

# **Untersuchung ganzzahliger linear–algebraischer Ansätze zur Lösung des Erreichbarkeitsproblems in Stellen/Transitions–Netzen**

Diplomarbeit

zur Erlangung des Grades einer  
Diplom–Informatikerin  
im Studiengang Informatik

vorgelegt von

Kerstin Susewind

Betreuer: Prof. Dr. rer. nat. Kurt Lautenbach,  
Universität Koblenz–Landau, Fachbereich 4, Institut für Informatik  
Erstgutachter: Prof. Dr. rer. nat. Kurt Lautenbach,  
Universität Koblenz–Landau, Fachbereich 4, Institut für Informatik  
Zweitgutachter: Dr. rer. nat. Stephan Philippi,  
Universität Koblenz–Landau, Fachbereich 4, Institut für Informatik

Koblenz, im Mai 2011



# Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde noch keiner Prüfungsbehörde in gleicher oder ähnlicher Form vorgelegt.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

.....  
(Ort und Datum) (Unterschrift)



# Inhaltsverzeichnis

Abbildungsverzeichnis	ix
Tabellenverzeichnis	xiii
Algorithmenverzeichnis	xv
<b>1 Einführung</b>	<b>1</b>
1.1 Ziel und Aufbau der Arbeit . . . . .	2
1.2 Benötigte Software . . . . .	3
<b>2 Petri-Netze</b>	<b>5</b>
2.1 Stellen/Transitions-Netze . . . . .	5
2.1.1 Invarianten . . . . .	7
2.1.1.1 Invariantenberechnung . . . . .	11
2.1.2 Deadlocks und Traps . . . . .	11
2.2 Das allgemeine Erreichbarkeitsproblem in Petri-Netzen . . . . .	12
2.2.1 Erreichbarkeit in S/T-Netzen . . . . .	13
2.2.2 Kriterien zur Vereinfachung des Erreichbarkeitsproblems . . . . .	15
<b>3 Ganzzahlige Lösungen von Gleichungssystemen</b>	<b>17</b>
3.1 Basen und Erzeugendensysteme . . . . .	17
3.1.1 Exkurs: Inhomogene Gleichungssysteme . . . . .	19
3.1.2 Algorithmen . . . . .	20
3.1.2.1 Berechnung eines Level 3-Erzeugendensystems . . . . .	20
3.1.2.2 Berechnung eines Level 4-Erzeugendensystems . . . . .	24
3.2 Ganzzahlige lineare Optimierung . . . . .	27
3.2.1 Algorithmen . . . . .	29
<b>4 Das Erreichbarkeitsproblem und lineare Algebra: Existierende Algorithmen</b>	<b>31</b>
4.1 Das Verfahren von Alexander Kostin . . . . .	31

4.2	Das Verfahren von Ramachandran und Kamath . . . . .	38
<b>5</b>	<b>Verfeinerung des Erreichbarkeitsproblems im Zusammenspiel mit linearer Algebra</b>	<b>43</b>
5.1	Lebendigkeit und Reproduzierbarkeit der leeren Markierung . . . . .	43
5.2	Starke und schwache Erreichbarkeit . . . . .	46
5.3	Nachweis schwacher Erreichbarkeit mittels T-Invarianten . . . . .	47
5.3.1	Verwendung von Erzeugendensystemen . . . . .	47
5.3.2	Lösung einer ganzzahligen linearen Extremwertaufgabe . . . . .	53
5.4	Nachweis starker Erreichbarkeit mittels T-Invarianten . . . . .	54
5.4.1	Verwendung von Erzeugendensystemen . . . . .	55
5.4.2	Lösung einer ganzzahligen linearen Extremwertaufgabe . . . . .	55
5.4.3	Zusammenfassung . . . . .	55
5.5	Zusätzlicher Informationsgewinn . . . . .	56
5.5.1	Ermitteln minimaler Kreise über S-Invarianten . . . . .	59
5.5.2	Umformung in ein zyklensfreies Netz . . . . .	64
5.5.3	Exkurs: Eine „universale“ Markierung . . . . .	68
5.5.3.1	Realisierbarkeit der minimalen T-Invarianten im Originalnetz . . . . .	69
<b>6</b>	<b>Lösungsansatz I: Berechnung eines Synchronisationsgraphen</b>	<b>73</b>
6.1	Auswahl einer geeigneten Linearkombination von T-Invarianten . . . . .	75
6.2	Vom erweiterten Synchronisationsgraphen zum Synchronisationsgraphen	80
6.3	Berechnung der kreisfreien Synchronisationsgraphen . . . . .	88
6.4	Ermitteln und „Verbieten“ der Kreise . . . . .	93
6.5	Zusammenfassung und Fazit . . . . .	98
<b>7</b>	<b>Lösungsansatz II: Entfaltung des Netzes in kreisfreie Schichten</b>	<b>99</b>
7.1	Grundlagen der Entfaltung . . . . .	99
7.2	Entfaltung in ein vollständig kreisfreies Netz . . . . .	104
7.2.1	Nebenbedingungen des ganzzahligen linearen Programms . . . . .	104
7.2.2	Nachweis schwacher Erreichbarkeit . . . . .	105
7.2.3	Nachweis starker Erreichbarkeit . . . . .	109
7.3	Entfaltung mit Erhaltung der Kreise auf der untersten Schicht . . . . .	116
7.3.1	Nebenbedingungen des ganzzahligen linearen Programms . . . . .	116
7.3.2	Nachweis schwacher Erreichbarkeit . . . . .	117
7.3.3	Nachweis starker Erreichbarkeit . . . . .	119
7.4	Bewertung des Lösungsansatzes . . . . .	122

<b>8</b>	<b>Lösungsansatz II: Anwendungsbeispiele</b>	<b>125</b>
8.1	Beispiele für starke Erreichbarkeit . . . . .	125
8.1.1	Netz B1 . . . . .	126
8.1.2	Netz B2 . . . . .	129
8.1.3	Netz B3 . . . . .	135
8.1.4	Netz B4 . . . . .	139
8.2	Beispiele für schwache Erreichbarkeit . . . . .	146
8.2.1	Netz B5 . . . . .	146
8.2.2	Netz B6 . . . . .	150
8.2.3	Netz B7 . . . . .	153
<b>9</b>	<b>Fazit und Ausblick</b>	<b>161</b>
	<b>Literaturverzeichnis</b>	<b>167</b>





# Abbildungsverzeichnis

1.1	Ein einfaches Stellen/Transitions-Netz mit Start- und Zielmarkierung . . . . .	2
2.1	S/T-Netz N1 und zugehörige Inzidenzmatrix [N1] . . . . .	9
2.2	Netz N1: Realisierung der T-Invariante $\vec{y}^2$ . . . . .	10
2.3	S/T-Netz ES1 und zugehörige Inzidenzmatrix [ES1] . . . . .	10
2.4	S/T-Netz N2 und Erreichbarkeit: Die Zustandsgleichung ist kein hinreichendes Kriterium. . . . .	14
2.5	Beschränktheit eines Netzes ohne S-Invarianten . . . . .	16
3.1	Ganzzahlige Lösungen — Gesetz der Komplexität [KJ87] . . . . .	20
3.2	Berechnung des standardisierten $\mathbb{Q}^+$ -Erzeugendensystems für die T-Invarianten des Netzes N1 aus Abbildung 2.1 . . . . .	22
3.3	Berechnung des $\mathbb{N}$ -Erzeugendensystems für die T-Invarianten des Netzes N1 aus Abbildung 2.1 . . . . .	26
4.1	S/T-Netz N3 aus [Kos08] . . . . .	32
4.2	$N3_c$ — um $t_c$ erweitertes Netz N3. . . . .	33
4.3	Minimale T-Invarianten von Netz $N3_c$ mit jeweiligen Netzdarstellungen . . . . .	34
4.4	S/T-Netz N4 . . . . .	38
4.5	Netz N4 erweitert um eine zusätzliche Schicht. . . . .	39
4.6	Netz N4 erweitert um fünf zusätzliche Schichten. . . . .	41
5.1	Netz N5 und erweitertes Netz $N5_{AB}$ . . . . .	44
5.2	Netz N2 – schwache Erreichbarkeit . . . . .	46
5.3	Netz $N6_{AB}$ . . . . .	48
5.4	Netz $N6_{ABC}$ mit Anfangsmarkierung $M6_C$ . . . . .	48
5.5	Netz $N4_{ABC}$ . . . . .	52
5.6	Unzulässige Reduktion der einfachen Stelle $p_5$ . . . . .	57
5.7	Netz $N3_{AB}$ . . . . .	59
5.8	Netz $R3 := N3_{AB}$ ohne einfache Stellen . . . . .	59
5.9	Beispiel: Ersetzen nicht-einfacher Stellen . . . . .	61

5.10	ES3 — erweiterter Synchronisationsgraph zu R3 . . . . .	63
5.11	Nicht-einfache Stellen repräsentierende „Parallelstellen“ . . . . .	66
5.12	$R3^{cf}$ — zyklensfreies Netz R3 . . . . .	68
5.13	S/T-Netz R3 mit universaler Markierung $M3_U$ . . . . .	70
6.1	Netz ES4 . . . . .	76
6.2	Netz $N7_{AB}$ . . . . .	78
6.3	Netz ES7 . . . . .	79
6.4	Beispiel: Behandlung nicht-einfacher Stellen bei der Umwandlung in einen Synchronisationsgraphen . . . . .	83
6.5	Synchronisationsgraph $ES7^*$ unter Einbeziehung der Schaltheufigkeiten von T-Invariante $\vec{r}$ . . . . .	83
6.6	Ersetzen der Transitionen $T_i$ durch Gruppen von Transitionen, darge- stellt für $T_1$ aus Netz $ES7^*$ . . . . .	84
6.7	Pseudo-Synchronisationsgraph PG4 . . . . .	87
6.8	Pseudo-Synchronisationsgraph PG1 . . . . .	92
7.1	Netz $N1 _1$ . . . . .	101
7.2	Netz $N1 _1$ mit hinter Stellen aufgeschnittenen Kanten . . . . .	103
7.3	Zyklensfreies Netz $N1 _1^{cf}$ . . . . .	105
7.4	Erweiterung um eine Schicht: Netz $R2 _1^{cf}$ . . . . .	107
7.5	Erweiterung um 2 Schichten: Netz $R2 _2^{cf}$ . . . . .	108
7.6	S/T-Netz $N6_{AB}$ . . . . .	112
7.7	S/T-Netz $N6_{AB} _4^{cf}$ . . . . .	113
7.8	Netz $R2 _2$ . . . . .	119
7.9	S/T-Netz $N6_{AB} _5$ . . . . .	123
8.1	Netz $B1_{AB}$ . . . . .	126
8.2	Erweiterter Synchronisationsgraph zu Netz $B1_{AB}$ . . . . .	127
8.3	Netz $B1_{AB} _1^{cf}$ . . . . .	128
8.4	Netz $B1_{AB} _1$ . . . . .	129
8.5	Netz $B2_{AB}$ . . . . .	129
8.6	Erweiterter Synchronisationsgraph zu Netz $B2_{AB}$ . . . . .	131
8.7	Netz $B2_{AB} _1^{cf}$ . . . . .	132
8.8	Netz $B2_{AB} _2$ . . . . .	134
8.9	Netz $B3_{AB}$ . . . . .	135
8.10	Erweiterter Synchronisationsgraph zu Netz $B3_{AB}$ . . . . .	135
8.11	Netz $B3_{AB} _4^{cf}$ . . . . .	138

---

8.12	Netz $B_{3AB} _4$ . . . . .	140
8.13	Netz $B_{4AB}$ . . . . .	141
8.14	Erweiterter Synchronisationsgraph zu Netz $B_{4AB}$ . . . . .	141
8.15	Netz $B_{4AB} _2^{cf}$ . . . . .	143
8.16	Netz $B_{4AB} _3$ . . . . .	145
8.17	Netz $B_{5AB}$ . . . . .	146
8.18	Erweiterter Synchronisationsgraph zu Netz $B_{5AB}$ . . . . .	147
8.19	Netz $B_{5AB} _2^{cf}$ . . . . .	148
8.20	Netz $B_{5AB} _2$ . . . . .	149
8.21	Netz $B_{6AB}$ . . . . .	150
8.22	Erweiterter Synchronisationsgraph zu Netz $B_{6AB}$ . . . . .	151
8.23	Netz $B_{6AB} _5^{cf}$ . . . . .	154
8.24	Netz $B_{6AB} _5$ . . . . .	155
8.25	Netz $B_{7AB}$ . . . . .	156
8.26	Netz $B_{7AB} _4^{cf}$ . . . . .	158
8.27	Netz $B_{7AB} _4$ . . . . .	159



# Tabellenverzeichnis

3.1	Basen und Erzeugendensysteme, angelehnt an [KJ87] . . . . .	18
5.1	S–Invarianten des Netzes ES3 . . . . .	63
6.1	“Kritische“ Übergänge repräsentiert durch S–Invarianten in ES, in diesem Fall für ES3 . . . . .	74
6.2	S–Invarianten von ES4 . . . . .	77
6.3	Minimale T–Invarianten von N7 <sub>ABC</sub> . . . . .	79
6.4	Minimale S–Invarianten von ES7 . . . . .	79
6.5	Inzidenzmatrix [PG1*  – $\vec{b}$ ] . . . . .	97
7.1	Lösungen des GLOP für das schichtweise erweiterte Netz N6 <sub>AB</sub> <sup>cf</sup> . . . . .	114
7.2	Lösungen des GLOP für das schichtweise erweiterte Netz R2 mit Kreisen auf Schicht 0 . . . . .	118
7.3	Lösungen des GLOP für das schichtweise erweiterte Netz N6 <sub>AB</sub> . . . . .	121
8.1	Lösungen des GLOP für das schichtweise erweiterte Netz B1 <sub>AB</sub> <sup>cf</sup> . . . . .	128
8.2	Lösungen des GLOP für das schichtweise erweiterte Netz B1 <sub>AB</sub> mit Kreisen auf Schicht 0 . . . . .	129
8.3	Lösungen des GLOP für das schichtweise erweiterte Netz B2 <sub>AB</sub> <sup>cf</sup> . . . . .	133
8.4	Lösungen des GLOP für das schichtweise erweiterte Netz B2 <sub>AB</sub> mit Kreisen auf Schicht 0 . . . . .	133
8.5	Lösungen des GLOP für das schichtweise erweiterte Netz B3 <sub>AB</sub> <sup>cf</sup> . . . . .	137
8.6	Lösungen des GLOP für das schichtweise erweiterte Netz B3 <sub>AB</sub> mit Kreisen auf Schicht 0 . . . . .	139
8.7	Lösungen des GLOP für das schichtweise erweiterte Netz B4 <sub>AB</sub> <sup>cf</sup> . . . . .	144
8.8	Lösungen des GLOP für das schichtweise erweiterte Netz B4 <sub>AB</sub> mit Kreisen auf Schicht 0 . . . . .	144
8.9	Lösungen des GLOP für das schichtweise erweiterte Netz B5 <sub>AB</sub> <sup>cf</sup> . . . . .	148
8.10	Lösungen des GLOP für das schichtweise erweiterte Netz B5 <sub>AB</sub> mit Kreisen auf Schicht 0 . . . . .	149

8.11	Lösungen des GLOP für das schichtweise erweiterte Netz $B6_{AB}^{cf}$	152
8.12	Lösungen des GLOP für das schichtweise erweiterte Netz $B6_{AB}$ mit Kreisen auf Schicht 0	153
8.13	Lösungen des GLOP für das schichtweise erweiterte Netz $B7_{AB}^{cf}$	157
8.14	Lösungen des GLOP für das schichtweise erweiterte Netz $B7_{AB}$ mit Kreisen auf Schicht 0	157

# Algorithmenverzeichnis

3.1	LEVEL 3 — (Minimale Invarianten nach Silva/Martínez) . . . . .	23
3.2	LEVEL 4 — (N-Erzeugendensystem nach Contejean/Devie) . . . . .	25
5.1	Überprüfung auf schwache Erreichbarkeit . . . . .	50
5.2	Entfernen einfacher Stellen . . . . .	58
5.3	Transformation eines S/T-Netzes in einen erweiterten Synchronisationsgraphen . . . . .	62
5.4	Ermitteln der aufzuschneidenden Kanten anhand von S-Invarianten . . . . .	64
5.5	Transformation in ein zyklensches Netz . . . . .	67
5.6	Berechnung einer universalen Markierung $M_U$ . . . . .	70
6.1	Transformation des erweiterten Synchronisationsgraphen in einen Synchronisationsgraphen . . . . .	81
6.2	Transformation in einen Pseudo-Synchronisationsgraphen . . . . .	85
7.1	ADD-LAYER — Hinzufügen einer neuen Schicht . . . . .	100
7.2	Nachweis schwacher Erreichbarkeit im kreisfreien Netz . . . . .	108
7.3	Nachweis starker Erreichbarkeit für die Netze aus dieser Arbeit (kreisfrei auf Schicht 0) . . . . .	115
7.4	Nachweis schwacher Erreichbarkeit mit Kreisen auf Schicht 0 . . . . .	118
7.5	Nachweis starker Erreichbarkeit für die Netze aus dieser Arbeit (mit Kreisen auf Schicht 0) . . . . .	121





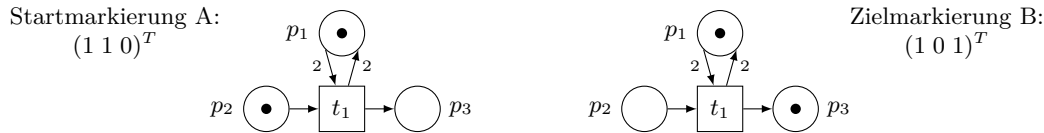
# 1 Einführung

Das allgemeine Erreichbarkeitsproblem in Stellen/Transitions-Netzen behandelt die Frage, ob eine bestimmte Zielmarkierung  $B$  aus einer anderen Markierung  $A$  erreichbar ist. Bei dieser Zielmarkierung kann es sich um einen erwünschten, häufig aber auch um einen strikt zu vermeidenden Zustand eines Netzes handeln. Erst Anfang der 80er Jahre konnte die Entscheidbarkeit dieses Erreichbarkeitsproblems bewiesen werden [Mül85, May81]. Die Entscheidung der Erreichbarkeit und aller Fragen bezüglich des Verhaltens von Petri-Netzen sind EXPSPACE-hart, sie benötigen also mindestens exponentiellen Platzaufwand [Esp98, Lip76]. Kriterien zur Vereinfachung der Erreichbarkeitsfrage und damit des Rechenaufwands für in der Praxis relevante Netzklassen stellen daher seit Jahrzehnten ein großes Forschungsgebiet dar (vgl. [Sta90], S. 52).

Die lineare Algebra als Teilgebiet der Mathematik ist ein sehr elegantes und vielfältig einsetzbares Handwerkszeug. Entscheidende Anstöße, wie sie zudem zur Beschreibung und Analyse von Petri-Netzen herangezogen werden kann, lieferte Kurt Lautenbach mit seiner Dissertation und weiteren Veröffentlichungen (z. B. [Lau73, Lau87, Lau02]). Das dynamische Verhalten von Stellen/Transitions-Systemen und damit auch die Erreichbarkeit einer Markierung ist zum einen von der Struktur des Netzes und zum anderen von der initialen Markierung abhängig. Die Struktur eines allgemeinen Stellen/Transitions-Netzes lässt sich fast exakt durch seine Inzidenzmatrix repräsentieren. Eine Ausnahme bildet eine Null in der Matrix: Sie kann sowohl *keine* Verbindung zwischen einer Stelle  $p_i$  und einer Transition  $t_j$  als auch eine Schlinge  $p_i \rightarrow t_j \rightarrow p_i$  mit gleichen Kantengewichten beschreiben. Netze können also Kreise enthalten, die zunächst in der Inzidenzmatrix und damit in der linearen Algebra nicht erkennbar sind. Während bei kreisfreien Netzen die Zustandsgleichung als hinreichendes Kriterium für Erreichbarkeit herangezogen werden kann, ist sie bei Netzen *mit* Kreisen nur ein notwendiges Kriterium: Abbildung 1.1 zeigt ein S/T-Netz mit Startmarkierung  $A$  und Zielmarkierung  $B$ . Für die Zustandsgleichung, hier

$$\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \cdot (t_1) = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix},$$

existiert zwar mit  $t_1 = 1$  eine nichtnegative Lösung, jedoch ist  $t_1$  bei gegebener Startmarkierung  $A$  nicht aktiviert und damit auch die Zielmarkierung  $B$  niemals erreichbar. Kreise im Netz stellen also ein Problem bei linear-algebraischen Erreichbarkeitsuntersuchungen in Stellen/Transitions-Netzen dar.



**Abbildung 1.1:** Ein einfaches Stellen/Transitions-Netz mit Start- und Zielmarkierung

## 1.1 Ziel und Aufbau der Arbeit

Gibt es trotz dieser Hürde rein linear-algebraische Verfahren zur Entscheidung des Erreichbarkeitsproblems in Stellen/Transitions-Netzen? Kommt man ohne Simulationsschritte oder Berechnung des Erreichbarkeitsgraphen aus? In dieser Arbeit werden bereits existierende Verfahren diskutiert sowie diese Fragestellung auf Basis von T-Invarianten in Kombination mit der Reproduzierbarkeit der leeren Markierung [Lau02] und struktureller Eigenschaften bestimmter Netzklassen neu untersucht. Der Fokus liegt hierbei auf der theoretischen Machbarkeit und nicht auf der Performance der vorgestellten Verfahren, zumal in der ganzzahligen linearen Algebra und ganzzahligen Programmierung bei sehr großen Netzen in der Regel mit exponentiellem Rechenaufwand zu rechnen ist.

Die vorliegende Arbeit ist folgendermaßen gegliedert: Alle notwendigen Grundlagen und Begriffe aus der Petri-Netz-Theorie, der ganzzahligen linearen Algebra und ganzzahligen linearen Optimierung sind in den Kapiteln 2 und 3 zusammengestellt. Insbesondere in Kapitel 3 wurde Wert darauf gelegt, verwendete Algorithmen genauer zu erklären, da sie in späteren Kapiteln wieder aufgegriffen und modifiziert werden. Bisherige Ansätze zur Lösung des Erreichbarkeitsproblems stehen in Kapitel 4 zur Diskussion. Anschließend wird in Kapitel 5 eine von Kurt Lautenbach entwickelte Verfeinerung des Erreichbarkeitsproblems in zwei Problemklassen, aufbauend auf der bereits erwähnten Reproduzierbarkeit der leeren Markierung, vorgestellt. Zwei neue, auf dieser Idee basierende Lösungsansätze werden in den Kapiteln 6 und 7 diskutiert. Für den Lösungsansatz II enthält Kapitel 8 ausführliche Beispiele. Die Ergebnisse sind abschließend in Kapitel 9 zusammengefasst.

## 1.2 Benötigte Software

Bei dieser Arbeit handelt es sich um eine theoretische Arbeit. Dennoch mussten häufig lineare Gleichungssysteme und ganzzahlige lineare Optimierungsprobleme gelöst und damit auf geeignete Software zurückgegriffen werden. Sofern nicht anders angegeben, wurden alle Stellen/Transitions-Netze im Petri-Netz-Tool „NeMo“ der Arbeitsgruppe Lautenbach modelliert und minimale T- und S-Invarianten, Deadlocks und Traps mit dem Tool berechnet. Modifizierte Algorithmen wurden in Java vom Autor implementiert bzw. aus dem „NeMo“-Quellcode extrahiert und geringfügig verändert. Für die Lösung ganzzahliger Optimierungsprobleme hat sich das Programmpaket von Michael Gensty [Gen05] bewährt. Die Ergebnisse wurden mit dem frei verfügbaren Tool *lp\_solve* [BEN11] gegengetestet.



## 2 Petri–Netze

Prof. Dr. Carl Adam Petri entwickelte die nach ihm benannten *Petri–Netze* zur Modellierung verteilter Systeme und lieferte mit seiner Dissertation im Jahre 1962 „entscheidende Anstöße zur weiteren Entwicklung und Anwendung der Netztheorie“ ([Bau96], S. 14).

Es gibt mittlerweile eine Reihe verschiedener Netzklassen, die sich in ihrer Ausdrucksmächtigkeit unterscheiden. In dieser Arbeit werden nur die Stellen/Transitions–Netze, die einfachste Netzklasse, untersucht. Es wird davon ausgegangen, dass der Leser<sup>1</sup> bereits mit Petri–Netzen vertraut ist. Darum werden in den folgenden Abschnitten nur die für diese Arbeit notwendigen netztheoretischen Begriffe und Schreibweisen definiert. Eine gute Einführung in die Grundlagen der Petri–Netz–Theorie bietet zum Beispiel [Bau96].

### 2.1 Stellen/Transitions–Netze

Die folgenden formalen Definitionen für Stellen/Transitions–Netze orientieren sich am Skript zur Vorlesung „Duality and Diagnosis in Higher Level Petri Nets“ von Prof. Dr. Kurt Lautenbach (vgl. [Lau05]); es erfolgte in den meisten Fällen lediglich eine Übersetzung ins Deutsche.

**Definition 2.1 (Stellen/Transitions–Netz)** *Ein Stellen/Transitions–Netz, nachfolgend S/T–Netz genannt, ist ein Vier–Tupel  $N = (S, T, F, W)$  mit:*

- $S$  und  $T$  sind endliche, nicht–leere und disjunkte Mengen,  
 $S$  ist die Menge der Stellen,  
 $T$  ist die Menge der Transitionen.
- $F \subseteq (S \times T) \cup (T \times S)$  ist die Menge der gerichteten Kanten.
- $W : F \rightarrow \mathbb{N}$  weist jeder Kante ein Gewicht zu.  
Falls  $W : F \rightarrow \{1\}$  gilt, wird  $N = (S, T, F)$  als Abkürzung verwendet. ■

---

<sup>1</sup>Aus Gründen der leichteren Lesbarkeit wird durchgehend die männliche Formulierung verwendet.

**Definition 2.2 (Vorbereich und Nachbereich)** Der Vorbereich eines Knotens  $x \in S \cup T$  ist definiert als

$$\bullet x := \{y \in S \cup T \mid (y, x) \in F\},$$

der Nachbereich als

$$x^\bullet := \{y \in S \cup T \mid (x, y) \in F\}. \quad \blacksquare$$

**Definition 2.3 (Markierung, markiert, unmarkiert, leere Markierung)** Sei  $N = (S, T, F, W)$  ein  $S/T$ -Netz.

- Eine Markierung von  $N$  ist eine Abbildung  $M : S \rightarrow \mathbb{N}_0$ .
- $M(p)$  gibt die Anzahl der Marken auf Stelle  $p$  unter Markierung  $M$  an.
- $p \in S$  ist markiert durch  $M \Leftrightarrow M(p) \geq 1$ .
- $H \subseteq S$  ist markiert durch  $M \Leftrightarrow$  Mindestens ein  $p \in H$  ist durch  $M$  markiert.
- Andernfalls heißen  $p$  bzw.  $H$  unmarkiert.
- Im Fall  $M(p) = 0 \ \forall p \in S$  liegt die leere Markierung vor. ■

**Definition 2.4 (aktiviert, schalten, neue Markierung)** Eine Transition  $t \in T$  ist genau dann aktiviert durch  $M$  (geschrieben  $M[t)$ ), wenn

$$\forall p \in \bullet t : M(p) \geq W((p, t)).$$

Gilt  $M[t)$ , so kann  $t$  schalten. Dies resultiert in einer neuen Markierung  $M'$  ( $M[t)M'$ ) mit

$$M'(p) := \begin{cases} M(p) - W((p, t)) & \text{falls } p \in \bullet t \setminus t^\bullet \\ M(p) + W((t, p)) & \text{falls } p \in t^\bullet \setminus \bullet t \\ M(p) - W((p, t)) + W((t, p)) & \text{falls } p \in t^\bullet \cap \bullet t \\ M(p) & \text{sonst} \end{cases}$$

für alle  $p \in S$ . ■

**Definition 2.5 (erreichbare Markierung, Folgemarkierung)** Die Menge  $[M_0\rangle$  aller Markierungen, die von einer (Anfangs-)Markierung  $M_0$  aus erreichbar sind, ist definiert durch

$$\begin{aligned} M_0 &\in [M_0\rangle \\ M &\in [M_0\rangle \wedge M[t)M' \Rightarrow M' \in [M_0\rangle. \end{aligned}$$

$[M_0\rangle$  nennt man auch die Folgemarkierungen von  $M_0$ . ■

**Definition 2.6 (Schaltsequenz, Schalthäufigkeit, S/T-System)** Für Transitionen  $t_1, \dots, t_n \in T$  ist  $\sigma = t_1 \dots t_n$  genau dann eine Schaltsequenz, wenn Markierungen  $M_0, M_1, \dots, M_n$  existieren, so dass gilt:

$$M_0[t_1]M_1[t_2] \dots [t_n]M_n \text{ (kurz: } M_0[\sigma]M_n\text{)}.$$

$M_0[\sigma]$  besagt, dass  $\sigma$  von  $M_0$  aus startet. Die Schalthäufigkeit  $\bar{\sigma}(t)$  von  $t$  gibt an, wie oft  $t$  in  $\sigma$  vorkommt. Für eine Anfangsmarkierung  $M_0$  von  $N$  beschreibt das Paar  $(N, M_0)$  ein S/T-System bzw. ein markiertes S/T-Netz. ■

Eine besondere Schaltsequenz ist  $M[\sigma]M$ , die die Markierung  $M$  reproduziert. Diese Schaltsequenz nennt man *Transitions-Invariante*. Der folgende Abschnitt beschäftigt sich mit diesen und den *Stellen-Invarianten*.

### 2.1.1 Invarianten

Ein S/T-Netz kann durch seine sogenannte *Inzidenzmatrix*  $[N]$  beschrieben werden. Die Zeilen der Matrix repräsentieren die Stellen, die Spalten repräsentieren die Transitionen des Netzes. Ein Eintrag in Zeile  $i$  und Spalte  $j$  in der Matrix beschreibt, wie sich die Anzahl der Marken auf Stelle  $p_i$  durch das Schalten von Transition  $t_j$  ändert.

**Definition 2.7 (Inzidenzmatrix)** Die Inzidenzmatrix eines S/T-Netzes  $N$  ist eine Matrix  $[N] : (S \times T) \rightarrow \mathbb{Z}$  geordnet nach  $S$  und  $T$  mit

$$[N]_{p,t} := \begin{cases} -W((p, t)) & \text{falls } p \in \bullet t \setminus t \bullet \\ W((t, p)) & \text{falls } p \in t \bullet \setminus \bullet t \\ -W((p, t)) + W((t, p)) & \text{falls } p \in t \bullet \cap \bullet t \\ 0 & \text{sonst} \end{cases}$$

Neben der Darstellung des gesamten Netzes als Matrix können die Stellen- und Transitionsmengen auch als Vektoren notiert werden:

**Definition 2.8 (Stellenvektor, Transitionsvektor)**

- Ein Stellenvektor ist ein Spaltenvektor  $\vec{v} : S \rightarrow \mathbb{Z}$  geordnet nach  $S$ .  $\vec{v}(p)$  bezeichnet die Markenzahl auf Stelle  $p \in S$  in  $\vec{v}$ .
- Ein Transitionsvektor ist ein Spaltenvektor  $\vec{w} : T \rightarrow \mathbb{Z}$  geordnet nach  $T$ .  $\vec{w}(t)$  bezeichnet die Schalthäufigkeit von Transition  $t \in T$  in  $\vec{w}$ .

- Markierungen werden als Stellenvektoren dargestellt.
- Schalthäufigkeitsvektoren werden durch Transitionsvektoren beschrieben.
- $\vec{0}$  ( $\vec{1}$ ) ist der Spaltenvektor, dessen Einträge alle 0 (1) sind. ■

Zwei Vektoren lassen sich komponentenweise vergleichen und damit in eine Ordnung eingliedern. Hierzu führen wir<sup>2</sup> Vergleichssymbole ein:

**Definition 2.9 (größer als ( $\succ$ ), kleiner als ( $\prec$ ), nicht vergleichbar)** Seien  $\vec{a} = (a_1 \dots a_n)^T$  und  $\vec{b} = (b_1 \dots b_n)^T$  zwei Vektoren aus  $\mathbb{N}_0^n$ .  $\vec{a}$  ist größer als  $\vec{b}$ , geschrieben  $\vec{a} \succ \vec{b}$ , wenn gilt:

$$\forall i \in [1, \dots, n] : a_i \geq b_i \text{ und } \exists i \in [1, \dots, n] : a_i \neq b_i.$$

Damit ist auch  $\vec{b}$  kleiner als  $\vec{a}$ , geschrieben  $\vec{b} \prec \vec{a}$ .

Die Vektoren  $\vec{a}$  und  $\vec{b}$  sind nicht vergleichbar gemäß dieser Ordnung  $\succ$ , falls gilt:

$$\exists i \in [1, \dots, n] : a_i < b_i \text{ und } \exists j \neq i \in [1, \dots, n] : a_j > b_j.$$

Falls zwei Vektoren in allen Komponenten gleich sind, schreiben wir wie gewohnt  $\vec{a} = \vec{b}$ , bei Ungleichheit  $\vec{a} \neq \vec{b}$ . ■

Im Folgenden werden nun die sogenannten Stellen- und Transitions-Invarianten vorgestellt, die Vektoren gemäß Definition 2.8 mit speziellen Eigenschaften sind.

**Definition 2.10 (S-Invariante, T-Invariante, Trägermenge, überdeckt)** Sei  $\vec{x}$  ein Stellenvektor,  $\vec{y}$  ein Transitionsvektor von  $N$ .

- $\vec{x}$  ist eine Stellen-Invariante (S-Invariante)  $\Leftrightarrow \vec{x} \neq \vec{0}$  und  $\vec{x}^T[N] = \vec{0}^T$ .
- $\vec{y}$  ist eine Transitions-Invariante (T-Invariante)  $\Leftrightarrow \vec{y} \neq \vec{0}$  und  $[N]\vec{y} = \vec{0}$ .

Die Trägermenge (engl. Support)  $\|\vec{x}\|$  ( $\|\vec{y}\|$ ) einer S-Invariante (T-Invariante)  $\vec{x}$  ( $\vec{y}$ ) ist definiert als  $\|\vec{x}\| = \{p \in S \mid \vec{x}(p) \neq 0\}$  ( $\|\vec{y}\| = \{t \in T \mid \vec{y}(t) \neq 0\}$ ).

- $N$  ist von  $\vec{x}$  ( $\vec{y}$ ) überdeckt  $\Leftrightarrow \|\vec{x}\| = S$  ( $\|\vec{y}\| = T$ ).

Die Trägermenge einer Invariante ist genau dann minimal, wenn sie keine andere Trägermenge einer Invariante beinhaltet außer sich selbst und der leeren Menge. ■

<sup>2</sup>Hierbei ist stets der Autor zusammen mit dem Leser gemeint.



Weiterhin sind folgende Eigenschaften für Invarianten definiert:

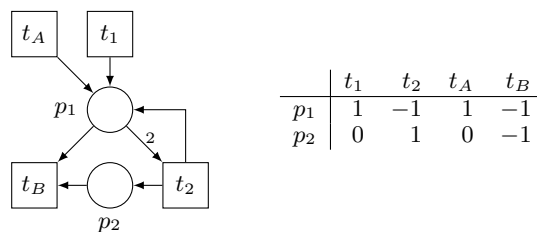
**Definition 2.11 (nichtnegativ, kanonisch, minimal)** Sei  $\vec{x}$  eine S-Invariante und sei  $\vec{y}$  eine T-Invariante von  $N$ .

- $\vec{x}$  ( $\vec{y}$ ) ist nichtnegativ  $\Leftrightarrow \forall p \in S : \vec{x}(p) \geq 0$  ( $\forall t \in T : \vec{y}(t) \geq 0$ ).
- $\vec{x}$  ( $\vec{y}$ ) ist kanonisch  $\Leftrightarrow$  der größte gemeinsame Teiler aller Komponenten von  $\vec{x}$  ( $\vec{y}$ ) ist 1.
- $\vec{x}$  ( $\vec{y}$ ) ist minimal  $\Leftrightarrow \vec{x}$  ( $\vec{y}$ ) ist kanonisch mit minimalem Support. ■

T-Invarianten sind Häufigkeitsvektoren für Schaltsequenzen, nach deren Durchführung die Ausgangsmarkierung wieder erreicht wird. Ob eine T-Invariante eine Markierung reproduziert, hängt von dieser Markierung ab.

