Ausführen von Übergangsoperationen reale Aktionen anzustoßen, ohne daß hierdurch der aktuelle Zustand des Objektmodells verändert würde[42].

l) Der Übergang von einer Referenz- zu einer Folgefaktenmenge läßt sich auch durch eine komplex strukturierte Übergangsfunktion $üf_{v.c}$ darstellen. Diese Übergangsfunktion entspricht genau dem Schaltakt $sa_{r.v.c.f} = FAK_r[tr_v, vb_c\rangle FAK_f$. Da die Zulässigkeit der Variablenbindungsfunktion vb_c von der Referenzfaktenmenge FAK_r abhängt, auf die sie angewendet wird, muß die Übergangsfunktion partiell definiert sein. Sie legt nur für solche Referenzfaktenmengen Bilder fest, für welche die Variablenbindungsfunktion vb_c aus dem Schaltakt $sa_{r.v.c.f}$ auch zulässig ist[43]. Daher gilt mit $FAK_r = (FAK_{r.1}, \ldots, FAK_{r.U})$ und $FAK_f = (FAK_{f.1}, \ldots, FAK_{f.U})$ für die Übergangsfunktion[44]:

$$
\begin{aligned}
üf_{v.c}\colon \quad & pot_r(PFAK_1) \times \ldots \times pot_r(PFAK_U) \rightarrow_{part} pot_f(PFAK_1) \times \ldots \times pot_f(PFAK_U) \\
& (FAK_{r.1}, \ldots, FAK_{r.U}) \rightarrow_{part} üf_{v.c}(FAK_{r.1}, \ldots, FAK_{r.U}) = (FAK_{f.1}, \ldots, FAK_{f.U})
\end{aligned}
$$

Dennoch wird die ursprünglich eingeführte Notation $FAK_r[tr_v, vb_c\rangle FAK_f$ aufgrund ihrer Kompaktheit bevorzugt. Darüber hinaus spricht für sie, daß sie die Gerichtetheit des operationsbedingten Übergangs von der Referenz- auf die Folgefaktenmenge durch ihr Infix "[...⟩" besonders deutlich hervortreten läßt.

m) Die Anschlußprämisse wurde oben nicht nur eingeführt, um die Varietät von Übergangsoperationen zu reduzieren. Vielmehr zielte sie inbesondere auch darauf ab, das Übergangsschema ÜS input-deterministisch auszugestalten[45]. Eine Operation[46] heißt genau dann input-deterministisch, wenn ihr Output bei gegebenem Input eindeutig fixiert ist. Der Output einer Operation, die das Übergangsschema ÜS erfüllt, ist die Folgefaktenmenge FAK_f, die aus einer Operationsausführung resultiert. Der Operationsinput erstreckt sich dagegen auf die Kombination aus einer vorliegenden Referenzfaktenmenge FAK_r und aus einer zulässigen Variablenbindungsfunktion vb_c. Hierdurch werden die Argumente aller Eingangsformeln einer Übergangsoperation tr_v eindeutig bestimmt. Dabei werden alle Einflußvariablen, die in den Argumenten der teilevaluierten atomaren Formeln aus den Multimengen $MTAE_{u.v}$ für Prädikatssymbole $Prä_u$ aus dem Einflußbereich EB_v enthalten sind, mit Hilfe der Variablenbindungsfunktion durch konkrete formale Objekte ersetzt. Diese Objekte müssen aufgrund des Inklusionstests aus den Argumenten derjenigen Fakten stammen, die zur Referenzfaktenmenge FAK_r gehören. Der Operationsoutput in Gestalt der Folgefaktenmenge steht per constructionem bei gegebener Referenzfaktenmenge und Variablenbindungsfunktion eindeutig fest. Dabei stellt die Anschlußprämisse sicher, daß die Ersetzung aller Ausgangsvariablen in den Argumenten von teilevaluierten atomaren Formeln für Prädikatssymbole des Nachbereichs durch die vorliegende Referenzfaktenmenge und die ausgewählte Variablenbindungsfunktion für alle Einflußvariablen eindeutig bestimmt ist. Folglich ist der Input-Determinismus jeder Übergangsoperation, die das Übergangsschema ÜS erfüllt, sichergestellt[47].

n) Eine Operation wird genau dann als zustands-deterministisch bezeichnet, wenn ihr Output für jeden Zustand desjenigen Systems, in dem die Operation zur Ausführung gelangen kann, eindeutig festliegt. Das Übergangsschema bezieht sich auf ein prädikatenlogisches Formelsystem, dessen aktuellen Zustand jeweils die Referenzfaktenmenge FAK_r ausdrückt. Eine Übergangsoperation tr_v wäre daher zustands-deterministisch, falls ihr Output in Gestalt der Folgefaktenmenge FAK_f schon bei gegebener Referenzfaktenmenge FAK_r eindeutig bestimmt wäre. Dies ist jedoch im allgemeinen nicht der Fall, weil die Ersetzung von Variablen in den Argumenten von teilevaluierten atomaren Formeln für Prädikatssymbole aus dem Operationsbereich der Übergangsoperation erst durch die Auswahl einer Variablenbindungsfunktion eindeutig fixiert ist. Im Regelfall existieren für eine Übergangsoperation unter derselben Referenzfaktenmenge jedoch mehrere zulässige Variablenbindungsfunktionen. Dies folgt aus der Offenheit der Funktionenfamilie TA_{SIG} einer Termauswertung, die allen Variablenbindungsfunktionen vb_c zugrundeliegt. Folglich verhält sich jede Übergangsoperation, die das Übergangsschema ÜS erfüllt, im allgemeinen nicht zustands-deterministisch.

o) Eine Übergangsoperation tr_v kann beliebig, aber höchstens endlich viele[48] Übergangsmodi vb_c besitzen. Wie viele und welche Übergangsmodi definiert sind, hängt von der jeweils aktuellen Referenzfaktenmenge FAK_r ab, auf welche die Übergangsoperation tr_v angewendet werden soll. Denn diese Referenzfaktenmenge determiniert, welche Variablenbindungsfunktionen vb_c angesichts der Prä-, Haupt- und Posttests sowie hinsichtlich des Bestimmungsgleichungen überhaupt zulässig sind. Dies schließt auch den Grenzfall ein, daß eine Übergangsoperation keinen Übergangsmodus besitzt. Das ist genau dann der Fall, wenn im aktuellen Modellzustand keine erfolgreiche Operationsausführung definiert ist, weil keine zulässige Variablenbindungsfunktion vb_c existiert. Die Anzahl aller Übergangsmodi vb_c, die eine Übergangsoperation tr_v bezüglich einer Referenzfaktenmenge FAK_r besitzt, wird als $C_{v.r}$ mit $C_{v.r} \in \mathcal{N}_0$ notiert. $MOD_{v.r} = \{vb_c: c = 1,...,C_{v.r}\}$ für $C_{v.r} \in \mathcal{N}_+$ und $MOD_{v.r} = \varnothing$ für $C_{v.r} = 0$ sind die Mengen aller Übergangsmodi, in denen eine Übergangsoperation tr_v auf eine Faktenmenge FAK_r zulässig angewendet werden kann.

p) Falls für eine Übergangsoperation tr_v wegen $C_{v.r} \geq 2$ mehrere Variablenbindungsfunktionen vb_c bezüglich derselben Referenzfaktenmenge FAK_r zulässig sind, kann die Übergangsoperation in mehreren Modi ausgeführt werden. Es besteht keine Auszeichnung eines der korrespondierenden Schaltakte $sa_{r.v.c.f}$. Um die Übergangsoperation tatsächlich ausführen zu können, muß aber genau ein Übergangsmodus durch die Auswahl von genau einer Variablenbindungsfunktion vb_c realisiert werden. Diese Modus- bzw. Funktionsauswahl bleibt im Übergangsschema ÜS grundsätzlich offen; sie erfolgt willkürlich. Für diese Auswahl ist auch keine Wahrscheinlichkeitsverteilung über der Gesamtheit aller zulässigen Variablenbindungsfunktionen definiert. Daher verhalten sich Übergangsoperationen, die das Übergangsschema erfüllen, in bezug auf die Zustände prädikatenlogischer Formelsysteme nicht nur nicht-deterministisch, sondern im allgemeinen auch nicht-stochastisch.

q) Das Übergangsschema zeichnet sich dadurch aus, daß es Übergangsoperationen spezifiziert, die einerseits input-deterministisch ausfallen. Andererseits verhalten sich die Übergangsoperationen im allgemeinen weder zustands-deterministisch noch zustands-stochastisch. Diese Operationscharakteristik wird fortan wird unter dem Begriff der semi-deterministischen Operationsweise subsumiert.

r) Nur in Sonderfällen verhält sich eine Übergangsoperation zustands-deterministisch. Sie erstrecken sich auf diejenigen seltenen Konstellationen, in denen eine Übergangsoperation tr_v nur in genau einem Modus vb_c ausgeführt werden kann. Dies ist erstens der Fall, wenn die Operationsdefinition zwar noch Einflußformeln mit Variablen in ihren Argumenten enthält, aber ihr Haupttest derart eng formuliert ist, daß er nur noch durch genau eine Variablenbindungsfunktion vb_c für alle Einflußvariablen erfüllt wird. Zweitens können alle teilevaluierten atomaren Formeln aus den Multimengen $MTAV_{u.v}$, $MTAI_{u.v}$ und $MTAN_{u.v}$ für Prädikatssymbole $Prä_u$ ihres Operationsbereichs OB_v variablenfreie Grundtermformeln darstellen. Dann verfügt die betroffene Übergangsoperation tr_v nur noch über genau einen Operationsmodus $tr_{v.c} = (tr_v, id)$ mit $c = C_{v.r} = 1$. Das Spektrum denkmöglicher Variablenbindungsfunktionen reduziert sich auf genau eine Funktion: die identische Abbildung $vb_c = id$[49].

s) Ein Übergangsschema heißt global definiert, wenn sein Operationsbereich alle Prädikatssymbole umfaßt, die in einem prädikatenlogischen Objektmodell definiert sind. Ein lokales Übergangsschema liegt dagegen vor, wenn sein Operationsbereich eine echte Teilmenge der Menge aller Prädikatssymbole des Objektmodells darstellt. Falls ein Übergangsschema lokal definiert ist, wird sein Operationsbereich auch als Nachbarschaft bezeichnet. Das Petrinetz-Konzept beruht grundsätzlich auf einem lokal definierten Übergangsschema. Dies schließt allerdings nicht aus, daß sich jedes Petrinetz in ein äquivalentes Petrinetz mit einem globalen Übergangsschema transformieren läßt[50]. Da diese Globalität aber der konzeptionellen Lokalität von Petrinetzen widerspricht, wird ihre formale Denkmöglichkeit fortan ausgeschlossen.

t) Ein Übergangsschema ist dadurch charakterisiert, Veränderungen von Faktenmengen als Übergänge $FAK_r[tr_v,vb_c\rangle FAK_f$ zu definieren. Kommt die Lokalität des Übergangsschemas hinzu, so muß eine nicht-leere Schnittmenge $FAK_r \cap FAK_f$ von Fakten existieren, die von der Operationsausführung nicht betroffen sind. Es handelt sich genau um diejenigen Fakten, deren zugrundeliegenden Prädikatssymbole nicht zum Wirkungsbereich der Übergangsoperation gehören. Das Übergangsschema berücksichtigt auch diese Prädikatssymbole. Denn die o.a. Konstruktion von Entfernungs- und Ergänzungsanweisung stellt mit Hilfe der beiden intermediären Faktenmengen $FAT_{u.-}$ und $FAK_{u.+}$ sicher, daß alle unveränderten Fakten aus der Referenz- in die Folgefaktenmenge übernommen werden. Zu diesem Zweck wurden die beiden Anweisungen jeweils für alle Prädikatssymbole $Prä_u$ aus der Prädikatssymbolemenge PRÄ des zugrundeliegenden Objektmodells definiert[51].

u) Die Variablen des Übergangsschemas sind lokal definiert[52]. Dies bedeutet einerseits, daß alle Vorkommnisse derselben Variablen[53] innerhalb *derselben* Übergangsoperation durch eine Variablenbindungsfunktion jeweils auf dasselbe formale Objekt abgebildet werden müssen[54]. Andererseits kann dieselbe Variable in *unterschiedlichen* Übergangsoperationen, für die jeweils eigene Variablenbindungsfunktionen angesetzt werden, durch verschiedene formale Objekte ersetzt werden[55]. Diese Lokalität der Variablendefinition erweist sich aus der Perspektive der praktischen Gestaltung von prädikatenlogischen Objektmodellen als hilfreich. Denn es braucht nicht auf mögliche Fernwirkungen[56] geachtet zu werden, die von Vorkommnissen derselben Variable in unterschiedlichen Übergangsoperationen ausgehen könnten[57]. Darüber hinaus stimmt die Lokalität der Variablendefinition mit der Behandlung von Variablenvorkommnissen in der Programmiersprache PROLOG überein[58]. Daher läßt sich später das Übergangsschema ohne Schwierigkeiten durch ein PROLOG-Klauselschema implementieren.

v) Referenz- und Folgefaktenmenge werden immer als endliche Mengen vorausgesetzt. Im Normalfall sind beide Mengen nicht leer. Als Grenzfälle läßt das Übergangsschema jedoch auch Übergangsoperationen mit leerer Referenz- oder leerer Folgefaktenmenge zu.

w) Inkonsistente Formulierungen von Prä-, Haupt- oder Posttests würden dazu führen, daß für die zugehörige Übergangsoperation niemals eine zulässige Variablenbindungsfunktion gefunden werden kann. Eine solche Übergangsoperation, deren Ausführung grundsätzlich unmöglich ist, widerspräche jedoch der inhaltlichen Festlegung von Übergangsoperationen, Veränderungen bewirken zu können. Daher werden inkonsistente Operationsformulierungen a priori ausgeschlossen.

x) Der Wirkungsbereich einer Übergangsoperation darf nicht leer sein[59]. Dies schließt allerdings nicht aus, daß entweder der Vor- oder aber der Nachbereich einer Operation leer ist. Eine Übergangsoperation heißt ein Generator, wenn ihr Vorbereich leer ist. Sie kann u.a. auf eine leere Referenzfaktenmenge angewendet werden und hierbei eine nicht-leere Folgefaktenmenge erzeugen. Ein Reduktor liegt dagegen vor, wenn der Nachbereich einer Übergangsoperation leer ist. Sie läßt sich u.a. anwenden, um den Umfang von nicht-leeren Referenzfaktenmengen zu verringern. Im Extremfall können die Referenzfaktenmengen sogar in die leere Folgefaktenmenge transformiert werden.

y) Eine elementare Operationswirkung liegt entweder vor, wenn aus der Referenzfaktenmenge FAK_r ein Fakt $fakt_r(mu_{u.r}, prä_u(ob_1,...,ob_{Ku}))$ ersatzlos entfernt wird, so daß ein korrespondierendes Fakt $fakt_f(mu_{u.f}, prä_u(ob_1,...,ob_{Ku}))$ mit $mu_{u.f}>0$ in der Folgefaktenmenge FAK_f nicht enthalten ist. Oder es wird in die Folgefaktenmenge FAK_f ein neues Fakt $fakt_f(mu_{u.f}, prä_u(ob_1,...,ob_{Ku}))$ aufgenommen, bezüglich dessen kein korrespondierendes Fakt $fakt_r(mu_{u.r}, prä_u(ob_1,...,ob_{Ku}))$ mit $mu_{u.r}>0$ in der Referenzfaktenmenge FAK_r vorhanden war.

z) Eine einfache Operationswirkung besteht in der Ersetzung eines alten Fakts $fakt_r(mu_{u.r}, prä_u(ob_{1.r},...,ob_{Ku.r}))$ aus der Referenzfaktenmenge FAK_r durch ein neues Fakt $fakt_f(mu_{u.f}, prä_u(ob_{1.f},...,ob_{Ku.f}))$ aus der Folgefaktenmenge FAK_f in der Weise, daß beide Fakten gültige Vorkommnisse desselben Prädikatssymbols $Prä_u(st_1,...,st_{Ku})$ darstellen. Dabei kann sich die elementare Veränderung auf zwei grundsätzlich verschiedenen Ebenen abspielen[60]. Entweder unterscheiden sich altes und neues Fakt nur in ihren - stets positiv unterstellten[61] - Multiplizitäten $mu_{u.r}$ bzw. $mu_{u.f}$ ($mu_{u.r} \neq mu_{u.f}$). Beide Fakten gelten für dieselbe konstante atomare Formel $prä_u(ob_{1.r},...,ob_{Ku.r}) = prä_u(ob_{1.f},...,ob_{Ku.f}) = prä_u(ob_1,...,ob_{Ku})$. Hierdurch wird lediglich die Anzahl der identischen Vorkommnisse derselben Formel $prä_u(ob_1,...,ob_{Ku})$ verändert. Oder aber altes und neues Fakt stimmen zwar in ihren Multiplizitäten $mu_{u.r}$ bzw. $mu_{u.f}$ überein ($mu_{u.r} = mu_{u.f}$). Aber sie erstrecken sich auf konstante atomare Formeln mit verschiedenen Argumenten: $prä_u(ob_{1.r},...,ob_{Ku.r}) \neq prä_u(ob_{1.f},...,ob_{Ku.f})$. Solche einfachen Operationswirkungen lassen sich nur dadurch berücksichtigen, daß das betroffene Prädikatssymbol $Prä_u(st_1,...,st_{Ku})$ zugleich Ein- und Ausgangsprädikat der Operation ist. Daher wurde im Übergangsschema die Überlappung von Vor- und Nachbereich zugelassen.

A) Der Informationsbereich IB_v einer Übergangsoperation tr_v kann leer sein. Falls er nicht leer ist, steuern seine Informationsprädikate zwei wesentliche Beiträge zur Beeinflussung der Operationswirkung bei. Sie betreffen die Formulierung von kontextabhängigen Übergangsoperationen und die Variablenbindung durch formale Objekte.

B) Zunächst lassen sich Informationsprädikate benutzen, um in den Bestimmungsgleichungen "$vb_c(X_k) = vb_c(bestimme_k(X_{k.1},...,X_{k.Hk}))$" mit $X_{k.h} \in VA(MTAE_{u.v})$ für alle $h \in \{1,...,H_k\}$ die Substitution einer Variablen X_k aus einem Formelvorkommnis $präq(te_1,...,te_{Kq})$ des Nachbereichs NB_v der Übergangsoperation tr_v durch Variablenbindungsfunktionen vb_c zu beeinflussen. Hierbei spielen solche Informationsprädikate $Prä_u(st_1,...,st_{Ku})$ aus dem Informationsbereich IB_v EB_v eine Rolle, zu denen Formelvorkommnisse $prä_u(te_1,...,te_{Ku})$ mit mindestens einer Variable X_h in ihren Argumenten gehören. Über den Inklusionstest für Informationsprädikate werden die Variablen X_h dieser Prädikatsvorkommnisse $prä_u(te_1,...,te_{Ku})$ an die korrespondierenden formalen Objekte derjenigen Fakten gebunden, welche zur aktuellen Faktenmenge der betrachteten Informationsprädikate gehören und den Inklusionstest erfüllen (Informationsfakten). Auf diese Weise wird das Spektrum zulässiger Variablenbindungsfunktionen vb_c auf solche eingeschränkt, die sich mit den formalen Objekten in den Argumenten aller involvierten Informationsfakten vereinbaren lassen. Daher können Informationsprädikate durch die Informationsfakten aus ihren aktuellen Faktenmengen das Ergebnis einer Operationsausführung beeinflussen. Die Informationsfakten, die eine Operation durch ihre Bestimmungsgleichung zwecks Bindung ihrer Variablen X_k konsultiert, werden dabei grundsätzlich unverändert in den aktuellen Faktenmengen der involvierten Informationsprädikate belassen. Daher wird die Verfügbarkeit der jeweils benutzen Informationsfakten für andere Operationen nicht beeinträchtigt[62]. Zugleich wird auf diese Weise verhindert, daß während der Operationsausführung die Existenz der konsultierten Informationsfakten unklar bleibt[63]. Ebenso werden unerwünschte Fernwirkungen[64] auf die Ausführungen anderer Operationen ausgeschlossen.

C) Darüber hinaus bieten Informationsprädikate die Gelegenheit, Übergangsoperationen kontextabhängig zu definieren. Zu diesem Zweck dienen die Prätests für Informationsprädikate, die eine Operationsausführung nur dann zulassen, wenn charakteristische Informationsmuster über den aktuellen Zustand des Objektmodells vorliegen[65]. Diese Informationsmuster können jeweils als ein operationsspezifischer Anwendungskontext aufgefaßt werden. Dadurch läßt sich erreichen, daß Operationen solange unbeachtet bleiben, wie ihr spezifischer Anwendungskontext nicht vorliegt[66]. Auf diese Weise können umfangreiche Operationenmengen durch Einführung von Anwendungskontexten übersichtlich strukturiert[67] werden[68].

D) Die Kontextabhängigkeit von Operationsdefinitionen schließt auch den Grenzfall ein, in dem eine Übergangsoperation nur dann ausgeführt werden darf, wenn keine Informationen über die Gültigkeit von bestimmten, jeweils operationsspezifisch festgelegten Prädikatssymbolen aus dem Objektmodell vorliegen. Diese Möglichkeit wird später genutzt, um für Synthetische Netze Inhibitorkanten einzuführen[69], ohne das Übergangsschema selbst grundsätzlich erweitern zu müssen. Hierdurch wird die Ausdrucksmächtigkeit Synthetischer Netze auf das Modellierungspotential von TURING-Automaten ausgedehnt. Es handelt sich keineswegs um eine triviale Konzeptmodifizierung. Vielmehr resultieren gravierende Konsequenzen für die Modellierungsgüte[70] und -effizienz[71]. Daher kommt der Definition von Informationsprädikaten im allgemeinen Übergangsschema erhebliche Bedeutung für die spätere Beurteilung des Modellierungsbeitrags Synthetischer Netze zu.

E) Ein weiterer bemerkenswerter Spezialfall der Kontextabhängigkeit von Operationsdefinitionen erlaubt die "natürliche"[72] Formulierung von Nebenbedingungen[73]. Eine Nebenbedingung besteht aus zwei Komponenten: einem nebenbedingungsspezifischen Prädikatssymbol (Nebenbedingungsprädikat) und einer Vorschrift für die Faktenmenge dieses Prädikatssymbols (Nebenbedingungsvorschrift). Die Nebenbedingung ist genau dann erfüllt, wenn die aktuelle Faktenmenge des Nebenbedingungsprädikats der Nebenbedingungsvorschrift gerecht wird. Nebenbedingungen beeinflussen Übergangsoperationen derart, daß die ersten erfüllt sein müssen, damit die zweiten überhaupt ausgeführt werden dürfen (Geltungsaspekt). Darüber hinaus dürfen die Nebenbedingungen durch die Operationsausführung nicht verändert werden (Invarianzaspekt)[74].

Für die Erfassung einer Nebenbedingung als Anwendungskontext einer Übergangsoperation wird zunächst das Nebenbedingungsprädikat durch ein Informationsprädikat repräsentiert. Dem Invarianzaspekt wird dieses Informationsprädikat von vornherein gerecht, weil es grundsätzlich nicht an den veränderungsbewirkenden Entfernungs- und Ergänzungsanweisungen des Übergangsschemas beteiligt ist[75]. Zusätzlich wird im Übergangsschema für das nebenbedingungsspezifische Informationsprädikat ein Prätest eingerichtet. Dieser Prätest wird so formuliert, daß er von der aktuellen Faktenmenge des Informationsprädikats genau dann erfüllt wird, wenn dieselbe Faktenmenge auch die Nebenbedingungsvorschrift erfüllt. Der Prätest fällt daher inhaltlich mit der Nebenbedingungsvorschrift zusammen. Hierdurch wird der Geltungsaspekt unmittelbar realisiert. Dabei kann sich der Prätest sowohl auf die Forderung nach einer bestimmten - minimalen, maximalen oder fixen - Fakten*anzahl*[76] als auch auf die inhaltliche Auszeichnung einer bestimmten Fakten*menge* erstrecken.

F) Durch die "UNDO"-Komponente[77] können Übergangsoperationen die Integritätsbedingungen eines Objektmodells berücksichtigen. Denn sie veranlaßt, die hypothetische Ausführung einer Übergangsoperation zurückzusetzen, falls sie dazu führen würde, die im Posttest ausgedrückten Integritätsbedingungen zu verletzen. Dabei wird stets vorausgesetzt, daß der Modellzustand, der zu Beginn der hypothetischen Operationsausführung vorlag, alle Integritätsbedingungen erfüllt hat[78]. Unter dieser Voraussetzung besitzen Übergangsoperationen integritätswahrenden Charakter. Sie werden entweder vollständig ausgeführt, falls sie einen integren Zustand des Objektmodells in einen ebenso integren Folgezustand transformieren. Oder sie werden im Falle drohender Integritätsverletzung überhaupt nicht ausgeführt. Eine dritte Möglichkeit existiert nicht[79].

G) Allerdings werden Übergangsoperationen in dieser Arbeit nur mit einer relativen Integritätssicherungsfähigkeit ausgestattet. Sie können nur die Einhaltung jener Integritätsbedingungen sicherstellen, die in ihren Posttests explizit ausgedrückt sind[80]. Dazu gehören beispielsweise beim Petrinetz-Konzept die Anforderungen, daß das Schalten einer Transition nicht dazu führen darf, die Mindest- oder Höchstkapazitäten ihrer Ein- und Ausgangsstellen zu unter- bzw. zu überschreiten. Diese Berücksichtigung von Kapazitätsrestriktionen spielt auch beim transaktionsorientierten Entwurf von Informationsverarbeitungssystemen für die Koordinierung von betrieblichen Produktionsprozessen eine große Rolle[81]. Praktische Erfahrungen bei der Gestaltung komplexer Systeme legen jedoch nahe, daß sich nicht alle Integritätsbedingungen, die für ein Objektmodell formuliert werden können, als Posttests von Übergangsoperationen formulieren lassen[82]. Daher ist zwecks Integritätswahrung ein weiteres Konzept vorzuhalten. Es soll die Rückführung in einen zulässigen Systemzustand sicherstellen, falls eine Übergangsoperation in einen unzulässigen Folgezustand geführt hat, ohne daß dies anhand ihres Posttests erkannt worden wäre. Für die Integritätswiederherstellung kommt das Triggerkonzept in Betracht.

H) Das Triggerkonzept[83] kann benutzt werden, um die absolute Integrität eines Objektmodells sicherzustellen[84]. Unter "absoluter" Integrität wird dabei die Erfüllung *aller* Integritätsbedingungen verstanden, die für ein Objektmodell zur Auszeichnung seiner zulässigen oder für den Ausschluß seiner unzulässigen Modellzustände formuliert worden sind[85]. Wenn eine Integritätsbedingung durch eine prädikatenlogische Formel $ib_w(te_1,...,te_{Kw})$ ausgedrückt wird, so wird ihre Verletzung durch die Gültigkeit ihres Negats $\neg ib_w(te_1,...,te_{Kw})$ angezeigt. Daher läßt sich eine Triggerinformation "verletzt(ib_w)", die eine Integritätsverletzung ausdrückt, stets in der Gestalt eines Subjugats in ein prädikatenlogisches Modell integrieren:

$$\neg ib_w(te_1,...,te_{Kw}) \rightarrow \text{verletzt}(ib_w)$$

Zusätzlich wird vorausgesetzt, daß weder das Negat $\neg ib_w(te_1,...,te_{Kw})$ in der Faktenmenge FAK_0 unmittelbar enthalten ist noch aus ihr mittelbar abgeleitet werden kann. Dadurch wird ein ab initio integritätswidriges Objektmodell ausgeschlossen[86]. Die Triggerinformation "verletzt(ib_w)" läßt sich im einfachsten Fall dazu benutzen, einem Modellbenutzer anzuzeigen, daß eine Integritätsverletzung eingetreten ist[87]. Dabei liegt aber noch keine wesentliche Erweiterung gegenüber dem bereits eingeführten Prädikat "verletzt(ib_w)" vor. Allenfalls kann die Meldung an den Modellbenutzer so gestaltet werden, daß die prädikatenlogische Formel "verletzt(ib_w)" in eine besonders benutzerfreundliche Darstellungsweise transformiert wird. Gehaltreicher sind dagegen solche Konstruktionen, die entweder im Falle einer Integritätsverletzung einen integren Modellzustand wiederherstellen[88] oder von vornherein das Entstehen von integritätsverletzenden Modellzuständen verhindern können[89]. Die beiden letzten Varianten des "Integritätsmanagements"[90] werden bei der absoluten Integritätswahrung angestrebt. Bei der späteren Modellierung von Flexiblen Fertigungssystemen werden solche integritätswahrenden Konstruktionen für den Fall von Betriebsunterbrechungen exemplarisch behandelt.

I) Jede Übergangsoperation wird als ein punktförmiger, atomarer Veränderungsakt konzipiert. Sie besitzt daher die epistemische Qualität eines zeit- und strukturlosen Ereignisses. Die Beschaffenheit des Objektmodells ist *während* der Ausführung einer Übergangsoperation grundsätzlich nicht definiert. Die Referenz- und die Folgefaktenmenge FAK_r bzw. FAK_f, die durch die Übergangsoperationen eines Übergangsschemas miteinander verknüpft werden, repräsentieren daher immer unmittelbar aufeinander folgende Zustände des Objektmodells.

J) Übergangsoperationen besitzen diskursiven Charakter. Denn die Übergänge zwischen zwei Faktenmengen sind jeweils als schrittweises Ausführen der operationskonstituierenden Tests und Aktionen definiert. Daher zeichnen sich Übergangsoperationen intern durch eine sequentielle Ablaufstruktur aus. Dies widerspricht nicht der Ereignishaftigkeit von Übergangsoperationen.

Denn eine Übergangsoperation erweist sich nur aus dem Blickwinkel der Zustände eines Objektmodells, zwischen denen sie vermittelt, als ein atomarer Übergang. Dies schließt keineswegs aus, daß auf einer detaillierteren Betrachtungsebene die Übergangsoperation selbst aus verschiedenen Komponenten zusammengesetzt wird[91)].

K) Trotz ihrer internen Sequentialität können Übergangsoperationen benutzt werden, um in Systemen nebenläufige Prozesse zu realisieren. Diese Eigenschaft wird später für die Definition nebenläufiger Schaltakte von Transitionen in Synthetischen Netzen genutzt.

L) Übergangsoperationen entsprechen aus informationstechnischer Sicht dem Konzept der transaktionsorientierten Informationsverarbeitung[92)]: Jede Übergangsoperation läßt sich als eine Transaktion auf der Faktenmenge eines Objektmodells auffassen. Die Übereinstimmung mit dem Transaktionskonzept erstreckt sich im wesentlichen auf fünf Punkte: Erstens gelten sowohl Übergangsoperationen als auch Transaktionen als atomare[93)] zustandstransformierende[94)] Akte, während derer Ausführung der Zustand des transformierten Systems nicht definiert ist. Zweitens entspricht die "UNDO"-Komponente von Übergangsoperationen der recovery-Fähigkeit von Transaktionen[95)]. Beide dienen zwecks Integritätswahrung[96)] dazu, den ursprünglichen zulässigen Systemzustand wiederherzustellen, falls die Operations- bzw. Transaktionsausführung durch Verletzung mindestens einer Integritätsbedingung in einen unzulässigen Systemzustand geführt hätte und dies durch einen Posttest erkannt wurde[97)]. Drittens werden auch seitens der transaktionsorientierten Informationsverarbeitung komplexere Mechanismen[98)] vorgehalten, um Integritätsverletzungen aufzuheben, die im Rahmen einer einzelnen Transaktionsausführung nicht behandelt werden konnten. Dies entspricht den speziellen Übergangsoperationen zur Wiederherstellung der absoluten Modellintegrität. Fünftens besitzen Übergangsoperationen und Transaktionen intern eine sequentielle Ablaufstruktur[99)], werden aber oftmals genutzt, um nebenläufige Prozesse in verteilten Systemen zu realisieren[100)]. Viertens können sowohl Übergangsoperationen als auch Transaktionen auch im Fall ihrer erfolgreichen Ausführung zu demjenigen Systemzustand zurückführen, auf den sie ursprünglich angewendet worden sind[101)].

M) Hinzu kommt eine Übereinstimmung mit manchen speziellen Transaktionskonzepten, die sich dadurch auszeichnen, Transaktionen durch die Geltung von Vor- und Nachbedingungen zu charakterisieren[102)]. Ihnen entsprechen die Vor- bzw. Nachbereiche der Übergangsoperationen einschließlich der zugehörigen Prädikatssymbole und Multimengen aus teilevaluierten Formeln[103)]. Schließlich läßt sich das Triggerkonzept der transaktionsorientierten Informationsverarbeitung auch mit der Hilfe von Übergangsoperationen verwirklichen. Allerdings unterscheiden sich Übergangsoperationen und Transaktionen auch in einem wesentlichen Aspekt: Erste wurden als zeitlose Ereignisse definiert; zweite dagegen im allgemeinen als zeitverbrauchende Vorgänge konzipiert[104)].

N) Die weitgehende Übereinstimmung von Übergangsoperationen mit informationstechnischen Transaktionen ist insofern beachtlich, als das Konzept der tansaktionsorientierten Informationsverarbeitung - einschließlich des Triggerkonzepts - neuerdings auch zunehmende Beachtung für die Koordinierung von Produktionsprozessen in Fertigungssystemen findet[105)]. Dabei stehen solche Prozeßkoordinierungsaufgaben im Vordergrund, die nicht stets in derselben Weise erfüllt werden können, sondern in Abhängigkeit vom jeweils aktuellen Systemzustand variieren[106)]. Genau diese Charakteristik erfüllt aber auch die hier interessierende Koordinierung von Maschinenbelegungen in Flexiblen Fertigungssystemen. Daneben wird dem transaktionsorientierten Informationsverarbeitungskonzept auch aus der Perspektive einer Datenflußsteuerung von Produktionsprozessen Aufmerksamkeit zuteil[107)], bei der die Ausführung einzelner Arbeitsgänge durch das Eintreten koordinierungsrelevanter Ereignisse angestoßen wird. Dies entspricht einer Variante des oben angesprochenen Triggerkonzepts, die nicht mehr an Integritätsverletzungen, sondern an Ereignisgeschehnissen ausgerichtet ist[108)].

O) Die Formulierung des Übergangsschemas ÜS bedeutet einen qualitativen Sprung von der deklarativen Semantik, die der Prädikatenlogik zugrundeliegt[109)], zu einer operationalen Semantik. Die hier gewählte operationale Semantik besitzt prozeduralen Charakter[110)]. Denn die auszuführenden Operationen wurden mit der Hilfe von Übergangsprozeduren formuliert[111)]. Der prozedurale Kern des Übergangsschemas ÜS äußert sich in den Anweisungsoperatoren ":=", die sich im Rahmen der Prädikatenlogik nicht rein deklarativ erklären lassen[112)].

P) Die Ausführung einer Übergangsoperation wird durch die Transformation der Referenz- in eine Folgefaktenmenge eindeutig und vollständig beschrieben. Eine andere als die hierdurch spezifizierte Veränderung der aktuellen Faktenmenge bewirkt die Operationsausführung grundsätzlich nicht[113)]. Die Faktenmengen entsprechen inhaltlich den Extensionen für alle Prädikatssymbole eines prädikatenlogischen Formelsystems. Daher konstituiert das Übergangsschema eine operationale Semantik auf rein extensionaler Basis. Dies genügt der früher dargelegten Präferenz extensionaler Modellierungskonzepte.

Q) Das Übergangsschema ÜS ist die abstrakte Beschreibung aller konkreten Übergangsoperationen tr_v, deren Ausführungen jeweils einen schemaspezifischen Übergang von der Faktenmenge FAK_r zur Faktenmenge FAK_f mit $f \neq r$ bewirken. Alle Übergangsoperationen, die das Übergangsschema erfüllen, heißen zulässige Übergangsoperationen. Sie werden vereinfacht auch als Operationen bezeichnet[114)]. Anstatt das abstrakte Übergangsschema anzusprechen, kann auch auf diejenigen konkreten Übergangsoperationen Bezug genommen werden, die durch dieses Schema definiert werden.

R) Mit "abstrakt(...)" wird der Abstraktionsoperator für schematische Beschreibungen notiert. Dabei wird erstens von den konkreten Prädikats- und Anweisungsformulierungen einer Übergangsoperation tr_v abgesehen. Zweitens wird die Auswahl einer Variablenbindungsfunktion vb_c vernachlässigt, die notwendig ist, um die Folgefaktenmenge bei der Anwendung einer Übergangsoperation tr_v auf eine Referenzfaktenmenge eindeutig zu determinieren. Unter diesen Voraussetzungen ergibt sich als abstrakte Charakterisierung des hier entfalteten Übergangsschemas[115)]:

$$\text{ÜS} = \text{abstrakt}(\{(tr_v, vb_c) \colon FAK_r [tr_v, vb_c\rangle FAK_f\})$$

S) Umgekehrt kann mit "konkret(...)" als Konkretisierungsoperator jede Übergangsoperation tr_v aufgefaßt werden als ein Element aus der Bildmenge konkret(ÜS), die aus der Operatoranwendung auf das Übergangsschema ÜS resultiert: $tr_v \in$ konkret(ÜS). Sie ist durch die Konkretisierung allerdings nicht eindeutig determiniert, sondern bleibt immer noch gegenüber der Kombination mit einer zulässigen Variablenbindungsfunktion vb_c offen.

