

**Arbeitsbericht Nr. 12**

**Netztheoretische Fundierung  
von parallelen Algorithmen für die Lösung  
linear-ganzzahliger OR-Modelle**

von  
**Dr. Stephan Zelewski**

**2. Auflage des Arbeitsberichts 9/1986**

**Köln 1986**

**Alle Rechte vorbehalten.**

Abstract

Modelle aus dem Bereich des Operations Research werden zur Zeit noch vorwiegend mit sequentiellen Lösungsalgorithmen bearbeitet. Die zukünftig erwartete Verbreitung besonders leistungsfähiger Computer mit paralleler Architektur läßt jedoch Ausschau nach entsprechend parallel angelegten Algorithmuskonzepten halten. Einen Ansatz in dieser Richtung stellen netztheoretisch fundierte Lösungsalgorithmen dar. Ein solches Konzept wird auf der Basis der Petrinetz-Theorie näher untersucht. Es gestattet die Abbildung linear-ganzzahliger OR-Modelle auf Stelle/Transition-Netze. Die Erreichbarkeitsanalyse dieser Netze führt zur Lösung der ursprünglich algebraisch formulierten OR-Modelle. Aufgrund der inhärenten Parallelität ("Nebenläufigkeit") von Petrinetzen lassen sich Lösungsalgorithmen mit paralleler Ablaufstruktur konzipieren. Das Gestaltungspotential solcher Netz-Algorithmen wird - unter Berücksichtigung der Einsatzmöglichkeit von Erkenntnissen aus der Künstlichen Intelligenz-Forschung - näher untersucht und an einem Beispiel verdeutlicht.

Inhaltsübersicht

	Seite
1 Einführung in den explorativen Ansatz	1
2 Petrinetze zur Darstellung von linear-ganzzahligen OR-Modellen	5
2.1 Darstellung und Aufbereitung linear-ganzzahliger OR-Modelle	5
2.2 Stelle/Transition-Netze als spezielle Ausprägung von Petrinetzen	9
2.3 Abbildung linear-ganzzahliger OR-Modelle auf Stelle/Transition-Netze	12
3 Erreichbarkeitsanalysen zur Lösung von Netzmodellen	15
3.1 Transformation von Petrinetzen in Erreichbarkeitsgraphen	15
3.2 Konzepte für parallele Algorithmen zur Analyse von Erreichbarkeitsgraphen	16
3.2.1 Allgemeine Grundlagen	16
3.2.1.1 Das Parallelitätspotential von Petrinetzen	16
3.2.1.2 Das blackboard-Konzept	19
3.2.1.3 Verfeinerungen der Steuerungskomponente	21
3.2.1.3.1 Basisstrategien	21
3.2.1.3.2 Wissensbasierte Metastrategien	29
3.2.2 Spezielle Ausformung zweier exemplarischer Algorithmus-Konzepte	33
3.2.2.1 Schaltphasenorientiertes Konzept	33
3.2.2.2 Transitionenorientiertes Konzept	44
4 Anhang: Exemplarische Umsetzung eines OR-Modells in ein Netz-Modell	52
Literaturverzeichnis	57

## 1 Einführung in den explorativen Ansatz

Die strukturelle Einfachheit linearer Denkschemata und die konventionelle von Neumann-Architektur Automatischer Informationsverarbeitungssysteme haben in der Vergangenheit dazu geführt, Algorithmen für die Lösung von betriebswirtschaftlichen Problemen, die mit Hilfe von Modellen des Operations Research formalsprachlich beschrieben sind, in sequentieller Weise zu entwerfen. Diese "natürliche" Ablaufstruktur äußert sich beispielsweise in Programmablaufplänen nach DIN 66001 oder Pseudocodes, in denen die Operationen zur Algorithmusausführung stets zeitlich vollständig geordnet vorliegen (sequentielle Algorithmen).

Erst Entwicklungen im Bereich der Informatik, welche auf die Konstruktion paralleler Automaten-Architekturen abzielen, und Erkenntnisse der Neurobiologie über nichtlineare Prozesse bei der menschlichen Informationsverarbeitung haben Bemühungen des Operations Research stimuliert, parallele Algorithmen zu entwickeln<sup>1)</sup>. Ein Algorithmus gilt als parallel (i.w.S.), wenn zumindest ein Teil seiner Operationen zeitlich nicht geordnet ist. Solche "nebenläufigen" (concurrent) Operationen können zeitgleich (parallel i.e.S.), zeitlich überlappend, zeitlich ineinander verschachtelt oder auch in jeder beliebigen zeitlichen Reihenfolge ausgeführt werden.

---

1) Vgl. z.B. Pan (1986), S.130ff., im Hinblick auf parallele Algorithmen für die Lösung von linear-rationalzahligen Modellen ("linearen Programmen"). Einen breiten Überblick über die Forschungsarbeiten auf dem Gebiet der parallelen Algorithmen bietet die annotierte Bibliographie von Kindervater (1985), S. 113ff.

Ziel der Entwicklung solcher paralleler Algorithmen ist es, die Effizienz der Lösung von Problemen dadurch zu erhöhen, daß die Zeitdauern zur Ermittlung der zugehörigen Modelllösungen<sup>2)</sup> reduziert werden<sup>3)</sup>. Voraussetzung hierfür ist, daß Automatische Informationsverarbeitungssysteme mit paralleler Multiprozessor-Architektur bereitstehen, die es ermöglichen, nebenläufige Operationen parallel abzuarbeiten. Hierdurch wird - günstigstenfalls<sup>4)</sup> - die Zeitdauer zur Ausführung einer Operationen-Menge von der Summe aller Operationsdauern - wie bei der sequentiellen Vorgehensweise - auf die Zeitdauer der Operation mit maximaler Ausführungszeit reduziert. Hierbei wird unterstellt, daß jeder nebenläufigen Operation ein Prozessor für ihre Abarbeitung zugeordnet werden kann.

Eine besondere Schwierigkeit bei der Entwicklung solcher paralleler Algorithmen stellt der Sachverhalt dar, daß die modellhafte Problembeschreibung deklarativ, nonprozeduraler Art ist. Infolgedessen determiniert die Sachstruktur einer Klasse von OR-Modellen noch keine bestimmte Ablaufstruktur ihrer Lösungsalgo-

- 
- 2) Der Einfachheit halber wird fortan auf die Differenzierung zwischen der Lösung der ursprünglich gegebenen Probleme und die Lösung der problemabbildenden Modelle verzichtet. Von möglichen Validitätsmängeln der Modelllösungen wird abstrahiert, ohne die Bedeutung dieses Aspekts für die Bewältigung konkreter Probleme zu verkennen.
  - 3) Der Effizienzbegriff wird fortan auf seine zeitliche Dimension reduziert. Andere Effizienzfacetten bleiben unberücksichtigt, obwohl sie bei einer umfassenderen betriebswirtschaftlichen Effizienz-(Wirtschaftlichkeits-)Analyse berücksichtigt werden müssen. Hierzu zählen insbesondere der erhöhte Ressourcenanspruch paralleler Algorithmen, da sie das Vorhalten hoher Verarbeitungskapazität in der Gestalt von Multiprozessor-Architekturen erfordern. Ferner werden Nutzenaspekte der Algorithmanwendung - z.B. hinsichtlich der Lösungsgüte - nicht näher untersucht.
  - 4) Vom erheblichen Zeitbedarf für die Koordinierung von nebenläufigen Operationen, der zur Zeit ein wesentliches, noch nicht befriedigend gelöstes Problem bei der Entwicklung paralleler Automaten-Architekturen und Algorithmen darstellt, wird abgesehen. Gleiches gilt für die Verlängerung von Ausführungsdauern aufgrund des Umstandes, daß nebenläufige Operationen nicht notwendig zeitgleich (parallel i.e.S.), sondern auch "nur" überlappend oder verschachtelt, u.U. - etwa bei mangelnder Prozessorkapazität - sogar auch sequentiell ausgeführt werden können.

rithmen. Die Beschreibung der Komponenten, die ein Problem konstituieren und durch die Elemente eines OR-Modells abgebildet werden, implizieren keine eindeutige zeitliche Anordnung der Operationen, die zur algorithmischen Modellösung ausgeführt werden müssen.

Infolgedessen können parallele Algorithmen nicht unmittelbar aus einer - vermeintlich objektiven - inhärenten Modellparallelität abgeleitet werden. Vielmehr stellt die zeitliche Strukturierung des Zusammenhangs der Operationen, die im Rahmen eines modellösenden Algorithmus ausgeführt werden müssen, eine schöpferische Zugabe der Algorithmenkonstruktion dar. Die implizite Modellparallelität kann nur im Sinne eines Grenzbegriffs als dasjenige - vorläufige - Ausmaß<sup>5)</sup> an Parallelität verstanden werden, das durch bereits bekannte, nicht notwendig implementierte Lösungsalgorithmen explizit verwirklicht wird. Daher besitzt die Entwicklung paralleler Algorithmen explorativen Charakter. Durch sie wird die - nur virtuell vorhandene - Modellparallelität erforscht, im strengen Sinne sogar erst "erschaffen".

In diesem Kontext wird der Beitrag der Petrinetz-Theorie erörtert, ein Fundament für die Entwicklung paralleler Lösungsalgorithmen zu bilden, die sich auf die Klasse der linear-ganzzahligen OR-Modelle ohne Extremalziele erstrecken. Der besondere Reiz dieser netztheoretischen Exploration liegt darin begründet, daß sie die flexible Wahl unterschiedlicher Parallelitätsgrade, in einem bedingten Sinne sogar die Konstruktion "maximal" paralleler Algorithmen ermöglicht.

Die Eigenart dieses Ansatzes wird im wesentlichen durch die Transformation eines ursprünglich gegebenen OR-Modells, das in der üblichen Form eines linear-ganzzahligen Programms vorliegt, in ein Netz-Modell geprägt. Im Gegensatz zur strikten Nonprozeduralität des OR-Modells impliziert das Netz-Modell bereits eine be-

---

5) Auf die Schwierigkeit, einen Maßstab "objektiver" Parallelität von Algorithmen oder Modellen zu definieren, sei nur kurz hingewiesen. Hier wird das Ausmaß der Parallelität in intuitiver Weise durch den Grad des Verzichts auf die zeitliche Ordnung von Operations-Ausführungen umschrieben.

stimmte, obgleich nur sehr gering ausgeprägte zeitliche Strukturierung später konstruierter Lösungsalgorithmen. Sofern diese transformationsbedingte Strukturvorgabe als gegeben hingenommen wird, kann mit Hilfe der Petri-netz-Theorie das maximal mögliche Ausmaß an Parallelität für alle - netztheoretisch fundierten - Lösungsalgorithmen abgeleitet werden.

Die nachfolgenden Ausführungen besitzen - über die vorangehenden Ausführungen hinaus - auch insofern einen nur explorativen, d.h. nicht konstruktiven Charakter, als dem Verf. weder seitens der Hardware noch seitens software-unterstützter Implementierungssprachen die Voraussetzungen zur Entwicklung konkreter paralleler Algorithmen zur Verfügung stehen. Daher kann lediglich ein netztheoretisches Fundament vorgestellt werden, das eine konzeptionelle Basis zukünftiger Entwürfe für parallele Algorithmen zur Lösung linear-ganzzahliger OR-Modelle darstellt.

Aus dem gleichen Grund bleiben Effizienzbetrachtungen auf das abstrakte Argument der Vorteilhaftigkeit paralleler Operationsausführung beschränkt. Ob netztheoretisch fundierte Algorithmen tatsächlich effizienter sind als konventionelle sequentielle oder - auf anderer Basis konzipierte - parallele Algorithmen, bleibt der komplexitätstheoretischen Analyse konkreter Algorithmenentwürfe und der Laufzeitmessung ihrer Implementierungen vorbehalten. Hierbei muß auch die Ressourcenbeanspruchung durch die Transformation von OR-Modellen in Netz-Modelle als effizienz mindernde Einflußgröße in Rechnung gestellt werden.

Unter der Voraussetzung der Hard- und Software, die derzeit breit zur Verfügung steht, kann sogar davon ausgegangen werden, daß sich netztheoretisch fundierte Algorithmen als recht ineffizient erweisen. Zukünftig erwartete, u.a. vom japanischen Projekt der Entwicklung von Computern der 5. Generation stimulierte Konstruktionen von hochgradig parallelen Automaten-Architekturen lassen jedoch darauf hoffen, daß solche Algorithmen das Effizienzpotential ihrer parallelen Ablaufstruktur wirksamer entfalten können.

## 2 Petrinetze zur Darstellung von linear-ganzzahligen OR-Modellen

### 2.1 Darstellung und Aufbereitung linear-ganzzahliger OR-Modelle

Betrachtet seien linear-ganzzahlige OR-Modelle für (Entscheidungs-)Probleme ohne Extremalziele<sup>6)</sup>. Sie lassen sich als m-zeilige Ungleichungssysteme<sup>7)</sup> in Standardform (US-SF) darstellen als:

$$\underline{A} \cdot \underline{x} \leq \underline{b}$$

6) Die Linearitätsprämisse wird mit den meisten anderen praktisch handhabbaren OR-Modellen geteilt, obwohl es sich oftmals um eine bedeutsame Vereinfachung der zugrundeliegenden Realprobleme handelt. Die Ganzzahligkeit spielt keine wesentliche Rolle, weil alle rationalen Zahlen durch Multiplikation mit einer entsprechend groß gewählten Konstante in ganze Zahlen transformiert werden können und sich die meisten betriebswirtschaftlichen Probleme mit rationalen Zahlen modellieren lassen. Der Ausschluß von Extremalzielen ist dagegen erheblich. Denn hierdurch werden Optimierungsmodelle für Entscheidungs-Probleme, welche die Minimierung oder Maximierung von Zielfunktionen erfordern, ausgegrenzt. Allerdings können reale betriebliche Probleme noch insofern berücksichtigt werden, als sich die involvierten Zielvorstellungen durch Satisfizierungsziele beschreiben lassen, die nur das Erreichen eines Zufriedenheitsniveaus erfordern. Solche Satisfizierungsziele können - ebenso wie die sachlichen Nebenbedingungen des Entscheidungsfelds - als Ungleichungen dargestellt werden. Fortan werden Satisfizierungsziele und sachliche Nebenbedingungen infolge ihrer formal gleichartigen Behandlung unter den Oberbegriff der Restriktionen subsumiert.

7) Solche Ungleichungssysteme umfassen auch Gleichungen, die als Paare von Ungleichungen mit - bis auf die Indizierung - gleichen Variablen und Koeffizienten, aber entgegengesetzter Ungleichungsrelation formuliert werden.



Hierbei bedeuten - mit  $N$  und  $Z$  als Mengen aller natürlichen (einschließlich der Null) bzw. ganzen Zahlen -  $\underline{x}$  den  $m$ -stelligen Spalten-Vektor<sup>8)</sup> der Entscheidungsvariablen  $x_i \in N$ <sup>9)</sup> ( $i = 1, 2, \dots, m$ ),  $\underline{b}$  den  $n$ -stelligen Spalten-Vektor der Restriktionskonstanten  $b_j \in Z$  ( $j = 1, 2, \dots, n$ ) und  $\underline{A}$  die  $n$ -zeilige/ $m$ -spaltige Matrix der Restriktionskoeffizienten  $a_{i,j} \in Z$ .

Wegen der Kommutativität und Assoziativität der konjunkativen Verknüpfung von Ungleichungen läßt sich das o.a. Ungleichungssystem in zwei Teilsysteme zerlegen, um es auf die Transformation in ein Netz-Modell vorzubereiten:

$$\underline{A} \cdot \underline{x} \leq \underline{b} \quad \Leftrightarrow \quad ( \underline{A}_P \cdot \underline{x} \leq \underline{b}_P \quad \wedge \quad \underline{A}_N \cdot \underline{x} \leq \underline{b}_N )$$

$\underline{b}_P$  und  $\underline{b}_N$  sind die  $r$ -stelligen bzw.  $n-r$ -stelligen Spalten-Vektoren der positiven bzw. nicht-positiven Restriktionskoeffizienten.  $\underline{A}_P$  und  $\underline{A}_N$  bezeichnen die Teilmatrizen der Koeffizienten der jeweils zugehörigen Restriktionen.

Da die algebraische Ausprägung der Petrinetz-Theorie wesentlich auf homogenen Gleichungssystemen mit natürlichzahligen Variablen und ganzzahligen Koeffizienten basiert, bedarf das inhomogene zweiteilige Ungleichungssystem der weiterführenden Äquivalenzumformung

---

8) Zeilen-Vektoren werden durch das Superskript "T" für die transponierte Form der zugehörigen Spalten-Vektoren dargestellt.

9) Ganzzahlige Entscheidungsvariablen lassen sich in Differenzen jeweils zweier natürlichzahliger Entscheidungsvariablen überführen. Vgl. hierzu die entsprechende Substitution bei der dritten Äquivalenzumformung in Fußnote 10).

mung<sup>10)</sup>. Es resultiert ein (gemischtes) Ungleichungssystem (US), das sich aus einem zweiteiligen linear-homogenen Gleichungssystem (GS) und einem System von Ungleichungen für die Schlupfvariablen<sup>11)</sup> als Restriktionssystem (RS) zusammensetzt:

$$\begin{aligned} \underline{AP} \cdot \underline{x} - \underline{xp} + \underline{xp}' &= \underline{0} & \wedge & & \underline{xp} - \underline{xp}' &\leq \underline{bp} \\ \wedge & & \underline{AN} \cdot \underline{x} + \underline{xn} &= \underline{0} & \wedge & & \underline{xn} &\geq -\underline{bn} \end{aligned}$$

$\underline{0}$  steht für den - je nach Verwendung r- oder n-r-stelligen - Null(spalten)vektor. Der r-stellige Spaltenvektor  $\underline{xp}$ , der r-stellige Spaltenvektor  $\underline{xp}'$  und der n-r-stellige Spaltenvektor  $\underline{xn}$  umfassen die Schlupfvariablen  $x_i$  mit  $i=m+1, m+2, \dots, m+r$ , die Schlupfvariablen  $x_i$  mit  $i=m+n+1, m+n+2, \dots, m+n+r$  bzw. die Schlupfvariablen  $x_i$  mit  $i=m+r+1, m+r+2, \dots, m+n$ <sup>12)</sup>. Jede nicht-triviale Lösung  $\underline{xh}^T = (\underline{x}^T, \underline{xp}^T, \underline{xp}'^T, \underline{xn}^T) \neq \underline{0}^T$  für das Ungleichungssystem US ist zugleich eine Lösung des ursprüng-

10) Den Umformungen liegen mit den Variablen  $x, x', z \in \mathbb{N}$  und den Konstanten  $a, b \in \mathbb{Z}$  sowie  $d \in \mathbb{N}$  folgende Äquivalenzen zugrunde:

$$\begin{aligned} & a \leq b \wedge b \leq 0 & // z=-b \\ \Leftrightarrow & a \leq -z \wedge z \geq 0 \\ \Leftrightarrow & a + z \leq 0 \wedge z \geq 0 \\ \Leftrightarrow & a + z + d = 0 \wedge z \geq 0 \wedge d \geq 0 \\ \Leftrightarrow & a + (z + d) = 0 \wedge z + d \geq z & // x=z+d \geq 0 \\ \Leftrightarrow & a + x = 0 \wedge x \geq z & // z=-b \\ \Leftrightarrow & a + x = 0 \wedge x \geq -b \geq 0 \\ & a \leq b \wedge b = 0 \\ \Leftrightarrow & a \leq 0 \\ \Leftrightarrow & a + d = 0 \wedge d \geq 0 & // x=d \\ \Leftrightarrow & a + x = 0 \wedge x \geq 0 \\ & a \leq b \wedge b \geq 0 & // z=b \\ \Leftrightarrow & a \leq z \wedge z = b \geq 0 \\ \Leftrightarrow & a - z \leq 0 \wedge z = b \\ \Leftrightarrow & a - z + d = 0 \wedge z = b \wedge d \geq 0 \\ \Leftrightarrow & a - (z - d) = 0 \wedge z - d \leq b & // x-x'=z-d \\ \Leftrightarrow & a - x + x' = 0 \wedge x - x' \leq b \end{aligned}$$

11) Die Schlupfvariablen erfüllen den Zweck, die n Ungleichungen in Gleichungen zu überführen.