**Definition 2.12 (realisierbar)** Sei  $\vec{y}$  eine nichtnegative T-Invariante von  $N$ .  $\vec{y}$  heißt realisierbar in  $N$ , wenn es eine Schaltsequenz  $\sigma$  mit  $\bar{\sigma} = \vec{y}$  und eine in  $N$  erreichbare Markierung  $M$  mit  $M[\sigma]M$  gibt (vgl. [Sta90], S. 112). ■

**Beispiel 2.1 (T-Invariante)** Abbildung 2.1 zeigt ein einfaches S/T-Netz mit zugehöriger Inzidenzmatrix. Das Netz hat die nichtnegativen, minimalen T-Invarianten



**Abbildung 2.1:** S/T-Netz  $N_1$  und zugehörige Inzidenzmatrix  $[N_1]$

$\vec{y}^1 = (2 \ 1 \ 0 \ 1)^T$ ,  $\vec{y}^2 = (0 \ 1 \ 2 \ 1)^T$ . Für die Invariante  $\vec{y}^2$  ist in Abbildung 2.2 eine mögliche Schaltfolge unter der leeren Markierung als Startmarkierung dargestellt. Offensichtlich wird die leere Markierung durch die Schaltfolge  $\sigma = t_A t_A t_2 t_B$  reproduziert. Für das gegebene Netz  $N_1$  sind beide minimalen T-Invarianten unter der leeren Markierung als Startmarkierung realisierbar, ebenso auch  $\vec{y}^3 = (1 \ 1 \ 1 \ 1)^T$ , die um den größten gemeinsamen Teiler von 2 reduzierte Summe von  $\vec{y}^1$  und  $\vec{y}^2$ . Damit ist  $\vec{y}^3$  kanonisch, aber nicht minimal, da die Trägermenge nicht minimal ist. □

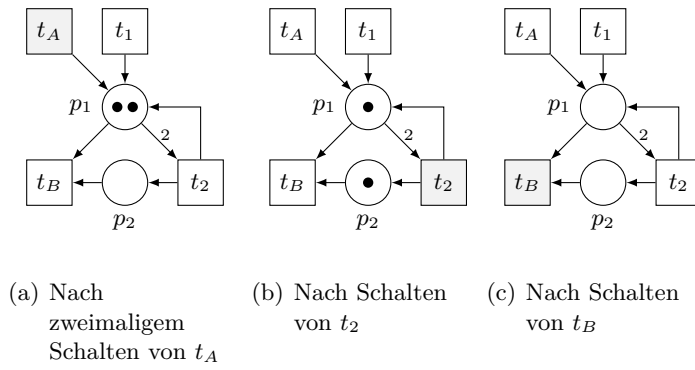


Abbildung 2.2: Netz N1: Realisierung der T-Invariante  $\vec{y}^2$

Die S-Invarianten beziehen sich auf die Anzahl der Marken auf bestimmten Stellenmengen:

**Satz 2.1** Sei  $(N, M_0)$  ein S/T-System und  $\vec{x}$  eine S-Invariante. Dann gilt

- $\forall M \in [M_0] : \vec{x} \circ M = \vec{x} \circ M_0$
- $\forall M : (\vec{x} \circ M \neq \vec{x} \circ M_0) \Rightarrow M \notin [M_0]$  ■

Anschaulich ausgedrückt bedeutet dies, dass die gewichtete Markensumme auf den in der Trägermenge angegebenen Stellen unabhängig von allen Schaltvorgängen konstant bleibt. Diese Gewichtung erhält man durch die Skalarmultiplikation der Anfangsmarkierung mit der S-Invariante.

**Beispiel 2.2 (S-Invariante)** Das in Abbildung 2.3 dargestellte Netz besitzt die nichtnegative S-Invariante  $\vec{x}^1 = (0 \ 0 \ 2 \ 0 \ 1 \ 0)^T$  mit Support  $||\vec{x}^1|| = \{p_{1\triangleleft 2}, p_{1\triangleright 2}\}$ . Für

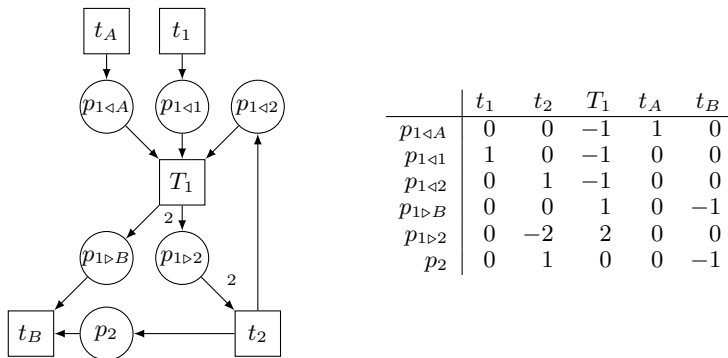


Abbildung 2.3: S/T-Netz ES1 und zugehörige Inzidenzmatrix [ES1]

die Anfangsmarkierung  $(1\ 1\ 1\ 0\ 0\ 0)^T$  erhält man in Kombination mit der S-Invariante  $\vec{x}^1$  eine Summe von  $(1\ 1\ 1\ 0\ 0\ 0)^T \circ (0\ 0\ 2\ 0\ 1\ 0)^T = 2$ . Insgesamt werden auf den Stellen  $\{p_{1 \leftarrow 2}, p_{1 \rightarrow 2}\}$  bei allen folgenden Schaltvorgängen also nie mehr als zwei Marken liegen. Für eine andere Anfangsmarkierung ergibt sich in der Regel auch eine andere gewichtete Markensumme.  $\square$

T- und S-Invarianten können selbst als Petri-Netz dargestellt werden. Diese sogenannte *Netzdarstellung* enthält aber nur einen Teil der Transitionen bzw. Stellen des ursprünglichen Netzes  $N$ :

**Definition 2.13 (Netzdarstellung der Invarianten)** Die Netzdarstellung  $N_{\vec{x}} = (S_{\vec{x}}, T_{\vec{x}}, F_{\vec{x}}, W_{\vec{x}})$  einer S-Invariante  $\vec{x}$  eines Netzes  $N$  ist definiert durch

- $S_{\vec{x}} := \|\vec{x}\|$
- $T_{\vec{x}} := \bullet S_{\vec{x}} \cup S_{\vec{x}} \bullet$
- $F_{\vec{x}} := F \cap ((S_{\vec{x}} \times T_{\vec{x}}) \cup (T_{\vec{x}} \times S_{\vec{x}}))$
- $W_{\vec{x}}$  ist die Beschränkung von  $W$  auf  $F_{\vec{x}}$ .

Die Netzdarstellung  $N_{\vec{y}} = (S_{\vec{y}}, T_{\vec{y}}, F_{\vec{y}}, W_{\vec{y}})$  einer T-Invariante  $\vec{y}$  eines Netzes  $N$  ist definiert durch

- $T_{\vec{y}} := \|\vec{y}\|$
- $S_{\vec{y}} := \bullet T_{\vec{y}} \cup T_{\vec{y}} \bullet$
- $F_{\vec{y}} := F \cap ((S_{\vec{y}} \times T_{\vec{y}}) \cup (T_{\vec{y}} \times S_{\vec{y}}))$
- $W_{\vec{y}}$  ist die Beschränkung von  $W$  auf  $F_{\vec{y}}$ . ■

### 2.1.1.1 Invariantenberechnung

Die Invarianten eines Stellen/Transitions-Netzes lassen sich linear-algebraisch auf Basis der in Definition 2.10 gegebenen Gleichungen berechnen. Abhängig von den gewünschten Eigenschaften, z. B. *nichtnegativ*, *kanonisch* und *minimal*, lassen sich diese klassifizieren und unterschiedliche Algorithmen kommen zur Anwendung. Eine Auswahl wird in Abschnitt 3.1.2 vorgestellt, da die Invariantenberechnung einen wichtigen Teil dieser Arbeit ausmacht.

### 2.1.2 Deadlocks und Traps

Neben den T- und S-Invarianten gibt es noch weitere Netzeigenschaften, die einen Zusammenhang zwischen *Struktur* und *Dynamik* herstellen. Sowohl Traps als auch

Deadlocks werden zunächst als *strukturelle* Eigenschaften eines S/T-Netzes  $N$  definiert und in dieser Arbeit nach dem Verfahren von Ezpeleta/Couvreur/Silva [ECS93] berechnet:

**Definition 2.14 (Trap, Deadlock)** Sei  $N = (S, T, F, W)$  ein S/T-Netz.

Eine nichtleere Stellenmenge  $H \subseteq S$  ist ein Trap  $\Leftrightarrow H^\bullet \subseteq \bullet H$ .

Eine nichtleere Stellenmenge  $H \subseteq S$  ist ein Deadlock  $\Leftrightarrow \bullet H \subseteq H^\bullet$ .

Ist ein Trap markiert, so bleibt er markiert.

Ist ein Deadlock unmarkiert, so bleibt er unmarkiert.

Die Trägermengen aller nichtnegativen  $S$ -Invarianten  $\vec{i}$  sind sowohl Traps als auch Deadlocks. ■

Erst bei einer gegebenen Markierung lassen sich Aussagen darüber treffen, ob das Vorhandensein von Deadlocks und Traps Auswirkungen auf das *dynamische* Verhalten des Netzes und damit auch auf die Erreichbarkeit einer Markierung  $M_B$  aus einer anderen Markierung  $M_A$  hat. Die Zusammenhänge werden in Kapitel 5.1 genauer erläutert und bilden die Basis für alle darauf folgenden Erreichbarkeitsuntersuchungen. Zunächst wird nun das allgemeine Erreichbarkeitsproblem definiert.

## 2.2 Das allgemeine Erreichbarkeitsproblem in Petri-Netzen

Die Frage nach der Erreichbarkeit einer Markierung  $M$  in einem Petri-Netz  $N$  ist gleichzustellen mit der Frage, ob das Netz einen ganz bestimmten Zustand annehmen kann (vgl. [Sta90], S. 43). Beispielsweise könnte es ein Zustand des Systems sein, der unbedingt erreicht oder unbedingt vermieden werden soll. Dieses *allgemeine Erreichbarkeitsproblem* verlangt also nach einem Algorithmus, der die Fragestellung

“Ist im Petri-Netz  $N$  eine bestimmte Markierung  $M$  erreichbar?”

beantwortet. Für beschränkte Netze  $N$  kann die Frage durch Konstruktion des zugehörigen Erreichbarkeitsgraphen entschieden werden. *Beschränkt* bedeutet, dass bei gegebener Anfangsmarkierung in jeder Folgemarkierung die Markenzahl auf jeder Stelle eine obere Grenze  $k$  nie überschreitet. Der Erreichbarkeitsgraph ist in diesem Fall endlich und die Markierung  $M$  kann darin gesucht werden. Der Berechnungsaufwand ist dennoch nicht zu unterschätzen:

“Die Zeitkompliziertheit des Problems, den Erreichbarkeitsgraphen eines beschränkten Petri-Netzes zu berechnen, ist überexponentiell, d.h. diese Funktion wächst schneller als jede Exponentialfunktion. [...] Man kann zeigen, daß das Erreichbarkeitsproblem für beschränkte Petri-Netze die gleiche Kompliziertheit hat wie das Problem, den Erreichbarkeitsgraphen zu berechnen, also ebenfalls überexponentiell ist. Das impliziert, daß Algorithmen, die auf einer Lösung von Erreichbarkeitsproblemen beruhen, nicht effizient, d.h. in vielen praktischen Fällen nicht anwendbar sind. Damit stellt sich die Aufgabe, nach Kriterien für die Erreichbarkeit von Markierungen zu suchen, die einfacher zu verifizieren sind, dafür nur für eingeschränkte aber praktisch relevante Netzklassen gültig sind.” — [Sta90], S. 51f.

Die Existenz eines Algorithmus, der die Erreichbarkeitsfrage für beliebige S/T-Netze beantwortet, wurde erst in den 80er Jahren nachgewiesen [Mül85, May81]. Im angeführten Zitat wird hervorgehoben, dass der Rechenaufwand für gewisse Netzklassen mit bestimmten Eigenschaften eingeschränkt werden kann. Wie sieht dies im Fall der S/T-Netze aus? Stellen/Transitions-Netze und ihre Dynamik sind leicht linear-algebraisch darstellbar. Aber gibt es auch rein linear-algebraische Berechnungen zur Lösung des Erreichbarkeitsproblems? Auf Besonderheiten dieser Frage in S/T-Netzen geht der folgende Abschnitt ein. Der Spezialfall der beschränkten S/T-Netze wird in Kapitel 2.2.2 behandelt.

### 2.2.1 Erreichbarkeit in S/T-Netzen

Bereits in Kapitel 2.1.1 wurde gezeigt, wie ein S/T-Netz linear-algebraisch in Form seiner Inzidenzmatrix dargestellt werden kann. Auch S- und T-Invarianten und Markierungen lassen sich als Vektoren schreiben. Geht eine Markierung durch eine Schaltsequenz aus einer anderen hervor, so gilt die folgende *Zustandsgleichung*:

**Satz 2.2** Sei  $N$  ein S/T-Netz,  $M_1$  und  $M_2$  Markierungen und  $\sigma$  eine Schaltsequenz von  $N$ , so dass  $M_1[\sigma\rangle M_2$ . Dann gilt die Zustandsgleichung

$$M_1 + [N] \cdot \bar{\sigma} = M_2.$$

Die Reihenfolge, in der die Transitionen schalten, ist in  $\bar{\sigma}$  nicht erkennbar. ■

Ist eine Markierung  $M_2$  aus einer Markierung  $M_1$  erreichbar, so muss es demnach einen nichtnegativen Schalthäufigkeitsvektor geben, durch den die Zustandsgleichung erfüllt

ist. Hinreichend ist dieses Kriterium jedoch nicht, wie Abbildung 2.4 zeigt: Der Vektor  $\bar{\sigma} = (1\ 1\ 1\ 1\ 1)^T$  erfüllt die Zustandsgleichung, jedoch gelangt man durch je einmaliges Schalten aller Transitionen nicht von Markierung  $M_{2A}$  nach  $M_{2B}$ ; bereits nach dem Schalten von  $t_4$  oder  $t_5$ , die unter  $M_{2A}$  feuern können, ist keine weitere Transition mehr aktiviert. Die aktuelle Markierung ist dann *tot*, ebenso z. B.  $t_3$  in  $(N_2, M_{2A})$ .

**Definition 2.15 (lebendig, schwach lebendig, tot)** Sei  $(N, M_0)$  ein S/T-System;

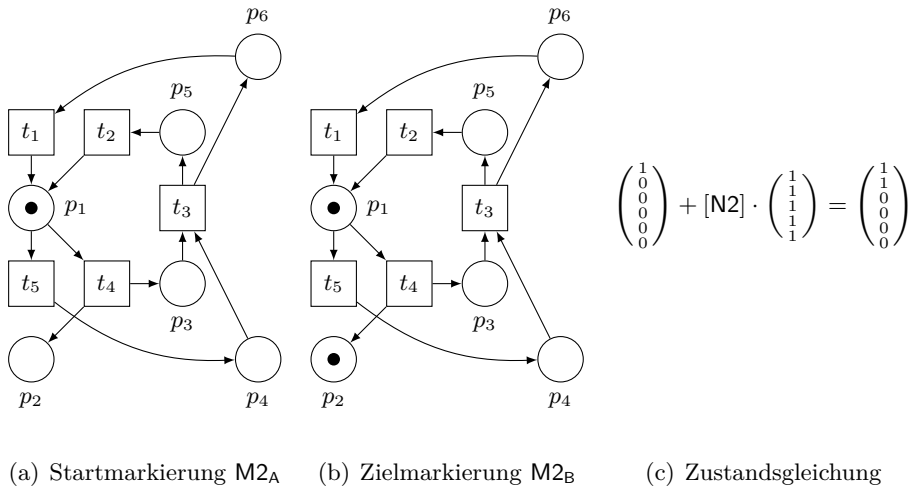
eine Transition  $t$  ist lebendig in  $M_0$  oder  $(N, M_0) \Leftrightarrow \forall M \in [M_0] \exists M' \in [M] : M'[t]$

eine Transition  $t$  ist tot in  $M_0$  oder  $(N, M_0) \Leftrightarrow \forall M \in [M_0] : \neg M[t]$

$(N, M_0)$  oder  $M_0$  ist tot  $\Leftrightarrow \neg \exists t \in T : M_0[t]$

$(N, M_0)$  oder  $M_0$  ist schwach lebendig  $\Leftrightarrow \forall M \in [M_0] \exists t \in T : M[t]$

$(N, M_0)$  oder  $M_0$  ist lebendig  $\Leftrightarrow \forall t \in T : t$  ist lebendig in  $M_0$



**Abbildung 2.4:** S/T-Netz  $N_2$  und Erreichbarkeit: Die Zustandsgleichung ist kein hinreichendes Kriterium.

Der Grund sind die Kreise im Netz: Der Schalthäufigkeitsvektor sagt, wie bereits erwähnt, nichts über die Reihenfolge aus, in der die Transitionen schalten müssten. Im Netz  $N_2$  können  $t_1$  und  $t_2$  unter der Startmarkierung nicht feuern und darum nicht ausreichend Marken auf  $p_1$  zur Verfügung stellen, um sowohl  $t_4$  als auch  $t_5$  zu aktivieren. Dieses Problem wird nicht erkannt, da alle Transitionen des Netzes auf Kreisen liegen. Mit der Zustandsgleichung kann hier also weder ausgedrückt noch erkannt werden, ob die Marken *vor* dem Durchlaufen des Kreises zur Verfügung stehen oder erst

beim Durchlaufen erzeugt werden. Am offensichtlichsten wird dies bereits in der Inzidenzmatrix durch eine Null ausgedrückt: Diese kann einerseits auf *keine* Verbindung zwischen einer Stelle und Transition hinweisen, andererseits aber auch eine *Schlinge*<sup>3</sup> beschreiben. Die Transition  $t_1$  aus Abbildung 1.1 von Seite 2 beschreibt einen solchen Fall. Sie müsste sich die von ihr selbst produzierten zwei Marken leihen, um überhaupt aktiviert zu sein. Dies kann die lineare Algebra nicht ausdrücken.

### 2.2.2 Kriterien zur Vereinfachung des Erreichbarkeitsproblems

Neben der Zustandsgleichung gibt es weitere Eigenschaften eines Netzes, durch die das Erreichbarkeitsproblem vereinfacht werden kann. Die Zustandsgleichung kann als „Schnelltest“ betrachtet werden; existiert keine nichtnegative Lösung  $\vec{\sigma}$  für die Gleichung

$$M_1 + [N] \cdot \vec{\sigma} - M_2 = \vec{0},$$

so kann in keinem Fall Erreichbarkeit vorliegen. Mit Hilfe von S-Invarianten lässt sich die Beschränktheit einzelner Stellen(-mengen) oder des gesamten Netzes nachweisen. Im letzteren Fall ist die Endlichkeit des Erreichbarkeitsgraphen bewiesen — eine wichtige Erkenntnis, wenn er zur Klärung der Erreichbarkeitsfrage konstruiert werden soll.

**Definition 2.16 (beschränkt)** Sei  $(N, M_0)$  ein S/T-System,  $k \in \mathbb{N}$ ;

<i>eine Stelle</i> $p \in S$ <i>ist beschränkt in</i> $M_0$	$\Leftrightarrow \forall M \in [M_0] : M(p) \leq k$
$(N, M_0)$ <i>oder</i> $M_0$ <i>ist beschränkt</i>	$\Leftrightarrow \forall p \in S : p$ <i>ist beschränkt in</i> $M_0$
$N$ <i>heißt strukturell beschränkt</i>	$\Leftrightarrow N$ <i>ist in allen Anfangsmarkierungen beschränkt</i> ■

**Satz 2.3** Sei  $N$  ein S/T-Netz und  $\vec{i} \in \mathbb{N}^n$  (d.h.  $\forall p \in S : \vec{i}(p) > 0$ ) ein positiver Stellenvektor, so dass  $[N]^T \cdot \vec{i} \leq \vec{0}$ ). Dann ist  $N$  strukturell beschränkt [Mur89]. ■

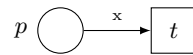
**Satz 2.4** Sei  $N$  ein S/T-Netz. Ist  $N$  von S-Invarianten überdeckt, so ist es strukturell beschränkt. ■

S-Invarianten beschreiben gemäß Satz 2.1 Stellenmengen, auf denen die gewichtete Markensumme unabhängig von der aktuellen Markierung konstant ist. Eine maximale

<sup>3</sup>Eine Schlinge ist ein Teilnetz, das aus einer Stelle, einer Transition und einer Kante von der Stelle zur Transition und zurück besteht. Ein Netz ohne Schlingen heißt *schlingenf*rei oder *rein*.

Obergrenze  $k$  für alle beschränkten Stellen — falls die S-Invariante überdeckend ist — entspricht damit dieser Markensumme, berechnet mit Hilfe der Anfangsmarkierung. Auch S-Invarianten sind damit eine Art „Schnelltest“, bevor der Erreichbarkeitsgraph berechnet wird. Die Umkehrung von Satz 2.4 gilt allerdings nicht, wie das folgende Beispiel zeigt:

**Beispiel 2.3 (Beschränktheit ohne S-Invarianten)** Das Netz in Abbildung 2.5



**Abbildung 2.5:** Beschränktheit eines Netzes ohne S-Invarianten

ist strukturell beschränkt, da unter Startmarkierung  $M_0$  die Transition  $t$  nur genau so oft schalten kann, wie  $M(p) \geq x$  gilt. Das Netz besitzt jedoch keine S-Invarianten.  $\square$

Neben diesen linear-algebraisch überprüfbaren Kriterien haben, wie bereits erwähnt, bestimmte Netzklassen noch Eigenschaften, die das Erreichbarkeitsproblem vereinfachen können. Insbesondere ist der Zusammenhang zwischen Lebendigkeit und Erreichbarkeit von großer Bedeutung. Dieses Thema wird in Kapitel 5 wieder aufgegriffen.



# 3 Ganzzahlige Lösungen von Gleichungssystemen

In dieser Arbeit ist die Berechnung der S- und T-Invarianten beliebiger S/T-Netze von zentraler Bedeutung. Insbesondere gilt für die gesamte Arbeit, dass alle Lösungsvektoren nichtnegativ sein müssen. Diese Einschränkung des Lösungsraums führt zu einem erhöhten Rechenaufwand, da alle unzulässigen Lösungen aussortiert werden müssen oder bei der Berechnung gar nicht erst auftreten dürfen.

Grundsätzlich gibt es zwei linear-algebraische Vorgehensweisen, um ganzzahlige Lösungen von Gleichungssystemen zu finden. Bei der Berechnung von *Basen* oder *Erzeugendensystemen* wird eine bestimmte Anzahl von Lösungsvektoren ermittelt, aus denen sich unendlich viele neue Lösungen durch Linearkombination bilden lassen. Einen Überblick über diese Methode, mögliche — und bewusst ausgewählte, leicht verständliche — Algorithmen und deren Komplexität liefert Kapitel 3.1.

Eine zweite Möglichkeit der Berechnung ist die *ganzzahlige lineare Programmierung*. Bei dieser Methode, im Englischen *Integer Programming* genannt, erhält man nur genau eine nichtnegative Lösung, kann dem Gleichungssystem jedoch zusätzliche Einschränkungen in Form von (Un-)Gleichungen beifügen, um die Gestalt der Lösung genauer zu definieren. Eine Einführung in diese Methodik liefert Kapitel 3.2.

## 3.1 Basen und Erzeugendensysteme

Bei Basen bzw. Erzeugendensystemen handelt es sich um eine feste Anzahl  $p$  von Basisvektoren bzw. Generatoren  $\vec{x}^i = (x_1^i \dots x_n^i) \neq \vec{0}$  als (in diesem Fall) *ganzzahlige* Lösungen eines linearen homogenen Gleichungssystems. Das besondere an diesen Vektoren ist, dass sich aus ihnen unendlich viele neue ganzzahlige Lösungen  $\vec{x} = (x_1 \dots x_n)$  durch Linearkombination erzeugen lassen. Die Komposition neuer Lösungen wird in [KJ87] beschrieben durch

$$\vec{x} = \sum_{i=1}^p \lambda_i \vec{x}^i.$$

Level	$\lambda_i \in$	Vektoren $\vec{x}^i$	Lösungsmenge $\{\vec{x}^i\}$	kombinierte Lösung $\vec{x}$
4	$\mathbb{N}$	$\vec{x}^i \succ \vec{0}, \vec{x}_j^i \in \mathbb{N}_0$	$\{\vec{x}^i\}$ eindeutig	$\vec{x} \succ \vec{0}, x_j \in \mathbb{N}_0$
3	$\mathbb{Q}^+$	$\vec{x}^i \succ \vec{0}, \vec{x}_j^i \in \mathbb{N}_0$	$\{\vec{x}^i\}$ eindeutig	$\vec{x} \succ \vec{0}, x_j \in \mathbb{N}_0$
2	$\mathbb{Z}$	$\vec{x}^i \in \mathbb{Z}^n$	$\{\vec{x}^i\}$ Basis	$\vec{x} \in \mathbb{Z}^n$
1	$\mathbb{Q}$	$\vec{x}^i \in \mathbb{Z}^n$	$\{\vec{x}^i\}$ Basis	$\vec{x} \in \mathbb{Z}^n$

Tabelle 3.1: Basen und Erzeugendensysteme, angelehnt an [KJ87]

Der Rechenaufwand zur Gewinnung dieser Basisvektoren und Generatoren lässt sich durch die Faktoren  $\lambda_i$  charakterisieren. Je enger man deren zulässigen Zahlenbereich fasst, desto mehr Generatoren werden letztendlich benötigt, um den kompletten Lösungsraum abzubilden. In Tabelle 3.1 sind vier Level für unterschiedliche Festlegungen der  $\lambda_i$  gegeben.

**Beispiel 3.1 (Erzeugendensysteme)** Anhand des Beispiels aus Abbildung 2.1 von Seite 9 lassen sich die Unterschiede für die Level 1 bis 4 gut hervorheben. Für diese Arbeit sind insbesondere die Level 3 und 4 interessant. Bei den Leveln 1 und 2 handelt es sich um ganzzahlige Basen des linearen homogenen Gleichungssystems. Die Basiselemente sind jeweils zwingend linear unabhängig voneinander und ihre Anzahl ist eindeutig durch den Rang der Inzidenzmatrix festgelegt [Row11]. In beiden Fällen sind alle Lösungen aus  $\mathbb{Z}^n$  zulässig, also auch Vektoren mit negativen Komponenten. Level 1 beschreibt das Ergebnis des gewohnten Gaußschen Eliminationsverfahrens [Wei11]. Eine mögliche, vom Computeralgebrasystem *Mathematica*<sup>1</sup> ermittelte, Basis ist

$$\mathcal{L}_1 = \{(2 \ 1 \ 0 \ 1)^T, (-1 \ 0 \ 1 \ 0)^T\}.$$

Die für Level 2 mit Hilfe der Hermite–Normalform<sup>2</sup> berechnete Basis  $\mathcal{L}_2$  ist ähnlich,

$$\mathcal{L}_2 = \{(0 \ 1 \ 2 \ 1)^T, (-1 \ 0 \ 1 \ 0)^T\},$$

lediglich ein Basiselement ist unterschiedlich. Ab Level 3 handelt es sich bei den Ge-

<sup>1</sup>Informationen zu *Mathematica* gibt es unter <http://www.wolfram.com>, Abruf: 27.02.2011.

<sup>2</sup>Durch Hermite–Normalform–Algorithmen werden ganzzahlige  $(m \times n)$ –Matrizen ähnlich dem Gauß–Verfahren in obere oder untere Dreiecksform gebracht. Durch Anwendung der gleichen Umformungen auf eine  $(n \times n)$ –Einheitsmatrix erhält man eine ganzzahlige Basis des zugrunde liegenden Gleichungssystems. Einen Überblick über Hermite–Normalform–Algorithmen bietet [Sus08].

neratoren um für S/T-Netze interessante Lösungen,

$$\mathcal{L}_3 = \{(0 \ 1 \ 2 \ 1)^T, (2 \ 1 \ 0 \ 1)^T\},$$

denn das sogenannte standardisierte  $\mathbb{Q}^+$ -Erzeugendensystem besteht aus nichtnegativen Vektoren. Es handelt sich präziser gesagt um eine eindeutige Menge von ganzzahligen, nichtnegativen Lösungen, die auch die Minimalitätsbedingung für Invarianten (vgl. Def. 2.11) erfüllen. Die auf Seite 9 vorgestellte weitere T-Invariante  $\vec{y}^3$  ist zum einen eine gültige Level 3-Linear kombination mit  $\lambda_1, \lambda_2 = \frac{1}{2}$ , zum anderen ein notwendiger Generator in  $\mathcal{L}_4$ ,

$$\mathcal{L}_4 = \{(0 \ 1 \ 2 \ 1)^T, (2 \ 1 \ 0 \ 1)^T, (1 \ 1 \ 1 \ 1)^T\},$$

da hier nur  $\lambda_i \in \mathbb{N}$  als Faktoren zur Erzeugung weiterer Lösungen zugelassen sind.  $\square$

Somit ist schon anhand dieses kleinen Beispiels leicht ersichtlich, dass mit zunehmender Einschränkung des Zahlenbereichs der  $\lambda_i$  die Anzahl der Generatoren von Level 3 zu Level 4 in der Regel steigt.

### 3.1.1 Exkurs: Inhomogene Gleichungssysteme

Mit allen im nächsten Abschnitt vorgestellten Algorithmen lassen sich grundsätzlich auch Lösungen inhomogener Gleichungssysteme  $A \cdot \vec{x} = \vec{b}$  berechnen [Pas86]. Falls vorhanden, bestehen Lösungen inhomogener Systeme aus der Summe einer partiellen Lösung  $\vec{x}_{part}$  und gegebenenfalls einer Linearkombination  $\vec{x}_{hom}$  für den homogenen Lösungsraum,

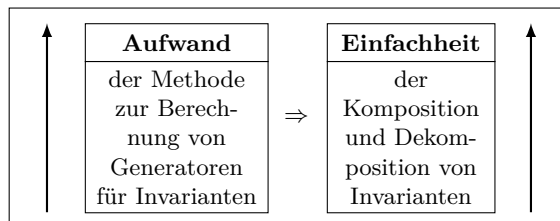
$$\vec{x}_{inhom} = \vec{x}_{part} + \underbrace{\sum_{i=1}^p \vec{x}^i}_{x_{hom}}.$$

Zur Berechnung des inhomogenen Lösungsraums ergänzt man die Spalten von  $A$  um die negierte rechte Seite  $\vec{b}$ :

$$[A \mid -\vec{b}] \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_{n+1} \end{pmatrix} = \vec{0}$$

Es müssen keine ganzzahligen, nichtnegativen partiellen Lösungen existieren. Im positiven Fall sind aber alle errechneten Generatoren  $\vec{x}$ , für die  $x_{n+1} = 1$  gilt, partielle Lösungen des inhomogenen Gleichungssystems. Aus Linearkombinationen der Generatoren lassen sich sämtliche Lösungen gewinnen, jedoch nur mit der zusätzlichen Bedingung  $x_{n+1} = 1$ . Auf Level 4 ist diese Bedingung leicht erfüllbar: Da es sich um ein

$\mathbb{N}$ -Erzeugendensystem handelt, kann zu jeder partiellen Lösung nur eine ganzzahlige Linearkombination ( $\lambda_i \in \mathbb{N}$ ) homogener Lösungen addiert werden, ohne die Anforderung  $x_{n+1} = 1$  zu gefährden. Die Komposition und Dekomposition einer allgemeinen Lösung  $\vec{x}_{inhom}$  ist hier also relativ simpel. Aber schon ab Level 3 stößt man auf das



**Abbildung 3.1:** Ganzzahlige Lösungen — Gesetz der Komplexität [KJ87]

Problem, dass für die  $\lambda_i$  auch rationale Werte möglich sind und ab Level 2 die Basisvektoren nicht zwingend nichtnegativ sein müssen. Demnach können auch zwei oder mehr partielle Lösungen addiert werden, solange die Bedingung  $x_{n+1} = 1$  nicht verletzt wird. Leichte Komposition neuer partieller Lösungen aus Generatoren geht also einher mit größerem Berechnungsaufwand der Generatoren, was in Abbildung 3.1 dargestellt ist. Dieses *Gesetz der Komplexität* gilt selbstverständlich auch für homogene lineare Gleichungssysteme. Die Level 4-Generatoren sind offenbar am besten geeignet, den Lösungsraum eines inhomogenen linearen Gleichungssystems zu beschreiben. Im ungünstigsten Fall ist der Rechenaufwand exponentiell.

### 3.1.2 Algorithmen

Für S/T-Netze sind wie bereits erwähnt die Level 3 und 4 aus Tabelle 3.1 interessant, bei denen nichtnegative, ganzzahlige Erzeugendensysteme berechnet werden. In dieser Netzklasse kann schließlich eine Transition nicht 1, 5-mal schalten oder eine Stelle  $\frac{2}{3}$  Marken besitzen. Zur Berechnung der Level 3-Generatoren wird der Algorithmus von Silva und Martínez [MS82, CS91] im nächsten Abschnitt vorgestellt, ein Berechnungsverfahren für ein  $\mathbb{N}$ -Erzeugendensystem [CD94] ist in Kapitel 3.1.2.2 aufgeführt. Zudem werden diese Algorithmen in ihrer nachfolgend aufgeführten oder gegebenenfalls modifizierten Form für alle Berechnungen von Erzeugendensystemen in dieser Arbeit verwendet.

#### 3.1.2.1 Berechnung eines Level 3-Erzeugendensystems

Der Algorithmus von Silva/Martínez, abgebildet auf Seite 23, basiert grundsätzlich auf einem Verfahren, bei dem nach und nach durch *elementare Zeilenumformungen*

alle *Spalten* einer  $(m \times n)$ -Matrix  $A$  (für S-Invarianten  $[N]$ , für T-Invarianten  $[N]^T$ ) annulliert, d.h. alle Einträge auf Null gesetzt werden. Die gleichen Operationen erfolgen simultan auf einer weiteren Matrix, die als Einheitsmatrix  $Y$  vom Format  $(m \times m)$  initialisiert und an  $A$  angehängt wird und am Ende die Lösungsvektoren enthält.

Die Zeilenanzahl dieser erweiterten Matrix  $U = [A|Y]$  wächst und schrumpft im Laufe des Algorithmus, da vielfach Zeilen addiert werden, um ein Spaltenelement auf Null zu setzen. Die Auswahl der nächsten zu annullierenden Spalte erfolgt nach einer bestimmten Heuristik, basierend auf der Anzahl ihrer positiven ( $\pi_k$ ) und negativen ( $v_k$ ) Elemente, um den Zuwachs neuer Zeilen gering zu halten: Zu Beginn werden die Spalten annulliert, die nur ein positives und negatives Element enthalten. Solche Spalten stellen einfache *Ketten* im Petri-Netz dar, d.h. entweder Transitionen mit nur einer Ein- und Ausgangsstelle oder umgekehrt Stellen mit einer Ein- und Ausgangstransition. Nach dieser Ketten-Elimination wird immer eine neue Spalte  $k$  ausgewählt, die zu einem minimalen Zuwachs neuer Zeilen führt. Die neue, während der Bearbeitung von Spalte  $k$  entstehende Matrix wird mit  $U^k = [A^k|Y^k]$  bezeichnet.

Zunächst werden die Zeilen der Matrix  $U^{alt}$  vom vorherigen Durchlauf auf drei Matrizen verteilt:  $U^-$  bzw.  $U^+$  enthalten die Zeilen mit negativem bzw. positivem Element in Spalte  $k$ ,  $U^k$  enthält zu Beginn alle Zeilen  $i$  aus  $U^{alt}$  mit  $A_{i,k}^{alt} = 0$ . Alle Zeilen von  $U^-$  werden anschließend mit allen Zeilen aus  $U^+$  so zu neuen Linearkombinationen addiert (insgesamt also  $\pi_k \cdot v_k$  Kombinationen), dass der Eintrag in den Spalten  $k$  dieser neuen Zeilen annulliert wird.

Der  $Y^k$ -Anteil jedes so entstandenen Vektors wird auf die Minimalität seiner Trägermenge untersucht, bevor die ganze Zeile der Matrix  $U^k$  hinzugefügt werden darf. Dieser Minimalitätstest verläuft in zwei Schritten: Ein Schnelltest überprüft zunächst, ob die Zeile weniger Einträge  $\neq 0$  ( $card(\|Y\|)$ ) hat als der  $rankUpperbound + 1$  der ursprünglichen Matrix<sup>3</sup>. Im zweiten Schritt wird die Trägermenge dieses  $Y^k$ -Anteils mit allen Trägermengen anderer Zeilen aus  $Y^k$ ,  $Y^+$  und  $Y^-$  verglichen. Ist sie auch hier minimal, so wird die Zeile noch durch den größten gemeinsamen Teiler aller ihrer Komponenten geteilt und dann  $U^k$  angefügt.

Sind alle Spalten annulliert oder wird keine weitere Spalte  $k$  mehr gefunden, die bearbeitet werden kann (falls  $\pi_k = 0$  oder  $v_k = 0$ ), so stoppt der Algorithmus. Alle Zeilen  $i$  in  $Y^k$ , in denen der  $A^k$ -Anteil der Nullvektor ist, geschrieben  $A_{i,-}^k = \vec{0}^T$ , bilden die Lösungsmenge. Einen Beispielablauf des Algorithmus zeigt Abbildung 3.2 auf der folgenden Seite.

<sup>3</sup> $card(\|Y\|)$  beschreibt hier die Anzahl der Elemente in der Trägermenge der Zeile  $Y$ . Der  $rankUpperbound$  bezeichnet die Anzahl der Spalten  $\neq \vec{0}$  der Matrix und ist damit eine Abschätzung, wie groß ihr Rang maximal sein kann.