T) Jede zulässige Übergangsoperation tr_v ist eine Konkretisierung des Übergangsschemas ÜS. Eine Konkretisierung erfolgt erstens dadurch, daß alle Prä-, Haupt- und Posttests des Schemas ÜS jeweils entweder auf ihre impliziten Voreinstellungen gesetzt oder aber durch explizite Testbedingungen eindeutig bestimmt werden[116)]. Zweitens ist es erforderlich, den Vor-, Informations- und Nachbereich durch die Angabe aller Eingangs-, Informations- bzw. Ausgangsprädikate eindeutig festzulegen. Drittens gilt es, die Multimengen teilevaluierter atomarer Formeln eindeutig zu determinieren, welche mit den Eingangs-, Informations- bzw. Ausgangsprädikaten korrespondieren. Viertens müssen alle Ausgangsvariablen, die in den vorgenannten Multimengen vorkommen, durch Bestimmungsgleichungen gebunden wurden. Diese Konkretisierungsleistungen werden fortan als materieller Gehalt der formalen Konkretisierungsoperation "konkret" vorausgesetzt.

U) Übergangsoperationen, die das Übergangsschema ÜS konkretisieren, brauchen nicht alle Komponenten dieses Schemas zu umfassen. Die Varietät der Konkretisierungsoptionen wird konstituiert durch folgende Festlegungen:

- Jede Operation muß mindestens entweder eine Entfernungs- oder aber eine Ergänzungsanweisung besitzen[117)].
- Eine Operation kann eine Bestimmungsgleichung nur dann enthalten, wenn sie einen nicht-leeren Einfluß- und einen ebenso nicht-leeren Nachbereich besitzt[118)].
- Falls eine Operation eine Entfernungsanweisung enthält, dann muß zu ihr auch ein Inklusionstest für alle Prädikatssymbole ihres Vorbereichs vorhanden sein. Ein Inklusionstest für Eingangsprädikate ohne zugehörige Entfernungsanweisung ist unzulässig.
- Wenn eine Operation eine Bestimmungsgleichung enthält, dann muß sie auch einen Inklusionstest für alle Prädikatssymbole ihres Informationsbereichs besitzen. Ein Inklusionstest für Informationsprädikate ohne zugehörige Bestimmungsgleichung ist durchaus zulässig[119)].
- Explizite Prä-, Haupt- und Posttests sind fakultativ. Wenn sie nicht aufgeführt werden, gelten ihre impliziten Voreinstellungen.
- Falls eine Operation einen Haupttest mit Bezug auf Eingangsprädikate enthält, dann muß sie auch eine Entfernungsanweisung besitzen.
- Sofern eine Operation einen Haupttest mit Bezug auf Informationsprädikate umfaßt, dann muß sie auch eine Bestimmungsgleichung oder einen Inklusionstest[120)] für Informationsprädikate aufweisen.
- Wenn eine Operation einen Posttest mit Bezug auf Ein- oder Ausgangsprädikate umfaßt, dann muß sie für alle involvierten Ein- bzw. Ausgangsprädikate je eine Entfernungs- bzw. Ergänzungsanweisung aufweisen.

Den Normalfall stellen Übergangsoperationen dar, die jeweils sowohl mindestens eine Entfernungs- als auch mindestens eine Ergänzungsanweisung besitzen. Solche Operationen bewirken Veränderungen von Faktenmengen, bei denen die Eliminierung alter Fakten und die Erzeugung neuer Fakten miteinander kombiniert werden[121)].

V) Der schematische Charakter[122)] des Konstrukts ÜS geht daraus hervor, daß die Konkretisierungsoperation keine gewöhnliche rechtseindeutige, sondern eine rechtsmehrdeutige Abbildung darstellt. Ihrem Argument - dem Übergangsschema ÜS - ordnet sie im allgemeinen eine Vielzahl unterschiedlicher Konkretisierungen zu. Jede von ihnen stellt eine Übergangsoperation tr_v dar. Die Übergangsoperationen lassen sich durch die konkrete Ausformulierung ihrer Prädikate und Anweisungen beliebig voneinander unterscheiden. Dadurch können die verschiedenen Übergangsoperationen tr_v, wenn sie jeweils zusammen mit einer zulässigen Variablenbindung vb_c auf dieselbe Referenzfaktenmenge FAK_r angewendet werden, zu unterschiedlichen Folgefaktenmengen FAK_f führen.

W) Übergangsoperationen, die das Übergangsschema ÜS konkretisieren, stellen eine Verallgemeinerung und Komplizierung der Operationen op aus SIG-Algebren dar. Die algebraischen Operationen sind als rechtseindeutige Abbildungen (Funktionen) definiert, die jedem Argument höchstens ein Bild zuordnen. Im Regelfall handelt es sich um vollständig definierte Funktionen, die jedes Argument auf genau einen Funktionswert abbilden. Übergangsoperationen stellen dagegen im allgemeinen um rechtsmehrdeutige[123)] und nur partiell definierte Abbildungen dar. Denn dieselbe Übergangsoperation tr_v bildet einerseits dasselbe Argument - die Referenzfaktenmenge FAK_r - auf verschiedene Folgemarkierungen FAK_f ab. Diese hängen jeweils davon ab, in welchem Modus $vb_{v.c}$ die Übergangsoperation tr_v ausgeführt wird. Andererseits muß eine Übergangsoperation keineswegs unter allen kombinatorisch möglichen Referenzfaktenmengen FAK_r ausführbar sein. Vielmehr ist die Operationsausführung für alle Referenzfaktenmengen nicht definiert, in denen mindestens einer ihrer Prä-, Inklusions-, Haupt- oder Posttests verletzt wird. Darüber hinaus werden die relativ einfach strukturierten Abbildungsvorschriften der algebraischen Operationen durch das komplexe Test-, Bestimmungsgleichungs- und Anweisungsgefüge des Übergangsschemas abgelöst.

X) Das Übergangsschema ist so allgemein formuliert, daß es alle bisher in der Literatur zum Petrinetz-Konzept beschriebenen Transitionen - einschließlich der später vorgestellten Transitionen von Synthetischen Netzen - als zulässige Übergangsoperationen einschließt.

Y) Darüber hinaus besitzt das Übergangsschema einen formalen Gehaltsüberschuß, der von den bisher vorgelegten, etablierten Transitionsbeschreibungen für Petrinetze nicht in Anspruch genommen wird. Er gewährt einen Entwicklungsspielraum für neuartige Transitionsdefinitionen in Petrinetzen. Dazu gehört z.B. die Möglichkeit, in den Posttestbedingungen für Prädikatssymbole $Prä_u$ aus dem Vorbereich VB_v einer Übergangsoperation tr_v zu prüfen, ob die Folgefaktenmengen $FAK_{u.f}$ jeweils eine Mindestanzahl von gültigen konstanten Formelvorkommnissen aufweisen. Auf diese Weise können Mindestkapazitäten für die Stellen eines Petrinetzes eingeführt werden, die in konventionell definierten Petrinetzen nicht bekannt sind[124)].

Z) Durch das Übergangsschema werden keine impliziten Annahmen über die Implementierung von Übergangsoperationen getroffen. Der Verf. setzt als normative Basisprämisse voraus, daß formale Konzepte grundsätzlich unabhängig von möglichen konkreten Konzeptimplementierungen definiert sein sollen[125)]. Daher sieht er in der implementierungsinvarianten Formulierung des Übergangsschemas einen wesentlichen Vorzug seiner abstrakten Herleitung von Synthetischen Netzen aus Prädikatenlogik und Signaturkonzept. Hierdurch unterscheidet es sich von den üblichen Definitionsschemata für die Transitionen von Prädikat/Transition-Netzen, die sich im allgemeinen - zumindest implizit - an Vorstellungen über die konkreten Implementierungsmöglichkeiten von Netzen ausrichten[126)].

Anmerkungen zum Kapitel:

1) Darüber hinaus erlaubt die SIG-Dynamik, die hervorgehobene Stellung der Ausgangsmarkierung von Petrinetzen in das Signaturkonzept einzubetten.

2) Die Schaltregel wird zwar definiert, aber nicht in der formalen Netzdefinition selbst. Statt dessen wird sie gewöhnlich in einem "Annex" nachgereicht, in dem formal- und natürlichsprachliche Formulierungen vermengt werden; vgl. z.B. REISIG (1986a), S. 71 u. 151f. Das Verhältnis dieses Nachtrags zur Netzdefinition bleibt zumeist im Dunkeln. Es ist nicht klar zu erkennen, ob es sich "bloß" um eine Erläuterung der Netzdefinition, um einen nachgeholten Bestandteil der Netzdefinition, um ein selbständiges Konstrukt neben der Netzkonstruktion oder um ein aliud handelt.

3) Vgl. KRÄMER (1987a), S. 278f.; RECK (1988), S. 81; REISIG (1989a), S. 10f.

4) Das Symbol "tr" für eine Übergangsoperation stellt strenggenommen den Namen der Übergangsoperation dar. Diese Differenzierung zwischen einer Operation und ihrem Namen spielt hier jedoch noch keine Rolle, weil die interne Struktur von Übergangsoperationen nicht weiter ausdifferenziert wird. Bei der späteren Konkretisierung von Übergangsoperationen durch Transaktionen gilt dies allerdings nicht mehr. Dort wird zwischen dem Namen "tr_v" einer Transaktion und der Spezifizierung der internen Transaktionsstruktur unterschieden.
Die Verwendung des Symbols "tr" kann hier als mnemotechnische Anlehnung an den *tr*ansformierenden Operationscharakter verstanden werden. Zugleich verweist sie auf die enge inhaltliche Verwandtschaft der Übergangsoperationen mit der *transaktionsorientierten* Informationsverarbeitung. Auf weite Sicht wird jedoch bereits auf den *Tr*ansitionsbegriff des Petrinetz-Konzepts Bezug genommen. Denn das Übergangsschema wird später im Kontext Synthetischer Netze als Schaltregel für Transitionen interpretiert. Darüber hinaus wird im Symbol "t*r*" auf den schematischen Charakter der hier vorgelegten allgemeinen Definition von Übergangsoperationen angespielt. Er besteht darin, daß diese Definition im allgemeinen keine konkrete Operation, sondern ein abstraktes Konzept darstellt, das durch mehrere gleichartige Operationen vollständig konkretisiert werden kann. Für diese Operationen wird durch das Operationsschema eine gemeinsame *R*egel für die Operationsausführung spezifiziert.

5) Vgl. dazu die Einführung prädikatenlogischer Objektmodelle.

6) Die gesamten nachfolgenden Ausführungen dieser Arbeit - also auch dieses Kapitels - erstrecken sich auf Faktenmultimengen. Infolge der Schwerfälligkeit dieser Bezeichnungsweise und aufgrund der Möglichkeit, Multimengen auch als (binäre) Mengen zu notieren, werden sie fortan der Einfachheit halber auch kurz als Faktenmengen angesprochen.

7) Vgl. dazu die Einführung von Faktenmengen und ihre Fortentwicklung zu zeitlich variablen Faktenmengen.

8) Diese Notationsweise wird in diesem Kapitel auch für alle weiteren Multimengen präsupponiert.

9) Diese als gültig bekannten Vorkommnisse von konstanten interpretierten atomaren Formeln werden auch kurz als Prädikatsvorkommnisse bezeichnet.

10) Fakten stellen bereits metasprachliches Wissen über die Gültigkeit objektsprachlicher Formeln dar. Daher bedeutet Wissen über die Transformationsmöglichkeit von Mengen solcher metasprachlichen Fakten seinerseits metametasprachliches Wissen. Der Einfachheit halber wird es fortan auch nur als metasprachliches Wissen angesprochen.
Der meta-metasprachliche Charakter von Übergangsoperationen verweist auf die Präsupposition von metasprachlichem Wissen: Eine notwendige Bedingung der Möglichkeit ihrer Anwendung liegt darin, daß überhaupt eine Faktenmenge FAK_T über aktuell gültige Formeln eines prädikatenlogischen Objektmodells vorliegt. Die Quelle *dieses* metasprachlichen Wissens über Prädikatsgültigkeiten wird durch ein Übergangsschema nicht offengelegt. Wenn es möglich sein soll, ein Übergangsschema anwenden zu können, dann muß dieses metasprachliche Wissen schemaunabhängig eingeführt worden sein. Genau dies leistet die ausgezeichnete Faktenmenge FAK_0, die seitens der algebraisch-prädikatenlogischen Spezifikation $SPEC_{PSIG}$ festgelegt wird. Eine solche Ausgangsfaktenmenge FAK_0 wird hier stets als bekannt vorausgesetzt. Später wird sie in das Konzept Synthetischer Netze als korrespondierende Ausgangsmarkierung M_0 aufgenommen.

11) Das Konzept zeitvariabler Faktenmengen wird vorausgesetzt, um Wissensveränderungen als zeitliche Variationen von Formelgültigkeiten erfassen zu können.

12) Daher entsprechen die hier betrachteten Übergangsoperationen der Konzeption von "Übergangsregeln", wie sie im Rahmen des Situationskalküls der KI-Forschung verfolgt wird. Vgl. zur prädikatenlogischen Formulierung solcher Übergangsregeln FIDELAK (1988b), S. 10f. (dort wird auch der metasprachliche Regelcharakter durch die Verwendung des metasprachlichen Wahrheitsoperators "T" besonders deutlich).

13) Die Bezeichnungen Grund- und Ausgangszustand werden synonym verwendet.

14) Auch diese totale Unbekanntheit der Gültigkeit objektsprachlicher Formeln stellt wohldefiniertes metasprachliches Wissen dar.

15) Da hier nur das Ergebnis, aber nicht der Prozeß der Variablenbindung interessiert, braucht keine sortierte Variablenfamilie betrachtet zu werden.

16) Pseudocodes stellen semi-formalsprachliche Notationen dar. Sie bestehen jeweils aus einer codespezifischen, nicht-leeren und endlichen Menge von wohldefinierten formalsprachlichen Konstrukten, die durch reservierte Ausdrücke gekennzeichnet werden. Die reservierten Ausdrücke werden mit Großbuchstaben notiert und erklären sich durch ihre mnemotechnisch Formulierung selbst. Beispielsweise drücken nachfolgend die reservierten Ausdrücke "DO" und "IF" aus, daß eine nicht-leere Liste von Anweisungen ausgeführt bzw. eine Fallunterscheidung vorgenommen werden soll. Es werden jeweils ein einleitender und ein abschließender reservierter Ausdruck so miteinander kombiniert, daß sie ein Codemodul mit einer wohldefinierten Funktion einschließen. So kennzeichnet z.B. das Ausdruckspaar "DO ... ENDDO" Beginn und Ende eines Moduls, in dem auszuführende Anweisungen aufgelistet werden. Falls es sich um mehrere Anweisungen handelt wird vorausgesetzt, daß sie in sequentieller Weise nacheinander ausgeführt werden. Innerhalb der Codemodule kann ein beliebiger Text eingefügt werden, der die Modulfunktion näher spezifiziert. Dieser Text kann sowohl formal- als auch natürlichsprachlicher Art sein. Falls er formalsprachliche Qualität besitzt, wird er nicht durch die zugrundeliegende Pseudocode-Notation selbst festgelegt. Statt dessen muß er im Kontext einer Pseudocode-Anwendung definiert sein. Die formalsprachlichen Texte, die im nachfolgenden Übergangsschema zwischen die reservierten Ausdrücke der Pseudocode-Notation eingefügt sind, wurden überwiegend in den voranstehenden Kapiteln zusammen mit der Entfaltung des Konzepts Synthetischer Netze eingeführt. Spezielle formalsprachliche Konstrukte - wie der Anweisungsoperator ":=" - können auch in der Erläuterung einer Pseudocode-Notation festgelegt werden.
Vgl. zu vielfältig variierenden Vorschlägen für Pseudocode-Notationen, die der Verf. im Groben übernommen, im Detail jedoch an seine Bedürfnisse angepaßt hat, JORDAN (1978), S. 82ff.; JACKSON,M.A. (1979), S. 30ff.; HOROWITZ (1981), S. 5ff., insbesondere S. 7ff.; SOWA (1984), S. 398ff.; vgl. auch die Verwendung einer ausdrucksstärkeren Pseudocode-Notation in dieser Arbeit.

17) Unter einer Voreinstellung (default value) wird die eindeutige a priori-Festlegung eines Wertes für eine Größe verstanden, der so lange gilt, wie die Größe nicht durch eine andere Wertfestlegung a posteriori bestimmt wird.
Voreinstellungen spielen sowohl eine theoretisch als auch eine praktisch wichtige Rolle. Ihre praktische Bedeutung resultiert aus dem Umstand, daß sie die Formulierung von Übergangsoperationen, die das allgemeine Übergangsschema konkretisieren, erheblich vereinfachen. Denn die Voreinstellungen "vererben" sich automatisch in alle Übergangsoperationen, die aus einem Schema abgeleitet worden sind, solange die Voreinstellungen nicht widerrufen und durch andere Werte ersetzt werden. Dieser Vererbungsmechanismus wird seitens des objektorientierten Programmierstils - z.B. bei Verwendung von semantischen Rahmen (frames) oder bei Einsatz der Programmiersprache SMALLTALK - intensiv genutzt; vgl. z.B. WAND (1989), S. 550 u. 555; VON ZIMMERMANN (1990), S. 243ff. Der theoretische Aspekt von Voreinstellungen besteht darin, daß alle "typischen" Konkretisierungen des Übergangsschemas genau als diejenigen Übergangsoperationen auszeichnen, in denen alle Voreinstellungen unverändert übernommen werden. Dabei wird auf die frühere Charakterisierung von Schemata als generische Kennzeichnungen für Klassen von ähnlichen Objekten zurückgegriffen. Denn im Kontext generischer Kennzeichnungen läßt sich die typkonstituierende Funktion von Voreinstellungen präzise entfalten; vgl. HEYER (1987), S. 210f.

18) Vgl. dazu die Definition der Vergleichsrelation "≥" für Multimengen.

19) Es brauchen keineswegs alle Variablen $X_{k.h}$ aus den Mengen $VA(MTAE_{u.v})$ im Argument der Bestimmungsfunktionen $bestimme_k(X_{k.1},...,X_{k.Hk})$ enthalten zu sein. Es gilt daher nur die Teilmengenbeziehung: $\{X_{k.1},...,X_{k.Hk}\} \subseteq (\cup(u \in IEB_v): VA(MTAE_{u.v}))$ mit $H_k \leq \#(\cup(u \in IEB_v): VA(MTAE_{u.v}))$. Mit Hilfe des früher eingeführten besonderen Mengenbildungsoperators "::" kann dieser Sachverhalt auch vereinfacht durch $bestimme_k(X_{k.h}:: X_{k.h} \in VA(MTAE_{u.v}) \wedge u \in IEB_v))$ ausgedrückt werden. Variablen aus den Mengen $VA(MTAE_{u.v})$, welche die Bestimmungsfunktion nicht beeinflussen, werden im Argument der Bestimmungsfunktion nicht angeführt.

20) An früherer Stelle wurde die Vereinfachung eingeführt, daß zwei Terme genau dann als gleich behandelt werden dürfen, wenn sie unter derselben Termauswertung auf die gleichen formalen Objekte abgebildet werden: $te_1 = te_2 :\Leftrightarrow eval_i(te_1) = eval_i(te_2)$. Den Auswertungsfunktionen "$eval_i$" aus dem Signaturkonzept sowie den daraus abgeleiteten Teilauswertungs- und Komplementfunktionen entsprechen in der algebraisch erweiterten Prädikatenlogik die Variablenbindungsfunktionen vb_c. Daher lassen sich unter Abstraktion von der prädikatenlogischen Variablenbindung die Bestimmungsgleichungen auch kompakter als $X_k = bestimme_k(X_{k.1},...,X_{k.H})$ notieren. Diese verkürzte Schreibweise wird später für die Beschriftung der Transitionen von Synthetischen Netzen verwendet werden. Dabei kommt den Bestimmungsgleichungen oftmals die Aufgabe zu, der abhängigen Variablen X_k ein Wert in Abhängigkeit von den unabhängigen Variablen $X_{k.1},...,X_{k.H}$ zuzuweisen. Falls dieser operative Gleichungscharakter hervorgehoben werden soll, können die Bestimmungsgleichungen mit Hilfe des Anweisungsoperators ":=" ebenso als $X_k := bestimme_k(X_{k.1},...,X_{k.H})$ dargestellt werden. Es handelt sich nur um eine modifizierte Notationsweise, die hinsichtlich der jeweils zulässigen Variablenbindungen keine Veränderungen bewirkt.

21) Prozeduren werden hier als lineare Folgen von Anweisungen aufgefaßt. Jede Anweisung symbolisiert einen Befehl an eine Automatische Informationsverarbeitungsanlage, der von dieser Anlage unmittelbar ausgeführt werden kann (einfache Anweisung). Eine Prozedur kann aus genau einer Anweisung bestehen. Wenn eine Prozedur mehrere Anweisungen umfaßt (Komplexanweisung), werden sie - entsprechend ihrer horizontalen oder vertikalen Notation - von links nach rechts bzw. von oben nach unten abgearbeitet. Hierin liegt der lineare oder sequentielle Charakter der hier vorausgesetzten Prozeduren. (Auf nicht-lineare, nebenläufige Prozeduren wird in dieser Arbeit an anderer Stelle eingegangen).

22) Eine Anweisung "Anw" beruht auf der imperativen Semantik eines real implementierten Automatischen Informationsverarbeitungssystems. Sie spezifiziert einen Befehl, ihrem konkret spezifizierten Anweisungsinput einen eindeutigen Anweisungsoutput zuzuordnen. Die Zuordnungsvorschrift stellt eine anweisungsspezifische Funktion "anw" dar, welche den Anweisungsinput auf den zugehörigen Anweisungsoutput abbildet. Mit Hilfe des Anweisungsoperators ":=" in Infixnotation gilt:

Anw: <Anweisungsoutput> := anw(<Anweisungsinput>)

Der Anweisungsoperator ":=" symbolisiert, daß dem linksstehenden, noch unbekannten Ausdruck die Ausprägung des rechtsstehenden, bereits bekannten Ausdrucks zugeordnet werden soll.

23) Vgl. zu informationstechnischen Details der Ausgestaltung solcher UNDO-Anweisungen ALAGIC (1986), S. 201f.

24) Allerdings besteht eine notationelle Diskrepanz: Die Bedingungs- und Gleichungsformeln aus der Deklaration des Übergangsschemas sind stets mit Hilfe einer Variablenbelegung vb_c ausgedrückt, die für jeweils einen Übergangsmodus charakteristisch ist. Die Elemente aus der Menge RES_v werden dagegen als prädikatenlogische Formeln stets ohne solche Variablenbelegungen vb_c ausgedrückt. Daher werden jede Formel $for_z(te_1,...,te_{Kz})$ aus der Menge RES_v und jeder darin enthaltene teilevaluierte Term te_k in der korrespondierenden Formel aus dem Übergangsschema ÜS durch die Applikation der Variablenbelegung vb_c auf die Formel bzw. auf den Term derart ersetzt, daß die Formelnotation des Übergangsschemas resultiert.
Beispielsweise wird die Bestimmungsgleichung $vb_c(X_k)=vb_c(bestimme_k(X_{k.h}:: X_{k.h}\in VA(MATEu.v) \wedge u\in IEB_v))$ aus dem Übergangsschema in einer ihrer denkmöglichen Konkretisierungen für eine Übergangsoperation tr_v betrachtet: $vb_c(X_1)=vb_c(3\bullet X_2+2\bullet X_3)$ mit $X_1\in VA(MTAN_{1.v})$, $X_2\in VA(MTAE_{2.v})$, $X_3\in VA(MTAE_{3.v})$, "1"$\in INB_v$, "2"$\in IEB_v$, "3"$\in IEB_v$ und $bestimme_1(X_2,X_3):\Leftrightarrow 3\bullet X_2+2\bullet X_3$. Mit der konkreten Bestimmungsgleichung $vb_c(X_1)= vb_c(3\bullet X_2+2\bullet X_3)$ des Übergangsschemas korrespondiert in der Formelmenge RES_v die prädikatenlogische Formel $X_1=3X_2+2X_3$. Dabei wird die vereinfachte, aber prädikatenlogisch übliche Notationsweise für Gleichungen genutzt, die von Variablenbindungsfunktionen abstrahiert.

25) Denn es wurde Disjunktheit von Wirkungs- und Informationsbereich vorausgesetzt; zugleich ist der Informationsbereich ein Teilbereich des Einflußbereichs.

26) Faktenmultimengen werden in dieser Arbeit der Einfachheit halber auch als Faktenmengen bezeichnet, wenn aus dem Kontext ersichtlich ist, daß es sich um Multimengen handelt und die Differenzierung zwischen Multimengen und Mengen aktuell keine Rolle spielt.

27) Diese Verknüpfungen wurden im o.a. Übergangsschema bereits vorausgesetzt.

28) Durch das Übergangsschema ist sichergestellt, daß die Differenz $FAK_{u.x}-vb_c(MTA_{u.v})$ nur dann berechnet werden kann, falls $FAK_{u.x}\geq vb_c(MTA_{u.v})$ erfüllt ist. Daher braucht die Definition der Differenz $FAK_{u.x}-vb_c(MTA_{u.v})$ hier keine besondere Vorkehrungen für Komplikationen treffen, die sonst aus negativen Multiplizitäten $mu_{u.-}<0$ resultieren könnten. Denn wegen $FAK_{u.x}\geq vb_c(MTA_{u.v})$ ist stets $mu_{u.x}\geq mu_{u.v}$ für alle $mu_{u.x}-mu_{u.v}$ gewährleistet. Des weiteren braucht der Fall $(mu_{u.v},prä_u(ob_1,...,ob_{Ku}))\in vb_c(MTA_{u.v})$ und $fakt_x(mu_{u.x},prä_u(ob_1,...,ob_{Ku}))\neq FAK_{u.x}$ für die Differenz $FAK_{u.x}-vb_c(MTA_{u.v})$ nicht berücksichtigt zu werden, weil die Voraussetzung $FAK_{u.x}\geq vb_c(MTA_{u.v})$ die Existenzmöglichkeit eines solchen Falls ausschließt.

29) Der Index "c" wird in mnemotechnischer Anlehnung an die Farben ("*c*olours") von Höheren Netzen, insbesondere von Gefärbten Netzen, gebildet. Dort können Transitionen - als Pendants der hier thematisierten Übergangsoperationen - in unterschiedlichen Modi geschaltet werden, die jeweils durch eine Schaltfarbe gekennzeichnet sind. Darauf wird anläßlich der Definition von Schaltakten zurückgekommen. Vgl. auch die entsprechenden Ausführungen zu den Schaltfarben von Synthetischen Netzen.

30) Dies gilt allerdings nur unter der einschränkenden Voraussetzung, daß die besondere operationale (prozedurale) Semantik von PROLOG-Programmen und exogen vorgegebene Beschränkungen für maximale Suchtiefen oder für maximale Ressourcenverbräuche nicht dazu führen, tatsächlich existierende zulässige Variablenbelegungen zu übersehen. Solche "pathologischen" Fälle dürfen aber nicht dem Unifizierungskonzept im allgemeinen zugerechnet werden, sondern resultieren aus seiner besonderen PROLOG-Implementierung in Verbindung mit realen Beschränkun-

gen für Prozedurausführungen in Automatischen Informationsverarbeitungssystemen. Auf diese Schwierigkeiten - wie z.B. das Verfangen in Endlosschleifen trotz der Existenz zulässiger Unifizierungen - wurde bereits hingewiesen.

31) Vgl. dazu die Anmerkungen zur Konstruktivität des PROLOG-Inferenzkonzepts.

32) Deswegen bedeutet die o.a. Realisierung der Variablenbindungsfunktionen durch das kombinierte Unifizierungs- und Resolutionskonzept der Programmiersprache PROLOG keine implementierungsabhängige Konstitution des Übergangsschemas. Statt dessen wurde oben zunächst für das Übergangsschema die Zulässigkeit von Variablenbindungsfunktionen mit Bezug auf SIG-Termauswertungen und hieraus abgeleitete Teilauswertungsfunktionen in allgemeiner, implementierungsunabhängiger Weise definiert. Erst *nachdem* das Schema allgemein eingeführt worden ist, wird es später auf PROLOG-Basis in einer vereinfachten, aber äquivalenten Reformulierung implementiert.

33) Diese Vereinbarung liegt auch allen vom Verf. untersuchten Petrinetzen vom Typ der Prädikat/Transition-Netze zugrunde. Allerdings wird sie dort nicht offengelegt, sondern nur implizit angewandt.

34) Diese Identität kann auf zwei Weisen sichergestellt werden. Erstens lassen sich Ausgangs- und Einflußvariable zwar durch Symbole "X_a" bzw. "X_e" mit $a \neq e$ unterscheiden, aber durch eine Bestimmungsgleichung $vb_c(X_a) = vb_c(X_e)$ miteinander identifizieren. Dabei reduziert sich die Bestimmungsfunktion "$\text{bestimme}_k(X_{h.k}: \ldots)$" in der allgemeinen Formulierung des o.a. Übergangsschemas ÜS auf die identische Abbildung "id": $\text{bestimme}_a(X_e) = \text{id}(X_e) = X_e$ mit der einen ($h = H_k = 1$) unabhängigen Variablen $X_{k.1} = X_e$. Dann gilt die triviale Bestimmungsgleichung $vb_c(X_a) = vb_c(X_e)$ oder - in vereinfachter Notation - $X_a = X_e$. Zweitens können Ausgangs- und Einflußvariable von vornherein durch dasselbe Symbol "X" notiert werden. Der Verf. bevorzugt den zweiten Ansatz aufgrund seiner Kompaktheit.

35) In diesem Fall wird auch von einer erfolgreichen Operationsausführung gesprochen.

36) Da das Konstrukt $MTAO_v$ als *Familie* - und nicht als Vereinigungsmenge - aller Multimengen $MTA_{u.v}$ aus dem Operationsbereich einer Übergangsoperation definiert wurde, enthält es notwendig *alle* Informationen über die Variablenbindungen in diesen Multimengen durch die jeweils angewandte Variablenbindungsfunktion vb_c. Dies ist insofern wichtig festzustellen, als Vor- und Nachbereich derselben Übergangsoperation nicht disjunkt zu sein brauchen. Daher könnte die Verwendung der Vereinigungsmenge zu Informationsverlusten bei der Aggregation der Multimengen $MTA_{u.v}$ führen.

37) Dann wird auch von einem erfolglosen Ausführungsversuch geredet.

38) Die allgemeine dreistellige Übergangsformel "$ob_r[op\rangle ob_f$" drückt die Ausführung einer beliebigen formalen Operation "op" auf einem beliebigen formalen Objekt "ob_r" aus, das sich im Referenzzustand "r" befindet. Es wird durch die Operationsausführung in seinen Folgezustand "f" transformiert.

39) Der Modus vb_c wird auch verdeutlichend als Übergangsmodus bezeichnet.

40) Vgl. dazu die analoge Definition von Schaltakten $sa_{r.a.f}$ für Synthetische Netze. Dort wird lediglich das Paar (tr_v, vb_c) zu einem Schaltschritt SS_a zusammengefaßt, so daß in der Schaltschrittnotation die Teilindizes "v.c" durch den Teilindex "a" ersetzt werden. Darüber hinaus wird die Aktivierung des Schaltschritts in die Schaltaktdefinition einbezogen. Dies ist hier für Übergangsoperationen jedoch nicht möglich, weil für sie das Aktivierungskonzept nicht spezifiziert wurde.

41) Vgl. dazu die Erläuterung zu den Farben ("colours") von Höheren Netzen.

42) Dadurch lassen sich z.B. im Falle von Integritätsverletzungen Meldungen an einen Modellbenutzer ausgeben. Vgl. dazu die Anmerkungen zu Integritätsverletzungen. Vgl. ebenso die Erweiterung Synthetischer Netze durch Transitionen, die sowohl von netzexternen Aktionen beeinflußt werden können als auch solche Aktionen auszulösen vermögen.

43) Dies wird später im Petrinetz-Konzept mit Hilfe der Aktivierungsbedingung explizit berücksichtigt.

44) Dabei werden die Referenz- und die Folgefaktenmenge FAK_r bzw. FAK_f, die ursprünglich als Familien oder Vereinigungsmengen von prädikatspezifischen Faktenmultimengen definiert sind, hier als U-stellige Tupel aus den prädikatspezifischen Faktenmultimengen aufgefaßt. Diese geringfügige formale Darstellungsvariation besitzt keine inhaltliche Relevanz. Denn zwischen Familien und Tupeln von Faktenmultimengen besteht ohnehin kein nennenswerter Unterschied. Da die prädikatspezifischen Faktenmultimengen disjunkt sind, fallen die Familien und Tupel auch mit der Vereinigungsmenge dieser Faktenmultimengen zusammen.

45) An späterer Stelle werden Erweiterungen des Grundkonzepts Synthetischer Netze diskutiert, die zu einem stochastischen Übergangsschema führen. Es wird dort allerdings auch aufgezeigt, daß die stochastische "Bereicherung" zu wesentlichen Abstrichen an einigen zentralen Aspekten des Petrinetz-Konzepts führen. Daher behandelt der Verf. sie nur am Rande.

46) Damit ist eine beliebige Operation allgemeinster Art gemeint, die keineswegs das hier definierte Übergangsschema erfüllen muß.

47) Wäre die Anschlußprämisse nicht erfüllt, so könnten mehrere verschiedene Variablenbindungsfunktionen dieselbe Substitution aller Einflußvariablen durch formale Objekte realisieren. Diese Funktionen würden sich noch durch ihre verschiedenen Bindungen der Ausgangsvariablen unterscheiden. Dann wäre die Übergangsoperation nicht input-deterministisch.
An späterer Stelle werden Erweiterungen des Grundkonzepts Synthetischer Netze diskutiert, die sich nicht mehr input-deterministisch verhalten. Es wird dort allerdings auch aufgezeigt, daß diese stochastischen "Bereicherungen" zu wesentlichen Abstrichen an einigen zentralen Aspekten des Petrinetz-Konzepts führen. Daher behandelt der Verf. sie nur am Rande.

48) Die Mengen formaler Objekte OB_i wurden als endliche Mengen vorausgesetzt. Variablenbelegungen bestehen aus der Substitution von Variablen der Sorten $sort_i$ durch formale Mengen aus den zugehörigen Mengen OB_i. Daher können nur endlich viele unterschiedliche Kombinationen für denkmögliche Variablenbelegungen gebildet werden.

49) Die Variablenbindungsfunktionen wurden als kanonisch erweiterte Komplementfunktionen keval eingeführt. Diese Komplementfunktionen sind nicht nur für Tupel aus Variablen, sondern für alle teilevaluierten Termtupel definiert. Daher ist es berechtigt, auch bei variablenfreien Argumenten teilevaluierter Formeln noch von der Anwendung einer Variablenbindungsfunktion zu sprechen. In einem solchen Fall werden alle vorhandenen - d.h. gar keine - Variablen durch die Funktionsapplikation gebunden.

50) Dies läßt sich dadurch erreichen, daß alle Transitionen eines Petrinetzes durch genau eine Transition mit einer komplexen Schaltregel substituiert werden. Diese Transition umfaßt alle Stellen des Petrinetzes in ihrem Operationsbereich. Das oben eingeführte Übergangsschema ist so gehaltreich, daß es die Möglichkeit eröffnet, Schaltregeln für Transitionen beliebig so komplex zu formulieren, daß sie das gesamte Schaltverhalten eines Petrinetzes hervorbringen. Vgl. dazu auch die Überlagerung von Transitionen mit Produktionsregelsystemen.

51) Bei der späteren Implementierung des Übergangsschemas durch das Softwarepaket PASIPP entfallen die Teilanweisungen $FAK_{u.-} := FAK_{u.r}$ für alle $u \in (PRÄ\text{-}VB_v)$ und $FAK_{u.+} := FAK_{u.-}$ für alle $u \in (PRÄ\text{-}NB_v)$. Dort gewährleistet die Konstruktion einer dynamischen Datenbank für alle PROLOG-Fakten, daß alle unveränderten Fakten automatisch aus der Referenz- in die Folgefaktenmenge übernommen werden. Dies entspricht einer Lösung des "frame"-Problems durch das Prinzip, daß alle nicht explizit veränderten Größen implizit als unverändert präsupponiert werden. Dieselbe Präsuppositionslösung des "frame"-Problems findet sich auch im Bereich der KI-Forschung. Sie wurde z.B. bei der Entwicklung des "intelligenten" Planungsautomaten STRIPS implementiert; vgl. FIKES (1971), S. 613f.; WILKINS (1984), S. 275; HERTZBERG (1986), S. 157; FIDELAK (1988b), S. 12ff.; WINTER,RO. (1991), S. 145. Vgl. zu diesem Automaten auch FIKES (1971), S. 608ff.; ZELEWSKI (1986a), S. 191, 494f. u. 530f. (sowie die dort angeführten Quellen); WINTER,RO. (1991), S. 144ff.
Das "frame"-Problem betrifft die Schwierigkeit, zwei zunächst konfliktionär erscheinende Anforderungen miteinander zu vereinbaren. Einerseits soll in Objektmodellen mit zeitlich veränderlichen Modellzuständen das Wissen über die aktuellen Modellzustände jederzeit *vollständig* repräsentiert werden. Andererseits wird angestrebt, beim Übergang zwischen Modellzuständen nicht das gesamte zustandsrepräsentierende Wissen reformulieren zu müssen, sondern sich auf die Darstellung der übergangsspezifischen Veränderungen beschränken zu können. Diese beiden Anforderungen werden durch das o.a. Prinzip zugleich erfüllt. Detailliertere Behandlungen des "frame"-Problems finden sich vor allem im Kontext der Erforschung Künstlicher Intelligenz; vgl. MCCARTHY,J. (1969), S. 487; SANDEWALL (1972), S. 195ff.; KRAMOSIL (1975), S. 53; RAPHAEL (1976), S. 147; DOYLE (1980), S. 8; HAYES,P. (1981), S. 223ff.; RAULEFS (1984), S. 34; WILKINS (1984), S. 273ff.; HERTZBERG (1985), S. 11ff.; REINFRANK (1985b), S. 7ff.; HERTZBERG (1986), S. 151f. u. 157; GEORGEFF (1986), S. 71 u. 74; ZELEWSKI (1986a), S. 363f.; FIDELAK (1988b), S. 11ff.; MORRIS,P. (1988), S. 383; SHOHAM (1988c), S. 281; BREWKA (1989), S. 96; DORN (1989), S. 7; BIBEL (1989), S. 49, 51f. u. 61; WINTER,RO. (1991), S. 145.