12) Die Einführung des Vektors  $\underline{xp}'$  ist erforderlich, um für Restriktionen mit positiven Restriktionskonstanten zuzulassen, daß die Summe der Produkte aus Restriktionskoeffizienten und Entscheidungsvariablen (die "linke Restriktionsseite") negative Werte annehmen kann, ohne daß der Bezug des Schlupfvariablen-Vektors  $\underline{xp}$  auf den Konstanten-Vektor  $\underline{bp}$  aufgegeben werden muß. Dieser Bezug ist für die spätere Erörterung von Lösungsalgorithmen vorteilhaft, weil er für die Erkenntnis von unzulässigen Lösungen herangezogen werden kann.

lichen linear-ganzzahligen OR-Modells US-SF<sup>13</sup>). Die Umkehrung gilt, sofern  $\underline{x} = \underline{0}$  keine zulässige Lösung dieses OR-Modells darstellt. Dies wird fortan vereinfachend unterstellt, um die Lösungsäquivalenz der Ungleichungssysteme US und US-SF zu gewährleisten<sup>14</sup>).

Diese kompakte, zur Ausgangsform äquivalente Darstellungsweise linear-ganzzahliger OR-Modelle (ohne Extremalziele<sup>15</sup>) läßt sich in bezug auf eine beliebige, j-te Zeile des Ungleichungssystems übersichtlicher wiedergeben durch:

$$\sum_{i=1}^m a_{i.j} \cdot x_i \leq b_j \quad \Leftrightarrow$$

$$\text{für } b_j > 0 : \quad \sum_{i=1}^m a_{i.j} \cdot x_i - x_{m+j} + x_{m+n+j} = 0 \quad \wedge \quad x_{m+j} + x_{m+n+j} \leq b_j$$

$$\text{für } b_j = 0 : \quad \sum_{i=1}^m a_{i.j} \cdot x_i + x_{m+j} = 0 \quad \wedge \quad x_{m+j} \geq 0$$

$$\text{für } b_j < 0 : \quad \sum_{i=1}^m a_{i.j} \cdot x_i + x_{m+j} = 0 \quad \wedge \quad x_{m+j} \geq -b_j$$

- 
- 13) Triviale Lösungen  $(\underline{x}^T, \underline{x}_p^T, \underline{x}_p'^T, \underline{x}_n^T) = \underline{0}^T$  des Ungleichungssystems US, die zugleich triviale Lösungen des hierin enthaltenen linear-homogenen Gleichungssystems GS sind, werden nicht berücksichtigt, da sie notwendig immer gelten, ohne zulässige Lösungen für das ursprüngliche Ungleichungssystem US-SF des linear-ganzzahligen OR-Modells darstellen zu müssen. Daher ist stets zu prüfen, ob - neben der Lösungsmenge, die für das Ungleichungssystem US ermittelt wird, - auch der Nullvektor  $\underline{x} = \underline{0}$  zur Lösungsmenge des Ungleichungssystems US-SF gehört.
- 14) Es wird implizit unterstellt, daß die obligate Prüfung der Zulässigkeit des Nullvektors  $\underline{x}$  als Lösung von Ungleichungssystem US-SF stets vorgenommen wird.
- 15) Auf diesen Zusatz wird fortan der Einfachheit halber verzichtet.

## 2.2 Stelle/Transition-Netze als spezielle Ausprägung von Petrinetzen

Petrinetze, welche die Entwicklung paralleler Lösungsalgorithmen für die voranstehend spezifizierten OR-Modelle fundieren sollen, werden hier in der speziellen Gestalt der Stelle/Transition-Netze<sup>16)</sup> betrachtet. Ein Stelle/Transition-Netz läßt sich algebraisch als ein 6-Tupel  $STN = (S, T, F, K, W, M_0)$  mit nachfolgend beschriebenen Eigenschaften definieren.

$S$  ist eine endliche Menge von Objekten  $s_j$ , die als Stellen bezeichnet werden. Sie zeichnen sich durch die Möglichkeit aus, durch bewegliche Objekte - sogenannte "Marken" - belegt zu werden.

$T$  ist eine endliche Menge von Objekten  $t_i$ , die Transitionen heißen. Sie unterscheiden sich von Stellen grundsätzlich durch ihren dynamischen Charakter. Sie können während eines "Schaltprozesses" Marken von (vorgelagerten) Stellen abziehen und auf (nachgelagerten) Stellen ablegen.

$F \subseteq S \times T \cup T \times S$  ist die Flußrelation aus geordneten 2-Tupeln  $(s_j, t_i)$  oder  $(t_i, s_j)$ , die Stellen und Transitionen miteinander verknüpft. Dies geschieht in der Richtung des Markenflusses, der zwischen den Stellen durch das Schalten der Transitionen verursacht wird.

Es wird  $S \cap T = \emptyset$  und  $S \cup T \neq \emptyset$  vorausgesetzt. Ferner wird unterstellt, daß die Flußrelation Stellen und Transitionen vollständig verknüpft, d.h. es gibt weder isolierte Stellen noch Transitionen. Folglich ergibt die Vereinigung von Definitions- und Wertebereich der Flußrelation die Objektmenge  $S \cup T$ .

Die graphische Repräsentation des Tripels  $(S, T, F)$  ist ein bipartiter, gerichteter Graph ("Netz") mit zwei artverschiedenen Knoten: den Stellen und den Transitionen, die als Kreise bzw. als Quadrate symbolisiert werden<sup>17)</sup>. Die gerichteten Kanten zwischen Stellen und

16) Vgl. Jantzen (1980), S.167ff.; Reisig (1985), S. 62ff.

17) Vgl. hierzu die exemplarische Darstellung in Abb. 1 des Anhangs.

Transitionen werden jeweils durch ein 2-Tupel der Flußrelation (Kantenursprung als erste, Kantenspitze als zweite Tupel-Komponente) definiert.

$K : S \rightarrow N \cup \{u\}$  ordnet jeder Stelle die Kapazität maximal auf ihr zulässiger Marken zu. Fortan wird eine unbeschränkt ("u") hohe Markenzapazität für jede Stelle angenommen, so daß diese Netzgröße nicht mehr explizit behandelt werden muß. Durch die Gewichtsfunktion  $W : F \rightarrow N$  wird jedes 2-Tupel der Flußrelation - also jede Netzkante - mit dem Kantengewicht  $w$  versehen. Es gibt die Anzahl der Marken an, die bei jedem Schalten der zur Kante gehörigen ("adjazenten") Transition in Kantenrichtung bewegt werden.  $M_0 : S \rightarrow N$  schreibt als Ausgangsmarkierung jeder Stelle eine Anzahl von Marken zu, die sich auf ihr im Ausgangszustand des Netzes befinden. Die Markierungsfunktion  $M_0$  wird oftmals in der inhaltlich äquivalenten Form des Markierungsvektors  $\underline{M}_0$  verwendet, der mit  $\underline{M}_0^T = (M_0(s_1), \dots, M_0(s_n))$  in seiner  $j$ -ten Komponente die Markenanzahl  $M_0(s_j)$  der Stelle  $s_j$  unter der Ausgangsmarkierung  $M_0$  ausweist<sup>18)</sup>.

Die statische Netzstruktur, die aus den Knotenmengen  $S$  und  $T$ , der Kantenmenge  $F$  sowie der Gewichtsfunktion  $W$  gebildet wird<sup>19)</sup>, läßt sich durch die Inzidenzmatrix  $\underline{C}$  mit  $n$  Zeilen für die Stellen  $s_j$  ( $j = 1, \dots, n$ ) und  $m$  Spalten für die Transitionen  $t_i$  ( $i = 1, \dots, m$ ) beschreiben, wobei für jeden ihrer Koeffizienten  $c_{i,j}$  gilt:

$$c_{i,j} = \begin{cases} -w(s_j, t_i) & ; \text{ falls } (s_j, t_i) \in F \\ 0 & ; \text{ falls } (s_j, t_i) \notin F \wedge (t_i, s_j) \notin F \\ +w(t_i, s_j) & ; \text{ falls } (t_i, s_j) \in F \end{cases}$$

Die dynamische Netzstruktur wird einerseits durch die Ausgangsmarkierung  $\underline{M}_0$  im Sinne einer Randbedingung festgelegt. Andererseits ist sie implizit durch die Schaltregel der Transitionen definiert. Sie besteht für Stelle/Transition-Netze aus einer Abbildung  $SR$  von Vor-

18) Unter den Markierungsbegriff wird nachfolgend sowohl die Markierungsfunktion als auch der Markierungsvektor subsumiert.

19) Die Kapazitäts-Abbildung wird infolge der o.a. Voraussetzung nicht weiter berücksichtigt, bildet jedoch grundsätzlich auch ein Element der statischen Netzstruktur.

gängermarkierungen  $\underline{M}$  und schaltenden Transitionen  $t_i$  auf Nachfolgermarkierungen  $\underline{M}'$ :

$$\text{SR : } \quad N^n \times T \quad \rightarrow \quad N^n$$

$$(\underline{M}, t_i) \rightarrow \underline{M}' = \underline{M} + \underline{C} \cdot \underline{t}_i \quad ; \quad \text{falls}$$

$$M(s_j, t_i) \geq w(s_j, t_i) \quad \text{für alle } s_j \in V(t_i)$$

Hierbei umfaßt der Vorbereich  $V(t_i)$  der Transition  $t_i$  alle Stellen  $s_j$ , die ihr - gemäß der Flußrelation  $F$  - unmittelbar vorgelagert sind. Entsprechend ist der Nachbereich  $N(s_j)$  der Stelle  $s_j$  als die Menge aller Transitionen  $t_i$  definiert, die der Stelle  $s_j$  unmittelbar nachgelagert sind. Derart unmittelbar vor- und nachgelagerte Stellen und Transitionen werden als inzident bezeichnet.  $\underline{t}_i$  ist der Schalt-Vektor der Transition  $t_i$ . Er besitzt als  $m$ -stelliger Spalten-Vektor nur an der  $i$ -ten Stelle eine 1, sonst besteht er aus den Komponenten 0.

Die zweiteilige Schaltregel bewirkt, daß eine Transition  $t_i$  nur dann schalten kann, wenn sich auf den Stellen  $s_j$  ihres Vorbereichs jeweils mindestens so viele Marken befinden, wie durch das Schalten der Transition über die Kanten  $(s_j, t_i)$  mit den Gewichten  $w(s_j, t_i)$  abgezogen würden (Aktivierungsbedingung). Wenn eine derart aktivierte Transition schaltet, zieht sie von ihren unmittelbar vorgelagerten Stellen so viele Marken ab und belegt ihre unmittelbar nachgelagerten Stellen mit so vielen Marken, wie durch die Gewichte der adjazenten Kanten bestimmt und durch die Koeffizienten der Inzidenzmatrix  $\underline{C}$  wiedergegeben wird. Eine aktivierte Transition kann, muß aber nicht schalten<sup>20)</sup>.

Wenn die Folge  $f = (t_{i(1)}, t_{i(2)}, \dots, t_{i(z)})$  aus  $z$  - nicht notwendig verschiedenen<sup>21)</sup> - Transitionen im aktuellen Netzzustand, der durch die Markierung  $\underline{M}$  beschrieben ist, geschaltet wird, resultiert die Markierung  $\underline{M}'$  nach Schalten der letzten (der  $z$ -ten) Transition. Diese Markierung wird ermittelt, indem die Schaltregel jeweils für das Schalten einer Transition

20) Auf diesen Umstand wird später zurückgekommen; vgl. S. 18 u. 22.

21) In einer solchen Folge kann dieselbe Transition mehrmals - auch unmittelbar hintereinander - geschaltet werden.

$t_i(h)$  mit  $h=2, \dots, z$  aus der Folge  $f$  auf diejenige Markierung angewendet wird, die von ihrer Vorgänger-Transition  $t_i(h-1)$  erzeugt wurde. Wenn der  $m$ -stellige Spalten-Vektor  $\underline{t}_f$  in seiner  $i$ -ten Komponente angibt, wie oft Transition  $t_i$  in der Folge  $f$  geschaltet wird, gilt:

$$\underline{M}' = \underline{M} + \underline{C} \cdot \underline{t}_f$$

### 2.3 Abbildung linear-ganzzahliger OR-Modelle auf Stelle/Transition-Netze

Von der eingangs dargestellten und äquivalent transformierten Standardform linear-ganzzahliger OR-Modelle (US-SF) wird nur das abschließend abgeleitete linear-homogene Gleichungssystem GS unmittelbar auf ein Stelle/Transition-Netz abgebildet. Das ergänzende Restriktionensystem RS für die involvierten Schlupfvariablen wird erst bei den netztheoretisch fundierten Lösungsalgorithmen berücksichtigt.

Das Gleichungssystem GS läßt sich explizit darstellen als:

$$\begin{aligned} a_{1.1}x_1 + \dots + a_{m.1}x_m - x_{m+1} + 0 \dots 0 + x_{m+n+1} + 0 \dots 0 &= 0 \\ &\dots \\ a_{1.r}x_1 + \dots + a_{m.r}x_m + 0 \dots 0 - x_{m+r} + 0 \dots 0 + x_{m+n+r} &= 0 \\ a_{1.r+1}x_1 + \dots + a_{m.r+1}x_m + 0 \dots 0 + x_{m+r+1} + 0 \dots 0 &= 0 \\ &\dots \\ a_{1.n}x_1 + \dots + a_{m.n}x_m + 0 \dots 0 + x_{m+n} + 0 \dots 0 &= 0 \end{aligned}$$

In vektorisierter Form lautet dieses Gleichungssystem:

$$\underline{AH} \cdot \underline{xh} = \underline{0}$$

mit  $\underline{AH}$  als Erweiterung der Koeffizientenmatrix  $\underline{A}$  um die Koeffizienten  $-1$ ,  $0$  oder  $+1$  in den Spalten  $m+1, \dots, m+n+r$  und mit  $\underline{xh}$  als Erweiterung des Vektors  $\underline{x}$  der Entscheidungsvariablen um die Schlupfvariablen aus den Vektoren  $\underline{xp}$ ,  $\underline{xp}'$  und  $\underline{xn}$ . Ferner wird  $\underline{xh} \neq \underline{0}$  vorausgesetzt, um die Analyse auf nicht-triviale Lösungen einzuschränken.

Dieses algebraische OR-Modell wird in ein Stelle/Transition-Netz  $STN=(S,T,F,W,M_0)$  als äquivalentes Netz-Modell überführt, indem jede Gleichungs-Zeile als Stelle  $s_j$  und jede Spalte, die jeweils durch die Variable  $x_i$  gekennzeichnet ist, als Transition dargestellt wird. Es resultiert ein Netz mit  $m$  Stellen und  $q = m+n+r$  Transitionen. Im einzelnen gilt:

- Der Transition  $t_i$  mit  $i=1,2,\dots,m$  ist die Stelle  $s_j$  genau dann unmittelbar vorgelagert, wenn im Gleichungssystem  $a_{i,j} < 0$  gilt. Im Netz werden diese beiden Knoten durch die Kante  $(s_j, t_i)$  mit dem Gewicht  $w(s_j, t_i) = -a_{i,j}$  verknüpft.
- Der Transition  $t_i$  mit  $i=1,2,\dots,m$  ist die Stelle  $s_j$  genau dann unmittelbar nachgelagert, wenn im Gleichungssystem  $a_{i,j} > 0$  gilt. Im Netz werden diese beiden Knoten durch die Kante  $(t_i, s_j)$  mit dem Gewicht  $w(t_i, s_j) = a_{i,j}$  verknüpft.
- Zwischen der Transition  $t_i$  mit  $i=1,2,\dots,m$  und der Stelle  $s_j$  verläuft genau dann keine Kante, wenn im Gleichungssystem  $a_{i,j} = 0$  gilt.
- Der Transition  $t_i$  mit  $i=m+1, m+2, \dots, m+r$  ist die Stelle  $s_j$  genau dann unmittelbar vorgelagert, wenn im Gleichungssystem die Variable  $x_i$  in der  $j$ -ten Zeile (entsprechend  $a_{i,j} = -1$ ) angeführt ist. Im Netz werden diese beiden Knoten durch die Kante  $(s_j, t_i)$  mit dem Gewicht  $w(s_j, t_i) = 1$  verknüpft.
- Der Transition  $t_i$  mit  $i=m+n+1, m+n+2, \dots, m+n+r$  ist die Stelle  $s_j$  genau dann unmittelbar nachgelagert, wenn im Gleichungssystem die Variable  $x_i$  in der  $j$ -ten Zeile (entsprechend  $a_{i,j} = 1$ ) enthalten ist. Im Netz werden diese beiden Knoten durch die Kante  $(t_i, s_j)$  mit dem Gewicht  $w(t_i, s_j) = 1$  verbunden.
- Der Transition  $t_i$  mit  $i=m+r+1, m+r+2, \dots, m+n$  ist die Stelle  $s_j$  genau dann unmittelbar nachgelagert, wenn im Gleichungssystem die Variable  $x_i$  in der  $j$ -ten Zeile (entsprechend  $a_{i,j} = 1$ ) angeführt ist. Im Netz werden diese beiden Knoten durch die Kante  $(t_i, s_j)$  mit dem Gewicht  $w(t_i, s_j) = 1$  verknüpft.
- Als Ausgangsmarkierung wird  $\underline{M}_0 = \underline{0}$  gewählt. Es wird also unterstellt, daß anfangs keine Stelle von Marken belegt ist.



Per constructionem stimmt die Inzidenzmatrix  $\underline{C}$  des Netzes STN mit der Koeffizientenmatrix  $\underline{AH}$  des homogenen Gleichungssystems GS überein. Aus diesem Sachverhalt kann eine fundamentale Korrespondenz zwischen den Lösungen des homogenen Gleichungssystems und Schaltfolgen des Netzes hergeleitet werden.

Gesucht ist die Menge NM aller Null-Schaltfolgen  $f$  des Netzes STN, welche - angewendet auf die Ausgangsmarkierung  $\underline{M}_0=0$  - diese Ausgangsmarkierung durch  $\underline{M}'=0$  reproduzieren. Mit der o.a. Schaltgleichung für Folgen von Transitionen und  $\underline{t}_f$  als Vektor der zugehörigen Schaltanzahlen muß gelten:

$$\begin{aligned} \underline{M}' &= \underline{M}_0 + \underline{C} \cdot \underline{t}_f = 0 + \underline{C} \cdot \underline{t}_f = \underline{C} \cdot \underline{t}_f \quad \wedge \quad \underline{M}'=0 \\ &\Leftrightarrow \quad \underline{C} \cdot \underline{t}_f = 0 \end{aligned}$$

Zugleich erfüllt jede Lösung  $\underline{x}_h$  des homogenen Gleichungssystems per definitionem die Bedingung:

$$\underline{AH} \cdot \underline{x}_h = 0$$

Aus  $\underline{C} = \underline{AH}$  folgt unmittelbar, daß jede Null-Schaltfolge  $f$  im Netz STN durch den Vektor  $\underline{t}_f$  ihrer Schaltanzahlen eine mit  $\underline{t}_f$  identische Lösung  $\underline{x}_h$  des homogenen Gleichungssystems GS definiert. Die Lösungsmenge LM des letzten ist also durch die Menge NM aller Null-Schaltfolgen des Netzes STN vollständig determiniert.