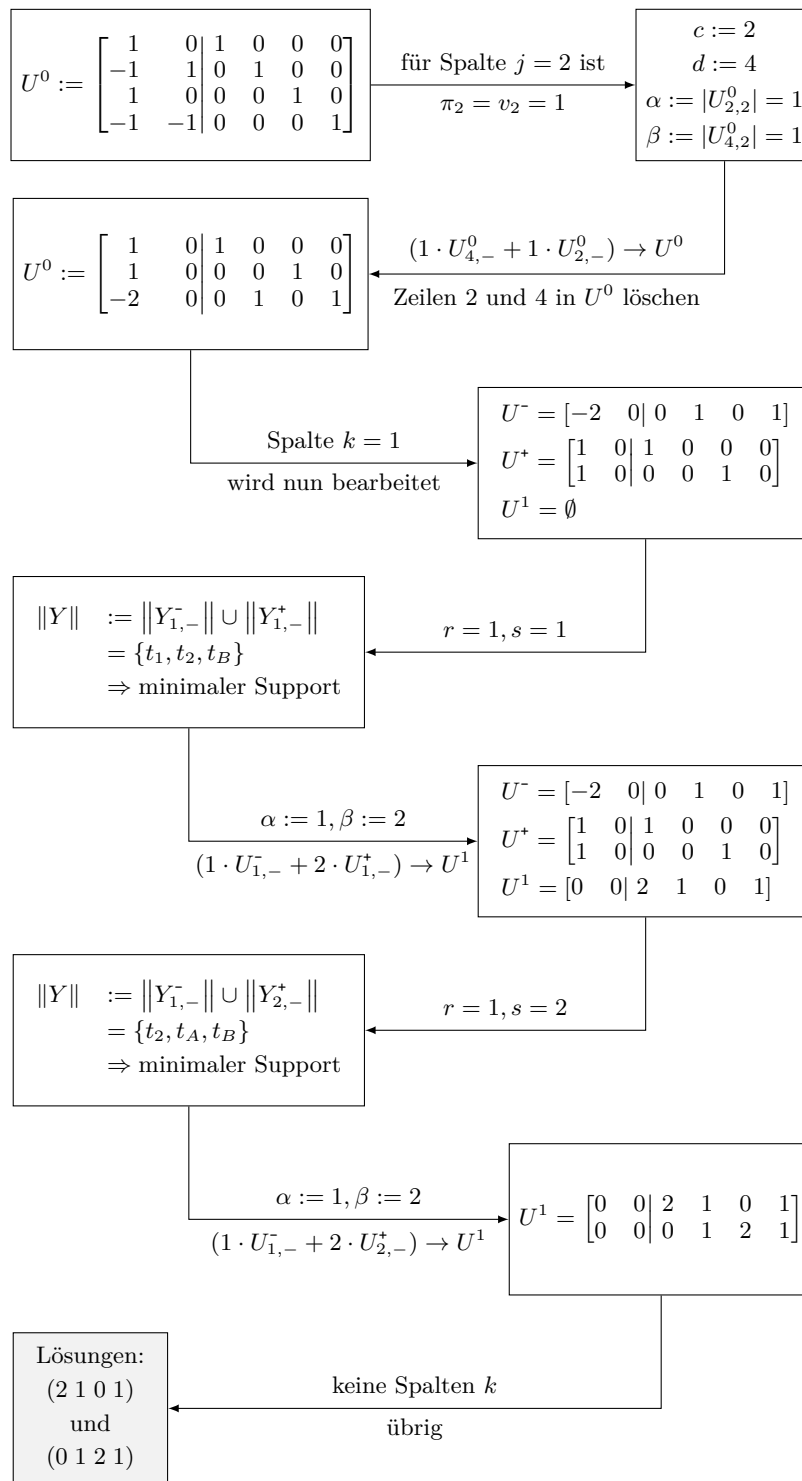


Abbildung 3.2: Berechnung des standardisierten  $Q^+$ -Erzeugendensystems für die T-Invarianten des Netzes N1 aus Abbildung 2.1

**Algorithmus 3.1:** LEVEL 3–ERZUEGENDENSYSTEM NACH SILVA/MARTÍNEZ [MS82, CS91]

---

**Eingabe:**  $U^0 := [A^0|Y^0]$  ▶  $Y^0 := I_m$ ,  $A^0$  hat Dimension  $m \times n$ .

**Ausgabe:** Alle Zeilen  $i$  in  $Y^k$  mit  $A_{i,-}^k = \vec{0}$  als Lösungsmenge.

---

```

1 for Spalten  $j := 1$  to  $n$  do ▶ „Ketten“ löschen.
2    $\pi_j :=$  Anzahl der positiven Elemente in  $j$ ;  $v_j :=$  Anzahl der negativen Elemente in  $j$ ;
3   if ( $\pi_j = v_j = 1$ ) then
4      $c :=$  Zeilenindex des positiven Elements in Spalte  $j$ ;
5      $d :=$  Zeilenindex des negativen Elements in Spalte  $j$ ;
6      $\alpha := |U_{c,j}^0|$ ;  $\beta := |U_{d,j}^0|$ ;
7     Füge die Zeile  $\alpha \cdot U_{d,-}^0 + \beta \cdot U_{c,-}^0$  zu  $U^0$  hinzu; Lösche die Zeilen  $c$  und  $d$  aus  $U^0$ ;
8   endif
9 endfor
10 Wähle die erste Spalte  $k$  in  $U^{alt} := U^0$  mit  $\pi_k \cdot v_k - (\pi_k + v_k) < 0$  oder eine Spalte  $k$ , die
     $\pi_k \cdot v_k$  minimiert (in beiden Fällen muss gelten:  $\pi_k, v_k > 0$ );
11 while (noch bearbeitbare Spalten  $k$  übrig) do ▶ Restliche Spalten annullieren
12    $U^- :=$  alle Zeilen  $i$  in  $U^{alt}$ , in denen  $A_{i,k}^{alt} < 0$ ;
13    $U^+ :=$  alle Zeilen  $i$  in  $U^{alt}$ , in denen  $A_{i,k}^{alt} > 0$ ;
14    $U^k :=$  alle Zeilen in  $U^{alt}$ , in denen  $A_{i,k}^{alt} = 0$ ;
15    $r :=$  Index der ersten Zeile von  $U^-$ ;
16   while ( $r \leq$  Zeilenanzahl( $U^-$ )) do
17      $s :=$  Index der ersten Zeile von  $U^+$ ;
18     while ( $s \leq$  Zeilenanzahl( $U^+$ )) do
19        $\|Y\| := \|Y_{r,-}^-\| \cup \|Y_{s,-}^+\|$ ;
20       if ( $card(\|Y\|) \leq rankUpperbound+1$ ) then ▶ Schnelltest
21         if ( $\|Y\|$  enthält keinen Support einer anderen Zeile von  $Y^k$ ,  $Y^+$  oder  $Y^-$ )
22           then ▶ Genaue Minimalitätsprüfung
23              $\alpha := |U_{s,k}^+|$ ;  $\beta := |U_{r,k}^-|$ ;
24             Füge die Zeile  $\alpha \cdot U_{r,-}^- + \beta \cdot U_{s,-}^+$  der Matrix  $U^k$  hinzu;
25             Teile diese Matrixzeile durch den ggT aller ihrer Elemente  $\neq 0$ ;
26             Speichere  $\|Y\|$ , den Support dieser Zeile;
27           endif
28         endif
29        $s := s + 1$ ;
30     endwhile
31    $r := r + 1$ ;
32 endwhile
33 Leere die Matrizen  $U^-$  und  $U^+$ ;  $U^{alt} := U^k$ ;
34 Wähle die erste Spalte  $k$  in  $U^{alt}$  mit  $\pi_k \cdot v_k - (\pi_k + v_k) < 0$  oder eine Spalte  $k$ , die  $\pi_k \cdot v_k$ 
    minimiert (in beiden Fällen muss gelten:  $\pi_k, v_k > 0$ );
35 endwhile

```

---

Für die Berechnung der T-Invarianten des Netzes aus Abbildung 2.1 ist es erforderlich, die Inzidenzmatrix [N1] zunächst zu transponieren, damit in  $Y$  auch wirklich T-Invarianten gespeichert werden. Die Lösungsmenge für dieses kleine Beispiel ist schnell gefunden. Bei großen Netzen, für die viele Linearkombinationen aus Zeilen von  $U^+$  und  $U^-$  gebildet werden müssen und nur wenige den Minimalitätstest *nicht* bestehen, kann der Aufwand explosionsartig ansteigen. Im schlimmsten Fall ist exponentieller Rechenaufwand zu erwarten.

### 3.1.2.2 Berechnung eines Level 4-Erzeugendensystems

Für die Berechnung des N-Erzeugendensystems wurde für diese Arbeit ein weiterer leicht verständlicher Algorithmus<sup>4</sup> ausgewählt, der auf Seite 25 abgebildet ist. Er erhält eine  $(m \times n)$ -Matrix  $A$  als Eingabe. Dann wird ein Stack  $\mathcal{P}$  mit  $n$  Einheitsvektoren initialisiert und zu diesen nach bestimmten Regeln so lange Einheitsvektoren addiert und damit Lösungen konstruiert, bis  $\mathcal{P}$  leer ist und alle gefundenen Lösungen Schritt für Schritt zu einer Menge  $\mathcal{B}$  hinzugefügt wurden. Die Einheitsvektoren werden nacheinander, beginnend mit  $\vec{e}_1 = (1\ 0 \dots 0)$ , auf  $\mathcal{P}$  geschoben. Zur Vermeidung redundanter Berechnungen werden an Position  $i$  im Stack die ersten  $(i-1)$  Elemente des dort liegenden Vektors *eingefroren*, d. h. die ersten  $\vec{e}_j$  Einheitsvektoren,  $j \leq (i-1)$ , dürfen nicht auf diesen Vektor addiert werden. In Algorithmus 3.2 wird hierzu das  $(n \times n)$ -Array *Frozen* initialisiert, das festlegt, welche Komponenten eines Vektors eingefroren sind. Im Beispiel auf Seite 26 sind die eingefrorenen Komponenten unterstrichen. An Position 4 im Stack dürfen beispielsweise die Einheitsvektoren  $\vec{e}_1$  bis  $\vec{e}_3$  nicht hinzuaddiert werden. Auch in späteren Rechenschritten sind an Position  $i$  im Stack genau  $(i-1)$  Elemente eines Vektors eingefroren, aber nicht zwingend die ersten  $(i-1)$ .

Solange  $\mathcal{P}$  nicht leer ist, wird das oberste Element  $\vec{t}$  vom Stack geholt und überprüft, ob es eine Lösung ist. Falls ja, wird es  $\mathcal{B}$  hinzugefügt, falls nicht, wird es bearbeitet. Neben der Regel, dass eingefrorene Komponenten des aktuellen Vektors  $\vec{t}$  ignoriert werden, müssen noch zwei weitere Bedingungen vor der Bearbeitung erfüllt sein: Die Summe  $(\vec{t} + \vec{e}_i)$  darf *nicht* größer sein (vgl. Definition 2.9) als irgendein Element der Lösungsmenge  $\mathcal{B}$  — dadurch ist die Minimalität gewährleistet — und es muss das Skalarprodukt  $(A \cdot \vec{t}) \circ (A \cdot \vec{e}_i) < 0$  gelten<sup>5</sup>. Sind die Bedingungen nicht erfüllt, wird  $\vec{t}$  verworfen; sind sie erfüllt, wird die Summe  $(\vec{t} + \vec{e}_i)$  auf  $\mathcal{P}$  geschoben. Der Einheitsvektor  $\vec{e}_i$  darf nun an höheren Positionen im Stack nicht mehr hinzuaddiert werden. Der

---

<sup>4</sup>Der Algorithmus ist zu finden in [CD94].

<sup>5</sup>Eine Erklärung dieser letzten Bedingung ist ebenfalls in [CD94] zu finden. Für diese Arbeit ist eine genaue Erläuterung der geometrischen Zusammenhänge, aus denen diese Bedingung entstand, nicht notwendig und würde den Rahmen sprengen.



Algorithmus stoppt, wenn  $\mathcal{P}$  leer ist. Die gefundenen Lösungen — falls vorhanden — sind in  $\mathcal{B}$  enthalten. Einen Ablauf für die Berechnung der T-Invarianten von Netz N1 zeigt Abbildung 3.3. Für die Berechnung der S-Invarianten müsste dem Algorithmus die transponierte Inzidenzmatrix  $A^T$  übergeben werden.

**Algorithmus 3.2:** LEVEL 4-ERZUGENDENSYSTEM NACH CONTEJEAN/DEVIE ([CD94])

---

**Eingabe:**  $A$  – Matrix,  $n$  – Anzahl der Spalten von  $A$   
**Ausgabe:**  $\mathcal{B}$ ,  $\mathbb{N}$ -Erzeugendensystem zu  $A \cdot \vec{x} = \vec{0}$

---

```

1 for  $i := 1$  to  $n$  do                                     ▶ Initialisierung von  $\mathcal{P}$  und Frozen
2    $\mathcal{P} := \text{push}(\vec{e}_i, \mathcal{P});$ 
3   for  $j := 1$  to  $n$  do
4     if  $(j < i)$  then  $\text{Frozen}[i, j] := \text{true}$ 
5     else  $\text{Frozen}[i, j] := \text{false}$ 
6     endif
7   endfor
8 endfor
9  $\mathcal{B} := \emptyset;$                                        ▶ Initialisierung von  $\mathcal{B}$  (Liste der Lösungsvektoren)
10 while  $(\mathcal{P} \neq \emptyset)$  do
11    $\vec{t} := \text{pop}(\mathcal{P});$                                    ▶ oberstes Element vom Stack nehmen
12    $h := \text{height}(\mathcal{P}) - 1;$                              ▶ Größe des Stack verringert sich damit um 1
13   if  $(A \cdot \vec{t} = \vec{0})$  then  $\mathcal{B} := \text{add}(\mathcal{B}, \vec{t})$      ▶  $\vec{t}$  ist eine Lösung und wird  $\mathcal{B}$  hinzugefügt
14   else
15     for  $i := 1$  to  $n$  do
16        $F[i] := \text{Frozen}[h + 1, i];$                        ▶ Lade „Verbotsliste“ von  $\vec{t}$ 
17     endfor
18     for  $i := 1$  to  $n$  do                                     ▶ füge  $\vec{t} + \vec{e}_i$  zu  $\mathcal{P}$  hinzu, falls erlaubt
19       if  $(F[i] = \text{false} \text{ and } (A \cdot \vec{t}) \circ (A \cdot \vec{e}_i) < 0 \text{ and } \min(\vec{t} + \vec{e}_i, \mathcal{B}))$ 
20       then begin
21          $\mathcal{P} := \text{push}(\vec{t} + \vec{e}_i, \mathcal{P});$                  ▶  $\vec{t}$  kann später noch zu einer Lösung werden
22          $h := h + 1;$                                        ▶ Größe des Stack erhöht sich um 1
23         for  $j := 1$  to  $n$  do
24            $\text{Frozen}[h, j] := F[j];$ 
25         endfor
26          $F[i] := \text{true};$                                        ▶ Vektor  $\vec{e}_i$  darf an höheren Positionen des
27       end                                                     Stacks nicht mehr addiert werden.
28     endif
29   endfor
30 endif
31 endwhile

```

---

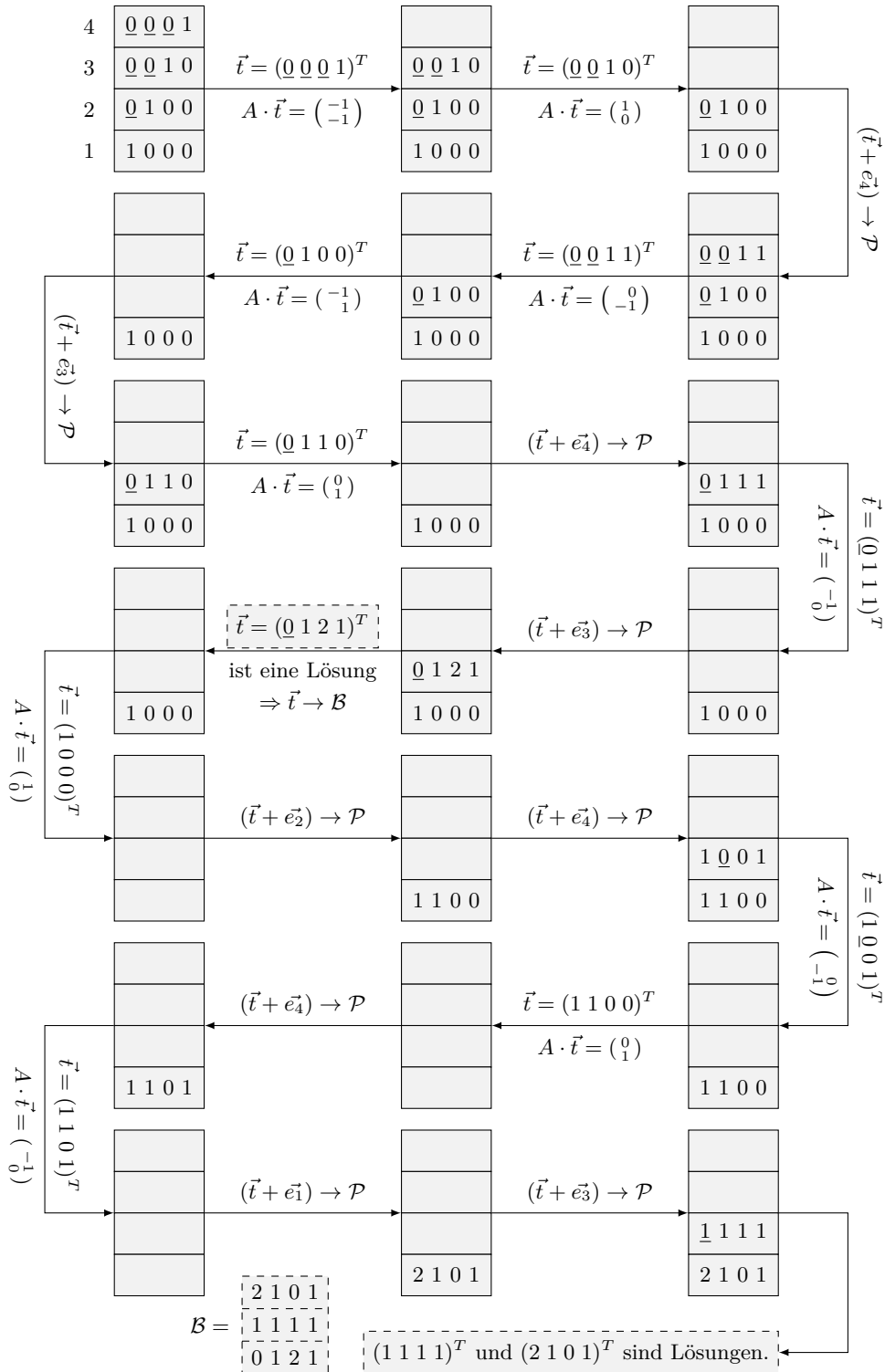


Abbildung 3.3: Berechnung des N-Erzeugendensystems für die T-Invarianten des Netzes N1 aus Abbildung 2.1

Auch bei diesem Algorithmus ist leicht ersichtlich, dass der Aufwand schnell sehr groß werden kann. Bei Lösungsvektoren, deren Komponenten große natürliche Zahlen sind, muss immer wieder der gleiche Vektor  $\vec{t}$  vom Stack genommen und Einheitsvektoren hinzuaddiert werden. Gleichzeitig muss dieser Vektor  $\vec{t}$  jedes Mal auf die zu erfüllenden Bedingungen getestet werden. Auch hier ist der Aufwand im ungünstigsten Fall exponentiell, obwohl die Komposition und Dekomposition von T-Invarianten mit Hilfe eines N-Erzeugendensystems einfach und übersichtlich ist und damit durchaus ihre Vorzüge hat.

## 3.2 Ganzzahlige lineare Optimierung

Eine Alternative zur Berechnung ganzer Erzeugendensysteme für die Gewinnung nichtnegativer T-Invarianten ist die ganzzahlige lineare Optimierung, bei der *genau eine* nichtnegative Lösung des Gleichungssystems errechnet wird. Ein lineares Optimierungsproblem besteht im Allgemeinen aus der Lösung einer linearen Extremwertaufgabe, bei der eine (lineare)  $n$ -stellige Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  mit

$$f(x_1, x_2, \dots, x_n) := c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n$$

( $c_i \in \mathbb{R}$  sind Konstanten) gegeben ist, die maximiert oder minimiert werden soll. Die  $x_i$  sind gewissen Nebenbedingungen (im Englischen Constraints genannt) unterworfen, welche ebenfalls als lineare Gleichungen oder Ungleichungen vorliegen.

Formal ist das sogenannte *lineare Optimierungsproblem (LOP)* nach [Lau04] folgendermaßen definiert:

**Definition 3.1 (Lineares Optimierungsproblem, LOP)** *Das lineare Optimierungsproblem (LOP) (oftmals auch lineares Programm genannt)*

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &:= c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n \rightarrow \text{Min} \\ (\text{kurz: } f(\vec{x}) &= \vec{c}^T \cdot \vec{x} \rightarrow \text{Min}) \\ A \cdot \vec{x} &= \vec{b} \\ \vec{x} &\succ \vec{0} \end{aligned}$$

wird LOP in Normalform genannt. Die Funktion  $f$  heißt Zielfunktion (ZF).  $A \cdot \vec{x} = \vec{b}$  und  $\vec{x} \succ \vec{0}$  mit  $A \in \mathbb{R}^{m \times n}$ ,  $\vec{x} \in \mathbb{R}^{n \times 1}$  und  $\vec{b} \in \mathbb{R}^{m \times 1}$  nennt man Nebenbedingungen (NB) des LOP. Die Lösungen  $\vec{x}$ , die der Nichtnegativitätsbedingung  $\vec{x} \succ \vec{0}$  genügen, heißen zulässige Lösungen des linearen Gleichungssystems. ■

Für nichtnegative Lösungen eines LOP wird zumeist die Simplexmethode (vgl. [Lau04], S. 13) verwendet. Bei der Berechnung ganzzahliger, nichtnegativer T-Invarianten  $\vec{x}$  hat die Optimierungsaufgabe die Zusatzforderung  $\forall i : x_i \in \mathbb{N}_0$ . Mit dieser Zusatzbedingung wird das LOP zu einem ganzzahligen LOP (GLOP, vgl. [Lau04]) erweitert:

**Definition 3.2 (Ganzzahliges Lineares Optimierungsproblem, GLOP)** *Das ganzzahlige lineare Optimierungsproblem (GLOP), oftmals auch ganzzahliges lineares Programm genannt, ist definiert als*

$$\begin{aligned} f(\vec{x}) &= \vec{c}^T \cdot \vec{x} \rightarrow \text{Min} \\ A \cdot \vec{x} &= \vec{b} \\ \forall i : x_i &\in \mathbb{N}_0. \end{aligned}$$

■

Auch hier wird zunächst mit Hilfe der Simplexmethode eine positiv rationale Lösung berechnet. Dann ändert man mit darauf aufbauenden Algorithmen, z.B. *Branch-and-Bound* oder *Cutting-Plane* (vgl. [Gen05]) die erhaltenen Werte auf die nächstgelegenen ganzen Zahlen so ab, dass alle Nebenbedingungen erfüllt sind. Die Lösung eines GLOP muss in der Regel nicht einem Level 3- oder Level 4-Generator entsprechen. Die Gestalt des Ergebnisses hängt allein von der Arbeitsweise des Programms ab, das die Extremwertaufgabe unter Berücksichtigung der Nebenbedingungen löst.

**Beispiel 3.2 (GLOP)** Für das Netz N1 von Seite 9 könnte man als Extremwertaufgabe die Berechnung einer T-Invarianten  $\vec{j}$  mit dem Zusatz  $\bar{\sigma}(t_A) = \bar{\sigma}(t_B) = 1$  und minimaler Schalthäufigkeit aller Transitionen stellen  $((j_1 \ j_2 \ j_3 \ j_4)^T$  entspricht hier  $(t_1 \ t_2 \ t_A \ t_B)^T$ ):

$$\begin{aligned} f(\vec{j}) &= \vec{1}^T \cdot \vec{j} \rightarrow \text{Min} \\ [\text{N1}] \cdot \vec{j} &= \vec{0} \\ \vec{j} &\succ \vec{0} \\ j_3 &= 1 \\ j_4 &= 1 \end{aligned}$$

Die optimale, die Zielfunktion minimierende Lösung ist  $\vec{j} = (1 \ 1 \ 1 \ 1)^T$ , was einem der drei Level 4-Generatoren entspricht. □

### 3.2.1 Algorithmen

Für die Bearbeitung ganzzahliger linearer Extremwertaufgaben wurden in dieser Arbeit ausschließlich das Programmpaket von Michael Gensty [Gen05] und die unter [BEN11] zu findende Software verwendet. Genstys Paket namens *IntegerProgramming* wurde im Computeralgebrasystem Mathematica implementiert und nutzt optional das Cutting-Plane- oder — in diesem Fall — das Branch-and-Bound-Verfahren (BAB) zur Berechnung einer ganzzahligen, nichtnegativen Lösung. Als Eingabeparameter erwartet der Branch-and-Bound-Algorithmus

1. die Angabe, ob die Zielfunktion zu minimieren oder maximieren ist,
2. die Matrix  $A$ , die das Gleichungssystem repräsentiert,
3. die *rechte Seite*  $\vec{b}$ ,
4. die Zielfunktion und
5. weitere Nebenbedingungen.

Für das Beispiel aus dem letzten Abschnitt sähe die Eingabe folgendermaßen aus:

```
BAB[
  "Minimum",
  {{1,-1,1,-1},{0,1,0,-1}},
  {0,0},
  t[1]+t[2]+t[3]+t[4],
  {t[3]==1,t[4]==1}
]
```

Als Ergebnis liefert das Programm  $\{\{1,1,1,1\},\{4\}\}$ , also zum einen den Lösungsvektor  $(1\ 1\ 1\ 1)^T$  und zum anderen den Wert der Zielfunktion, der in diesem Fall „4“ ist.

Dieses Kapitel bildet den Abschluss der mathematischen und netztheoretischen Grundlagen. Es wurde deutlich, dass sowohl beim Lösen der Zustandsgleichung als auch beim Berechnen von Invarianten — unter Umständen mit bestimmten Nebenbedingungen — lineare Gleichungssysteme gelöst werden müssen. Ab dem nächsten Kapitel werden die hier aufgeführten Methoden nun im Rahmen der Erreichbarkeitsfrage in S/T-Netzen angewendet.



# 4 Das Erreichbarkeitsproblem und lineare Algebra: Existierende Algorithmen

Im letzten Jahrzehnt beschäftigten sich einige Netztheoretiker mit der Erforschung linear-algebraischer Methoden zur Lösung des Erreichbarkeitsproblems [Kos08, RK04, WW10]. Zwei der entwickelten Verfahren [Kos08, RK04] werden in den nächsten Abschnitten beispielhaft vorgestellt, da einige ihrer Bestandteile den in den Kapiteln 6 und 7 diskutierten Ansätzen stark ähneln. Beide Methoden besitzen ihre Schwächen und sind nicht ausgereift genug, um für beliebige S/T-Netze die Erreichbarkeitsfrage befriedigend zu lösen.

## 4.1 Das Verfahren von Alexander Kostin

Das Verfahren von Alexander Kostin [Kos08] ist eine Kombination von linear-algebraischen Berechnungen und Simulationsschritten zur Lösung des Erreichbarkeitsproblems für schlingenfreie Netze. Es startet mit der Erweiterung eines S/T-Netzes  $N$  zu einem Netz  $N_c$  durch eine zusätzliche Transition  $t_c$ , die aus der Differenz von Startmarkierung  $M_0$  und Zielmarkierung  $M$  (mit  $M_0 \neq M$ ) entsteht. Wenn  $t_c$  unter Markierung  $M$  einmal schaltet, wird  $M_0$  wieder erreicht. Falls es also eine Schaltsequenz  $M_0[\sigma]M[t_c]M_0$  gibt, liegt Erreichbarkeit vor; gleichzeitig muss es aber auch eine T-Invariante

$$\vec{F} = \begin{pmatrix} \bar{\sigma} \\ t_c \end{pmatrix}$$

geben mit

$$[N_c] \cdot \begin{pmatrix} \bar{\sigma} \\ t_c \end{pmatrix} = \vec{0}$$

und  $t_c = 1$ . Eine T-Invariante  $\vec{F}$  und eine realisierbare, zu  $\bar{\sigma}$  passende Schaltsequenz  $\sigma$  zu finden ist nun die eigentliche Aufgabe, jedoch wird die Erreichbarkeitsanalyse durch die theoretisch unendliche Menge von T-Invarianten erheblich verkompliziert.

Welche Linearkombination von T-Invarianten ist eine passende? Kostins Ansatz beschreibt eine Idee, wie die minimalen T-Invarianten von  $N_c$  in Klassen eingeteilt und zu möglicherweise passenden Linearkombinationen  $\vec{F}$  addiert werden können. Trotz dieses großen linear-algebraischen Anteils wird hier für jede berechnete Kombination durch Simulationsschritte überprüft, ob sie die gesuchte Schaltsequenz  $\sigma t_c$  ist, die die Anfangsmarkierung reproduziert. Es sei noch darauf hingewiesen, dass das Verfahren im Fall keiner vorhandenen minimalen T-Invariante mit  $t_c \geq 1$  sofort stoppt, da damit bereits Nicht-Erreichbarkeit nachgewiesen ist. Für das Netz N3 in Abbildung 4.1 wird nun der Ablauf des Algorithmus beispielhaft erläutert. Das Verfahren gliedert sich in die folgenden Schritte:

1. Erweiterung von  $N$  zu  $N_c$
2. Berechnung und Klassifizierung der T-Invarianten
3. Untersuchung auf strukturelle Beschränktheit
4. Simulation der kombinierten T-Invariante  $\vec{F}$

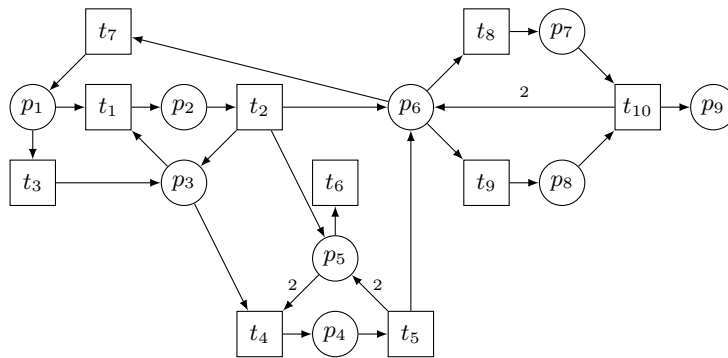


Abbildung 4.1: S/T-Netz N3 aus [Kos08]

**Erweiterung von N3 zu N3<sub>c</sub>**

Für Netz N3 seien die Anfangs- und Zielmarkierung  $M3_0 = (200000000)^T$  und  $M3 = (200000001)^T$  der Form  $(p_1 \dots p_9)^T$  gegeben. Durch  $M3_0 - M3 = (00000000-1)^T$  erhält man das um die Transition  $t_c$  erweiterte Netz  $N3_c$ , wobei  $t_c$  eine Marke von Stelle  $p_9$  nimmt.

**Berechnung und Klassifizierung der T-Invarianten**

Im nächsten Schritt werden die minimalen T-Invarianten mit Algorithmus 3.1 berechnet. Das Level 3-Erzeugendensystem besteht aus den T-Invarianten  $\vec{F}^1$  bis  $\vec{F}^3$  der Form  $(t_1 \dots t_{10} t_c)^T$ , die in Abbildung 4.3 mit zugehörigen Netzdarstellungen gegeben sind. Anschließend werden diese klassifiziert. Dazu wird zunächst ein sogenannter un-



gerichteter *Relationsgraph* erzeugt, dessen Knoten die Invarianten sind. Zwei Knoten werden verbunden, wenn sie mindestens eine Stelle in ihren Netzdarstellungen gemeinsam haben. Für das obige Beispiel sieht der Relationsgraph folgendermaßen aus:

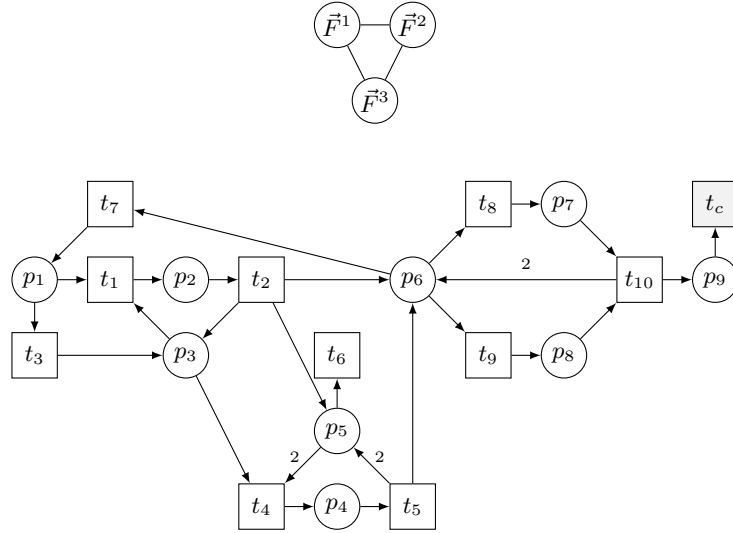


Abbildung 4.2:  $N3_c$  — um  $t_c$  erweitertes Netz  $N3$ .

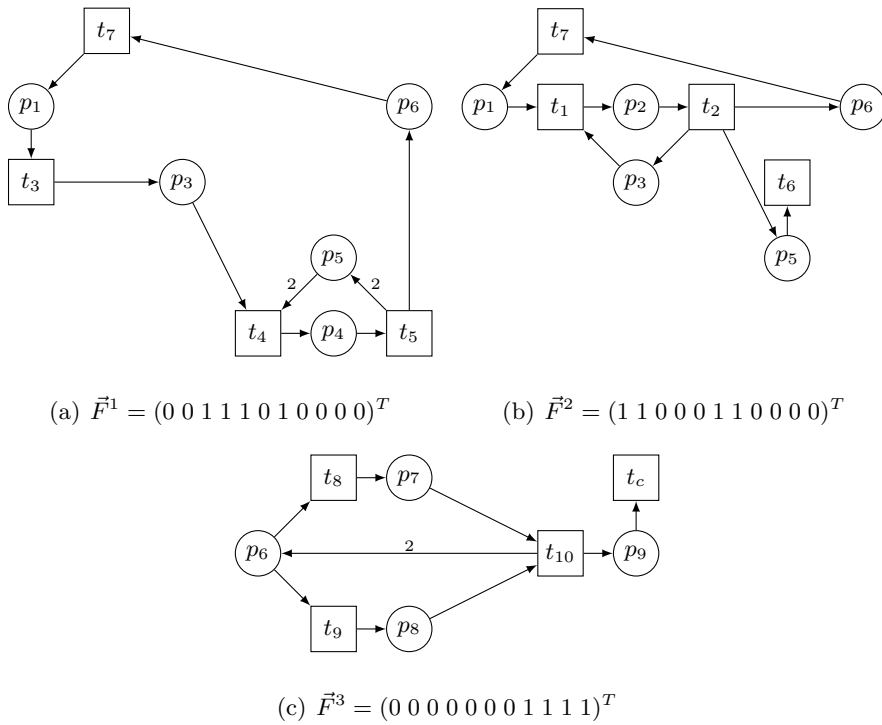
Grundsätzlich kann ein Relationsgraph eines Petri-Netzes auch aus nicht miteinander verbundenen Teilgraphen, sogenannten *Zusammenhangskomponenten*, bestehen. Das ist genau dann der Fall, wenn verschiedene Gruppen von T-Invarianten keinerlei gemeinsame Stellen haben. Diese grobe Einteilung ist der erste Schritt der Klassifizierung: Nur Invarianten, die in der gleichen Zusammenhangskomponente liegen, können sich prinzipiell gegenseitig Marken leihen, um realisierbar zu werden. T-Invarianten sind schließlich nur unter bestimmten Markierungen realisierbar, die jeweils erst einmal erreicht werden müssen. Im zweiten Schritt werden darum alle Zusammenhangskomponenten getrennt voneinander untersucht, in denen mindestens eine T-Invariante mit  $t_c \geq 1$  existiert. Diese ist in diesem Beispiel durch  $\vec{F}^3$  gegeben. In jeder Zusammenhangskomponente wird die Menge  $\Phi_c$  der in ihr enthaltenen T-Invarianten wie folgt anhand von  $t_c$  klassifiziert:

$$\{\vec{F}^j \mid t_c = 0\} = \text{nicht komplementäre, minimale T-Invarianten} \quad (1)$$

$$\{\vec{F}^j \mid t_c = 1\} = \text{singuläre, komplementäre T-Invarianten} \quad (2)$$

$$\{\vec{F}^j \mid t_c > 1\} = \text{komplementäre, minimale T-Invarianten} \quad (3)$$

Mit Minimalität ist hierbei die Minimalität der Trägermenge gemeint. Für obiges Beispiel ist  $\Phi_c = \{\vec{F}_1, \vec{F}_2, \vec{F}_3\}$ . Da zum Nachweis der Erreichbarkeit die Transition  $t_c$  genau



**Abbildung 4.3:** Minimale T-Invarianten von Netz  $N3_c$  mit jeweiligen Netzdarstellungen

einmal schalten muss, ist die zugehörige realisierbare T-Invariante  $\vec{F}$  eine Linearkombination:

$$\vec{F} = \vec{F}_c + \sum_{j=1}^n k_j \vec{F}_{nc}^j, \quad k_j \in \mathbb{N}_0 \quad (4)$$

Hierbei ist

$$\begin{aligned} \vec{F}_c &= \text{T-Invariante aus (2) oder Linearkombination aus} \\ &\quad \text{T-Invarianten von (2) und (3) mit } t_c = 1 \\ &\Rightarrow \vec{F}_c = \vec{F}^3 \end{aligned}$$

$$\begin{aligned} \vec{F}_{nc}^j &= \text{eine von insgesamt } n \text{ T-Invarianten aus (1)} \\ &\Rightarrow \vec{F}_{nc}^1 = \vec{F}^1, \vec{F}_{nc}^2 = \vec{F}^2 \end{aligned}$$

Für das Netz  $N3_c$  erhält man somit folgende Linearkombination:

$$\vec{F} = \vec{F}_3 + k_1 \cdot \vec{F}^1 + k_2 \cdot \vec{F}^2, \quad k_j \in \mathbb{N}_0$$

Nun stellt sich die Frage, wie oft jede T–Invariante in die Linearkombination eingehen muss. Wie bereits erwähnt, müssen die T–Invarianten sich eventuell gegenseitig Marken leihen, um realisierbar zu werden. Durch Simulation, ausgehend von der Startmarkierung, werden diese Markendefizite ermittelt. Unter der Startmarkierung  $M3_0$  erhält man für  $\vec{F}^1$  bis  $\vec{F}^3$  folgenden Bedarf, wobei positive Einträge in einer Tabellenzelle angeben, wie viele Marken sich eine T–Invariante auf der zugehörigen Stelle leihen muss; negative Einträge geben an, wie viele Marken auf der jeweiligen Stelle von der T–Invariante produziert werden:

	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$p_7$	$p_8$	$p_9$
$\vec{F}^1$	-1	-	-1	-1	<b>2</b>	-1	-	-	-
$\vec{F}^2$	-1	-1	<b>1</b>	-	-1	-1	-	-	-
$\vec{F}^3$	-	-	-	-	-	<b>2</b>	-1	-1	-1

Beispielsweise muss  $\vec{F}^1$  sich auf  $p_5$  zwei Marken leihen,  $\vec{F}^2$  auf  $p_3$  eine und  $\vec{F}^3$  auf  $p_6$  wiederum zwei.  $\vec{F}^1$  kann die eine Marke auf  $p_3$  bereitstellen, die  $\vec{F}^2$  benötigt. Zusammen können  $\vec{F}^1$  und  $\vec{F}^2$  die zwei Marken auf  $p_6$  liefern, die  $\vec{F}^3$  braucht. Ob  $\vec{F}^2$  (als einzige Möglichkeit!) zwei Marken auf  $p_5$  für  $\vec{F}^1$  erzeugen kann, hängt davon ab, ob die Netzdarstellung von  $\vec{F}^2$  *strukturell beschränkt* ist.