52) Später wird das Übergangsschema durch Transaktionen konkretisiert, die Transitionen eines Petrinetzes i.e.S. zugeordnet werden. Daher entspricht die hier thematisierte Lokalität der Variablendefinition für das Übergangsschema dort dem Sachverhalt, daß die Variablen jeweils lokal in bezug auf eine Transaktion und deren korrespondierende Transition definiert werden. Diese Lokalität der Definition von transaktions- bzw. transitionszugehörigen Variablen wird im Rahmen des Petrinetz-Konzepts zwar selten explizit ausgesprochen, liegt aber implizit allen Prädikat/Transition-Netzen zugrunde. Zu den seltenen Ausnahmen, welche die Lokalität der Variablendefinition in Petrinetzen ausdrücklich ansprechen, gehören MURATA,TA. (1988b), S. 485 u. 493(ff.); REISIG (1989b), S. 3 ("variables are bound in the environment of transitions").

53) Ausgenommen ist die anonyme Variable "_", die bei einer Variablenbelegung durch überhaupt kein formales Objekt ersetzt wird.

54) Dies entspricht der Variablenkonvention für PROLOG-Klauseln; vgl. CORDES (1988), S. 13.

55) Dies stimmt abermals mit der Variablenkonvention in der Programmiersprache PROLOG überein; vgl. LEVI,G. (1986), S. 403 (dort wird die Variable "x" in den Ausdrücken "<-P(x,b)" und "P(x,x)" in unterschiedlicher Weise durch x=a bzw. x=b gebunden); CORDES (1988), S. 13. Dagegen weicht die lokale, jeweils auf eine Übergangsoperation beschränkte Reichweite von Variablenbelegungen deutlich von der sonst üblichen Konvention ab, *alle* Vorkommnisse derselben Variable in einer vorgegebenen Formelzusammenhang durch dieselbe Konstante zu ersetzen. Vgl. zu dieser globalen Variablenbelegung z.B. die "Einsetzungsregel" bei VON WEIZSÄCKER (1985), S. 69f.

56) Vgl. die Anmerkung zu "toxischen" Fernwirkungen.

57) Wenn dieselbe Variable in unterschiedlichen Übergangsoperationen vorkommt und nicht lokal definiert wäre, müßte sie in allen jenen Übergangsoperationen - sofern sie ausgeführt werden - durch dasselbe formale Objekt substituiert werden. Ein solcher globaler Variablencharakter würde an den Modellierungsträger die Anforderung stellen, bei der Gestaltung eines prädikatenlogischen Objektmodells die Gesamtheit aller potentiellen Auswirkungen eines Variablenvorkommnisses auf alle weiteren Vorkommnisse derselben Variablen zu überblicken. Dies kann bei umfangreicheren Objektmodellen in der Regel nicht gewährleistet werden. Vgl. dazu auch den Hinweis bei VALETTE (1979b), S. 157, daß die Verwendung globaler Variablen des öfteren zu unübersichtlichen und schwer verifizierbaren Konstruktionen führt. Vgl. darüber hinaus MATTERN (1989a), S. 206f., zu Schwierigkeiten, die bei der Verwendung globaler Variablen in nebenläufigen Simulationskonzepten auftreten können.

58) Darauf wurde bereits in voranstehenden Anmerkungen hingewiesen.

59) Ein leerer Wirkungsbereich widerspräche der grundlegenden Definition von Operationen durch Veränderungen von Faktenmengen. Die einzigen Anweisungen, welche die Veränderung einer Faktenmenge bewirken können, stellen die Entfernungs- und die Ergänzungsanweisung dar. Erste setzt mindestens ein Eingangs-, letzte mindestens ein Ausgangsprädikat voraus. Folglich ist eine veränderungsbewirkende Operationsausführung nur dann möglich, wenn für die Operation mindestens ein Eingangs- oder mindestens ein Ausgangsprädikat definiert ist, d.h. wenn ihr Vor- bzw. ihr Nachbereich nicht leer ist. Da der Wirkungsbereich die Vereinigung von Vor- und Nachbereich ist und beide zusammen niemals leer sein dürfen, muß der Wirkungsbereich nicht-leer sein; q.e.d.

60) Es wird auch dann noch von einer einfachen Operationswirkung gesprochen, wenn Veränderungen auf beiden Ebenen miteinander kombiniert werden.

61) Falls genau eine der beiden Multiplizitäten den Wert Null annimmt, liegt eine elementare Operationswirkung im Sinne der voranstehenden Erläuterung vor.

62) Mehrere Übergangsoperationen, die zeitgleich oder zeitüberlappend auf dasselbe Informationsfakt Bezug nehmen, stören sich nicht, falls sie innerhalb eines Automatischen Informationsverarbeitungssystems implementiert werden, das nebenläufige Leseoperationen auf einem gemeinsamen Speicher zuläßt. Dies wird hier vorausgesetzt. Daher läßt sich jedes Informationsprädikat als ein Speicherbezirk auffassen, dessen aktueller Inhalt aus der Faktenmenge des Informationsprädikats besteht und durch mehrere Informationsverarbeitungsprozesse nebenläufig gelesen werden kann.

63) Dies ist insofern bedeutsam, als solche Operationsbeeinflussungen durch Informationen, die selbst qua Voraussetzung unverändert bleiben, in sonst üblichen Beiträgen aus der Perspektive des Petrinetz-Konzepts nicht adäquat behandelt werden. Dort wird die Konsultation (mindestens) eines Informationsfakts im allgemeinen überhaupt nicht explizit thematisiert. Aber implizit wird diese Informationsauswertung zumeist so dargestellt, daß das Informationsprädikat einer Stelle und das zu konsultierende Informationsfakt einer Marke auf dieser Stelle zugeordnet werden. Die Operation, deren Ausführung durch das Informationsfakt beeinflußt werden soll, wird durch eine Transition erfaßt, welche die Marke des Informationsfakts zunächst von der Stelle entfernt, die faktische Informations auswertet und abschließend die Marke auf derselben Stelle wieder unverändert ablegt.
Während der Operationsausführung - in der Zeitspanne zwischen dem Ausführungsbeginn der Entfernungs- und dem Ausführungsende der Einfügungsanweisung - bleibt es jedoch aus der Netzperspektive ungeklärt, was mit dem Informationsfakt geschehen ist. Es wurde zwar durch die Entfernungsanweisung (irgendwann) aus der alten Faktenmenge des Informationsprädikats eliminiert, doch noch nicht zur Wiederherstellung der unveränderten Faktenmenge zur intermediären Faktenmenge hinzugefügt. Es resultiert eine temporäre Definitionslücke, in welcher der Verbleib des konsultierten Informationsfakts vollkommen im dunkeln bleibt. Diese Unklarheit stellt schon an sich Anlaß zur Kritik dar. Darüber hinaus impliziert sie aber auch eine Zugriffslücke, falls weitere Übergangsoperationen existieren, die während der Ausführung der hier betrachteten Operation ebenfalls auf das Informationsfakt zugreifen müßten, um es ebenfalls zu konsultieren (oder um es zu entfernen). Da dieses Informationsfakt in der o.a. Zeitspanne in der intermediären Faktenmenge des Informationsprädikats überhaupt nicht mehr enthalten ist, ohne daß sein Verbleib geklärt wäre, muß der Zugriffsversuch der anderen Operationen auf dieses Fakt scheitern. Es resultiert als Fernwirkung, daß jene Operationen entweder überhaupt nicht oder nur verzögert ausgeführt werden können. Beides ist unerwünscht, weil diese indirekte Wirkungsweise vom Übergangsschema in keiner Weise abgedeckt wird.

Die Zugriffslücke könnte zwar im Rahmen des Petrinetz-Konzepts dadurch zugedeckt werden, daß unendlich schnelle Ausführungen und augenblickliche Aufeinanderfolgen aller Anweisungen innerhalb der Übergangsprozedur einer Operation unterstellt werden. Dann würde der oben skizzierte, temporär nicht klar definierte Zwischenzustand der intermediären Faktenmenge überhaupt nicht auftreten. Die gesamte Ausführung der Übergangsprozedur wäre laut Voraussetzung auf einen Zeitpunkt reduziert. Aber dann wird ein konzeptionell sehr wohl vorhandener problematischer Zwischenzustand durch die künstliche - und realitätsferne - Prämisse unendlicher Ausführungsgeschwindigkeiten und simultaner Anweisungsausführungen lediglich fiktiv aufgelöst. Eine tatsächliche Bewältigung der konzeptionellen Schwachstelle "Zugriffslücke" wird hierdurch aber nicht erreicht. Darüber hinaus widerspricht die unterstellte simultane Ausführung mehrerer Anweisungen dem oben eingeführten Prozedurkonzept, in dem die einzelnen Anweisungen einer Prozedur sequentiell aufeinander folgen. Der mögliche Einwand, die Anweisungen der Übergangsprozedur ließen sich auch als nebenläufiger Prozeß gestalten, greift hier nicht. Denn bei der hier betrachteten Konsultation von Informationsfakten durch deren Entfernung und reziproke Hinzufügung zur Faktenmenge sind die Entfernungs- und Einfügungsanweisung weder logisch noch informationstechnisch voneinander unabhängig. Vielmehr besteht die Präzedenzbeziehung, daß die Ausführung der Einfügungsanweisung frühestens begonnen werden kann, nachdem die Ausführung der Entfernungsanweisung abgeschlossen ist. Denn die erstgenannte Anweisungsausführung benötigt als Input die intermediäre Faktenmenge, die den Output der letztgenannten Anweisungsausführung darstellt; vgl. dazu die spätere Definition eines Übergangsschemas.
Aufgrund der voranstehend skizzierten Schwierigkeiten folgt der Verf. der üblichen Vorgehensweise nicht. Statt dessen hat er das Konzept der Informationsprädikate eingeführt. Für sie bestehen die Unklarheiten hinsichtlich des temporären Verbleibs von Informationsfakten und die Zugriffslücke nicht.

64) Fernwirkungen sind hier prinzipiell unerwünscht, weil unterstellt wird, daß das Übergangsschema ÜS jede Übergangsoperation *vollständig* definiert. Dann darf die Ausführung einer Übergangsoperation keine anderen Auswirkungen haben als diejenigen, die aufgrund des Übergangsschemas explizit spezifiziert sind. Der hierdurch definierten Lokalität von Übergangsoperationen würden die o.a. Fernwirkungen grundsätzlich widersprechen. Vgl. dazu auch die inhaltlich analoge, aber auf den Kontext der Programmiertechnik bezogene Problematisierung von "toxischen" Neben- oder Fernwirkungen und die daraus abgeleitete Forderung nach neben- bzw. fernwirkungsfreien Konstrukten bei BERZHEIM (1975a), S. 119; BERZHEIM (1975b), S. 237 u. 239; ZELEWSKI (1986a), S. 215f.; REUTER (1987), S. 406 u. 411.

65) Damit erlauben Übergangsoperationen auch eine einfache Darstellung von Produktionsregeln aus dem Bereich der Künstlichen Intelligenz. Denn solche Produktionsregeln zeichnen sich durch Anwendungsbedingungen aus, die jeweils durch regelspezifische Informationsmuster definiert werden. Daher lassen sich Produktionsregeln als Übergangsoperationen mit nicht-leeren Informationsbereichen und zugehörigen Prätests betrachten. Vgl. dazu die Einbindung des Produktionsregelansatzes in das Konzept Synthetischer Netze. Dort wird besonders deutlich, daß Produktionsregeln als Konkretisierungen des Übergangsschemas aufgefaßt werden können.

66) Der operationsspezifische Anwendungskontext wird jeweils in der Definition der Prätestbedingung definiert. Da bei jeder Operationsausführung zunächst die Gültigkeit des Prätestprädikats überprüft wird, werden aufgrund der Konstruktionsweise des Übergangsschemas alle weiteren Schritte der Operationsausführung unterbunden, solange dieses Prädikat ungültig bleibt.

67) Die Struktur der Anwendungskontexte braucht keineswegs disjunkt und flach zu sein. Vielmehr können sich verschiedene Anwendungskontexte auch - auf derselben Kontextebene - gegenseitig überlappen. Darüber hinaus ist es möglich, durch Über- und Unterordnungsbeziehungen zwischen Anwendungskontexten eine tief gegliederte, hierarchische Kontextstruktur in mehreren Ebenen aufzubauen.

68) In dieser Hinsicht besteht eine interessante Beziehung zur Künstlichen Intelligenz. Die Produktionsregeln des dort weit verbreiteten regelbasierten Wissensrepräsentationsschemas können als spezielle Konkretisierungen des hier vorgestellten Übergangsschemas aufgefaßt werden. Dies wurde bereits angesprochen.
Seitens der KI-Forschung wird des öfteren beklagt, daß die Produktionsregelmengen größerer Expertensysteme schnell unübersichtlich und hinsichtlich ihrer Auswertung ineffizient werden. Daher werden interne Strukturierungskonzepte für solche Produktionsregelmengen gefordert. Für diesen Zweck läßt sich das o.a. Konzept der Anwendungskontexte in der Weise nutzen, daß die Strukturierung der Anwendungskontexte für die Prätests der Informationsprädikate von Übergangsoperationen als Kontextstrukturierungen entsprechender Produktionsregeln ausgelegt werden.
Vgl. auch die Hinweise auf Ansätze des Petrinetz-Konzepts, mit deren Hilfe sich Anwendungskontexte für Produktionsregeln von Expertensystemen explizit definieren lassen. Aus dieser Perspektive der Strukturierung großer Wissensbestände stellen Petrinetze eine bemerkenswerte Befruchtung für die KI-Forschung dar.

69) Inhibitorkanten bedeuten einen "Nulltest": Das Prätestprädikat genau dann gültig ist, wenn die Extension desjenigen Prädikats, das im betrachteten Netz der Stelle im Ursprung der Inhibitorkante zugeordnet ist, leer ist. Mit dieser leeren Faktenmenge korrespondiert eine Markierung, unter die Anzahl der Markenkopien, die sich auf der betrachteten Stelle befinden, "Null" beträgt.

70) Die Ausdruckskraft von TURING-Automaten reicht aus, um die Modellierungsmächtigkeit aller anderen rekursiv strukturierten Konzepte im Prinzip zu erreichen. Da alle - derzeit bekannten - formalen Konzepte eine rekursive Struktur besitzen oder auf vereinfachten Unterfällen des allgemeinen Rekursionsschemas beruhen, erhalten Petrinetze auf diese Weise die Modellierungsmächtigkeit jedes beliebigen anderen formalen Modellierungskonzepts.

71) Die relativ effizienten Auswertungsalgorithmen, die auf der Inzidenzmatrix eines Netzes arithmetische Operationen ausführen, können auf Petrinetze mit Inhibitorkanten grundsätzlich nicht mehr angewendet werden. Dies bedeutet eine Beschränkung auf weitaus weniger effiziente Konstruktions- und Auswertungsalgorithmen für Erreichbarkeitsgraphen oder auf Simulationsalgorithmen, bei denen ein Dilemma zwischen Auswertungseffizienz und Auswertungsspanne besteht. Näheres zu diesen Aspekten der Modellierungseffizienz an anderer Stelle.

72) Vgl. die Vorbehalte bezüglich der Schwierigkeiten, den Begriff "natürlicher" Repräsentationen inhaltlich präzise zu definieren. Auch hier wird an ein intuitives Vorverständnis des Rezipienten appelliert.

73) Nachfolgend wird das Thema der Nebenbedingungen verfeinert fortgeführt, das bereits im Kontext von Stelle/Transition-Netzen aus gröberer Perspektive angesprochen wurde.

74) Der Invarianzaspekt läßt sich in einem starken und in einem schwachen Sinn verstehen. Eine schwache Invarianz liegt vor, wenn nur gefordert wird, daß eine Nebenbedingung *nach* einer Operationsausführung unverändert vorliegen muß. Damit besteht der Freiheitsgrad, daß die Nebenbedingung *während* der Operationsausführung verändert werden darf. Da Nebenbedingungen hier als Prädikate formuliert werden, bedeutet eine solche operationsbegleitende Nebenbedingungsveränderung, daß die Faktenmenge des nebenbedingungsspezifischen Prädikatssymbols während der Operationsausführung variiert wird. Hierdurch könnten Fernwirkungen auf andere Operationen ausgehen, die auf dasselbe Nebenbedingungsprädikat zugreifen. Solche Fernwirkungen träten immer dann ein, wenn die Veränderung der Faktenmenge des Nebenbedingungsprädikats bewirkte, daß die anderen Operationen überhaupt nicht mehr oder nur noch mit veränderten Ergebnissen ausgeführt werden könnten. Auf die Unerwünschtheit solcher Fernwirkungen wurde bereits in einer früheren Anmerkung eingegangen. Daher wird für Nebenbedingungen hier stets Invarianz im starken Sinn gefordert: Die Nebenbedingungen müssen nicht nur nach, sondern auch während der gesamten Operationsausführung unverändert vorliegen.

75) Darauf wurde bereits hingewiesen. Der Invarianzaspekt wird durch die allgemein übliche Behandlung von Nebenbedingungen im Kontext des Petrinetz-Konzepts nicht zufriedenstellend abgedeckt. Dort wird eine Nebenbedingung zumeist als eine Stelle dargestellt, für welche die konstante Markierung durch genau eine Marke gefordert wird. Andere Markenanzahlen oder - im Fall von Prädikat/Transition-Netzen - gar nebenbedingungsspezifische Anforderungen an die Argumentausprägung(en) der Marke(n) werden gewöhnlich nicht thematisiert. Aber selbst wenn der letzte - nur selten verwirklichte - Fall betrachtet wird, bleibt der Umgang mit Nebenbedingungen im Petrinetz-Konzept dennoch weiterhin unbefriedigend. Dies liegt an der Zugriffslücke, die bereits in einer früheren Anmerkung besprochen wurde. Ihre Problematik wird nachfolgend - nunmehr jedoch mit speziellem Bezug auf die Formulierung von Nebenbedingungen - nochmals verdeutlicht.
Ausgangspunkt ist eine Übergangsoperation, deren Ausführungsmöglichkeit von der Erfüllung einer Nebenbedingung abhängt. Die Nebenbedingung wird durch ein Nebenbedingungsprädikat und eine Faktenmenge für dieses Prädikatssymbol formuliert, welche die Nebenbedingungsvorschrift widerspiegelt. Die Übergangsoperation wird in einem Petrinetz als Transition dargestellt. Die Operationsausführung entspricht dann dem Schalten dieser operationsabbildenden Transition. Im korrespondierenden Netzmodell entsteht eine 1-Schleife, auf deren immanente Probleme bereits hingewiesen wurde. Falls die Nebenbedingung erfüllt und die Übergangsoperation tatsächlich ausgeführt wird, sorgen eine Entfernungs- und eine Ergänzungsanweisung in der Operationsdefinition dafür, daß die Faktenmenge des Nebenbedingungsprädikats zunächst um eine Teilmenge - gewöhnlich ein einzelnes Fakt - verringert und danach wieder um dieselbe Faktenteilmenge vermehrt wird. Es ist aber "unnatürlich", durch die Ausführung zweier genau entgegengesetzter Anweisungen im Ergebnis eine Nichtveränderung "bewirken" zu wollen. Darüber hinaus erscheint es selbstwidersprüchlich, eine veränderungsbezogene Operationsausführung mit der Nichtveränderung der Gültigkeit einer Nebenbedingung zu kombinieren. Diese Selbstwidersprüchlichkeit äußert sich darin, daß die intendierte Invarianz der Nebenbedingungsgültigkeit nur durch einen inadäquaten Kompensationsmechanismus sichergestellt werden kann: Weder die Entfernungs- noch die Ergänzungsanweisung in der Operationsdefinition entsprechen einem Aspekt aus der unverändert gültigen Nebenbedingung. Sie leisten daher - jede für sich genommen - keine "natürliche" Repräsentation der betroffenen Nebenbedingung und ihrer unveränderten Gültigkeit. Statt dessen bewirkt zunächst die Entfernungsanweisung den Repräsentationsfehler, die Nebenbedingung als ungültig erscheinen zu lassen. Erst die anschließende Ergänzungsanweisung heilt diesen Repräsentationsfehler nachträglich so, daß das Schalten der operationsdarstellenden Transition *insgesamt* die Gültigkeit der Nebenbedingung nicht verändert. Die zweifache Veränderung des Gültigkeitsstatus der Nebenbedingung *während* der Schaltens der Transition verletzt eklatant die vorausgesetzte Invarianz ihrer Gültigkeit. Nur durch die gegenseitige Kompensierung der entgegensetzt gerichteten Gültigkeitsveränderungen resultiert am Schaltende die korrekte Gültigkeit der Nebenbedingung.

Zwar könnte eingewandt werden, wegen der Punktförmigkeit der Schaltakte von Transitionen sei es belanglos, was während des Schaltens einer Transition geschehe. Doch vermag sich der Verf. dieser Ansicht aus drei Gründen nicht anzuschließen. Erstens hält er ein Repräsentationskonzept auch dann für nachteilhaft, wenn es so angelegt ist, daß sich seine Repräsentationsfehler nicht auszuwirken vermögen. Denn es kommt aus der Perspektive der Repräsentationsnatürlichkeit nicht darauf an, ob sich Repräsentationsunzulänglichkeiten tatsächlich auszuwirken vermögen. Vielmehr interessiert hier nur, ob die Repräsentation selbst der Intuition einer "natürlichen" Wiedergabe des jeweils darzustellenden Sachverhalts entspricht. Genau dies ist aber bei der Verwendung von 1-Schleifen für die Repräsentation von Nebenbedingungen nicht der Fall. Denn der komplizierte, intransparente Mechanismus, bei dem sich zwei Repräsentationsfehler gegenseitig kompensieren, läßt sich in keiner Weise als eine "natürliche" Darstellung für den einfachen und klaren Sachverhalt ausgeben, daß die Gültigkeit einer Nebenbedingung fortbesteht. Die Undurchsichtigkeit dieses Kompensationsmechanismus wird auch dadurch unterstrichen, daß es im Petrinetz-Konzept widerstreitende Auffassungen darüber gibt, wie solche 1-Schleifen beim Schalten von Transitionen "korrekt" zu berücksichtigen sind. Zweitens sind in der Zeitspanne zwischen dem Ausführungsbeginn der Entfernungs- und dem Ausführungsende der Ergänzungsanweisung die gleichen unerwünschten Fernwirkungen möglich, die bereits oben für die Operationsbeeinflussung durch Informationsprädikate unter dem Aspekt der Zugriffslücke angesprochen wurden. Aufgrund dieser Fernwirkungen, die hinsichtlich ihrer Konsequenzen oftmals nicht mehr abzuschätzen sind, ist die hier skizzierte Repräsentationsweise von Nebenbedingungen nicht nur unnatürlich, sondern auch intransparent. Drittens werden mitunter Transitionen mit positiven Schaltdauern versehen. Der Verf. folgt diesem Ansatz zwar nicht, doch muß seine gelegentliche Verwendung in Rechnung gestellt werden. Wenn eine solche Transition zu einer 1-Schleife für die Repräsentation einer Nebenbedingung gehört, wird das Problem der zuvor angesprochenen Zugriffslücke noch verschärft. Dann stellt das Schalten der Transition keinen punktförmigen Schaltakt mehr dar, sondern bildet einen zeitlich ausgedehnten Schaltvorgang. Während dieses Schaltvorgangs ist der Gültigkeitsstatus der Nebenbedingung nicht mehr wohldefiniert. Denn es bleibt offen, wann die Entfernungs- und die Ergänzungsanweisungen tatsächlich ausgeführt sind. Daher kann in dem Zeitraum, in dem der Schaltvorgang der Transition aus der 1-Schleife andauert, die Aktivierung aller anderen Transitionen, die von der Gültigkeit oder Ungültigkeit der Nebenbedingung abhängen, nicht mehr klar festgestellt werden.
Bei der Verwendung von Informationsprädikaten und Prätests werden die vorgenannten Schwierigkeiten dagegen grundsätzlich vermieden. Denn die Faktenmengen dieser Informationsprädikate sind nach Maßgabe der Konstruktion des Übergangsschemas weder von Entfernungs- noch von Ergänzungsanweisungen betroffen. Informationsfakten werden zwar während der Operationsausführung zur Kenntnis genommen, verbleiben aber während der gesamten Ausführungsdauer unverändert in den Faktenmengen ihrer Informationsprädikate. Daher können auch keine Zugriffslücken eintreten. Folglich drohen auch keine unerwünschten Fernwirkungen. Diese veränderungs- und fernwirkungsfreie Konsultierung von Informationsfaktenmengen stellt genau die gesuchte "natürliche" Repräsentation von Nebenbedingungen in prädikatenlogischer Formulierung dar. Sie erweist sich ebenso einfach und übersichtlich wie die zu repräsentierende Nebenbedingung selbst. Da eine derart natürliche Repräsentationsweise für Nebenbedingungen existiert, ist die Verwendung von 1-Schleifen für diesen Repräsentationszweck nicht nur unnatürlich, sondern schlicht überflüssig. Damit wird aber keineswegs ausgeschlossen, 1-Schleifen weiterhin für andere Repräsentationszwecke zu verwenden. Denn sie werden auf jeden Fall dann benötigt, wenn das Schalten einer Transition aus der Faktenmenge eines Prädikatssymbols ein altes Fakt entfernen und durch ein neues Fakt ersetzen soll.

76) Die Inhibitorkanten, die schon an früherer Stelle angesprochen wurden, stellen einen Sonderfall der hier thematisierten Nebenbedingungen dar. Sie konkretisieren den Geltungsaspekt dahingehend, daß die Faktenanzahl Null für dasjenige Informationsprädikat gefordert wird, das dem Ursprung der Inhibitorkante zugeordnet ist.

77) Die Bezeichnungsweise "UNDO" ist dem Rücksetzungskonzept für integritätsverletzende Datenbanktransaktionen entlehnt; vgl. zu ihrer expliziten Verwendung z.B. REUTER (1987), S. 398f.; NETT (1988), S. 56f.; VON ZIMMERMANN (1990), S. 257f.

78) Daher muß für die Ausgangsfaktenmenge FAK_0, die bereits in einer früheren Anmerkung angesprochen wurde, alle Integritätsbedingungen erfüllen. Vgl. zur analogen Integritätspräsupposition für die Ausgangszustände von Datenbanksystemen, auf die datenverarbeitende Transaktionen angewendet werden sollen, REUTER (1987), S. 404.
Die Einhaltung aller Integritätsbedingungen durch die Ausgangsfaktenmenge kann bei der späteren Definition des Kernkonzepts Synthetischer Netze zunächst noch nicht berücksichtigt werden. Denn die qualitativ-semantische Unterscheidung zwischen der Erfüllung und der Verletzung einer Integritätsbedingung läßt durch keines der rein syntaktisch definierten Netzkonstrukte des Kernkonzepts erfassen. Später wird aber die Konzepterweiterung durch faktische Transitionen erlauben, auch solche qualitativen Unterscheidungen in formaler Weise auszudrücken. Dann kann auch innerhalb der Definition eines Synthetischen Netzes die Meta-Integritätsbedingung ausgedrückt werden, daß alle voranstehend erörterten Objekt-Integritätsbedingungen seitens der Ausgangsfaktenmenge erfüllt werden müssen.

79) Diese Ausschließlichkeit entweder vollständiger oder aber unterlassener Operationsausführung wird besonders deutlich bei REUTER (1987), S. 398 u. 403, und von NETT (1988), S. 55, herausgestellt (die sich allerdings auf Transaktionen in Datenbanksystemen beziehen).

80) Vgl. zur Integritätswahrung durch Bedingungen, deren Erfüllung jeweils am Ende einer (hypothetischen) Operationsausführung überprüft wird, REUTER (1987), S. 396f. u. 399; KOTZ (1989), S. 1.

81) Vgl. MERTENS (1986a), S. 332.

82) Vgl. zu praktischen Einschränkungen der Möglichkeit, durch Übergangsoperationen (Transaktionen) die Integrität komplexer (informationsverarbeitender) Systeme absolut sicherzustellen, REUTER (1987), S. 407; KOTZ (1989), S. 1 u. 40.

83) Das Triggerkonzept wird vor allem im Kontext der transaktionsorientierten Informationsverarbeitung behandelt; vgl. LUM (1982), S. 338; BERTHOLD (1983), S. 20ff.; MERTENS (1984), S. 28ff. u. 42; MERTENS (1986a), S. 326f. u. 330f.; ZELEWSKI (1986a), S. 320, 328, 569f., 830, 834 u. 1243f.; SCHOLZ (1988), S. 166f. u. 170; MERTENS (1988e), o.S., Abb. 13 (nach S. 20); BECKSTEIN (1988a), S. 160f.; SCHEER (1989c), S. 59ff.; SCHEER (1990a), S. 19 u. 21ff.; SCHEER (1990c), S. 63f. u. 83; SCHEER (1991d), S. 119, 175f. u. 185f.
Vgl. insbesondere zur Anwendung des Triggerkonzepts auf den hier interessierenden Aspekt der Integritätswahrung REUTER (1987), S. 386ff.; ESTER (1989), S. 9, 63ff., 89ff., 98ff., 115, 118ff. u. 128f.; KOTZ (1989), S. 40 u. 42ff., insbesondere S. 48f. u. 56f.; JARKE (1989b), S. 13ff.; LEIKAUF (1989), S. 142; WINTER,RO. (1991), S. 212 u. 223; SCHEER (1991d), S. 163.
Darüber hinaus liegt das Triggerkonzept den "Dämonen" zugrunde, die seitens der KI-Forschung z.B. im Rahmen der "blackboard"-Architektur von nebenläufigen Informationsverarbeitungssystemen verwendet werden; vgl. zu solchen "Dämonen" (Agenten o.ä.) PACINI (1983), S. 862ff.; ZELEWSKI (1986a), S. 214 u. 328ff.; ZELEWSKI (1986c), S. 28 u. 33ff.; BECKSTEIN (1988a), S. 160; ZELEWSKI (1988b), S. 357f.; KOTZ (1989), S. 52f.; FOLDENAUER (1990), S. 55.

84) Damit wird der Empfehlung von MERTENS (1986a), S. 325, entsprochen, der aktionsorientierten Informationsverarbeitung verstärkte Aufmerksamkeit zukommen zu lassen.

85) Vgl. dazu die prädikatenlogische Darstellung von Integritätsbedingungen.

86) Dies entspricht der Prämisse, bei der integritätswahrenden Anwendung des Triggerkonzepts stets von integren ("konsistenten") Informationsgesamtheiten (Datenbanken, Wissensbasen u.ä.) auszugehen; vgl. ESTER (1989), S. 65.

87) Das kann z.B. dadurch geschehen, das ein entsprechendes Prädikat, das zur Informationsübermittlung an Modellbenutzer vorgesehen ist, gültig wird. Vgl. ESTER (1989), S. 18 ("Fehlermeldung" und "Rückfragen beim Benutzer"), 49 ("Inkonsistenz wird dem Benutzer angezeigt") u. 65 ("Warnung an den Benutzer"); LEIKAUF (1989), S. 137 ("Warnung") u. 142f.; MOERKOTTE (1990), S. 3f.
Es wird auch von *schwachen* Integritätsbedingungen (oder Konsistenzbedingungen) gesprochen, falls ihre Verletzung zwar unerwünscht ist, aber - wie in den vorgenannten Quelle ausgeführt - unter entsprechender Verletzungsanzeige hingenommen wird; vgl. ESTER (1989), S. 18; LEIKAUF (1989), S. 137 u. 145.

88) Vgl. REUTER (1987), S. 386f. ("Folgeaktionen"); KOTZ (1989), S. 43 ("Reaktion auf Konsistenzverletzung"); ESTER (1989), S. 9 ("bei Inkonsistenzen geeignete Anpassungen") sowie S. 18, 54ff. u. 65 ("Folge-Update(s)"); JEUSFELD (1990), S. 10 ("forward recovery"); MOERKOTTE (1990), S. 4.

89) Vgl. ESTER (1989), S. 65 ("Ablehnen"); JEUSFELD (1990), S. 10 ("refuse the transaction"); vgl. darüber hinaus die Konstruktionen, die mit der Hilfe von faktischen Transitionen für die Erfassung quantifizierter Formeln eingeführt werden.

90) Unter "Integritätsmanagement" oder "Konsistenzmonitoring" wird hier jedes Konzept verstanden, das für eine vorgegebene Informationsgesamtheit zwei Funktionen erfüllt. Erstens kann es die Integrität (Konsistenz) der Informationsgesamtheit in deren Ausgangszustand überprüfen. Zweitens ist es in der Lage, bei verarbeitungsbedingten Veränderungen der Informationsgesamtheit über die Erhaltung ihrer Integrität (Konsistenz) zu wachen. Vgl. zu Konzepten für ein solches "Integritätsmanagement" oder "Konsistenzmonitoring" NAKANO,R. (1983), S. 557ff.; LIU,N. (1987), S. 124 u. 127ff.; SAAKE (1988), S. 235ff.; GLOVER (1988), S. 199ff.; OWSNICKI-KLEWE (1988), S. 77ff.; HARMON (1989), S. 356; ZELEWSKI (1989c), S. 7ff., und ZELEWSKI (1989e), S. 73ff. u. 85ff. (jeweils auf der Basis von Petrinetzen); JEUSFELD (1990), S. 4ff.; KRALLMANN (1990a), S. 23.

91) Vgl. dazu die analoge, aber auf Transaktionen bezogene Feststellung von NETT (1988), S. 55, ihre Atomarität bedeute lediglich, "daß jede Transaktion *gegenüber der Außenwelt* als unteilbar ... erscheint" (kursive Hervorhebung durch den Verf.).

92) Vgl. zur ereignis- oder (trans)aktionsorientierten Informationsverarbeitung BERNSTEIN (1980), S. 18ff.; LUM (1982), S. 338; BERTHOLD (1983), S. 20ff.; MERTENS (1984), S. 28ff. u. 42; ALDINGER (1985a), S. 110; MERTENS (1986a), S. 323ff.; ZELEWSKI (1986a), S. 569ff.; HOHENSTEIN (1986), S. 186 u. 205ff.; HOHENSTEIN (1987), S. 79ff.; REUTER (1987), S. 396ff.; MEYER-WEGENER (1987), S. 120ff.; SCHEER (1987a), S. 35f. u. 227ff.; HELBERG (1987), S. 86f.; ASCHE (1988), S. 20; SCHOLZ (1988), S. 165ff.; NETT (1988), S. 55ff.; MEYER-WEGENER (1988), S. 47ff.; MERTENS (1988a), S. 197; WILDEMANN (1988c), S. 54f. u. 62f.; KOTZ (1989), S. 1ff.; LIPECK (1989), S. 5ff.;

FAHRION (1989), S. 364ff.; JABLONSKI (1990), S. 100ff.; SCHEER (1990a), S. 19 u. 22ff.; SCHEER (1990c), S. (58,) 60ff. u. 83; VON ZIMMERMANN (1990), S. 256ff.; SCHEER (1991d), S. 174ff.
Vgl. zu ersten Ansätzen, das Konzept der aktionsorientierten Informationsverarbeitung in PPS-Systeme einfließen zu lassen, ALDINGER (1985a), S. 110; MERTENS (1986a), S. 324ff., insbesondere S. 327ff.; GÜNTHER,H. (1986), S. 241ff.; MERTENS (1988e), S. 15; WILDEMANN (1988c), S. 63. Vgl. allerdings auch den Hinweis von HELBERG (1987), S. 86f., daß die aktionsorientierte Informationsverarbeitung bei PPS-Systemen der betrieblichen Praxis noch kaum Beachtung gefunden hat.

93) Vgl. zur Atomarität (konzeptionellen Unteilbarkeit) von Transaktionen HOHENSTEIN (1986), S. 205; HOHENSTEIN (1987), S. 79; REUTER (1987), S. 398, 402 u. 405; MEYER-WEGENER (1987), S. 121; NETT (1988), S. 55; ESTER (1989), S. 17; LIPECK (1989), S. 5; JABLONSKI (1990), S. 103; SCHEER (1990c), S. 83; VON ZIMMERMANN (1990), S. 256f.; SCHEER (1991d), S. 174.
Die Atomarität von Transaktionen spiegelt sich auch bei MERTENS (1988e), o.S., Abb. 13 (nach S. 20), wider. Er unterscheidet, ob eine Verarbeitungsprozedur bei Eintreffen eines Triggersignals unmittelbar begonnen wird ("Transaktion") oder ob mehrere solche Prozeduren gesammelt und periodisch als Prozedursammlungen ausgeführt werden ("Batch"). Es wird hier von Verfeinerungen des Transaktionskonzepts abgesehen, die durch Verschachtelung von Subtransaktionen zu einem nicht-atomaren Transaktionsbegriff führen; vgl. dazu etwa NETT (1988), S. 59f.