Eine Standard-Methode der Petrinetz-Theorie, solche Null-Schaltfolgen zu ermitteln, ist die Invarianten-Analyse. Denn jede Null-Schaltfolge stellt eine Netzinvariante dar, die dadurch gekennzeichnet ist, daß die Netzmarkierung von der Ausführung dieser Schaltfolge insgesamt nicht verändert wird. Da die Invarianten-Analyse auf algebraische Methoden zur Lösung linear-homogener Gleichungssysteme zurückgreift<sup>22)</sup>, wird dieser Weg hier jedoch nicht eingeschlagen. Denn diese Methoden könnten ebenso gut auf das o.a. homogenen Gleichungssystem GS angewendet werden, die aus der Transformation der ursprünglich gegebenen linear-ganzzahligen OR-Modelle resultieren. Ein Umweg über die Petrinetz-Theorie wäre überflüssig.

22) Vgl. z.B. Memmi (1981), insbesondere S. 18ff.; Pascoletti (1985), S. 119ff.

### 3 Erreichbarkeitsanalysen zur Lösung von Netzmodellen

#### 3.1 Transformation von Petrinetzen in Erreichbarkeitsgraphen

Eine Alternative zur Invarianten-Analyse stellt die Petrinetz-Theorie durch das Instrument der Erreichbarkeitsgraphen zur Verfügung, die das dynamische Verhalten eines Petrinetzes vollständig beschreiben. Es handelt sich um konventionelle (nonpartite) gerichtete Graphen mit Netzmarkierungen  $\underline{M}$  als Knoten. In ihnen ist genau dann eine Kante vom Knoten  $\underline{M}$  zum Knoten  $\underline{M}'$  gerichtet und mit dem Namen der Transition  $t_i$  beschriftet, wenn im zugrundeliegenden Petrinetz (Stelle/Transition-Netz) die Markierung  $\underline{M}$  durch das Schalten der Transition  $t_i$  in die Markierung  $\underline{M}'$  überführt wird.

Fortan wird das oben eingeführte Stelle/Transition-Netz  $STN = (S, T, F, W, M_0)$  unterstellt. Dann ist der Ursprung des Erreichbarkeitsgraphen EG durch die Ausgangsmarkierung  $\underline{M}_0$  festgelegt. Da für die (nicht-triviale) Lösung des homogenen Gleichungssystems GS alle Null-Schaltfolgen im Netz STN gesucht werden, sind im Erreichbarkeitsgraphen EG alle endlichen Wege aus gerichteten, zusammenhängenden Folgen alternierender Markierungen (Knoten) und Transitionen (Kanten) zu ermitteln, die von der Nullmarkierung  $\underline{0}$  als Ausgangsmarkierung  $\underline{M}_0$  (Startknoten) ausgehen und zu eben dieser Nullmarkierung als Ergebnis  $\underline{M}'$  des Schaltens aller betroffenen Transitionen (Zielknoten) zurückführen. Es werden also Zyklen im Erreichbarkeitsgraphen gesucht, die als Start-Ziel-Wege durch die Ausgangsmarkierung (Nullmarkierung)  $\underline{M}_0 = \underline{0}$  führen.

Für die Feststellung solcher Start-Ziel-Wege muß der Erreichbarkeitsgraph konstruiert werden. Da entsprechende Algorithmen im wesentlichen nur aus der systematischen Prüfung der Aktivierungsbedingungen und aus der Anwendung der Schaltregel auf aktivierte Transitionen bestehen, wird hier nicht näher auf sie eingegangen,

sondern auf die einschlägige Literatur verwiesen<sup>23)</sup>. Näher ausgeführt wird hingegen, durch welche Konzepte diese Konstruktion eines Erreichbarkeitsgraphen und die Ermittlung seiner Start-Ziel-Wege herangezogen werden kann, um parallele Algorithmen für die Lösung der ursprünglich gegebenen OR-Modelle zu gestalten.

### 3.2 Konzepte für parallele Algorithmen zur Analyse von Erreichbarkeitsgraphen

#### 3.2.1 Allgemeine Grundlagen

##### 3.2.1.1 Das Parallelitätspotential von Petrinetzen

Erreichbarkeitsgraphen können grundsätzlich mit Algorithmen der konventionellen Graphentheorie dahingehend untersucht werden, ob in ihnen mindestens ein Start-Ziel-Weg von der Nullmarkierung weg und wieder zu dieser zurück existiert. Die vollständige Abbildung der dynamischen Struktur eines Netz-Modells durch seinen Erreichbarkeitsgraphen bedeutet wegen der ebenso vollständigen Abbildung des zugrundeliegenden OR-Modells durch das Netz-Modell, daß durch die Menge aller Start-Ziel-Wege des Erreichbarkeitsgraphen die Menge aller Lösungen des OR-Modells gegeben ist. Netztheoretisch fundierte Algorithmen (Netzalgorithmen) zur Ermittlung von Start-Ziel-Wegen in Erreichbarkeitsgraphen erlauben daher immer die Ermittlung mehrerer, bei Bedarf auch aller Lösungen der zugrundeliegenden OR-Modelle.

Neben dem vorherrschenden Typ sequentieller Algorithmen wurden für solche Wegermittlungen in Graphen auch schon mehrere parallele Algorithmen entwickelt<sup>24)</sup>. Diese Ansätze nehmen jedoch keinen spezifischen Bezug auf die inhärente Parallelität der hier betrachteten Petrinetze.

<sup>23)</sup> Vgl. z.B. Jantzen (1980), S. 171; Reisig (1985), S. 67, der auf S. 67ff. wesentliche Eigenschaften solcher Graphen näher analysiert. Die dort beschriebene Konstruktionsweise von Erreichbarkeitsgraphen ist in die Algorithmus-Konzepte eingeflossen, die auf S. 36ff. u. 45ff. vorgestellt werden.

<sup>24)</sup> Vgl. Imai (1979), S. 416ff.; Nakagawa (1979), S. 628ff.; Kanal (1981), S. 570f.

Die Parallelität von Petrinetzen basiert auf der lokalen Definition der Schaltregel für die Transitionen. Sie läßt es zu, unter einer gegebenen Netzmarkierung - dem jeweils aktuellen Knoten des Erreichbarkeitsgraphen - Aktivierung und ggf. auch Schaltwirkung einer jeden Transition unabhängig<sup>25)</sup> von allen anderen Netzelementen zu analysieren. Folglich besteht die Möglichkeit, in jedem Knoten den Erreichbarkeitsgraphen erstens alle Transitionen hinsichtlich ihrer Aktivierung parallel zu untersuchen und zweitens innerhalb der Restmenge aktivierter Transitionen auch noch deren Schaltwirkungen als resultierende Nachfolger-Markierungen (Folgeknoten) parallel zu berechnen.

Da bei dieser Vorgehensweise mehrere neue Knoten im Erreichbarkeitsgraphen erzeugt werden können, besteht ein drittes Parallelitätspotential darin, nicht - wie oben noch vorausgesetzt - jeweils nur einen aktuellen Knoten zu analysieren. Vielmehr lassen sich alle Knoten, die hinsichtlich Aktivierung und Schaltwirkungen ihrer Transitionen noch nicht vollständig untersucht wurden, hinsichtlich dieser beiden Aspekte parallel betrachten.

Ein Algorithmus, der die drei vorgenannten Parallelitätspotentiale vollständig ausschöpft, verwirklicht die maximale inhärente Parallelität des zugrundeliegenden Petrinetzes. Ein solcher Algorithmus müßte mit der Konstruktion des Erreichbarkeitsgraphen verschränkt sein, weil die Konstruktion der Kanten des Erreichbar-

---

25) Diese Unabhängigkeit gilt nur hinsichtlich der Konstruktion des - hier im Mittelpunkt stehenden - Erreichbarkeitsgraphen. Im zugrundeliegenden Petrinetz können Transitionen, die unter derselben Markierung aktiviert sind, sowohl voneinander unabhängig schalten (nebenläufig aktivierte Transitionen) als auch durch ihr eigenes Schalten die Aktivierungsbedingungen anderer Transitionen aufheben (konfliktnär aktivierte Transitionen). Die letztgenannte Abhängigkeit von Transitionen spielt für Erreichbarkeitsgraphen jedoch keine Rolle, solange in ihnen beim Übergang zwischen zwei Markierungen jeweils nur eine Transition geschaltet wird, da grundsätzlich alle Schaltmöglichkeiten - auch die konfliktnären - enumeriert werden. Von dieser Voraussetzung wird fortan ausgegangen - bis auf die später erörterte Ausnahme der Verdichtung von Erreichbarkeitsgraphen.

keitsgraphen einerseits mit der Analyse der Aktivierungsbedingungen und Schaltwirkungen der Transitionen des Petrinetzes andererseits inhaltlich übereinstimmt. Folglich kann ein Algorithmus, der die inhärente Netz-Parallelität ausschöpfen soll, nicht erst auf den bereits vorliegenden Erreichbarkeitsgraphen aufsetzen, sondern muß mit dessen Konstruktion parallel einhergehen. Hierin liegt der vierte und letzte Parallelisierungsansatz der nachfolgenden Überlegungen.

Es liegt in der Hand des Konstrukteurs von Netzalgorithmen, in welchem Umfang die Parallelitätspotentiale der Petrinetz-Theorie ausgeschöpft werden. Beispielsweise ist es möglich, Aktivierung und Schaltwirkungen von Transitionen im Erreichbarkeitsgraphen parallel zu untersuchen, soweit es sich um die Betrachtung desselben Knotens (derselben Markierung) handelt, während unterschiedliche Knoten sequentiell abgearbeitet werden. Ein weitergehender Verzicht auf potentielle Parallelität besteht darin, nicht alle Transitionen, die infolge ihrer Aktivierung geschaltet werden können, tatsächlich zu schalten, sondern stets nur eine von ihnen auszuwählen. In diesem Freiheitsgrad variabler Parallelisierung liegt ein bemerkenswerter Gestaltungsspielraum von netztheoretisch fundierten Algorithmen.

Werden die Prüfung der Aktivierungsbedingung und die Ermittlung der Schaltwirkung einer Transition jeweils als eine elementare Operation von parallelen Algorithmen zur simultanen Konstruktion und Auswertung des Erreichbarkeitsgraphen von Petrinetzen aufgefaßt, so bewirkt dies eine - für die Petrinetz-Theorie spezifische - zeitliche Vorstrukturierung solcher Algorithmen. Sie war im originären, zu lösenden homogenen Gleichungssystem GS noch nicht enthalten.

Erstens wird das Gleichungssystem GS spaltenweise in Transitionen eines Petrinetzes zerlegt, welche die atomaren Elemente der dynamischen Netzstruktur darstellen. Somit bilden sie auch die Basis der Ablaufstruktur von hierauf aufbauenden Konstruktions- und Auswertungsalgorithmen. Zweitens erlangen die Schaltwirkungen einer Transition nicht zu jedem beliebigen Zeitpunkt Rele-

vanz, sondern nur dann, wenn die Transition unter der jeweils aktuellen Netzmarkierung aktiviert ist. Somit bestimmt die dynamische Netzstruktur auch die zeitlichen Beziehungen zwischen den Schalt-Operationen der Transitionen. Drittens beeinflußt die zeitliche Struktur dieser Operations-Ausführungen die zeitliche Struktur der Erzeugung neuer Netzmarkierungen, also auch der korrespondierenden neuen Knoten im Erreichbarkeitsgraphen. Folglich wird die Ablaufstruktur der Knoten-Analyse im Erreichbarkeitsgraphen durch die dynamische Struktur des zugrundeliegenden Petrinetzes determiniert<sup>26)</sup>.

### 3.2.1.2 Das blackboard-Konzept

Um die das oben skizzierte Parallelitätspotential von Petrinetzen nutzen zu können, muß eine Grundstruktur definiert werden, auf der die elementaren Operationen der Analyse von Aktivierungsbedingungen und Schaltwirkungen ausgeführt werden können. Eine - nach Einschätzung des Verf. - vielversprechende Möglichkeit bietet hier das das blackboard-Konzept aus der Künstlichen Intelligenz-Forschung. Es wurde speziell zur Realisierung paralleler Algorithmen (ursprünglich im Zusammenhang mit der Erkennung akustischer Sprachmuster) eingeführt. Es stehen bereits subtile Konzepte für die Implementierung solcher Algorithmen und ihrer Ablaufsteuerung zur Verfügung<sup>27)</sup>.

Trotz seiner im Detail ausgefeilten Implementierungen erweist sich das blackboard-Konzept strukturell als sehr einfach. Auf einer deklarativen Informationsbasis - dem "blackboard" - operieren mehrere, hinsichtlich ihrer Anzahl grundsätzlich nicht beschränkte prozedura-

26) Allerdings handelt es sich nur um eine partielle Determination, da - je nach gewählter Knoten-Erzeugungsstrategie - unterschiedliche, jeweils mit der dynamischen Netzstruktur kompatible zeitliche Abfolgen der Knotenbildung im Erreichbarkeitsgraphen realisiert werden können.

27) Vgl. Erman (1975), S. 483ff.; Hayes-Roth (1979), S. 285ff.; Hayes-Roth (1985), S. 260ff.

le Einheiten, die mitunter auch als "Dämonen" bezeichnet werden. Ein solcher Dämon vermag jeweils eine spezifische Operationen-Sequenz auf der Informationsbasis auszuführen, wenn in dieser Basis ein ebenso spezifisches Informationsmuster als Ausführungsbedingung vorliegt. Ist dieses der Fall, kann der Dämon seine Operationen autonom ausführen ("Informations-getriggerte" Variante) oder erst nach Aufruf durch eine übergeordnete blackboard-Steuerung - die Kontrollkomponente - tätig werden ("Kontroll-getriggerte" Variante). Das Ergebnis der Operationsausführung stellt eine Veränderung des Informationsinhalts der Informationsbasis dar.

Die Informationsbasis wird hier für parallele Netzalgorithmen zweidimensional strukturiert. Auf der ersten Ebene wird das Petrinetz als Repräsentation (Netz-Modell) des zu lösenden linear-homogenen Gleichungssystems GS implementiert. Hinzu kommt das Restriktionensystem RS, welches das Gleichungssystem ergänzt, um die Lösungsäquivalenz hinsichtlich des ursprünglichen linear-ganzzahligen OR-Modells US(-SF) zu wahren. Auf der zweiten Ebene wird durch die Aktivität der Dämonen der zugehörige Erreichbarkeitsgraph simultan konstruiert und ausgewertet.

Die Dämonen inkorporieren - in einer noch näher zu bestimmenden Weise - die Operationen der Analyse von Aktivierungsbedingungen und der Ausführung von Schaltwirkungen der Transitionen. Die übergeordnete Steuerungskomponente des blackboards sorgt für die ursprüngliche Initialisierung des Petrinetzes mit der Ausgangsmarkierung und entsprechend des Erreichbarkeitsgraphen mit dem Startknoten für diese Nullmarkierung.

Die Steuerungskomponente gewährleistet auch die Einhaltung des Restriktionensystems RS. Zu diesem Zweck werden die bereits erzeugten Wege im Erreichbarkeitsgraphen laufend überwacht<sup>28)</sup>, ob die auf ihnen geschalteten Transitionen  $t_i$  noch im Bereich zulässiger Schaltanzahlen liegen. Ist dies nicht der Fall und kann

---

28) Dies kann durch Dämonen sui generis erfolgen, die sich - im Gegensatz zu den petrinetzbezogenen Dämonen - nur auf den Erreichbarkeitsgraphen erstrecken.

auch mit Sicherheit nicht mehr in diesen Bereich zurückgekehrt werden, wird der betroffene Weg in seinem bis dahin erreichten Endknoten als unfruchtbar markiert und nicht mehr weiterverfolgt<sup>29)</sup>. Bei dieser Betrachtung läßt sich auf die grundlegende Monotonie-Eigenschaft von Petrinetzen zurückgreifen, daß bei fortgesetztem Erzeugen von Nachfolger-Markierungen die Schaltanzahl jeder Transition nur streng monoton steigen oder konstant bleiben kann.

### 3.2.1.3 Verfeinerungen der Steuerungskomponente

#### 3.2.1.3.1 Basisstrategien

Betrachtet sei eine beliebige Transition  $t_i \in T$ , der im homogenen Gleichungssystem GS die Spalte mit der Variablen  $x_i$  (mit  $i = 1, \dots, q$ ) entspricht. Der Wert dieser Variablen ergibt sich aus der Schaltanzahl der Transition im Erreichbarkeitsgraphen auf dem jeweils betrachteten, am Startknoten der Ausgangsmarkierung beginnenden Weg. Die expliziten Restriktionen  $x_i > 0$  (falls  $i = m+r+1, \dots, m+n$  und  $b_{i-m} = 0$  oder falls  $i = m+1, \dots, m+r, m+n+1, \dots, q$ ) und die durch  $x_i \in \mathbb{N}$  nur implizit gegebenen Restriktionen  $x_i \geq 0$  (für  $i = 1, \dots, m$ ) werden immer erfüllt, da im Startknoten für alle Transitionen  $x_i = 0$  (mit  $i = 1, \dots, q$ ) gilt und die Schaltanzahl der o.a. Monotonie unterliegt. Die Restriktionen  $x_i \geq -b_{i-m}$  (falls  $i = m+r+1, \dots, m+n$  und  $b_{i-m} < 0$ ) werden im Startknoten wegen  $x_i = 0$  immer verletzt, können aber wegen der Monotonie der Schaltanzahlen mit zunehmender Entwicklung des Erreichbarkeitsgraphen noch erfüllt werden.

---

29) Stattdessen kann auch der Endknoten aus der Agenda der noch zu untersuchenden Knoten des Erreichbarkeitsgraphen gestrichen werden, wenn alle zu ihm führenden, beim Knoten der Nullmarkierung beginnenden Wege als unfruchtbar nachgewiesen werden. Von dieser Variante gehen die unten vorgestellten Algorithmus-Konzepte aus.



Erst wenn ein Start-Ziel-Weg konstruiert wurde, der zur Nullmarkierung zurückgeführt hat, lassen sich durch die Steuerungskomponente die Restriktionen  $x_i - x_{i+n} \leq b_{i-m}$  (falls  $i = m+1, \dots, m+r$ ) und die Restriktionen  $x_i \geq -b_{i-m}$  (falls  $i = m+r+1, \dots, m+n$  und  $b_{i-m} < 0$ ) überprüfen. Werden sie alle erfüllt, ist eine Lösung  $\underline{x}$  des ursprünglichen linear-ganzzahligen OR-Modells US(-SF) gefunden. Denn der Variablen-Vektor  $\underline{xh}^T = (\underline{x}^T, \underline{xp}^T, \underline{xp}'^T, \underline{xn}^T)$ , der mit der Komponente  $\underline{x}$  die gesuchte Modellösung umfaßt, erfüllt sowohl das abgeleitete homogene Gleichungssystem GS als auch das Restriktionensystem RS zur Sicherung der Lösungsäquivalenz des Netz-Modells. Liegt dagegen die Verletzung von mindestens einer dieser Restriktionen vor, dann repräsentiert der Start-Ziel-Weg im Erreichbarkeitsgraphen keine zulässige Lösung des OR-Modells.