### Untersuchung auf strukturelle Beschränktheit

Ist eine Netzdarstellung einer T–Invariante strukturell beschränkt, so kann die T–Invariante unter Umständen nicht die Anzahl von Marken bereitstellen, die eine andere benötigt. Durch den durch Satz 2.3 von Seite 15 gegebenen Schnelltest kann für die Netzdarstellung jeder T–Invariante festgestellt werden, ob dies zutrifft. Hierzu wird mit Hilfe von Algorithmus 3.1 nach nichtnegativen S–Invarianten gesucht.  $\vec{F}^3$  geht ohnehin nur einmal in die Linearkombination  $\vec{F}^{\vec{i}}$  ein und muss somit nicht untersucht werden. Für die beiden anderen T–Invarianten  $\vec{F}_{nc}^j$  erhält man:

$\vec{F}^1$ : hat zwei minimale S–Invarianten  $(0\ 0\ 2\ 1\ 0)^T$  und  $(1\ 1\ 1\ 0\ 1)^T$ , in der Summe also eine S–Invariante  $\vec{i} = (1\ 1\ 3\ 1\ 1)^T$  und ist strukturell beschränkt. Sie geht also maximal einmal in die Linearkombination (4) ein. Damit legen wir  $k_1 = 1$  fest, denn  $\vec{F}^1$  muss einmal realisiert werden, um  $\vec{F}^2$  eine Marke auf  $p_3$  zu leihen.

$\vec{F}^2$ : ist nicht von S–Invarianten überdeckt; der Schnelltest war nicht erfolgreich und eine Aussage zu struktureller Beschränktheit kann noch nicht getroffen werden.

Falls  $\vec{F}^2$  strukturell unbeschränkt ist, gilt die Unbeschränktheit für mindestens eine der  $n$  Stellen  $p_i$  in der Netzdarstellung von  $\vec{F}^2$ . Dies ist gegeben, wenn

$$\exists \vec{t} \in \mathbb{N}_0^m, \vec{t} \neq \vec{0} : [N] \cdot \vec{t} = \Delta M \geq \vec{0}, \quad \Delta M \neq \vec{0},$$

wobei  $\Delta M$  ein  $n$ -stelliger Stellen-Vektor ist, der die Veränderung der Marken im Netz durch das Feuern der Transitionen in  $\vec{t}$  (Schaltreihenfolge irrelevant) angibt:

$$\Delta M = (\Delta m_1 \dots \Delta m_i \dots \Delta m_n)$$

Für obiges Beispiel werden mindestens zwei Marken auf Stelle  $p_5$  benötigt. Damit ist

$$\Delta M = (0 \ 0 \ 0 \ 2 \ 0)^T.$$

Die Bedingungen werden in einem ganzzahligen linearen Optimierungsproblem zusammengefasst:

$$v := t_1 + t_2 + t_6 + t_7 \rightarrow \text{Min}$$

$$[N] \cdot \vec{t} = \Delta M$$

$$v \geq 1, t_i \geq 0 \ \forall i \in \{1, 2, 6, 7\}$$

Insgesamt sollen also so wenige Transitionen wie möglich schalten; darum ist die Zielfunktion zu minimieren. Mindestens eine Transition ( $v \geq 1$ ) muss jedoch feuern. Das lineare Programm liefert als Lösung den Vektor  $(t_1 \ t_2 \ t_6 \ t_7)^T = (2 \ 2 \ 0 \ 2)^T$ ,  $\vec{F}^2$  kann also Marken auf  $p_5$  für  $\vec{F}^1$  bereitstellen. Nun errechnet man eine obere Schranke dafür, wie häufig  $\vec{F}^2$  maximal realisiert werden muss, damit genügend Marken erzeugt werden:

$$k_2 = \max \left( \left\lceil \frac{t_i}{\vec{F}^2(i)} \right\rceil \right), i = 1, 2, 6, 7,$$

d.h.

$$k_2 = \max \left( \left\lceil \frac{2}{1} \right\rceil, \left\lceil \frac{2}{1} \right\rceil, \left\lceil \frac{0}{1} \right\rceil, \left\lceil \frac{2}{1} \right\rceil \right) = \max(2, 2, 0, 2) = 2.$$

T-Invariante  $\vec{F}^2$  geht somit zweimal in die Linearkombination  $\vec{F}$  ein:

$$\vec{F} = \vec{F}^3 + \vec{F}^1 + 2 \cdot \vec{F}^2$$

### Simulation von $\vec{F}$

Sobald eine Linearkombination gefunden ist, erfolgt der Simulationsschritt. Es wird nun versucht, von  $M3_0$  ausgehend mit den in  $\vec{F}$  gegebenen Schalthäufigkeiten eine Schalt-

sequenz zu finden, die letztlich  $M3_0$  reproduziert. Wird eine solche Sequenz unter *allen* möglichen Linearkombinationen der Zusammenhangskomponente nicht gefunden, liegt keine Erreichbarkeit vor. Für Netz N3 ist die Erreichbarkeit von M3 aus  $M3_0$  nachweisbar:  $t_3t_1t_2t_7t_1t_2t_4t_5t_8t_9t_{10}t_6t_6t_7t_7t_c$  ist eine passende Schaltsequenz, die  $M3_0$  im erweiterten Netz  $N3_c$  reproduziert.

### Zusammenfassung

Für den in diesem Kapitel diskutierten Beispielablauf ist Kostins Algorithmus leicht nachvollziehbar, da es insgesamt nur drei minimale T-Invarianten gab, die alle in die finale Linearkombination eingingen. So simpel ist die Lösung des Erreichbarkeitsproblems im Allgemeinen nicht: Bereits nach der Berechnung und Klassifikation der minimalen T-Invarianten finden Simulationsschritte statt, die für jede Netzdarstellung einer Invariante überprüfen, ob sie unter der Startmarkierung realisierbar ist oder sich Marken leihen muss. Der anschließende Schritt — geeignete T-Invarianten auszuwählen und zu überprüfen, ob sie sich gegenseitig die notwendigen Marken leihen können — erfolgt zwar systematisch, ist aber dennoch bei großen Netzen alles andere als trivial. Unter Umständen müssen immer wieder neue Linearkombinationen aus T-Invarianten gebildet und Schaltsequenzen simuliert werden. Allein die Addition von T-Invarianten aus Gruppe (3) derart, dass am Ende stets  $t_c = 1$  gewährleistet ist, ist unter Umständen mit erheblichem Aufwand verbunden (vgl. hierzu die Berechnung des Level 4-Erzeugendensystems ab Seite 24).

Die Häufigkeit, wie oft eine T-Invariante in eine Linearkombination eingeht, ist gegebenenfalls durch die mögliche strukturelle Beschränktheit ihrer Netzdarstellung limitiert. Falls sie ausreichend Marken auf ausgewählten Stellen bereitstellen kann, muss sie auch nicht häufiger als das für sie ermittelte Maximum realisiert werden: Ein Überschuss an Marken auf einer Stelle ist ebenfalls unerwünscht, da er vielleicht nicht von anderen T-Invarianten getilgt werden kann. Damit ist die Anzahl der Linearkombinationen innerhalb einer Zusammenhangskomponente endlich. Im schlimmsten Fall müssen alle konstruierbaren Linearkombinationen gebildet und simuliert werden:

“We can note only that the algorithm will spend most of its time calculating minimal-support T-invariants, solving ILP problems, and trying to find reachability paths for calculated T-invariants.” — [Kos06], S. 620

## 4.2 Das Verfahren von Ramachandran und Kamath

Das in diesem Kapitel vorgestellte Verfahren von Ramachandran und Kamath basiert auf der Tatsache, dass in azyklischen S/T-Netzen die Zustandsgleichung nicht nur ein notwendiges, sondern sogar ein hinreichendes Kriterium für die Erreichbarkeit einer Markierung  $M_B$  aus einer Markierung  $M_A$  ist. Erreichbarkeit ist in diesem Fall also genau dann gegeben, wenn die Gleichung

$$[N] \cdot \vec{y} = M_B - M_A = \Delta M$$

eine nichtnegative, ganzzahlige Lösung  $\vec{y}$  besitzt. Ein beliebiges, aber auch — wie bereits in Kapitel 4.1 gefordert — schlingenfreies Petri-Netz wird Schritt für Schritt in mehrere Schichten entfaltet, genauer gesagt vielfach kopiert und die Kopien (Schichten) nach bestimmten Regeln miteinander verbunden. Auf diese Weise wird das Netz in ein azyklisches Netz transformiert und die Zustandsgleichung ist als hinreichendes Kriterium anwendbar. Ist  $M_B$  aus  $M_A$  erreichbar, so wird der Algorithmus nach einer bestimmten Anzahl von  $n$  zusätzlichen Schichten stoppen, da eine passende Lösung  $\vec{y}$  für die Zustandsgleichung gefunden wurde. Die einzelnen Schritte werden anhand von Netz **N4** mit Startmarkierung  $M_{4A} = (2\ 0\ 0\ 0)^T$  und Zielmarkierung  $M_{4B} = (0\ 0\ 0\ 2)^T$  der Form  $(p_1 \dots p_4)^T$  genauer erläutert. Es gliedert sich in die folgenden Schritte:

1. Transformation/Entfaltung in ein azyklisches Netz
2. Berechnung einer T-Invariante bzw. Lösung der Zustandsgleichung

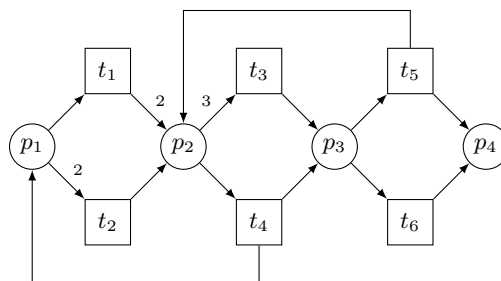


Abbildung 4.4: S/T-Netz N4

### Transformation/Entfaltung in ein azyklisches Netz

Bei der Transformation in ein azyklisches Netz werden als neue Schichten exakte Kopien der Stellen des Originalnetzes erzeugt. Die Transitionen des ursprünglichen Netzes bzw. deren Kopien verbinden die einzelnen Schichten so miteinander, dass ihre eingehenden Kanten von den Stellen auf Schicht  $n$  kommen, ausgehende Kanten zu den Stellen in Schicht  $n + 1$  führen. Zusätzliche „Helfer“-Transitionen  $c_x$  verbinden noch

die exakten Kopien der Stellen auf zwei aufeinander folgenden Schichten miteinander, um gegebenenfalls übrig gebliebene Marken weiter zu reichen. Präzise formuliert existieren folgende Knoten im Netz, deren Label jeweils auf Schicht und Originalknoten hinweist:

- $p_{x|n}$ : Kopie von Stelle  $p_x$  auf Schicht  $n$
- $t_{y|n,n+1}$ : Kopie von Transition  $t_y$ , die die Schichten  $n$  und  $n + 1$  verbindet:
  - $\bullet t_{y|n,n+1} = \{p_{i|n} \mid p_i \in \bullet t_y\}$ ,
  - $t_{y|n,n+1}^\bullet = \{p_{j|n+1} \mid p_j \in t_y^\bullet\}$ .
- $c_{x|n,n+1}$ : „Helfer“-Transition, die die Stellen  $p_{x|n}$  und  $p_{x|n+1}$  verbindet.

Das Netz N4 nach dem ersten Entfaltungsschritt zeigt Abbildung 4.5. Zur Vergleich-

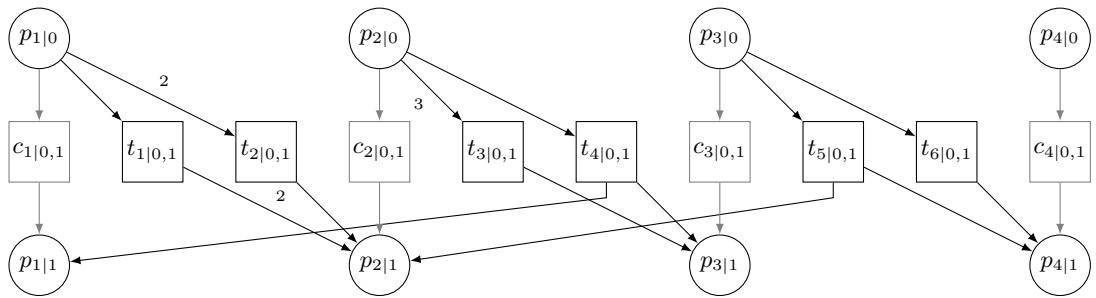


Abbildung 4.5: Netz N4 erweitert um eine zusätzliche Schicht.

barkeit mit den in Folgekapiteln vorgestellten Verfahren wird dieses Netz noch um zwei zusätzliche Transitionen  $t_A$  und  $t_B$  erweitert. Transition  $t_A$  legt die Marken auf das Netz, die der Anfangsmarkierung  $M_{4A}$  entsprechen, d.h. zwei Marken auf  $p_1$ . Durch Transition  $t_B$  wird die Zielmarkierung  $M_{4B}$  wieder vom Netz genommen, d.h. zwei Marken von  $p_4$ . Transition  $t_A$  erzeugt hierbei immer die Startmarkierung auf der untersten Schicht und  $t_B$  ist stets mit der aktuell höchsten Schicht verbunden. Der Vorteil dieser zwei zusätzlichen Transitionen ist, dass nun nach nichtnegativen T-Invarianten gesucht werden kann, in denen  $t_A$  und  $t_B$  genau einmal schalten. Bei positivem Suchergebnis wird die leere Markierung reproduziert, was dem Erreichen der Zielmarkierung aus der Startmarkierung entspricht. Zudem ist dann auch die Zustandsgleichung erfüllt.

### Berechnung einer T-Invariante bzw. Lösung der Zustandsgleichung

Für die Berechnung der T-Invariante bietet sich wiederum ein ganzzahliges lineares Optimierungsproblem mit dem zusätzlichen Constraint  $\bar{\sigma}(t_A) = \bar{\sigma}(t_B) = 1$  an. Da nur genau eine Lösung gesucht wird, sollte dieses Verfahren dem der Ermittlung von Erzeugendensystemen vorgezogen werden. Alle Lösungen  $\bar{\sigma}(t_A), \bar{\sigma}(t_B) \neq 1$  interessieren nicht und sollten erst gar nicht berechnet werden.

Für das Netz  $N_4$  werden insgesamt fünf zusätzliche Schichten benötigt, bis das lineare Programm eine Lösung findet. Das finale Netz ist in Abbildung 4.6 dargestellt, der Lösungspfad grau unterlegt. Die Schaltreihenfolge der Transitionen  $t_i$  entspricht der Schaltsequenz, die im Originalnetz notwendig ist, um  $M_{4B}$  aus  $M_{4A}$  zu erreichen. In diesem Fall wäre eine mögliche Sequenz also  $t_2t_4t_1t_5t_3t_6$ .

### Stärken und Schwächen des Verfahrens

Für ein gegebenes Erreichbarkeitsproblem, bei dem der Schalthäufigkeitsvektor bereits bekannt ist, ist dieses Verfahren sehr einfach anzuwenden: Man fügt maximal so viele neue Schichten  $d$  wie die Summe aller Schalthäufigkeiten der Transitionen hinzu. Dies ist die Obergrenze, wenn man vom ungünstigsten Fall ausgeht, in dem auf einer Schicht nur genau eine Transition schaltet. Konfliktfreie Transitionen, die nicht um Marken konkurrieren, können auf der gleichen Schicht feuern. Gibt es nach maximal  $d$  Schichten eine nichtnegative Lösung für die Zustandsgleichung für einen gegebenen Schalthäufigkeitsvektor, so liegt Erreichbarkeit vor. Wird keine solche Lösung gefunden, so ist die Zielmarkierung für diese Schalthäufigkeiten nicht erreichbar.

Allgemein ist die Zielmarkierung also *genau dann* erreichbar, *wenn* für das um eine bestimmte Anzahl von Schichten erweiterte Netz eine nichtnegative Lösung der Zustandsgleichung existiert.

An dieser Stelle zeigt sich auch die Schwäche des Verfahrens: Ist eine Obergrenze  $d$  bekannt, so ist die Konstruktion des entfalteten Netzes und die Lösung der Zustandsgleichung lediglich eine Frage des Rechenaufwands. Wenn jedoch keine maximale Schichtenanzahl bekannt ist, d.h. der Lösungsraum durch die Schalthäufigkeiten der Transitionen nicht eingeschränkt wurde, existiert im Fall der Nicht-Erreichbarkeit auch kein Abbruchkriterium für den Algorithmus. Es ist letztendlich also nicht feststellbar, nach welcher Schicht von Nicht-Erreichbarkeit auszugehen ist.



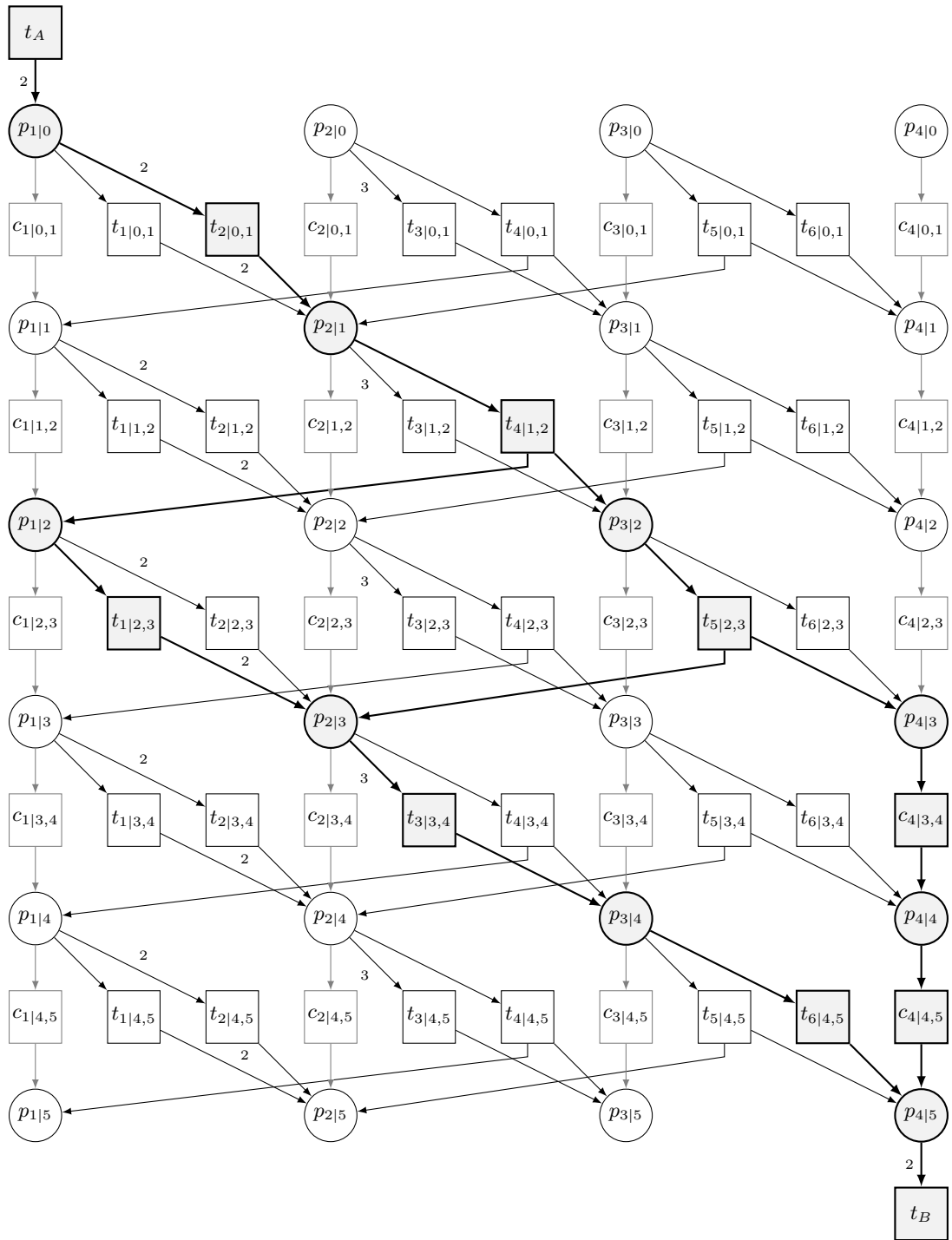


Abbildung 4.6: Netz N4 erweitert um fünf zusätzliche Schichten.



# 5 Verfeinerung des Erreichbarkeitsproblems im Zusammenspiel mit linearer Algebra

In Kapitel 4 wurde bereits thematisiert, warum man bei einer linear-algebraischen Annäherung an das Erreichbarkeitsproblem schnell an seine Grenzen stößt, weil entweder Simulationsschritte erforderlich sind (Abschnitt 4.1) oder ohne Einschränkung des Lösungsraums (Abschnitt 4.2) kein Abbruchkriterium für den Algorithmus existiert.

Dieses Kapitel greift diese Thematik wieder auf. Der nächste Abschnitt beschreibt zunächst den Zusammenhang zwischen Lebendigkeit und der Reproduzierbarkeit der leeren Markierung in einem beliebigen S/T-Netz, indem das Netz um zwei zusätzliche Transitionen erweitert wird, die Anfangs- und Zielmarkierung repräsentieren. Aufbauend auf der Reproduzierbarkeit der leeren Markierung kann das allgemeine Erreichbarkeitsproblem verfeinert und *starke* von *schwacher* Erreichbarkeit unterschieden werden. Die Schwierigkeit, in diesen zwei neuen Erreichbarkeitsklassen die Reproduzierbarkeit der leeren Markierung allein über die Berechnung von T-Invarianten nachzuweisen, ist in den Kapiteln 5.3 und 5.4 beschrieben.

Unter Zuhilfenahme der Eigenschaften bestimmter Netzklassen (Synchronisationsgraphen) und Netzeigenschaften (S-Invarianten) wird abschließend — als Grundlage für die Kapitel 6 und 7 — ein Verfahren zur Berechnung der minimalen Kreise in einem S/T-Netz und dessen Transformation in ein zyklensfreies Netz vorgestellt.

## 5.1 Lebendigkeit und Reproduzierbarkeit der leeren Markierung

Die Fragestellung, ob eine Markierung  $M_B$  aus einer Markierung  $M_A$  erreichbar ist, wurde bereits in den Abschnitten 4.1 (durch Hinzufügen einer weiteren Transition) und 4.2 (durch Erweiterung um zwei Transitionen) in die Berechnung von T-Invarianten transformiert. Letztere Methode führt sogar dazu, dass durch die zwei zusätzlichen Transitionen nach Schaltfolgen gesucht wird, die die leere Markierung reproduzieren.

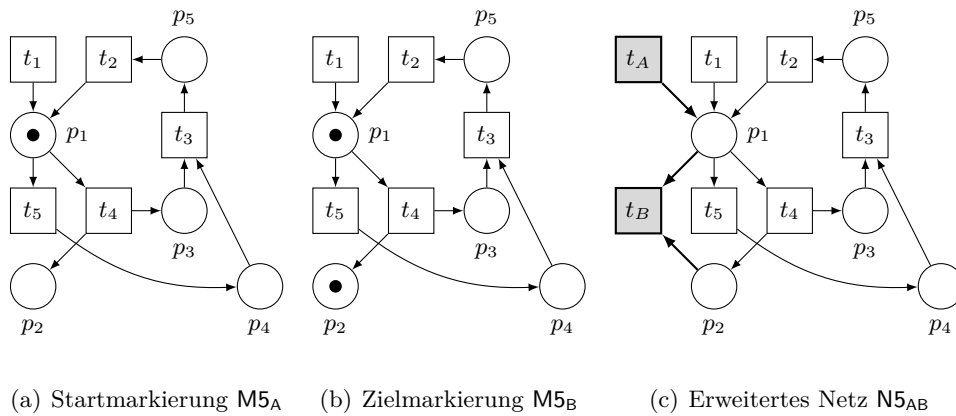
Dieses Verfahren wird im Detail in [Lau02] beschrieben und seine Korrektheit bewiesen. Die Erweiterung eines Petri-Netzes um diese zwei weiteren Transitionen wollen wir nun zunächst formal definieren:

**Definition 5.1 (erweitertes S/T-Netz  $N_{AB}$ )** Ein erweitertes S/T-Netz  $N_{AB}$  wird aus einem S/T-Netz  $N = (S, T, F, W)$  mit Anfangsmarkierung  $M_A$  und Zielmarkierung  $M_B$  folgendermaßen gebildet:  $N_{AB} := (S, T^*, F^*, W)$  mit:

- $T^* := T \cup \{t_A, t_B\}$
- $\forall p \in S, M_A(p) \geq 1 : F^* := F \cup \{(t_A, p)\}; W((t_A, p)) := M_A(p);$
- $\forall p \in S, M_B(p) \geq 1 : F^* := F \cup \{(p, t_B)\}; W((p, t_B)) := M_B(p);$

Die Anfangsmarkierung von  $N_{AB}$  ist die leere Markierung  $\emptyset$ . ■

Eine zusätzliche Transition  $t_A$  legt also so Marken auf die Stellen von  $N_{AB}$ , dass die Anfangsmarkierung  $M_A$  erzeugt wird. Eine weitere Transition  $t_B$  nimmt die Marken wieder herunter, die der Zielmarkierung  $M_B$  entsprechen. Im Netz  $N$  von Markierung  $M_A$  nach  $M_B$  zu gelangen ist also nun gleichzusetzen mit dem Ziel, im erweiterten Netz  $N_{AB}$  die leere Markierung zu reproduzieren, indem  $t_A$  und  $t_B$  jeweils genau einmal schalten. In Abbildung 5.1 sind ein S/T-Netz  $N_5$  mit Anfangs- und Zielmarkierung  $M_{5A}$  und  $M_{5B}$  und das zugehörige erweiterte Netz  $N_{5AB}$  dargestellt.



**Abbildung 5.1:** Netz  $N_5$  und erweitertes Netz  $N_{5AB}$

Die Reproduzierbarkeit der leeren Markierung in einem beliebigen S/T-Netz kann vom Vorhandensein einer nichtnegativen T-Invariante  $\vec{r}$  und der Trap- und Deadlockfreiheit der Netzdarstellung von  $\vec{r}$  abhängig gemacht werden (vgl. [Lau02]):

**Theorem 5.1 (Reproduzierbarkeit der leeren Markierung)** Sei  $N$  ein  $S/T$ -Netz und  $\vec{r}$  eine nichtnegative  $T$ -Invariante von  $N$ . Dann ist die leere Markierung  $\emptyset$  in  $N$  durch Realisierung von  $k \cdot \vec{r}$  für ein  $k \in \mathbb{N}$  reproduzierbar  $\Leftrightarrow$  Die Netzdarstellung  $N_{\vec{r}}$  von  $\vec{r}$  enthält weder Traps noch Deadlocks. ■

**Lemma 5.1 (aktiviert unter der leeren Markierung)** Jede Transition, die unter der leeren Markierung als Startmarkierung im  $S/T$ -System  $(N, \emptyset)$  schalten kann, kann beliebig oft in  $(N, \emptyset)$  schalten. ■

Wendet man Theorem 5.1 auf das Netz in Abbildung 5.1(c) an, so erkennt man bereits für  $\vec{r} = \vec{1}$  und  $k = 1$  die Reproduzierbarkeit der leeren Markierung. Die Netzdarstellung dieser  $T$ -Invariante entspricht  $N5_{AB}$  und hat weder Deadlocks noch Traps. Auch die Aussage von Lemma 5.1 ist leicht ersichtlich: Die Transitionen  $t_A$  und  $t_1$  sind als Randtransitionen in  $(N5_{AB}, \emptyset)$  aktiviert und können beliebig oft schalten.

Das Vorhandensein von Deadlocks und Traps in einem  $S/T$ -System  $(N, \emptyset)$  hat zudem direkte Auswirkungen auf dessen Lebendigkeit; Details und Beweise sind erneut in [Lau02] zu finden:

**Theorem 5.2 (Lebendigkeit, Deadlockfreiheit)** Das  $S/T$ -System  $(N, \emptyset)$  ist lebendig  $\Leftrightarrow N$  enthält keinen Deadlock. ■

**Korollar 5.1** Das  $S/T$ -System  $(N, \emptyset)$  ist lebendig in beide Richtungen  $\Leftrightarrow N$  enthält weder Traps noch Deadlocks. ■

Lebendigkeit und Reproduzierbarkeit der leeren Markierung in einem solchen  $S/T$ -System  $(N, \emptyset)$  hängen also zusammen — alle Transitionen im Netz müssen immer wieder schalten können, auch  $t_A$  und  $t_B$  —, die Erreichbarkeit ist aber an die in Theorem 5.1 aufgeführten Bedingungen geknüpft: Möglicherweise müssen  $t_A$  und  $t_B$  in  $\vec{r}$  zur Reproduktion der leeren Markierung ungleich oft und mehr als einmal feuern. Im nächsten Abschnitt werden nun zwei neue Begriffe eingeführt, die die Definition der Erreichbarkeit unter diesem Gesichtspunkt präzisieren.

## 5.2 Starke und schwache Erreichbarkeit

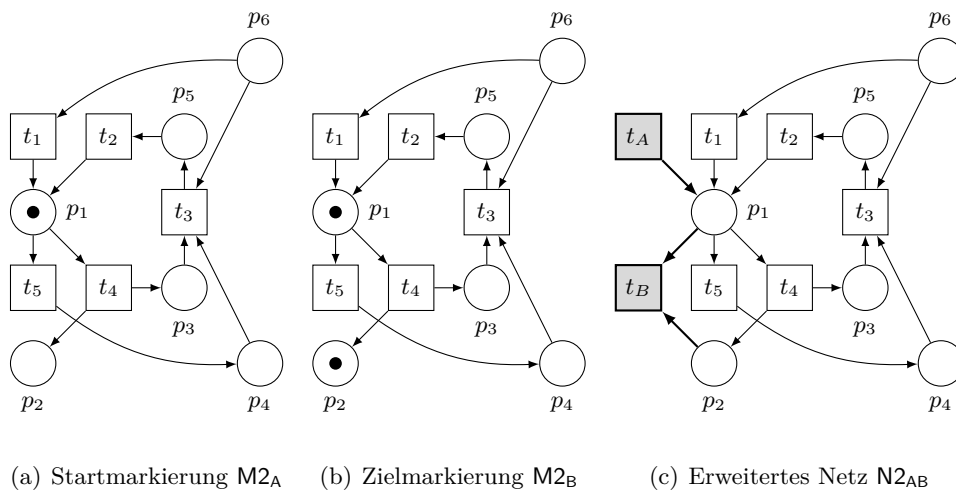
Aufbauend auf der Reproduzierbarkeit der leeren Markierung definierte Kurt Lautenbach die folgenden Begriffe der *starken* und *schwachen* Erreichbarkeit, durch die das Erreichbarkeitsproblem in zwei verschiedene Problemklassen verfeinert wird. Die „Chance“, dass eine Schaltsequenz von  $M_A$  nach  $M_B$  existieren könnte, wird als die Klasse der *schwach erreichbaren Fälle* definiert:

**Definition 5.2 (schwache Erreichbarkeit)** Sei  $N_{AB}$  ein erweitertes S/T-Netz gemäß Definition 5.1. Die Markierung  $M_B$  heißt von  $M_A$  aus schwach erreichbar

$\Leftrightarrow$

$N_{AB}$  besitzt eine nichtnegative T-Invariante  $\vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$ , deren Netzdarstellung weder Deadlocks noch Traps enthält und die für ein  $k \cdot \vec{r}$ ,  $k \in \mathbb{N}$  realisierbar ist. ■

Für ein beliebiges Netz erweitert man dieses also zunächst um  $t_A$  und  $t_B$  und überprüft, ob eine T-Invariante  $\vec{r}$  mit den gewünschten Eigenschaften existiert. Für Netz  $N_{2AB}$  (vgl. S. 14) in Abbildung 5.2 ist schwache Erreichbarkeit gegeben. Es besitzt genau eine minimale T-Invariante  $\vec{r} = (1 \ 1 \ 1 \ 1 \ 1 \ 1)^T$ . Ist diese aber auch realisierbar? Die Klasse der *stark erreichbaren* Fälle (als Unterklasse der schwach erreichbaren Fälle) beschäftigt sich mit dieser Realisierbarkeit von T-Invarianten:



**Abbildung 5.2:** Netz  $N_2$  – schwache Erreichbarkeit

**Definition 5.3 (starke Erreichbarkeit)** Sei  $N_{AB}$  ein S/T-Netz gemäß Definition 5.1, in dem schwache Erreichbarkeit gegeben ist. Die Markierung  $M_B$  heißt von  $M_A$  aus stark erreichbar

$\Leftrightarrow$

$N_{AB}$  besitzt eine nichtnegative, realisierbare T-Invariante  $\vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$ , deren Netzdarstellung weder Deadlocks noch Traps enthält. ■

Für Netz N2 ist keine starke Erreichbarkeit gegeben, da Transition  $t_A$  gemäß Definition 5.3 nur einmal schalten darf. Unabhängig davon, ob danach  $t_4$  oder  $t_5$  feuert: das Netz ist nach dem nächsten Schaltvorgang tot. Hier zeigt sich deutlich die in Theorem 5.1 beschriebene Eigenschaft: die leere Markierung ist nicht für  $1 \cdot \vec{r}$ , jedoch für  $2 \cdot \vec{r}$  reproduzierbar; alle Transitionen müssen also zweimal schalten.

### 5.3 Nachweis schwacher Erreichbarkeit mittels T-Invarianten

Wenn ein Algorithmus ausgehend von T-Invarianten auf starke Erreichbarkeit untersuchen soll, muss schwache Erreichbarkeit zuvor nachgewiesen worden sein. Eine T-Invariante mit den gewünschten Eigenschaften zu finden ist jedoch keine triviale Angelegenheit, da sie nicht unbedingt minimal sein muss. Es kann sich demnach auch um eine Linearkombination minimaler T-Invarianten handeln. Grundsätzlich lässt sich eine solche T-Invariante über die Berechnung von Erzeugendensystemen und manchmal auch aus der Lösung einer linearen Optimierungsaufgabe gewinnen. Die Herausforderungen beider Verfahren werden in den nächsten beiden Abschnitten diskutiert.

#### 5.3.1 Verwendung von Erzeugendensystemen

Der in Abschnitt 3.1.2.1 vorgestellte Algorithmus 3.1 berechnet ein Level 3-Erzeugendensystem für ein gegebenes Netz  $N_{AB}$ . Jedoch ist aus diesem Erzeugendensystem möglicherweise nicht erkennbar, ob eine T-Invariante mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  existiert. Das S/T-Netz  $N_{6AB}$  in Abbildung 5.3 hat beispielsweise folgende minimale T-Invarianten:

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_A$	$t_B$
$\vec{r}^1$	0	0	14	7	7	0	0	3
$\vec{r}^2$	126	63	21	0	14	0	0	6
$\vec{r}^3$	63	0	21	0	14	0	63	6
$\vec{r}^4$	0	0	0	0	14	21	0	3

Keiner der Generatoren erfüllt die Anforderung für schwache Erreichbarkeit. Zudem

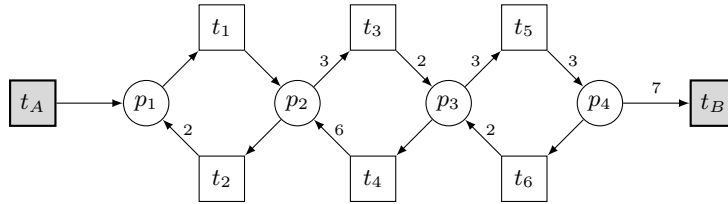


Abbildung 5.3: Netz  $N_{6AB}$

ist auf den ersten Blick nicht ersichtlich, ob sich aus diesen T-Invarianten eine Linearkombination mit den erforderlichen Schalthäufigkeiten für  $t_A$  und  $t_B$  bilden lässt. Zur Erinnerung: Als Faktoren für die Linearkombination sind Zahlen aus  $\mathbb{Q}^+$  zugelassen und die Kombinationsmöglichkeiten dadurch vielfältig. Erschwerend kommt hinzu, dass in keiner minimalen T-Invariante die Schalthäufigkeiten von  $t_A$  und  $t_B$  gleich sind, was die Suche vereinfachen würde. Letzteres Problem lässt sich durch eine zusätzliche Stelle  $p_C$  lösen, die dem Netz folgendermaßen hinzugefügt wird:

**Definition 5.4 (erweitertes S/T-Netz  $N_{ABC}$ )** Ein erweitertes S/T-Netz  $N_{ABC}$  wird aus einem S/T-Netz  $N_{AB} = (S, T, F, W)$  folgendermaßen gebildet:

$N_{ABC} := (S^*, T, F^*, W)$  mit

- $S^* := S \cup \{p_C\}$ ,
- $F^* := F \cup \{(t_B, p_C), (p_C, t_A)\}$ ,
- $W((t_B, p_C)) = W((p_C, t_A)) := 1$ .