94) Vgl. zum zustandstransformierenden Charakter von Transaktionen ALAGIC (1986), S. 173; REUTER (1987), S. 397f. u. 402; NETT (1988), S. 55; LIPECK (1989), S. 5; SCHEER (1990c), S. 83.

95) Vgl. zum "Zurücksetzen", "(backward) recovery" oder "roll back" bei Transaktionen, die zu einer Integritätsverletzung führen würden, REUTER (1987), S. 397ff. u. 408; HOHENSTEIN (1987), S. 83; MEYER-WEGENER (1987), S. 121; NETT (1988), S. 55ff. u. 60ff.; KOTZ (1989), S. 1; JEUSFELD (1990), S. 10.

96) Vgl. zum konsistenz- oder integritätswahrenden Charakter von Transaktionen BERNSTEIN (1980), S. 18(ff.); ALAGIC (1986), S. 173; HOHENSTEIN (1986), S. 205 u. 207; REUTER (1987), S. 396ff.; HOHENSTEIN (1987), S. 79; NETT (1988), S. 55 u. 57f.; ASCHE (1988), S. 20; ESTER (1989), S. 17; KOTZ (1989), S. 1; JABLONSKI (1990), S. 103; SCHEER (1990c), S. 83; VON ZIMMERMANN (1990), S. 256f.; SCHEER (1991d), S. 174; WECK (1991f), S. 141.

97) Vgl. zu dieser Einschränkung die Erläuterung der relativen Integritätswahrung.

98) Vgl. zu solchen recovery-Mechanismen NETT (1988), S. 60ff.; KOTZ (1989), S. 46ff., insbesondere S. 57ff.

99) Vgl. zur internen Sequentialität von Transaktionen ALAGIC (1986), S. 173; REUTER (1987), S. 404; NETT (1988), S. 58.

100) Vgl. zur Parallelität (Nebenläufigkeit) von transaktionsorientiert gestalteten Informationsverarbeitungssystemen BERNSTEIN (1980), S. 18ff.; ALAGIC (1986), S. 175; REUTER (1987), S. 402 u. 406ff.; MEYER-WEGENER (1987), S. 121ff.; MEYER-WEGENER (1988), S. 47 u. 49f.; NETT (1988), S. 55; ASCHE (1988), S. 20.

101) Vgl. zur möglichen Identität von Referenz- und Folgezustand bei erfolgreichen Transaktionsausführungen REUTER (1987), S. 404.

102) Vgl. HOHENSTEIN (1986), S. 206; HOHENSTEIN (1987), S. 80ff.; EHRICH (1988), S. 121.

103) Übergangsoperationen erweisen sich jedoch insofern als strukturreicher, als sie auch noch durch ihre Informationsbereiche determiniert sein können.

104) Vgl. zur zeitlichen Ausdehnung von Transaktionen HOHENSTEIN (1986), S. 205; HOHENSTEIN (1987), S. 79; REUTER (1987), S. 405.

105) Vgl. MERTENS (1986a), S. 324ff.; SCHEER (1987b), S. 174f.; SCHOLZ (1988), S. 165ff.; MEYER-WEGENER (1988), S. 47; KOTZ (1989), S. 180.

106) Vgl. MERTENS (1986a), S. 323f.; SCHOLZ (1988), S. 166.

107) Vgl. ZELEWSKI (1986a), S. 567ff. u. 830; SPUR (1988c), S. 12, wo von einer "datengetriebenen Fabrik" gesprochen wird.

108) Vgl. zur ereignisgesteuerten Transaktionsverarbeitung und ihrer Bezugnahme auf das Triggerkonzept TSICHRITZIS (1982), S. 464ff.; LUM (1982), S. 338; JARKE (1984), S. 67; ZELEWSKI (1986a), S. 568ff., 830 u. 834; SCHEER (1990a), S. 22ff.; SCHEER (1990c), S. 60ff.
Allerdings braucht keineswegs jeder Ereigniseintritt im Produktionssystem die Ausführung eines Arbeitsgangs zu triggern. Andernfalls drohte die früher thematisierte Nervosität von PPS-Systemen. Vielmehr lassen sich Informationen über Ereignisgeschehnisse im gesteuerten Produktionssystem vorübergehend in einer Aktionsdatei sammeln. Durch die Vorgabe von Schwellenwerten können Bedingungen spezifiziert werden, bei deren Erfüllung produktionssteuernde Maßnahmen ausgelöst werden. Dabei können dann die Informationen, die bis dahin in der Aktionsdatei zwischengespeichert wurden, im Stapelbetrieb abgearbeitet werden. Daher erfolgt die Transaktionsverarbei-

tung auf der Basis des Triggerkonzepts zumeist nicht als reine Ereignissteuerung, sondern als ereignisorientierte Mischung zwischen ereignisinduzierter und stapelweiser Informationsverarbeitung. SCHEER (1990c), S. 63, spricht in diesem Zusammenhang von einer "asynchronen Online-Verarbeitung". Eine reine Ereignissteuerung wird dagegen von SCHEER (1991d), S. 113f., 119, 126 u. 129f., skizziert.

109) Vgl. die Anmerkungen zur deklarativen Semantik von Prädikaten.

110) Diese Prozeduralität kommt allen Übergangsoperationen tr_v zu, die aus dem allgemeinen Übergangsschema ÜS abgeleitet sind. Diese Übergangsoperationen werden später als Transaktionen den Transitionen von Synthetischen Netzen zugeordnet. Daher erhalten auch die Transitionen Synthetischer Netze einen prozeduralen Charakter. Diese prozedurale Facette von Transitionen wird im Rahmen der Netzliteratur nur selten explizit behandelt. Zu den wenigen Ausnahmen gehört RAZOUK (1985c), S. 12. Dort wird ausdrücklich auf kleine Programmodule (small program segment) hingewiesen, die sich den Transitionen eines Netzes zuordnen lassen. Die Programmodule stellen Softwareprozeduren dar, die beim Schalten ihrer Transitionen ausgeführt werden und dabei Veränderungen im Netzmodell bewirken.

Da sich die Softwareprozeduren in verschiedensten Programmiersprachen implementieren lassen, wurde oben für die Darstellung des Übergangsschemas eine Pseudocode-Notation gewählt. Der Pseudocode läßt sich als eine abstrakte Repräsentation der gemeinsamen Schnittmenge fast aller höherer Programmiersprachen auffassen. Daher erfolgt die konzeptionelle Formulierung des Übergangsschemas zwar im Hinblick auf seine spätere informationstechnische Implementierung. Aber eine spezielle Implementierungsumgebung - z.B. eine spezielle Programmiersprache - wird dabei noch nicht vorausgesetzt. Dieser Hinweis erlangt insofern Bedeutung, als an früherer Stelle eine deutliche Abgrenzung von anderen Übergangsschemata erfolgte, die - zumindest implizit - implementierungsabhängig formuliert sind.

111) Für die Formulierung der operationalen Semantik wurde eine prozedurale Basis gewählt, weil sich die operationsspezifizierenden Testbedingungen, Bestimmungsprädikate und Prozeduren stets mit Hilfe der Automatischen Informationsverarbeitung implementieren lassen. Der Verf. sieht auch keine wirksame Alternative, operationale Semantiken ohne Bezugnahme auf prozedurale Konzepte zu formulieren. Es könnte zwar der Einwand erhoben werden, operationale Semantiken ließen sich auch mit der Hilfe nonprozeduraler, nämlich deklarativer (Programmier-)Sprachen spezifizieren. Dieses Argument beruht jedoch auf einem Mißverständnis. Diese Sprachen erweisen sich zwar an ihrer objektsprachlichen Oberfläche als nonprozedural, indem nur deklarativ definierte Ausdrücke zugelassen werden. Jede wirksame Anwendung eines deklarativen objektsprachlichen Konstrukts setzt allerdings eine Implementierungsumgebung voraus, die ihrerseits die deklarative durch eine prozedurale Semantik ersetzt, um die deklarierten Konstrukte effizient ausführen zu können. Ein treffliches Beispiel bietet die "deklarative" Programmiersprache PROLOG, deren Implementierungsvarianten stets auf einer prozeduralen Semantik für die Abarbeitung von PROLOG-Konstrukten beruhen. Hierauf wurde bereits mehrfach hingewiesen. Der Verf. kennt keine Implementierung einer deklarativ definierten Objektsprache, die ohne prozedurale Interpretationen der objektsprachlichen Konstrukte auskäme.

112) Die zeitliche Differenz zwischen dem Anweisungsinput, der *vor* einer Anweisungsausführung vorliegt, und dem Anweisungsoutput, der *nach* der Anweisungsausführung gegeben ist, läßt sich nicht deklarativ ausdrücken.

113) Vgl. dazu die Anmerkungen zum Ausschluß von Fernwirkungen.

114) Diese Operationen, die Übergangsoperationen in einem prädikatenlogischen Objektmodell darstellen, sind von den früher definierten Operationen der SIG-Algebren zu unterscheiden. Auf deren wechselseitiges Verhältnis wird an anderer Stelle näher eingegangen.

115) Der Begriff des Übergangsschemas ist in seiner allgemeinsten Formulierung "$ÜS=abstrakt(\{(tr_v,vb_c): FAK_r\,[tr_v,vb_c\rangle FAK_f\})$" so offen formuliert, daß er keine inhaltliche Festlegung auf einen bestimmten Typ konkreter Übergangsoperationen involviert. Damit kann er für die Definition beliebiger operationaler Semantiken herangezogen werden. Einzige Voraussetzung hierbei ist, daß sich diese Semantiken auf metasprachliches Wissen über die Gültigkeit von Formeln in prädikatenlogisch formulierten Objektmodellen beziehen. Der gesamte formale Komplex aus einer objektsprachlichen Syntax für die prädikatenlogische Modellformulierung und aus einer metasprachlichen operationalen Semantik wird als eine formale operationale Logik für die Objektmodellierung bezeichnet.
Gegenüber der o.a. allgemeinsten Darstellung des Übergangsschemas stellt die eingangs vorgelegte Definition eines Übergangsschemas bereits eine spezielle Schemaausprägung dar. Sie ist speziell darauf zugeschnitten, die Schaltregeln des Petrinetz-Konzepts formal präzise erfassen zu können. Allerdings schätzt der Verf. sie immer noch als so allgemein ein, daß sie auch Übergangsoperationen aus anderen Konzepten auszudrücken vermag. Dieser Aspekt wird in der vorliegenden Arbeit jedoch nicht weiter vertieft. Darüber hinaus könnten alternative Übergangsschemata untersucht werden, die vom hier vorgestellten Übergangsschema abweichen. Es wäre interessant zu analysieren, ob solche formal unterschiedlichen Übergangsschemata auch materiell signifikante Konsequenzen hinsichtlich der Güte oder Effizienz von prädikatenlogisch basierten Objektmodellierungen besitzen. Die prädikatenlogische Modellierungsbasis müßte allerdings beibehalten werden, um Differenzen der Modellierungsgüte bzw. -effizienz dem

Wechsel zwischen alternativen Übergangsschemata zurechnen zu können. In dieser Voraussetzung sieht der Verf. jedoch keine erhebliche Einschränkung, weil die Ausdruckskraft der Prädikatenlogik hinreicht, um alle betriebswirtschaftlich relevanten Modellierungsobjekte zufriedenstellend zu beschreiben.

116) Die Inklusionstests liegen immer eindeutig expliziert vor.

117) Diese Mindestanforderung folgt aus der Definition von Übergangsoperationen durch Veränderungen als Veränderungen von Faktenmengen. Die einzigen mengenbezogenen Veränderungsmaßnahmen, die im Übergangsschema ÜS vorgesehen sind, stellen die Entfernungs- und die Ergänzungsanweisung dar. Daher ist eine von beiden in jeder Operationsdefinition obligatorisch. Damit korrespondiert auch die oben erfolgte Festlegung, daß der Wirkungsbereich einer Übergangsoperation niemals leer sein darf.

118) Ob die Operation in diesem Fall eine Bestimmungsgleichung enthält, hängt davon ab, ob mindestens eine Multimenge $MTAN_{u,v}$ für ein Ausgangsprädikat $Prä_u(st_1,...,st_{Ku})$ aus dem Nachbereich NB_v mindestens ein Prädikatsvorkommnis $prä_u(te_1,...,te_{Ku})$ enthält, das in seinem Argument mindestens eine Ausgangsvariable X_k umfaßt. Falls diese Bedingung erfüllt ist, dann kann die Operation nicht nur, sondern sie *muß* dann sogar je eine Bestimmungsgleichung für jede der voranstehend charakterisierten Variablen X_k umfassen. Allerdings lassen sich diese Bestimmungsgleichungen einsparen, falls die Ausgangsvariablen jeweils mit einer Einflußvariable identifiziert werden.

119) Es besteht daher eine auffällige Asymetrie bezüglich der voranstehenden Entfernungsanweisung. Der Grund hierfür liegt in Informationsprädikaten, die zur Formulierung von Nebenbedingungen herangezogen werden, um die Zulässigkeit einer Operationsausführung einzuschränken, ohne jedoch das Ergebnis der Operationsausführung zu beeinflussen. Solche Informationsprädikate setzen Inklusionstests voraus, ohne jedoch Bestimmungsgleichungen nach sich zu ziehen. Die hierdurch bedingte Asymetrie setzt sich in den nachfolgenden Vereinbarungen für Übergangsschemata partiell fort.

120) Das "oder" wird - wie es in dieser Arbeit stets der Fall ist - im Sinne des inklusiven logischen "oder" gebraucht. Also können sowohl eine Bestimmungsgleichung als auch ein Inklusionstest für Informationsprädikate vorliegen.

121) Der Sonderfall, in dem eine Übergangsoperation nur mindestens eine Entfernungs- (Ergänzungs-), aber keine Ergänzungsanweisung (Entfernungsanweisung) besitzen, bedeuten eine reine Fakteneliminierung (Faktenerzeugung). Solche Operationen wurden bereits als Reduzierungsoperationen (Generierungsoperationen) angesprochen.

122) KRÄMER (1987a), S. 272, spricht in einem analogen Zusammenhang von einem Ereignisschema ("event schemes").

123) Daher lassen sich Übergangsoperationen tr_v im allgemeinen Fall *nicht* als Funktionen auffassen. Sie verletzen die konstitutive Funktionseigenschaft der Rechtseindeutigkeit. Erst der Schaltakt $sa_{v,c}=(tr_v,vb_c)$ schafft durch die zusätzliche Auswahl einer Variablenbindungsfunktion vb_c die Rechtseindeutigkeit von Funktionen. Daher konnten auch Schaltakte $sa_{v,c}$ als Übergangsfunktionen $üf_{v,c}$ reformuliert werden.

124) Etablierte Netzdefinitionen beruhen, wenn sie als Übergangsoperationen aus dem o.a. Übergangsschema reformuliert werden, allenfalls auf Posttests für die Faktenmengen der Prädikatssymbole aus dem Nachbereich einer Übergangsoperation. Ein solcher Posttest fällt an, wenn die Faktenmenge eines Ausgangsprädikats infolge endlicher Markenkapazität der prädikatzugehörigen Stelle im Petrinetz eine Höchstanzahl von Fakten nicht überschreiten darf. Aufgrund von Symmetrieüberlegungen enthält das Übergangsschema ÜS aber auch zusätzlich die Möglichkeit von Posttests für Eingangsprädikate. Sie lassen sich z.B. nutzen, um Mindestkapazitäten für die Stellen von Eingangsprädikaten einzuführen. Dann ließe sich fordern, daß durch das Schalten von Transitionen die Extensionen der Eingangsprädikate nicht so weit reduziert werden dürfen, daß die Anzahl der faktenrepräsentierenden Marken die vorgegebenen Mindestkapazitäten unterschreiten würde. Die Möglichkeit von Markenmindestkapazitäten wurde bereits im Kontext von Stelle/Transition-Netzen angesprochen. Sie spielt aber in dieser Arbeit keine Rolle.

125) Andernfalls drohte stets die Gefahr, ein Konzept aufgeben oder zumindest anpassen zu müssen, sobald die konkreten Bedingungen seiner Implementierung variieren. In der Stabilität implementierungsunabhängiger Konzeptformulierungen trotz variabler Implementierungsbedingungen sieht der Verf. einen erheblichen Vorzug. Er kann z.B. dadurch motiviert werden, daß bei variierenden Implementierungsbedingungen Ressourcenverbräuche für die Konzeptanpassung eingespart werden können. Dabei muß allerdings vorausgesetzt werden, daß die zusätzliche Ressourcenbindung für eine abstrakte, implementierungsunabhängige Konzeptformulierung gegenüber einer konkreteren, auf bestimmte Implementierungsprämissen gestützte Konzeptformulierung kleiner ausfällt als die vorgenannten Ressourcenverbräuche bei wechselnden Implementierungsbedingungen. Eine genauere Abwägung dieser beiden gegenläufigen Aspekte des Ressouceneinsatzes für Konzeptformulierungen steht außerhalb des Erkenntnisrahmens dieser Arbeit. Die Abwägung wird hier durch die o.a. normative Basisprämisse zugunsten der Bevorzugung implementierungsunabhängiger Konzeptformulierungen ersetzt.

126) Die implizite Implementierungsabhängigkeit konventioneller Definitionsschemata für Transitionen äußert sich vor allem in der Transitionsbeschriftung von Prädikat/Transition-Netzen. Darin werden gewöhnlich zwei konzeptio-

nell verschiedene Aspekte von Übergangsoperationen konfundiert. Der erste Aspekt erstreckt sich auf die Überprüfung, ob die Marken, die von den Eingangsstellen einer Transition durch deren Schalten abgezogen werden sollen, bestimmte Bedingungen erfüllen. Diese Marken repräsentieren auf den Eingangsstellen einer Transition Fakten der stellenzugehörigen Prädikatssymbole. Der Abzug einer solchen Marke entspricht daher dem Entfernen des repräsentierten Fakts aus der aktuellen Faktenmenge. Der Aspekt der Bedingungsprüfung wird in dem Übergangsschema ÜS durch den Haupttest für Eingangsprädikate erfaßt. Der zweite Aspekt betrifft diejenigen Marken, die auf Ausgangsstellen einer Transition abgelegt werden sollen und während des Schaltens bestimmte Eigenschaften zugeordnet erhalten. Dies wird im Übergangsschema ÜS durch die Bestimmungsgleichungen für die Ausgangsvariablen berücksichtigt. Sie determinieren die Markeneigenschaften, indem die markenzugehörigen Variablen durch eine Variablenbindungsfunktion auf formale Objekte als konkrete Eigenschaftsausprägungen abgebildet werden.
Die zwei unterschiedlichen Komponenten des Übergangsschemas - der Haupttest und die Bestimmungsgleichungen - entsprechen der grundsätzlichen Verschiedenartigkeit von Bedingungsprüfungen und Variablenbindungen. In gewöhnlichen Prädikat/Transition-Netzen werden diese beiden unterschiedlichen Aspekte jedoch nicht auseinander gehalten. Statt dessen werden undifferenzierte Transitionsbeschriftungen zugelassen, die sowohl die Funktion eines Haupttestprädikats als auch einer Bestimmungsgleichung erfüllen können. Diese erstaunliche Konfundierung von Bedingungsprüfungen und Variablenbindungen könnte aus der impliziten Annahme resultieren, die Netzimplementierung erfolge auf der Basis des prädikatenlogischen Unifizierungskonzepts. Denn in einem solchen Unifizierungskontext werden Bedingungsprüfungen und Variablenbindungen in der gleichen Weise behandelt. In beiden Fällen wird nach einer zulässigen Unifizierung der Variablen - in den Bedingungsformeln bzw. in den Bestimmungsgleichungen - gesucht. Aus *dieser* Unifizierungsperspektive besteht tatsächlich kein Unterschied bei der Behandlung von Bedingungsprüfungen und Variablenbindungen. Auch der Verf. stützt sich intensiv auf das prädikatenlogische Unifizierungskonzept. Er hat es sowohl für die Realisierung von SIG-Termauswertungen angeregt als auch - später - der Implementierung von Synthetischen Netzen durch einen Dialekt der Programmiersprache PROLOG zugrundegelegt. Dennoch hält er daran fest, das allgemeine Übergangsschema für Übergangsoperationen von solchen Implementierungsentscheidungen frei zu halten. Genau dies geschieht durch die oben wohlunterschiedenen Komponenten des Haupttests für Bedingungsprüfungen einerseits sowie der Bestimmungsanweisung für Eigenschaftszuordnungen andererseits.

Literaturverzeichnis zu Band 4

<u>Vorbemerkungen:</u>

- Jedes Werk wird durch die Angabe eines Referenztitels (1. Zeile) und durch seine bibliographischen Angaben (folgende Zeilen) aufgeführt. In den Quellenangaben dieser Arbeit wird immer auf den Referenztitel Bezug genommen.

- Die Referenztitel bestehen nur aus den Autorennachnamen und den Erscheinungsjahren, solange hierdurch eine eindeutige Identifizierung der jeweils zugehörigen Werke möglich ist. Andernfalls dienen zusätzliche - abgekürzte - Autorenvornamen oder alphabetische Zusätze zu den Erscheinungsjahren der eindeutigen Identifizierung.

- Um eine einheitliche Quellenangabe in allen Bänden des Projekts PEMOPS zu gewährleisten, bezieht sich die eindeutige Identifizierung durch Autorenvornamen und alphabetische Zusätze zu den Erscheinungsjahren auf den Gesamtkorpus aller verarbeiteten Quellen. Daher kann es dazu kommen, daß innerhalb eines Bandes Lücken klaffen. Sie resultieren daraus, daß die scheinbar fehlenden Quellen im Gesamtkorpus zwar enthalten sind, aber im jeweils betroffenen Band nicht verwendet wurden.

- Die Titel fremdsprachlicher Werke werden grundsätzlich in der Notation des Originals wiedergegeben. Allerdings gelten drei Ausnahmen:
  - Titel, die sich nicht mit dem deutschsprachigen Alphabet ausdrücken lassen, werden in ihrer lautsprachlichen Umschreibung durch das deutschsprachige Alphabet wiedergegeben. Dies gilt insbesondere für Werke mit chinesischen oder kyrillischen Schriftzeichen.
  - Falls die Titel im Original durchgängig mit Großbuchstaben dargestellt werden, erfolgt hier eine Notation in der jeweils sprachspezifischen Groß-/Kleinschreibung von Titeln. Dies trifft vor allem auf anglophone Werke zu, in deren Titeln die jeweils sinnbestimmenden Worte durch Großbuchstaben eingeleitet werden.
  - Accents und andere diakritische Zeichenbestandteile, die nicht im deutschsprachigen Alphabet enthalten sind, werden grundsätzlich ausgelassen.

- In das Literaturverzeichnis wurden alle Quellen aufgenommen, auf die in den Anmerkungen zum laufenden Text verwiesen wurde.

- Weitere Publikationen, die sich auf die Thematik des Petrinetz-Konzepts beziehen, aber in den vorgenannten Quellen nicht angesprochen wurden, finden sich im Band 10 des Projekts PEMOPS zur Petrinetz-Literatur.

- Die Literaturauswertung wurde 1992 abgeschlossen (vgl. das Vorwort in Band 1).

Abrahamson (1979)
Abrahamson,K.: Modal Logic of Concurrent Nondeterministic Programs; in: Kahn,G. (Hrsg.): Semantics of Concurrent Computation, Proceedings of the International Symposium, 2.-4.07.1979 in Evian, Lecture Notes in Computer Science 70, Berlin - Heidelberg - New York 1979, S. 21-33.

Ackermann,W. (1957)
Ackermann,W.: Philosophische Bemerkungen zur mathematischen Logik und zur mathematischen Grundlagenforschung; in: Ratio, (1.) Jg. 1957/58, Heft 1, S. 1-20.

Ackoff (1962)
Ackoff,R.L.: Scientific Method - optimizing applied research decisions, New York - London - Sydney 1962.

Aida (1983)
Aida,H.; Tanaka,H.; Moto-oka,T.: A Prolog Extension for Handling Negative Knowledge; in: New Generation Computing, Vol. 1 (1983, S. 87-91.

Alagic (1986)
Alagic,S.: Relational Database Technology, New York - Berlin - Heidelberg ... 1986.

Aldinger (1985a)
Aldinger,L.: Leitstandunterstützte kurzfristige Fertigungssteuerung bei Einzel- und Kleinserienfertigung, Berlin - Heidelberg - New York - Tokyo 1985.

Alexy (1978a)
Alexy,R.: Theorie der juristischen Argumentation - Die Theorie des rationalen Diskurses als Theorie der juristischen Begründung, Frankfurt 1978.

Alisch (1987)
Alisch,L.-M.: Bemerkungen zur strukturalistischen Theorienlogik und Logik der Praxis; in: Müller,N.; Stachowiak,H. (Hrsg.): Problemlösungsoperator Sozialwissenschaft - Anwendungsorientierte Modelle der Sozial- und Planungswissenschaften in ihrer Wirksamkeitsproblematik, Band I, Stuttgart 1987, S. 236-290.

Allen,J. (1983)
Allen,J.F.: Maintaining Knowledge about Temporal Intervals; in: Communications of the ACM, Vol. 26 (1983), S. 832-843.

Allen,J. (1984)
Allen,J.F.: Towards a General Theory of Action and Time; in: Artificial Intelligence, Vol. 23 (1984), S. 123-154.

Appelrath (1983)
Appelrath,H.-J.: Konzepte der Wissensbereitstellung in Expertensystemen: Inferenzmechanismen auf relationalen Datenbanken, Dissertation, Universität Dortmund, Dortmund 1983.

Appelt (1980)
Appelt,D.E.: A Planner for Reasoning about Knowledge and Action; in: o.V.: Proceedings of the First Annual National Conference on Artificial Intelligence, AAAI-80, 18.-21.08.1980 in Stanford, o.O. (Menlo Park), 1980, S. 131-133.

Apt (1982)
Apt,K.R.; van Emden,M.H.: Contributions to the Theory of Logic Programming; in: Journal of the Association for Computing Machinery, Vol. 29 (1982), S. 841-862.

Arens (1985)
Arens,B.: Die Non-standard Analysis: Eine Rehabilitierung des Unendlichkleinen in den Grundlagen der Mathematik; in: Zeitschrift für allgemeine Wissenschaftstheorie, Bd. 16 (1985), S. 147-150.

Arlabosse (1986)
Arlabosse,F.: Reasoning with Time and about Time: Some Aspects of Temporal Logic; in: Mamdani,A.; Efstathiou,J. (Hrsg.): Expert Systems and Optimisation in Process Control, Based on papers delivered at a seminar, 3.-5.12.1985, Aldershot - Brookfield 1986, S. 215-219.

Asche (1988)
Asche,W.: Neues Zentrum für Online-Transaktionsverarbeitung - Manipulationen an und mit Datenbanken dominieren - Das Netzwerk hat Einfluß auf Durchsatzrate und Antwortzeit; in: VDI nachrichten, 42. Jg. (1988), Nr. 46, S. 20.

Astesiano (1988)
Astesiano,E.; Giovini,A.; Reggio,G.: Data in a Concurrent Environment; in: Vogt,F.H. (Hrsg.): Concurrency 88, International Conference on Concurrency, 18.-19.10.1988 in Hamburg, Proceedings, Lecture Notes in Computer Science 335, Berlin - Heidelberg - New York ... 1988, S. 140-159.

Atanassov (1989)
Atanassov,K.; Atanassova,L.; Dimitrov,E.; Gargov,G.; Kazalarski,I.; Marinov,M.; Petkov,S.: Generalized Nets and Expert Systems; in: Kleinschmidt,P.; Radermacher,F.J.; Schweitzer,W.; Wildemann,H. (Hrsg.): Methods of Operations Research 59, XII. Symposium on Operations Research, 09.-11.09.1987 in Passau, Proceedings, Frankfurt 1989, S. 303-310.

Attardi (1981)
Attardi,G.; Simi,M.: Consistency and Completeness of OMEGA, a Logic for Knowledge Representing; in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, Vol. 1, o.O. (Menlo Park) 1981, S. 504-510.

Azema (1984)
Azema,P.; Juanole,G.; Sanchis,E.; Montbernard,M.: Specification and Verification of Distributed Systems Using PROLOG Interpreted Petri Nets; in: o.V.: Proceedings of the IEEE Software Engineering Conference 1984, New York 1984, S. 510-518.

Bachem (1980)
Bachem,A.: Komplexitätstheorie im Operations Research; in: Zeitschrift für Betriebswirtschaft, 50. Jg. (1980), S. 812-844.

Back,R. (1986)
Back,R.J.R.: A Computational Interpretation of Truth Logic; in: Synthese, Vol. 66 (1986), S. 15-34.

Baier,W. (o.J.)
Baier,W.; Niemeier,J.: OFFICE-NET - ein rechnergestütztes Werkzeug zur Analyse und Planung von Arbeitssystemen im Büro, Paper, Fraunhofer-Institut für Arbeitswirtschaft und Organisation, Stuttgart o.J.

Bancilhon (1986)
Bancilhon,F.; Khoshafian,S.: A Calculus for Complex Objects; in: o.V.: Proceedings of the Fifth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, 24.-26.03.1986 in Cambridge (Massachusetts), New York 1986, S- 53-59.

Bandini (1988)
Bandini,S.; Carniel,B.; Pomello,L.: Narrative Context Model for Representing Deep Knowledge; in: Trappl,R. (Hrsg.): Cybernetics and Systems'88, Proceedings of the Ninth European Meeting on Cybernetics and Systems Research, 05.-08.04.1988 in Wien, Dordrecht - Boston - London ... 1988, Part 2, S. 869-876.

Barker (1990)
Barker,J.R.: Novel Logic and Architectures for Molecular Computing; in: Eckmiller,R.; Hartmann,G.; Hauske,G. (Hrsg.): Parallel Processing in Neural Systems and Computers, International Conference on Parallel Processing in Neural Systems and Computers, 19.-21.03.1990 in Düsseldorf, Amsterdam - New York - Oxford ... 1990, S. 519-524.

Barr,A. (1981)
Barr,A.; Feigenbaum,E.A. (Hrsg.): The Handbook of Artificial Intelligence, Vol. 1, Stanford - Los Altos 1981.

Barr,R. (1989)
Barr,R.S.; Christiansen,M.G.: A Parallel Auction Algorithm: A Case Study in the Use of Parallel Object-Oriented Programming; in: Sharda,R.; Golden,B.L.; Wasil,E.; Balci,O.; Stewart,W. (Hrsg.): Impacts of Recent Computer Advances on Operations Research, New York - Amsterdam - London 1989, S. 23-32.

Barth,G. (1987)
Barth,G.: Prolog: Programmieren auf der Basis von Logik; in: Informationstechnik, 29. Jg. (1987), S. 217-226.

Bast (1985)
Bast,H.: Resolution auf Kleinrechnern: "Taschenrechner" und 1-Klauseln-Verfahren auf einem Z80-System, Informatik Berichte Nr. 45, Institut für Informatik, Universität Bonn, Bonn 1985.

Battiston (1988)
Battiston,E.; De Cindio,F.; Mauri,G.: OBJSA Nets: a class of high-level nets having objects as domains; in: Rozenberg,G. (Hrsg.): Advances in Petri Nets 1988, Lecture Notes in Computer Science 340, Berlin - Heidelberg - New York ... 1988, S. 20-43.

Bauer,F. (1989)
Bauer,F.L.: 100 Jahre Peano-Zahlen; in: Informatik-Spektrum, Bd. 12 (1989), S. 340-341.

Baumeister,H. (1987)
Baumeister,H.; Ganzinger,H.; Heeg,G.; Rüger,M.: Smalltalk-80; in: Informationstechnik, 29. Jg. (1987), S. 241-251.

Baur (1976)
Baur,W.: Zeitlich beschränkte Turingmaschinen und polynomiale Reduktion; in: Specker,E.; Strassen,V. (Hrsg.): Komplexität von Entscheidungsproblemen, Lecture Notes in Computer Science 43, Berlin - Heidelberg - New York 1976, S. 11-19.

Beckstein (1988a)
Beckstein,C.: Zur Logik der Logik-Programmierung - Ein konstruktiver Ansatz -, Dissertation, Universität Erlangen-Nürnberg, Erlangen 1988. (Auch erschienen als: Informatik-Fachberichte 199, Berlin - Heidelberg - New York ... 1988.)

Beetz (1986)
Beetz,M.: Wissensrepräsentationstechniken und Inferenzmaschinen - ein klassifizierender Überblick, Forschungsbericht FB-TA-85-14, WISDOM-Verbundprojekt/Triumph-Adler AG, o.O. (Nürnberg) 1986.

Behmann (1922)
Behmann,H.: Beiträge zur Algebra der Logik, insbesondere zum Entscheidungsproblem; in: Mathematische Annalen, 86. Bd. (1922), S. 163-229.

Beierle (1988a)
Beierle,C.: Semantische Aspekte des algebraischen Programmierens; in: Informationstechnik, 30. Jg. (1988), S. 446-455.

Beierle (1988b)
Beierle,C.: PROTOS-L: Design and Implementation; in: Appelrath,H.-J.; Cremers,A.B.; Schiltknecht,H. (Hrsg.): Prolog Tools for Building Expert Systems: First Work Shop, 12.-14.09.1988 in Morcote, Basel 1988, S. 159-181.

Ben-Ari (1981)
Ben-Ari,M.; Manna,Z.; Pnueli,A.: The Temporal Logic of Branching Time; in: o.V.: Conference Record of the 8th Annual ACM Symposium on Principles of Programming Languages, New York 1981, S. 164-176.

Bernstein (1980)
Bernstein,P.A.; Shipman,D.W.; Rothnie,J.B.: Concurrency Control in a System for Distributed Databases (SDD-1); in: ACM Transactions on Database Systems, Vol. 5 (1980), S. 18-51.

Berthold (1983)
Berthold,H.J.: Aktionsdatenbanken in einem kommunikationsorientierten EDV-System; in: Informatik-Spektrum, Bd. 6 (1983), S. 20-26.

Berzheim (1975a)
Berzheim,H.: "Fernwirkungsfreie" Strukturierte Programmierung - Ein Neuer Kalkül für die Softwaretechnologie; in: Online, 13. Jg. (1975), Nr. 3, S. 118-121.

Berzheim (1975b)
Berzheim,H.: "Fernwirkungsfreie" Strukturierte Programmierung - Ein neuer Kalkül für die Softwaretechnologie Teil II; in: Online, 13. Jg. (1975), Nr. 4, S. 234-240.

Bezem (1988)
Bezem,M.: Logic Programming and PROLOG; in: CWI Quarterly, Vol. 1 (1988), No. 3, S. 15-29.

Bibel (1982a)
Bibel,W.: Automated Theorem Proving, Braunschweig - Wiesbaden 1982.

Bibel (1982b)
Bibel,W.: Deduktionsverfahren; in: Bibel,W.; Siekmann,J.H. (Hrsg.): Künstliche Intelligenz, Frühjahrsschule Teisendorf, 15.-24.03.1982 in Teisendorf, Informatik-Fachberichte 59, Berlin - Heidelberg - New York 1982, S. 99-140.

Bibel (1984)
Bibel,W.: Automatische Inferenz; in: Retti,J.; Bibel,W.; Buchberger,B.; Buchberger,E.; Horn,W.; Kobsa,A.; Steinacker,I.; Trappl,R.; Trost,H.: Artificial Intelligence - Eine Einführung, Stuttgart 1984, S. 145-167.

Bibel (1986)
Bibel,W.: A Deductive Solution for Plan Generation; in: New Generation Computing, Vol. 4 (1986), S. 115-132.

Bibel (1989)
Bibel,W.; del Cerro,L.F.; Fronhöfer,B.; Herzig,A.: Plan Generation by Linear Proofs: On Semantics; in: Metzing,D. (Hrsg.): GWAI-89, 13th German Workshop on Artificial Intelligence, 18.-22.09.1989 in Eringerfeld, Proceedings, Informatik-Fachberichte 216, Berlin - Heidelberg - New York ... 1989, S. 49-62.

Binmore (1987)
Binmore,K.: Modeling Rational Players; in: Economics and Philosophy - Part I, Vol. 3 (1987), S. 179-214.

Binmore (1988)
Binmore,K.: Modeling Rational Players - Part II; in: Economics and Philosophy, Vol. 4 (1988), S. 9-55.

Binot (1988)
Binot,J.-L.; Burgard,W.; de Zegher,I.; Donner,D.; Michaux,G.: Integration of a frame based extension into a Prolog environment; in: Appelrath,H.-J.; Cremers,A.B.; Schiltknecht,H. (Hrsg.): Prolog Tools for Building Expert Systems: First Work Shop, 12.-14.09.1988 in Morcote, Basel 1988, S. 10-28.

Bitner (1975)
Bitner,J.R.; Reingold,E.M.: Backtracking Programming Techniques; in: Communications of the ACM, Vol. 18 (1975), S. 651-655.

Blau (1973)
Blau,U.: Zur 3-wertigen Logik der natürlichen Sprache; in: Bechert,J.; Grewendorf,G.; Mayerthaler,E.; Mayerthaler,W.; Meggle,G.; Reis,M.; Schwischay,B.; Stimm,H. (Hrsg.): Papiere zur Linguistik, Nr. 4, München 1973, S. 20-96.