Über die vorgenannten Basisstrategien hinaus umfaßt die Steuerungskomponente (mindestens) eine weitere (Basis-)Strategie, die der Erzeugung von Knoten des Erreichbarkeitsgraphen dient. Als Extreme lassen sich anführen:

- Aus der Menge der Transitionen, die im jeweils aktuellen Knoten aktiviert vorliegen, wird stets nur eine geschaltet und der hieraus resultierende Knoten als neuer aktueller Knoten angesetzt. Es handelt sich um die Strategie der Tiefensuche, da versucht wird, einen tiefen, möglichst schnell zur Nullmarkierung zurückführenden Weg zu konstruieren. Diese Steuerung ist tendenziell vorteilhaft, wenn das rasche Auffinden von nur einer Lösung angestrebt wird.
- Alle aktivierten Transitionen werden im jeweils aktuellen Knoten ausgeführt, so daß in der Regel eine Vielzahl neuer Knoten resultiert. Es liegt eine Strategie der Breitensuche vor, wenn zuerst alle Folgeknoten eines aktuellen Knotens erzeugt und danach - sukzessiv oder parallel - als neue aktuelle Knoten abgearbeitet werden. Der Erreichbarkeitsgraph

wird sofort in seiner vollen Breite konstruiert und ausgewertet. Hierdurch läßt sich zwar tendenziell nicht so schnell wie bei der Tiefensuche eine Lösung in der Gestalt eines Start-Ziel-Wegs auffinden, dafür aber das Spektrum dieser Wege, d.h. die gesamte Lösungsmenge systematisch untersucht.

Innerhalb der Breitensuche kann noch differenziert werden, ob die Steuerungskomponente jeweils nur einen der erzeugten Knoten als neuen aktuellen Knoten festlegt, eine bestimmte Anzahl neuer aktueller Knoten (maximal) selektiert oder grundsätzlich alle erzeugten Knoten als aktuell - und somit als Analyseobjekte der nächsten Algorithmusstufe - auswählt. Diese drei Substrategien bedeuten - gemäß ihrer Reihenfolge - ein zunehmendes Ausmaß an Parallelität des blackboard-Algorithmus.

Bei der Tiefensuche werden zur Algorithmus-Implementierung - abgesehen von den zusätzlichen Prozessoren für die Steuerungskomponente - so viele Prozessoren benötigt, wie transitionsbezogene Dämonen implementiert werden sollen. Bei der Breitenstrategie gilt dies nur für den Fall, daß stets nur genau ein aktueller Knoten im Erreichbarkeitsgraphen untersucht wird. Bei höherer Parallelität durch simultane Analyse mehrerer Knoten müssen von den transitionsbezogen definierten Dämonen jeweils so viele Kopien durch eigenständige Prozessoren implementiert werden, wie aktuelle Knoten maximal parallel abgearbeitet werden sollen.

Die Steuerungskomponente bedarf allerdings noch einer Verfeinerung, da die übliche Reduzierung des Erreichbarkeitsgraphen auf einen endlichen Überdeckungsgraphen<sup>30)</sup> bei der netztheoretisch fundierten Lösung linear-ganzzahliger OR-Modelle unzulässig ist. Denn die Konstruktion solcher Überdeckungsgraphen führt zur Substitution der Markierung von Stellen im Petrinetz, deren Markenanzahlen über alle Schranken wachsen können, durch spezielle Unendlichkeits-Symbole. Hiermit gehen aber Informationen des Erreichbarkeitsgraphen verloren, welche für die Ableitung der o.a. Lösungen erforderlich sind. Weil zur Bewahrung dieser Informationen auf das

---

30) Vgl. die Quellenangaben in Fußnote 23).

Konzept der Überdeckungsgraphen verzichtet wird, nimmt der Erreichbarkeitsgraph unendliche Größe an. Dies wird durch die kritischen Transitionen  $t_k$  mit  $k = m+r+1, \dots, q$  verursacht, die auf ihren unmittelbaren Nachfolgerstellen - wegen derer unbeschränkten Marken-Kapazität - durch unendlich häufiges Schalten unendliche viele Marken abzulegen vermögen.

Dennoch muß die Suche im Erreichbarkeitsgraphen nach Lösungen des OR-Modells auf ein endliches Ausmaß reduziert werden, um die Konstruktion praktisch anwendbarer, endlicher Lösungsalgorithmen zu gewährleisten. Diese Forderung wird durch erfüllt, daß in jedem Knoten des Erreichbarkeitsgraphen zunächst nur diejenigen aktivierten Transitionen des zugrundeliegenden Petri-netzes geschaltet werden, denen im OR-Modell Entscheidungsvariablen  $x_i$  mit  $i = 1, \dots, m$  entsprechen. Falls jedoch keine solche Transition aktiviert ist, werden - aber erst dann - aktivierte Transitionen  $t_h$  geschaltet, welche den Schlupfvariablen  $x_h$  mit  $h = m+1, \dots, q$  zugeordnet sind.

Hierdurch wird das Potential unendlich häufiger Schaltakte der kritischen Transitionen  $t_k$  verringert, aber noch nicht eliminiert. Denn es können im Erreichbarkeitsgraphen immer (Teil-)Folgen von Knoten konstruiert werden, in denen keine der Transitionen  $t_i$  aktiviert ist, das Schalten der kritischen Transitionen  $t_k$  jedoch unendlich viele Folgeknoten hervorbringt. Solche unendlichen Folgen lassen sich aber dadurch ausschließen, daß ein Folgeknoten nur dann erzeugt wird, wenn unter der Markierung dieses Folgeknotens mit Sicherheit wieder eine Transition  $t_i$  aktiviert ist oder wenn es sich um die gesuchte Nullmarkierung handelt (Wesentlichkeitsbedingung)<sup>31)</sup>.

Bei dieser Erzeugung eines Folgeknotens wird durch entsprechendes Schalten der Transitionen  $t_h$  mit  $h = m+1, \dots, q$  zusätzlich darauf geachtet, daß die Aktivierungsbedingung der zugehörigen Transition  $t_i$  ( $i=1, \dots, m$ ) mit der minimal möglichen Anzahl von Marken auf je-

31) Vgl. zur konkreten Realisierung dieser Vorgehensweise die unten exemplarisch angeführten Algorithmen.

der Stelle des zugrundeliegenden Petrinetzes erfüllt wird (Minimalitätsbedingung). Hierdurch läßt sich erreichen, von der Nullmarkierung, die als Zielknoten für jede Lösung des OR-Modells wieder erreicht werden muß, so wenig wie nötig abzuweichen.

Wenn eine der vorgenannten Transitionen  $t_i$  ( $i = \dots, m$ ) im Knoten  $\underline{M}_h$  geschaltet wird, ist es erstens möglich, daß im resultierenden Folgeknoten  $\underline{M}_{h.i}$  mindestens eine Stelle  $s_j$  mit  $j = r+1, \dots, n$  mehr Marken aufweist, als durch irgendeine Transition  $t_k$  aus dem Nachbereich  $N(s_j)$  der Stelle  $s_j$  durch Schalten wieder abgezogen werden kann. Zweitens möge hinzukommen, daß der Ausgangsknoten  $\underline{M}_h$  einen bereits im Erreichbarkeitsgraphen erzeugten Referenzknoten  $\underline{M}_r$ , in dem dieselbe Transition  $t_i$  aktiviert war und geschaltet wurde, echt überdeckt. Dies ist der Fall, wenn im Knoten  $\underline{M}_h$  jede Stelle mindestens so viele Marken besitzt wie im Referenzknoten und mindestens eine Stelle - nämlich die Stelle  $s_j$  - mit mehr Marken belegt ist als im Referenzknoten.

Unter den beiden vorgenannten Voraussetzungen läßt sich vom Folgeknoten  $\underline{M}_{h.i}$  aus die intendierte Nullmarkierung mit Sicherheit nicht mehr erreichen<sup>32)</sup>. Daher wird dieser Knoten als unfruchtbar markiert. Die Konstruktion des Erreichbarkeitsgraphen wird an diesem Punkt nicht weiter fortgesetzt.

---

32) Vom Ausgangsknoten  $\underline{M}_h$  gehen per constructionem genau dieselben Schaltfolgen aus wie von dem Referenzknoten. Alle Schaltfolgen, die vom Referenzknoten aus zum Zielknoten der Nullmarkierung zurückführen, sind auch für den Ausgangsknoten aktiviert. Da der Ausgangsknoten den Referenzknoten echt überdeckt, kann jedoch keine dieser Schaltfolgen vom Ausgangsknoten zum Zielknoten führen. Denn jede dieser Schaltfolgen verursacht an ihrem Ende auf der o.a. Stelle  $s_j$  diejenige positive Markenanzahl, um die der Ausgangsknoten in dieser Stelle den Referenzknoten übertrifft. Alle übrigen Schaltfolgen, die vom Referenzknoten aus nicht zum Zielknoten der Nullmarkierung zurückführen, können den Ausgangsknoten erst recht nicht in den Zielknoten leiten. Denn aufgrund der analogen Argumentation wird am Ende jeder dieser Schaltfolgen, wenn sie auf den Ausgangsknoten angewendet wird, zumindest wieder auf der Stelle  $s_j$  eine positive Markenanzahl erzeugt.

Durch die Minimalitätsbedingung wird einerseits keine tatsächlich existierende Lösung des OR-Modells ausgeschlossen. Andererseits gewährleistet sie in Verbindung mit der Begrenzung der Erreichbarkeitsgraphen-Konstruktion durch unfruchtbare Knoten -, daß nur ein endlicher Ausschnitt des Erreichbarkeitsgraphen für die Lösungsermittlung untersucht werden muß. Die Endlichkeit der Analyse gilt allerdings nur, sofern das zugrundeliegende linear-ganzzahlige OR-Modell endlich viele Lösungen besitzt<sup>33)</sup>.

Die Suche im Erreichbarkeitsgraphen bleibt gagegen (potentiell) unendlich, wenn auch das vorgegebene OR-Modell unendlich viele Lösungen aufweist. Dies ist korrekt, weil dann die Enumeration sämtlicher Lösungen ebenfalls unendlich sein muß. Netztheoretisch fundierte Algorithmen lassen sich unter dieser Voraussetzung endlich gestalten, wenn ein modellfremdes Abbruchskriterium hinzugefügt<sup>34)</sup> oder - wie etwa bei der Tiefenstrategie üblich - von vornherein auf die Enumeration aller Lösungen durch Ermittlung nur der ersten Lösung verzichtet wird.

Der Erreichbarkeitsgraph läßt sich hinsichtlich seiner Breite und Tiefe verdichten, indem nicht nur jeweils eine aktivierte Transition längs einer Kante zwischen zwei Knoten geschaltet wird, sondern eine maximale Menge nebenläufig aktivierter Transitionen. Zwei

---

33) Die endlich vielen Lösungen des OR-Modells determinieren durch die Schaltfolgen, die ihnen im Petri-netz entsprechen und welche die zugehörigen Restriktionen einhalten, einen allseits abgeschlossenen Raum zulässiger Netzmarkierungen im n-dimensionalen Raum der natürlichen Zahlen. Die Minimalitätsbedingung führt dazu, daß von diesen Markierungen bei der Lösungssuche im Erreichbarkeitsgraphen nicht beliebig weit abgewichen werden darf. Der Markierungsraum bleibt also weiterhin allseits abgeschlossen. Folglich ist die Enumeration aller möglichen Netzmarkierungen, die in diesem Raum enthalten sind, endlich. Da durch den Erreichbarkeitsgraphen - infolge der Selektivität der Schaltregel - nur eine Teilmenge des Raums möglicher Markierungen erzeugt wird, ist dieser Graph (und damit auch die Suche in ihm) erst recht endlich.

34) Hierbei kann es sich z.B. um die Vorgabe einer maximalen Rechenzeit, einer maximalen Anzahl erzeugter (untersuchter) Knoten oder einer maximalen Anzahl ermittelter Lösungen handeln.

Transitionen sind nebenläufig aktiviert, wenn sie unabhängig voneinander ausgeführt werden können, d.h. wenn die Schaltwirkung der einen Transition die Erfüllung der Aktivierungsbedingung der anderen Transition nicht aufhebt und umgekehrt. Eine maximale Menge nebenläufig aktivierter Transitionen liegt vor, wenn alle Transitionen paarweise nebenläufig aktiviert sind und diese Transitionen-Menge keine echte Teilmenge einer anderen Menge nebenläufig aktivierter Transitionen ist.

Die Strategie der Knotenbildung im Erreichbarkeitsgraphen wird als verdichtet bezeichnet, wenn unter einer Markierung stets maximale Mengen nebenläufig aktivierter Transitionen geschaltet werden und jedem solchen Schalt-Schritt eine Kante im Erreichbarkeitsgraphen zugeordnet wird. Ein Schalt-Schritt, der aus  $k$  Transitionen besteht, ersetzt  $k!$  einzelne Schalt-Aktivitäten (Kanten) in einem äquivalenten nicht-verdichteten Erreichbarkeitsgraphen. Denn die Wirkung dieses Schalt-Schritts führt - wie sich leicht aus der Kommutativität und der Assoziativität der additiven Berechnung von schaltbedingten Folgemarkierungen ableiten läßt - zu demselben Ergebnis wie die  $k!$  Permutationen, die im nicht-verdichteten Erreichbarkeitsgraphen als  $k$ -elementige sequentielle Schaltfolgen der Transitionen des Schalt-Schritts gebildet werden können. Auf diese Weise werden  $k!$  Teilwege mit jeweils der Länge von  $k$  Kanten im nicht-verdichteten Erreichbarkeitsgraphen zu einer einzigen Kante für den entsprechenden nebenläufigen Schalt-Schritt im verdichteten Erreichbarkeitsgraphen zusammengefaßt. Die nebenläufige Ausführung der  $k$  Transitionen in einem Schalt-Schritt bedeutet eine fünfte Dimension des Parallelitätspotentials von Petri-netzen für netztheoretisch fundierte Algorithmen.

Die verdichtete Strategie weist allerdings den Nachteil auf, daß bei ihrer Anwendung auf eine Markierung kurz vor der Rückkehr zur Nullmarkierung die Möglichkeit besteht, durch Schalten eines Schrittes aus mehreren Transitionen über diesen Zielknoten hinweg zu schalten, das erfolgreiche Auffinden eines Start-Ziel-Weges also zu übersehen<sup>35)</sup>.

---

35) Dieser Effekt kann jedoch leicht korrigiert werden, wenn beim Hinwegschaten ein Nachfolger-Knoten des Startknotens als neuer Knoten erzeugt wird, der im verdichteten Erreichbarkeitsgraphen bereits enthalten ist. Es muß bei jeder Erzeugung eines bereits vorhandenen Knotens durch einen mehrelementigen Schalt-Schritt geprüft werden, ob die sequentielle Ausführung aller nebenläufig aktivierten Transitionen dieses Schrittes in einem nicht-verdichteten Erreichbarkeitsgraphen zu einer alternierenden Folge von Markierungen und Schaltakten einzelner Transitionen führen würde, welche die gesuchte Nullmarkierung als Element enthält. Die Reihenfolge der Transitionen kann hierbei eine beliebige der o.a.  $k!$  Permutationen sein. Allerdings kann leider nicht gewährleistet werden, daß beim Hinwegschaten über die Nullmarkierung immer eine Markierung erreicht wird, die im Erreichbarkeitsgraphen schon als Knoten enthalten ist. Daher wird diese Verdichtungsoperation nur noch am Rande weiterverfolgt.

### 3.2.1.3.2 Wissensbasierte Metastrategien

Metastrategien können in der Absicht herangezogen werden, die Effizienz der Steuerungskomponente des blackboard-Konzepts bei der Ausführung von Basisstrategien für die Lösungssuche zu steigern. Grundgedanke solcher Ansätze ist es, Wissen über die Struktur der zu lösenden linear-ganzzahligen OR-Modelle sowie über die Struktur von Petrinetzen und ihrer Erreichbarkeitsgraphen in die Gestaltung der Steuerungskomponente von Netzalgorithmen einzubringen. Oftmals handelt es sich um heuristisches Wissen, dessen Anwendung die intendierte Effizienzsteigerung nicht gewährleisten, sondern nur tendenziell fördern kann.

In Anlehnung an das Konzept der Produktionsregelsysteme, das seitens der Künstlichen Intelligenz-Forschung für den Zweck der Wissensrepräsentation in breitem Umfang eingesetzt wird, erfolgt hier die Darstellung des Wissens für die Bildung von Metastrategien mit der Hilfe von "WENN ... DANN"-Regeln. Da solche Regeln auch von der spezifischen Struktur der jeweils betrachteten Modell(sub)klasse abhängen können, wird nicht angestrebt, sie für die Klasse der linear-ganzzahligen OR-Modelle in umfassender Weise darzustellen. Vielmehr werden nur einige exemplarische Regeln angeführt, die sich in unterschiedlicher Weise zu wissensbasierten Metastrategien kombinieren und durch weitere, hier nicht vorgestellte Regeln ergänzen lassen.

Da die Ausgangsmarkierung  $\underline{M}_0$  als Nullmarkierung festgelegt ist, liegt im Startknoten unter  $\underline{M}_0$  zumeist keine der Transitionen  $t_i$  mit  $i = 1, \dots, m+r$  aktiviert vor. Denn diese Transitionen ziehen in der Regel von mindestens einer Stelle, die ihnen unmittelbar vorgelagert ist, mindestens eine Marke ab, können also unter



der Markierung  $\underline{M}_0$  nicht aktiviert sein<sup>36)</sup>. Daher empfiehlt es sich für die Tiefensuche, zu Beginn eines Lösungsalgorithmus nicht die Transitionen der Entscheidungsvariablen  $x_i$  mit  $i = 1, \dots, m$ , sondern zunächst die Transitionen  $t_h$  mit  $h = m+r+1, \dots, q$  zu betrachten, die per constructionem immer eine Marke auf ihrer unmittelbar nachgelagerten Stelle ablegen:

WENN Strategie = "Tiefensuche" und  $\underline{M}_h = \underline{M}_0$ , DANN wähle eine Transition  $t_h$  mit  $h = m+r+1, \dots, q$ .

Diese Metastrategie läßt sich auf die Breitensuche nur durch die Abschwächung übertragen, daß auch die Schaltmöglichkeit der Transitionen  $t_h$  mit  $h = 1, \dots, m$  berücksichtigt wird:

WENN Strategie = "Breitensuche" und  $\underline{M}_h = \underline{M}_0$ , DANN betrachte die Transitionen  $t_i$  mit  $i = 1, \dots, m$  und - falls keine von diesen aktiviert ist - die Transitionen  $t_h$  mit  $h = m+r+1, \dots, q$ .

Nur wenn das zusätzliche Wissen über das zu lösende OR-Modell vorliegt, daß im Ungleichungssystem  $US(-SF)$  keine Entscheidungsvariable  $x_i$  existiert, deren Koeffizienten  $a_{i.j}$  für alle  $j = 1, \dots, n$  positiv sind (Nichtpositivitätsbedingung), läßt sich die Metastrategie im Hinblick auf die grundsätzlich relevante Transitionenmenge unabhängig von der basalen Suchstrategie formulieren:

WENN  $\underline{M}_h = \underline{M}_0$  und Nichtpositivitätsbedingung erfüllt, DANN wähle eine Transition  $t_h$  mit  $h = m+r+1, \dots, q$  bei Tiefen- und alle Transitionen  $t_h$  bei Breitensuche.

36) Dies gilt im strengen Sinn nur für die Transitionen  $t_h$  mit  $h = m+1, \dots, m+r$ . Denn den Transitionen  $t_i$  mit  $i = 1, \dots, m$  können - je nach Ausprägung der Koeffizientenmatrix  $\underline{A}$  des ursprünglichen Ungleichungssystems  $US(-SF)$  - in der Spalte der zugehörigen Entscheidungsvariablen  $x_i$  in Sonderfällen ausnahmslos positive Werte zugeordnet sein, so daß diese Transitionen immer - also auch unter der Ausgangsmarkierung  $\underline{M}_0$  - aktiviert sind. Da bei der Breitensuche alle aktivierten Transitionen geschaltet werden, scheidet infolge dieser Aktivierungsmöglichkeit die nachfolgend vorgestellte Metastrategie aus.