Die Anfangsmarkierung  $M_C$  von  $N_{ABC}$  besteht aus einer Marke auf  $p_C$ . ■

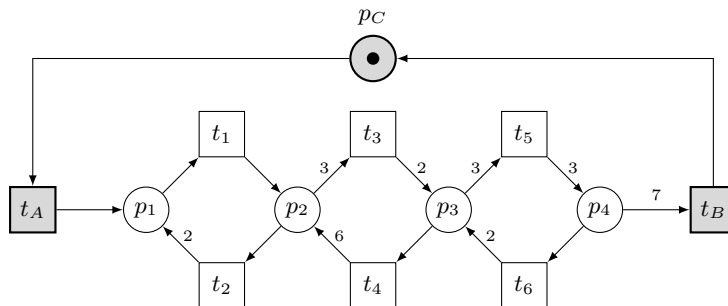


Abbildung 5.4: Netz  $N_{6ABC}$  mit Anfangsmarkierung  $M_{6C}$



Berechnet man nun für  $N6_{ABC}$  erneut das Level 3-Erzeugendensystem, so erhält man folgende minimale T-Invarianten:

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_A$	$t_B$
$\vec{r}^1$	9	0	41	19	21	0	9	9
$\vec{r}^2$	120	57	21	0	14	0	6	6
$\vec{r}^3$	9	0	3	0	40	57	9	9

Trotzdem erhält man aber keine Aussage darüber, ob eine T-Invariante mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  existiert. Man könnte nun bei dieser geringen Anzahl von Generatoren versuchen, eine passende Linearkombination durch „Ausprobieren“ zu finden. Besser ist jedoch, auf die Berechnung eines Level 4-Erzeugendensystems zurückzugreifen. Der Vorteil dabei sind die Faktoren aus  $\mathbb{N}$  für die Linearkombinationen: Es können nur ganzzahlige Vielfache minimaler T-Invarianten addiert werden. Falls eine T-Invariante mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  existiert, muss sie aufgrund der neuen Stelle  $p_C$  auf jeden Fall unter den Level 4-Generatoren zu finden sein. Werden keine passenden T-Invarianten gefunden — auch nicht als Linearkombination von Level 4-Generatoren —, so liegt keine schwache Erreichbarkeit vor.

Es bleibt noch zu bedenken, dass der Berechnungsaufwand des  $\mathbb{N}$ -Erzeugendensystems erheblich größer sein kann als die Ermittlung minimaler T-Invarianten. Insbesondere bei Netz  $N6_{ABC}$ , dessen  $\mathbb{Q}^+$ -Generatoren bereits große ganzzahlige Werte enthalten, arbeitet Algorithmus 3.2 äußerst ineffektiv: Immer wieder muss der gleiche Einheitsvektor hinzuaddiert und dazu stets ein Element vom Stack genommen und wieder darauf gelegt werden. Jedoch lässt sich der Algorithmus durch eine zusätzliche Abfrage verkürzen. Es sind nur T-Invarianten interessant, in denen — auch in Zwischenschritten der Berechnung — niemals  $\vec{r}(t_A), \vec{r}(t_B) > 1$  gilt. Übergibt man Algorithmus 3.2 (S. 25) neben der Inzidenzmatrix noch die Zeilenindizes  $tAIndex$  von  $t_A$  und  $tBIndex$  von  $t_B$  im Lösungsvektor, in diesem Fall also 7 und 8, so kann vor Zeile 19 noch folgende Abfrage eingefügt werden:

```

if (( $i = tAIndex$  or  $i = tBIndex$ ) and  $\vec{t}(i) \neq 0$ ) then
    continue;
endif

```

Dadurch werden nur Lösungsvektoren berechnet, in denen die Komponenten  $\vec{r}(t_A), \vec{r}(t_B)$  nur Werte  $\leq 1$  annehmen können. Falls im aktuell betrachteten Vektor an einer dieser Positionen schon der Wert 1 gesetzt ist, wird diese Komponente übersprungen („continue“), im Sinne des Algorithmus quasi „eingefroren“. Die  $\mathbb{N}$ -Generatoren von

Netz  $N_{6ABC}$ , die diese Bedingung erfüllen, sind:

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_A$	$t_B$
$\vec{r}^1$	5	2	1	0	4	5	1	1
$\vec{r}^2$	5	2	3	1	3	2	1	1

Zum Test auf schwache Erreichbarkeit wird die Stelle  $p_C$  wieder gelöscht, das Netz also auf  $N_{AB}$  zurückgesetzt. Die Netzdarstellung von  $\vec{r}^1$  enthält weder Deadlocks noch Traps und weist bereits schwache Erreichbarkeit nach. Allerdings ist diese T–Invariante nicht realisierbar: Nach dem Schalten der Transitionen  $t_1$  bis  $t_3$  ist das Netz tot, ta  $t_A$  nur einmal feuern darf. Der Generator  $\vec{r}^2$  ist allerdings realisierbar und würde starke Erreichbarkeit nachweisen. Eine mögliche Schaltsequenz wäre:

$$t_A t_1 t_2 (2 \cdot t_1) t_2 (2 \cdot t_1) t_3 t_4 (2 \cdot t_3) t_5 (2 \cdot t_6) (2 \cdot t_5) t_B$$

Insgesamt kann der Test auf schwache Erreichbarkeit wie in Algorithmus 5.1 dargestellt ablaufen. Eingabe ist die Inzidenzmatrix eines erweiterten Netzes  $N_{ABC}$ . Die Erzeugendensysteme werden wie gewohnt durch die Algorithmen 3.1 und 3.2 (mit der oben angegebenen Modifizierung) ermittelt. Gegebenenfalls reicht aber ein Generator  $\vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  nicht aus, da seine Netzdarstellung noch Deadlocks oder Traps enthält. Möglicherweise müssen noch T–Invarianten  $\vec{x}$  mit  $\vec{x}(t_A) = \vec{x}(t_B) = 0$  hinzuaddiert werden. Hierbei genügt es T–Invarianten  $\vec{x}$  zu wählen, die in der gleichen Zusammenhangskomponente liegen, also einen Einfluss auf die gleiche Menge von Stellen haben. Der Relationsgraph wird wie bei Kostins Verfahren auf Seite 33 erzeugt.

**Algorithmus 5.1:** ÜBERPRÜFUNG AUF SCHWACHE ERREICHBARKEIT

```

Eingabe:  $[N_{ABC}]$  – Inzidenzmatrix von Netz  $N_{ABC}$ ,  $[N_{AB}]$  – Inzidenzmatrix von Netz  $N_{AB}$ 
Ausgabe: ("schwach erreichbar", T–Invariante  $\vec{r}$ ) oder ("nicht schwach erreichbar")
1  $L3 :=$  Liste aller Level 3–Generatoren von  $[N_{ABC}]$ ;
2 if ( $L3 = \emptyset$ ) or ( $\vec{r}(t_A) = \vec{r}(t_B) = 0 \quad \forall \vec{r} \in L3$ ) then
3   return ("nicht schwach erreichbar")
4 else ▶ L3 nach passenden T–Invarianten durchsuchen
5    $L3_{AB} :=$  Liste aller T–Invarianten  $\vec{r}$  in  $L3$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$ ;
6   if ( $L3_{AB} \neq \emptyset$ ) then
7     if (Netzdarstellung  $N_{AB\vec{r}}$  mindestens eines Vektors in  $L3_{AB}$  hat keine Deadlocks und Traps)
8       then return ("schwach erreichbar",  $\vec{r}$ );
9     else
10      for  $i := 1$  to  $|L3_{AB}|$  do ▶ L3–Invarianten kombinieren

```

```

10      $\vec{r} := L3_{AB}[i];$  ▶  $i$ -te T-Invariante auswählen
11      $L3_0 :=$  Liste aller T-Invarianten  $\vec{x}$  in  $L3$  mit  $\vec{x}(t_A) = \vec{x}(t_B) = 0$ , die in der
        Zusammenhangskomponente von  $\vec{r}$  liegen;
12     if ( $L3_0 \neq \emptyset$ ) then
13         for  $j := 1$  to  $|L3_0|$  do
14              $\vec{r} := \vec{r} + L3_0[j];$ 
15             if (Netzdarstellung  $N_{AB\vec{r}}$  hat keine Deadlocks und Traps) then
16                 return ("schwach erreichbar",  $\vec{r}$ );
17             endif
18         endfor
19     endif
20 endfor
21 endif
22 endif
23 ▶ Suche in  $L3$  war ergebnislos
24  $L4 :=$  Liste aller Level 4-Generatoren von  $[N_{ABC}]$  mit  $\vec{r}(t_A) = \vec{r}(t_B) \leq 1$ ;
25  $L4_{AB} :=$  Liste aller Level 4-Generatoren von  $[N_{ABC}]$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$ ;
26 if ( $L4_{AB} \neq \emptyset$ ) then
27     if (Netzdarstellung  $N_{AB\vec{r}}$  mindestens eines Vektors in  $L4_{AB}$  hat keine Deadlocks und Traps)
        then return ("schwach erreichbar",  $\vec{r}$ );
28     else
29         for  $i := 1$  to  $|L4_{AB}|$  do ▶ L4-Invarianten kombinieren
30              $\vec{r} := L4_{AB}[i];$ 
31              $L4_0 :=$  Liste aller T-Invarianten  $\vec{x}$  in  $L4$  mit  $\vec{x}(t_A) = \vec{x}(t_B) = 0$ , die in der
                Zusammenhangskomponente von  $\vec{r}$  liegen;
32             if ( $L4_0 \neq \emptyset$ ) then
33                 for  $j := 1$  to  $|L4_0|$  do
34                      $\vec{r} := \vec{r} + L4_0[j];$ 
35                     if (Netzdarstellung  $N_{AB\vec{r}}$  hat keine Deadlocks und Traps) then
36                         return ("schwach erreichbar",  $\vec{r}$ );
37                     endif
38                 endfor
39             endif
40         endfor
41     endif
42     return ("nicht schwach erreichbar");
43 else ▶  $L4_{AB} = \emptyset$ 
44     return ("nicht schwach erreichbar");
45 endif
46 endif

```

Ob es — wie in Algorithmus 5.1 — sinnvoll ist, zuerst Level 3- und danach Level 4-Generatoren zu berechnen, hängt konkret vom untersuchten Netz ab. Prinzipiell ist es ausreichend, nur die T-Invarianten des  $\mathbb{N}$ -Erzeugendensystems mit  $\vec{r}(t_A) = \vec{r}(t_B) \leq 1$  zu ermitteln und auf die Level 3-Generatoren zu verzichten.

**Beispiel 5.1 (schwache Erreichbarkeit)** Wendet man den Algorithmus auf das um  $t_A$  und  $t_B$  erweiterte Netz  $N_4$  von Seite 38 an, so muss dieses zunächst um die zusätzliche Transition  $p_C$  erweitert werden (vgl. Abbildung 5.5). Im ersten Schritt

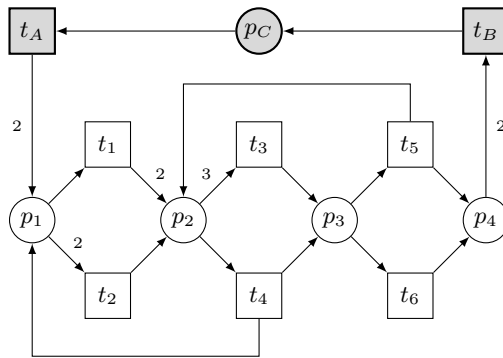


Abbildung 5.5: Netz  $N_{4ABC}$

werden nun dessen Level 3-Generatoren berechnet:

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_A$	$t_B$
$\vec{r}^1$	2	0	2	0	2	0	1	1
$\vec{r}^2$	0	2	0	2	0	2	1	1
$\vec{r}^3$	0	8	4	6	10	0	5	5
$\vec{r}^4$	5	0	3	1	0	4	2	2

Somit erhält man

$$L3_{AB} = \{(2\ 0\ 2\ 0\ 2\ 0\ 1\ 1)^T, (0\ 2\ 0\ 2\ 0\ 2\ 1\ 1)^T\}.$$

Keine der beiden Netzdarstellungen dieser T-Invarianten ist frei von Traps (Zeile 7 im Algorithmus) und es gibt keinerlei Generatoren, in denen  $t_A$  und  $t_B$  nicht schalten, daher ist  $L3_0 = \emptyset$  (Zeile 11 im Algorithmus) für alle Zusammenhangskomponenten. Dennoch kann noch nicht entschieden werden, ob schwache Erreichbarkeit vorliegt. Aus den zwei T-Invarianten in  $L3_{AB}$  lässt sich möglicherweise eine passende Linearkombination bilden. Um dies zu überprüfen, wird darum im nächsten Schritt das  $\mathbb{N}$ -Erzeugendensystem von Netz  $N_{4ABC}$  mit der zusätzlichen Bedingung  $\vec{r}(t_A) = \vec{r}(t_B) \leq 1$

(Zeile 24 im Algorithmus) berechnet, welche sich aus der auf Seite 49 diskutierten Änderung ergibt. Man erhält folgende Level 4-Generatoren:

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_A$	$t_B$
$\vec{r}^1$	0	2	0	2	0	2	1	1
$\vec{r}^2$	1	1	1	1	1	1	1	1
$\vec{r}^3$	2	0	2	0	2	0	1	1

Die Netzdarstellung  $N4_{AB\vec{r}^2}$  der neu hinzugekommenen T-Invariante  $\vec{r}^2$  (eine Linearkombination von  $\vec{r}^1$  und  $\vec{r}^3$ ) hat keine Deadlocks und Traps. Somit ist schwache Erreichbarkeit für  $N4_{AB}$  nachgewiesen.  $\square$

### Fazit:

Aus obigen Beispielen ist ersichtlich, dass schwache Erreichbarkeit spätestens über die Berechnung eines N-Erzeugendensystems mit den erwähnten Modifizierungen überprüfbar ist, falls nach der Ermittlung der Level 3-Generatoren keine passende T-Invariante gefunden wurde. Dennoch ist bei großen Netzen der Rechenaufwand keinesfalls zu unterschätzen.

### 5.3.2 Lösung einer ganzzahligen linearen Extremwertaufgabe

Betrachtet man das Finden einer geeigneten T-Invariante als ganzzahlige lineare Extremwertaufgabe, zeigt sich sehr schnell die Schwäche dieses Verfahrens. Ein passender Algorithmus liefert *keine* oder *genau eine* Lösung, und diese kann — muss aber nicht! — eine T-Invariante mit den gewünschten Eigenschaften sein. Für Netz  $N4_{ABC}$  sieht die Eingabe für Genstys Programmpaket [Gen05] in Mathematica folgendermaßen aus:

```

BAB[
  "Minimum",
  {{-1,-2,0,1,0,0,2,0},{2,1,-3,-1,1,0,0,0},{0,0,1,1,-1,-1,0,0},
   {0,0,0,0,1,1,0,-2},{0,0,0,0,0,0,-1,1}},
  {0,0,0,0,0},
  t[1]+t[2]+t[3]+t[4]+t[5]+t[6],
  {t[7]==1,t[8]==1}
]
```

Eingabe sind die Inzidenzmatrix  $[N_{4ABC}]$ , die rechte Seite  $\vec{0}$ , die zu minimierende Zielfunktion  $\tau[1] + \dots + \tau[6]$  (die Transitionen  $t_1$  bis  $t_6$  sollen in der Summe möglichst wenig schalten) und die Nebenbedingung  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  (hierbei durch  $\tau[7]$  und  $\tau[8]$  repräsentiert).

Die Ausgabe ist  $\{0, 2, 0, 2, 0, 2, 1, 1\}, \{6\}$ , also eine der minimalen T-Invarianten des vorherigen Abschnitts, und ein Zielfunktionswert von 6. Die Netzdarstellung dieser Lösung ist wie bereits erwähnt nicht frei von Traps, so dass sie keine schwache Erreichbarkeit nachweist. Zudem kann nur mit dieser einen Lösung nicht entschieden werden, ob man möglicherweise durch andere T-Invarianten doch noch ein positives Ergebnis erhält.

Für das Beispiel in Abbildung 5.4 auf Seite 48 wiederum erhält man die Lösung  $\{5, 2, 3, 1, 3, 2, 1, 1\}$ , also eine T-Invariante, die schwache Erreichbarkeit nachweist.

**Fazit:**

Die Berechnung von T-Invarianten auf diese Weise liefert nicht garantiert eine Lösung, deren Netzdarstellung frei von Deadlocks und Traps ist. Schwache Erreichbarkeit ist also nicht immer nachweisbar. Nur durch zusätzliche Constraints könnte der Lösungsraum weiter eingeschränkt und eine passende T-Invariante ermittelt werden. In der Regel ist es nicht trivial, die notwendigen Constraints zu finden, da sich die Schalthäufigkeiten der von  $t_A$  und  $t_B$  verschiedenen Transitionen kaum einschränken lassen.

## 5.4 Nachweis starker Erreichbarkeit mittels T-Invarianten

Vor der Untersuchung auf starke Erreichbarkeit wird im Folgenden immer davon ausgegangen, dass bereits schwache Erreichbarkeit nachgewiesen wurde. Es existiert also eine T-Invariante, deren Netzdarstellung frei von Deadlocks und Traps ist. Gemäß Theorem 5.1 ist mit dieser T-Invariante oder einem Vielfachen  $k$  die leere Markierung im Netz in jedem Fall reproduzierbar. Man kann also Simulationsschritte ausführen und zunächst  $t_A$  nur einmal schalten lassen. Reichen an irgendeiner Stelle die Token nicht aus, so muss  $t_A$  mehr als einmal feuern und  $t_B$  am Ende entsprechend oft. In diesem Fall ist keine starke Erreichbarkeit gegeben, da damit im ursprünglichen Netz (ohne  $t_A$  und  $t_B$ ) nur aus einem *Vielfachen* der Anfangsmarkierung ein *Vielfaches* der Zielmarkierung erreicht wurde. Lässt sich die starke Erreichbarkeit aber auch rein linear-algebraisch nachweisen? Wieder stehen die Berechnung von Erzeugendensystemen und das Lösen eines ganzzahligen linearen Programms zur Verfügung.

### 5.4.1 Verwendung von Erzeugendensystemen

Bereits in Abschnitt 5.3.1 wurde erklärt, dass allein durch die Berechnung eines  $\mathbb{Q}^+$ - oder  $\mathbb{N}$ -Erzeugendensystems nicht immer eine passende T-Invariante gefunden wird. Unter Umständen müssen Linearkombinationen gebildet werden. Genauso ist damit in der Regel auch keine starke Erreichbarkeit unmittelbar nachweisbar. Eine passende T-Invariante  $\vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  liefert noch keine Information darüber, ob nicht erst ein Vielfaches  $k \cdot \vec{r}$ ,  $k > 1$ , ein realisierbarer Schalthäufigkeitsvektor ist. Die T-Invariante  $\vec{r}$  des Beispiels in Abbildung 5.2 auf Seite 46 ist erst für  $2 \cdot \vec{r}$  realisierbar. Da aber die Generatoren der  $\mathbb{Q}^+$ - und  $\mathbb{N}$ -Erzeugendensysteme immer kanonisch — der größte gemeinsame Teiler aller ihrer Komponenten ist 1 — sind, kann  $2 \cdot \vec{r}$  niemals unter den direkt berechneten Lösungen sein.

**Fazit:**

Zum Nachweis starker Erreichbarkeit in einem Netz  $N_{AB}$  kommt man letztendlich nicht ohne Simulationsschritte aus, wenn zuvor schwache Erreichbarkeit über die Berechnung von Erzeugendensystemen und dem Nachweis einer passenden T-Invariante  $\vec{r}$  für das Netz  $N_{ABC}$  nachgewiesen wurde. Jedoch ist sichergestellt, dass ein Vielfaches  $k \cdot \vec{r}$  auf jeden Fall realisierbar ist. Von  $k$  abhängig weist man durch die Simulationsschritte demnach starke ( $k = 1$ ) oder nur schwache ( $k > 1$ ) Erreichbarkeit nach.

### 5.4.2 Lösung einer ganzzahligen linearen Extremwertaufgabe

Wie beim Nachweis schwacher Erreichbarkeit durch Lösen eines ganzzahligen linearen Programms ist auch in der Regel keine starke Erreichbarkeit unmittelbar überprüfbar. Es wird vom Programm entweder keine oder nur eine Lösung ermittelt, die zwar durch den Constraint  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  stets eine notwendige Bedingung erfüllt, jedoch in ihrer Netzdarstellung noch Deadlocks und/oder Traps enthalten kann.

**Fazit:**

Im günstigsten Fall erhält man eine geeignete T-Invariante, über die durch Simulationsschritte starke Erreichbarkeit bewiesen werden kann. In der Regel ist dies aber nicht der Fall.

### 5.4.3 Zusammenfassung

Die letzten beiden Abschnitte verdeutlichen, warum die Berechnung von Erzeugendensystemen oder das Lösen einer linearen Extremwertaufgabe für ein Netz  $N_{ABC}$  allgemein nicht zum Nachweis starker Erreichbarkeit ausreicht. Die gesuchte T-Invariante

muss nicht in der Lösungsmenge enthalten sein. Selbst wenn T–Invarianten in verschiedene Klassen sortiert und Linearkombinationen gebildet werden, gibt es kein Patentrezept für das schnelle Auffinden der gewünschten Lösung. Letztendlich kommt man ohne Simulation, die die Realisierbarkeit der T–Invariante überprüft, nicht aus. Der Grund hierfür ist nach wie vor das Vorhandensein von Kreisen im Netz und wann eine T–Invariante aktiviert ist: Die lineare Algebra kann dann nicht unterscheiden zwischen „ausreichend Marken haben und nehmen“ und „Marken nicht haben und deshalb leihen müssen und später zurückgeben“.

Wie lassen sich diese Probleme lösen? Es ist leicht ersichtlich, dass präzisere Informationen benötigt werden, um sowohl die T–Invarianten einzugrenzen als auch Simulationsschritte zu vermeiden. Die Erweiterung des Netzes zu einem Synchronisationsgraphen ist eine solche Möglichkeit. Aufgrund der für diese Netzklasse geltenden Eigenschaften kann das Netz dann trotz der noch vorhandenen Kreise einfacher für eine Erreichbarkeitsanalyse herangezogen werden. Eine entsprechende Idee wird in Kapitel 6 diskutiert. Ein zweiter Ansatz ist die Transformation des Netzes in ein zyklonfreies Äquivalent. Sind die Kreise eliminiert, so kann die Zustandsgleichung als hinreichendes Kriterium herangezogen werden. Verfahren zum Auffinden aller Kreise und zur Umformung des Netzes werden im nächsten Abschnitt beschrieben.

## 5.5 Zusätzlicher Informationsgewinn

Bei der Gewinnung zusätzlicher Information über ein S/T–Netz, beispielsweise die Anzahl seiner Kreise, ist es sinnvoll, das Netz zunächst in ein verhaltensgleiches, vereinfachtes Netz zu transformieren. Grundsätzlich wird in dieser Arbeit davon ausgegangen, dass das Netz *zusammenhängend* ist, es also für zwei beliebige Knoten  $x$  und  $y$  aus  $S \cup T$  einen ungerichteten Weg von  $x$  nach  $y$  gibt. Es gibt also keine vollständig voneinander getrennten Teilnetze. Zur Reduzierung des Rechenaufwands sollten zudem so wenig Knoten und Kanten wie möglich betrachtet werden müssen.

Ein erster Schritt, das Netz — falls möglich — unter Erhaltung der Verhaltensgleichheit um einige Knoten und Kanten zu reduzieren, kann durch die Elimination *einfacher Stellen* erfolgen:

**Definition 5.5 (einfache Stelle, nicht–einfache Stelle, Grenzstelle)** Sei  $N$  ein S/T–Netz. Eine Stelle  $p \in S$  ist eine

$$\begin{aligned} \text{einfache Stelle} \quad & \Leftrightarrow \quad |\bullet p| = |p \bullet| = 1 \text{ und } W((t_x, p)) = W((p, t_y)), \\ & t_x \in \bullet p, t_y \in p \bullet, t_x \neq t_y; \end{aligned}$$



nicht-einfache Stelle  $\Leftrightarrow |\bullet p| = |p\bullet| = 1$ ;  
 Grenzstelle  $\Leftrightarrow |\bullet p| > 1 \vee |p\bullet| > 1$ . ■

Einfache Stellen besitzen also *genau eine* Ein- und Ausgangskante mit gleichem Kantengewicht, haben nicht die gleiche Vor- und Nachtransition und reichen somit Marken im Netz lediglich weiter. Nicht-einfache Stellen besitzen auch je eine Eingangs- und Ausgangskante, sind aber bezüglich der Kantengewichte nicht eingeschränkt und können somit auch Schlingen jeglicher Art beschreiben. Grenzstellen sammeln und verteilen Marken im Netz.

Bei der Elimination der einfachen Stellen werden sowohl Vor- und Nachtransition zu einer neuen Transition als auch deren Vor- und Nachbereiche zusammengefasst. Gegebenenfalls dadurch entstandene mehrfache Kanten zwischen zwei Knoten werden durch eine Kante und die Summe aller Kantengewichte ersetzt. Hierbei gilt es allerdings zu beachten, dass diese Reduktion einfacher Stellen nur erlaubt ist, wenn dadurch keine neuen Deadlocks oder Traps im Netz entstehen, da sich sonst das Verhalten des Netzes ändern würde. Diesen Sachverhalt veranschaulicht Abbildung 5.6. Links ist das ursprüngliche Netz abgebildet, rechts dessen um die einfache Stelle  $p_5$  reduzierte Kopie. Durch das Entfernen von  $p_5$  wurden die Transitionen  $t_3$  und  $t_4$  verschmolzen und die von  $p_2$  ausgehenden Kanten durch eine Kante mit Gewicht 3 ersetzt. Dadurch ist ein neuer Trap  $\{p_2, p_4\}$  entstanden, denn

$$\{p_2, p_4\}^\bullet = \{t_2, t_3\} \subset \{t_A, t_1, t_2, t_3\} = \bullet\{p_2, p_4\}.$$

Im linken Netz ist Transition  $t_3$  durch je einmaliges Schalten von  $t_A$  und  $t_2$  aktiviert, im reduzierten Netz ist dies nicht möglich. Vor jeder Elimination einer einfachen Stelle muss also überprüft werden, ob neue Deadlocks oder Traps entstehen können. Algorithmus 5.2 beschreibt den Reduktionsvorgang, wobei die Überprüfung auf Deadlocks und Traps nicht explizit aufgeführt ist.

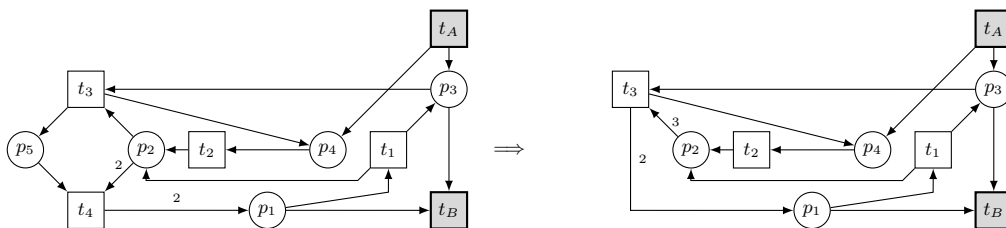


Abbildung 5.6: Unzulässige Reduktion der einfachen Stelle  $p_5$

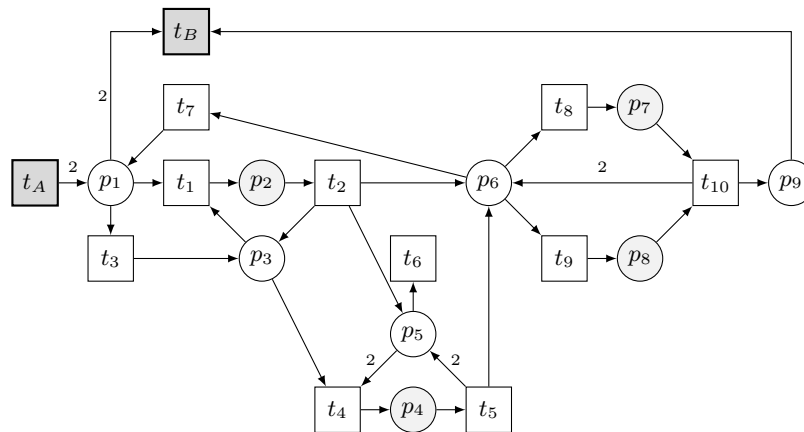
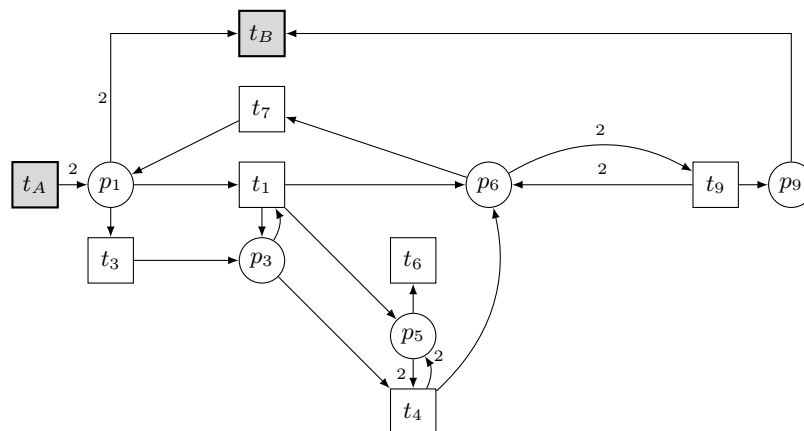
**Algorithmus 5.2:** ENTFERNEN EINFACHER STELLEN**Eingabe:** S/T-Netz  $N = (S, T, F, W)$ **Ausgabe:**  $R := N$  ohne alle einfachen Stellen, die eliminiert werden dürfen

```

1 for  $i := 1$  to  $|S|$  do
2   if ( $p_i$  ist einfache Stelle und der durch ihre Reduktion bedingte Umbau des Netzes erzeugt
3     keine neuen Deadlocks oder Traps im Netz) then
4      $t_x :=$  Vortransition von  $p_i$ ;
5      $t_y :=$  Nachtransition von  $p_i$ ;
6     if ( $t_x = t_A$  oder  $t_y = t_B$ ) then
7       continue;
8     endif
9     for all  $p_j \in t_y^\bullet$  do
10      Füge die Kante  $(t_x, p_j)$  mit Kantengewicht  $W((t_y, p_j))$  dem Netz hinzu;
11      Lösche die Kante  $(t_y, p_j)$ ;
12      if ( $|(t_x, p_j)| > 1$ )
13         $w :=$  Summe der Kantengewichte aller Kanten von  $t_x$  nach  $p_j$ ;
14        Füge eine neue Kante  $(t_x, p_j)$  mit Kantengewicht  $w$  hinzu.
15        Lösche alle anderen Kanten  $(t_x, p_j)$ ;
16      endif
17    endfor
18    for all  $p_j \in {}^\bullet t_y$  außer  $p_i$  do
19      Füge die Kante  $(p_j, t_x)$  mit Kantengewicht  $W((p_j, t_y))$  dem Netz hinzu;
20      Lösche die Kante  $(p_j, t_y)$ ;
21      if ( $|(p_j, t_x)| > 1$ )
22         $w :=$  Summe der Kantengewichte aller Kanten von  $p_j$  nach  $t_x$ ;
23        Füge eine neue Kante  $(p_j, t_x)$  mit Kantengewicht  $w$  hinzu.
24        Lösche alle anderen Kanten  $(p_j, t_x)$ ;
25      endif
26    endfor
27    Lösche  $p_i$ , seine aus- und eingehenden Kanten und  $t_y$ ;
28 endif
endfor

```

**Beispiel 5.2 (Elimination einfacher Stellen)** In Abbildung 5.7 sind die einfachen Stellen  $p_2, p_4, p_7$  und  $p_8$  in Netz  $N_{3AB}$  (das um  $t_A$  und  $t_B$  ergänzte Netz von Seite 32) grau unterlegt. Nach dem Reduktionsvorgang um diese vier Stellen und weitere vier Transitionen hat das Netz die in Abbildung 5.8 dargestellte Form.  $\square$

Abbildung 5.7: Netz  $N_{3AB}$ Abbildung 5.8: Netz  $R_3 := N_{3AB}$  ohne einfache Stellen

Bei allen weiteren Umformungs- oder Reduktionsschritten in Folgekapiteln wird nun davon ausgegangen, dass das zugrunde liegende Netz auf diese Art und Weise zu einem Netz  $R$  reduziert wurde.

### 5.5.1 Ermitteln minimaler Kreise über S-Invarianten

Die entscheidende Frage ist nun, wie man die Kreise im Netz findet. Zum einen lassen sie sich mit gängigen Graphenalgorithmus berechnen. Zum anderen können aber auch einige Eigenschaften der Petri-Netze selbst für die Berechnung der Kreise hilfreich sein. Es sind zunächst nur die minimalen Zyklen interessant, d.h. Zyklen mit minimaler Anzahl von Transitionen. Oft können zwei oder mehr dieser minimalen Kreise zu größeren vereint werden, da sie in Grenzstellen beginnen und enden. Ein guter Schritt ist es, diese Grenzstellen derart „aufzutrennen“, dass nur minimale Zyklen gefunden

werden. Dies gelingt durch die Transformation des Netzes in einen (erweiterten) Synchronisationsgraphen:

**Definition 5.6 (Synchronisationsgraph, Weg, Kreis)** *Ein Synchronisationsgraph ist ein S/T-Netz  $N = (S, T, F)$  mit der folgenden Eigenschaft:  $\forall p \in S : |\bullet p| = |p \bullet| = 1$ . Eine Folge  $\pi = (p_0, p_1, \dots, p_n)$  von Stellen in  $N$  heißt Weg der Länge  $n$ , falls für  $i = 1, 2, \dots, n$  gilt:  $p_{i-1} \bullet = \bullet p_i$ .  $\pi$  heißt Kreis in  $N$ , wenn gilt:*

- $\forall i, j \in N : 0 \leq i \leq n, 0 \leq j \leq n, i \neq j, p_i \neq p_j$
- $\bullet p_0 = \bullet p_n$ . ■

In einem Synchronisationsgraphen hat jede Stelle also genau eine eingehende und eine ausgehende Kante und alle Kantengewichte sind 1. Zwei wichtige Eigenschaften dieser Netzklasse beschreiben die folgenden Sätze:

**Satz 5.1** *Sei  $N$  ein Synchronisationsgraph und sei  $\pi = (p_0, p_1, \dots, p_n)$  ein Kreis von  $N$ . Dann gilt für alle Markierungen  $M \in [M_0\rangle$  :*

$$\sum_{i=0}^n M(p_i) = \sum_{i=0}^n M_0(p_i),$$

*d.h. die Markensumme ist für alle von der Anfangsmarkierung aus erreichbaren Markierungen immer gleich.* ■

**Satz 5.2** *Sei  $N$  ein Synchronisationsgraph.  $N$  ist lebendig  $\Leftrightarrow$  Jeder Kreis  $\pi$  von  $N$  enthält mindestens eine unter Startmarkierung  $M_0$  markierte Stelle.* ■

Die Kreise in einem Synchronisationsgraphen können folglich durch die Berechnung von S-Invarianten ermittelt werden (Satz 5.1). Auch Satz 5.2 kommt später noch zur Anwendung: Besitzt ein Synchronisationsgraph keine Kreise, so ist er damit automatisch lebendig: Jede Stelle hat aufgrund der Definition der Netzklasse auf jeden Fall eine Vortransition. In einem kreisfreien Synchronisationsgraphen muss es demnach mindestens eine Randtransition geben, die immer aktiviert ist und das Netz mit ausreichend vielen Marken versorgen kann und somit lebendig hält.

Man könnte nun ein S/T-Netz  $R$  in einen Synchronisationsgraphen transformieren, es müssten aber auch alle Kantengewichte auf 1 reduziert werden. Stattdessen genügt es, einen *erweiterten Synchronisationsgraphen* zu erzeugen:

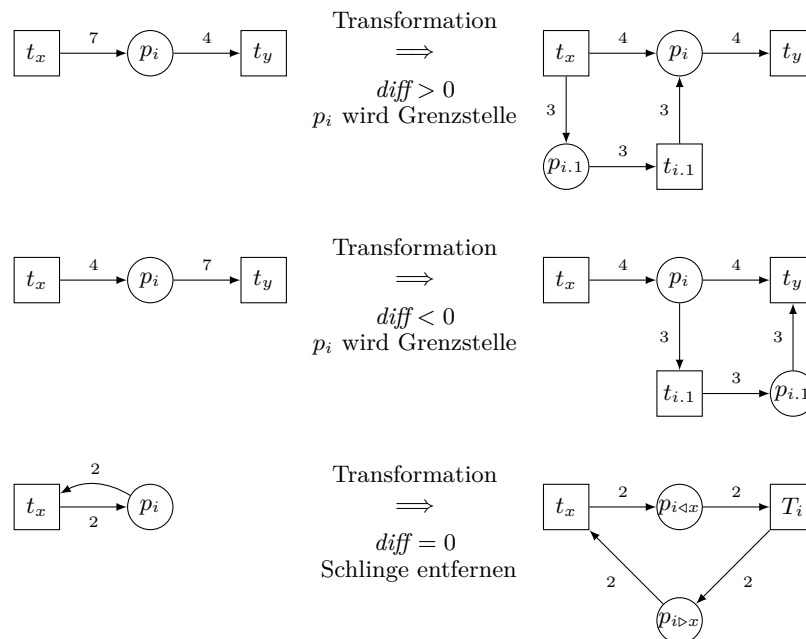
**Definition 5.7 (erweiterter Synchronisationsgraph)** Ein erweiterter Synchronisationsgraph ist ein  $S/T$ -Netz  $N = (S, T, F, W)$  mit folgenden Eigenschaften:

- $\forall p \in S : |\bullet p| = |p \bullet| = 1$
- $\forall p \in S, t_x \in \bullet p, t_y \in p \bullet : W((t_x, p)) = W((p, t_y)) \geq 1$  ■

Die Stellen eines erweiterten Synchronisationsgraphen sind also ausschließlich einfache Stellen. Durch mögliche Kantengewichte größer als 1 ist Satz 5.1 so nicht mehr zutreffend, es gilt die allgemeiner gefasste Eigenschaft für  $S$ -Invarianten  $\vec{i}$  dieses Netzes:

$$\forall M \in [M_0] : \vec{i} \circ M = \vec{i} \circ M_0$$

Nicht-einfache Stellen und Grenzstellen sollen nun ersetzt werden. Algorithmus 5.3 beschreibt das gesamte Verfahren der Transformation eines Netzes  $R$  in einen *erweiterten* Synchronisationsgraphen  $ES$ . Zunächst werden hier die nicht-einfachen Stellen durch zusätzliche Stellen und Transitionen entweder in Grenzstellen umgewandelt oder — falls sie Schlingen beschreiben — geeignet ersetzt. In Abbildung 5.9 sind die verschiedenen Fälle dieser Transformation mit konkreten Kantengewichten visualisiert.