Blau (1985)
Blau,U.: Die Logik der Unbestimmtheiten und Paradoxien; in: Erkenntnis, Vol. 22 (1985), S. 369-459.

Bobrow (1984)
Bobrow,D.G.: If Prolog Is the Answer, What Is the Question?; in: o.V.: Fifth Generation Computer Systems 1984 - Proceedings of the International Conference, 06.-09.11.1984 in Tokyo, Tokyo - Amsterdam 1984, S. 138-148.

Böhling (1974)
Böhling,K.H.; von Braunmühl,B.: Komplexität bei Turingmaschinen, Mannheim - Wien - Zürich 1974.

Böhringer (1988)
Böhringer,B.; Chiopris,C.; Futo,I.: Wissensbasierte Systeme mit Prolog, Bonn - Reading - Massachusetts ... 1988.

Bollinger (1989)
Bollinger,T.; Hedtstück,U.; Rollinger,C.-R.: Reasoning for Text Understanding - Knowledge Processing in the 1st LILOG-Prototype; in: Metzing,D. (Hrsg.): GWAI-89, 13th German Workshop on Artificial Intelligence, 18.-22.09.1989 in Eringerfeld, Proceedings, Informatik-Fachberichte 216, Berlin - Heidelberg - New York ... 1989, S. 203-212.

Bolour (1982)
Bolour,A.; Anderson,T.L.; Dekeyser,L.J.; Wong,H.K.T.: The Role of Time in Information Processing: A survey; in: SIGART Newsletter, No. 80 (1982), S. 28-48.

Boolos (1980)
Boolos,G.; Jeffrey,R.: Computability and Logic, 2. Aufl., Cambridge - London - New York ... 1980.

Bossi (1989)
Bossi,A.; Cocco N.: Verifying correctness of logic programs; in: Diaz,J.; Orejas,F. (Hrsg.): TAPSOFT'89, Proceedings of the International Joint Conference on Theory and Practice of Software Development, 13.-17.03.1989 in Barcelona, Volume 2: Advanced Seminar on Foundations of Innovative Software Development II and Colloquium on Current Issues in Programming Languages (CCIPL), Lecture Notes in Computer Science 352, Berlin - Heidelberg - New York ... 1989, S. 96-110.

Brachman (1983a)
Brachman,R.J.; Amarel,S.; Engelman,C.; Engelmore,R.S.; Feigenbaum,E.A.; Wilkins,D.E.: What Are Expert Systems?; in: Hayes-Roth,F.; Waterman,D.A.; Lenat,D.B. (Hrsg.): Building Expert Systems, Reading - London - Amsterdam ... 1983, S. 31-57.

Brauer (1968)
Brauer,W.; Indermark,K.: Algorithmen, rekursive Funktionen und formale Sprachen, Mannheim - Wien - Zürich 1968.

Brauer (1990a)
Brauer,W.: Grenzen maschineller Berechenbarkeit; in: Informatik-Spektrum, Bd. 13 (1990), S. 61-70.

Braun,W. (1985)
Braun,W.: Konstruktive Betriebswirtschaftslehre, - Eine wissenschaftliche Einführung, Wiesbaden 1985.

Brennenstuhl (1980)
Brennenstuhl,W.: Ziele der Handlungslogik; in: Lenk,H. (Hrsg.): Handlungstheorien interdisziplinär I - Handlungslogik, formale und sprachwissenschaftliche Handlungstheorien, München 1980, S. 35-66.

Brewka (1989)
Brewka,G.: Nonmonotonic Logics - A Brief Overview; in: AI Communications, Vol. 2 (1989), S. 88-97.

Brinkmann (1989)
Brinkmann,G.: Analytische Wissenschaftstheorie - Einführung sowie Anwendung auf einige Stücke der Volkswirtschaftslehre, München - Wien 1989.

Brouwer (1913)
Brouwer,L.E.J.: Intuitionism and Formalism; in: Bulletin of the American Mathematical Society, Vol. 20 (1913/14), S. 81-96.

Brouwer (1919)
Brouwer,L.E.J.: Intuitionistische Mengenlehre.; in: Gutzmer.A. (Hrsg.): Jahresbericht der Deutschen Mathematiker-Vereinigung, 28. Bd., Leipzig 1919, S. 203-208.

Brouwer (1925)
Brouwer,L.E.J.: Über die Bedeutung des Satzes vom ausgeschlossenen Dritten in der Mathematik, insbesondere in der Funktionentheorie.; in: Journal für die reine und angewandte Mathematik, Bd. 154 (1925), S. 1-7.

Brouwer (1928)
Brouwer,L.E.J.: Intuitionistische Betrachtungen über den Formalismus; in: Sitzungsberichte der Preussischen Akademie der Wissenschaften, Jahrgang 1928, physikalisch-mathematische Klasse, Berlin 1928, S. 48-52.

Brucker (1981)
Brucker,P.: Scheduling, Wiesbaden 1981.

Bucher (1987)
Bucher,T.G.: Einführung in die angewandte Logik, Berlin - New York 1987.

Bürckert (1986)
Bürckert,H.-J.; Hewrold,A.: Unifikation; in: Informatik-Spektrum, Bd. 9 (1986), S. 186-187.

Bürckert (1987a)
Bürckert,H.-J.: Unifikationstheorie; in: Bläsius,K.H.; Bürckert,H.-J. (Hrsg.): Deduktionssysteme - Automatisierung des logischen Denkens, München - Wien 1987, S. 104-114.

Bullers (1980)
Bullers,W.I.; Nof,S.Y.; Whinston,A.B.: Artificial Intelligence in Manufacturing Planning and Control; in: AIIE Transactions, Vol. 12 (1980), S. 351-363.

Bunge (1985a)
Bunge,M.: Treatise on Basic Philosophy, Volume 7, Epistemology & Methodology III: Philosophy of Science and Technology, Part I: Formal and Physical Sciences, Dordrecht - Boston - Lancaster 1985.

Burkhard (1982a)
Burkhard,H.-D.: On Fairness in Petri Nets; in: o.V.: Berichte vom 20. Mathematischen Kolloquium in Rostock, o.O. 1982, S. 85-96.

Cardoza (1976)
Cardozy,E.; Lipton,R.; Meyer,A.R.: Exponential Space Complete Problems for Petri Nets and Commutative Semigroups: Preliminary Report; in: o.V.: Conference Record of The Eigth Annual ACM Symposium on Theory of Computing, Papers presented at the Symposium, 3.-5.05.1976 in Hershey, New York 1976, S. 50-54.

Carnap (1930a)
Carnap,R.: Die alte und die neue Logik; in: Erkenntnis, 1. Bd. (1930/31), S. 12-26.

Carnap (1931c)
Carnap,R.: Die logizistische Grundlegung der Mathematik; in: Erkenntnis, 2. Bd. (1931), S. 91-105.

Carnap (1959a)
Carnap,C.: Induktive Logik und Wahrscheinlichkeit, bearbeitet von W. Stegmüller, Wien 1959.

Carnap (1959b)
Carnap.R.: Beobachtungssprache und theoretische Sprache; in: Logica - Studia Paul Bernays Dedicata, Vol. 34 (1959), S. 32-44.

Carnap (1960a)
Carnap,R.: Einführung in die symbolische Logik, 2. Aufl., Wien 1960.

Carnap (1968)
Carnap,R.: Logische Syntax der Sprache, 2. Aufl., Wien - New York 1968.

Cerf (1972)
Cerf,V.G.: Multiprocessors, Semaphores, and a Graph Model of Computation, Dissertation, Computer Science Department, University of California, Los Angeles 1972.

Chang,C.L. (1973)
Chang,C.-L.; Lee,R.C.-T.: Symbolic Logic and Mechanical Theorem Proving, New York - London 1973.

Chang,C.L. (1976)
Chang,C.L.: DEDUCE --- A Deductive Query Language for Relational Data Bases; in: Chen,C.H. (Hrsg.): Pattern Recognition and Artificial Intelligence, Proceedings of the Joint Workshop on Pattern Recognition and Artificial Intelligence, 01.-03.06.1976 in Hyannis, New York - San Francisco - London 1976, S. 108-134.

Chang,C.L. (1979)
Chang,C.L.: Resolution Plans in Theorem Proving; in: o.V.: IJCAI-79, Proceedings of the Sixth International Conference on Artificial Intelligence, 20-23.08.1979 in Tokyo, Vol. 1, o.O. (Stanford) 1979, S. 143-148.

Charniak (1976a)
Charniak,E.: Inference and Knowledge Part 1; in: Charniak,E.; Wilks,Y. (Hrsg.): Computational Semantics - An Introduction to Artificial Intelligence and Natural Language Comprehension, Amsterdam - New York - Oxford 1976, S. 1-21.

Church (1932)
Church,A.: A Set of Postulates for the Foundation of Logic; in: Annals of Mathematics, 2nd Series, Vol. 33 (1932), S. 346-366.

Church (1936a)
Church,A.: An Unsolvable Problem of Elementary Number Theory; in: American Journal Of Mathematics, Vol. 58 (1936), S. 345-363.

Church (1936b)
Church,A.: A Note on the Entscheidungsproblem; in: The Journal of Symbolic Logic, Vol. 1 (1936), S. 40-41.

Church (1936c)
Church,A.: A Correction to A Note on the Entscheidungsproblem; in: The Journal of Symbolic Logic, Vol. 1 (1936), S. 101-102.

Church (1952)
Church,A.; Quine,W.V.: Some Theorems on Definability and Decidability; in: The Journal of Symbolic Logic, Vol. 17 (1952), S. 179-187.

Church (1956)
Church,A.: Introduction to Mathematical Logic, Princeton 1956.

Church (1976)
Church,A.: Comparison of Russels's Resolution of the Semantical Antinomies with that of Tarski; in: The Journal of Symbolic Logic, Vol. 41 (1976), S. 747-760.

Clark,K. (1978)
Clark,K.L.: Negation as Failure; in: Gallaire,H.; Minker,J. (Hrsg.): Logic and Data Bases, New York - London 1978, S. 293-322.

Clark,K. (1979)
Clark,K.L.; McCabe,F.G.: The Control Facilities of IC-PROLOG; in: Michie,D. (Hrsg.): Expert Systems in the Micro-electronic Age, Proceedings of the 1979 AISB Summer School, Edinburgh 1979, S. 122-149.

Cleave (1987)
Cleave,J.P.: Logic and Inexactness; in: Srzednicki,J.T.J. (Hrsg.): Stephan Körner - Philosophical Analysis and Reconstruction - Contributions to Philosophy, Dordrecht - Boston - Lancaster 1987, S. 137-159.

Clocksin (1981)
Clocksin,W.F.; Mellish,C.S.: Programming in Prolog, Berlin - Heidelberg - New York 1981.

Clocksin (1990)
Clocksin,W.F.; Mellish,C.S.: Programmieren in Prolog, (Übersetzung der 3. Aufl. 1987), Berlin - Heidelberg - New York ... 1990.

Codd (1970)
Codd,E.F.: A Relational Model of Data for Large Shared Data Banks; in: Communications of the ACM, Vol. 13 (1970), S. 377-387.

Codd (1971a)
Codd,E.F.: Normalized Data Base Structure: A Brief Tutorial; in: Codd,E.F.; Dean,A.L. (Hrsg.): Proceedings of the 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control, 11.-12.11.1971 in San Diego, New York 1971, S. 1-17.

Codd (1971b)
Codd,E.F.: A Data Base Sublanguage Founded on the Relational Calculus; in: Codd,E.F.; Dean,A.L. (Hrsg.): Proceedings of the 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control, 11.-12.11.1971 in San Diego, New York 1971, S. 35-68.

Coeho (1983)
Coeho,H.: PROLOG: A Programming Tool for Logical Domain Modeling; in: Sol,H.G. (Hrsg.): Processes and Tools for Decision Support, Proceedings of the Joint IFIP WG 8.3/IASA Working Conference on Processes and Tools for Decision Support, 19.-21.07.1982 in Laxenburg, Amsterdam - New York - Oxford 1982, S. 37-45.

Cohen,P. (1982)
Cohen,P.R.; Feigenbaum,E.A. (Hrsg.): The Handbook of Artificial Intelligence, Vol. III, London - Los Altos 1982.

Cohn (1987)
Cohn,A.G.: Many Sorted Logic = Unsorted Logic + Control?; in: Bramer,M.A. (Hrsg.): Research and Development in Expert Systems III, Proceedings of Expert Systems'86, the Sixth Annual Technical Conference of the British Computer Society Spezialist Group on Expert Systems, 15.-18.12.1986 in Brighton, Cambridge (Großbritannien) - London - New York ... 1987, S. 184-194.

Cohors-Fresenborg (1977)
Cohors-Fresenborg,E.: Mathematik mit Kalkülen und Maschinen, Braunschweig 1977.

Colmerauer (1983)
Colmerauer,A.; Kanoui,H.; van Caneghem,M.: Prolog, bases theoriques et developpements actuels; in: T.S.I. - Technique et Science Informatiques, Vol. 2 (1983), S. 271-311.

Commission of the European Communities (1989)
Commission of the European Communities: 1989 ESPRIT Workprogramme (European Strategic Programme for R&D in Information Technology), Brüssel 1989.

Cook,S. (1973)
Cook,S.A.: An Observation on Time-Storage Trade Off; in: o.V.: Conference Record of the 5th Annual ACM Symposium on Theory of Computing, New York 1973, S. 29-33.

Cook,S. (1983)
Cook,S.A.: An Overview of Computational Complexity; in: Communications of the ACM, Vol. 26 (1983), 401-408.

Coomann (1983)
Coomann,H.: Die Kohärenztheorie der Wahrheit - Eine kritische Darstellung der Theorie Reschers vor ihrem historischen Hintergrund, Dissertation, Universität Düsseldorf, Frankfurt - Bern - New York 1983.

Cooprider (1976)
Cooprider,L.W.: Petri Nets and the Representation of Standard Synchronizations, Department of Computer Science, Carnegie-Mellon University, Pittsburgh 1976.

Cordes (1988)
Cordes,R.; Kruse,R.; Langendörfer,H.; Rust,H.: Prolog - Eine methodische Einführung, Braunschweig - Wiesbaden 1988.

Correnz (1987)
Correnz,W.; Ingenerf,J.; Richter,M.M.: Bemerkungen über ML und seine polymorphe Typenstruktur; in: Informationstechnik, 29. Jg. (1987), S. 235-240.

Craig (1952)
Craig,W.; Quine,W.V.: On Reduction to a Symmetric Relation; in: The Journal of Symbolic Logic, Vol. 17 (1952), S. 188.

Crespi-Reghizzi (1976a)
Crespi-Reghizzi,S.; Mandrioli,D.: Some Algebraic Properties of Petri Nets; in: Alta Frequenza, Vol. 45 (1976), N. 2, S. 130-137.

Crespi-Reghizzi (1976b)
Crespi-Reghizzi,S.; Mandrioli,D.: Commutative Grammars; in: Calcolo, Vol. 13 (1976), No. 2; S. 173-189.

Dahl (1983)
Dahl,V.: Logic Programming as a Representation of Knowledge; in: Computer, Vol. 16 (1983), S. 106-111.

Daiser (1984)
Daiser,W.: Künstliche Intelligenz Forschung und ihre epistemologischen Grundlagen, Dissertation, Universität Düsseldorf, Frankfurt - Bern - New York ... 1984.

Dangelmaier (1987a)
Dangelmaier,W.; Becker,B.-D.; Himmelstoß,P.: Concepts and Experiences Buliding a Knowledge Base for Strategy Construction in Materialflow Networks; in: Retti,T.; Wichmann,K.E. (Hrsg.): Simulation in CIM and Artificial Itelligence, Proceedings of the European Simulation Multiconference ESM'87, 1987 in Wien, o.O. 1987, S. 83-87.

Dauben (1983)
Dauben,J.W.: Georg Cantor und die Mächtigkeit der Mengen; in: Spektrum der Wissenschaft, o.Jg. (1983), Heft 8, S. 112-122.

Dauchet (1988)
Dauchet,M.: Termination of Rewriting is Undecidable in the One-Rule Case; in: Chytil,M.P.; Janiga,L.; Koubek,V. (Hrsg.): MFCS'88 - Mathematical Foundations of Computer Science, Proceedings of the 13th Symposium, 29.08.-02.09.1988 in Carlsbad, Lecture Notes in Computer Science 324, Berlin - Heidelberg - New York ... 1988, S. 262-270.

Davis,M. (1958)
Davis,M.: Computability and Unsolvability, New York - Torornto - London 1958.

Davis,M. (1973a)
Davis,M.: Hilbert's Tenth Problem is Unsolvable; in: The American Mathematical Monthly, Vol. 80 (1973), S. 233-269.

Davis,M. (1973b)
Davis,M.: Hilbert's 10th Problem; in: Scientific American, Vol. 229 (1973), No. 5, S. 84-91.

Davis,M. (1983)
Davis,M.: The Prehistory and Early History of Automated Deduction; in: Siekmann,J.; Wrightson,G. (Hrsg.): Automation of Reasoning 1: Classical Papers on Computational Logic 1957-1966, Berlin - Heidelberg - New York 1983, S. 1-28.

de Kleer (1986a)
de Kleer,J.: An Assumption-based TMS; in: Artificial Intelligence, Vol. 28 (1986), S. 127-162.

de Kleer (1986b)
de Kleer,J.: Extending the ATMS; in: Artificial Intelligence, Vol. 28 (1986), S. 163-196.

Delahaye (1987)
Delahaye,J.-P.: Formal Methods in Artificial Intelligence, Oxford 1987.

Dewdney (1984)
Dewdney,A.K.: Computer-Kurzweil: Eine Computerfamilie für den fleißigen Biber ...; in: Spektrum der Wissenschaft, o.Jg. (1984), Heft 11, S. 8-16.

Digricoli (1981)
Digricoli,V.J.: The Efficacy of RUE-Resolution - Experimental Results and Heuristic Theory; in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, Vol. 1, o.O. (Menlo Park) 1981, S. 539-547.

Dilger (1979)
Dilger,W.: Theorembeweiser als Bestandteil der Problemlösungskomponente; in: Kolvenbach,M.; Lötscher,A.; Lutz,H.D. (Hrsg.): Künstliche Intelligenz und natürliche Sprache - Sprachverstehen und Problemlösen mit dem Computer, Bd. 42 der Forschungsberichte des Instituts für deutsche Sprache in Mannheim, Tübingen 1979, S. 265-288.

Dimitrovici (1990a)
Dimitrovici,C.; Hummert,U.; Petrucci,L.: The properties of algebraic net schemes in some semantics; in: o.V.: 11th International Conference on Application and Theory of Petri Nets, 27.-29.06.1990 in Paris, o.O. 1990, S. 180-203.

DIN 44300 (1972)
Fachnormenausschuß Informationsverarbeitung (FNI) im Deutschen Normenausschuß (DNA): DIN 44300 - Informationsverarbeitung - Begriffe, Berlin - Köln 1972.

DIN 44300 (1985)
Normenausschuß Informationsverarbeitungssysteme (NI) im DIN Deutsches Institut für Normung e.V.: Entwurf DIN 44300 - Informationsverarbeitung - Begriffe, Teil 1 bis 9, Berlin 1985.

DIN 5474 (1973)
Deutscher Normenausschuß - Ausschuß für Einheiten und Formelgrößen (Hrsg.): DIN 5474, Zeichen der mathematischen Logik, Berlin - Köln 1973.

Dinkelbach (1973)
Dinkelbach,W.: Modell - ein isomorphes Abbild der Wirklichkeit?; in: Grochla,E.; Szyperski,N. (Hrsg.): Modell- und computer-gestützte Unternehmungsplanung, Wiesbaden 1973, S. 151-162.

Dorn (1988a)
Dorn,J.: Der Ereignis-Prozessor; in: Dillmann,R.; Swiderski,D. (Hrsg.): WIMPEL'88, 1. Konferenz über Wissensbasierte Methoden für Produktion, Engineering und Logistik, 28.-30.06.1988 in München, Stuttgart 1988, S. 205-219.

Dorn (1988b)
Dorn,J.: Repräsentation von Abhängigkeiten durch Skripte; in: Hertzberg,J.; Günter,A. (Hrsg.): Beiträge zum 2. Workshop Planen und Konfigurieren, Arbeitspapiere der GMD 310, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1988, S. 203-225.

Dorn (1989)
Dorn,J.: Wissensbasierte Echtzeitplanung, Dissertation, Technische Universität Berlin, Braunschweig - Wiesbaden 1989.

Doyle (1980)
Doyle,J.; London,P.: A Selected Descriptor-Indexed Bibliography to the Literature on Belief Revision; in: SIGART Newsletter, No. 71 (1980), S. 7-23.

Dressler (1989b)
Dressler,O.; Freitag,H.: Truth Maintenance Systeme; in: Künstliche Intelligenz: Forschung, Entwicklung, Erfahrungen., 3. Jg. (1989), Heft 2, S. 13-19.

Duggan (1988)
Duggan,J.; Browne,J.: ESPNET: expert-system-based simulator of Petri nets; in: IEE Proceedings, Vol. 135 (1988), Part D (Control Theory and Applications), No. 4, S. 239-247.

Dummett (1963)
Dummett,M.: Die philosophische Bedeutung von Gödels Theorem; in: Ratio, 5. Bd. (1963), S. 124-137.

Dutta,A. (1984)
Dutta,A.; Basu,A.: An Artificial Intelligence Approach to Model Management in Decision Support Systems; in: Computer, Vol. 17 (1984), Heft September, S. 89-97.

Ebbinghaus (1978)
Ebbinghaus,H.-D.; Flum,J.; Thomas,W.: Einführung in die mathematische Logik, Darmstadt 1978.

Ehrich (1988)
Ehrich,H.-D.; Drosten,K.; Gogolla,M.: Towards an Algebraic Semantics for Database Specification; in: Meersman,R.A.; Sernadas,A.C. (Hrsg.): Data and Knowledge (DS-2), Proceedings of the Second IFIP 2.6 Working Conference on Database Semantics, 'Data and Knowledge' (DS-2), 03.-07.11.1986 in Albufeira, Amsterdam - New York - Oxford ... 1988, S. 119-135.

Ehrig (1984a)
Ehrig,H.: An Algebraic Specification Concept for Modules, Bericht-Nr. 84-04, Institut für Software und Theoretische Informatik, Technische Universität Berlin, Berlin 1984.

Ehrig (1984b)
Ehrig,H.; Thatcher,J.W.; Lucas,P.; Zilles,S.N.: Denotational and Initial Algebra Semantics of the Algebraic Specification Language Look, Bericht-Nr. 84-22, Institut für Software und Theoretische Informatik, Technische Universität Berlin, Berlin 1984.

Ehrig (1985a)
Ehrig,H.; Mahr,B.: Fundamentals of Algebraic Specification 1 - Equations and Initial Semantics, Berlin - Heidelberg - New York ... 1985.

Ehrig (1987)
Ehrig,H.; Parisi-Presicce,F.; Boehm,P.; Rieckhoff,C.; Dimitrovici,C.; Große-Rhode,M.: Algebraic Data Type and Process Specifications Based on Projection Spaces, Bericht-Nr. 84-8, Institut für Software und Theoretische Informatik, Technische Universität Berlin, Berlin 1987.

Elcock (1983a)
Elcock,E.W.: How Complete Are Knowledge-Representation Systems?; in: Computer, Vol. 16 (1983), S. 114-118.

Endres (1988)
Endres,E. (Hrsg.): Architecture of the GRASPIN Environment, ESPRIT Project 125 - GRASPIN, Technical Paper GRA 80/3, Siemens AG, Berlin - München 1988.

Ernst (1969)
Ernst,G.W.; Newell,A.: GPS: A Case Study in Generality and Problem Solving, New York - London 1969.

Erschler (1986)
Erschler,J.; Esquirol,P.: Decision-Aid in Job Shop Scheduling: A Knowledge Based Approach; in: o.V.: IEEE International Conference on Robotics and Automation, San Francisco 1986, S. 1651-1656.

Esser,H. (1977a)
Esser,H.; Klenovits,K.; Zehnpfennig: Wissenschaftstheorie 1: Grundlagen und Analytische Wissenschaftstheorie, Stuttgart 1977.

Essler (1982a)
Essler,W.K.: Wissenschaftstheorie I - Definition und Reduktion, 2. Aufl., Freiburg - München 1982.

Estenfeld (1987)
Estenfeld,K.: Prolog-Implementierungen - Konzepte und Realisierungen; in: Informatik-Spektrum, Bd. 10 (1987), S. 67-78.

Ester (1989)
Ester,M.: Konsistenzwerkzeuge für PROLOG-Wissensbasen, Dissertation, Technische Hochschule Zürich, Zürich 1989.

Fahrion (1989)
Fahrion,R.: Wirtschaftsinformatik - Grundlagen und Anwendungen, Heidelberg 1989.

Faidt (1989)
Faidt,K.; Flohr,S.; Bleisinger,R.: Repräsentation und Verarbeitung von zeitlichem Wissen; in: Retti,J.; Leidlmair,K. (Hrsg.): 5. Österreichische Artificial-Intelligence-Tagung, 28.-31.03.1989 in Igls, Proceedings, Informatik-Fachberichte 208, Berlin - Heidelberg - New York ... 1989, S. 303-312.

Feigenbaum (1983)
Feigenbaum,E.A.; McCorduck,P.: The Fifth Generation - Artificial Intelligence and Japan's Computer Challenge to the World, Reading - Menlo Park - London ... 1983.

Fidelak (1986a)
Fidelak,M.: Wissensdarstellung und -verarbeitung auf der Basis von Petri-Netzen, Diplomarbeit am Fachbereich Informatik/Universität Bonn, Bonn 1986. (Auch veröffentlicht als: Arbeitspapiere der GMD Nr. 225, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1986.)

Fidelak (1986b)
Fidelak,M.: Petri-Netze - Eine formale Sprache zur Wissensrepräsentation; in: Rundbrief des Fachausschusses 1.2 Künstliche Intelligenz & Mustererkennung in der Gesellschaft für Informatik, Nr. 43 (1986), S. 32-38.

Fidelak (1987a)
Fidelak,M.: Petri Nets for Logical Representation; in: Chorafas,D.N.; Rowe,A.J. (Hrsg.): 1st International Symposium on Artificial Intelligence and Expert Systems, 18.-22.05.1987 in Berlin, Proceedings, Part A, Berlin 1987, S. 95-116.

Fidelak (1988b)
Fidelak,M.; Lischka,C.; Voß,H.: Repräsentation der Dynamik technisch-physikalischer Systeme; in: Hoschka,P. (Hrsg.): Forschungsgruppe Expertensysteme - Aus der Arbeit der Forschungsgruppe Expertensysteme, Arbeitspapiere der GMD 337, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1988, 5. Beitrag.

Fikes (1971)
Fikes,R.E.; Nilsson,N.J.: STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving; in: o.V.: Advance Papers, Second International Joint Conference on Artificial Intelligence, 01.-03.09.1971 in London, o.O. (Portsmouth) 1971, S. 608-620.

Fikes (1972)
Fikes,R.E.; Hart,P.E.; Nilsson,N.J.: Some New Directions in Robot Problem Solving; in: Meltzer,B.; Michie,D. (Hrsg.): Machine Intelligence 7, Edinburgh 1972, S. 405-430.

Fischer,P. (1965)
Fischer,P.C.: On Formalisms for Turing Machines; in: Journal of the Association for Computing Machinery (ACM), Vol. 12 (1965), S. 570-580.

Fisher,E. (1985)
Fisher,E.L.: Logic-Based Factory Design; in: o.V.: Proceedings of the IEEE International Conference on Robotics and Automation, Silver Springs 1985, S. 176-181.

Fitting (1987)
Fitting,M.: Resolution for Intuitionistic Logic; in: Ras,Z.W.; Zemankova,M. (Hrsg.): Methodologies for Intelligent Systems, Proceedings of the Second International Symposium, 14.-17.10.1987 in Charlotte, New York - Amsterdam - London 1987, S. 400-407.

Fleischhack (1989)
Fleischhack,H.; Weber,A.: Rule Based Programming, Predicate Transition Nets and the Modeling of Office Procedures and Flexible Manufacturing Systems; in: o.V.: 10th International Conference on Application and Theory of Petri Nets, 28.-30.06.1989 in Bonn, o.O. 1989, Supplement to the Proceedings, S. 23-42.

Foldenauer (1990)
Foldenauer,J.: Wissensbasierte Analyse von Fließlinien - Ein Expertensystem zur Beurteilung von verketteten Prozessen, Dissertation (unter dem Titel "Ein constraint-basiertes Analysesystem für Fließfertigungen"), Universität Karlsruhe, Düsseldorf 1990.

Fonio (1987)
Fonio,H.-R.: Symbolic Execution of Nets Using Rewrite Systems, Technical Paper GMD 32/1, ESPRIT Project 125 - GRASPIN, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1987.

Forbus (1981)
Forbus,K.D.: Qualitative Reasoning About Physical Processes; in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, Vol. 1, o.O. (Menlo Park) 1981, S. 326-330.

Fraenkel (1930)
Fraenkel,A.: Die heutigen Gegensätze in der Grundlegung der Mathematik; in: Erkenntnis, 1. Bd. (1930/31), S. 286-302.

Freedman (1988a)
Freedman,P.; Malowany,A.: The Analysis and Optimization of Repetition within Robot Workcell Sequencing Problems; in: o.V.: Proceedings of the 1988 IEEE International Conference on Robotics and Automation, 24.-29.04.1988 in Philadelphia, Vol. 2, Silver Spring 1988, S. 1276-1281.

Freedman (1988b)
Freedman,P.; Malowany,A.: SAGE: A Decision Support System for the Sequencing of Operations within a Robotic Workcell; in: Decision Support Systems, Vol. 4 (1988), S. 329-343.

Frege (1879)
Frege,G.: Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens, Halle 1879.

Friedrichs,J. (1987)
Friedrichs,J.: Expertensystemansatz zum Entwurf verteilter Systeme, Mitteilung DFVLR-Mitt. 87-16, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt, zugleich Diplom-Arbeit, Institut für Flugmechanik, Technische Universität Braunschweig, Köln 1987.

Fronhöfer (1987)
Fronhöfer,B.: PLANLOG: A Language Framework for the Integration of Procedural and Logical Programming; in: o.V.: IJCAI 87, Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 23.-28.08.1987 in Mailand, o.O. (Los Altos) 1987, Vol. 1, S. 15-17.

Fuchi (1982)
Fuchi,K.: Aiming for Knowledge Information Processing Systems; in: Moto-oka,T. (Hrsg.): Fifth Generation Computer Systems, Proceedings of the International Conference on Fifth Generation Computer Systems, 19.-22.10.1981 in Tokyo, Amsterdam - New York - Oxford 1982, S. 107-120.

Fusaoka (1983)
Fusaoka,A.; Seki,H.; Takahashi,K.: A Description and Reasoning of Plant Controllers in Temporal Logic; in: Bundy,A. (Hrsg.): IJCAI-83, Proceedings of the Eigth International Joint Conference on Artificial Intelligence, 08.-12.08.1983 in Karlsruhe, Vol. 1, o.O. (Los Altos) 1983, S. 405-408.

Futo (1990)
Futo,I.; Gergely,T.: Artificial Intelligence in Simulation, New York - London - Toronto ... 1990.

Gabbay (1980)
Gabbay,D.; Pnueli,A.; Shelah,S.; Stavi,J.: On the Temporal Analysis of Fairness; in: o.V.: Conference Record of 7th ACM Symposium on Principles of Programming Languages, im Januar 1980, New York 1980, S. 163-173.

Gäng (1967)
Gäng,P.: Pragmatische Information; in: Grundlagen aus Kybernetik und Geisteswissenschaft, Bd. 8 (1967), S. 77-90.

Gallaire (1986)
Gallaire,H.: Merging Objects and Logic Programming: Relational Semantics; in: o.V.: proceedings aaai-86, fifth annual national conference on artificial intelligence, 11.-15.08.1986 in Philadelphia, Vol. 2: engineering, Los Altos 1986, S. 754-758.

Garey (1979)
Garey,M.R.; Johnson,D.S.: Computers and Intractability - A Guide to the Theory of NP-Completeness, San Francisco 1979.

Genrich (1973b)
Genrich,H.J.: Formale Eigenschaften des Entscheidens und Handelns, Interner Bericht 09/73-11-29, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1973.

Genrich (1976b)
Genrich,H.J.: The Petri Net Representation of Mathematical Knowledge, Interner Bericht ISF-76-5, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1976.

Genrich (1979b)
Genrich,H.J.: Ein Kalkül des Planens und Handelns; in: Petri,C.A. (Hrsg.): Ansätze zur Organisationstheorie Rechnergestützter Informationssysteme, Bericht Nr. 111, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1979, S. 77-92.

Genrich (1980b)
Genrich,H.J.; Stankiewicz-Wiechno,E.: A Dictionary of Some Basic Notions of Net Theory; in: Brauer,W. (Hrsg.): Net Theory and Applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, 8.-19.10.1979 in Hamburg, Lecture Notes in Computer Science 84, Berlin - Heidelberg -New York 1980, S. 519-531.

Genrich (1987a)
Genrich,H.J.: Predicate/Transition Nets; in: Brauer,W.; Reisig,W.; Rozenberg,G. (Hrsg.): Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets 1986, Part I, Proceedings of an Advanced Course, 8.-19.09.1986 in Bad Honnef, Lecture Notes in Computer Science 254, Berlin - Heidelberg - New York ... 1987, S. 207-247.

Genrich (1988b)
Genrich,H.J.: Equivalence Transformations of PrT-Nets; in: o.V.: Application and Theory of Petri Nets - 9th European Workshop, 22.-24.06.1988 in Venedig, o.O. 1988, Vol. I, S. 229-248.

Gentzen (1936)
Gentzen,G.: Die Widerspruchsfreiheit der reinen Zahlentheorie.; in: Mathematische Annalen, 112. Bd. (1935/36), S. 493-565.

Gentzen (1938)
Gentzen,G.: Die gegenwärtige Lage in der mathematischen Grundlagenforschung - Neue Fassung des Widerspruchfreiheitsbeweises für die reine Zahlentheorie, Leipzig 1938.

Georgeff (1986)
Georgeff,M.P.: The Representation of Events in Multiagent Domains; in: o.V.: Proceedings AAAI-86, Fifth International Conference on Artificial Intelligence, 11.-15.08.1986 in Philadelphia, Los Altos 1986, Vol. 1, S. 70-75.

Gerlach,H. (1987)
Gerlach,H.: Executing P/T-Nets Using Rewrite Rules, Technical Paper KAI 24/1, ESPRIT Project 125 - GRASPIN, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1987.

Gerlach,H. (1988)
Gerlach,H.: Rewriting applied to Pr/E-nets over algebraic specifications with constructors, ESPRIT Project 125 - GRASPIN, Technical Paper KAI 29/1, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern 1988.

Gerum (1979)
Gerum,E.: Prinzipien des Konstruktivismus; in: Raffee,H.; Abel,B. (Hrsg.): Wissenschaftstheoretische Grundfragen der Wirtschaftswissenschaften, München 1979, S. 205-208.

Gethmann (1977)
Gethmann,C.F.; Hegselmann,R.: Das Problem der Begründung zwischen Dezisionismus und Fundamentalismus; in: Zeitschrift für allgemeine Wissenschaftstheorie, Bd. 8 (1977), S. 342-368.

Gethmann (1980a)
Gethmann,C.F. (Hrsg.): Theorie des wissenschaftlichen Argumentierens, Frankfurt 1980.

Gethmann (1980b)
Gethmann,C.F.: Die Logik der Wissenschaftstheorie; in: Gethmann,C.F. (Hrsg.): Theorie des wissenschaftlichen Argumentierens, Frankfurt 1980, S. 15-42.

Gethmann (1987)
Gethmann,C.F.: Letztbegründungen vs. lebensweltliche Fundierung des Wissens und Handelns; in: Köhler,W.R.; Kuhlmann,W.; Rohs,P. (Hrsg.): Philosophie und Begründung, Frankfurt 1987, S. 268-302.

Giloi (1989)
Giloi,K.: The FIRST Parallel Operating Prolog Engine; in: o.V.: Jahresbericht 1988 - Gesellschaft für Mathematik und Datenverarbeitung mbH, Sankt Augustin 1989, S. 42-54.

Ginsberg (1986)
Ginsberg,M.L.: Multi-valued Logics; in: o.V.: proceedings aaai-86, fifth national conference on artificial intelligence, 11.-15.08.1986 in Philadelphia, Vol. 1: sciene, Los Altos 1986, S. 243-247.

Giordana (1985)
Giordana,A.; Saitta,L.: Modeling Production Rules by Means of Predicate Transition Networks; in: Information Sciences, Vol. 35 (1985), S. 1-41.

Glover (1988)
Glover,F.; Greenberg,H.J.: Logical Testing for Rule-Base Management; in: Annals of Operations Research, Vol. 12 (1988), S. 199-215.

Gödel (1930)
Gödel,K.: Die Vollständigkeit der Axiome des logischen Funktionenkalküls; in: Monatshefte für Mathematik und Physik, Bd. 37 (1930), S. 349-360.

Gödel (1931)
Gödel,K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I; in: Monatshefte für Mathematik und Physik, 38. Bd. (1931), S. 173-198.

Goldberg (1984a)
Goldberg,A.; Pohl,I.: Is Complexity Theory of Use to AI?; in: Elithorn,A.; Banerji,R. (Hrsg.): Artificial and Human Intelligence, Edited Review Papers presented at the International NATO Symposium on Artificial and Human Intelligence, im Oktober 1981 in Lyon, Amsterdam - New York - Oxford 1984, S. 43-55.