Es werden nur nicht-triviale Lösungen des Gleichungssystems GS gesucht, denen Schaltfolgen im zugrundeliegenden Petrinetz entsprechen, in denen mindestens eine der Transitionen  $t_i$  mit  $i = 1, \dots, m$  mindestens einmal schalten muß. Daher liegt es nahe, zu Beginn der Konstruktion des Erreichbarkeitsgraphen Transitionen  $t_h$  mit  $h = m+r+1, \dots, q$  so lange zu schalten, bis die Aktivierungsbedingung von mindestens einer Transition  $t_i$  erfüllt wird.

WENN  $M_h = M_0$ , DANN schalte Transitionen  $t_h$  mit  $h = m+r+1, \dots, q$  so lange, bis die Aktivierungsbedingung von mindestens einer Transition (bei Tiefensuche) oder von allen Transitionen (bei Breitensuche)  $t_i$  mit  $i = 1, \dots, m$  erfüllt ist.

Die Restriktionen  $x_i \geq -b_{i-m}$  für  $i = m+r+1, \dots, m+n$  werden zu Beginn der Konstruktion und Analyse des Erreichbarkeitsgraphen notwendig verletzt. Das Auffinden einer Lösung kann bei Tiefensuche dadurch beschleunigt werden, daß versucht wird, möglichst schnell für die Erfüllung dieser Restriktionen zu sorgen. Ist dies in einem Knoten des Erreichbarkeitsgraphen der Fall, muß es wegen der Monotonie der Schaltanzahlen in Schaltfolgen notwendig auch für alle Folgeknoten gelten. Die Erfüllung dieser Restriktionen wird tendenziell am schnellsten erreicht, wenn zunächst aktivierte Transitionen  $t_i$  mit  $i = m+r+1, \dots, m+n$  geschaltet werden, weil jeder ihrer Schaltakte den Wert der Variablen  $x_i$  um 1 erhöht.

WENN Strategie = "Tiefensuche" und mindestens eine Transition  $t_i$  mit  $i = m+r+1, \dots, m+n$  aktiviert ist, DANN wähle für die Konstruktion eines Folgeknotens eine solche Transition  $t_i$ .

Aus der Minimalitätsbedingung wurde als Basisstrategie abgeleitet, daß die Konstruktion des Erreichbarkeitsgraphen abgebrochen werden kann, sobald sich auf einer Stelle  $s_j$  mit  $j = r+1, \dots, n$  in einem Knoten  $M_h$  mehr Marken befinden, als von einer Transition aus ihrem Nachbereich  $N(s_j)$  von dieser Stelle maximal abgezogen werden. Bei der Tiefensuche kann versucht werden, die Begrenzung des lösungsrelevanten Ausschnittes des Er-

reichbarkeitsgraphen durch einen solchen Knoten  $M_h$  möglichst schnell zu realisieren. Hierzu müssen in Knoten, in denen keine Transitionen  $t_i$  mit  $i = 1, \dots, m$  aktiviert sind, Schaltfolgen initiiert werden, welche die Transitionen  $t_h$  mit  $h = m+r+1, \dots, m+n$  in möglichst großer Anzahl enthalten. Die Effizienz der Tiefensuche kann durch diese Metastrategie gesteigert werden, muß es aber nicht<sup>37)</sup>. Die Strategie läßt sich als Regel ausdrücken durch:

WENN Strategie = "Tiefensuche" und im untersuchten Knoten keine Transitionen  $t_i$  mit  $i = 1, \dots, m$  aktiviert sind, DANN initiiere eine solche Schaltfolge, welche Transitionen  $t_h$  mit  $h = m+r+1, \dots, m+n$  in möglichst großer Anzahl umfaßt.

Es besteht eine gewisse Komplementarität zwischen den Strategien der Breiten- und der Tiefensuche. Die erste besitzt das größere Potential zur Verwirklichung hochgradig paralleler Netzalgorithmen, weil im Erreichbarkeitsgraphen oftmals mehrelementige Mengen noch zu untersuchender Knoten vorliegen, deren Elemente parallel abgearbeitet werden können. Hinzu kommt, daß in jedem solchen Knoten zumeist mehrere Transitionen parallel hinsichtlich Aktivierung und Schaltwirkung analysiert werden können. Bei der Tiefensuche wird dagegen immer nur ein Knoten aus der Menge der noch zu untersuchenden Knoten betrachtet und nur ein Folgeknoten durch Schalten von einer aktivierten Transition erzeugt. Das Parallelitätspotential erschöpft sich hier in der parallelen Prüfung von Aktivierungsbedingungen<sup>38)</sup>.

- 
- 37) Der möglichen Effizienzsteigerung wirkt entgegen, daß zunächst keine Lösung des OR-Modells gesucht, sondern nur der Lösungsraum für die spätere Lösungssuche verkleinert wird. Die Lösungssuche im reduzierten Lösungsraum benötigt zwar weniger Ressourcen als die im ursprünglichen Lösungsraum, aber die Reduzierung des Erreichbarkeitsgraphen verursacht zusätzlichen, den vorgenannten Effekt kompensierenden Aufwand. Es sind statistische Analysen erforderlich, welcher dieser beiden Einflüsse bei welchen - noch konkret zu bildenden - Teilklassen von linear-ganzzahligen OR-Modellen überwiegt.
- 38) Bei verdichteter Strategie wird stattdessen eine maximale Menge nebenläufig aktivierter Transitionen in einem Schritt geschaltet. Dies bedeutet ein weiteres Parallelitätspotential der Tiefensuche.

Dieser Überlegenheit der Breitensuche im Hinblick auf die Parallelisierungsmöglichkeiten steht jedoch der Sachverhalt gegenüber, daß die Tiefensuche bei den Metakzepten ein größere Anzahl von Ansätzen erkennen läßt, die Steuerungskomponente durch Wissensbasierung effizienter zu gestalten. Denn in der Auswahl der einen Transition, die unter der Markierung des jeweils untersuchten Knotens aktiviert ist und geschaltet wird, liegt ein erhebliches Potential, die Richtung der Suche im Erreichbarkeitsgraphen zu beeinflussen. Dies ist bei der Breitensuche nicht der Fall, da hier im jeweils untersuchten Knoten alle aktivierten Transitionen geschaltet werden. Dies gilt zumindest in bezug auf die Transitionen  $t_i$ , welche den Entscheidungsvariablen  $x_i$  mit  $i = 1, \dots, m$  entsprechen.

Es kann nicht allgemeingültig entschieden werden, ob sich im Gesamteffekt das größere Parallelitätspotential zugunsten der Breitensuche oder das größere Potential wissensbasierter Metakonzepte zugunsten der Tiefensuche auswirkt.

### 3.2.2 Spezielle Ausformung zweier exemplarischer Algorithmus-Konzepte

#### 3.2.2.1 Schaltphasenorientiertes Konzept

Durch das schaltphasenorientierte Konzept werden die blackboard-Dämonen nach dem Kriterium geliedert, ob sie als Operation die Prüfung einer Aktivierungsbedingung (Aktivierungsphase) oder die Berechnung einer Schaltwirkung (Aktionsphase) ausführen. Es wird unterstellt, daß jeder Dämon jeweils eine der beiden vorgenannten Operationen für genau eine Transition des zugrundeliegenden Petrinetzes ausführt.

Die erste Gruppe der Aktivierungs-Dämonen kann in jedem aktuellen Knoten des Erreichbarkeitsgraphen - unabhängig von der repräsentierten Netzmarkierung - parallel auf alle Transitionen angewendet werden. Aus der zweiten Gruppe der Aktions-Dämonen lassen sich dagegen in einem bestimmten aktuellen Knoten nur diejenigen

Exemplare parallel anwenden, um die Schaltwirkungen der zugehörigen Transitionen zu ermitteln, für die zuvor in diesem Knoten die Aktivierung ihrer Transitionen nachgewiesen wurde. Daher geht in bezug auf jeweils dieselbe Transition die Aktivität des Aktivierungs-Dämons der des Aktions-Dämons notwendig voraus. Dagegen können die parallelen Aktivitäten der Aktivierungs-Dämonen einerseits und der Aktions-Dämonen andererseits ineinander verschachtelt sein, also auch (partiell) parallel geschehen. Sie müssen es aber nicht, weil die Aktivitäten dieser beiden Dämonen-Gruppen auch in jedem Knoten des Erreichbarkeitsgraphen streng sequentiell aufeinander folgen können.

Bei Verschachtelung von Aktivierungs- und Aktions-Dämonen schreiben die ersten, sobald sie die Aktivierung ihrer zugehörigen Transition festgestellt haben, diesen Sachverhalt in einen Zwischenspeicher, auf den die Aktions-Dämonen zugreifen. Sobald einer der letztgenannten in diesem Speicher die Information über die Aktivierung der ihm zugeordneten Transition vorfindet, beginnt er mit seiner Prozedur, deren Schaltwirkung zu ermitteln.

Das schaltphasenorientierte Konzept ist für die Anwendung der verdichteten Strategie geeignet, weil die Ermittlung der maximalen Mengen nebenläufig aktivierter Transitionen erst nach Feststellung aller Transitionen, die unter der jeweils betrachteten Markierung aktiviert sind, beginnen kann und vor dem Berechnen der Schaltwirkungen von Transitionen abgeschlossen sein muß<sup>39)</sup>.

---

39) Wenn die verdichtete Strategie zur Erzeugung von Knoten im Erreichbarkeitsgraphen verwendet wird, muß jeder Aktions-Dämon um die - problematische (vgl. Fußnote 35) - Prüfung erweitert werden, ob die Nullmarkierung überschritten wurde. Bei positivem Testergebnis ist der vom Dämon ausgeführte Schalt-Schritt aus nebenläufig aktivierten Transitionen in eine Schaltfolge seiner vereinzelter Transitionen aufzulösen, die durch den Knoten für die Nullmarkierung führt.

Die Trennung von Dämonen für die Aktivierungs- und die Aktionsphase ermöglicht eine solche zwischengeschaltete Ermittlung der potentiellen Schalt-Schritte. Allerdings ist dann die zeitliche Verschachtelung der Operationen von Aktivierungs- und Aktions-Dämonen nicht mehr möglich.

Stattdessen müssen die Aktions-Dämonen warten, bis die Aktivierungs-Dämonen alle aktivierten Transitionen ermittelt haben und die Steuerungskomponente hieraus die maximalen Mengen nebenläufig aktivierter Transitionen gebildet hat. Für jeden Schalt-Schritt, der tatsächlich ausgeführt werden soll, ruft die Steuerungskomponente die Aktions-Dämonen aller Transitionen auf, die Elemente dieses Schalt-Schritts sind. Diese Aktions-Dämonen können in einer beliebigen Reihenfolge<sup>40)</sup> auf den betrachteten Knoten des Erreichbarkeitsgraphen angewendet werden. Sie bestimmen in der Gesamtheit ihrer Schaltwirkungen den Nachfolger-Knoten.

Im folgenden wird ein paralleler blackboard-Algorithmus skizziert, dessen Dämonen schaltphasenorientiert gruppiert sind und dessen Erreichbarkeitsgraph nicht-verdichtet entwickelt wird. Die Angehörigen der beiden Dämonen-Gruppen werden in jedem Knoten des Erreichbarkeitsgraphen verschränkt aktiv. Als Strategie der übergeordneten Steuerungskomponente wird vorausgesetzt, daß alle aktivierten Transitionen geschaltet werden, allerdings unter den einschränkenden Vorgaben, die oben aus den spezifischen Eigenarten von Netz-Modellen für die Lösung linear-ganzzahliger OR-Modelle

---

40) Diese Reihenfolge kann sich z.B. nach dem Status der Dämonen richten, der entweder "frei" oder - durch die Beteiligung an der Untersuchung eines anderen, parallelen Schalt-Schritts - "belegt" sein kann. Im ersten Fall würde die Aktivität des Dämons initiiert. Wenn mehrere Dämonen frei sind, wäre einer zufällig oder nach Maßgabe einer vorgegebenen Prioritätsordnung auszuwählen. Die belegten Dämonen werden dagegen nicht aktiviert, bis auch sie frei werden. Da mehrere Schalt-Schritte um die Aktivitäten dieser Dämonen konkurrieren können, ist ein Synchronisierungsmechanismus einzurichten, im einfachsten Fall wieder ein Zufallsmechanismus oder eine dynamische Prioritätsordnung, die jeweils aus Schritt-spezifischen Eigenschaften (z.B. den Indices der zugehörigen Transitionen) berechnet wird.

abgeleitet wurden<sup>41)</sup>. Jeweils nur eine der resultierenden Markierungen wird als neuer aktueller Knoten im Erreichbarkeitsgraphen weiter untersucht. Es handelt sich also um eine eingeschränkte Breitensuche mit niedrigem Parallelitätsgrad. Die Algorithmus-Beschreibung erfolgt in Pseudocode<sup>42)</sup>. Die Realisierung der nebenläufig operierenden Dämonen erfordert zur parallelen Implementierung dieses Algorithmus  $3m+1$  autonome Prozessoren.

SEQBEGIN: schaltphasenorientierter Algorithmus

```

erzeuge im Erreichbarkeitsgraphen den Knoten  $kn = \underline{m}_0$ 43)
setze die Menge der erzeugten Knoten des
Erreichbarkeitsgraphen auf  $KN := \{kn\}$ 
setze die Menge der zu untersuchenden Knoten auf
 $U := \{kn\}$ 
setze die Menge der erzeugten Kanten des
Erreichbarkeitsgraphen auf  $KA := \emptyset$ 

```

41) Vgl. die Anmerkungen auf S. 23ff.

42) Vgl. Jordan (1978), S. 82ff.; Jackson (1979), S. 30ff. Der Ausdruck "PARBEGIN ... PAREND" bedeutet, daß die Operations-Statements, die zwischen den Klammersymbolen "PARBEGIN" und "PAREND" aufgeführt sind, im Gegensatz zu ihrer hier erfolgenden sequentiellen Auflistung beim Ablauf des blackboard-Algorithmus parallel ausgeführt werden. Notwendige Voraussetzung hierfür ist, daß jede dieser Operationen durch einen autonom agierenden Prozessor implementiert wird. In analoger Weise stehen die Ausdrücke "SEQBEGIN ... SEQEND", "ITERBEGIN ... ITEREND", "IFBEGIN ... IFEND" und "CASEBEGIN ... CASEEND" für die Strukturblöcke einer Sequenz, einer Iteration, einer Bedingungsprüfung bzw. einer Fallunterscheidung.

43) Die Knoten des Erreichbarkeitsgraphen korrespondieren mit den Markierungen des zugrundeliegenden Petrinetzes (Stelle/Transition-Netzes). Daher werden fortan mit "M" - je nach Analysekontext - sowohl Knoten als auch die entsprechenden Markierungen angesprochen. Es handelt sich jeweils nur um eine unterschiedliche materielle Interpretation der formal identischen Symbole "M". Ferner werden die Markierungs-Vektoren im Erreichbarkeitsgraphen gewöhnlich als Zeilen-Vektoren dargestellt, während sie im algebraischen Kalkül der Petrinetz-Theorie - sofern sie als Vektoren aufgefaßt werden - als Spalten-Vektoren definiert sind. Um die Übersichtlichkeit der Algorithmus-Beschreibung nicht durch formale Symbole für die Kennzeichnung von (Spalten-)Vektoren und transponierten (Zeilen-)Vektoren zu beeinträchtigen, wird nachfolgend das Symbol "M" als Zeilen-Vektor aufgefaßt. Entsprechend bezeichnet auch 0 einen Nullvektor in transponierter Form.

ITERBEGIN: Knotenanalyse

DO UNTIL  $U = \emptyset$

ITER SEQBEGIN: Knotenanalyse

wähle aus der Menge  $U$  der zu untersuchenden Knoten genau einen Knoten  $kn_h = \underline{M}_h$  aus

setze die Menge  $AT(\underline{M}_h)$  der Transitionen  $t_i$  mit  $i = 1, \dots, m$ , die im Knoten  $\underline{M}_h$  aktiviert sind, auf  $AT(\underline{M}_h) := \emptyset$

aktiviere die Dämonen  $D_1, \dots, D_m$

PARBEGIN: Dämonen der Entscheidungsvariablen

DO WHILE mindestens ein Dämon  $D_g$  mit  $g = 1, \dots, m$  aktiviert ist

PAR SEQBEGIN: Dämon  $D_1$

IFBEGIN: Aktivierungsphase

IF Aktivierungsbedingung von Transition  $t_1$  im Knoten  $\underline{M}_h$  erfüllt durch:

$M_h(s_j, t_1) \geq w(s_j, t_1)$  für alle  $s_j \in V(t_1)$

THEN SEQBEGIN: Aktivierungsphase

setze  $AT(\underline{M}_h) := AT(\underline{M}_h) \cup \{t_1\}$

aktiviere Dämon  $D_{m+1}$

THEN SEQEND: Aktivierungsphase

IFEND: Aktivierungsphase

desaktiviere Dämon  $D_1$

PAR SEQEND: Dämon  $D_1$

... (PAR SEQBEGIN ... PAR SEQEND: Dämon  $D_g$  analog zu  $D_1$  für  $g = 2, \dots, m$ )

PAR SEQBEGIN: Dämon  $D_{m+1}$

IFBEGIN: Aktionsphase

IF  $t_1 \in AH(\underline{M}_h)$

THEN SEQBEGIN: Aktionsphase

berechne Nachfolger-Knoten  $\underline{M}_{h.1}$  durch Schalten von Transition  $t_1$  unter der Markierung des Knoten  $\underline{M}_h$ , d.h. ermittle mit Inzidenzmatrix  $\underline{C}$  des Petrinetzes und  $\underline{t}_1$  als Schalt-Vektor der Transition  $t_1$ :

$\underline{M}_{h.1} := \underline{M}_h + \underline{C} \cdot \underline{t}_1$

IFBEGIN: Schaltwirkung

IF  $\underline{M}_{h.1} \in KN$

THEN SEQBEGIN: Schaltwirkung

erzeuge die Kante<sup>44)</sup>  $ka = (\underline{M}_h, t_1, \underline{M}_{h.1})$

setze  $KA := KA \cup \{ka\}$

THEN SEQEND: Schaltwirkung

44)  $(\underline{M}_h, t_1, \underline{M}_{h.1})$  bedeutet eine Kante, die vom Knoten  $\underline{M}_h$  zum Knoten  $\underline{M}_{h.1}$  gerichtet und mit der schaltenden Transition  $t_1$  beschriftet ist.



```

ELSE SEQBEGIN: Schaltwirkung
  erzeuge den Knoten  $kn = \underline{M}_{h.1}$ 
  erzeuge die Kante  $ka = (\underline{M}_h, t_1, \underline{M}_{h.1})$ 
  setze  $KN := KN \cup \{kn\}$ 
  setze  $KA := KA \cup \{ka\}$ 
  IFBEGIN: Begrenzungstest
    IF  $\underline{M}_{h.1}(s_j) > \max \{w(s_j, t_k) \mid \dots$ 
       $\dots t_k \in N(s_j)\}$ 
      für  $j = r+1 \vee \dots \vee j = n$ 
      und es existiert ein Referenz-
      knoten  $\underline{M}_r \in KN$  mit:
       $\underline{M}_h \succ \underline{M}_r \wedge \underline{M}_h(s_j) > \underline{M}_r(s_j) \wedge \underline{M}_r$ 
      wird durch Schalten der
      Transition  $t_1$  im Erreichbar-
      keitsgraphen in einen Folge-
      knoten überführt

    THEN markiere Knoten  $\underline{M}_{h.1}$  als
      einen unfruchtbaren Endknoten
      des lösungsrelevanten Erreich-
      barkeitsgraphen-Ausschnitts:
       $\underline{M}_{h.1} := \underline{M}_{h.1}^*$ 
    ELSE setze  $U := U \cup \{kn\}$ 
  IFEND: Begrenzungstest
ELSE SEQEND: Schaltwirkung
IFEND: Schaltwirkung
  THEN SEQEND: Aktionsphase
  IFEND: Aktionsphase
  deaktiviere Dämon  $D_{m+1}$ 
  PAR SEQEND: Dämon  $D_{m+1}$ 
  ... (PAR SEQBEGIN ... PAR SEQEND: Dämon  $D_g$ 
    analog zu  $D_{m+1}$  für  $g = m+2, \dots, 2m$ )
  PAREND: Dämonen der Entscheidungsvariablen
  IFBEGIN: Schlupfvariablen
  IF  $AT(\underline{M}_h) = \emptyset$  (d.h. im Knoten  $\underline{M}_h$  war keine
    Transition  $t_i$  einer Entschei-
    dungsvariablen  $x_i$  mit
     $i = 1, \dots, m$  aktiviert)

  THEN SEQBEGIN: Schlupfvariablen
    aktiviere die Dämonen  $D_{2m+1}, \dots, D_{3m}, D_{3m+1}$ 
  PARBEGIN: Dämonen der Schlupfvariablen
    DO WHILE mindestens ein Dämon  $D_g$  mit  $g =$ 
       $2m+1, \dots, 3m, 3m+1$  aktiviert ist