**Abbildung 5.9:** Beispiel: Ersetzen nicht-einfacher Stellen

**Algorithmus 5.3:** TRANSFORMATION EINES S/T–NETZES IN EINEN ERWEITERTEN SYNCHRONISATIONSGRAPHEN

**Eingabe:** S/T–Netz  $R = (S, T, F, W)$

**Ausgabe:**  $ES := R$  transformiert in einen erweiterten Synchronisationsgraphen

```

1  for  $i := 1$  to  $|S|$                                      ▶ nicht-einfache Stellen ersetzen
2      if ( $p_i$  ist nicht–einfache Stelle) then
3           $t_x :=$  Vortransition von  $p_i$ ;
4           $t_y :=$  Nachtransition von  $p_i$ ;
5           $diff := W((t_x, p_i)) - W((p_i, t_y))$ ;
6          if ( $diff > 0$ ) then
7               $S := S \cup \{p_{i.1}\}$ ;
8               $T := T \cup \{t_{i.1}\}$ ;
9               $F := F \cup \{(t_x, p_{i.1}), (p_{i.1}, t_{i.1}), (t_{i.1}, p_i)\}$ ;
10              $W((t_x, p_{i.1})) = W((p_{i.1}, t_{i.1})) = W((t_{i.1}, p_i)) := diff$ ;
11              $W((t_x, p_i)) := W((p_i, t_y))$ ;
12         else if ( $diff < 0$ ) then
13              $S := S \cup \{p_{i.1}\}$ ;
14              $T := T \cup \{t_{i.1}\}$ ;
15              $F := F \cup \{(p_i, t_{i.1}), (t_{i.1}, p_{i.1}), (p_{i.1}, t_y)\}$ ;
16              $W((p_i, t_{i.1})) = W((t_{i.1}, p_{i.1})) = W((p_{i.1}, t_y)) := |diff|$ ;
17              $W((p_i, t_y)) := W((t_x, p_i))$ ;
18         else if ( $diff = 0$ ) then                             ▶ Schlinge
19              $T := T \cup \{T_i\}$ ;
20              $S := S \cup \{p_{i \triangleright x}, p_{i \triangleleft x}\}$ ;
21              $F := F \cup \{(t_x, p_{i \triangleleft x}), (p_{i \triangleleft x}, T_i), (T_i, p_{i \triangleright x}), (p_{i \triangleright x}, t_x)\}$ ;
22              $W((t_x, p_{i \triangleleft x})) := W((p_i, t_x))$ ;  $W((p_{i \triangleleft x}, T_i)) := W((p_i, t_x))$ ;
23              $W((T_i, p_{i \triangleright x})) := W((p_i, t_x))$ ;  $W((p_{i \triangleright x}, t_x)) := W((p_i, t_x))$ ;
24              $F := F \setminus \{(p_i, t_y), (t_x, p_i)\}$ ;  $S := S \setminus \{p_i\}$ ;
25         endif
26     endif
27 endif
28 for  $i := 1$  to  $|S|$  do
29     if ( $p_i$  ist Grenzstelle) then                             ▶ Grenzstellen ersetzen
30          $T := T \cup \{T_i\}$ ;
31         for all  $t_j \in \bullet p_i$  do
32              $S := S \cup \{p_{i \triangleleft j}\}$ ;
33              $F := F \cup \{(t_j, p_{i \triangleleft j}), (p_{i \triangleleft j}, T_i)\}$ ;
34              $W((t_j, p_{i \triangleleft j})) = W((p_{i \triangleleft j}, T_i)) := W((t_j, p_i))$ ;
35              $F := F \setminus \{(t_j, p_i)\}$ ;
36         endfor
37     for all  $t_j \in p_i^\bullet$  do

```

```

38     S := S ∪ {pi▷j};
39     F := F ∪ {(Ti, pi▷j), (pi▷j, tj)};
40     W((Ti, pi▷j)) = W((pi▷j, tj)) := W((pi, tj));
41     F := F \ {(pi, tj)};
42   endfor
43   S := S \ {pi};
44   endif
45 endfor

```

► Grenzstelle löschen

Für Netz R3 von Seite 59 erhält man das in Abbildung 5.10 dargestellte Netz ES3. Im nächsten Schritt werden nun dessen S-Invarianten, insgesamt neun Stück und in Tabelle 5.1 gelistet, berechnet. Diese beschreiben gemäß Satz 5.1 alle minimalen Kreise. Über diese Zyklen kann das ursprüngliche Netz nun in ein kreisfreies Äquivalent umgeformt werden. Ein passendes Verfahren beschreibt der nächste Abschnitt.

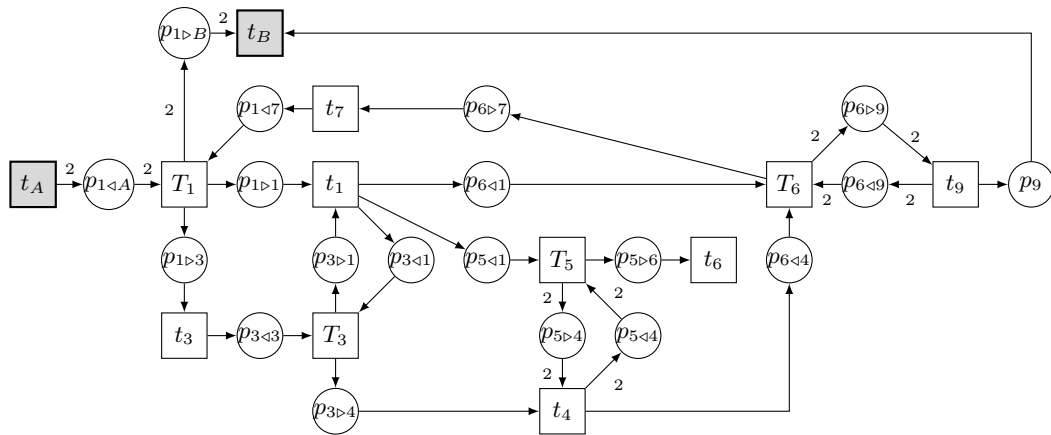


Abbildung 5.10: ES3 — erweiterter Synchronisationsgraph zu R3

$p$	9	1◁A	1▷3	1▷1	1◁7	1▷B	3◁3	3▷1	3◁1	3▷4	5◁1	5▷4	5◁4	5▷6	6▷6	6▷7	6◁1	6◁4	6◁9	6▷9
$i^1$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
$i^2$	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
$i^3$	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
$i^4$	0	0	1	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0
$i^5$	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
$i^6$	0	0	0	2	2	0	0	0	0	2	1	0	0	2	0	2	0	0	0	0
$i^7$	0	0	0	1	1	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0
$i^8$	0	0	1	0	1	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0
$i^9$	0	0	2	0	2	0	2	2	0	2	1	0	0	2	0	2	0	0	0	0

Tabelle 5.1: S-Invarianten des Netzes ES3

### 5.5.2 Umformung in ein zyklensfreies Netz

Die Positionen, an denen nun die für  $R$  ermittelten Kreise aufzuschneiden sind, damit man ein zyklensfreies Netz erhält, können ebenfalls über die S-Invarianten des zugehörigen Netzes  $ES$  ermittelt werden. Algorithmus 5.4 beschreibt in Pseudocode ein Verfahren zum Auffinden dieser Schnittstellen.

Hierbei wurde insbesondere Wert darauf gelegt, deren Anzahl gering zu halten: Im Regelfall laufen mehrere Kreise durch die gleiche Kante; wird diese aufgeschnitten, werden also sofort mehrere Zyklen unterbrochen. Bei diesem Algorithmus wollen wir o.B.d.A. stets die Kanten *hinter* Transitionen des ursprünglichen Netzes aufschneiden. Von Transitionen  $y$  ausgehende Kanten sind im Netz  $ES$  durch Stellen der Art  $p_{x \triangleleft y}$  repräsentiert oder führen zu einer nicht eliminierten einfachen Stelle  $p$ . Vor Stellen  $p_{x.1}$ , die durch die Transformation nicht-einfacher Stellen entstanden sind, soll nicht aufgeschnitten werden. Das Löschen einer Stelle  $p_{x \triangleleft y}$  in  $ES$  entspricht dem Löschen der Kante  $(t_y, p_x)$  in  $R$ . Die Anzahl der minimalen Kreise, die durch die Stellen  $p_{x \triangleleft y}$  bzw. noch vorhandene einfache Stellen verlaufen, wird über eine Häufigkeitstabelle ermittelt. Algorithmus 5.4 erwartet neben dem Netz  $ES$  noch zwei weitere Eingabeparameter:

$L$ :  $n$ -stellige Liste der S-Invarianten von  $ES$ ; wird stets aktuell gehalten.

$KH$ : Array aller einfachen Stellen ( $\neq p_{x.1}$ ) und Stellen der Art  $p_{x \triangleleft y}$  (als Schlüssel) und ihren Auftretishäufigkeiten  $h$  in den S-Invarianten (als Werte). Ein Element von  $KH$  ist somit ein Tupel  $(p, h)$ , wobei  $KH(p)$  den zu  $p$  gehörigen Wert  $h$  liefere.  $KH$  wird in Bezug auf die Liste  $L$  stets aktuell gehalten (`update(KH)`).

Einzelne Schritte des Ablaufs müssen bei einer Implementation noch spezifiziert werden, jedoch verdeutlicht der Pseudocode die Zusammenhänge zwischen den Netzen  $R$  und  $ES$  besser und wurde darum hier bevorzugt.

**Algorithmus 5.4:** ERMITTELN DER AUFZUSCHNEIDENDEN KANTEN ANHAND VON S-INvarianten

<b>Eingabe:</b> Netz $ES$ , $L$ , $KH$ .	
<b>Ausgabe:</b> Liste $K$ der Stellen, die in $ES$ gelöscht werden.	
1	$K := \emptyset$ ;
2	<b>for</b> $e := 1$ <b>to</b> $n$ <b>do</b>
3	$m :=$ Anzahl der Einträge in $L[e]$ , die $> 0$ sind;
4	<b>if</b> ( $m = 2$ mit $L[e] = \{p_i, p_j\}$ ) <b>then</b>
5	<b>if</b> ( $p_i$ vom Typ $p_{x \triangleleft y}$ und $p_j$ vom Typ $p_{x \triangleright y}$ ) <b>then</b>
6	$K := K \cup \{p_i\}$ ; <span style="float: right;">► Schlinge.</span>



```

7   else
8      $K := K \cup \{p_i\};$  ▶ Kreis aus zwei einfachen Stellen.
9   endif
10  endif
11  endfor
12  while ( $ES \setminus K$  hat noch Kreise) do ▶ Stellen  $K$  in  $ES$  löschen und weiterrechnen.
13     $L := S$ -Invarianten von  $ES \setminus K$ ;
14    update ( $KH$ );
15    Sei  $max$  die aktuell größte Auftrittshäufigkeit  $h$  in  $KH$ ;
16    if ( $max$  kommt mehr als einmal in  $KH$  vor) then
17      if ( $max = KH(p_{x \triangleleft y})$  irgendeiner Stelle  $p_{x \triangleleft y}$  mit Parallelstelle  $p_{x \triangleleft x.1}$ ) then
18         $K := K \cup \{p_{x \triangleleft y}, p_{x \triangleleft x.1}\};$  ▶ Nicht-einfache Stellen zuerst abarbeiten.
19      else if ( $max = KH(p_{x \triangleleft y})$  irgendeiner Stelle  $p_{x \triangleleft y}$ ) then
20         $K := K \cup \{p_{x \triangleleft y}\};$  ▶ Dann immer vor normalen Grenzstellen aufschneiden.
21      else if ( $max = KH(p)$  einer einfachen Stelle  $p$ ) then
22         $K := K \cup \{p\};$  ▶ Erst dann vor einfachen Stellen aufschneiden.
23      endif
24      else if (es gibt genau einen größten Wert  $max = KH(p)$  und  $p$  ist vom Typ  $p_{x \triangleleft y}$  mit
25        Parallelstelle  $p_{x \triangleleft x.1}$ ) then
26         $K := K \cup \{p_{x \triangleleft y}, p_{x \triangleleft x.1}\};$  ▶ Nicht-einfache Stellen zuerst berücksichtigen.
27      else
28         $K := K \cup \{p\};$  ▶ Sonst Stelle  $p$  zu  $K$  hinzufügen.
29      endif
30    endwhile

```

Im ersten Schritt werden zunächst genau die Stellen als zu eliminierende Stellen gespeichert, die Schlingen oder einen Kreis mit zwei einfachen Stellen im Netz  $R$  repräsentieren. Falls damit noch nicht alle Kreise im Netz aufgeschnitten sind, werden im zweiten Schritt nach und nach diejenigen Stellen ausgewählt, durch die laut Häufigkeitstabelle die meisten Kreise verlaufen. Falls zwei oder mehr Werte in der Tabelle gleich sind, werden die Stellen in einer bestimmten Reihenfolge abgearbeitet. Es gibt in  $R$  drei Klassen von Stellen, deren eingehende Kanten in  $ES$  wie folgt repräsentiert werden:

- nicht-einfache Stellen: durch  $p_{x \triangleleft y}$  mit einer „Parallelstelle“  $p_{x \triangleleft x.1}$ ,
- Grenzstellen: durch  $p_{x \triangleleft y}$ ,
- einfache Stellen: unverändert.

Nicht-einfache Stellen  $p_x$  wurden bei der Konstruktion des erweiterten Synchronisationsgraphen — wie in Abbildung 5.9 auf Seite 61 dargestellt — gegebenenfalls durch

eine zusätzliche Stelle  $p_{x.1}$  und Transition  $t_{x.1}$  zunächst in Grenzstellen umgewandelt und dann durch Transitionen  $T_x$  und deren Ein- und Ausgangsstellen ersetzt. Durch diese Aufteilung der ursprünglichen Stelle verlaufen durch  $p_{x \triangleleft x.1}$  und  $p_{x \triangleleft y}$  gleich viele minimale S-Invarianten (falls vorhanden). Wenn der Fall  $max = KH(p_{x \triangleleft y})$  eintritt und eine Parallelstelle ( $p_{x \triangleleft x.1}$ ) existiert, müssen also beide Stellen im Synchronisationsgraphen zu  $K$  hinzugefügt werden, da nur beide Stellen gemeinsam die Kante  $t_y \rightarrow p_x$  im Originalnetz repräsentieren. Der Sachverhalt ist in Abbildung 5.11 veranschaulicht. Die minimalen S-Invarianten  $\vec{i}$  und Häufigkeitswerte für alle Stellen der Art  $p_{x \triangleleft y}$  für

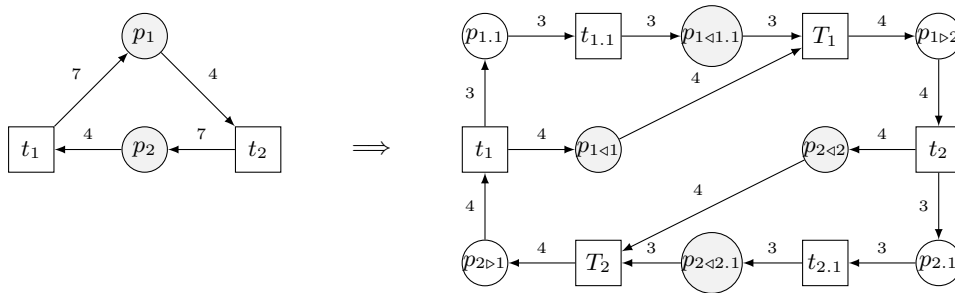


Abbildung 5.11: Nicht-einfache Stellen repräsentierende „Parallelstellen“

dieses Netz sind in der folgenden Tabelle aufgeführt:

	$p_{1 \triangleleft 1.1}$	$p_{2 \triangleleft 2.1}$	$p_{1.1}$	$p_{2.1}$	$p_{1 \triangleleft 1}$	$p_{1 \triangleright 2}$	$p_{2 \triangleleft 2}$	$p_{2 \triangleright 1}$
$\vec{i}^1$	0	0	0	0	1	1	1	1
$\vec{i}^2$	0	4	0	4	3	3	0	3
$\vec{i}^3$	4	0	4	0	0	3	3	3
$\vec{i}^4$	4	4	4	4	0	3	0	3
$KH(p)$	–	–	–	–	2	–	2	–

Die beiden Häufigkeitswerte sind hier gleich. Es wird also die erste, eine nicht-einfache Stelle repräsentierende, Stelle der Art  $p_{x \triangleleft y}$ , hier  $p_{1 \triangleleft 1}$ , in  $K$  übernommen, gleichzeitig aber auch ihre Parallelstelle  $p_{1 \triangleleft 1.1}$ . Ohne diese beiden Stellen ist das Netz kreisfrei.

Die Häufigkeitstabelle wird mit jedem Durchlauf der *while*-Schleife aktualisiert, d.h. die Stelle(n) im Netz gelöscht, S-Invarianten berechnet und die Werte in  $KH$  angepasst.

**Beispiel 5.3 (Ermitteln der Schnittstellen)** Auf Seite 63 ist Netz ES3 mit seinen neun S-Invarianten gegeben. Die Häufigkeitstabelle  $KH$  sieht für dieses Netz folgen-

dermaßen aus:

$$KH := \begin{array}{c|c|c|c|c|c|c|c|c|c} p & p_{1 \triangleleft 7} & p_{3 \triangleleft 3} & p_{3 \triangleleft 1} & p_{5 \triangleleft 1} & p_{5 \triangleleft 4} & p_{6 \triangleleft 1} & p_{6 \triangleleft 4} & p_{6 \triangleleft 9} \\ \hline h & 6 & 3 & 2 & 2 & 1 & 2 & 4 & 1 \end{array}$$

Sie lässt sich schnell berechnen, indem für jede Stelle vom Typ  $p_{x \triangleleft y}$  deren Vorkommen in den neun S-Invarianten gezählt wird. Stellen mit Häufigkeit Null liegen nicht auf einem Kreis und sind darum hier nicht aufgeführt. Die in Netz R3 vorhandenen Schlingen werden in ES3 durch die S-Invarianten  $\vec{i}^1$  bis  $\vec{i}^3$  repräsentiert. Die zu löschenden Stellen sind damit

$$K = \{p_{6 \triangleleft 9}, p_{5 \triangleleft 4}, p_{3 \triangleleft 1}\}.$$

Entfernt man diese in ES3, so hat das aktualisierte Netz immer noch Kreise. Der größte Wert  $max = 5$  in  $KH$  wird nach wie vor für Stelle  $p_{1 \triangleleft 7}$  gefunden. Nun wird  $p_{1 \triangleleft 7}$  in  $K$  aufgenommen. Schlussendlich ist ES3 nach Löschen dieser Stelle kreisfrei. Für das Originalnetz R3 bedeutet dies nun, dass folgende Kanten aufgeschnitten werden:  $t_9 \rightarrow p_6$ ,  $t_4 \rightarrow p_5$ ,  $t_1 \rightarrow p_3$  und  $t_7 \rightarrow p_1$ .  $\square$

Mit diesen Informationen wird nun ein zyklenfreies Netz wie folgt konstruiert: Die durch die Stellen in  $K$  beschriebenen Kanten werden in  $R$  ersetzt, so dass an Stelle jeder aufzuschneidenden Kante  $(t_y, p_x)$  je zwei Transitionen  $(t'_{y,x}, t''_{y,x})$  und eine Stelle  $s_{y,x}$  wie in Algorithmus 5.5 beschrieben treten.

**Algorithmus 5.5:** TRANSFORMATION IN EIN ZYKLENFREIES NETZ

```

Eingabe: Netz  $R = (S, T, F, W)$ ,  $K$ .
Ausgabe: zyklenfreies Netz  $R^{cf}$ 

1 for  $i := 1$  to  $|K|$  do
2    $p_x := K[i]$ ;  $t_y := \bullet p_x$ ;
3   if ( $p_x$  Stelle vom Typ  $p_{x \triangleleft x.1}$ ) then continue;           ▶ „Parallelstelle“ ignorieren.
4   endif
5    $S := S \cup \{s_{y,x}\}$ ;                                       ▶ 2 neue Stellen und 1 Transition hinzufügen.
6    $T := T \cup \{t'_{y,x}, t''_{y,x}\}$ ;
7    $F := F \cup \{(t_y, s_{y,x}), (s_{y,x}, t'_{y,x}), (t''_{y,x}, p_x)\}$ ;
8    $W((t_y, s_{y,x})) := W((t_y, p_x))$ ;                         ▶ Kantengewichte übernehmen.
9    $W((s_{y,x}, t'_{y,x})) := W((t_y, p_x))$ ;
10   $W((t''_{y,x}, p_x)) := W((t_y, p_x))$ ;
11   $F := F \setminus \{(t_y, p_x)\}$ ;                               ▶ Kante  $(t_y, p_x)$  löschen.
12 endfor

```

**Beispiel 5.4 (Erzeugen eines zyklensfreien Netzes)** Die in Beispiel 5.3 ermittelten Stellen  $K$  werden Algorithmus 5.5 übergeben und aus Netz R3 ein neues, kreisfreies Netz  $R3^{cf}$  erzeugt. Das Ergebnis ist in Abbildung 5.12 dargestellt, die neu hinzugekommenen Stellen und Transitionen sind grau unterlegt. An der Dynamik des Netzes wird nichts verändert, wenn jedes Paar  $(t'_{y,x}, t''_{y,x})$  immer genauso häufig schaltet wie die zugehörige Transition  $t_y$ . □

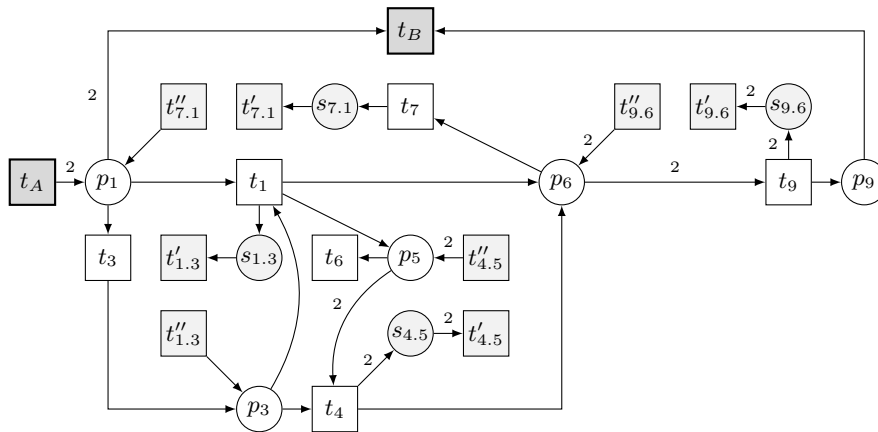


Abbildung 5.12:  $R3^{cf}$  — zyklensfreies Netz R3

### 5.5.3 Exkurs: Eine „universale“ Markierung

Nun stellt sich die Frage, ob aus den S-Invarianten des erweiterten Synchronisationsgraphen noch weitere hilfreiche Informationen gewonnen werden können. S-Invarianten beschreiben zunächst nur Stellenmengen, deren gewichtete Markensumme immer gleich der Markensumme bei der Anfangsmarkierung ist. Es gibt aber noch eine weitere Besonderheit: Aufgrund der Struktur dieses erweiterten Synchronisationsgraphen besitzt dieser selbst eine überdeckende T-Invariante, bei der jede Transition einmal schaltet, aber auch die Netzdarstellungen seiner S-Invarianten sind gleichzeitig T-Invarianten: Eingangs- und Ausgangskantengewichte jeder Stelle sind identisch, so dass also bei einmaligem Schalten jeder Transition immer genau so viele Marken von Stellen herunter genommen wie darauf gelegt werden. Auch der erweiterte Synchronisationsgraph  $ES$  ist darum lebendig, wenn alle seine Kreise ausreichend markiert sind. Ausreichend bedeutet hier: Auf eine beliebige Stelle im Kreis müssen genau so viele Marken gelegt werden, wie ihre Nachfolgetransition benötigt, um aktiviert zu sein.

Die Auswahl, welche Stelle im jeweiligen Kreis zu Beginn markiert wird, ist willkürlich. Jedoch liefert Algorithmus 5.4 auf Seite 64 bereits eine minimal notwendige

Anzahl von Stellen: In der von ihm erzeugten Liste  $K$  sind all die Stellen enthalten, durch die die durch S–Invarianten beschriebenen minimalen Kreise des Netzes verlaufen. Sind genau diese Stellen ausreichend markiert, so ist das Netz lebendig.

**Beispiel 5.5** Für das Netz in Abbildung 5.10 auf Seite 63 wurde die Stellenmenge

$$K = \{p_{6 \leftarrow 9}, p_{5 \leftarrow 4}, p_{3 \leftarrow 1}, p_{1 \leftarrow 7}\}$$

ermittelt. Eine passende Startmarkierung für ES3 besteht damit aus je einer Marke auf  $p_{1 \leftarrow 7}$  und  $p_{3 \leftarrow 1}$  und je zwei Marken auf  $p_{5 \leftarrow 4}$  und  $p_{6 \leftarrow 9}$ . Eine mögliche Schaltfolge zur Realisierung der überdeckenden T–Invariante des Netzes lautet damit

$$t_A T_1 t_3 T_3 t_1 T_5 t_4 T_6 t_7 t_9 t_6 t_B,$$

jede Transition schaltet genau einmal und die Startmarkierung wird reproduziert.  $\square$

### 5.5.3.1 Realisierbarkeit der minimalen T–Invarianten im Originalnetz

An dieser Stelle bietet sich ein Vergleich mit dem Verfahren von Alexander Kostin an: Auf Seite 35 ist der Markenbedarf aufgelistet, der den minimalen T–Invarianten des Originalnetzes N3 zur Realisierbarkeit fehlt. Benötigt wurden eine Marke auf  $p_3$  sowie jeweils zwei Marken auf  $p_5$  und  $p_6$ . Vergleicht man diese Werte mit obiger Liste  $K$ , so erkennt man die Parallelen: Der Markenbedarf entspricht der für die Kreise in ES3 erforderlichen Startmarkierung — mit Ausnahme von Stelle  $p_{1 \leftarrow 7}$ , die in Kostins Verfahren kein Äquivalent besitzt, da dort durch die Anfangsmarkierung bereits genug Marken auf  $p_1$  vorhanden sind.

Es liegt also die Vermutung nahe, dass aus dem Netz  $ES$  auch eine Art „universale“ Markierung für das Netz  $R$  abgelesen werden kann, unter der alle minimalen T–Invarianten und deren Linearkombinationen in  $R$  realisierbar sind:

Um die Markierung für die entsprechenden Stellen in  $R$  zu erhalten, betrachtet man die Eingangstellen von Transitionen  $T_i$  und einfache Stellen in  $ES$ , die in  $R$  nicht eliminiert werden durften. Einfache Stellen, die mit  $t_A$  oder  $t_B$  verbunden sind, werden ignoriert. Für den Vorbereich der  $T_i$  gilt: Das Maximum aller eingehenden Kantengewichte wird als Markierung für die Stelle  $p_i$  in  $R$  übernommen — mit einer Ausnahme: Das Kantengewicht  $W((p_{i \leftarrow i.1}, T_i))$  einer „Parallelstelle“ wird zusätzlich auf die Markierung von  $p_i$  addiert — symbolisch also das ursprüngliche Kantengewicht in  $R$  wiederhergestellt, welches bei der Ersetzung nicht–einfacher Stellen aufgeteilt

wurde. Auf diese Weise werden immer so viele Marken auf die Stellen gelegt, wie deren Vortransition liefern kann (im Fall der einfachen Stellen) bzw. maximal durch einfaches Schalten einer Transition im Vorbereich einer Grenzstelle produziert werden können.

Unter der Markierung  $M_U$  sind also alle minimalen T-Invarianten eines Netzes  $R$  und folglich auch deren Linearkombinationen realisierbar:  $M_U$  liefert zu Beginn bereits jeder T-Invariante genau die Marken, die sie sich sonst „leihen“ müsste, um realisierbar zu sein.  $M_U$  übersteigt unter Umständen aber die minimal erforderliche Markierung — eine T-Invariante kann auch mit weniger Marken realisierbar sein.

**Algorithmus 5.6:** BERECHNUNG EINER UNIVERSALEN MARKIERUNG  $M_U$

**Eingabe:** Netz  $ES = (S_S, T_S, F_S, W_S)$ ,  $R = (S_R, T_R, F_R, W_R)$

**Ausgabe:** universale Markierung  $M_U$

```

1  $M_U(p_i) := \vec{0}$ ; ▶ zunächst eine leere Markierung erzeugen.
2 for  $i := 1$  to  $|S_S|$  do
3   if ( $p_i$  ist aus  $R$  übernommene einfache Stelle und  $\bullet p_i \neq t_A$  und  $p_i \bullet \neq t_B$ ) then
4      $M_U(p_i) := W((t, p_i))$ ; ▶ Hierbei sei  $t = \bullet p_i$ 
5   else if ( $p_i$  ist vom Typ  $p_{i \triangleleft y}$  mit Parallelstelle  $p_{i \triangleleft i.1}$  und  $\bullet p_{i \triangleleft y} \neq t_A$ ) then
6      $M_U(p_i) := \max(W((p_{i \triangleleft y}, T_i)) + W((p_{i \triangleleft i.1}, T_i)), M_U(p_i))$ ;
7   else if ( $p_i$  ist vom Typ  $p_{i \triangleleft y}$  ohne Parallelstelle und  $\bullet p_{i \triangleleft y} \neq t_A$ ) then
8      $M_U(p_i) := \max(W((p_{i \triangleleft y}, T_i)), M_U(p_i))$ ;
9   endif
10 endfor

```

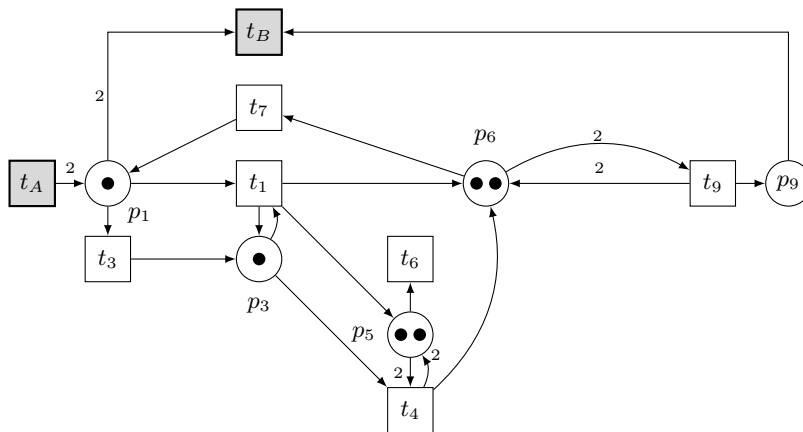


Abbildung 5.13: S/T-Netz  $R_3$  mit universaler Markierung  $M_{3U}$

**Beispiel 5.6 (Universale Markierung)** In Abbildung 5.13 ist Netz R3 mit universaler Markierung dargestellt. Alle Transitionen des Netzes — bis auf  $t_B$  — sind unter dieser Markierung aktiviert. In der Tat sind auch alle minimalen T-Invarianten des Netzes  $((t_9, t_A, t_B), (t_1, t_6, t_7)$  und  $(t_3, t_4, t_7))$  und somit auch beliebige Linearkombinationen realisierbar;  $M_U$  wird stets reproduziert.

Alle Netzdarstellungen dieser T-Invarianten enthalten Kreise. Es werden genau die oder sogar mehr Marken geliehen, die sonst beim Durchlaufen der T-Invariante auf der jeweiligen Stelle produziert und die nächste Transition aktivieren würden. Durch die zwei Marken auf  $p_6$  werden beispielsweise Transition  $t_9$  ausreichend Marken geliehen, die sie nach dem Schalten sofort zurückgibt.  $\square$





## 6 Lösungsansatz I: Berechnung eines Synchronisationsgraphen

Im letzten Abschnitt wurde eine universale Markierung gefunden, unter der alle T-Invarianten in einem Netz  $R$  realisierbar sind. Wie sieht es aber nun mit der Reproduzierbarkeit der leeren Markierung aus? In der universalen Markierung wurden für jede Transition möglicherweise notwendige Marken zur Verfügung gestellt, die bei der leeren Startmarkierung nicht verfügbar sind. Bei der Reproduktion der leeren Markierung gilt für jede Stelle: Es müssen genauso viele Marken produziert wie konsumiert werden. Die Summen aller eingehenden und ausgehenden Kantengewichte einer Stelle in Abhängigkeit von den Schalthäufigkeiten der Transitionen müssen daher gleich sein. Wie findet man aber nun eine die leere Markierung reproduzierende T-Invariante?

Ein erster Versuch, schwache und starke Erreichbarkeit mittels T-Invarianten nachzuweisen, wurde in den Abschnitten 5.3 und 5.4 diskutiert. Nun wird ein diese Methoden erweiterndes Verfahren vorgestellt, das wir auf eine Unterklasse der (um einfache Stellen reduzierten) S/T-Netze  $N_{ABC}$  mit folgenden Eigenschaften anwenden wollen:

- (1) Unter den minimalen T-Invarianten von  $N_{ABC}$  existiert eine T-Invariante  $\vec{r}_{AB}$  mit  $\vec{r}_{AB}(t_A) = \vec{r}_{AB}(t_B) = 1$ , deren Netzdarstellung noch Deadlocks und/oder Traps enthält. Diese T-Invariante muss daher zum Nachweis schwacher und starker Erreichbarkeit noch mit anderen kombiniert werden.
- (2) Ist  $\vec{r}_{AB}$  nicht die einzige minimale T-Invariante ihrer Zusammenhangskomponente im Relationsgraphen, in der  $t_A$  und  $t_B$  genau einmal feuern, so muss durch die Berechnung eines N-Erzeugendensystems mit der zusätzlichen Bedingung  $\vec{r}(t_A) = \vec{r}(t_B) \leq 1$  ein weiterer Vektor gefunden werden, der eine Kombination dieser minimalen T-Invarianten ist. In diesem Fall wird mit dieser neuen Linearkombination fortgefahren.
- (3) Mit einem solchen Vektor aus (2) noch zu kombinierende T-Invarianten sind demnach alle diejenigen, in denen  $t_A$  und  $t_B$  nicht schalten.

Das Verfahren gliedert sich in die folgenden Schritte: Zunächst werden die S-Invarianten im Netz  $ES$  für einen weiteren Informationsgewinn herangezogen. Dieses neue

Wissen, Hinweise auf die erforderliche Schalthäufigkeit einzelner Transitionen, wird zur präziseren Auswahl einer geeigneten Linearkombination minimaler T-Invarianten verwendet. Anstatt nun die Linearkombination im Netz zu simulieren, kann deren Netzdarstellung als sogenannter „Pseudo-Synchronisationsgraph“ modelliert und deren Realisierbarkeit durch die Berechnung eines in diesem Netz existierenden kreisfreien Synchronisationsgraphen linear-algebraisch nachgewiesen werden.

Zur Einführung betrachten wir erneut die minimalen T-Invarianten von Netz R3:

	$t_1$	$t_3$	$t_4$	$t_6$	$t_7$	$t_9$	$t_A$	$t_B$
$\vec{F}^1$	0	1	1	0	1	0	0	0
$\vec{F}^2$	1	0	0	1	1	0	0	0
$\vec{F}^3$	0	0	0	0	0	1	1	1

Für dieses Netz ist bereits bekannt, dass starke Erreichbarkeit vorliegt (siehe Abschnitt 4.1) und welche Linearkombination von T-Invarianten den Nachweis liefert. Wir untersuchen nun die S-Invarianten des erweiterten Synchronisationsgraphen ES3 (abgebildet in Tabelle 5.1 auf Seite 63) und in ihnen insbesondere die Übergänge an *Grenzstellen* repräsentierenden Transitionen  $T_i$ , die in R3 Schnittstellen zwischen unterschiedlichen minimalen T-Invarianten entsprechen.

Tabelle 6.1 zeigt zwei S-Invarianten von ES3, in denen es einen interessanten Übergang von  $p_{5 \triangleleft 1}$  nach  $p_{5 \triangleright 4}$  gibt, was dem Übergang von Transition  $t_1$  über  $p_5$  nach  $t_4$  in R3 entspricht. Marken, die von  $t_1$  auf  $p_5$  gelegt werden, haben in der S-Invariante offenbar eine größere Gewichtung als die Marken, die  $t_4$  von  $p_5$  nimmt. Übersetzt man dies ins Originalnetz R3, liegt die Vermutung nahe, dass  $t_1$  auch eine „größere“ Gewichtung als  $t_4$  hat:  $t_1$  muss zweimal schalten, damit  $t_4$  einmal feuern kann.