Goltz,U. (1983c)
Goltz,U.; Vogt,U.: Processes of Relation Nets; in: Special Interest Group "Petri Nets and Related System Models" (Gesellschaft für Informatik), Newsletter 14 (1983), S. 10-19.

Goos (1987a)
Goos,G.; Dietrich,R.; Kursawe,P.: Prolog-Arbeiten in Karlsruhe; in: Brauer,W.; Wahlster,W. (Hrsg.): Wissensbasierte Systeme. 2. Internationaler GI-Kongreß, 20.-21.10.1987 in München, Proceedings, Informatik-Fachberichte, Berlin - Heidelberg - New York ... 1987, S. 89-104.

Gostelow (1971)
Gostelow,K.P.: Flow of Control, Resource Allocation, and the Proper Termination of Programs, Dissertation an der University of California, Los Angeles 1971.

Grabowski,J. (1980c)
Grabowski,J.: Lineare Methoden in der Theorie der Vektoradditionssysteme (III), Bericht PSF 1297, Sektion Mathematik, Humboldt-Universität Berlin (Ost), Berlin (Ost) 1980.

Graham,N. (1983)
Graham,N.: Künstliche Intelligenz - Wie Sie Ihren Computer zum Denken bringen, Sprendlingen 1983.

Grewendorf (1987)
Grewendorf,G.; Hamm,F.; Sternefeld,W.: Sprachliches Wissen - Eine Einführung in moderne Theorien der grammatischen Beschreibung, Frankfurt 1987.

Günther,A. (1974b)
Günther,A.; Heidrich,C.H.: Konstruktion einer zeitbezogenen Semantikstruktur für eine Frage-Antwort Interaktion; in: Ungeheuer,G. (Hrsg.): Kommunikationsforschung und Phonetik , Festschrift zum fünfzigjährigen Bestehen des Instituts für Kommunikationsforschung und Phonetik an der Universität Bonn, Hamburg 1974, S. 125-140.

Günther,H. (1986)
Günther,H.O.: The Design of an Hierarchical Model for Production Planning and Scheduling; in: Axsäter,S., Schneeweiss,C.; Silver,E. (Hrsg.): Multi-Stage Production Planning and Inventory Control, Lecture Notes in Economics and Mathematical Systems 266, Berlin - Heidelberg - New York - Tokyo 1986, S. 227-260.

Habel (1983)
Habel,C.: Logische Systeme und Repräsentationsprobleme; in: Neumann,B. (Hrsg.): GWAI-83, 7th German Workshop on Artificial Intelligence, 19.-23.09.1983 in Dassel/Solling, Informatik-Fachberichte 76, Berlin - Heidelberg - New York ... 1983, S. 118-142.

Habermas (1973b)
Habermas,J.: Legitimationsprobleme im Spätkapitalismus, Frankfurt 1973.

Habermas (1988)
Habermas,J.: Nachmetaphysisches Denken - Philosophische Aufsätze, 2. Aufl., Frankfurt 1988.

Hack,M. (1975a)
Hack,M.H.T.: Decidability Questions For Petri Nets, Dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1975.

Harel (1979)
Harel,D.: First-Order Dynamic Logic, Lecture Notes in Computer Science 68, Berlin - Heidelberg - New York 1979.

Harmon (1985)
Harmon,P.; King,D.: Expert Systems - Artificial Intelligence in Business, New York - Chichester - Brisbane ... 1985.

Harmon (1989)
Harmon,P.; Maus,R.; Morrissey,W.: Expertensysteme: Werkzeuge und Anwendungen, München - Wien 1989.

Hartmanis (1965)
Hartmanis,J.; Stearns,R.E.: On the Computational Complexity of Algorithms; in: Transactions of the American Mathematical Society, Vol. 117 (1965), S. 285-306.

Hasenjaeger (1952)
Hasenjaeger,G.: Über *-Unvollständigkeit in der Peano-Arithmetik; in: The Journal of Symbolic Logic, Vol. 17 (1952), S. 81-97.

Hasenjaeger (1968)
Hasenjaeger,G.: Logik und Ontologie; in: Klibansky,R. (Hrsg.): Contemporary Philosophy - A Survey, Vol. I: Logic and Foundations of Mathematics, Firenze 1968, S. 241-249.

Hayes,P. (1981)
Hayes,P.J.: The Frame Problem and Related Problems in Artificial Intelligence; in: Webber,B.L.; Nilsson,N.J. (Hrsg.): Readings in Artificial Intelligence - A Collection of Articles, Palo Alto 1981, S. 223-230.

Hegselmann (1982)
Hegselmann,R.; Raub,W.: Zur Logikabhängigkeit wissenschaftstheoretischer Paradoxien; in: Erkenntnis, Vol. 17 (1982), S. 349-359.

Heinen (1983)
Heinen,E.: Betriebswirtschaftliche Kostenlehre - Kostentheorie und Kostenentscheidungen, 6. Aufl., Wiesbaden 1983.

Helberg (1987)
Helberg,P.: PPS als CIM-Baustein - Gestaltung der Produktionsplanung und -steuerung für die computerintegrierte Produktion, Berlin 1987.

Henschen (1974)
Henschen,L.; Wos,L.: Unit Refutations and Horn Sets; in: Journal of the Association for Computing Machinery, Vol. 21 (1974), S. 590-605.

Hermes (1937)
Hermes,H.: Definite Begriffe und berechenbare Zahlen; in: Semesterberichte zur Pflege des Zusammenhangs von Universität und Schule aus den mathematischen Seminaren, 10. Bd. (1937), S. 110-123.

Hermes (1952)
Hermes,H.: Maschinen zur Entscheidung von mathematischen Problemen; in: Mathematisch = physikalische Semesterberichte, Bd. II (1952), S. 179-189.

Hermes (1961)
Hermes,H.: Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit - Einführung in die Theorie der rekursiven Funktionen, Berlin - Göttingen - Heidelberg 1961.

Hermes (1968)
Hermes,H.: Prädikatenlogik und Theorie der rekursiven Funktionen; in: Klibansky,R. (Hrsg.): Contemporary Philosophy - A Survey, I: Logic and Foundations of Mathematics, Firenze 1968, S. 254-265.

Hermes (1978)
Hermes,H.: Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit - Einführung in die Theorie der rekursiven Funktionen, 3. Aufl., Berlin - Göttingen - Heidelberg 1978.

Herschel (1974)
Herschel,R.: Einführung in die Theorie der Automaten, Sprachen und Algorithmen, München - Wien 1974.

Hertzberg (1985)
Hertzberg,J.: Über Künstliche Intelligenz und die reale Welt, KI-Bericht Nr. 2, Institut für Informatik, Universität Bonn, Bonn 1985.

Hertzberg (1986)
Hertzberg,J.: Planerstellungs-Methoden der Künstlichen Intelligenz; in: Informatik-Spektrum, Bd. 9 (1986), S. 149-161.

Hertzberg (1989)
Hertzberg,J.: Planen - Einführung in die Planerstellungsmethoden der Künstlichen Intelligenz, Mannheim - Wien - Zürich 1989.

Heyderhoff (1984)
Heyderhoff,P.: GRASPIN: A Coherent Methodology for Software Development, ESPRIT Project 125 - GRASPIN, Technical Paper GMD 11/2, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1984.

Heyer (1987)
Heyer,G.: Generische Kennzeichnungen - Zur Logik und Ontologie generischer Bedeutung, Dissertation, Universität Bochum 1983, München - Wien 1987.

Heyting (1931)
Heyting,A.: Die intuitionistische Grundlegung der Mathematik; in: Erkenntnis, 2. Bd. (1931), S. 106-115.

Heyting (1968a)
Heyting,A.: L.E.J. Brouwer; in: Klibansky,R. (Hrsg.): Contemporary Philosophy - A Survey, I: Logic and Foundations of Mathematics, Firenze 1968, S. 309-315.

Heyting (1968b)
Heyting,A.: Intuitionism in Mathematics; in: Klibansky,R. (Hrsg.): Contemporary Philosophy - A Survey, I: Logic and Foundations of Mathematics, Firenze 1968, S. 316-323.

Hilbert (1900)
Hilbert,D.: Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Kongreß zu Paris 1900.; in: Nachrichten von der Königl(ichen) Gesellschaft der Wissenschaften zu Göttingen. Mathematisch-physikalische Klasse aus dem jahre 1900, Göttingen 1900, S. 253-297.

Hilbert (1925)
Hilbert,D.: Über das Unendliche; in: Mathematische Annalen, 95. Bd. (1925), S. 161-190.

Höss (1986)
Höss,K.; Küchenhoff,V.; Pichler,C.; Schmauch,C.; Thuy,o.Vn.: Objekt-orientierte Programmierung in Prolog; in: Prolog - Interface, Vol. 1 (1986), Heft 3, S. 3-11.

Hofbauer (1989)
Hofbauer,D.; Kutsche,R.-D.: Grundlagen des maschinellen Beweisens - Eine Einführung für Informatiker und Mathematiker, Braunschweig - Wiesbaden 1989.

Hohenstein (1986)
Hohenstein,U.; Neugebauer,L.; Saake,G.: An Extended Entity-Relationship Model for Non-Standard Datenbases; in: Heuer,A. (Hrsg.): Workshop über Relationale Datenbanken, 16.-20.06.1986 in Lessach, Informatik-Bericht 86/3, Institut für Informatik, Universität Clausthal-Zellerfeld, Clausthal-Zellerfeld 1986, S. 185-211. (Anmk. des Verf.: "Datenbases" im Original; gemeint ist wohl: "Databases".)

Hohenstein (1987)
Hohenstein,U.; Neugebauer,L.; Saake,G.; Ehrich,H.-D.: Three-Level-Specification of Databases Using an Extended Entity-Relationship Model; in: Wagner,R.R.; Traunmüller,R.; Mayr,H.C. (Hrsg.): Informationsbedarfsermittlung und -analyse für den Entwurf von Informationssystemen, Fachtagung EMISA, 02.-03.07.1987 in Linz, Proceedings, Informatik-Fachberichte 143, Berlin - Heidelberg - New York ... 1987, S. 58-88.

Hopcroft (1974)
Hopcroft,J.E.: Complexity of Computer Computations; in: Rosenfeld,J.L. (Hrsg.): Information Processing 74, Proceedings of the IFIP Congress 1974, 05.-10.08.1974 in Stockholm, Amsterdam - London - New York 1974, S. 620-626.

Hopcroft (1984)
Hopcroft,J.E.: Turingmaschinen; in: Spektrum der Wissenschaft, o.Jg. (1984), Heft 7, S. 34-49.

Horowitz (1981)
Horowitz,E.; Sahni,S.: Algorithmen - Entwurf und Analyse, Berlin - Heidelberg - New York 1981.

Howell (1988a)
Howell,R.R.; Rosier,L.E.; Yen,H.-C.: A Taxonomy of Fairness and Temporal Logic Problems for Petri Nets; in: Chytil,M.P.; Janiga,L.; Koubek,V. (Hrsg.): MFCS'88 - Mathematical Foundations of Computer Science, Proceedings of the 13th Symposium, 29.08.-02.09.1988 in Carlsbad, Lecture Notes in Computer Science 324, Berlin - Heidelberg - New York ... 1988, S. 351-359. (Auch erschienen als: Technical Report No. 88-4, Department of Computer Science, Iowa State University, o.O. 1988.)

Hsiang (1983a)
Hsiang,J.; Dershowitz,N.: Rewrite Methods for Clausal and Non-Clausal Theorem Proving; in: Diaz,J. (Hrsg.): Automata, Languages and Programming, 10th Colloquium, 18.-22.07.1983 in Barcelona, Lecture Notes in Computer Science 154, Berlin - Heidelberg - New York .... 1983, S. 331-346.

Hsiang (1983b)
Hsiang,J.: Topics in Automated Theorem Proving and Program Generation, Dissertation, University of Illinois, Urbana-Champaign 1983, Ann Arbor 1983.

Huber,M. (1987)
Huber,M.; Varsek,I.: Extended Prolog for Order-Sorted Resolution; in: o.V.: Proceedings 1987 Symposium on Logic Programming, 31.08.-04.09.1987 in San Francisco, o.O. (New York) 1987, S. 34-41.

Hutchinson,S. (1986)
Hutchinson,S.A.; Kak,A.C.: FProlog: A Language to Integrate Logic and Functional Programming for Automated Assembly; in: o.V.: Proceedings of the IEEE International Conference on Robotics and Automation, Washington 1986, S. 904-909.

Igel (1989a)
Igel,B.: Applikative Beschreibungsmethoden in verteilten Prozeßsteuerungen; in: Henn,R.; Stieger,K. (Hrsg.): PEARL 89 - Workshop über Realzeitsysteme, 10. Fachtagung des PEARL-Vereins e.V., 07.-08.12.1989 in Boppard, Proceedings, Informatik-Fachberichte 231, Berlin - Heidelberg - New York ... 1989, S. 172-195.

Irani,K. (1985)
Irani,K.B.; Shin,D.G.: A Many-Sorted Resolution based on an Extension of a First-Order Language; in: o.V.: Proceedings of the Ninth International Joint Conference on Artificial Intelligence, IJCAI-85, 18.-23.08.1985 in Los Angeles, Los Altos 1985, Vol. 2, S. 1175-1177.

Israel (1983)
Israel,D.J.: The Role of Logic in Knowledge Representation; in: Computer, Vol. 16 (1983), No. 2, S. 37-41.

Itzinger (1976)
Itzinger,O.: Methoden der maschinellen Intelligenz, München - Wien 1976.

Jablonski (1990)
Jablonski,S.: Datenverwaltung in verteilten Systemen - Grundlagen und Lösungskonzepte, Informatik-Fachberichte 233, Berlin - Heidelberg - New York ... 1990.

Jackson,M.A. (1979)
Jackson,M.A.: Grundsätze des Programmentwurfs, Darmstadt 1979.

Janas (1979)
Janas,J.M.: How to Not Say "Nil" - Improving Answers to Failing Queries in Data Base Systems; in: o.V.: IJCAI-79, Proceedings of the Sixth International Conference on Artificial Intelligence, 20-23.08.1979 in Tokyo, Vol. 1, o.O. (Stanford) 1979, S. 429-434.

Janich (1974)
Janich,P.; Kambartel,F.; Mittelstraß,J.: Wissenschaftstheorie als Wissenschaftskritik, Frankfurt 1974.

Jarke (1984)
Jarke,M.; Vassiliou,Y.: Coupling Expert Systems With Database Management Systems; in: Reitman,W. (Hrsg.): Artificial Intelligence Applications For Business, Proceedings of the NYU Symposium, 18.-20.05.1983 in New York, Norwood 1984, S. 65-85.

Jarke (1989b)
Jarke,M.; Jeusfeld,M.; Rose,T.: Software Process Modeling as a Strategy for KBMS Implementation, Bericht MIP-8933, Fakultät für Mathematik und Informatik, Universität Passau, Passau 1989.

Jensen (1982a)
Jensen,K.: High-Level Petri Nets, Report DAIMI PB-151, Computer Science Department, Aarhus University, Aarhus 1982.

Jensen (1987a)
Jensen,K.: Coloured Petri Nets; in: Brauer,W.; Reisig,W.; Rozenberg,G. (Hrsg.): Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets 1986, Part I, Proceedings of an Advanced Course, 8.-19.09.1986 in Bad Honnef, Lecture Notes in Computer Science 254, Berlin - Heidelberg - New York ... 1987, S. 248-299.

Jeusfeld (1990)
Jeusfeld,M.; Krüger,E.: Deductive Integrity Maintenance in an Object-Oriented Setting, Bericht MIP-9013, Fakultät für Mathematik und Informatik, Universität Passau, Passau 1990.

Jones,N. (1977)
Jones,N.D.; Landweber,L.H.; Lien,Y.E.: Complexity of some problems in Petri Nets; in: Theoretical Computer Science, Vol. 4 (1977), S. 277-299.

Jordan (1978)
Jordan,W.; Urban,H.: Strukturierte Programmierung, Berlin - Heidelberg - New York 1978.

Kambartel (1974a)
Kambartel,F. (Hrsg.): Praktische Philosophie und konstruktive Wissenschaftstheorie, Frankfurt 1974.

Kambartel (1974b)
Kambartel,F.: Wie ist praktische Philosophie konstruktiv möglich? - Über einige Mißverständnisse eines methodischen Verständnisses praktischer Diskurse; Kambartel,F. (Hrsg.): Praktische Philosophie und konstruktive Wissenschaftstheorie, Frankfurt 1974, S. 9-33.

Kambartel (1975)
Kambartel,F.: Wissenschaftstheorie und Wissenschaftspraxis; in: Weingart,P. (Hrsg.): Wissenschaftsforschung, Frankfurt - New York 1975, S. 162-183.

Kamlah,W. (1967)
Kamlah,W.; Lorenzen,P.: Logische Propädeutik - Vorschule des vernünftigen Redens, Mannheim - Wien - Zürich 1967.

Kanitscheider (1971)
Kanitscheider,B.: Geometrie und Wirklichkeit, Berlin 1971.

Keller,R. (1972a)
Keller,R.M.: Vector Replacement Systems: A Formalism for Modeling Asynchronous Systems, Technical Report 117, Department of Electrical Engineering, Computer Sciences Laboratories, Princeton University, Princeton 1972.

Kinnebrock (1988)
Kinnebrock,W.: Turbo Prolog, München - Wien 1988.

Kirchner (1988)
Kirchner,C.; Kirchner,H.; Meseguer,J.: Operational Semantics of OBJ-3; in: Lepistö,T.; Salomaa,A. (Hrsg.): Automata, Languages and Programming, 15th International Colloquium, 11.-15.07.1988 in Tampere, Lecture Notes in Computer Science 317, Berlin - Heidelberg - New York ... 1988, S. 287-301.

Kleene (1952)
Kleene,S.C.: Introduction to Metamathematics, Amsterdam - Groningen 1952.

Kleene (1976)
Kleene,S.C.: The Wortk of Kurt Gödel; in: The Journal of Symbolic Logic, Vol. 41 (1976), S. 761-778.

Klein,S. (1989)
Klein,S.: Theorie der Unternehmungsplanung - Struktur und Beitrag einer anwendungsorientierten Planungstheorie, Stuttgart 1989.

Kleine Büning (1988a)
Kleine Büning,H.; Lettmann,T.: Perspektiven für die Logikprogrammierung; in: Rahmstorf,G. (Hrsg.): Wissensrepräsentation in Expertensystemen, Workshop, 16.-18.03.1987 in Herrenberg, Proceedings, Informatik-Fachberichte 172, Berlin - Heidelberg - New York ... 1988, S. 56-78.

Kleinhans (1989)
Kleinhans,A.M.: Wissensverarbeitung im Management - Möglichkeiten und Grenzen wissensbasierter Managementunterstützungs-, Planungs- und Simulationssysteme, Dissertation, Universität Stuttgart 1989, Frankfurt - Bern - New York ... 1989.

Knolmayer (1989)
Knolmayer,G.: Die Berücksichtigung des Zeitbezugs von Daten bei der Gestaltung computergestützer Informationssysteme; in: Hax,H.; Kern,W.; Schröder,H.-H. (Hrsg.): Zeitaspekte betriebswirtschaftlicher Theorie und Praxis, 50. Wissenschaftliche Jahrestagung des Verbandes der Hochschullehrer für Betriebswirtschaft e.V., 24.-28.05.1988 in Köln, Stuttgart 1989, S., 77-88.

Kobsa (1982)
Kobsa,A.: Wissensrepräsentation - Die Darstellung von Wissen im Computer, Bericht, Österreichische Studiengesellschaft für Kybernetik, o.O. (Wien) 1982.

Körner,S. (1968)
Körner,S.: Philosophie der Mathematik - Eine Einführung, München 1968.

Körner,S. (1970)
Körner,S.: Erfahrung und Theorie - Ein wissenschaftstheoretischer Versuch, Frankfurt 1970.

Konagaya (1984)
Konagaya,A.; Umemura,M.: Knowledge Information Processing Language: Shape Up; in: New Generation Computing, Vol. 2 (1984), S. 195-201.

Kornfeld (1981a)
Kornfeld,W.A.: The Use of Parallelism to Implement a Heuristic Search; in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, o.O. (Menlo Park) 1981, Vol. 1, S. 575-580.

Kosiol (1961a)
Kosiol,E.: Modellanalyse als Grundlage unternehmerischer Entscheidungen; in: Zeitschrift für handelswissenschaftliche Forschung - Neue Folge, 13. Jg. (1961), S. 318-334.

Kotz (1989)
Kotz,A.M.: Triggermechanismen in Datenbanksystemen, Informatik-Fachberichte 201, Berlin - Heidelberg - New York ... 1989.

Kowalski (1974)
Kowalski,R.: Predicate Logic as Programming Language; in: Rosenfeld,J.L. (Hrsg.): Information Processing 74, Proceedings of IFIP Congress 74, 05.-10.08.1974 in Stockholm, Amsterdam - London - New York 1974, S. 569-574.

Kowalski (1978)
Kowalski,R.: Logic for Data Description; in: Gallaire,H.; Minker,J. (Hrsg.): Logic and Data Bases, New York - London 1978, S. 77-103.

Kowalski (1979a)
Kowalski,R.: Algorithm = Logic + Control; in: Communications of the ACM, Vol. 22 (1979), S. 424-436.

Kowalski (1982a)
Kowalski,R.: Logic Programming in the Fifth Generation, Paper, Department of Computing, Imperial College, London 1982.

Kowalski (1983a)
Kowalski,R.: Logic for Problem Solving, 2. und zugleich 3. Aufl., New York - Amsterdam - Oxford 1983.

Kowalski (1984a)
Kowalski,R.: The Early History of Logic Programming, Paper, Department of Computing, Imperial College, London 1984.

Kowalski (1987a)
Kowalski,R.: Directions for Logic Programming; in: Brauer,W.; Wahlster,W. (Hrsg.): Wissensbasierte Systeme. 2. Internationaler GI-Kongreß, 20.-21.10.1987 in München, Proceedings, Informatik-Fachberichte, Berlin - Heidelberg - New York ... 1987, S. 128-146.

Krämer (1983a)
Krämer,B.: On the Stepwise Construction of Non-Sequential Software Systems Using a Net Based Description Language; in: in: o.V.: Papers presented at the 4th European Workshop on Applications and Theory of Petri Nets, 26.-29.09.1983 in Toulouse, o.O. 1983, S. 230-252.

Krämer (1984)
Krämer,B.: Formal and Semi-Graphic Specification of Non-Sequential Systems, ESPRIT Project 125 - GRASPIN, Technical Paper GMD 12/2, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1984.

Krämer (1985a)
Krämer,B.: Stepwise Construction of Non-Sequential Software Systems Using a Net-Based Specification Language; in: Rozenberg,G.; Genrich,H.; Roucairol,G. (Hrsg.): Advances in Petri Nets 1984, Lecture Notes in Computer Science 188, Berlin - Heidelberg - New York - Tokyo 1985, S. 307-330. (Anmk. des Verf.: überarbeitete Fassung von Krämer (1983a).)

Krämer (1986a)
Krämer,B.: Die Spezifikations- und Entwurfssprache SEGRAS; in: Handbuch der modernen Datenverarbeitung, 23. Jg. (1986), Heft 130, S. 107-116.

Krämer (1986b)
Krämer,B.: SEGRAS: The GRASPIN Specification Language - Preliminary Reference Manual, ESPRIT Project 125 - GRASPIN, Technical Paper GMD 26/2, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1986.

Krämer (1987a)
Krämer,B.; Schmidt,H.W.: Types and Modules for Net Specifications; in: Voss,K.; Genrich,H.J.; Rozenberg,G. (Hrsg.): Concurrency and Nets - Advances in Petri Nets, Berlin - Heidelberg - New York ... 1987, S. 269-286.

Krämer (1987b)
Krämer,B.: SEGRAS - A Formal and Semigraphical Language combining Petri Nets and Abstract Data Types in the Specification of Distributed Systems; in: o.V.: Proceedings of the 9th Annual International Conference on Software Engineering, im März 1987 in Monterey 1987, New York 1987, S. 116-125.

Krallmann (1990a)
Krallmann,H.: Modellierung betrieblicher Kommunikationsarchitekturen; in: computer magazin, 19. Jg. (1990), Heft 3/4, S. 22-23.

Kramosil (1975)
Kramosil,I.: A Note on Deduction Rules with Negative Premises; in: o.V.: Advance Papers of the Fourth International Conference on Artificial Intelligence (IJCAI-75), 03.-08.09.1975 in Tbilisi, Vol. 1, o.O. (Cambridge/Massachusetts), S. 53-56.

Kripke (1959)
Kripke,S.A.: A Completeness Theorem in Modal Logic; in: The Journal of Symbolic Logic, Vol. 24 (1959), S. 1-14.

Kung (1985)
Kung,C.-H.: High Parallelism and a Proof Procedure, I: Theoretical Considerations; in: Decision Support Systems, Vol. 1 (1985), S. 323-331.

Kuypers (1973)
Kuypers,W. (Hrsg.): Mathematik Sekundarstufe II - Einführung - Logik, Relationen, Zahlen, Düsseldorf 1973.

Lamport,L. (1983)
Lamport,L.: What Good is Temporal Logic?; in: Mason,R.E.A. (Hrsg.): Information Processing 83, Proceedings of the IFIP 9th World Computer Congress, 19.-23.09.1983 in Paris, Amsterdam - New York - Oxford 1983, S. 657-668.

Landweber (1978)
Landweber,L.H.; Robertson,E.L.: Properties of Conflict-Free and Persistent Petri Nets; in: Journal of the Association for Computing Machinery, Vol. 25 (1978), S. 352-364.

Laurent (1984)
Laurent,J.-P.: Control structures in expert systems; in: Technology and Science of Informatics, Vol. 3 (1984), No. 3, S. 147-162.

Lautenbach (1985b)
Lautenbach,K.: On Logical and Linear Dependencies, Arbeitspapiere der GMD 147, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1985.

Lee,R. (1983)
Lee,R.M.: Epistemological Aspects of Knowledge-Based Decision Sulport Systems; in: Sol.,H.G. (Hrsg.): Processes and Tools for Decision Support, Proceedings of the Joint IFIP WG 8.3 / IASA Working Conference on Processes and Tools for Decision Support, 19.-21.07.1982 in Laxenburg, Amsterdam - New York - Oxford 1983, S. 25-36. (Anmk. des Verf.: Druckfehler in "Sulport"; gemeint ist wohl: "Support".)

Lee,R. (1985)
Lee,R.M.: Database Inferencing for Decision Support; in: Decision Support Systems, Vol. 1 (1985), S. 57-68.

Lee,R. (1988a)
Lee,R.M.: Logic, Semantics and Data Base Modeling: An Ontology; in: Meersman,R.A.; Sernadas,A.C. (Hrsg.): Data and Knowledge (DS-2), Proceedings of the Second IFIP 2.6 Working Conference on Database Semantics, 'Data and Knowledge' (DS-2), 03.-07.11.1986 in Albufeira, Amsterdam - New York - Oxford ... 1988, S. 221-243.

Leikauf (1989)
Leikauf,P.: Konsistenzsicherung durch Verwaltung von Konsistenzverletzungen; in: Härder,T. (Hrsg.): Datenbanksysteme in Büro, Technik und Wissenschaft, GI/SI-Fachtagung, 01.-03.03.1989 in Zürich, Proceedings, Informatik-Fachberichte 204, Berlin - Heidelberg - New York ... 1989, S. 135-153.

Lenat (1984)
Lenat,D.B.: Software für Künstliche Intelligenz; in: Spektrum der Wissenschaft, o.Jg. (1984), Heft 11, S. 178-189.

Lenk (1973)
Lenk,H.: Metalogik und Sprachanalyse - Studien zur analytischen Philosophie, Freiburg 1973.

Lenk (1976)
Lenk,H.: Komplementbildungen und Boolesche sowie Brouwer-Verbände in der Handlungslogik; in: Kanitscheider,B. (Hrsg.): Sprache und Erkenntnis, Festschrift für Gerhard Frey zum 60. Geburtstag, Innsbruck 1976, S. 271-282.

Lenstra (1979)
Lenstra,J.K.; Rinnooy Kan,A.H.G.: Computational Complexity of Discrete Optimization Problems; in: Hammer,P.L.; Johnson,E.L.; Korte,B.H. (Hrsg.): Discrete Optimization I, Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications, im August 1977 in Banff und Vancouver; zugleich: Annals of Discrete Mathematics, Vol. 4 (1979), Amsterdam - New York - Oxford 1979, S. 121-140.

Lenstra (1982)
Lenstra,J.K.; Rinnooy Kan,A.H.G.; van Emde Boas,P.: An appraisal of computational complexity for operations researchers; in: European Journal of Operational Research, Vol. 11 (1982), S. 201-210.

Leßner (1980)
Leßner,G.: Elemente der Topologie und Graphentheorie, Freiburg - Basel - Wien 1980.

Levi,G. (1986)
Levi,G.: Logic Programming: The Foundations, the Approach and the Role of Concurrency; in: de Bakker,J.W.; de Roever,W.-P.; Rozenberg,G. (Hrsg.): Current Trends in Concurrency - Overviews and Tutorials, Lecture Notes in Computer Sciene 224, Berlin - Heidelberg - New York ... 1986, S. 396-441.

Lifschitz (1989)
Lifschitz,V.: Logical Foundations of Deductive Databases; in: Ritter,G.X. (Hrsg.): Information Processing 89, Proceedings of the IFIP 11th World Computer Congress, 28.08.-01.09.1989 in San Francisco, Amsterdam - New York - Oxford ... 1989, S. 315-321.

Lind (1988)
Lind,C.: Implementation eines Uhrmechanismus für Petri-Netze, Studienarbeit, Lehrstuhl für Praktische Informatik III, Universität Mannheim, Mannheim 1988.

Lipeck (1989)
Lipeck,U.W.: Dynamische Integrität von Datenbanken - Grundlagen der Spezifikation und Überwachung, Informatik-Fachberichte 209, Berlin - Heidelberg - New York 1989.

Liu,N. (1987)
Liu,N.K.; Dillon,T.: Detection of consistency and completeness in expert systems using Numerical Petri Nets; in: Gero,J.S.; Stanton,R. (Hrsg.): Artificial Intelligence Developments and Applications, Edited Selection of Papers to the Australian Joint Artificial Intelligence Conference, 02.-04.11.1987 in Sydney, Amsterdam - New York - Oxford ... 1987, S. 119-134.

Lorenz,Ku. (1979a)
Lorenz,Ku. (Hrsg.): Konstruktionen versus Positionen - Beiträge zur Diskussion um die Konstruktive Wissenschaftstheorie, Bd. I: Spezielle Wissenschaftstheorie, Berlin - New York 1979.

Lorenzen,P. (1962)
Lorenzen,P.: Metamathematik, Mannheim 1962.

Lorenzen,P. (1973a)
Lorenzen,P.; Inhetveen,R.: Die Einheit der Wissenschaften; in: Kambartel,F.; Mittelstraß,J. (Hrsg.): Zum normativen Fundament der Wissenschaft, Frankfurt 1973, S. 70-78.

Lorenzen,P. (1975a)
Lorenzen,P.; Schwemmer,O.: Konstruktive Logik, Ethik und Wissenschaftstheorie, 2. Aufl., Mannheim - Wien - Zürich 1975.

Lorenzen,P. (1978a)
Lorenzen,P.: Theorie der technischen und politischen Vernunft, Stuttgart 1978.

Lorenzen,P. (1980)
Lorenzen,P.: Die dialogische Begründung von Logikkalkülen; in: Gethmann,C.F. (Hrsg.): Theorie des wissenschaftlichen Argumentierens, Frankfurt 1980, S. 43-69.

Lorenzen,P. (1987)
Lorenzen,P.: Lehrbuch der konstruktiven Wissenschaftstheorie, Mannheim - Wien - Zürich 1987.

Loveland (1978)
Loveland,D.W.: Automated Theorem Proving: A Logical Basis, Amsterdam - New York - Oxford 1978.

Lucas (1961)
Lucas,J.R.: Minds, Machines, and Gödel; in: Philosophy, Vol. 36 (1961), S. 112-127.

Ludewig (1983)
Ludewig,J.; Schult,U.; Wankmüller,F.: Chasing the Busy-Beaver - Notes and Observations on a Competition to Find the 5-State Busy Beaver, Bericht Nr. 159, Abteilung Informatik II, Universität Dortmund, Dortmund 1983.

Lum (1982)
Lum,V.Y.; Choy,D.M.; Shu,N.C.: OPAS: An office procedure automation system; in: IBM Systems Journal, Vol. 21 (1982), S. 327-350.

Lusti (1990)
Lusti,M.: Wissensbasierte Systeme - Algorithmen, Datenstrukturen und Werkzeuge, Mannheim - Wien - Zürich 1990.

Luxemburg (1973)
Luxemburg,W.A.J.: What is Nonstandard Analysis?; in: The American Mathematical Monthly, Vol. 80 (1973), Papers in the Foundation of Mathematica, published as a supplemant to the American Mathematical Monthly, S. 38-67.

Machover (1983)
Machover,M.: Towards a New Philosophy of Mathematics; in: The British Journal for the Philosophy of Science, Vol. 34 (1983), S. 1-11.

Mainz (1984)
Mainz,U.: Netztheoretische Repräsentation prädikatenlogischer Begriffe und Methoden, Diplomarbeit am Institut für Informatik, Universität Bonn, Bonn 1984.

Malik (1986)
Malik,F.F.: Strategie des Managements komplexer Systeme - Ein Beitrag zur Management-Kybernetik evolutionärer Systeme, 2. Aufl., Bern - Stuttgart 1986.

Manna (1979)
Manna,Z.: The Modal Logic of Programs; in: Maurer,H.A. (Hrsg.): Automata, Languages and Programming, Sixth Colloquium, 16.-20.07.1979 in Graz, Lecture Notes in Computer Science 71, Berlin - Heidelberg - New York 1979, S. 385-409.

Marti-Oliet (1989)
Marti-Oliet,N.; Meseguer,J.: From Petri Nets to Linear Logic; in: Pitt,D.H.; Rydeheard,D.E.; Dybjer,P.; Pitts,A.M.; Poigne,A. (Hrsg.): Category Theory and Computer Science, 05.-08.09.1989 in Manchester, Proceedings, Lecture Notes in Computer Science 389, Berlin - Heidelberg - New York ... 1989, S. 313-340.

Matijasevic (1970)
Matijasevic,J.V.: Enumerable Sets Are Diophantine; in: Soviet Mathematics Doklady, Vol. 11 (1970), S. 354-358.

Mattern (1989a)
Mattern,F.; Mehl,H.: Diskrete Simulation - Prinzipien und Probleme der Effizienzsteigerung durch Parallelisierung; in: Informatik-Spektrum, Bd. 12 (1989), S. 198-210.

Mayr,E. (1980a)
Mayr,E.W.: Ein Algorithmus für das allgemeine Erreichbarkeitsproblem bei Petrinetzen und damit zusammenhängende Probleme, Dissertation an der Technischen Universität München, München 1980.

McCarthy,J. (1968)
McCarthy,J.: Programs with Common Sense; in: Minsky,M.L. (Hrsg.): Semantic Information Processing, Cambridge (Massachusetts) - London 1968, S. 403-418.

McCarthy,J. (1969)
McCarthy,J.; Hayes,P.J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence; in: Meltzer,B.; Michie,D.; Swann,M. (Hrsg.): Machine Intelligence 4, Edinburgh 1969, S. 463-502.

McCarthy,J. (1977)
McCarthy,J.: Epistemological Problems of Artificial Intelligence; in: o.V.: 5th International Conference on Artificial Intelligence - 1977, IJCAI-77, Proceedings of the Conference, 22.-25.08.1977 in Cambridge (Massachusetts), Vol. 2, o.O. (Pittsburgh) 1977, S. 1038-1044.

McCarthy,J. (1980)
McCarthy,J.: Circumscription - A Form of Non-Monotonic Reasoning; in: Artificial Intelligence, Vol. 13 (1980), S. 27-39.

McDermott (1979)
McDermott,D.; Doyle,J.: An Introduction to Non-Monotonic Logic; in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, o.O. (Stanford) 1979, Vol. 1, S. 562-567.

McDermott (1980)
McDermott,D.; Doyle,J.: Non-Monotonic Logic; in: Artificial Intelligence, Vol. 13 (1980), S. 41-72.

McDermott (1981)
McDermott,D.: A Temporal Logic for Reasoning About Processes and Plans, Research Report No. 196, Department of Computer Science, Yale University, New Haven 1981.

McDermott (1982a)
McDermott,D.: A Temporal Logic for Reasoning About Processes and Plans; in: Cognitive Science, Vol. 6 (1982), S. 101-155.

Mehlhorn (1977)
Mehlhorn,K.: Effiziente Algorithmen, Stuttgart 1977.

Meltzer (1970)
Meltzer,B.: The Semantics of Induction and the Possibility of Complete Systems of Inductive Inference; in: Artificial Intelligence, Vol. 1 (1970), S. 189-192.

Meltzer (1975)
Meltzer,B.: Vorbemerkungen zu einer Theorie der Effizienz von Beweisverfahren; in: Findler, N.V. (Hrsg.): Künstliche Intelligenz und heuristisches Programmieren, Wien - New York 1975, S. 15-32.

Mendelson (1964)
Mendelson,E.: Introduction to Mathematical Logic, New York - Cincinnati - Toronto ... 1964.