```

```

PAR SEQBEGIN: Dämon  $D_{2m+1}$ 
  ITERBEGIN45): Erzeugen der Aktivierung
                von Transition  $t_1$  im
                Folgeknoten  $M_{h.1}$ 
DO FOR jede Stelle  $s_j$  mit  $j = 1, \dots, n$ 
  ITER CASEBEGIN: Folgeknoten
    CASE1  $M_h(s_j) < w(s_j, t_1)$ 
    THEN IFBEGIN: Case1
      IF  $j = 1 \vee \dots \vee j = r$ 
      THEN SEQBEGIN: Case1
        berechne die Schaltanzahl  $d_{m+n+j}$ 
        für Transition  $t_i$  mit  $i = m+n+j$ 
        durch:
           $d_{m+n+j} := w(s_j, t_1) - M_h(s_j)$ 
           $M_{h.1}(s_j) := M_h(s_j) + d_{m+n+j}$ 
        THEN SEQEND: Case1
      ELSE SEQBEGIN: Case1
        berechne die Schaltanzahl  $d_{m+j}$ 
        für Transition  $t_i$  mit  $i = m+j$ 
        durch:
           $d_{m+j} := w(s_j, t_1) - M_h(s_j)$ 
           $M_{h.1}(s_j) := M_h(s_j) + d_{m+j}$ 
        ELSE SEQEND: Case1
      THEN IFEND: Case1
    CASE2  $M_{h.1}(s_j) = M_h(s_j)$ 
    THEN SEQBEGIN: Case2
      IFBEGIN: Case2
        IF  $j = 1 \vee \dots \vee j = r$ 
        THEN  $d_{m+j} := 0$  und  $d_{m+n+j} := 0$ 
        ELSE  $d_{m+j} := 0$ 
      IFEND: Case2
      setze  $M_{h.1}(s_j) := M_h(s_j)$ 
    THEN SEQEND: Case2

```

45) Auch diese Iteration kann - bei hinreichender Anzahl autonomer Prozessoren - als parallele Struktur PARBEGIN ... PAREND implementiert werden, weil die einzelnen Iterationsschritte voneinander unabhängig sind.

```

CASE3  $M_h(s_j) > w(s_j, t_1)$ 
THEN IFBEGIN: Case3
  IF  $j = 1 \vee \dots \vee j = r$ 
  THEN SEQBEGIN: Case3
    berechne die Schaltanzahl  $d_{m+j}$ 
    für Transition  $t_i$  mit  $i = m+j$ 
    durch:
       $d_{m+j} := M_h(s_j) - w(s_j, t_1)$ 
       $M_{h.1}(s_j) := M_h(s_j) - d_{m+j}$ 
    THEN SEQEND: Case3
  ELSE SEQBEGIN: Case3
     $d_{m+j} := 0$ 
     $M_{h.1}(s_j) := M_h(s_j)$ 
  ELSE SEQEND: Case3
  THEN IFEND: Case3
  ITER CASEEND: Folgeknoten
ITEREND: Erzeugen der Aktivierung von
  Transition  $t_1$  im Folgeknoten
   $\underline{M}_{h.1}$ 
  erzeuge den Knoten  $kn = \underline{M}_{h.1}$  mit
   $\underline{M}_{h.1} = (M_{h.1}(s_1), \dots, M_{h.1}(s_n))$ 
  setze  $KN := KN \cup \{kn\}$ 
  erzeuge die Kante  $ka$  mit46)
   $ka = (\underline{M}_h, (d_{m+1}t_{m+1}, \dots, d_q t_q), \underline{M}_{h.1})$ 
  setze  $KA := KA \cup \{ka\}$ 
  setze  $U := U \cup \{kn\}$ 
  deaktiviere Dämon  $D_{2m+1}$ 
PAR SEQEND: Dämon  $D_{2m+1}$ 
... (PAR SEQBEGIN ... PAR SEQEND: Dämon  $D_g$ 
  analog zu  $D_{2m+1}$  für  $g = 2m+2, \dots, 3m$ )

```

46) Die Kanten gewöhnlicher Erreichbarkeitsgraphen, die jeweils nur mit einer schaltenden Transition beschriftet sind, werden hier in der Weise verallgemeinert, daß sie die Schaltfolgen  $(d_{m+1}t_{m+1}, \dots, d_q t_q)$  als Aufschrift tragen. In diesen Schaltfolgen wird jede Transition  $t_i$  mit  $i = m+r+1, \dots, q$ , die auf Grund der Konstruktion des zugrundeliegenden Petri-netzes genau eine Marke auf ihrer nachgeordneten Stelle ablegt,  $d_i$ -mal geschaltet. Transitionen  $t_i$ , für die kein  $d_i > 0$  ermittelt wurde, gehören nicht zu den Schaltfolgen, da sie keinmal geschaltet werden ( $d_i = 0$ ).

Es handelt sich bei einer solchen multiplen Kante um eine verdichtete, aber äquivalente Darstellung einer Folge von alternierenden Knoten und (einfachen) Kanten in gewöhnlichen Erreichbarkeitsgraphen, die in denselben Anfangs- und Endknoten wie die o.a. multiple Kante beginnt bzw. endet und auf der dieselben Transitionen in derselben Anzahl geschaltet werden wie entlang der multiplen Kante.

```

PAR SEQBEGIN: Dämon  $D_{3m+1}$ 
  IFBEGIN: Nullmarkierung
    IF  $M_h(s_j) = 0$  für alle  $j = r+1, \dots, n$ 
      THEN SEQBEGIN: Nullmarkierung
        setze die Schaltanzahl  $d_{m+j}$  für alle
        Transitionen  $t_{m+j}$  mit  $j = 1, \dots, r$ 
        auf:  $d_{m+j} := M_h(s_j)$ 
        setze  $d_i$  für alle übrigen Transiti-
        onen mit  $i = 1, \dots, m, m+r+1, \dots, q$  auf:
         $d_i := 0$ 
        erzeuge die Kante  $ka$  mit
         $ka = (\underline{M}_h, (d_{m+1}t_{m+1}, \dots, d_qt_q), \underline{0})$ 
        setze  $KA := KA \cup \{ka\}$ 
      THEN SEQEND: Nullmarkierung
    ELSE die Nullmarkierung kann grund-
    sätzlich nicht erreicht werden
  IFEND: Nullmarkierung
  deaktiviere Dämon  $D_{3m+1}$ 
PAREND SEQEND: Dämon  $D_{3m+1}$ 
PAREND: Dämonen der Schlupfvariablen
THEN SEQEND: Schlupfvariablen
IFEND: Schlupfvariablen
setze  $U := U - \{\underline{M}_h\}$ 
ITER SEQEND: Knotenanalyse
ITEREND: Knotenanalyse
ermittle die Menge  $W$  aller Start-Ziel-Wege47) im
Erreichbarkeitsgraphen, die vom Knoten  $\underline{M}_0$  als Start-
knoten ausgehen und zum selben Knoten als Zielknoten
zurückführen
IFBEGIN: Lösungstest
  IF  $W = \emptyset$ 

```

47) Da der Erreichbarkeitsgraph per constructionem Zy-  
klen enthalten kann, ist die Menge  $W$  seiner Start-  
Ziel-Wege potentiell unendlich groß. Denn sobald  
ein Zyklus existiert, kann eine unendlich große We-  
ge(teil)menge dadurch erzeugt werden, daß dieser  
Zyklus beliebig oft, aber auf jedem Weg aus dieser  
Menge in unterschiedlicher Anzahl wiederholt wird.  
Damit die Konstruktion der Wegemenge  $W$  dennoch end-  
lich bleibt, sind durch Einführung des Iterations-  
Ausdrucks ITER(...) für jeden Zyklus Wegeteilmengen  
zu bilden. Zu jeder Wegeteilmenge, in deren Defini-  
tion mindestens ein Iterations-Ausdruck enthalten  
ist, gehören so viele Wege, wie durch null-, ein-,  
zweimalige usw. Wiederholung des jeweils repräsen-  
tierten Zyklus erzeugt werden können.

```

THEN gebe aus: "Es existiert keine Lösung des
linear-ganzzahligen OR-Modells,
weil es für das transformierte linear-homogene
Gleichungssystem keine oder nur unzulässige
nicht-triviale Lösungen gibt."

ELSE SEQBEGIN: Lösungstest

  setze die Lösungsmenge LM des linear-ganzzahligen
  OR-Modells auf LM :=  $\emptyset$ 

  ITERBEGIN: Lösungsermittlung

    DO UNTIL W =  $\emptyset$ 

      ITER SEQBEGIN: Lösungsermittlung

        wähle einen beliebigen48) Start-Ziel-Weg w
        aus der Menge W aus

        ermittle für jede Transition  $t_k$  mit  $k=m+1, \dots, m+r$ 
        und für jede Transition  $t_{k+n}$  die
        Anzahl  $a_k$  bzw.  $a_{k+n}$ , mit der sie im Weg w
        als - einfache oder multiple - Kanten-
        beschriftung enthalten ist

        IFBEGIN: Zulässigkeitstest  $T_1$ 

          IF  $a_k - a_{k+n} \leq b_{k-m}$  für alle  $k=m+1, \dots, m+r$ 
          THEN SEQBEGIN: Zulässigkeitstest  $T_1$ 

            ermittle für jede Transition  $t_h$  mit
             $h=m+r+1, \dots, m+n$  die Anzahl  $a_h$ , mit der
            sie im Weg w als - einfache oder multiple
            - Kantenbeschriftung enthalten ist

            IFBEGIN: Zulässigkeitstest  $T_2$ 

              IF  $a_h \geq -b_{h-m}$  für alle
               $h = m+r+1, \dots, m+n$ 

                THEN SEQBEGIN: Zulässigkeitstest  $T_2$ 

                  ermittle für jede Transition  $t_i$  mit
                   $i = 1, \dots, m$  die Anzahl  $a_i$ , mit der
                  sie im Weg w als Kantenbeschriftung
                  enthalten ist ~

                  bilde die Lösung  $\underline{x}$  des linear-
                  ganzzahligen OR-Modells mit:
                   $\underline{x}^T = (a_1, \dots, a_m)$ 
                  setze LM := LM  $\cup$  { $\underline{x}$ }

                THEN SEQEND: Zulässigkeitstest  $T_2$ 

              IFEND: Zulässigkeitstest  $T_2$ 

            THEN SEQEND: Zulässigkeitstest  $T_1$ 

```

48) Zur Beherrschung der potentiell unendlich großen Anzahl von Start-Ziel-Wegen im Erreichbarkeitsgraphen - vgl. Fußnote 47) - wird hierbei so verfahren, daß aus Wegeteilmengen, deren Definitionen mindestens einen Iterations-Ausdruck ITER(...) umfassen, zunächst immer solche Wege ausgewählt werden, auf denen der jeweils zugehörige Zyklus in der minimalen - noch nicht entlang eines bereits untersuchten Weges realisierten - Anzahl wiederholt wird.

```

ELSE streiche49) alle Wege  $w'$  aus der We-
menge  $W$ , die sich vom Weg  $w$  nur da-
durch unterscheiden, daß sie minde-
stens einen Zyklus, der Bestandteil
von Weg  $w$  ist, häufiger wiederholen,
als es bei Weg  $w$  der Fall ist, und auf
denen die in (mindestens) einem der
beiden vorangegangenen Zulässigkeits-
tests verletzten Restriktionen mit
Sicherheit nicht mehr in ihrer Gesamt-
heit erfüllt werden können

IFEND: Zulässigkeitstest  $T_1$ 
setze  $W := W - \{w\}$ 

ITER SEQEND: Lösungsermittlung
ITEREND: Lösungsermittlung
ELSE SEQEND: Lösungstest
IFEND: Lösungstest
IFBEGIN: Nulltest
IF  $\underline{x}^T = (x_1, \dots, x_m) = \underline{0}$  ist eine zulässige Lösung
von  $\underline{A} \cdot \underline{x} \leq \underline{b}$ 
THEN setze  $LM := LM \cup \{\underline{x}\}$ 
IFEND: Nulltest
IFBEGIN: Lösungsausgabe
IF  $LM = \emptyset$ 
THEN gebe aus: "Es existiert keine Lösung des
linear-ganzzahligen OR-Modells, weil die
nicht-trivialen "Lösungen" des linear-
homogenen Gleichungssystems unzulässig sind.
ELSE gebe aus: "Die Lösungsmenge des linear-
ganzzahligen OR-Modells ist:"
gebe aus:  $LM$ 
IFEND: Lösungsausgabe
SEQEND: schaltphasenorientierter Algorithmus

```

49) Vgl. zu den Voraussetzungen dieser Streichoperation die Ausführungen in den Fußnoten 47) und 48). Mit Sicherheit können - wegen der Monotonie der Schaltanzahlen beim Vorwärtsschreiten im Erreichbarkeitsgraphen - Restriktionen  $j$  mit  $j = 1, \dots, r$  nicht mehr durch zusätzliches Wiederholen von Zyklen erfüllt werden, wenn die jeweils betroffenen Zyklen keine negativen Werte für die Differenzen  $x_k - x_{k+n}$  mit  $k=m+j$  aufweisen. Denn dann läßt sich eine solche Restriktionsverletzung infolge des Überschreitens der Restriktionskonstanten  $b_j$  niemals mehr dadurch heilen, daß das Wiederholen der o.a. Zyklen das Ausmaß der Restriktionsverletzung reduziert und schließlich eliminiert.

### 3.2.2.2 Transitionenorientiertes Konzept

Beim transitionenorientierten Konzept erstreckt sich jeder blackboard-Dämon genau auf eine Transition, für die er die Aktivierungsbedingung prüft und danach - bei positivem Prüfergebnis - die Schaltwirkung in Gestalt der neuen Netzmarkierung und des entsprechenden neuen Knotens im Erreichbarkeitsgraphen ermittelt. In jedem Knoten des Erreichbarkeitsgraphen können alle Dämonen parallel agieren.

Dieses Konzept ist insofern übersichtlicher strukturiert als das schaltphasenorientierte<sup>50)</sup>, als eine 1:1-Zuordnung zwischen Transitionen des zugrundeliegenden Petrinetzes und der blackboard-Dämonen erfolgt. Des weiteren entfällt die mögliche, aber komplexe Verschachtelung der Aktivitäten von Exemplaren aus der Gruppe der Aktivierungs- und der Aktivitäts-Dämonen. Daher präferiert der Verf. das transitionenorientierte Konzept. Allerdings wird diese Transparenz durch den Nachteil erkauft, daß die verdichtete Strategie der Knotenerzeugung im Erreichbarkeitsgraphen nicht mehr möglich ist. Denn die unmittelbare Hintereinanderausführung von Test der Aktivierungsbedingung und Ermitteln der Schaltwirkung für eine Transition durch ihren Dämon verhindert es, zwischen diesen beiden Schaltphasen durch die Analyse aller in demselben Knoten des Erreichbarkeitsgraphen aktivierten Transitionen die maximalen Mengen nebenläufig aktivierter Transitionen zu bilden.

---

50) Es ist allerdings auch in der Hinsicht komplexer strukturiert, als seine Dämonen aus einer zweielementigen Operationen-Sequenz bestehen, während die Dämonen des schaltphasenorientierten Konzepts nur auf einer isolierten Operation beruhen.

Zu seiner Verdeutlichung wird ebenfalls ein exemplarischer paralleler Netzalgorithmus in Pseudocode skizziert<sup>51)</sup>. Für die Steuerungskomponente wird wieder die eingeschränkte Breitensuche unterstellt. Ihr kommt ein mittlerer Parallelitätsgrad zu, weil von den neu erzeugten Knoten jeweils maximal drei als neue aktuelle Knoten parallel weiterverfolgt werden. Es werden  $2m+4$  Prozessoren für die parallele Implementierung der nebenläufig operierenden Dämonen benötigt.

SEQBEGIN: transitionenorientierter Algorithmus

errichte im Erreichbarkeitsgraphen den Knoten  $kn = \underline{M}_0$

setze die Menge der erzeugten Knoten des Erreichbarkeitsgraphen auf  $KN := \{kn\}$

setze die Menge der zu untersuchenden Knoten auf  $U := \{kn\}$

setze die Menge der erzeugten Kanten des Erreichbarkeitsgraphen auf  $KA := \emptyset$

PARBEGIN: Knotenanalyse

DO UNTIL  $U = \emptyset$  52)

PAR SEQBEGIN: Prozessor  $P_1$

wähle aus der Menge  $U$  der zu untersuchenden Knoten genau einen Knoten  $\underline{M}_h$  aus

setze die Menge  $AT(\underline{M}_h)$  der Transitionen  $t_i$  mit  $i = 1, \dots, m$ , die im Knoten  $\underline{M}_h$  aktiviert sind, auf  $AT(\underline{M}_h) := \emptyset$

aktiviere die Dämonen  $D_1, \dots, D_m$

PARBEGIN: Dämonen der Entscheidungsvariablen

DO WHILE mindestens ein Dämon aktiviert ist

PAR SEQBEGIN: Dämon  $D_1$

IFBEGIN: Aktivität

IF Aktivierungsbedingung von Transition  $t_1$  im Knoten  $\underline{M}_h$  erfüllt durch:

$M_h(s_j, t_1) \geq w(s_j, t_1)$  für alle  $s_j \in V(t_1)$

51) Es gelten analog die Anmerkungen, die zum schaltphasenorientierten Algorithmus in den Fußnoten 43) bis 49) erfolgten.

52) Da die drei Prozessoren parallel auf die Menge  $U$  der noch zu untersuchenden Knoten zugreifen und durch diesen Zugriff die Extension dieser Menge verringern (sowie ggf. auch wieder erweitern), muß ihr Zusammenwirken durch einen speziellen Mechanismus der Steuerungskomponente synchronisiert werden. Z.B. kommt hierfür das Semaphoren-Konzept in Frage.