In der Tat ist dies auch der Fall: In R3 (vgl. Abbildung 5.8 auf S. 59) ist  $t_4$  neben  $t_1$  die einzige Eingangstransition von  $p_5$ . Transition  $t_4$  — die mit  $p_5$  eine Schlinge bildet — hat das bekannte Problem: Sie müsste sich zwei Marken auf  $p_5$  leihen, um schalten zu können, und gibt diese sofort wieder zurück. Nur durch zweimaliges Feuern von  $t_1$  können diese fehlenden Marken bereitgestellt werden. Ein neuer Ansatz ist nun, die

$p$	9	1 < A	1 > 3	1 > 1	1 < 7	1 > B	3 < 3	3 > 1	3 < 1	3 > 4	5 < 1	5 > 4	5 < 4	5 > 6	6 > 7	6 < 1	6 < 4	6 < 9	6 > 9
$\vec{z}^6$	0	0	0	2	2	0	0	0	0	0	2	1	0	0	2	0	2	0	0
$\vec{z}^9$	0	0	2	0	2	0	2	2	0	0	2	1	0	0	2	0	2	0	0

**Tabelle 6.1:** „Kritische“ Übergänge repräsentiert durch S-Invarianten in  $ES$ , in diesem Fall für ES3

T-Invariante, die  $t_1$  enthält, zweimal in die Linearkombination der drei minimalen T-Invarianten einfließen zu lassen. Als Linearkombination  $\vec{F} := \vec{F}^1 + 2 \cdot \vec{F}^2 + \vec{F}^3$  erhält man somit:

	$t_1$	$t_3$	$t_4$	$t_6$	$t_7$	$t_9$	$t_A$	$t_B$
$\vec{F}^1$	0	1	1	0	1	0	0	0
$\vec{F}^2$	1	0	0	1	1	0	0	0
$\vec{F}^3$	0	0	0	0	0	1	1	1
$\vec{F}$	2	1	1	2	3	1	1	1

Mit den Schalthäufigkeiten jeder Transition ist die Summe der eingehenden Kantengewichte jeder Stelle gleich der Summe ihrer ausgehenden Kantengewichte. Nun kann, wie bereits erwähnt, für diese Linearkombination zum einen durch Simulationsschritte auf starke Erreichbarkeit getestet werden. Eine andere Variante ist die Transformation der Netzdarstellung von  $\vec{F}$  in einen Synchronisationsgraphen, so dass die Sätze von Seite 60 über die Lebendigkeit von Synchronisationsgraphen wieder als einschränkendes Kriterium herangezogen werden können.

Zunächst muss allerdings noch geklärt werden: In welchen Fällen lassen sich Faktoren wie oben dargestellt aus den S-Invarianten für Linearkombinationen von T-Invarianten ablesen? Schließlich gibt es auch Netze mit mehr als drei minimalen T-Invarianten, in denen eine Linearkombination nicht eindeutig konstruierbar ist. Die Vorgehensweise zum Umgang mit dieser Problemstellung wird nun anhand von Beispielen erläutert.

## 6.1 Auswahl einer geeigneten Linearkombination von T-Invarianten

Für die Auswahl einer geeigneten Linearkombination werden zunächst nur die T-Invarianten in nähere Betrachtung gezogen, die in der gleichen Zusammenhangskomponente eines Relationsgraphen<sup>1</sup> liegen. Für eine Linearkombination aus dieser Zusammenhangskomponente muss schwache Erreichbarkeit nachweisbar sein. Im Folgenden wird davon ausgegangen, dass unter diesen T-Invarianten der Zusammenhangskomponente keine einzelne T-Invariante geeignet für den Test auf starke Erreichbarkeit ist. Eine Linearkombination zu finden — notfalls unter Einbeziehung des  $\mathbb{N}$ -Erzeugendensystems — ist also unumgänglich. Im ersten Schritt werden nun die S-Invarianten des Netzes  $ES$  und insbesondere Markenzahlen auf den *Grenzstellen* be-

<sup>1</sup>Zur Erinnerung: Die Knoten eines Relationsgraphen sind die minimalen T-Invarianten des Netzes. Zwei Knoten werden verbunden, wenn ihre Netzdarstellungen mindestens eine gemeinsame Stelle haben.

trachtet. Interessant sind alle Übergänge an Transitionen  $T_i$  der Form  $p_{i\leftarrow x} \rightarrow T_i \rightarrow p_{i\triangleright y}$ . Jeder Übergang, für den

$$W((p_{i\leftarrow x}, T_i)) > W((T_i, p_{i\triangleright y}))$$

gilt, ist für die Bildung von Linearkombinationen von Bedeutung. Im Originalnetz  $N_{AB}$  bedeutet ein solcher Sachverhalt, dass Transition  $t_x$   $W((p_{i\leftarrow x}, T_i))$ -mal schalten muss, damit Transition  $t_y$   $W((T_i, p_{i\triangleright y}))$ -mal feuern kann. Wenn Transitionen  $t_x$  und  $t_y$  zu unterschiedlichen T-Invarianten gehören, die sich gegenseitig bei ihrer Realisierbarkeit helfen, muss diese Gewichtung ebenso berücksichtigt werden wie auch bei einer T-Invariante, in der beide Transitionen feuern. Allerdings müssen die Übergänge an Transitionen  $T_i$ , die bei der Ersetzung nicht-einfacher Stellen entstanden sind, nicht betrachtet werden: Jede minimale T-Invariante, die durch Vor- und Nachbereich einer nicht-einfachen Stelle (jeweils nur eine Transition) verläuft, muss die passende Gewichtung („so viele Marken produzieren wie konsumieren“) in ihrem Schalthäufigkeitsvektor ausdrücken; ob sie letztendlich realisierbar ist, hängt von der gegebenen Markierung ab. Transitionen  $T_i$ , die eine Stelle der Art  $p_{i\leftarrow i.1}$  oder  $p_{i\triangleright i.1}$  in ihrem Vor- oder Nachbereich haben, werden also nicht betrachtet.

**Beispiel 6.1 (Linearkombination nicht ablesbar)** In Abbildung 6.1 ist der erweiterte Synchronisationsgraph für das Netz N4 von Seite 38 dargestellt. Start- und Zielmarkierung entsprechen den Angaben von Seite 38 und sind durch  $t_A$  und  $t_B$  repräsentiert. Die zugehörigen S-Invarianten sind in Tabelle 6.2 aufgeführt, kritische Übergänge

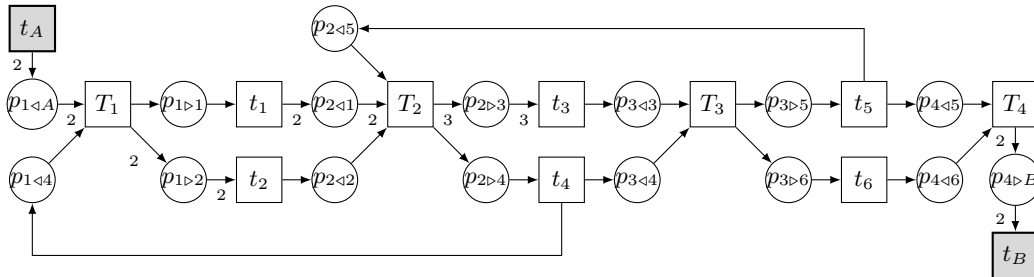


Abbildung 6.1: Netz ES4

sind hervorgehoben und betreffen die S-Invarianten  $\vec{i}^2$  und  $\vec{i}^3$ . Die Netzdarstellung von  $\vec{i}^2$  enthält die Transitionen  $t_2$  und  $t_4$ , die Netzdarstellung von  $\vec{i}^3$  die Transitionen  $t_3$  und  $t_5$ . Betrachtet man jedoch die T-Invarianten des um  $p_C$  erweiterten Netzes  $N4_{ABC}$  (abgebildet auf Seite 53), die für eine Erreichbarkeitsanalyse überhaupt in Frage kommen, und vergleicht sie mit den Transitionspaaren aus den S-Invarianten, erkennt man die

	1 < A	1 < 4	1 ▷ 1	1 ▷ 2	2 < 5	2 < 1	2 < 2	2 ▷ 3	2 ▷ 4	3 < 3	3 < 4	3 ▷ 5	3 ▷ 6	4 < 5	4 < 6	4 ▷ B
$\vec{r}^1$	0	2	2	0	0	1	0	0	2	0	0	0	0	0	0	0
$\vec{r}^2$	0	2	0	1	0	0	2	0	2	0	0	0	0	0	0	0
$\vec{r}^3$	0	0	0	0	3	0	0	1	0	3	0	3	0	0	0	0
$\vec{r}^4$	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0

Tabelle 6.2: S-Invarianten von ES4

Unbrauchbarkeit der Ergebnisse der S-Invariantenberechnung: Beide Transitionspaare gehören auch in  $N_{4ABC}$  zu je einer T-Invariante  $\vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$ . Keine dieser T-Invarianten kommt allein als Kandidat für eine Erreichbarkeitsanalyse in Frage, da die Netzdarstellungen noch Deadlocks und Traps enthalten, und muss daher noch mit der jeweils anderen kombiniert werden. Für dieses Netz musste bereits zum Nachweis der schwachen Erreichbarkeit ein N-Erzeugendensystem ermittelt werden. Dabei wird eine Linearkombination gefunden, in der jede Transition des Netzes einmal schaltet und mit der sofort auf starke Erreichbarkeit — ohne Beachtung der S-Invarianten — getestet werden kann.  $\square$

Bei Netzen, in denen tatsächlich eine T-Invariante  $\vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  noch mit mindestens einer anderen T-Invariante kombiniert werden muss, sind notwendige Voraussetzungen für die Einbeziehung der Information aus den S-Invarianten gegeben: Zum einen muss nun eine Linearkombination gefunden werden, deren Netzdarstellung frei von Deadlocks und Traps ist. Zum anderen muss an jeder Stelle, abhängig von der Schalthäufigkeit der mit ihr verbundenen Transitionen, die Summe ihrer Eingangskanten gleich der Summe ihrer Ausgangskantengewichte sein: Es müssen immer so viele Marken produziert wie konsumiert werden. In Folge lässt sich aus den S-Invarianten möglicherweise die Gewichtung ablesen, wie oft eine T-Invariante in eine Linearkombination eingeht. Aber: In der Regel ist starke Erreichbarkeit nicht nur durch *eine* spezifische zusammengesetzte T-Invariante nachweisbar, sondern verschiedene Linearkombinationen liefern das gleiche Ergebnis. Auch hier gibt es also eine Vielzahl von Kombinationsmöglichkeiten, die schlimmstenfalls *alle* getestet werden müssen.

**Beispiel 6.2 (Ermitteln einer Linearkombination)** Das in Abbildung 6.2 dargestellte S/T-Netz  $N_{7AB}$  stammt aus [Kos08], einfache Stellen wurden bereits eliminiert und  $t_A$  und  $t_B$  entsprechend der in [Kos08] gegebenen Anfangs- und Zielmarkierung hinzugefügt. Bis auf  $p_5$  handelt es sich bei allen Stellen um Grenzstellen. Das um  $p_C$  erweiterte Netz  $N_{7ABC}$  besitzt die in Tabelle 6.3 aufgeführten minimalen T-Invarianten.

Alle gehören zur gleichen Zusammenhangskomponente im Relationsgraphen.

Unter diesen T-Invarianten befindet sich mit  $\vec{r}^1$  nur genau eine, in der  $t_A$  und  $t_B$  jeweils einmal schalten. Allein ist diese T-Invariante nicht realisierbar, ihre Netzdarstellung enthält sowohl Deadlocks als auch Traps. Welche Generatoren mit  $\vec{r}^1$  kombiniert eine die leere Markierung reproduzierende T-Invariante ergeben, ist nicht unmittelbar erkennbar. Die Voraussetzungen für eine weitere Analyse mittels der S-Invarianten sind jedoch erfüllt. Im nächsten Schritt wird somit der erweiterte, in Abbildung 6.3 dargestellte, Synchronisationsgraph ES7 erzeugt. Die S-Invarianten von ES7 sind in Tabelle 6.4 abgebildet. Es gibt einen einzigen kritischen Übergang von Stelle  $p_{2 \llcorner 2}$  zu Stelle  $p_{2 \triangleright 6}$ . In  $N7_{AB}$  muss  $t_2$  folglich dreimal schalten, damit  $t_6$  zweimal schalten kann. Oder verständlicher ausgedrückt: Damit  $t_6$  aktiviert ist, muss  $t_2$  mindestens

$$\left\lceil \frac{3}{2} \right\rceil\text{-mal} = 2\text{-mal}$$

gefeuert haben. Die einzige T-Invariante aus Tabelle 6.3, in der  $t_2$  und  $t_6$  gemeinsam vorkommen, erfüllt diese Bedingung bereits — in ihrer Netzdarstellung muss die Gewichtung schließlich berücksichtigt sein. Sie kann also zu  $\vec{r}^1$  addiert werden. Andernfalls müsste man mindestens zwei andere T-Invarianten miteinander kombinieren. Die Linearkombination  $\vec{r} = \vec{r}^1 + \vec{r}^6 = (2 \ 2 \ 1 \ 0 \ 0 \ 1 \ 2 \ 0 \ 1 \ 1 \ 1)^T$  besitzt in ihrer Netzdarstellung weder Deadlocks noch Traps und die Summen der Eingangs- und Ausgangskantengewichte sind für jede Stelle ausgeglichen. Durch Simulation kann starke Erreichbarkeit für das Netz  $N7_{AB}$  nachgewiesen werden. Eine mögliche Schaltfolge, die die leere Markierung reproduziert, ist:

$$t_A, t_2, t_3, t_1, t_2, t_6, t_9, t_7, t_7, t_1, t_B$$

□

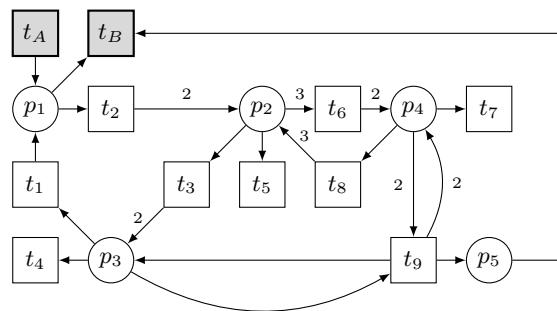


Abbildung 6.2: Netz  $N7_{AB}$

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_A$	$t_B$
$\vec{r}^1$	0	0	0	0	0	0	0	0	1	1	1
$\vec{r}^2$	2	2	1	0	3	0	0	0	0	0	0
$\vec{r}^3$	0	0	0	0	3	1	0	2	0	0	0
$\vec{r}^4$	1	1	2	3	0	0	0	0	0	0	0
$\vec{r}^5$	0	0	3	6	0	1	0	2	0	0	0
$\vec{r}^6$	2	2	1	0	0	1	2	0	0	0	0
$\vec{r}^7$	0	0	0	0	0	1	1	1	0	0	0

Tabelle 6.3: Minimale T-Invarianten von N7<sub>ABC</sub>

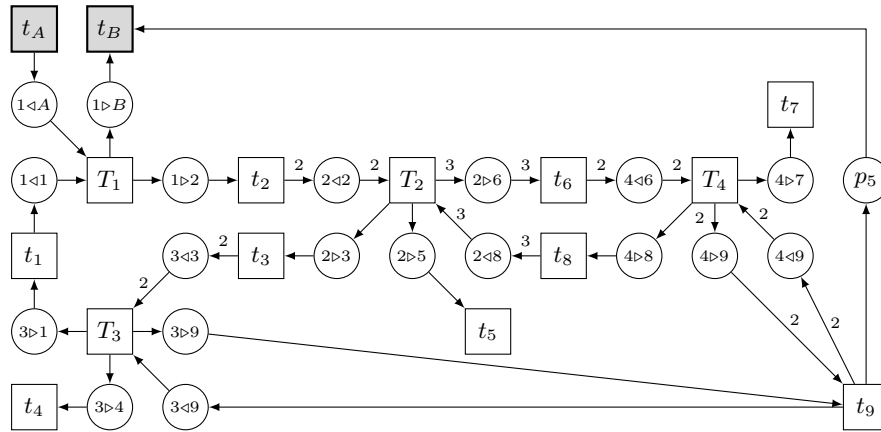


Abbildung 6.3: Netz ES7

**Fazit:**

Durch die Betrachtung der S-Invarianten im erweiterten Synchronisationsgraphen und den zusätzlichen Informationsgewinn kann in einigen Fällen eine Linearkombination, mit der auf starke Erreichbarkeit getestet werden soll, schneller und präziser gefunden werden als es bisher der Fall war. Diese „zielorientierte Analyse“ zeigt dennoch erneut, dass ein rein linear-algebraischer Nachweis starker Erreichbarkeit durch eine passende Linearkombination von T-Invarianten schwer möglich ist. Hat man jedoch eine Linearkombination gefunden, deren Netzdarstellung keine Deadlocks und Traps besitzt

	$1 \triangleleft 1$	$1 \triangleleft A$	$1 \triangleright B$	$1 \triangleright 2$	$2 \triangleleft 2$	$2 \triangleleft 8$	$2 \triangleright 3$	$2 \triangleright 5$	$2 \triangleright 6$	$3 \triangleleft 3$	$3 \triangleleft 9$	$3 \triangleright 1$	$3 \triangleright 4$	$3 \triangleright 9$	$4 \triangleleft 6$	$4 \triangleleft 9$	$4 \triangleright 8$	$4 \triangleright 9$	$4 \triangleright 7$	$p_5$
$\vec{i}^1$	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
$\vec{i}^2$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
$\vec{i}^3$	0	0	0	0	0	2	0	0	2	0	0	0	0	0	3	0	6	0	0	0
$\vec{i}^4$	2	0	0	2	1	0	2	0	0	1	0	2	0	0	0	0	0	0	0	0
$\vec{i}^5$	0	0	0	0	0	2	6	0	0	3	0	0	0	6	0	3	6	0	0	0
$\vec{i}^6$	6	0	0	6	3	0	0	0	2	0	6	6	0	0	3	0	0	3	0	0

Tabelle 6.4: Minimale S-Invarianten von ES7

und in der ein ausgeglichenes Eingangs-/Ausgangskantenverhältnis für jede Stelle gegeben ist, so kann mittels der in den nächsten Abschnitten vorgestellten Verfahren rein rechnerisch auf starke Erreichbarkeit getestet werden.

## 6.2 Vom erweiterten Synchronisationsgraphen zum Synchronisationsgraphen

Mit einer im letzten Abschnitt gefundenen Linearkombination  $\vec{r}$  könnte nun durch Simulationsschritte auf starke Erreichbarkeit getestet werden. Es gibt aber auch die Möglichkeit, den Nachweis rein rechnerisch zu erbringen, indem das Netz  $ES$  unter Berücksichtigung der Schalthäufigkeiten der Transitionen in der T-Invariante  $\vec{r}$  umgebaut wird. Auf das transformierte Netz lassen sich später die Eigenschaften von Synchronisationsgraphen anwenden. Zunächst wird  $ES^*$  aus  $ES$  folgendermaßen erzeugt:

- Jede Transition  $t_i$  in  $\vec{r}$  mit  $\vec{r}(t_i) =: j > 1$  wird dem Netz  $ES^*$  genau  $j - 1$ -mal als Kopie mit zugehörigem Vor- und Nachbereich und Kanten hinzugefügt. Durch die Ersetzung nicht-einfacher Stellen muss beim Kopieren noch ein Sachverhalt beachtet werden:
  - Falls  $t_i$  die Vor- oder Nachtransition einer nicht-einfachen Stelle  $p_x$  ist, ist sie mit einem Pfad  $p_{x.1} \rightarrow t_{x.1} \rightarrow p_{x \ll x.1}$  oder  $p_{x \gg x.1} \rightarrow t_{x.1} \rightarrow p_{x.1}$  verbunden. Bei der Umwandlung in den Synchronisationsgraphen wollen wir nur noch die Stellen  $p_{x.1}$  dieser Konstrukte erhalten.
- Jede Transition  $t_i$  in  $\vec{r}$  mit  $\vec{r}(t_i) = 0$  wird in Netz  $ES^*$  mit ihrem gesamten Vor- und Nachbereich gelöscht. Durch die Ersetzung nicht-einfacher Stellen muss wiederum beachtet werden:
  - Falls  $t_i$  die Vor- oder Nachtransition einer nicht-einfachen Stelle  $p_x$  ist, so muss auch das gesamte Konstrukt, das  $p_x$  repräsentiert, gelöscht werden.
- Außerdem werden alle Kantengewichte eliminiert: Jede Stelle  $p_x$  mit ein- und ausgehenden Kantengewichten  $l > 1$  wird  $(l - 1)$ -mal als Kopie eingefügt. Ein- und ausgehende Kantengewichte aller Kopien werden auf 1 gesetzt.

Algorithmus 6.1 beschreibt dieses Verfahren mit erforderlichen Umbenennungen von Stellen und Transitionen, um Namenskonflikte zu vermeiden. Beim Löschen von Stellen oder Transitionen wird implizit davon ausgegangen, dass ein- und ausgehende Kanten ebenfalls entfernt werden.



Im Folgenden seien Vor- und Nachbereich von Transitionen jeweils als Listen  $p$  gegeben; hierbei entspreche  $p[k]$  dem  $k$ -ten Element in der Liste. Abbildung 6.4 zeigt ein Beispiel für die Transformationen, die beim Ersetzen einer nicht-einfachen Stelle erfolgen.

**Algorithmus 6.1:** TRANSFORMATION DES ERWEITERTEN SYNCHRONISATIONSGRAPHEN IN EINEN SYNCHRONISATIONSGRAPHEN

**Eingabe:** erweiterter Synchronisationsgraph  $ES = (S, T, F, W)$ , Linearkombination  $\vec{r}$

**Ausgabe:** Synchronisationsgraph  $ES^* = (S^*, T^*, F^*, W^*)$

```

1  $ES^* := ES;$ 
2 for  $i := 1$  to  $|\vec{r}|$  do                                     ▶ Nur Transitionen in  $\vec{r}$  betrachten.
3   if  $\vec{r}(t_i) = 0$  then
4     if (Stelle vom Typ  $p_{x.1}$  ist Teil des Vorbereichs oder Nachbereichs von  $t_i$ ) then
5        $S^* := S^* \setminus \{\bullet T_x, T_x^\bullet\};$ 
6        $T^* := T^* \setminus \{T_x, t_{x.1}\};$ 
7     endif
8      $S^* := S^* \setminus \{\bullet t_i, t_i^\bullet\};$ 
9      $T^* := T^* \setminus \{t_i\};$ 
10    else if  $\vec{r}(t_i) \geq 1$  then
11      for  $j := 1$  to  $\vec{r}(t_i)$  do
12         $T^* := T^* \cup \{t_i^j\};$ 
13         $p := \bullet t_i;$                                        ▶ Liste der Stellen im Vorbereich von  $t_i$ .
14        for  $k := 1$  to  $|p|$  do                               ▶ Vorbereich von  $t_i^j$  umbauen
15          if (Stelle  $p[k]$  ist vom Typ  $p_{x \triangleright i}$ ) then       ▶  $T_x = \bullet p_{x \triangleright i}$ 
16            for  $l := 1$  to  $W((p_{x \triangleright i}, t_i))$  do
17               $S^* := S^* \cup \{p_{x \triangleright i}^{j,l}\};$ 
18               $F^* := F^* \cup \{(T_x, p_{x \triangleright i}^{j,l}), (p_{x \triangleright i}^{j,l}, t_i^j)\};$ 
19               $W^*((T_x, p_{x \triangleright i}^{j,l})) := 1;$ 
20               $W^*((p_{x \triangleright i}^{j,l}, t_i^j)) := 1;$ 
21            endfor
22          else if (Stelle  $p[k]$  ist vom Typ  $p_{x.1}$ ) then       ▶  $t_{x.1} = \bullet p_{x.1}$ 
23            for  $l := 1$  to  $W((p_{x.1}, t_i))$  do
24               $S^* := S^* \cup \{p_{x.1}^{j,l}\};$ 
25               $F^* := F^* \cup \{(T_x, p_{x.1}^{j,l}), (p_{x.1}^{j,l}, t_i^j)\};$ 
26               $W^*((T_x, p_{x.1}^{j,l})) = W^*((p_{x.1}^{j,l}, t_i^j)) := 1;$ 
27            endfor
28           $S^* := S^* \setminus \{p_{x \triangleright x.1}\};$ 
29           $T^* := T^* \setminus \{t_{x.1}\};$ 
30          else if (Stelle  $p[k]$  ist vom Typ  $p_x$ ) then       ▶ restliche Stellen,  $t_y = \bullet p_x$ 
31            for  $l := 1$  to  $W((p_x, t_i))$  do
32               $S^* := S^* \cup \{p_x^{j,l}\};$ 
    
```

```

33       $F^* := F^* \cup \{(t_y, p_x^{j,l}), (p_x^{j,l}, t_i^j)\};$ 
34       $W^*((t_y, p_x^{j,l})) := 1;$ 
35       $W^*((p_x^{j,l}, t_i^j)) := 1;$ 
36      endfor
37      endif
38      endfor
39       $p := t_i^\bullet;$  ▶ Liste der Stellen im Nachbereich von  $t_i$ .
40      for  $k := 1$  to  $|p|$  do ▶ Nachbereich von  $t_i^j$  umbauen
41          if (Stelle  $p[k]$  ist vom Typ  $p_{x \triangleleft i}$ ) then ▶  $T_x = p_{x \triangleleft i}^\bullet$ 
42              for  $l := 1$  to  $W((t_i, p_{x \triangleleft i}))$  do
43                   $S^* := S^* \cup \{p_{x \triangleleft i}^{j,l}\};$ 
44                   $F^* := F^* \cup \{(t_i^j, p_{x \triangleleft i}^{j,l}), (p_{x \triangleleft i}^{j,l}, T_x)\};$ 
45                   $W^*((t_i^j, p_{x \triangleleft i}^{j,l})) := 1;$ 
46                   $W^*((p_{x \triangleleft i}^{j,l}, T_x)) := 1;$ 
47              endfor
48          else if (Stelle  $p[k]$  ist vom Typ  $p_{x.1}$ ) then ▶  $t_{x.1} = p_{x.1}^\bullet$ 
49              for  $l := 1$  to  $W((t_i, p_{x.1}))$  do
50                   $S^* := S^* \cup \{p_{x.1}^{j,l}\};$ 
51                   $F^* := F^* \cup \{(t_i^j, p_{x.1}^{j,l}), (p_{x.1}^{j,l}, T_x)\};$ 
52                   $W^*((t_i^j, p_{x.1}^{j,l})) = W^*((p_{x.1}^{j,l}, T_x)) := 1;$ 
53              endfor
54               $S^* := S^* \setminus \{p_{x \triangleleft x.1}\};$ 
55               $T^* := T^* \setminus \{t_{x.1}\};$ 
56          else if (Stelle  $p[k]$  ist vom Typ  $p_x$ ) then ▶ restliche Stellen,  $t_y = p_x^\bullet$ 
57              for  $l := 1$  to  $W((t_i, p_x))$  do
58                   $S^* := S^* \cup \{p_x^{j,l}\};$ 
59                   $F^* := F^* \cup \{(t_i^j, p_x^{j,l}), (p_x^{j,l}, t_y)\};$ 
60                   $W^*((t_i^j, p_x^{j,l})) := 1;$ 
61                   $W^*((p_x^{j,l}, t_y)) := 1;$ 
62              endfor
63          endif
64      endfor
65      endfor
66       $S^* := S^* \setminus \{^\bullet t_i, t_i^\bullet\};$ 
67       $T^* := T^* \setminus \{t_i\};$ 
68      endif
69      endfor

```

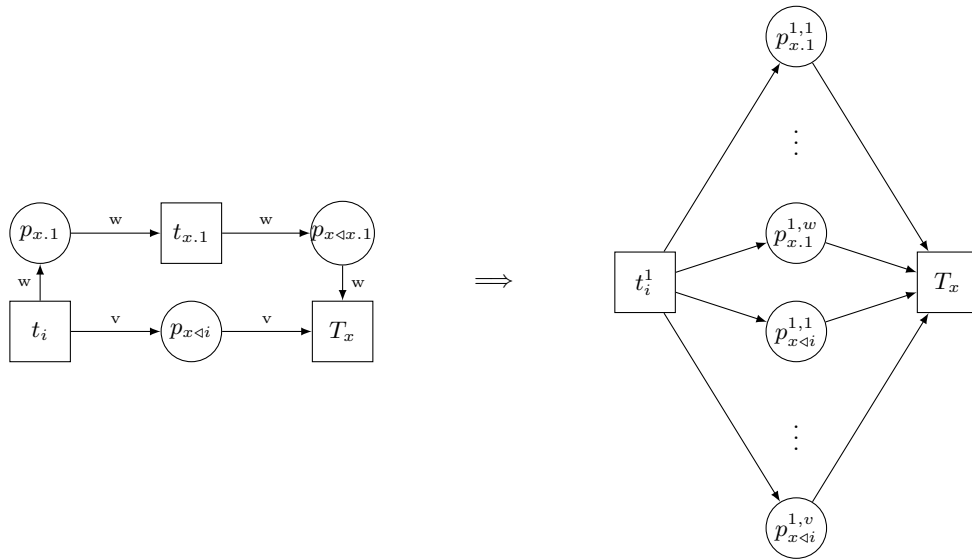


Abbildung 6.4: Beispiel: Behandlung nicht-einfacher Stellen bei der Umwandlung in einen Synchronisationsgraphen

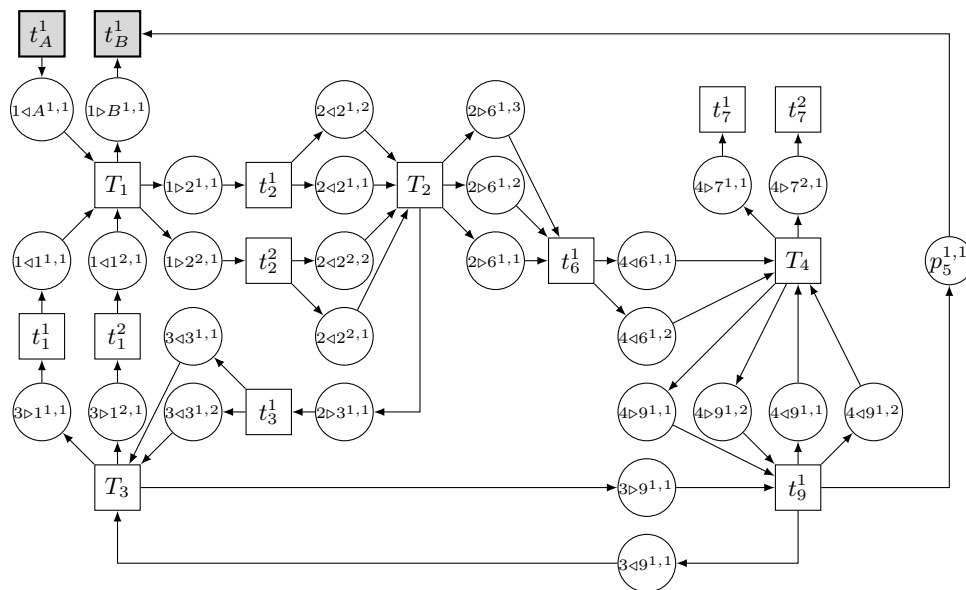


Abbildung 6.5: Synchronisationsgraph ES7\* unter Einbeziehung der Schalthäufigkeiten von T-Invariante  $\vec{r}$

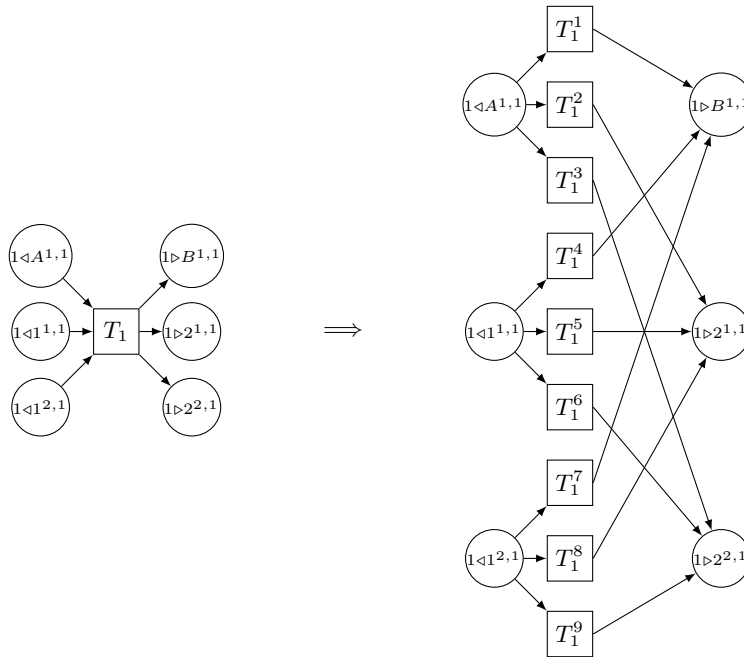
**Beispiel 6.3** Für das Netz aus Abbildung 6.3 von Seite 79 erhält man nach Anwendung des Algorithmus mit T-Invariante  $\vec{r} = (2 \ 2 \ 1 \ 0 \ 0 \ 1 \ 2 \ 0 \ 1 \ 1 \ 1)^T$  den in Abbildung 6.5 dargestellten Synchronisationsgraphen. Die Transitionen  $t_4$ ,  $t_5$  und  $t_8$ , die in  $\vec{r}$  nicht auftreten, wurden in das neue Netz nicht übernommen. Verdoppelt werden mussten die Transitionen  $t_1$  (in  $t_1^1$  und  $t_1^2$ ) sowie  $t_2$  und  $t_7$ . Bei der Ersetzung der Kantengewichte durch neue Stellen und Kanten mit Gewicht 1 wurde beispielsweise der Pfad  $T_2 \xrightarrow{3} p_{2 \triangleright 6} \xrightarrow{3} t_6$  durch insgesamt 3 neue Pfade ersetzt:

$$T_2 \rightarrow p_{2 \triangleright 6^{1,1}} \rightarrow t_6^1$$

$$T_2 \rightarrow p_{2 \triangleright 6^{1,2}} \rightarrow t_6^1$$

$$T_2 \rightarrow p_{2 \triangleright 6^{1,3}} \rightarrow t_6^1$$

Schlussendlich kann noch die Anzahl der Eingangs- und Ausgangskanten jeder Transition  $T_i$  kontrolliert werden. Ist sie gleich, so wurde der Synchronisationsgraph auf Basis der T-Invariante  $\vec{r}$  korrekt modelliert.  $\square$



**Abbildung 6.6:** Ersetzen der Transitionen  $T_i$  durch Gruppen von Transitionen, dargestellt für  $T_1$  aus Netz ES7\*.

### Konstruktion des „Pseudo-Synchronisationsgraphen“

Damit mit den ermittelten Schalthäufigkeiten nun linear-algebraisch auf starke Erreichbarkeit getestet werden kann, muss dieser Synchronisationsgraph wieder so in ein

dem Originalnetz ähnliches Netz transformiert werden, dass dieses die Netzdarstellung von  $\vec{r}$  repräsentiert und auch die ursprünglichen Schaltvorgänge durch die Grenzstellen reproduzierbar sind: Dazu ersetzt man Transitionen  $T_i$  durch Gruppen von neuen Transitionen derart, dass jede erdenkliche Schaltfolge von  $\bullet T_i$  nach  $T_i^\bullet$  möglich ist, denn insbesondere hier, wo Marken über Grenzstellen fließen, ist die genaue Schaltsequenz für  $\vec{r}$  schließlich noch nicht bekannt. Die Anzahl der Transitionen ergibt sich aus der Anzahl der Stellen im Vor- und Nachbereich von  $T_i$ : Bei  $x := \bullet T_i$  und  $y := T_i^\bullet$  wird  $T_i$  durch insgesamt  $|x|$  Gruppen zu je  $|y|$  Transitionen ersetzt. Von jeder Eingangsstelle von  $T_i$  sollen nun Marken zu jeder Ausgangsstelle transportiert werden können. Hierzu verbindet jede der  $|y|$  Transitionen einer Gruppe die Eingangsstelle dieser Gruppe mit je einer Ausgangsstelle. Ein Beispiel ist in Abbildung 6.6 dargestellt, Algorithmus 6.2 beschreibt den Umbauprozess.

**Algorithmus 6.2:** TRANSFORMATION IN EINEN PSEUDO-SYNCHRONISATIONSGRAPHEN

---

**Eingabe:** Synchronisationsgraph  $ES^* = (S^*, T^*, F^*)$   
**Ausgabe:** Pseudo-Synchronisationsgraph  $PG = (S, T, F)$

---

```

1   $PG := ES^*$ ;
2  for  $i := 1$  to  $|T|$  do
3      if (Transition  $T[i]$  vom Typ  $T_i$ ) then
4           $x := \bullet T_i$ ;
5           $y := T_i^\bullet$ ;
6           $k := 1$ ;
7          for  $n := 1$  to  $|x|$  do
8              for  $m := 1$  to  $|y|$  do
9                   $T := T \cup \{T_i^k\}$ ;
10                  $F := F \cup \{(x[n], T_i^k), (T_i^k, y[m])\}$ ;
11                  $k := k + 1$ ;
12             endfor
13         endfor
14          $T := T \setminus \{T_i\}$ ;
15     endif
16 endfor
    
```

▶ Nur Transitionen  $T_i$  betrachten.  
 ▶ Liste der Stellen im Vorbereich von  $T_i$ .  
 ▶ Liste der Stellen im Nachbereich von  $T_i$ .  
 ▶ Für genau  $|x|$  Gruppen...  
 ▶ werden jeweils  $|y|$ ...  
 ▶ ...Transitionen erzeugt,...  
 ▶ ...Kanten generiert und...  
 ▶ ...die neuen Transitionen nummeriert.  
 ▶ Die alte Transition  $T_i$  wird (inkl. Kanten) entfernt.