Mertens (1984)
Mertens,P.; Heigl,M.: Neuere Entwicklungen der computergestützten Produktionsplanung, Eignung - Verbindungen - Entwicklungspfade, Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung (Informatik) an der Universität Erlangen-Nürnberg, Bd. 17, Nr. 2, Erlangen 1984.

Mertens (1986a)
Mertens,P.; Hofmann,J.: Aktionsorientierte Datenverarbeitung; in: Informatik-Spektrum, Bd. 9 (1986), S. 323-333.

Mertens (1988a)
Mertens,P.: Industrielle Datenverarbeitung, Band 1: Administrations- und Dispositionsebene, 7. Aufl., Wiesbaden 1988.

Mertens (1988e)
Mertens,P.; Helmer,J.; Rose,H.; Wedel,T.: Ein Ansatz zu kooperierenden Expertensystemen bei der PPS, Skript, Universität Erlangen-Nürnberg, Nürnberg o.J. (1988).

Mertens II (1990)
Mertens,P.: Expertensysteme in der Produktion - Praxisbeispiele aus Disgnose und Planung - Entscheidungshilfen für den praktischen Einsatz, München - Wien 1990. (Anmk. des Verf.: Namensvetter des ansonsten angeführten Autors "Mertens,P.")

Meschkowski (1967)
Meschkowski,H.: Probleme des Unendlichen - Werk und Leben Georg Cantors, Braunschweig 1967.

Meyer,A. (1972)
Meyer,A.R.; Stockmeyer,L.J.: The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space; in: o.V.: Proceedings of the 13th Annual IEEE Symposium on Switching and Automata Theory, o.O. 1972, S. 125-129.

Meyer-Wegener (1987)
Meyer-Wegener,K.: Transaktionssysteme - verteilte Verarbeitung und verteilte Datenhaltung; in: Informationstechnik, 29. Jg. (1987), S. 120-126.

Meyer-Wegener (1988)
Meyer-Wegener,K.: Verteilte Verarbeituung mit Transaktionsmonitoren; in: computer magazin, 17. Jg. (1988), Heft 11, S. 47-50.

Minsky (1971)
Minsky,M.L.: Berechnung: Endliche und unendliche Maschinen, Stuttgart - Berlin - Köln ... 1971.

Mittelstaedt (1983)
Mittelstaedt,P.: Wahrheit, Wirklichkeit und Logik in der Sprache der Physik; in: Zeitschrift für allgemeine Wissenschaftstheorie, Bd. 14 (1983), S. 24-45.

Mittelstraß (1973)
Mittelstraß,J.: Das praktische Fundament der Wissenschaft und die Aufgabe der Philosophie; in: Kambartel,F.; Mittelstraß,J. (Hrsg.): Zum normativen Fundament der Wissenschaft, Frankfurt 1973, S. 1-69.

Mittelstraß (1974)
Mittelstraß,J.: Die Möglichkeit von Wissenschaft, Frankfurt 1974.

Mittelstraß (1984a)
Mittelstraß,J.: Forschung, Begründung, Rekonstruktion - Wege aus dem Begründungsstreit; in: Schnädelbach,H, (Hrsg.): Rationalität - Philosophische Beiträge, Frankfurt 1984, S. 117-140.

Mittelstraß (1984b)
Mittelstraß,J.: Gibt es eine Letztbegründung?; in: Janich,P. (Hrsg.): Methodische Philosophie - Beiträge zum Begründungsproblem der exakten Wissenschaften in Auseinandersetzung mit Hugo Dingler, Kolloquium, 2.-3.07.1981 in Marburg, Mannheim - Wien - Zürich 1984, S. 12-35.

Moerkotte (1990)
Moerkotte,G.: Inkonsistenzen in deduktiven Datenbanken - Diagnose und Reparatur, Informatik-Fachberichte 248, Berlin - Heidelberg - New York ... 1990.

Moore,R. (1980)
Moore,R.C.: Reasoning About Knowledge and Action, Technical Note No. 191, Stanford Research Institute International Inc., Menlo Park 1980.

Moore,R. (1982)
Moore,R.C.: The Role of Logic in Knowledge Representation and Commensense Reasoning; in: o.V.: Proceedings of the Second Annual National Conference on Artificial Intelligence AAAI-82, 18.-20.08.1982 in Pittsburgh, Menlo Park 1982, S. 428-433.

Morik (1982)
Morik,K.: Überzeugungssysteme der Künstlichen Intelligenz - Validierung vor dem Hintergrund linguistischer Theorien über implizite Äußerungen, Dissertation unter dem Titel "Validierung von Untersuchungssystemen der Künstlichen Intelligenz vor dem Hintergrund linguistischer Theorien über implizite Äußerungen", Universität Hamburg 1981, Tübingen 1982.

Morris,P. (1988)
Morris,P.H.: The Anomalous Extension Problem in Default Reasoning; in: Artificial Intelligence, Vol. 35 (1988), S. 383-399.

Moto-oka (1984)
Moto-oka,T.; Tanaka,H.; Aida,H.; Hirata,K.; Maruyama,T.: The Architecture of a Parallel Inference Engine -PIE-; in: o.V.: Fifth Generation Computer Systems 1984 - Proceedings of the International Conference, 06.-09.11.1984 in Tokyo, Tokyo - Amsterdam 1984, S. 479-488.

Müller,A. (1987)
Müller,A.: Produktionsplanung und Pufferbildung bei Werkstattfertigung, Dissertation unter dem Titel "Der Pufferbedarf im Rahmen der kurzfristigen Produktionsplanung bei Werkstattfertigung", Technische Hochschule Aachen 1986, Wiesbaden 1987.

Murata,Ta. (1986)
Murata,Ta.; Zhang,D.: A High-Level Petri Net Model for Parallel Interpretation of Logic Programs; in: o.V.: Proceedings of the IEEE Computer Society 1986 International Conference on Computer Languages, Washington 1986, S. 123-132.

Murata,Ta. (1987a)
Murata,Ta.; Peterka,G.: Application of Colored Petri Net T-Invariants to Logic Programming; in: o.V.: Proceedings of the Fifth International Conference on Systems Engineering, 1987 in Dayton, New York 1987, S. 381-384.

Murata,Ta. (1988a)
Murata,Ta.; Matsuyama,K.: Inconsistency Check of a Set of Clauses using Petri Net Reductions; in: Journal of the Franklin Institute, Vol. 325 (1988), No. 1, S. 73-93.

Murata,Ta. (1988b)
Murata,Ta.; Zhang,D.: A Predicate-Transition Net Model for Parallel Interpretation of Logic Programs; in: IEEE Transactions on Software Engineering, Vol. SE-14 (1988), S. 481-497.

Myhill (1952)
Myhill,J.: Some Philosophical Implications of Mathematical Logic; in: The Review of Metaphysics, Vol. 6 (1952), S. 165-198.

Myhill (1968)
Myhill,J.: The Formalization of Intuitionism; in: Klibansky,R. (Hrsg.): Contemporary Philosophy - A Survey, I: Logic and Foundations of Mathematics, Firenze 1968, S. 324-341.

Mylopoulos (1983)
Mylopoulos,J.; Levesque,H.: An Overview of Knowledge Representation; in: Neumann,B. (Hrsg.): GWAI-83, 7th German Workshop on Artificial Intelligence, 19.-23.09.1983 in Dassel/ Solling, Informatik-Fachberichte 76, Berlin - Heidelberg - New York ... 1983, S. 143-157.

Mylopoulos (1984)
Mylopoulos,J.; Levesque,H.J.: An Overview of Knowledge Representation; in: Brodie,M.L.; Mylopoulos,J.; Schmidt,J.W. (Hrsg.): On Conceptual Modelling - Perspectives from Artificial Intelligence, Databases, and Programming Languages, New York - Berlin - Heidelberg ... 1984, S. 3-17.

Nagel,E. (1964)
Nagel,E.; Newman,J.R.: Der Gödelsche Beweis, Wien - München 1964.

Nakano,R. (1983)
Nakano,R.: Integrity Checking in a Logic-Oriented ER Model; in: Davis,C.G.; Jajodia,S.; Ng,P.A.-B.; Yeh,R.T. (Hrsg.): Entity-Relationship Approach to Software Engineering, Proceedings of the Third International Conference on Entity-Relationship Approach, 05.-07.10.1983 in Anaheim, Amsterdam - New York - Oxford 1983, S. 551-564.

Nelson,R.J. (1982)
Nelson,R.J.: Artificial intelligence, philosophy and existence proofs; in: Hayes,J.E.; Michie,D.; Pao,Y.-H. (Hrsg.): Machine Intelligence 10: Intelligent Systems - Practice and Perspective, New York - Chichester - Brisbane ... 1982, S. 541-553.

Nett (1988)
Nett,E.: Transaktionsorientierte Systemarchitekturen - Ein Konzept zur Realisierung zuverlässiger, verteilter Datenverarbeitung; in: Gesellschaft für Mathematik und Datenverarbeitung mbH: Jahresbericht 1987, Sankt Augustin 1988, S. 54-65.

Newell (1982a)
Newell,A.: The Knowledge Level; in: Artificial Intelligence, Vol. 18 (1982), S. 87-127.

Nielsen,M. (1984b)
Nielsen,M.: Logic Programming - Viewed as Programming with Concurrency, Vortrag am 29.06.1984 anläßlich: 5th European Workshop on Application and Theory of Petri Nets, 27.-29.06.1984 in Aarhus.

Nilsson,N. (1980a)
Nilsson,N.J.: Principles of Artificial Intelligence, Palo Alto 1980.

Oberschelp (1962)
Oberschelp,A.: Untersuchungen zur mehrsortigen Quantorenlogik; in: Mathematische Annalen, 145. Bd. (1962), S. 297-333.

Oberweis (1987b)
Oberweis,A.; Schönthaler,F.; Seib,J.; Lausen,G.: Database Supported Analysis Tool for Predicate/Transition Nets; in: Special Interest Group on Petri Nets and related System Models, Gesellschaft für Informatik, Newsletter 28 (1987), S. 21-23.

Oberweis (1988a)
Oberweis,A.; Seib,J.: PASIPP: Ein Programm zur Analyse und Simulation von Prolog-beschrifteten Prädikate/Transitionen-Netzen (Benutzerhandbuch), o.O. (Mannheim) 1988.

Oberweis (1988b)
Oberweis,A.: Checking Database Integrity Constraints while Simulating Information System Behaviour; in: o.V.: Application and Theory of Petri Nets - 9th European Workshop, 22.-24.06.1988 in Venedig, o.O. 1988, Vol. I, S. 299-308.

Oberweis (1989a)
Oberweis,A.; Seib,J.; Lausen,G.: Ein Programm zur Analyse und Simulation von Prädikate-Transitionen-Netzen, Interner Bericht, Fakultät für Mathematik und Informatik, Universität Mannheim, Mannheim 1989.

Oberweis (1990a)
Oberweis,A.: Zeitstrukturen für Informationssysteme, Dissertation, Universität Mannheim, Mannheim 1990.

Ohlbach (1989)
Ohlbach,H.J.: Context Logic - An Introduction; in: Metzing,D. (Hrsg.): GWAI-89, 13th German Workshop on Artificial Intelligence, 18.-22.09.1989 in Eringerfeld, Proceedings, Informatik-Fachberichte 216, Berlin - Heidelberg - New York ... 1989, S. 27-36.

Ohsuga (1983)
Ohsuga,S.: Knowledge Based Man-Machine Systems; in: Johannsen,G.; Rijnsdorp,J.E. (Hrsg.): Analysis, Design and Evaluation of Man-Machine Systems, Proceedings of the IFAC/IFIP/IFORS/IEA Confernce, 27.-29.09.1982 in Baden-Baden, Oxford - New York - Toronto ... 1983, S. 303-310.

Opp,K. (1976)
Opp,K.-D.: Methodologie der Sozialwissenschaften - Einführung in Probleme ihrer Theorienbildung, Neuausgabe, Reinbek 1976.

Ottmann (1975a)
Ottmann,T.: Einfache universelle mehrdimensionale Turingmaschinen, Habilitationsschrift, Universität Karlsruhe, Karlsruhe 1975.

o.V. (1981b)
o.V. (The Xerox Learning Research Group): The Smalltalk-80 System; in: Byte, Vol. 6 (1981), Heft August 1981, S. 36-47.

o.V. (1989g)
o.V.: Analyse und Simulation von verteilten Systemen, Prospekt, Fakultät für Mathematik und Informatik, Universität Mannheim, Mannheim o.J. (1989).

o.V. (1989h)
o.V.: ESPRIT Project 125 GRASPIN - Personal Workstation for Incremental Graphical Specification and Formal Implementation of Nonsequential Systems - A joint European R&D project in the field of software technology, sponsored by the Commission of the European Communities in the frame of the European Strategic Programme for Research and Development in Information Technology (ESPRIT): Project Publications - Abstract Service, Report GRA 25/9, o.O. 1989.

o.V. (1990a)
o.V.: 11th International Conference on Application and Theory of Petri Nets, 27.-29.06.1990 in Paris, o.O. 1990.

Owsnicki-Klewe (1988)
Owsnicki-Klewe,B.: Configuration as a Consistency Maintenance Task; in: Hoeppner,W. (Hrsg.): Künstliche Intelligenz - GWAI-88, 12. Jahrestagung, 19.-23.09.1988 in Eringerfeld, Proceedings, Informatik-Fachberichte 181, Berlin - Heidelberg - New York ... 1988, S. 77-87.

Pacini (1983)
Pacini,G.; Turini,F.: Demonizing Production Systems; in: Bundy,A. (Hrsg.): IJCAI-83, Proceedings of the Eigth International Joint Conference on Artificial Intelligence, 08.-12.08.1983 in Karlsruhe, Los Altos 1983, Vol. 2, S. 862-866.

Paul,W. (1978)
Paul,W.J.: Komplexitätstheorie, Stuttgart 1978.

Perelman (1979)
Perelman,C.: Juristische Logik als Argumentationslehre, Freiburg - München 1979.

Peter (1957)
Peter,R.: Rekursive Funktionen, 2. Aufl., Berlin 1957.

Peterson,J. (1973a)
Peterson,J.L.: Modelling of Parallel Systems, Dissertation am Department of Electrical Engineering, Stanford University, Stanford 1973.

Peterson,J. (1976)
Peterson,J.L.: Computation Sequence Sets; in: Journal of Computer and System Sciences, Vol. 13 (1976), S. 1-24.

Peterson,J. (1981)
Peterson,J.L.: Petri Net Theory and the Modeling of Systems, Englewood Cliffs 1981.

Pique (1988)
Pique,J.-F.: Prolog II - A Step On the Prolog Road; in: AI Communications, Vol. 1 (1988), No. 2, S. 4-16.

Plümer (1986)
Plümer,L.: Und-Parallelismus und effizientes Backtracking von Prolog-Prozeduren; in: Hommel, G.; Schindler,S. (Hrsg.): GI - 16. Jahrestagung I: Informatik-Anwendungen - Trends und Perspektiven, 06.-10.10.1986 in Berlin, Proceedings, Informatik-Fachberichte 126, Berlin - Heidelberg - New York ... 1986, S. 137-150.

Pnueli (1977)
Pnueli,A.: The Temporal Logic of Programs; in: o.V.: Proceedings of the 18th Annual Symposium on Foundations of Computer Science, Providence 1977, S. 46-57.

Pnueli (1979)
Pnueli,A.: The Temporal Semantics of Concurrent Programs; in: Kahn,G. (Hrsg.): Semantics of Concurrent Computation, Proceedings of the International Symposium, 2.-4.07.1979 in Evian, Lecture Notes in Computer Science 70, Berlin - Heidelberg - New York 1979, S. 1-20.

Pnueli (1986)
Pnueli,A.: Applications of Temporal Logic to the Specification and Verification of Reactive Systems: A Survey of Current Trends; in: de Bakker,J.W.; de Roever,W.-P.; Rozenberg,G. (Hrsg.): Current Trends in Concurrency - Overviews and Tutorials, Lecture Notes in Computer Sciene 224, Berlin - Heidelberg - New York ... 1986, S. 510-584.

Popper (1984b)
Popper,K.R.: Objektive Erkenntnis - Ein evolutionärer Entwurf, 4. Aufl., Hamburg 1984.

Post,E. (1936)
Post,E.L.: Finite Combinatory Processes - Formulation I; in: The Journal of Symbolic Logic, Vol. 1 (1936), S. 103-105.

Post,E. (1944)
Post,E.,L.: Recursively Enumerable Sets of Positive Integers and Their Decison Problems; in: Bulletin of the American Mathematical Society, Vol. 50 (1944), S. 284-316.

Potthoff (1981)
Potthoff,K.: Einführung in die Modelltheorie und ihre Anwendungen, Darmstadt 1981.

Powers (1988)
Powers,D.M.W.; Davila,L.; Wrightson,G.: Implementing Connection Graphs for Logic Programming; in: Trappl,R. (Hrsg.): Cybernetics and Systems'88, Proceedings of the Ninth European Meeting on Cybernetics and Systems Research, 05.-08.04.1988 in Wien, Dordrecht - Boston - London ... 1988, Part 2, S. 957-964.

Preiß (1989)
Preiß,N.: Ein Konzept für die deduktive Erweiterung eines relationalen Datenbanksystems, Dissertation, Technische Hochschule Karlsruhe, Karlsruhe 1989.

PROLOG (o.J.)
o.V.: Turbo Prolog Owner's Handbook, o.O. o.J.

Putnam,H. (1966)
Putnam,H.: Minds and Machines; in: Hook,S. (Hrsg.): Dimensions of Mind, 3. Aufl., New York - London 1966, S. 138-164.

Putnam,H. (1973)
Putnam,H.: Recursive Functions and Hierarchies; in: Papers in the Foundation of Mathematics, published as a supplement to the American Mathematical Monthly, Vol. 80 (1973), S. 68-86.

Putnam,H. (1982a)
Putnam,H.: Vernunft, Wahrheit und Geschichte, Frankfurt 1982.

Putnam,H. (1982b)
Putnam,H.: Modell und Wirklichkeit; in: Conceptus, 16. Jg. (1982), Nr. 38, S. 9-30.

Pylyshyn (1980)
Pylyshyn,Z.W.; Bledsoe,W.W.; Feigenbaum,E.A.; Newell,A.; Nilsson,N.; Reddy,D.R.; Rosenfeld,A.; Winograd,T.; Winston,P.: Artificial Intelligence; in: Arden,B.W. (Hrsg.): What Can Be Automated? - The Computer Science and Engineering Research Study (COSERS), Cambridge (Massachusetts) - London 1980, S. 421-509.

Qadah (1987)
Qadah,G.Z.; Nussbaum,M.: Logic machines: A survey; in: o.V.: AFIPFS Conference Proceedings, Vol. 56: 1987 National Computer Conference, 15-18-06.1987 in Chicago, Reston 1987, S. 265-278.

Queille (1982b)
Queille,J.P.; Sifakis,J.: A Temporal Logic to Deal with Fairness in Transition Systems; in: o.V.: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, 3.-5.11.1982 in Chicago, New York 1982, S. 217-225.

Quine (1975b)
Quine,W.V.O.: Ontologische Relativität - und andere Schriften, Stuttgart 1975.

Quine (1985)
Quine,W.V.O.: Theorien und Dinge, Frankfurt 1985.

Rado (1962)
Rado,T.: On Non-Computable Functions; in: The Bell System Technical Journal, Vol. 41 (1962), S. 877-884.

Ramsey,F. (1965)
Ramsey,F.P.: The Foundations of Mathematics - and other Logical Essays, hrsg. von R.B: Braithwaite, 4. Druck, London 1965.

Raphael (1976)
Raphael,B.: The Thinking Computer - Mind Inside Matter, San Francisco 1976.

Raulefs (1982a)
Raulefs,P.: Expertensysteme; in: Bibel,W.; Siekmann,J.H. (Hrsg.): Künstliche Intelligenz, Frühjahrsschule Teisendorf, 15.-24.03.1982 in Teisendorf, Informatik-Fachberichte 59, Berlin - Heidelberg - New York 1982, S. 61-98.

Raulefs (1982b)
Raulefs,P.: Methoden der Künstlichen Intelligenz: Übersicht und Anwendungen in Expertensystemen; in: Nehmer,J. (Hrsg.): GI - 12. Jahrestagung, 05.-07.10.1982 in Kaiserslautern, Proceedings, Informatik-Fachberichte 57, Berlin - Heidelberg - New York 1982, S. 170-187.

Raulefs (1984)
Raulefs,P.: Foundation of Expert Systems for Conceptional Design in Mechanical Engineering, Memo SEKI-84-08, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern 1984.

Rautenberg (1979)
Rautenberg,W.: Klassische und nichtklassische Aussagenlogik, Braunschweig - Wiesbaden 1979.

Razouk (1985c)
Razouk,R.R.; Morgan,E.T.: The P-NUT System: An Environment for Modeling and Analyzing Concurrent Systems, Technical Report No. 85-10, Department of Information and Computer Science, University of California, Irvine 1985.

Reck (1988)
Reck,M.: Von informellen zu formalen Spezifikationen durch Petri-Netze und abstrakte Datentypen, Diplomarbeit, Lehrstuhl 1, Fachbereich Informatik, Universität Dortmund, Dortmund 1988.

Reichenbach (1977)
Reichenbach,H.: Gesammelte Werke in 9 Bänden, Band 1: Der Aufstieg der wissenschaftlichen Philosophie, hrsg. von Kamlah,A.; Reichenbach,M., Braunschweig 1977.

Reimer (1989)
Reimer,U.: FRM: Ein Frame-Repräsentationsmodell und seine formale Semantik - Zur Integration von Datenbank- und Wissensrepräsentationsansätzen, (erweiterte) Dissertation, Universität Konstanz, Informatik-Fachberichte 198, Berlin - Heidelberg - New York ... 1989.

Reinfrank (1985a)
Reinfrank,M.: Nicht-Monotones Argumentieren (Non-Monotonic Reasoning); in: Informatik-Spektrum, Bd. 8 (1985), S. 92.

Reinfrank (1985b)
Reinfrank,M.: An Introduction to Non-Monotonic Reasoning, MEMO-SEKI-85-02, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern 1985.

Reisig (1985b)
Reisig,W.: Petri Nets: An Introduction, Berlin - Heidelberg - New York - Tokyo 1985. (Anmk. des Verf.: englische Übersetzung der deutschen Originalfassung (1. Aufl.: 1982); deren 2. Aufl. hier als Reisig (1986a).)

Reisig (1985d)
Reisig,W.: Petri Nets with Individual Tokens; in: Theoretical Computer Science, Vol. 41 (1985), S. 185-213.

Reisig (1986a)
Reisig,W.: Petrinetze - Eine Einführung, 2. Aufl., Berlin - Heidelberg - New York Tokyo 1986. (Anmk. des Verf.: englische Übersetzung der 1. Aufl. (1982) erschienen als Reisig (1985b).)

Reisig (1989a)
Reisig,W.: Petri Nets and Abstract Data Types, Bericht TUM-I8904, Institut für Informatik, Technische Universität München, München 1989.

Reisig (1989b)
Reisig,W.: Cover Picture Story - The Container Crane System; in: Petri Net Newsletter (Gesellschaft für Informatik: Special Interest Group on Petri Nets and related System Models), No. 34 (1989), S. 3-9.

Reiter (1978b)
Reiter,R.: On Closed World Data Bases; in: Gallaire,H.; Minker,J. (Hrsg.): Logic and Data Bases, New York - London 1978, S. 55-76.

Reiter (1980)
Reiter,R.: A Logic for Default Reasoning; in: Artificial Intelligence, Vol. 13 (1980), S. 81-132.

Reiter (1984)
Reiter,R.: Towards a Logical Reconstruction of Relational Database Theory; in: Brodie,M.L.; Mylopoulos,J.; Schmidt,J.W. (Hrsg.): On Conceptual Modelling - Perspectives from Artificial Intelligence, Databases, and Programming Languages, New York - Berlin - Heidelberg ... 1984, S. 191-233.

Reitman (1964)
Reitman,W.R.: Heuristic decision procedures, open constraints, and the structure of ill-defined problems; in: Shelly,M.W.; Bryan,G. (Hrsg.): Human Judgements and Optimability, New York - London - Sydney 1964, S. 282-315.

Rescher (1968a)
Rescher,N.: Recent Developments in Philosophical Logic; in: Klibansky,R. (Hrsg.): Contemporary Philosophy - A Survey, Vol. I: Logic and Foundations of Mathematics, Firenze 1968, S. 31-40.

Rescher (1968b)
Rescher,N.: Conspectus of Recent Work in Many-Valued Logic; in: Klibansky,R. (Hrsg.): Contemporary Philosophy - A Survey, Vol. I: Logic and Foundations of Mathematics, Firenze 1968, S. 74-86.

Rescher (1968c)
Rescher,N.: Chronological Logic; in: Klibansky,R. (Hrsg.): Contemporary Philosophy - A Survey, Vol. I: Logic and Foundations of Mathematics, Firenze 1968, S. 123-134.

Rescher (1969)
Rescher,N.: Lawfulness as Mind-Dependent; in: Rescher,N. (Hrsg.): Essays in Honor of Carl G. Hempel - A Tribute on the Occasion of his Sixty-Fifth Birthday, Dordrecht 1969, S. 178-197.

Rescher (1971)
Rescher,N.; Urquhart,A.: Temporal Logic, Wien 1971.

Rescher (1977a)
Rescher,N.: Methodological Pragmatism - A Systems-Theoretic Approach to the Theory of Knowledge, New York 1977.

Rescher (1980a)
Rescher,N.: Scepticism - A critical reappraisal, Oxford 1980.

Rescher (1982a)
Rescher,N.: The Coherence Theory of Truth, Washington 1982. (Nachdruck der Ausgabe Oxford 1973.)

Rescher (1985a)
Rescher,N.: The Strife of Systems - An Essay on the Grounds and Implications of Philosophical Diversity, Pittsburgh 1985.

Rescher (1987a)
Rescher,N.: Induktion - Zur Rechtfertigung induktiven Schließens, München - Wien 1987.

Reuter (1987)
Reuter,A.: Maßnahmen zur Wahrung von Sicherheits- und Integritätsbedingungen; in: Lockemann,P.C.; Schmidt,J.W. (Hrsg.): Datenbank-Handbuch, Berlin - Heidelberg - New York ... 1987, S. 337-479.

Rheinwald (1989)
Rheinwald,R.: Zur These der Extensionalität in der Mathematik, Manuskript eines Vortrags anläßlich: 14. Internationales Wittgenstein-Symposium, 13.-20.08.1989 in Kirchberg, o.O. (Bielefeld) o.J. (1989).

Richter,M.M. (1978)
Richter,M.(M.): Logikkalküle, Stuttgart 1978.

Richter,M.M. (1988)
Richter,M.M.: Künstliche Intelligenz und Logik; in: Rahmstorf,G. (Hrsg.): Wissensrepräsentation in Expertensystemen, Workshop, 16.-18.03.1987 in Herrenberg, Proceedings, Informatik-Fachberichte 172, Berlin - Heidelberg - New York ... 1988, S. 16-40.

Riedemann (1979)
Riedemann,E.H.: The Control of Parallel Computations by Labelled Petri Nets: A Study in Terms of Multiple-Firing Automata and Parallel Program Schemata, Dissertation, Universität Dortmund, Dortmund 1979.

Robinson,J. (1965)
Robinson,J.A.: A Machine-Oriented Logic Based on the Resolution Principle; in: Communications of the ACM, Vol. 12 (1965), S. 23-41. (Anmk. des Verf.: auch veröffentlicht in: Siekmann, J.; Wrightson,G. (Hrsg.): Automation of Reasoning 1: Classical Papers on Computational Logic 1957-1966, Berlin - Heidelberg - New York 1983, S. 397-415.)

Robinson,J. (1975)
Robinson,J.A.: Über den Bau von Deduktionsmaschinen; in: Findler,N.V. (Hrsg.): Künstliche Intelligenz und heuristisches Programmieren, Wien - New York 1975, S. 3-13.

Rogers,H. (1967)
Rogers,H.: Theory of Recursive Functions and Effective Computability, New York - Saint Louis ... 1967.

Roggenbuck (1989)
Roggenbuck,S.; Gebhardt,R.; Ameling,W.: Prolog als Methodensprache in einer objektorientierten Programmierumgebung; in: Angewandte Informatik, 31. Jg. (1989), S. 181-188.

Rose,T. (1986)
Rose,T.; Appelrath,H.-J.; Bense,H.: Controlled Prolog - A Fornt-End to Prolog Incorporating Meta Knowledge; in: Rollinger,C.-R.; Horn,W. (Hrsg.): GWAI-86 und 2. Österreichische Artificial-Intelligence-Tagung, 22.-26.09.1986 in Ottenstein, Informatik-Fachbereichte 124, Berlin - Heidelberg - New York ... 1986, S. 301-311.

Rosen (1975)
Rosen,B.K.: Correctness of Parallel Programs: The Church-Rosser Approach; in: o.V.: Proving and Improving Programs, Papers of a Symposium, im Juli 1975, Arc et Senans 1975, S. 115-132.

Rosen (1976)
Rosen,B.K.: Correctness of Parallel Programs: The Church-Rosser Approach; in: Theoreticla Computer Science, Vol. 2 (1976), S. 183-207.

Rosenschein (1981)
Rosenschein,S.J.: Plan Synthesis: A Logical Perspective; in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, Vol. 1, o.O. (Menlo Park) 1981, S. 331-337.

Rosser (1936)
Rosser,B.: Extensions of some Theorems of Gödel and Church; in: The Journal of Symbolic Logic, Vol. 1 (1936), S. 87-91.

Ruspini (1987a)
Ruspini,E.H.: Issues in the Representation of Imprecision and Uncertainty in Information Systems; in: Bezdek,J.C. (Hrsg.): Analysis of Fuzzy Information, Vol. VV: Aertificial Intelligence and Decision Systems, Boca Raton 1987, S. 231-239.

Russell,B. (1921)
Russell,B.: Vorwort (zu: Wittgenstein,L.: Logisch-Philosophische Abhandlung.); in: Annalen der Natusphilosophie, Bd. 14 (1921), S. 186-198.

Ruzicka (1988)
Ruzicka,P.; Privara,I.: An Almost Linear Robinson Unification Algorithm; in: Chytil,M.P.; Janiga,L.; Koubek,V. (Hrsg.): MFCS'88 - Mathematical Foundations of Computer Science, Proceedings of the 13th Symposium, 29.08.-02.09.1988 in Carlsbad, Lecture Notes in Computer Science 324, Berlin - Heidelberg - New York ... 1988, S. 501-511.

Saake (1988)
Saake,G.; Lipeck,U.W.: Foundations of Temporal Integrity Monitoring; in: Rolland,C.; Bodart,F.; Leonard,M. (Hrsg): Temporal Aspects in Information Systems, Proceedings of the IFIP TC 8/WG 8.1 Working Conference on Temporal Aspects in Information Sytems, 13.-15.05.1987 in Sophia-Antipolis, Amsterdam - New York - Oxford ... 1988, S. 235-249.

Sahraoui (1987)
Sahraoui,A.; Atabakhche,H.; Courvoisier,M.; Valette,R.: Joining Petri Nets and Knowledge Based Systems for Monitoring Purposes; in: o.V.: Robotics and Automation, Proceedings of the 1987 IEEE International Conference on Robotics and Automation, 31.03.-3.04.1987 in Raleigh, Vol. 2, Washington 1987, S. 1160-1165.

Sahraoui (1988)
Sahraoui,A.E.K.: Timing Constraints Problems in Real-Time Systems: A Survey, Rapport LAAS N. 88346, Laboratoire d'Automatique et d'Analyse des Systemes du Centre National de la Recherche Scientifique, Toulouse 1988.

Sandewall (1972)
Sandewall,E.: An Approach to the Frame Problem and its Implementation; in: Meltzer,B.; Michie,D. (Hrsg.): Machine Intelligence 7, Edinburgh 1972, S. 195-204.

Sauer (1988a)
Sauer,J.; Appelrath,H.-J.: Wissensbasierte Feinplanung in PPS-Systemen; in: Dillmann,R.; Swiderski,D. (Hrsg.): WIMPEL'88, 1. Konferenz über Wissensbasierte Mathoden für Produktion, Engineering und Logistik, 28.-30.06.1988 in München, Stuttgart 1988, S. 363-379.

Sauer (1988b)
Sauer,J.: Model of the PROTOS-Production-Planning-Problem; in: Appelrath,H.-J.; Cremers,A. B.; Schiltknecht,H. (Hrsg.): Prolog Tools for Building Expert Systems: First Work Shop, 12.-14. 09.1988 in Morcote, Basel 1988, S. 247-262.

Sauerwein (1984)
Sauerwein,W.: Planerstellen mit besonderer Berücksichtigung der Zeit-Komponente, Diplomarbeit, Institut für Informatik, Universität Bonn, Bonn 1984.

Savage (1976)
Savage,J.E.: The Complexity of Computing, New York - London - Sydney ... 1976.

Scharf,A. (1987)
Scharf,A.: Expertensysteme - Repräsentation des Wissens; in: Hard and Soft, Ausgabe Mai 1987, S. 22-30.

Scheer (1987a)
Scheer,A.-W.: EDV-orientierte Betriebswirtschaftslehre, 3. Aufl, Berlin - Heidelberg - New York ... 1987. (Anmk. des Verf.: passagenweise über Scheer (1990c) hinausreichend.)

Scheer (1987b)
Scheer,A.-W.: Neue Architektur für EDV-Systeme zur Produktionsplanung und -steuerung; in: Adam,D. (Hrsg.): Neuere Entwicklungen in der Produktions- und Investitionspolitik, Wiesbaden 1987, S. 153-176.

Scheer (1989c)
Scheer,A.-W.: Enterprise-Wide Data Modelling - Information Systems in Industry, Berlin - Heidelberg - New York ... 1989.

Scheer (1990a)
Scheer,A.-W.; Herterich,R.; Klein,J.: INMAS - Eine individuell konfigurierbare Schnittstelle; in: Information Management, 5. Jg. (1990), Heft 1, S. 16-26.

Scheer (1990c)
Scheer,A.-W.: EDV-orientierte Betriebswirtschaftslehre - Grundlagen für ein effizientes Informationsmanagement, 4. Aufl., Berlin - Heidelberg - New York ... 1990.

Scheer (1991d)
Scheer,A.-W.: Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung, Berlin - Heidelberg - New York ... 1991.

Schefe (1982)
Schefe,P.: Some fundamental issues in knowledge representation; in: Wahlster,W. (Hrsg.): GWAI-82, 6th German Workshop on Artificial Intelligence, 27.09.-01.10.1982 in Bad Honnef, Informatik-Fachbereichte 58, Berlin - Heidelberg - New York 1982, S. 42-62.

Schlageter (1983)
Schlageter,G.; Stucky,W.: Datenbanksysteme: Konzepte und Modelle, 2. Aufl., Stuttgart 1983.

Schmidt,A. (1938)
Schmidt,A.: Über deduktive Theorien mit mehreren Sorten von Grunddingen; in: Mathematische Annalen, 115. Bd. (1937/38), S. 485-506.

Schmidt,A. (1951)
Schmidt,A.: Die Zulässigkeit der Behandlung mehrsortiger Theorien mittels der üblichen einsortigen Prädikatenlogik.; in: Mathematische Annalen, 123. Bd. (1951), S. 187-200.

Schmidt,H.W. (1984a)
Schmidt,H.W.: Towards a Net-Theoretic Notion of Type based on Predicate-Transition Nets; in: o.V.: Papers presented at the 5th European Workshop on Applications and Theory of Petri Nets, 27.-29.06.1984 in Aarhus, o.O. 1984, S. 194 u. 330-345.

Schmidt,H.W. (1984b)
Schmidt,H.-W.: Towards a Net-Theoretic Notion of Type based Predicate-Transition Nets, Arbeitspapiere der GMD Nr. 117, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1984. (Inhaltlich weitgehend, aber nicht vollständig identisch mit Schmidt,H. (1984a).)

Schmidt,H.W. (1985)
Schmidt,H.W.; Papazoglou,M.: Abstract Modular Implementation of Processes by Predicate-Event Systems, Technical Paper, ESPRIT Project 125 - GRASPIN, Sankt Augustin - Ivrea - Pisa ... 1985.

Schmitt,P. (1988)
Schmitt,P.H.: Vererbungshierarchien und Prädikatenlogik; in: Rahmstorf,G. (Hrsg.): Wissensrepräsentation in Expertensystemen, Workshop, 16.-18.03.1987 in Herrenberg, Proceedings, Informatik-Fachberichte 172, Berlin - Heidelberg - New York ... 1988, S. 78- 97.

Schmitz,L. (1988b)
Schmitz,L.: Objektorientierte Programmierung, Teil 2: Ein bißchen Smalltalk; in: PC Magazin, o.Jg. (1988), Nr. 52, S. 52-58.

Schnorr (1974)
Schnorr,C.P.: Rekursive Funktionen und ihre Komplexität, Stuttgart 1974.

Schnupp (1989)
Schnupp,P.; Nguyen Huu,C.T.; Bernhard,L.W.: Expert Systems Lab Course, Berlin - Heidelberg - New York ... 1989.

Schönfeld,W. (1988)
Schönfeld,W.: Anforderungen der Logik-Programmierung an die Wissensrepräsentation; in: Rahmstorf,G. (Hrsg.): Wissensrepräsentation in Expertensystemen, Workshop, 16.-18.03.1987 in Herrenberg, Proceedings, Informatik-Fachberichte 172, Berlin - Heidelberg - New York ... 1988, S. 41-55.