THEN SEQBEGIN: Aktivität

setze  $AT(\underline{M}_h) := AT(\underline{M}_h) \cup \{t_1\}$

berechne Nachfolger-Knoten  $\underline{M}_{h.1}$  durch Schalten von Transition  $t_1$  unter der Markierung des Knoten  $\underline{M}_h$ , d.h ermittle mit Inzidenzmatrix  $\underline{C}$  des Petrinetzes und  $\underline{t}_1$  als Schalt-Vektor der Transition  $t_1$ :

$\underline{M}_{h.1} := \underline{M}_h + \underline{C} \cdot \underline{t}_1$

IFBEGIN: Schaltwirkung

IF  $\underline{M}_{h.1} \in KN$

THEN SEQBEGIN: Schaltwirkung

erzeuge die Kante  $ka = (\underline{M}_h, t_1, \underline{M}_{h.1})$

setze  $KA := KA \cup \{ka\}$

THEN SEQEND: Schaltwirkung

ELSE SEQBEGIN: Schaltwirkung

erzeuge den Knoten  $kn = \underline{M}_{h.1}$

erzeuge die Kante  $ka = (\underline{M}_h, t_1, \underline{M}_{h.1})$

setze  $KN := KN \cup \{kn\}$

setze  $KA := KA \cup \{ka\}$

IFBEGIN: Begrenzungstest

IF  $\underline{M}_{h.1}(s_j) > \max \{w(s_j, t_k) \mid \dots$   
 $\dots t_k \in N(s_j)\}$   
 für  $j = r+1 \vee \dots \vee j = n$   
 und es existiert ein Referenzknoten  $\underline{M}_r \in KN$  mit:  
 $\underline{M}_h \geq \underline{M}_r \wedge \underline{M}_h(s_j) > \underline{M}_r(s_j) \wedge \underline{M}_r$   
 wird durch Schalten der Transition  $t_1$  im Erreichbarkeitsgraphen in einen Folgeknoten überführt

THEN markiere Knoten  $\underline{M}_{h.1}$  als einen unfruchtbaren Endknoten des lösungsrelevanten Erreichbarkeitsgraphen-Ausschnitts:

$\underline{M}_{h.1} := \underline{M}_{h.1}^*$

ELSE setze  $U := U \cup \{kn\}$

IFEND: Begrenzungstest

ELSE SEQEND: Schaltwirkung

IFEND: Schaltwirkung

THEN SEQEND: Aktivität

IFEND: Aktivität

desaktiviere Dämon  $D_1$

PAR SEQEND: Dämon  $D_1$

... (PAR SEQBEGIN ... PAR SEQEND: Dämon  $D_g$   
 analog zu  $D_1$  für  $g = 2, \dots, m$ )

PAREND: Dämonen der Entscheidungsvariablen

IFBEGIN: Schlupfvariablen

IF  $AT(\underline{M}_h) = \emptyset$  (d.h. im Knoten  $\underline{M}_h$  war keine  
Transition  $t_i$  einer Entschei-  
dungsvariablen  $x_i$  mit  
 $i = 1, \dots, m$  aktiviert)

THEN SEQBEGIN: Schlupfvariablen

aktiviere die Dämonen  $D_{m+1}, \dots, D_{2m}, D_{2m+1}$

PARBEGIN: Dämonen der Schlupfvariablen

DO WHILE mindestens ein Dämon  $D_g$  mit  
 $g = m+1, \dots, 2m$  aktiviert ist

PAR SEQBEGIN: Dämon  $D_{m+1}$

ITERBEGIN: Erzeugen der Aktivierung von  
Transition  $t_1$  im Folgeknoten  
 $\underline{M}_{h.1}$

DO FOR jede Stelle  $s_j$  mit  $j = 1, \dots, n$

ITER CASEBEGIN: Folgeknoten

CASE<sub>1</sub>  $M_h(s_j) < w(s_j, t_1)$

THEN IFBEGIN: Case<sub>1</sub>

IF  $j = 1 \vee \dots \vee j = r$

THEN SEQBEGIN: Case<sub>1</sub>

berechne die Schaltanzahl  $d_{m+n+j}$   
für Transition  $t_i$  mit  $i = m+n+j$   
durch:

$$d_{m+n+j} := w(s_j, t_1) - M_h(s_j)$$

$$M_{h.1}(s_j) := M_h(s_j) + d_{m+n+j}$$

THEN SEQEND: Case<sub>1</sub>

ELSE SEQBEGIN: Case<sub>1</sub>

berechne die Schaltanzahl  $d_{m+j}$   
für Transition  $t_i$  mit  $i = m+j$   
durch:

$$d_{m+j} := w(s_j, t_1) - M_h(s_j)$$

$$M_{h.1}(s_j) := M_h(s_j) + d_{m+j}$$

ELSE SEQEND: Case<sub>1</sub>

THEN IFEND: Case<sub>1</sub>

CASE<sub>2</sub>  $M_{h.1}(s_j) = M_h(s_j)$

THEN SEQBEGIN: Case<sub>2</sub>

IFBEGIN: Case<sub>2</sub>

IF  $j = 1 \vee \dots \vee j = r$

THEN  $d_{m+j} := 0$  und  $d_{m+n+j} := 0$

ELSE  $d_{m+j} := 0$

IFEND: Case<sub>2</sub>

setze  $M_{h.1}(s_j) := M_h(s_j)$

THEN SEQEND: Case<sub>2</sub>

```

CASE3  $M_h(s_j) > w(s_j, t_1)$ 
THEN IFBEGIN: Case3
  IF  $j = 1 \vee \dots \vee j = r$ 
  THEN SEQBEGIN: Case3
    berechne die Schaltanzahl  $d_{m+j}$ 
    für Transition  $t_i$  mit  $i = m+j$ 
    durch:
       $d_{m+j} := M_h(s_j) - w(s_j, t_1)$ 
       $M_{h.1}(s_j) := M_h(s_j) - d_{m+j}$ 
    THEN SEQEND: Case3
  ELSE SEQBEGIN: Case3
     $d_{m+j} := 0$ 
     $M_{h.1}(s_j) := M_h(s_j)$ 
  ELSE SEQEND: Case3
  THEN IFEND: Case3
  ITER CASEEND: Folgeknoten
ITEREND: Erzeugen der Aktivierung von
  Transition  $t_1$  im Folgeknoten
   $\underline{M}_{h.1}$ 
  erzeuge den Knoten  $kn = \underline{M}_{h.1}$  mit
   $\underline{M}_{h.1} = (M_{h.1}(s_1), \dots, M_{h.1}(s_n))$ 
  setze  $KN := KN \cup \{kn\}$ 
  erzeuge die Kante  $ka$  mit
   $ka = (\underline{M}_h, (d_{m+1}t_{m+1}, \dots, d_q t_q), \underline{M}_{h.1})$ 
  setze  $KA := KA \cup \{ka\}$ 
  setze  $U := U \cup \{kn\}$ 
  deaktiviere Dämon  $D_{m+1}$ 
PAR SEQEND: Dämon  $D_{m+1}$ 
... (PAR SEQBEGIN ... PAR SEQEND: Dämon  $D_g$ 
  analog zu  $D_{m+1}$  für  $g = m+2, \dots, 2m$ )
PAR SEQBEGIN: Dämon  $D_{2m+1}$ 
  IFBEGIN: Nullmarkierung
  IF  $M_h(s_j) = 0$  für alle  $j = r+1, \dots, n$ 
  THEN SEQBEGIN: Nullmarkierung
    setze die Schaltanzahl  $d_{m+j}$  für alle
    Transitionen  $t_{m+j}$  mit  $j = 1, \dots, r$ 
    auf:
       $d_{m+j} := M_h(s_j)$ 
    setze  $d_i$  für alle übrigen Transi-
    tionen mit  $i = 1, \dots, m, m+r+1, \dots, q$  auf:
       $d_i := 0$ 
    erzeuge die Kante  $ka$  mit
       $ka = (\underline{M}_h, (d_{m+1}t_{m+1}, \dots, d_q t_q), 0)$ 
    setze  $KA := KA \cup \{ka\}$ 
  THEN SEQEND: Nullmarkierung

```

```

ELSE die Nullmarkierung kann grundsätz-
      lich nicht erreicht werden
IFEND: Nullmarkierung
      deaktiviere Dämon  $D_{2m+1}$ 
      PAREND SEQEND: Dämon  $D_{2m+1}$ 
      PAREND: Dämonen der Schlupfvariablen
      THEN SEQEND: Schlupfvariablen
      IFEND: Schlupfvariablen
setze  $U := U - \{M_h\}$ 
PAR SEQEND: Prozessor  $P_1$ 
PAR SEQBEGIN: Prozessor  $P_2$ 
      ... (analog zu PAR SEQBEGIN ... PAR SEQEND für
           Prozessor  $P_1$ )
PAR SEQEND: Prozessor  $P_2$ 
PAR SEQBEGIN: Prozessor  $P_3$ 
      ... (analog zu PAR SEQBEGIN ... PAR SEQEND für
           Prozessor  $P_1$ )
PAR SEQEND: Prozessor  $P_3$ 
PAREND: Knotenanalyse
ermittle die Menge  $W$  aller Start-Ziel-Wege im
Erreichbarkeitsgraphen, die vom Knoten  $M_0$  als Start-
knoten ausgehen und zum selben Knoten als Zielknoten
zurückführen
IFBEGIN: Lösungstest
      IF  $W = \emptyset$ 
      THEN gebe aus: "Es existiert keine Lösung des
linear-ganzzahligen OR-Modells, weil es für
das transformierte linear-homogene Glei-
chungssystem keine oder nur unzulässige
nicht-triviale Lösungen gibt."
      ELSE SEQBEGIN: Lösungstest
      setze die Lösungsmenge  $LM$  des linear-ganzzahligen
OR-Modells auf  $LM := \emptyset$ 
      ITERBEGIN: Lösungsermittlung
      DO UNTIL  $W = \emptyset$ 
      ITER SEQBEGIN: Lösungsermittlung
      wähle einen beliebigen Start-Ziel-Weg  $w$  aus
der Menge  $W$  aus
      ermittle für jede Transition  $t_k$  mit  $k=m+1, \dots, m+r$ 
und für jede Transition  $t_{k+n}$  die
Anzahl  $a_k$  bzw.  $a_{k+n}$ , mit der sie im Weg  $w$ 
als - einfache oder multiple - Kanten-
beschriftung enthalten ist
      IFBEGIN: Zulässigkeitstest  $T_1$ 
      IF  $a_k - a_{k+n} \leq b_{k-m}$  für alle  $k=m+1, \dots, m+r$ 

```

```

THEN SEQBEGIN: Zulässigkeitstest T1
    ermittle für jede Transition th mit
    h=m+r+1,...,m+n die Anzahl ah, mit der
    sie im Weg w als - einfache oder multiple
    - Kantenbeschriftung enthalten ist
    IFBEGIN: Zulässigkeitstest T2
        IF    ah ≥ -bh-m für alle
            h = m+r+1,...,m+n
        THEN SEQBEGIN: Zulässigkeitstest T2
            ermittle für jede Transition ti mit
            i = 1,...,m die Anzahl ai, mit der
            sie im Weg w als Kantenbeschriftung
            enthalten ist
            bilde die Lösung x des linear-
            ganzzahligen OR-Modells mit:
             $\underline{x}^T = (a_1, \dots, a_m)$ 
            setze LM := LM ∪ {x}
        THEN SEQEND: Zulässigkeitstest T2
    IFEND: Zulässigkeitstest T2
THEN SEQEND: Zulässigkeitstest T1
ELSE streiche alle Wege w' aus der Wege-
menge W, die sich vom Weg w nur da-
durch unterscheiden, daß sie minde-
stens einen Zyklus, der Bestandteil
von Weg w ist, häufiger wiederholen,
als es bei Weg w der Fall ist, und auf
denen die in (mindestens) einem der
beiden vorangegangenen Zulässigkeits-
tests verletzten Restriktionen mit
Sicherheit nicht mehr in ihrer Gesamt-
heit erfüllt werden können
    IFEND: Zulässigkeitstest T1
    setze W := W - {w}
    ITER SEQEND: Lösungsermittlung
    ITEREND: Lösungsermittlung
    ELSE SEQEND: Lösungstest
IFEND: Lösungstest
IFBEGIN: Nulltest
    IF     $\underline{x}^T = (x_1, \dots, x_m) = \underline{0}$  ist eine zulässige Lösung
        von  $\underline{A} \cdot \underline{x} \leq \underline{b}$ 
    THEN setze LM := LM ∪ {x}
IFEND: Nulltest

```

```
IFBEGIN: Lösungsausgabe
  IF   LM =  $\emptyset$ 
  THEN gebe aus: "Es existiert keine Lösung des
               linear-ganzzahligen OR-Modells, weil die
               nicht-trivialen "Lösungen" des linear-
               homogenen Gleichungssystems unzulässig sind.
  ELSE gebe aus: "Die Lösungsmenge des linear-
               ganzzahligen OR-Modells ist:"
               gebe aus: LM
IFEND: Lösungsausgabe
SEQEND: transitionenorientierter Algorithmus
```

#### 4 Exemplarische Umsetzung eines OR-Modells in ein Netz-Modell

Gegeben sei als linear-ganzzahliges OR-Modell mit  $x_1, x_2, x_3 \in \mathbb{N}$ :

$$\begin{aligned} x_1 + 3x_2 - 4x_3 &\leq 4 \\ 2x_1 - x_2 + x_3 &\leq 3 \\ 2x_1 - 3x_2 + x_3 &\leq 0 \\ -4x_1 - x_2 + 4x_3 &\leq -1 \end{aligned}$$

Dieses Modell wird in ein linear-homogenes Gleichungssystem mit ergänzenden Restriktionen und  $x_1, \dots, x_9 \in \mathbb{N}$  äquivalent transformiert:

$$\begin{aligned} x_1 + 3x_2 - 4x_3 - x_4 + x_8 &= 0 & \wedge & & x_4 - x_8 &\leq 4 \\ 2x_1 - x_2 + x_3 - x_5 + x_9 &= 0 & \wedge & & x_5 - x_9 &\leq 3 \\ 2x_1 - 3x_2 + x_3 + x_6 &= 0 & \wedge & & x_6 &\geq 0 \\ -4x_1 - x_2 + 4x_3 + x_7 &= 0 & \wedge & & x_7 &\geq 1 \end{aligned}$$

Das linear-homogene Gleichungssystem wird in ein Stelle/Transition-Netz (Petrietz) mit 4 Stellen  $s_j$  und 9 Transitionen  $t_i$  überführt. In diesem Netz-Modell ist jeder Zeile des Gleichungssystem eine Stelle und jeder Spalte, in der jeweils nur eine Variable  $x_i$  steht, eine Transition zugeordnet. Das Ergebnis wird durch Abb. 1 auf S. 53 veranschaulicht.

Abb. 2 auf S. 54 gibt einen Ausschnitt aus dem zugehörigen nicht-verdichteten Erreichbarkeitsgraphen wieder. Aus den eingezeichneten Start-Ziel-Wegen läßt sich als Menge LM der Lösungen  $\underline{x}^T = (x_1, x_2, x_3)$  des originären OR-Modells ablesen:

$$LM = \{(0, 1, 0), (1, 1, 0), (1, 1, 1), (1, 2, 1), (2, 2, 1), (2, 3, 2)\}$$

Der Nullvektor  $\underline{x}^T = (0, 0, 0)$  stellt keine zulässige Lösung des OR-Modells dar - obwohl der korrespondierende Nullvektor  $\underline{x}_h = \underline{0}$  das linear-homogene Gleichungssystem erfüllt -, weil die vierte Restriktion mit  $b_4 = -1$  verletzt wird.

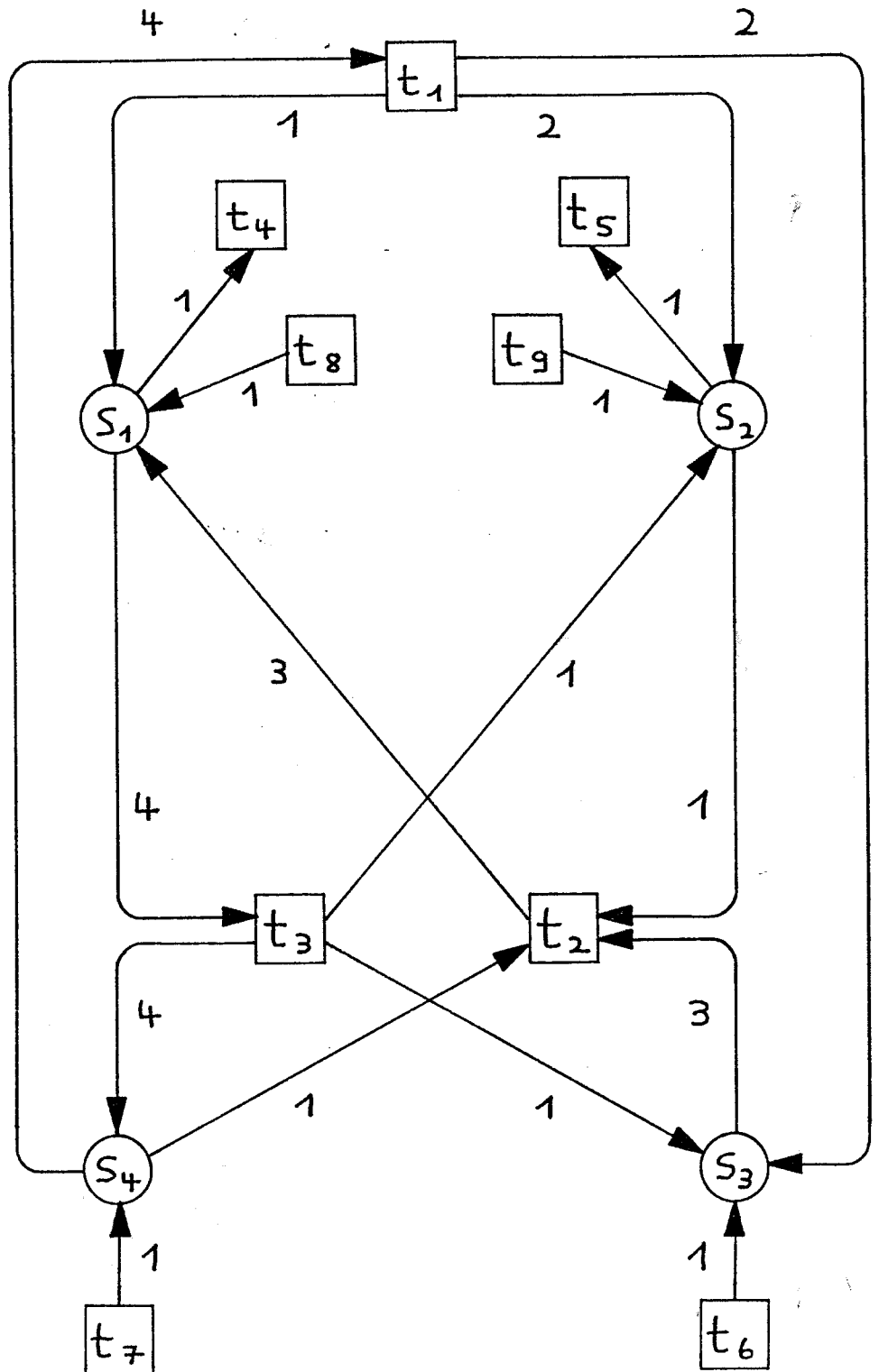


Abb. 1 : Stelle/Transition-Netz für das exemplarische linear-ganzzahlige OR-Modell



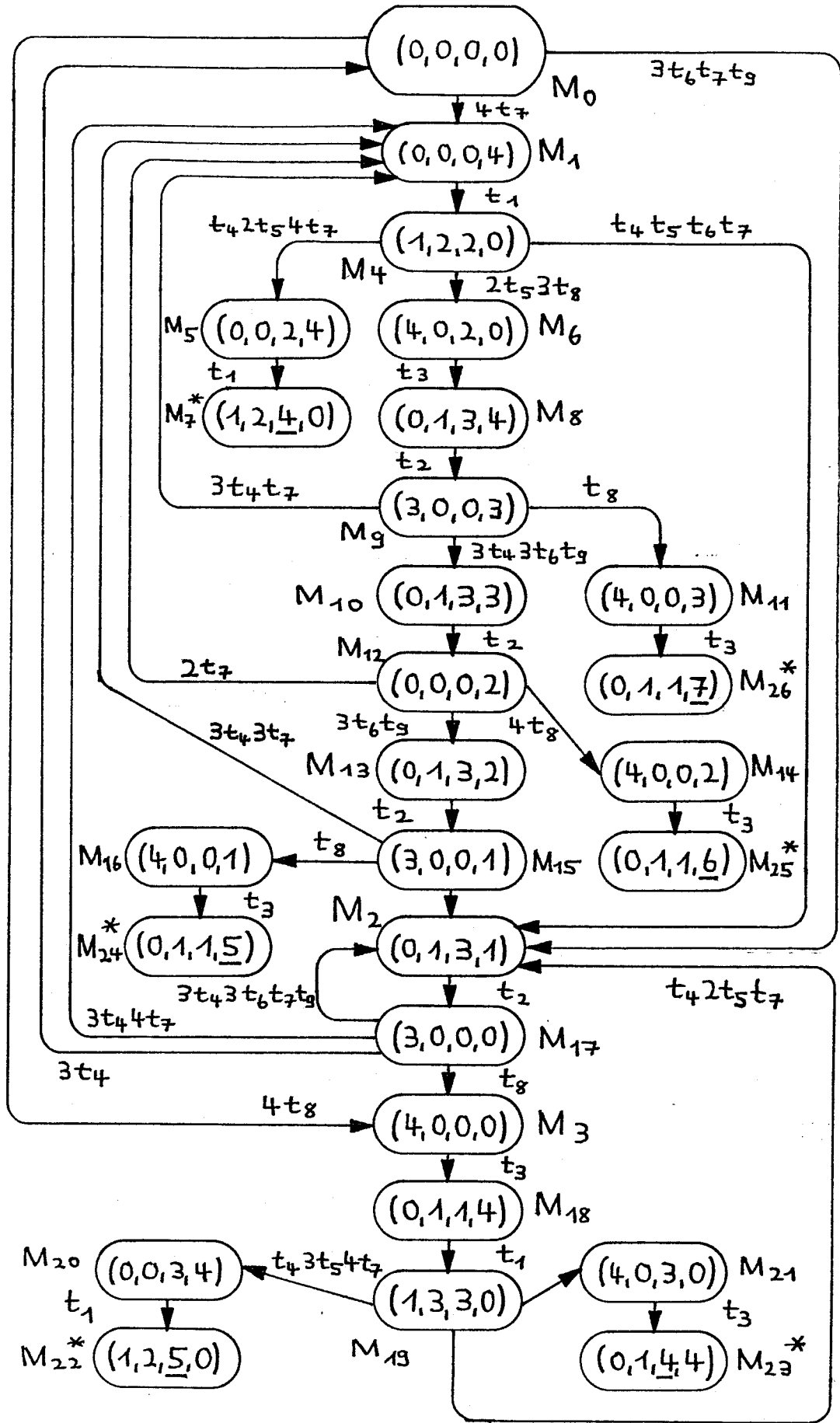


Abb. 2 : Erreichbarkeitsgraph für das Stelle/Transition-Netz aus Abb. 1

Der Erreichbarkeitsgraph aus Abb. 2 kann auf die komprimierte Form der Abb. 3 auf S. 56 reduziert werden, ohne daß Informationen hinsichtlich der Lösungen des o.a. (ersten) Ungleichungssystems verlorengehen. Es werden hierbei alle unverzweigten Schaltfolgen aus alternierenden Knoten und Kanten zu jeweils einer Kante zusammengefaßt, die mit den Bezeichnungen und Schaltanzahlen der Transitionen, welche durch die Schaltfolge geschaltet werden, beschriftet ist und dieselben Anfangs- und Endknoten wie die ersetzte Schaltfolge besitzt. Schaltfolgen, die zu unfruchtbaren Endknoten führen, werden eliminiert.

Der Übersichtlichkeit halber werden die Kantenbeschriftungen des komprimierten Erreichbarkeitsgraphen durch folgende Abkürzungen ersetzt:

$$\begin{array}{ll}
 tw_1 & := 4t_7 & tw_2 & := 3t_6t_7t_9 \\
 tw_3 & := 4t_8 & tw_4 & := t_1 \\
 tw_5 & := t_4t_5t_6t_7 & tw_6 & := t_2t_3 \\
 tw_7 & := 3t_4t_7 & tw_8 & := t_23t_4t_6t_9 \\
 tw_9 & := 2t_7 & tw_{10} & := t_23t_6t_9 \\
 tw_{11} & := 3t_4t_7 & tw_{12} & := 3t_43t_6t_9 \\
 tw_{13} & := t_2 & tw_{14} & := 3t_44t_7 \\
 tw_{15} & := 3t_43t_6t_7t_9 & tw_{16} & := 3t_4 \\
 tw_{17} & := t_8 & tw_{18} & := t_1t_3 \\
 tw_{19} & := t_42t_5t_7 & & 
 \end{array}$$

Es läßt sich leicht nachprüfen, daß im komprimierten Erreichbarkeitsgraphen genau so viele Start-Ziel-Wege  $w$  gebildet werden können, die alle Restriktionen des o.a. (zweiten) Ungleichungssystems erfüllen, wie das exemplarische OR-Modell Lösungen besitzt.

Als ein solcher zulässiger Start-Ziel-Weg und somit Lösung des OR-Modells ergibt sich z.B. gemäß Abb. 3  $w_1 = tw_1tw_4tw_5tw_{13}tw_{17}tw_{18}tw_{19}tw_{13}tw_{16}$ . Er repräsentiert die Lösung  $\underline{x}^T = (2, 2, 1)$ , da auf ihm die Transitionen  $t_1$ ,  $t_2$  und  $t_3$  zwei- bzw. zwei- bzw. einmal geschaltet werden. Dagegen ist der Start-Ziel-Weg  $w_2 = tw_1tw_4tw_5tw_6tw_8tw_{10}tw_{12}tw_{13}tw_{16}$  unzulässig, weil auf ihm die Transition  $t_4$  neunmal geschaltet wird, ohne daß die komplementäre Transition  $t_8$  einmal geschaltet wird. Dies widerspricht der Restriktion  $x_4 - x_8 \leq 4$ .

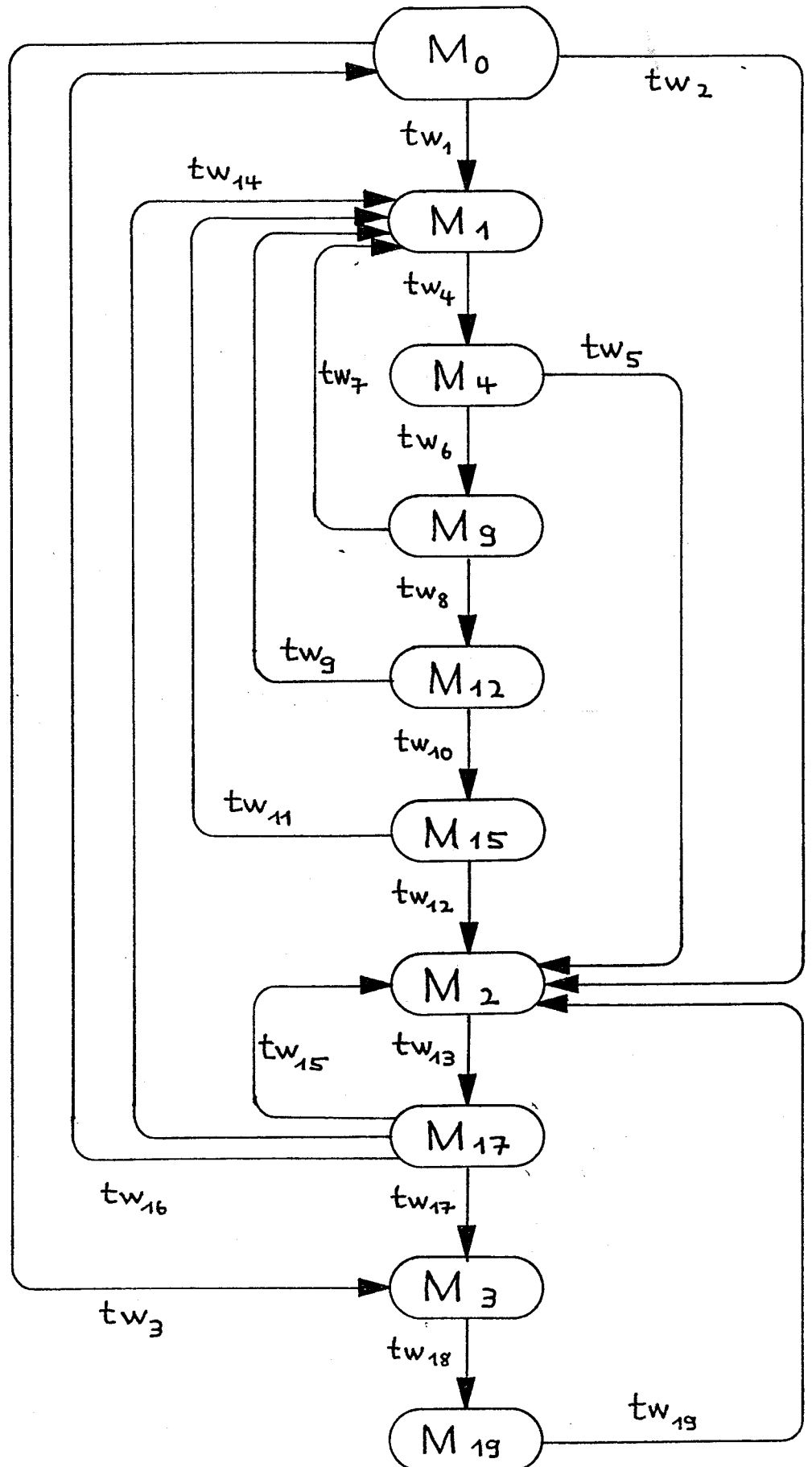


Abb. 3 : Komprimierte Darstellung des Erreichbarkeitsgraphen aus Abb. 2

Literaturverzeichnis

- Erman, D.L. u. V.R. Lesser: A Multi-Level Organization for Problem Solving Using Many, Diverse, Cooperating Sources of Knowledge, in: o.V.: Advance Papers of the Fourth International Joint Conference on Artificial Intelligence (IJCAI-75), 3.-8.09.1975 in Tbilisi, Vol. 2, o.O. (Cambridge/Massachusetts) 1975, S. 483-490.
- Hayes-Roth, B. u. F. Hayes-Roth: A Cognitive Model of Planning, in: Cognitive Science, Vol. 3 (1979), S. 275-310.
- Hayes-Roth, B.: A Blackboard Architecture for Control, in: Artificial Intelligence, Vol. 26 (1985), S. 251-321.
- Imai, M., Y. Yoshida u. T. Fukumura: A Parallel Searching Scheme for Multiprocessor Systems and its Application to Combinatorial Problems, in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, Vol. 1, o.O. (Stanford) 1979, S. 416-418.
- Jackson, M.A.: Grundsätze des Programmentwurfs, Darmstadt 1979.
- Jantzen, M. u. R. Valk: Formal Properties of Place/Transition Nets, 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. 165-212.
- Jordan, W. u. H. Urban: Strukturierte Programmierung, Berlin - Heidelberg - New York 1978.
- Kanal, L. u. V. Kupar: Branch & Bound Formulation for Sequential and Parallel Game Tree Searching: Preliminary Results, 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. 569-571.
- Kindervater, G.A.P. u. J.K. Lenstra: Parallel Algorithms, in: O'hEigeartaigh, M., J.K. Lenstra u. A.H.G. Rinnooy Kan (Hrsg.): Combinatorial Optimization - Annotated Bibliographies, Chichester - New York - Brisbane - Toronto - Singapore 1985, S. 106-128.
- Memmi, G.: Analyse des Reseaux de Petri par l'Algebre Lineaire, D.E.A. de Systemes Informatiques am Institut de Programmation der Universität Paris/Pierre et Marie Curie, Paris 1981.
- Nakagawa, S. u. T. Sakai: A Parallel Tree Search Method, in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, Vol. 2, o.O. (Stanford) 1979, S. 628-632.
- Pan, V. u. J. Reif: Efficient Parallel Linear Programming, in: Operations Research Letters, Vol. 5 (1986), S. 127-135.
- Pascoletti, K.-H.: Diophantische Systeme und Lösungsmethoden zur Bestimmung aller Invarianten in Petri-Netzen, Dissertation an der Universität Bonn, Bonn 1985.

Reisig, W.: Petri Nets - An Introduction, EATCS Monographs on Theoretical Computer Science Vol. 4, Berlin Heidelberg - New York - Tokyo 1985.

## Korrigenda

### 1) zu S. 7f.:

Die zwei Sätze "Jede nicht-triviale Lösung ... zu gewährleisten<sup>14)</sup>." sind - unter Fortfall der Fußnoten 13) und 14) - zu ersetzen durch:

Jede Lösung  $\underline{xh}^T = (\underline{x}^T, \underline{xp}^T, \underline{xp}'^T, \underline{xn}^T)$  für das Ungleichungssystem US ist zugleich eine Lösung des ursprünglichen linear-ganzzahligen OR-Modells US-SF.

### 2) zu S. 12:

Der letzte Satz "Ferner ... einzuschränken." ist ersatzlos zu streichen.

### 3) zu S. 15:

Im Anschluß an den 2. Abschnitt ist hinter "... führen." folgender Text einzufügen:

Durch die später eingeführten Konzepte der Wesentlichkeits- und Minimalitätsbedingung sowie der unfruchtbaren Knoten wird gewährleistet, daß die Menge der zu untersuchenden Start-Ziel-Wege endlich bleibt.

Zusätzlich ist zu prüfen, ob das zugrundeliegende Ungleichungssystem (US oder US-SF) den Nullvektor ( $\underline{xh}^T = \underline{0}^T$  bzw.  $\underline{x}^T = \underline{0}$ ) als triviale Lösung besitzt. Denn die Start-Ziel-Wege des Erreichbarkeitsgraphen sind per constructionem nicht-degeneriert, d.h. auf ihnen wird mindestens eine Transition mindestens einmal geschaltet. Dies bedeutet aufgrund der oben eingeführten Korrespondenz zwischen Transitionenschaltanzahlen und Variablen-Werten, daß mindestens eine Variable von Null verschieden sein muß, folglich triviale Modell-Lösungen im Erreichbarkeitsgraphen grundsätzlich nicht aufgefunden werden können.

Verzeichnis der Arbeitsberichte des  
Seminars für Allgemeine Betriebswirtschaftslehre,  
Industriebetriebslehre und Produktionswirtschaft der  
Universität zu Köln

(bis Sommer 1986: Seminar für Allgemeine  
Betriebswirtschaftslehre und Fertigungswirtschaft)

---

- Nr. 1: ZELEWSKI,STEPHAN: Entscheidungsmodelle zur Verschrottung von Fertigungshilfsmitteln, Köln 1984.
- Nr. 2: KERN,WERNER; ZELEWSKI,STEPHAN: Ein Zuordnungsmodell für Meßgeräte in Energie-Informationssystemen, Köln 1985.
- Nr. 3: KERN,WERNER; PETERS,ULRICH: Energiebewirtschaftung in industriellen Betrieben - Bericht über eine Befragung, Köln 1985.
- Nr. 4: BOOS,JOCHEN: Lokalisierung von Meßstellen für ein Informations-System zur Energiebewirtschaftung in industriellen Betrieben - Entwicklung eines OR-Modells mit einem Lösungsvorschlag -, Köln 1986.
- Nr. 5: ZELEWSKI,STEPHAN: Ansätze der Künstlichen Intelligenz-Forschung zur Unterstützung der Netzplantechnik, Köln 1986.
- Nr. 6: ZELEWSKI,STEPHAN: Schnittstellen bei betrieblichen Informationssystemen - eine Darstellung aus systemtheoretischer und betriebswirtschaftlicher Sicht -, Köln 1986.
- Nr. 7: ZELEWSKI,STEPHAN: Konzepte für Frühwarnsysteme und Möglichkeiten zu ihrer Fortentwicklung durch Beiträge der Künstlichen Intelligenz, Köln 1986.
- Nr. 8: ZELEWSKI,STEPHAN: Das Konzept der unscharfen Mengen unter besonderer Berücksichtigung ihrer linguistischen Interpretation - eine Lösung für unscharfe Probleme? -, Köln 1986.
- Nr. 9: ZELEWSKI,STEPHAN: Der tau-Wert: Aspekte eines neueren spieltheoretischen Ansatzes zur fairen Preisbildung aus kostenrechnerischer Perspektive, Köln 1986.
- Nr. 10: ZELEWSKI,STEPHAN: Competitive Bidding aus der Sicht des Ausschreibers - ein spieltheoretischer Ansatz -, Köln 1986.
- Nr. 11: ZELEWSKI,STEPHAN: Netztheoretische Ansätze zur Konstruktion und Auswertung von logisch fundierten Problembeschreibungen, Köln 1986.

- Nr. 12: ZELEWSKI,STEPHAN: Netztheoretische Fundierung von parallelen Algorithmen für die Lösung linear-ganzzahliger OR-Modelle, Köln 1986.
- Nr. 13: ZELEWSKI,STEPHAN: Intelligente Informationsbanksysteme - benutzerfreundliche Instrumente für die Informationsvermittlung? -, Köln 1986.
- Nr. 14: ZELEWSKI,STEPHAN: Komplexitätstheorie - ihr Beitrag zur Klassifizierung und Beurteilung von Problemen des Operations Research -, Köln 1986.
- Nr. 15: ZELEWSKI,STEPHAN: Der Informationsbroker, Köln 1986.
- Nr. 16: ZELEWSKI,STEPHAN: Soziale Verantwortbarkeit von Technologien, Köln 1986.
- Nr. 17: ZELEWSKI,STEPHAN: Expertensysteme - Übersicht über Konzeptionen und betriebswirtschaftliche Anwendungsmöglichkeiten -, Köln 1986.
- Nr. 18: ZELEWSKI,STEPHAN: Das Leistungspotential der Künstlichen Intelligenz für Industrieanwendungen - Ein Überblick -, Köln 1987.
- Nr. 19: ZELEWSKI,STEPHAN: Expertensysteme im "Büro der Zukunft" - Ein Überblick über Anwendungsperspektiven und Bewertungsaspekte -, Köln 1987.
- Nr. 20: KUMMER,SEBASTIAN: Computerunterstützung schöpferischer Forschungs- und Entwicklungsaktivitäten, Köln 1987.
- Nr. 21: ZELEWSKI,STEPHAN: Betriebswirtschaftliche Aspekte des industriellen Einsatzes von Expertensystemen - Anwendungsmöglichkeiten und Bewertung -, Köln 1988.
- Nr. 22: ZELEWSKI,STEPHAN: Expertensysteme für Prozeßplanung und -steuerung in der Fabrik der Zukunft - Ein Überblick über Konzepte und erste Prototypen -, Köln 1988.
- Nr. 23: ZELEWSKI,STEPHAN: Expertensysteme zur Sicherung der Betriebsbereitschaft in der Fabrik der Zukunft, Köln 1988.
- Nr. 24: ZELEWSKI,STEPHAN: Ansätze zur Bewertung des Einsatzes Künstlicher Intelligenz in Industrieunternehmen - aus produktiver und sozialer Sicht -, Köln 1988.