Scholz (1988)
Scholz,B.: CIM-Schnittstellen - Konzepte, Standards und Probleme der Verknüpfung von Systemkomponenten in der rechnerintegrierten Produktion, München - Wien 1988.

Schurz (1983a)
Schurz,G.: Wissenschaftliche Erklärung - Ansätze zu einer logisch-pragmatischen Wissenschaftstheorie, Dissertation, Universität Graz, Graz 1983.

Schwemmer (1981a)
Schwemmer,O. (Hrsg.): Vernunft, Handlung und Erfahrung - Über die Grundlagen und Ziele der Wissenschaften, München 1981.

Schwind (1982)
Schwind,C.B.: Natural Language Access to PROLOG Database Systems; in: Wahlster,W. (Hrsg.): GWAI-82, 6th German Workshop on Artificial Intelligence, 27.09.-01.10.1982 in Bad Honnef, Informatik-Fachberichte 58, Berlin - Heidelberg - New York 1982, S. 237-246.

Schwind (1985)
Schwind,C.B.: Temporal Logic in Artificial Intelligence; in: Laubsch,J. (Hrsg.): GWAI-84, 8th German Workshop on Artificial Intelligence, 08.-12.10.1984 in Wingst, Informatik-Fachberichte 103, S. 238-264.

Seiffert (1983a)
Seiffert,H.: Einführung in die Wissenschaftstheorie, Erster Band: Sprachanalyse - Deduktion - Induktion in Natur- und Sozialwissenschaften, 10. Aufl., München 1983.

Sergot (1986)
Sergot,M.J.; Sadri,F.; Kowalski,R.A.; Kriwaczek,F.; Hammond,P.; Cory,H.T.: The British Nationality Act as a Logic Program; in: Communications of the ACM, Vol. 29 (1986), S. 360-386.

Sethi (1974)
Sethi,R.: Testing for the Church-Rosser Property; in: Journal of the Association for Computing Machinery, Vol. 21 (1974), S. 671-679.

Shapiro,E. (1983)
Shapiro,E.; Takeuchi,A.: Object Oriented Programming in Concurrent Prolog; in: New Generation Computing, Vol. 1 (1983), S. 25-48.

Shapiro,S.C. (1979a)
Shapiro,S.C.: Techniques of Artificial Intelligence, New York - Cincinnati - Toronto ... 1979.

Shapiro,S.C. (1979b)
Shapiro,S.C.: Numerical Quantifiers and their Use in Reasoning with negative Information; in: o.V.: IJCAI-79, Proceedings of the Sixth International Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, Vol. 2, o.O. (Stanford) 1979, S. 791-796.

Sharpe (1985)
Sharpe,W.P.: Logic Programming for the Law; in: Braines,W.A. (Hrsg.): Research and Development in Expert Systems: Proceedings of the 4th Technical Conference of the British Computer Society Special Group on Expert Systems, 1985 in Warwick, Cambridge (Großbritannien) 1985, S. 217-228.

Sheffer (1913)
Sheffer,H.M.: A set of five independent postulates for Boolean algebras, with application to logical constants; in: Transactions of the American Mathematical Society, Vol. 14 (1913), S. 481-488.

Shepherdson (1963)
Shepherdson,J.C.; Sturgis,H.E.: Computability of Recursive Functions; in: Journal of the Association for Computing Machinery (ACM), Vol. 10 (1963), S. 217-255.

Shih (1987)
Shih,Y.-F.: Parallel Processing of Production Systems: An Integrated Software and Hardware Approach, Dissertation, University of Michigan, Ann Arbor 1987.

Shoham (1988c)
Shoham,Y.: Chronological Ignorance: Experiments in Nonmonotonic Reasoning; in: Artificial Intelligence, Vol. 36 (1988), S. 279-331.

Sickel (1977)
Sickel,S.: Variable Range Restrictions in Resolution Theorem Proving; in: Elcock,E.W.; Michie, D. (Hrsg.): Machine Intelligence 8: Machine Representations of Knowledge, New York - Chichester - Brisbane ... 1977, S. 73-85.

Siekmann (1982a)
Siekmann,J.; Szabo,P.: Universal Unification; in: Wahlster,W. (Hrsg.): GWAI-82, 6th German Workshop on Artificial Intelligence, 27.09.-01.10.1982 in Bad Honnef, Informatik-Fachberichte 58, Berlin - Heidelberg - New York 1982, S. 102-141.

Siekmann (1983a)
Siekmann,J.; Wrightson,G. (Hrsg.): Automation of Reasoning 1: Classical Papers on Computational Logic 1957-1966, Berlin - Heidelberg - New York 1983.

Siekmann (1987)
Siekmann,J.H.: Geschichte und Anwendungen; in: Bläsius,K.H.; Bürckert,H.-J. (Hrsg.): Deduktionssysteme - Automatisierung des logischen Denkens, München - Wien 1987, S. 3-21.

Siklossy (1974)
Siklossy,L.; Roach,J.: Collaborative problem-solving between optimistic and pessimistic problem solvers; in: Rosenfeld,J.L. (Hrsg.): Information Processing 74, Proceedings of IFIP Congress 74, 05.-10.08.1974 in Stockholm, Amsterdam - London - New York 1974, S. 814-817.

Siklossy (1975)
Siklossy,L.; Roach,J.: Model Verification and Improvement Using DISPROVER; in: Artificial Intelligence, Vol. 6 (1975), S. 41-52.

Sinachopoulos (1989)
Sinachopoulos,A.: Logics for Petri-nets: Partial order logics, Branching time logics and how to distinguish between them; in: Petri Net Newsletter (Gesellschaft für Informatik: Special Interest Group on Petri Nets and related System Models), No. 33 (1989), S. 9-14.

Skolem (1934)
Skolem,T.: Über die Nicht-charakterisierbarkeit der Zahlenreihe mittels endlich oder abzählbar unendlich vieler Aussagen mit ausschliesslich Zahlenvariablen; in: Fundamenta Mathematicae, Tom 23 (1934), S. 150-161. (Anmk. des Verf.: "charakterisierbarkeit" im Original.)

Slage (1972)
Slage,J.R.: Einführung in die heuristische Programmierung, Künstliche Intelligenz und intelligente Maschinen, München 1972.

Smolka (1988a)
Smolka,G.: Logic Programming with Polymorphically Order-Sorted Types; in: Appelrath,H.-J.; Cremers,A.B.; Schiltknecht,H. (Hrsg.): Prolog Tools for Building Expert Systems: First Work Shop, 12.-14.09.1988 in Morcote, Basel 1988, S. 183-212.

Smullyan (1963)
Smullyan,R.M.: A Unifying Principal in Quantification Theory; in: o.V.: Proceedings of the National Academy of Sciences of the United States of America, Vol. 49 (1963), S. 828-832.

Smullyan (1968)
Smullyan,R.M.: First-Order Logic, Berlin - Heidelberg - New York 1968.

Smullyan (1989)
Smullyan,R.: Logik-Ritter und andere Schurken - Diabolische Rätsel, interplanetarische Verwicklungen und Gödelsche Systeme, Frankfurt 1989.

Sowa (1984)
Sowa,J.F.: Conceptual Structures: Information Processing in Mind and Machine, Reading - Menlo Park - London ... 1984.

Specht,D. (1989)
Specht,D.: Wissensbasierte Systeme im Produktionsbetrieb, Habilitationsschrift, Technische Universität Berlin 1988, München - Wien 1989.

Spur (1988c)
Spur,G.: Einführungsstrategie zu CIM; in: VDI-Z(eitschrift), Bd. 130 (1988), Nr. 10, S. 12-14.

Stachowiak (1973)
Stachowiak,H.: Allgemeine Modelltheorie, Wien - New York 1973.

Stallman (1977)
Stallman,R.M.; Sussman,G.J.: Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis; in: Artificial Intelligence, Vol. 9 (1977), S. 135-196.

Stede (1984)
Stede,M. (et al.): Einführung in die künstliche Intelligenz, Bd. 2: Anwendungsgebiete, Sprendlingen 1984.

Stefik (1983a)
Stefik,M.; Aikins,J.; Balzer,R.; Benoit,J.; Birnbaum,L.; Hayes-Roth,F.; Sacerdoti,E.: Basic Concepts for Building Expert Systems; in: Hayes-Roth,F.; Waterman,D.A.; Lenat,D.B. (Hrsg.): Building Expert Systems, Reading - London - Amsterdam ... 1983, S. 59-86.

Stegmüller (1968)
Stegmüller,W.: Das Wahrheitsproblem und die Idee der Semantik - Eine Einführung in die Theorien von A. Tarski und R. Carnap, 2. Aufl., Wien - New York 1968.

Stegmüller (1969)
Stegmüller,W.: Metaphysik - Skepsis - Wissenschaft, 2. Aufl., Berlin - Heidelberg - New York 1969.

Stegmüller (1970a)
Stegmüller,W.: Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie, Bd. II: Theorie und Erfahrung (Erster Halbband), Berlin - Heidelberg - New York 1970.

Stegmüller (1973a)
Stegmüller, Unvollständigkeit und Unentscheidbarkeit - Die metamathematischen Resultate von Gödel, Church, Kleene, Rosser und ihre erkenntnistheoretische Bedeutung, 3. Aufl., Wien - New York 1973.

Stegmüller (1976b)
Stegmüller,W.: Hauptströmungen der Gegenwartsphilosophie - Eine kritische Einführung, Bd. 1, 6. Aufl., Stuttgart 1976.

Stegmüller (1983)
Stegmüller,W.: Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie, Bd. I: Erklärung - Begründung - Kausalität, 2. Aufl., Berlin - Heidelberg - New York 1983.

Stegmüller (1984b)
Stegmüller,W.; von Kibed,M.V.: Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie, Bd. III: Strukturtypen der Logik, Berlin - Heidelberg - New York ... 1984.

Stegmüller (1986a)
Stegmüller,W.: Hauptströmungen der Gegenwartsphilosophie - Eine kritische Einführung, Bd. II, 7. Aufl., Stuttgart 1986.

Stegmüller (1986b)
Stegmüller,W.: Hauptströmungen der Gegenwartsphilosophie - Eine kritische Einführung, Bd. III, 7. Aufl., Stuttgart 1986.

Stetter (1988)
Stetter,F.: Grundbegriffe der Theoretischen Informatik, Berlin - Heidelberg - New York ... 1988.

Thiel,C. (1980b)
Thiel,C.: Extensionalitätsaxiom; in: Mittelstraß,J. (Hrsg.): Enzyklopädie Philosophie und Wissenschaftstheorie, Band 1: A-G, Mannheim - Wien - Zürich 1980, S. 626-627.

Thiel,C. (1980c)
Thiel,C.: Über Ursprung und Problemlage des argumentationstheoretischen Aufbaus der Logik; in: Gethmann,C.F. (Hrsg.): Theorie des wissenschaftlichen Argumentierens, Frankfurt 1980, S. 117-135.

Thieler-Mevissen (1975)
Thieler-Mevissen,G.: Vollständigkeit und Korrektheit des netztheoretischen Kalküls für die Aussagenlogik, Interner Bericht 04/75-5-9, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1975.

Thieler-Mevissen (1977)
Thieler-Mevissen,G.: The Petri Net Calculus of Predicate Logic, Interner Bericht ISF-76-09, Institut für Informationssystemforschung, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1977.

Thuy (1989)
Thuy,N.H.C.; Schnupp,P.: Wissensverarbeitung und Expertensysteme, München - Wien 1989.

Tietze,J. (1988)
Tietze,J.: Einführung in die Wirtschaftsmathematik, Braunschweig - Wiesbaden 1988.

Townsend,C. (1987)
Townsend,C.: Mastering Expert Systems with Turbo Prolog, Indianapolis 1987.

Trost (1984)
Trost,H.: Wissensrepräsentation in der AI am Beispiel Semantischer Netze; in: Retti,J.; Bibel, W.; Buchberger,B.; Buchberger,E.; Horn,W.; Kobsa,A.; Steinacker,I.; Trappl,R.; Trost,H.: Artificial Intelligence - Eine Einführung, Stuttgart 1984, S. 47-72.

Tsichritzis (1971)
Tsichritzis,D.: Modular System Description, Technical Report No. 33, Department of Computer Science, University of Toronto, Toronto 1971.

Tsichritzis (1982)
Tsichritzis,D.: Form Management; in: Communications of the ACM, Vol. 25 (1982), S. 453-478.

Turing (1937a)
Turing,A.M.: On Computable Numbers, With an Application to the Entscheidungsproblem; in: o.V.: Proceedings of the London Mathematical Society, Second Series, Vol. 42, London 1937, S. 230-265.

Turing (1937b)
Turing,A.M.: On Computable Numbers, With an Application to the Entscheidungsproblem. A Correction; in: o.V. Proceedings of the London Mathematical Society, Second Series, Vol. 43, London 1937, S. 544-546.

Ursprung (1982)
Ursprung,H.W.: Die elementare Katastrophentheorie: Eine Darstellung aus der Sicht der Ökonomie, Lecture Notes in Economics and Mathematical Systems 195, Berlin - Heidelberg - New York 1982.

Vagin (1988)
Vagin,V.N.; Zakharov,V.N.; Rozenblyum,L.Y.: Logical Inference on Interpreted Petri Nets; in: Soviet Journal of Computer and Systems Sciences, Vol. 26 (1988), No. 3, S. 98-105.

Valette (1979b)
Valette,R.; Diaz,M.: A Methodology for Easily Provable Implementation of Synchronization Mechanisms; in: Syre,H.J. (Hrsg.): 1st European Conference on Parallel and Distributed Processing, 14.-16.02.1979 in Toulouse, Toulouse 1979, S. 156-162.

Van Emde Boas (1978)
Van Emde Boas,P.: The Connection between Model logic and Algorithmic logics; in: Winkowski,J. (Hrsg.): Mathematical Foundations of Computer Science 1978, Proceedings of the 7th Symposium, 4.-8.09.1978 in Zakopane, Lecture Notes in Computer Science 64, Berlin - Heidelberg - New York 1978, S. 1-15.

van Emden (1977)
van Emden,M.H.: Programming with Resolution Logic; in: Elcock,E.W.; Michie,D. (Hrsg.): Machine Intelligence 8: Machine Representations of Knowledge, New York - Chichester - Brisbane ... 1977, S. 266-299.

Varney (1988)
Varney,L.R.; Swiderski,D.: Real-Time Knowledge-Based Programming with IF/tasklog: An Executive for Asynchronous Multitasking in IF/Prolog, IF/PROmLOG, and C, Paper, Interface Computer GmbH, München o.J. (1988).

Varsek (1986)
Varsek,I.: Übersetzung logischer Programmiersprachen; in: Hommel,G.; Schindler,S. (Hrsg.): GI - 16. Jahrestagung I: Informatik-Anwendungen - Trends und Perspektiven, 06.-10.10.1986 in Berlin, Proceedings, Informatik-Fachberichte 126, Berlin - Heidelberg - New York ... 1986, S. 212-226.

Vaught (1973)
Vaught,R.L.: Some Aspects of the Theory of Models; in: The American Mathematical Monthly, Vol. 80 (1973), Papers in the Foundations of Mathematics, published as a supplement to the American Mathematical Monthly, S. 3-38.

Vautherin (1987b)
Vautherin,J.: Parallel Systems Specifications with Coloured Petri Nets and Algebraic Specifications.; in: Rozenberg,G. (Hrsg.): Advances in Petri Nets 1987, Lecture Notes in Computer Science 266, Berlin - Heidelberg - New York ... 1987, S. 293-308. (Überarbeitete Fassung von Vautherin (1986a).)

Vidal-Naquet (1982b)
Vidal-Naquet,G.; Roucairaol,G.; Berthelot,G.; Memmi,G.; Sifakis,J.; Girault,C.: Extensions et Abrevations des Reseaux de Petri, D.E.A. de Systemes Informatiques, Institut de Programmation, Universite Pierre et Marie Curie Paris, Paris 1982.

von Kutschera (1980)
von Kutschera,F.: Grundbegriffe der Handlungslogik; in: Lenk,H. (Hrsg.): Handlungstheorien interdisziplinär I - Handlungslogik, formale und sprachwissenschaftliche Handlungstheorien, München 1980, S. 67-106.

von Luck (1989)
v. Luck,K.; Meyer,R.; Pirlein,T.: Die logische Rekonstruktion eines Gegenstandbereiches - Eine Fallstudie -; in: Retti,J.; Leidlmair,K. (Hrsg.): 5. Österreichische Artificial-Intelligence-Tagung, 28.-31.03.1989 in Igls, Proceedings, Informatik-Fachberichte 208, Berlin - Heidelberg - New York ... 1989, S. 278-287.

von Martial (1988a)
v. Martial,F.; Victor,F.: An Interactive Planner for Open Systems; in: o.V.: The Fourth Conference on Artificial Intelligence Applications, Proceedings, 14.-18.03.1988 in San Diego, Silver Spring 1988, S. 293-298.

von Neumann/Burks (1966)
Burks,A.W. (Hrsg. und vervollständigender Autor): Theory of Self-Reproducing Automata - John von Neumann, Urbana - London 1966.

von Weizsäcker (1985)
von Weizsäcker,C.F.: Aufbau der Physik, München - Wien 1985.

von Wright (1966)
von Wright,G.H.: The Logic of Action - A Sketch; in: Rescher,N. (Hrsg.): The Logic of Decision and Action, Pittsburgh 1966, S. 121-136.

von Wright (1980)
von Wright,G.H.: Elemente der Handlungslogik; in: Lenk,H. (Hrsg.): Handlungstheorien interdisziplinär I - Handlungslogik, formale und sprachwissenschaftliche Handlungstheorien, München 1980, S. 21-34.

von Wright (1986)
von Wright,G.H.: Truth, Negation, and Contradiction; in: Synthese, Vol. 66 (1986), S. 3-14.

von Zimmermann (1990)
von Zimmermann,P.: Einsatz von objektorientierter Softwaretechnologie im Rechnungswesen; in: Scheer,A.-W. (Hrsg.): Rechnungswesen und EDV, 11. Saarbrücker Arbeitstagung 1990, Heidelberg 1990, S. 235-264.

Voß,A. (1989b)
Voß,A.: A layered algebraic specification technique for expert systems; in: Metzing,D. (Hrsg.): GWAI-89, 13th German Workshop on Artificial Intelligence, 18.-22.09.1989 in Eringerfeld, Proceedings, Informatik-Fachberichte 216, Berlin - Heidelberg - New York ... 1989, S. 368-378.

Wahlster (1977)
Wahlster,W.: Die Repräsentation von vagem Wissen in natürlichsprachlichen Systemen der Künstlichen Intelligenz , Bericht Nr. 38 (IFI-HH-B-38/77), Institut für Informatik, Universität Hamburg, Hamburg 1977.

Wahlster (1981)
Wahlster,W.: Natürlichsprachliche Argumentation in Dialogsystemen - KI-Verfahren zur Rekonstruktion und Erklärung approximativer Inferenzprozesse, Informatik-Fachberichte 48, Berlin - Heidelberg - New York 1981.

Wahlster (1982)
Wahlster,W.: Natürlichsprachliche Systeme - Eine Einführung in die sprachorientierte KI-Forschung; in: Bibel,W.; Siekmann,J.H. (Hrsg.): Künstliche Intelligenz, Frühjahrsschule Teisendorf, 15.-24.03.1982 in Teisendorf, Informatik-Fachberichte 59, Berlin - Heidelberg - New York 1982, S. 203-283.

Walker,A. (1983)
Walker,A.: Prolog/Ex1, An Inference Engine Which Explains Both Yes and No Answers; in: Bundy,A. (Hrsg.): IJCAI-83 - Proceedings of the Eigth International Conference on Artificial Intelligence, 08.-12.08.1983 in Karlsruhe, Vol. 2, o.O. (Los Altos) 1983, S. 526-528.

Walker,A. (1984)
Walker,A.: Databases, Expert Systems, and PROLOG; in: Reitman,W. Hrsg.): Artificial Intelligence Applications for Business, Proceedings of the NYU Symposium, 18.-20.05.1983 in New York, Norwood 1984, S. 87-109.

Walther,C. (1983)
Walther,C.: A Many-Sorted Calculus based on Resolution and Paramodulation; in: Bundy,A. (Hrsg.): IJCAI-83, Proceedings of the Eigth International Joint Conference on Artificial Intelligence, 08.-12.08.1983 in Karlsruhe, o.O. (Los Altos) 1983, Vol. 2, S. 882-891.

Walther,C. (1984)
Walther,C.: Ein mehrsortiger Resolutionskalkül mit Paramodulation, Interner Bericht 23/84, Institut für Informatik I, Universität Karlsruhe, Karlsruhe 1984.

Walther,C. (1985c)
Walther,C.: A Classification of Unification Problems in Many-Sorted Theories, Interner Bericht 10/85, Institut für Informatik I, Universität Karlsruhe, Karlsruhe 1985.

Walther,C. (1987)
Walther,C.: A Many-Sorted Calculus Based on Resolution and Paramodulation, London - Los Altos 1987.

Walther,C. (1989)
Walther,C.: Many-Sorted Resolution; in: Christaller,T. (Hrsg.): Künstliche Intelligenz, 5. Frühjahrsschule, KIFS-87, 28.03.-05.04.1987 in Günne, Proceedings, Informatik-Fachberichte 202, Berlin - Heidelberg - New York ... 1989, S. 65-102.

Wand (1989)
Wand,Y.: A Proposal for a Formal Model of Objects; in: Kim,W.; Lachovsky,F.H. (Hrsg.): Object-Oriented Concepts, Databases, and Applications, New York - Reading - Menlo Park ... 1989, S. 537-559.

Wandschneider (1974)
Wandschneider,D.: Zum Antinomienproblem der Logik; in: Ratio,. Bd. 16 (1974), S. 74-91.

Wang,Ha. (1952)
Wang,Ha.: Logic of Many-Sorted Theories; in: The Journal of Symbolic Logic, Vol. 17 (1952), S. 105-116.

Wang,Ha. (1957)
Wang,Ha.: A Variant to Turing's Theory of Computing Machines; in: Journal of the Association for Computing Machinery (ACM), Vol. 4 (1957), S. 63-92.

Wang,Ha. (1960a)
Wang,Ha.: Proving Theorems by Pattern Recognition I; in: Communications of the ACM, Vol. 3 (1960/Reprint 1971), S. 220-234.

Wang,Ha. (1965)
Wang,Ha.: Russel und seine Logik; in: Ratio, 7. Bd. (1965), S. 24-54.

Waterman (1970)
Waterman,D.A.: Generalization Learning Techniques for Automating the Learning of Heuristics; in: Artificial Intelligence, Vol. 1 (1970), S. 121-170.

Webber (1983)
Webber,B.L.: Logic and Natural Language; in: Computer, Vol. 16 (1983), S. 43-46.

Weck (1982)
Weck,M.: Werkzeugmaschinen, Bd. 3: Automatisierung und Steuerungstechnik, 2. Aufl., Düsseldorf 1982.

Weck (1991f)
Weck,M.; Pauls,A.: Datenbanken in Fertigungsleitsystemen; in: Pritschow,G.; Spur,G.; Weck,M. (Hrsg.): Leit- und Steuerungstechniken in flexiblen Produktionsanlagen, München - Wien 1991, S. 131-160.

Wedekind (1981)
Wedekind,H.: Datenbanksysteme I - Eine konstruktive Einführung in die Datenverarbeitung in Wirtschaft und Verwaltung, 2. Aufl., Mannheim - Wien - Zürich 1981.

Wedekind (1989c)
Wedekind,H.: Eine logische Analyse des Verhältnisses von Anwendungs- und Datenbanksystemen; in: Härder,T. (Hrsg.): Datenbanksysteme in Büro, Technik und Wissenschaft, GI/SI-Fachtagung, 01.-03.03.1989 in Zürich, Proceedings, Informatik-Fachberichte 204, Berlin - Heidelberg - New York ... 1989, S. 19-42.

Weidemann (1988)
Weidemann,H.: Überlegungen zu einer temporalen Modalanalyse; in: Menne,A.; Öffenberger,N. (Hrsg.): Modallogik und Mehrwertigkeit, Hildesheim - Zürich - Nerw York 1988, S. 86-104.

Weijland (1988)
Weijland,W.P.: Semantics for Logic Programs without Occur Check; in: Lepistö,T.; Salomaa,A. (Hrsg.): Automata, Languages and Programming, 15th International Colloquium, 11.-15.07.1988 in Tampere, Lecture Notes in Computer Science 317, Berlin - Heidelberg - New York ... 1988, S. 710-726.

Weizenbaum (1982)
Weizenbaum,J.: Die Macht der Computer und die Ohnmacht der Vernunft, 3. Aufl., Frankfurt 1982.

Westphal (1986)
Westphal,H.: Eine Beurteilung paralleler Modell für Prolog; in: Hommel,G.; Schindler,S. (Hrsg.): GI - 16. Jahrestagung I: Informatik-Anwendungen - Trends und Perspektiven, 06.-10.10.1986 in Berlin, Proceedings, Informatik-Fachberichte 126, Berlin - Heidelberg - New York ... 1986, S. 227-240.

Whitehead (1925)
Whitehead,A.N.; Russell,B.: Principia Mathematica, Vol. 1, 2. Aufl., Cambridge (Massachusetts) 1925.

Wildemann (1988c)
Wildemann,H.: Methodenintegration in Modularprogrammen zur Realisierung von CIM und JIT; in: Mertens,P.; Wiendahl,H.-P.; Wildemann,H. (Hrsg.): CIM-Komponenten zur Planung und Steuerung - Expertensysteme in der Produktion, München 1988, S. 39-96.

Wilkins (1984)
Wilkins,D.E.: Domain-independent Planning: Representation and Plan Generation; in: Artificial Intelliegnce, Vol. 22 (1984), S. 269-301.

Winograd (1984)
Winograd,T.: Software für Sprachverarbeitung; in: Spektrum der Wissenschaft, o.Jg. (1984), Heft 11, S. 88-102.

Winter,Ro. (1991)
Winter,Ro.: Mehrstufige Produktionsplanung in Abstraktionshierarchien auf der Basis relationaler Informationsstrukturen, Dissertation, Universität Frankfurt 1989, Berlin - Heidelberg - New York ... 1991.

Wittgenstein (1921)
Wittgenstein,L.: Logisch-Philosophische Abhandlung.; in: Annalen der Naturphilosophie, Bd. 14. (1921), S. 185-262.

Wohlrapp (1979)
Wohlrapp,H.: Analytischer versus konstruktiver Wissenschaftsbegriff; in: Lorenz,Ku. (Hrsg.): Konstruktionen versus Positionen - Beiträge zur Diskussion um die Konstruktive Wissenschaftstheorie, Bd. II: Allgemeine Wissenschaftstheorie, Berlin - New York 1979, S. 348-377.

Wollnik (1986)
Wollnik,M.: Implementierung computergestützter Informationssysteme - Perspektive und Politik informationstechnologischer Gestaltung, Dissertation, Universität Köln, Berlin - New York 1986.

Yamaguchi,T. (1985)
Yamaguchi,T.; Tezuka,Y.; Kakusho,O.: Parallel Processing of Resolution; in: in: o.V.: Proceedings of the Ninth International Joint Conference on Artificial Intelligence, IJCAI-85, 19.-23.09.1985 in Los Angeles, Vol. 2, Los Altos 1985, S. 1178-1180.

Yasuura (1984)
Yasuura,H.: On Parallel Computational Complexity of Unification; in: o.V.: Fifth Generation Computer Systems 1984, Proceedings of the International Conference, 06.-09.11.1984 in Tokyo, Tokyo - Amsterdam 1984, S. 235-243.

Yates (1970)
Yates,R.A.; Raphael,B.; Hart,T.P.: Resolution Graphs; in: Artificial Intelligence, Vol. 1 (1970), S. 257-289.

Zelewski (1986a)
Zelewski,S.: Das Leistungspotential der Künstlichen Intelligenz - Eine informationstechnisch-betriebswirtschaftliche Analyse, Band 1, 2 und 3, Dissertation (unter dem Titel: Das Leistungspotential der Künstlichen Intelligenz - Bestandsaufnahme und Bewertungsansätze aus informationstechnisch-betriebswirtschaftlicher Perspektive unter besonderer Berücksichtigung produktionswirtschaftlicher Aspekte -), Universität Köln 1985, Witterschlick (Bonn) 1986.

Zelewski (1986c)
Zelewski,S.: Netztheoretische Fundierung von parallelen Algorithmen für die Lösung linear-ganzzahliger OR-Modelle, Arbeitsbericht Nr. 12 (2. Aufl. des Arbeitsberichts 9/1986), Seminar für Allgemeine Betriebswirtschaftslehre, Industriebetriebslehre und Produktionswirtschaft, Universität Köln, Köln 1986.

Zelewski (1988b)
Zelewski,S.: Netztheoretische Fundierung von parallelen Algorithmen zur Bestimmung von Lösungen für linear-ganzzahlige OR-Modelle ohne Extremalziele; in: Angewandte Informatik, 30. Jg. (1988), S. 352-362.

Zelewski (1989a)
Zelewski,S.: Komplexitätstheorie - als Instrument zur Klassifizierung und Beurteilung von Problemen des Operations Research, Braunschweig - Wiesbaden 1989.

Zelewski (1989c)
Zelewski,S.: Petrinetze für die Konstruktion und Konsistenzanalyse von logisch orientierten Problembeschreibungen, Arbeitsbericht Nr. 28, Seminar für Allgemeine Betriebswirtschaftslehre, Industriebetriebslehre und Produktionswirtschaft, Universität Köln, Köln 1989.

Zelewski (1989e)
Zelewski,S.: Contributions of Net-Theory to the Modelling of OR-Problems from a Logically Based Point of View; in: Rivista di matematica per le scienze economiche e sociali, Anno 12 (1989), Fascicolo 2, S. 67-92.

Zelewski (1991b)
Zelewski,S.: Kritische Faktoren beim Einsatz von Expertensystemen; in: Zeitschrift für Betriebswirtschaft, 61. Jg. (1991), S. 237-258.

Zell,A. (1989)
Zell,A.: Zeitbeschränkte Logische Programmierung; in: Retti,J.; Leidlmair,K. (Hrsg.): 5. Österreichische Artificial-Intelligence-Tagung, 28.-31.03.1989 in Igls, Proceedings, Informatik-Fachberichte 208, Berlin - Heidelberg - New York ... 1989, S. 96-105.

Zervos (1977)
Zervos,C.R.; Irani,K.B.: Colored Petri Nets: Their Properties and Applications, Disseratation an der University of Michigan, Technical Report No. RADC-TR-77-246, Department of Electrical Engineering, University of Michigan, Ann Arbor 1977.

Zilles (1974)
Zilles,S.N.: Algebraic Specifications of Data Types, Project MAC Progress Report No. 11, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1974, S. 28-52.

Zisman (1978a)
Zisman,M.D.: Use of Production Systems for Modeling Asynchronous, Concurrent Processes; in: Waterman,D.A.; Hayes-Roth,F. (Hrsg.): Pattern-Directed Inference Systems, New York - San Francisco - London 1978, S. 53-68.

Institut für Produktionswirtschaft und Industrielle Informationswirtschaft

der Universität Leipzig

Verzeichnis der Arbeitsberichte

Nr. 1: ZELEWSKI, STEPHAN: Das Konzept technologischer Theorietransformationen - eine Analyse aus produktionswirtschaftlicher Perspektive, Leipzig 1994.

Nr. 2: SIEDENTOPF, JUKKA: Anwendung und Beurteilung heuristischer Verbesserungsverfahren für die Maschinenbelegungsplanung - Ein exemplarischer Vergleich zwischen Neuronalen Netzen, Simulated Annealing und genetischen Algorithmen, Leipzig 1994.

Nr. 3: ZELEWSKI, STEPHAN: Unternehmenskrisen und Konzepte zu ihrer Bewältigung, Leipzig 1994.

Nr. 4: SIEDENTOPF, JUKKA: Ein effizienter Scheduling-Algorithmus auf Basis des Threshold Accepting, Leipzig 1995.

Nr. 5: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 1: Exposition, Leipzig 1995.

Nr. 6: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 2: Bezugsrahmen, Leipzig 1995.

Nr. 7: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 3: Einführung in Stelle/Transition-Netze, Leipzig 1995.

Nr. 8: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 4: Verfeinerungen von Stelle/Transition-Netzen, Leipzig 1995.

Nr. 9: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 5: Einführung in Synthetische Netze, Teilband 5.1: Darstellung des Kernkonzepts, Leipzig 1995.

Nr. 10: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 5: Einführung in Synthetische Netze, Teilband 5.2: Auswertungsmöglichkeiten, Leipzig 1995.

Nr. 11: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 6: Erweiterungen von Synthetischen Netzen, Leipzig 1995.

Nr. 12: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 7: Fallstudie, Leipzig 1995.

Nr. 13: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 8: Charakterisierung des Petrinetz-Konzepts, Leipzig 1995.

Nr. 14: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 9: Beurteilung des Petrinetz-Konzepts, Leipzig 1995.

Nr. 15: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 10: Petrinetz-Literatur, Leipzig 1995.

Nr. 16: SIEDENTOPF, JUKKA: An Efficient Scheduling Algorithm Based upon Threshold Accepting, Leipzig 1995.

Nr. 17: SIEDENTOPF, JUKKA: The Threshold Waving Algorithm for Job Shop Scheduling, Leipzig 1995.

Nr. 18: ZELEWSKI, STEPHAN: Diskussionspapier zum Text "Zur wirtschaftlichen und sozialen Lage in Deutschland" einer evangelisch-katholischen Arbeitsgruppe, Leipzig 1995.

Nr. 19: SCHIMMEL, KATRIN; ZELEWSKI, STEPHAN: Untersuchung alternativer Auktionsformen hinsichtlich ihrer Eignung zur Koordination verteilter Agenten auf Elektronischen Märkten, Leipzig 1996.

Nr. 20: SIEDENTOPF, JUKKA: Feinterminierung unter restriktiven Laufzeitanforderungen - Ein exemplarischer Vergleich lokaler Suchverfahren (Teil I), Leipzig 1996.

Nr. 21: ZELEWSKI, STEPHAN: Strukturalistische Rekonstruktion von ökologisch induzierten Entwicklungen der produktionswirtschaftlichen Theoriebildung, Leipzig 1996.

Nr. 22: RÖßLER, HENRIK; SCHIMMEL, KATRIN: Zur Animation und Simulation hierarchischer Petrinetze., Leipzig 1996.

Nr. 23: RÖßLER, HENRIK; WURCH, MAIK: Implementierung des Modells eines Flexiblen Fertigungssystems, Teilbände 1-3, Leipzig 1996.

Nr. 24: SCHIMMEL, KATRIN: Abstimmung der Implementierungssoftware INCOME/STAR. Bericht zu Phase 1 des Projekts PEMVEK, Leipzig 1996/ 2. Auflage 1997.

Nr. 25: WURCH, MAIK: Modellierung eines Flexiblen Fertigungssystems sowie von Produktionsaufträgen. Bericht zu den Phasen 2 und 3 des Projekts PEMVEK, Leipzig 1996.

Nr. 26: SCHIMMEL, KATRIN: Der Einsatz elektronischer Märkte zur Koordination in Flexiblen Fertigungssystemen, Leipzig 1996.

Nr. 27: TÖPFER, ANDREAS: Vergleichende Wirtschaftlichkeitsbetrachtung von Windkraftanlagen im Raum Halle/Leipzig - Ergebniszusammenfassung, Leipzig 1996.

Nr. 28: WURCH, MAIK: Implementierung von Vickrey-Auktionen mit Hilfe von Petrinetzen, Leipzig 1996.

Nr. 29: WURCH, MAIK: Coordinating Electronic Markets by Auctions, Leipzig 1996.

Nr. 30: SCHIMMEL, KATRIN; WURCH, MAIK: Simulation eines Koordinations-Moduls in einem Flexiblen Fertigungssystem, Leipzig 1996.

Nr. 31: RÖßLER, HENRIK: XPNC - Auswahltool für parallele Schaltentscheidungen bei der Simulation von Petrinetzen, Leipzig 1997.

Nr. 32: ZELEWSKI, STEPHAN: Handelsinformationssysteme - erweiterte Fassung einer Rezension, Leipzig 1997.

Nr. 33: ZELEWSKI, STEPHAN: Erfahrungen mit Höheren Petrinetzen bei der Modellierung von Prozeßkoordinierungen in komplexen Produktionssystemen. Bericht zu Phase 7 des Projekts PEMVEK, Leipzig 1997. [in Arbeit]

Nr. 34: ZELEWSKI, STEPHAN: Optimierung in Petrinetz-Modellen - eine Analyse aus betriebswirtschaftlicher Sicht, Leipzig 1997. [in Arbeit]

Nr. 35: WURCH, MAIK: Simulation von Koordinations-Modulen unter Berücksichtigung strategischen Agentenverhaltens, Leipzig 1997. [in Arbeit]

Nr. 36: SCHIMMEL, KATRIN: Komponente für Erreichbarkeitsanalysen. Bericht zu Phase 6 des Projekts PEMVEK, Leipzig 1997.

Nr. 37: WURCH, MAIK: Modellierung der Prozeßkoordinierung. Bericht zu Phase 4 des Projekts PEMVEK, Leipzig 1997. [in Arbeit]

Nr. 38: BODE, JÜRGEN; FUNG, RICHARD Y.K.: Integrating Cost Considerations in Quality Function Deployment, Leipzig 1997.