---

**Beispiel 6.4** In Abbildung 6.7 ist das von Algorithmus 6.2 generierte Netz PG4 dargestellt, welches aus einem Synchronisationsgraphen ES4\* konstruiert wurde, der durch Elimination der Kantengewichte in ES4 von Seite 76 entstanden ist. Da hier  $\vec{r} = \vec{1}$

gilt, alle Transitionen des Netzes also schalten, mussten insgesamt fünf Stellen aufgrund der Kantengewichte  $> 1$  vervielfacht werden. Diese sind im Netz grau unterlegt.  $\square$

Die  $\vec{r}$  zugehörige T-Invariante in  $PG$  sei von nun an durch  $\vec{r}_{PG}$  bezeichnet, wobei die  $k$  Kopien einer ursprünglichen Transition  $t_i$  mit Schalthäufigkeit  $k$  je einmal, in der Summe also  $k$ -mal, feuern. Wir sprechen hier aus gutem Grund von einem „Pseudo-Synchronisationsgraphen“: In dem nun neu entstandenen Netz  $PG$  lässt sich in jedem Fall eine Schaltsequenz finden, die auf Basis der T-Invariante  $\vec{r}_{PG}$  schwache Erreichbarkeit nachweist: Realisierbar ist entweder  $\vec{r}_{PG}$ , dann läge starke Erreichbarkeit vor, oder ein Vielfaches von  $\vec{r}_{PG}$ . Wenn  $\vec{r}_{PG}$  selbst realisierbar ist, muss Folgendes gelten:

Jede Transition im Netz schaltet höchstens einmal:  $\vec{r}$  wurde so in  $\vec{r}_{PG}$  transformiert, dass mehrfache Schalthäufigkeiten einer Transition  $t_i$  durch das jeweils einmalige Schalten aller Kopien  $t_i^j$  realisiert werden können. Ebenso wurden alle Kantengewichte  $> 1$  eliminiert. Durch das Hinzufügen der zusätzlichen Transitionen in Algorithmus 6.2 ist die Korrektheit des Schaltvorgangs sichergestellt, er wird lediglich auf mehrere Zwischenschritte aufgeteilt.

Dies hat konkrete Auswirkungen auf die linear-algebraische Berechenbarkeit einer Lösung: Da jede Transition höchstens einmal schaltet, muss der Lösungsvektor charakteristisch sein, d.h. er darf als Komponenten nur Nullen (Schalthäufigkeit = 0) oder Einsen (Schalthäufigkeit = 1) enthalten. Zudem werden wir nun zeigen, dass eine realisierbare T-Invariante  $\vec{r}_{PG}$  folgende Eigenschaften erfüllen muss:

- (a) Die Netzdarstellung der Lösung ist ein Synchronisationsgraph.
- (b) Die Netzdarstellung der Lösung ist kreisfrei.

(a) *Die Netzdarstellung der Lösung ist ein Synchronisationsgraph.*

Bis auf die Eingangsstellen  $x_i$  und Ausgangsstellen  $y_i$  der aus  $T_i$  entstandenen Transitionsgruppen erfüllen alle anderen Stellen in  $PG$  bereits die Eigenschaft nur einer ein- und ausgehenden Kante eines Synchronisationsgraphen. Außerdem gilt bereits  $|\bullet x_i| = 1$  und  $|y_i \bullet| = 1$ .

- Jede Eingangsstelle  $x_i$  hat also genau eine eingehende Kante. Wenn jede Transition höchstens einmal schalten darf, kann jede Stelle  $x_i$  somit auch nur eine Marke erhalten, die von genau einer Transition aus der zu  $x_i$  gehörigen Transitionsgruppe wieder konsumiert wird. Damit gilt in jedem Fall  $|\bullet x_i| = |x_i \bullet| = 1$ .
- Jede Ausgangsstelle  $y_i$  der Transitionsgruppen muss eine Marke erhalten, damit ihre einzige Nachfolgetransition aktiviert wird. Würde eine Stelle mehr als eine

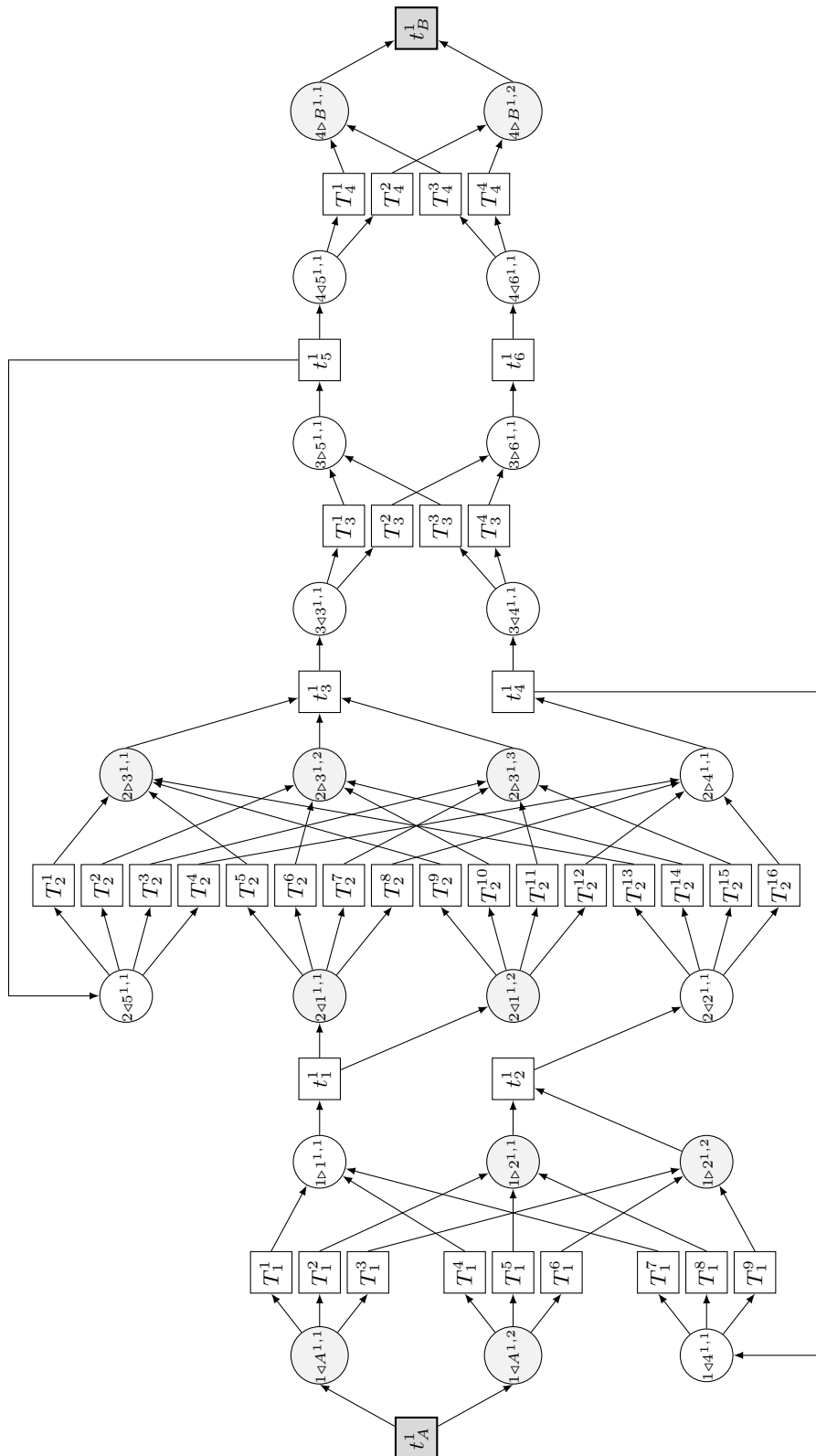


Abbildung 6.7: Pseudo-Synchronisationsgraph PG4

Marke von verschiedenen Vortransitionen aus den Transitionsgruppen bekommen, so würden diese zusätzlichen Marken auf anderen Ausgangsstellen fehlen. Eine andere Nachfolgetransition könnte demnach nicht schalten. Somit können also niemals zwei Transitionen aus den Transitionsgruppen Marken auf die gleiche Stelle legen und es gilt auch hier  $|\bullet y_i| = |y_i^\bullet| = 1$ .

Die Netzdarstellung einer realisierbaren T-Invariante  $\vec{r}_{PG}$  muss demnach ein Synchronisationsgraph sein.

(b) Die Netzdarstellung der Lösung ist kreisfrei.

Nun bleibt noch die Frage der Kreisfreiheit zu klären: Wir wissen bereits, dass ein Synchronisationsgraph lebendig ist, wenn zu Beginn alle seine Kreise ausreichend markiert sind. Die Netzdarstellung der gesuchten Lösung ist in jedem Fall ein Synchronisationsgraph, dessen Startmarkierung die leere Markierung ist. Dadurch ist zu Beginn kein Kreis ausreichend markiert. Zum Nachweis der starken Erreichbarkeit ist das Finden einer T-Invariante  $\vec{r}_{PG}$ , deren Netzdarstellung kreisfrei ist, also unabdingbar.

### 6.3 Berechnung der kreisfreien Synchronisationsgraphen

Alle T-Invarianten von  $PG$  lassen sich wie gewohnt durch Algorithmus 3.1 berechnen. Jedoch gelten aufgrund des Synchronisationsgraph-Kriteriums folgende Einschränkungen für Lösungsvektoren  $\vec{r}_{PG}$ : Im Nachbereich jeder Stelle  $x_i$  darf nur genau eine Transition schalten ( $T_i[j]$  sei hierbei die  $j$ -te Transition im Nachbereich von  $x_i$ ):

$$\sum_{j=1}^{|\bullet x_i|} \vec{r}_{PG}(T_i[j]) = 1$$

Gleichermaßen darf im Vorbereich jeder Stelle  $y_i$  nur eine Transition schalten:

$$\sum_{j=1}^{|\bullet y_i|} \vec{r}_{PG}(T_i[j]) = 1$$

**Beispiel 6.5** Für den Teilausschnitt des Netzes  $PG_7$  in Abbildung 6.6 auf Seite 84 erhält man folgende Regeln:

$$\begin{aligned} |x_i^\bullet|: \quad & \vec{r}_{PG}(T_1^1) + \vec{r}_{PG}(T_1^2) + \vec{r}_{PG}(T_1^3) = 1 \\ & \vec{r}_{PG}(T_1^4) + \vec{r}_{PG}(T_1^5) + \vec{r}_{PG}(T_1^6) = 1 \end{aligned}$$



$$\vec{r}_{PG}(T_1^7) + \vec{r}_{PG}(T_1^8) + \vec{r}_{PG}(T_1^9) = 1$$

$$\begin{aligned} |\bullet y_i|: \quad & \vec{r}_{PG}(T_1^1) + \vec{r}_{PG}(T_1^4) + \vec{r}_{PG}(T_1^7) = 1 \\ & \vec{r}_{PG}(T_1^2) + \vec{r}_{PG}(T_1^5) + \vec{r}_{PG}(T_1^8) = 1 \\ & \vec{r}_{PG}(T_1^3) + \vec{r}_{PG}(T_1^6) + \vec{r}_{PG}(T_1^9) = 1 \end{aligned}$$

□

Die Transitionen des Originalnetzes und deren  $k$  Kopien schalten je einmal. Insgesamt stellen alle potentiellen Lösungen  $\vec{r}_{PG}$  somit wie bereits erwähnt einen charakteristischen Vektor dar. Algorithmus 3.1 von Seite 23 berechnet systematisch Linearkombinationen, testet sie auf Minimalität und dividiert sie durch den größten gemeinsamen Teiler aller Komponenten. Die Inzidenzmatrix enthält nur drei verschiedenartige Einträge, nämlich 0, 1 oder  $-1$ . Durch diese Eigenschaften und obige Regeln gelten im Algorithmus nun diese Einschränkungen für die Bildung von Linearkombinationen:

- Für die Faktoren  $\alpha$  und  $\beta$  muss bei der Bildung jeder Linearkombination

$$\alpha \cdot U_{r,-}^- + \beta \cdot U_{s,-}^+$$

stets  $\alpha = \beta = 1$  gelten. Werte mit einem Betrag  $> 1$ , die auch in allen anderen Spalten als der aktuellen Spalte  $k$  durch Linearkombination entstehen können, führen sonst im Allgemeinen zu Werten  $> 1$  im neu kombinierten Vektor und dieser wäre nicht mehr charakteristisch. Vor Zeile 19 im Algorithmus kann genau dieser Sachverhalt durch eine zusätzliche Abfrage hergestellt werden: Sind die Einträge  $U_{r,j}^-$  und  $U_{s,j}^+$  in einer beliebigen anderen Spalte<sup>2</sup>  $j \neq k$  identisch und von Null verschieden, so würde die Addition der beiden Vektoren einen Wert

$$|U_{r,j}^- + U_{s,j}^+| > 1$$

erzeugen. Darum wird mit dem folgenden Code festgelegt, dass solche Vektoren nicht zulässig sind (**break**) und stattdessen mit der Bildung der nächsten Linearkombination fortgefahren wird (**continue**). Außerdem kann die Division durch den größten gemeinsamen Teiler von Zeile 25 eingespart werden.

```
boolean breakPoint := false;  
for Spalten  $j := 1$  to  $n + m$  do  
    if ( $j \neq k$  and  $U_{r,j}^- = U_{s,j}^+ \neq 0$ ) then
```

<sup>2</sup>Zur Erinnerung:  $U = [A|Y]$  ist eine  $(m \times (n + m))$ -Matrix.

```

breakPoint := true;
break;
endif
endfor
if breakPoint then
continue;
endif

```

- Die Bedingung, nur Lösungen mit  $\vec{r}_{PG}(t_A) = \vec{r}_{PG}(t_B) = 1$  zu berechnen, ist durch obigen Code gleichermaßen erfüllt: In den zulässigen T-Invarianten müssen  $t_A$  und  $t_B$  jeweils einmal schalten, um die leere Markierung zu reproduzieren. Ein Nicht-Schalten einer dieser Transitionen könnte nur — falls überhaupt möglich — durch unerlaubtes mehrfaches Schalten einer anderen Transition im Netz ausgeglichen werden.

Durch den zusätzlichen Code wird die Berechnung der „interessanten“ Lösungen erheblich verkürzt, da alle Linearkombinationen, die keine charakteristischen Vektoren darstellen, gar nicht erst gebildet werden. Da der größte Rechenaufwand in Algorithmus 3.1 durch das Bilden und Testen dieser Linearkombinationen entsteht, gewinnt er durch den zusätzlichen Code deutlich an Leistung. Die Anzahl und auch Gestalt der charakteristischen T-Invarianten lässt sich aber sogar im Voraus ermitteln:

### Aufwandsabschätzung für die Berechnung charakteristischer Vektoren

Im letzten Abschnitt wurde gezeigt, dass eine Transition  $T_i$  durch genau  $|x_i^\bullet|$  Transitionsgruppen mit je  $|y_i^\bullet|$  Transitionen ersetzt wird<sup>3</sup>. Da in zulässigen T-Invarianten nur eine Transition jeder Gruppe schalten darf, gibt es eine berechenbare Anzahl von Kombinationsmöglichkeiten:

**Satz 6.1** Sei PG ein Pseudo-Synchronisationsgraph gemäß Algorithmus 6.2 und sei L eine Liste der Transitionen vom Typ  $T_i$ , welche bei Erzeugung von Netz PG durch Transitionsgruppen ersetzt wurden. Sei  $|L|$  die Anzahl der Elemente von L. Dann ist die Anzahl der charakteristischen T-Invarianten  $\vec{r}_{PG}$  mit  $\vec{r}_{PG}(t_A) = \vec{r}_{PG}(t_B) = 1$  folgendermaßen berechenbar:

$$\prod_{k=1}^{|L|} (|\bullet L[k]|!)$$

■

<sup>3</sup>Durch ein ausgeglichenes Eingangs-/Ausgangskantenverhältnis gilt übrigens immer  $|x_i^\bullet| = |y_i^\bullet|$ .

**Beispiel 6.6** Bei der Erzeugung von Netz PG4 in Abbildung 6.7 wurden die Transitionen  $L = \{T_1, T_2, T_3, T_4\}$  durch Transitionsgruppen ersetzt. Die Anzahl der T-Invarianten gemäß Satz 6.1 ist damit:

$$\prod_{k=1}^{|L|} (|\bullet L[k]|!) = \prod_{k=1}^4 (|\bullet L[k]|!) = 3! \cdot 4! \cdot 2! \cdot 2! = 576$$

□

**Beispiel 6.7** Auch für das Netz ES7\* von Seite 83 ist  $L = \{T_1, T_2, T_3, T_4\}$ , die Anzahl der T-Invarianten jedoch aufgrund des größeren Vor- und Nachbereichs zweier Transitionen erheblich größer:

$$\prod_{k=1}^{|L|} (|\bullet L[k]|!) = \prod_{k=1}^4 (|\bullet L[k]|!) = 3! \cdot 4! \cdot 3! \cdot 4! = 20736$$

□

Die beiden Beispiele zeigen sehr deutlich, dass bereits bei sehr kleinen Netzen die Anzahl der T-Invarianten erheblich groß werden kann.

Tatsächlich müssen die T-Invarianten aber nicht berechnet, sondern können aufgrund der Regeln für die Transitionsgruppen manuell konstruiert werden:

**Satz 6.2** Sei PG ein Pseudo-Synchronisationsgraph gemäß Algorithmus 6.2 und sei  $L$  eine Liste der Transitionen vom Typ  $T_i$ , welche bei Erzeugung von Netz PG durch Transitionsgruppen ersetzt wurden. Sei  $|L|$  die Anzahl der Elemente von  $L$  und  $|\bullet T_i|!$  die Anzahl der Kombinationsmöglichkeiten in den Transitionsgruppen  $T_i$ . Dann ist die Lösungsmenge der charakteristischen T-Invarianten  $\vec{r}_{PG}$  mit

$$\vec{r}_{PG}(t_A) = \vec{r}_{PG}(t_B) = 1$$

folgendermaßen konstruierbar:

- Es wird vorab die Anzahl  $n$  der Lösungsvektoren gemäß Satz 6.1 berechnet und dementsprechend  $n$  Nullvektoren initialisiert.
- Jede Transition außerhalb einer Transitionsgruppe schaltet genau einmal. Dies wird in jeden Nullvektor eingetragen.

- Jede Kombinationsmöglichkeit von Transitionsgruppen  $T_i$  wird systematisch wiederholt in die T-Invarianten eingetragen. Bei  $|\bullet T_i|!$  Kombinationsmöglichkeiten für  $T_i$  sind dies genau  $\frac{n}{|\bullet T_i|}$  Iterationen der  $|\bullet T_i|!$  verschiedenen Kombinationen.

■

**Beispiel 6.8** Das Netz in Abbildung 6.8 ist der zu Netz 2.1 von Seite 9 gehörige Pseudo-Synchronisationsgraph. Genau eine Transition,  $T_1$ , wurde durch Transitionsgruppen ersetzt. Es gibt in diesem Fall sowohl  $n := |\bullet T_1|! = 3! = 6$  T-Invarianten sowie Kombinationsmöglichkeiten innerhalb der Transitionsgruppen. Gemäß Satz 6.2 werden nun also sechs T-Invarianten  $\vec{r}_{PG}$  initialisiert, für die jeweils gilt:

$$\vec{r}_{PG}(t_A^1) = \vec{r}_{PG}(t_B^1) = \vec{r}_{PG}(t_1^1) = \vec{r}_{PG}(t_2^1) = 1$$

Die sechs zulässigen Kombinationen für die Transitionen  $T_1^1, \dots, T_1^9$  sind:

$$(T_1^1, T_1^5, T_1^9), (T_1^1, T_1^6, T_1^8), (T_1^2, T_1^4, T_1^9), (T_1^2, T_1^6, T_1^7), (T_1^3, T_1^4, T_1^8), (T_1^3, T_1^5, T_1^7)$$

Jede Kombination wird nun in eine der sechs T-Invarianten eingetragen, für die erste Kombination erhält man also beispielsweise

$$\vec{r}_{PG}(T_1^1) = \vec{r}_{PG}(T_1^5) = \vec{r}_{PG}(T_1^9) = 1.$$

Damit sind alle möglichen charakteristischen T-Invarianten konstruiert.

□

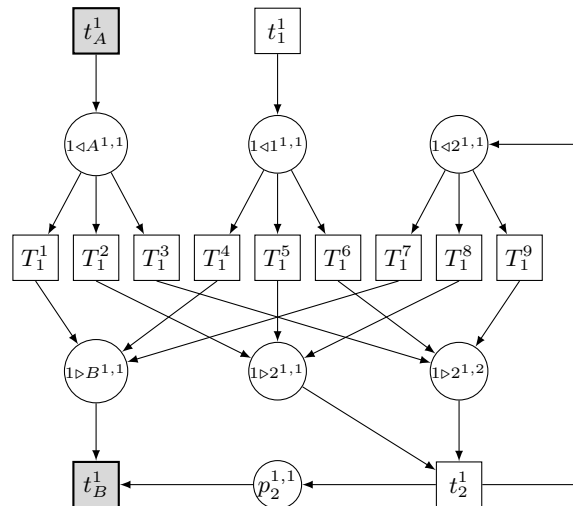


Abbildung 6.8: Pseudo-Synchronisationsgraph PG1

## 6.4 Ermitteln und „Verbieten“ der Kreise

Ein Aspekt, der noch nicht betrachtet wurde, ist die Frage nach der Kreisfreiheit der Netzdarstellungen dieser T-Invarianten. Im Nachhinein müssen viele Lösungen wieder verworfen werden und nur ein Bruchteil der T-Invarianten besitzt die gewünschte Eigenschaft: Jede Lösung gemäß Satz 6.1 ist zwar ein Synchronisationsgraph, jedoch nicht zwingend kreisfrei. Das gleiche Problem entsteht auch, wenn man statt der Berechnung *aller* T-Invarianten auf ein lineares Programm zurückgreift und nur *genau eine* Lösung ermittelt: Diese ist, ohne zusätzliche Constraints, die alle Kreise im Netz „verbieten“, im Allgemeinen nicht zyklensfrei und damit unbrauchbar. Grundsätzlich gibt es zwei Möglichkeiten, kreisfreie Lösungen zu berechnen:

- (a) Nachträgliches Löschen aller T-Invarianten aus der Lösungsmenge, deren Netzdarstellungen nicht kreisfrei sind.
- (b) „Verbieten“ aller Kreise vor der Konstruktion der T-Invarianten.

*Zu (a):*

Hat man alle T-Invarianten berechnet, kann man aus deren Menge genau diejenigen auswählen, deren Netzdarstellungen nicht kreisfrei sind. Jede Netzdarstellung einer Lösung kann mittels gängiger Graphenalgorithmien auf Zyklen getestet und die Lösung gegebenenfalls verworfen werden, die Berechnung konkreter Kreise ist gar nicht notwendig.

*Zu (b):*

Um nur T-Invarianten zu erhalten, deren Netzdarstellungen kreisfrei sind, müssen zunächst alle Kreise bekannt sein und dem Konstruktionsverfahren als zusätzliche Constraints hinzugefügt werden. In Abschnitt 5.5.1 wurde bereits beschrieben, wie sich die minimalen Kreise des Netzes über S-Invarianten berechnen lassen. Auch im Netz  $PG$  sind diese Kreise noch existent, lediglich das Schalten der Transition  $T_i$  wird durch das Schalten von Transitionen  $T_i^j$  ersetzt. Jedoch sind dies nicht die einzigen Kreise dieses Netzes. Neben den minimalen Kreisen gibt es noch eine endliche Anzahl weiterer elementarer Zyklen. Elementare Kreise sind solche Kreise, die bis auf Anfangs- und Endknoten keinen Knoten doppelt enthalten (vgl. Definition 5.6 auf Seite 60). Jeder minimale Kreis ist somit auch ein elementarer Kreis. Erst, wenn auch diese Zyklen im Netz „verboten“ sind, enthält man nur zyklensfreie Lösungen. Zur Berechnung aller elementaren Kreise sind die Algorithmen von Johnson [Joh75] und Tarjan [Tar72] empfehlenswert. Eine Implementation des Algorithmus von Johnson in Java ist unter [Mey11] zu finden.

**Beispiel 6.9** Im Netz aus Abbildung 6.7 (S. 87) existieren die folgenden minimalen Kreise (zur besseren Übersicht wurden verbindende Stellen eingespart):

- $T_1^7 \rightarrow t_1^1 \rightarrow T_2^8 \rightarrow t_4^1 \rightarrow T_1^7$
- $T_1^7 \rightarrow t_1^1 \rightarrow T_2^{12} \rightarrow t_4^1 \rightarrow T_1^7$
- $T_1^8 \rightarrow t_2^1 \rightarrow T_2^{16} \rightarrow t_4^1 \rightarrow T_1^8$
- $T_1^9 \rightarrow t_2^1 \rightarrow T_2^{16} \rightarrow t_4^1 \rightarrow T_1^9$
- $T_2^1 \rightarrow t_3^1 \rightarrow T_3^1 \rightarrow t_5^1 \rightarrow T_2^1$
- $T_2^2 \rightarrow t_3^1 \rightarrow T_3^1 \rightarrow t_5^1 \rightarrow T_2^2$
- $T_2^3 \rightarrow t_3 \rightarrow T_3^1 \rightarrow t_5 \rightarrow T_2^3$
- $T_2^4 \rightarrow t_4^1 \rightarrow T_3^3 \rightarrow t_5^1 \rightarrow T_2^4$

Einer der insgesamt zwölf zusätzlichen elementaren Kreise, berechnet mit dem Programm aus [Mey11], ist

$$T_1^7 \rightarrow t_1^1 \rightarrow T_2^5 \rightarrow t_3^1 \rightarrow T_3^1 \rightarrow t_5^1 \rightarrow T_2^4 \rightarrow t_4^1 \rightarrow T_1^7,$$

er ist nicht unter den acht minimalen S-Invarianten zu finden. Im gesamten Netz müssten also 20 elementare Kreise verboten werden.  $\square$

Wie verbietet man aber nun diese Kreise? Damit ein Kreis nicht durchlaufen werden kann, dürfen niemals alle Transitionen dieses Kreises schalten. Für jeden elementaren Kreis kann demnach folgende Regel aufgestellt werden:

**Satz 6.3** Sei  $PG$  ein Pseudo-Synchronisationsgraph gemäß Algorithmus 6.2,  $\vec{r}_{PG}$  eine charakteristische T-Invariante von  $PG$  und sei  $E$  die Liste aller elementaren Kreise von  $PG$ . Seien  $(y_1, \dots, y_n)$  die verschiedenen Transitionen eines Kreises  $e$  aus  $E$ . Dann ist dieser Kreis  $e$  genau dann nicht in  $\vec{r}_{PG}$  enthalten, wenn gilt:

$$\vec{r}_{PG}(y_1) + \vec{r}_{PG}(y_2) + \dots + \vec{r}_{PG}(y_n) \leq n - 1 \quad \blacksquare$$

**Beispiel 6.10** Für den ersten minimalen Kreis aus Beispiel 6.9,  $T_1^7 \rightarrow t_1^1 \rightarrow T_2^8 \rightarrow t_4^1 \rightarrow T_1^7$ , erhält man den Constraint  $\vec{r}_{PG}(T_1^7) + \vec{r}_{PG}(t_1^1) + \vec{r}_{PG}(T_2^8) + \vec{r}_{PG}(t_4^1) \leq 3$ , der das Schalten mindestens einer Transition dieses Kreises verbietet.  $\square$

Einen kreisfreien Lösungsvektor  $\vec{r}_{PG}$  erhält man nun einerseits über ein lineares Programm, andererseits können die Constraints dem oben beschriebenen Konstruktionsverfahren für charakteristische T-Invarianten hinzugefügt werden.

**Beispiel 6.11** Ein lineares Programm zur Berechnung einer Lösung  $\vec{r}_{PG1}$  im Netz  $PG1$  hat in diesem Fall folgende Struktur:

- (01)  $f(\vec{r}_{\text{PG1}}) = \vec{1} \circ \vec{r}_{\text{PG1}} \rightarrow \text{Min}$
- (02)  $[\text{PG1}] \cdot \vec{r}_{\text{PG1}} = \vec{0}$
- (03)  $\vec{r}_{\text{PG1}} \succ \vec{0}$
- (04)  $\vec{r}_{\text{PG1}} \prec \vec{1}$
- (05)  $\vec{r}_{\text{PG1}}(t_A^1) = \vec{r}_{\text{PG1}}(t_B^1) = \vec{r}_{\text{PG1}}(t_1^1) = \vec{r}_{\text{PG1}}(t_2^1) = 1$
- (06)  $\vec{r}_{\text{PG1}}(T_1^1) + \vec{r}_{\text{PG1}}(T_1^2) + \vec{r}_{\text{PG1}}(T_1^3) = 1$
- (07)  $\vec{r}_{\text{PG1}}(T_1^4) + \vec{r}_{\text{PG1}}(T_1^5) + \vec{r}_{\text{PG1}}(T_1^6) = 1$
- (08)  $\vec{r}_{\text{PG1}}(T_1^7) + \vec{r}_{\text{PG1}}(T_1^8) + \vec{r}_{\text{PG1}}(T_1^9) = 1$
- (09)  $\vec{r}_{\text{PG1}}(T_1^1) + \vec{r}_{\text{PG1}}(T_1^4) + \vec{r}_{\text{PG1}}(T_1^7) = 1$
- (10)  $\vec{r}_{\text{PG1}}(T_1^2) + \vec{r}_{\text{PG1}}(T_1^5) + \vec{r}_{\text{PG1}}(T_1^8) = 1$
- (11)  $\vec{r}_{\text{PG1}}(T_1^3) + \vec{r}_{\text{PG1}}(T_1^6) + \vec{r}_{\text{PG1}}(T_1^9) = 1$
- (12)  $\vec{r}_{\text{PG1}}(T_1^8) + \vec{r}_{\text{PG1}}(t_2^1) \leq 1$
- (13)  $\vec{r}_{\text{PG1}}(T_1^9) + \vec{r}_{\text{PG1}}(t_2^1) \leq 1$

Die ersten vier Zeilen beschreiben die Berechnung einer minimalen T–Invariante, die charakteristisch sein muss und aufgrund ihrer Eigenschaften in jedem Fall  $\neq \vec{0}$  und  $\neq \vec{1}$  ist. Jede Transition des ursprünglichen Netzes soll genau einmal feuern (Zeile 5). Die Zeilen 6 bis 11 repräsentieren die Regel, dass pro Transitionsgruppe nur eine Transition  $T_i^j$  auf je eine Ausgangsstelle schalten darf. Alle zu verbietenden elementaren Kreise ( $T_1^8 \rightarrow t_2^1 \rightarrow T_1^8$  und  $T_1^9 \rightarrow t_2^1 \rightarrow T_1^9$ ) sind schließlich in den Zeilen 12 und 13 als Constraints formuliert. Übrigens könnte die zu minimierende Zielfunktion aufgrund aller sehr eng gehaltenen Nebenbedingungen eingespart werden. In der Ausgabe des linearen Programms sind die schaltenden Transitionen

$$t_A^1, t_B^1, t_1^1, t_2^1, T_1^3, T_1^5, T_1^7,$$

die Netzdarstellung dieser Lösung ist ein kreisfreier Synchronisationsgraph und weist folglich starke Erreichbarkeit nach.  $\square$

**Beispiel 6.12** Bei der Berechnung *aller* minimalen T–Invarianten mit kreisfreier Netzdarstellung für Netz PG1 mit  $m$  Stellen und  $n$  Transitionen muss das Netz durch zusätzliche Stellen  $z_i$  und gegebenenfalls Transitionen, die gemeinsam die Constraints repräsentieren, zu einem neuen Netz PG1\* erweitert werden. Man erhält bei insgesamt  $c$  zusätzlichen Constraints dadurch insgesamt  $m + c$  Stellen und statt der normalen Gleichung  $[\text{PG1}] \cdot \vec{r}_{\text{PG1}} = \vec{0}$  ein Ungleichungssystem

$$[\text{PG1}^*] \cdot \vec{r}_{\text{PG1}^*} \leq \vec{b}, \quad \vec{b} = (b_1, \dots, b_m, b_{m+1}, \dots, b_c)^T,$$

wobei  $b_1 = \dots = b_m = 0$  gilt und die Werte  $b_{m+1}, \dots, b_c$  durch die Constraints festgelegt werden. Dieses Ungleichungssystem wird nun durch Subtraktion der rechten Seite  $\vec{b}$  in ein homogenes Gleichungssystem überführt,

$$[\text{PG1}^* | -\vec{b}] \cdot \begin{pmatrix} \vec{r}_{\text{PG1}^*} \\ 1 \end{pmatrix} = \vec{0},$$

wobei  $\vec{b}$  durch eine zusätzliche Transition  $b$  repräsentiert wird, die nur einmal schalten darf. Für die Modellierung der Constraints im Netz sind folgende Fälle zu unterscheiden:

- Einfache Constraints, die Gleichungen der Form

$$\sum_{k=1}^l \vec{r}_{\text{PG1}^*}(t_k) = 1, \quad l \in \mathbb{N}$$

darstellen, also Schalthäufigkeiten für ursprüngliche Transitionen oder innerhalb der Transitionsgruppen repräsentieren, werden dem Netz der Reihe nach als neue Stellen  $z_i$  mit  $\bullet z_i = \{t_1, \dots, t_l\}, z_i^\bullet = b$  hinzugefügt. Für unser Beispiel erhalten wir:

- $z_1$  mit  $\bullet z_1 = \{t_A^1\}, z_1^\bullet = b,$
- $z_2$  mit  $\bullet z_2 = \{t_B^1\}, z_2^\bullet = b,$
- $z_3$  mit  $\bullet z_3 = \{t_1^1\}, z_3^\bullet = b,$
- $z_4$  mit  $\bullet z_4 = \{t_2^1\}, z_4^\bullet = b,$
- $z_5$  mit  $\bullet z_5 = \{T_1^1, T_1^2, T_1^3\}, z_5^\bullet = b,$
- $z_6$  mit  $\bullet z_6 = \{T_1^4, T_1^5, T_1^6\}, z_6^\bullet = b,$
- $z_7$  mit  $\bullet z_7 = \{T_1^7, T_1^8, T_1^9\}, z_7^\bullet = b,$
- $z_8$  mit  $\bullet z_5 = \{T_1^1, T_1^4, T_1^7\}, z_8^\bullet = b,$
- $z_9$  mit  $\bullet z_6 = \{T_1^2, T_1^5, T_1^8\}, z_9^\bullet = b,$
- $z_{10}$  mit  $\bullet z_7 = \{T_1^3, T_1^6, T_1^9\}, z_{10}^\bullet = b.$

- Constraints, die Ungleichungen der Form

$$\sum_{k=1}^n \vec{r}_{\text{PG1}^*}(t_k) \leq b_i, \quad n \in \mathbb{N}$$

darstellen, also Kreise „verbieten“, müssen durch *Schlupfvariablen*  $sv$  ergänzt



	$t_A^1$	$t_B^1$	$t_1^1$	$t_2^1$	$T_1^1$	$T_1^2$	$T_1^3$	$T_1^4$	$T_1^5$	$T_1^6$	$T_1^7$	$T_1^8$	$T_1^9$	$sv_1$	$sv_2$	$b$
$p_{1 \triangleleft A^{1,1}}$	1	0	0	0	-1	-1	-1	0	0	0	0	0	0	0	0	0
$p_{1 \triangleleft 1^{1,1}}$	0	0	1	0	0	0	0	-1	-1	-1	0	0	0	0	0	0
$p_{1 \triangleleft 2^{1,1}}$	0	0	0	1	0	0	0	0	0	0	-1	-1	-1	0	0	0
$p_{1 \triangleright B^{1,1}}$	0	-1	0	0	1	0	0	1	0	0	1	0	0	0	0	0
$p_{1 \triangleright 2^{1,1}}$	0	0	0	-1	0	1	0	0	1	0	0	1	0	0	0	0
$p_{1 \triangleright 2^{1,2}}$	0	0	0	-1	0	0	1	0	0	1	0	0	1	0	0	0
$p_2^{1,1}$	0	-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
$z_1$	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
$z_2$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
$z_3$	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	-1
$z_4$	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1
$z_5$	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	-1
$z_6$	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	-1
$z_7$	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	-1
$z_8$	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	-1
$z_9$	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	-1
$z_{10}$	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	-1
$z_{11}$	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	-1
$z_{12}$	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	-1

 Tabelle 6.5: Inzidenzmatrix  $[\text{PG1}^* | -\vec{b}]$ 

werden. Statt  $t \leq b$  schreiben wir nun  $t + sv = b$ , wobei  $sv$  eine neue Transition repräsentiert. Für die Constraints aus unserem Beispiel erhält man damit:

- $z_{11}$  mit  $\bullet z_{11} = \{T_1^8, t_2^1, sv_1\}$ ,  $z_{11}^\bullet = b$ ,
- $z_{12}$  mit  $\bullet z_{12} = \{T_1^9, t_2^1, sv_2\}$ ,  $z_{12}^\bullet = b$ .

Die Inzidenzmatrix dieses erweiterten Netzes  $\text{PG1}^*$  ist in Tabelle 6.5 abgebildet. Berechnet man für dieses Netz die minimalen T-Invarianten, so erhält man zwei Lösungen  $(t_A^1 \ t_1^1 \ T_1^2 \ T_1^6 \ t_2^1 \ T_1^7 \ t_B^1)$  und  $(t_A^1 \ t_1^1 \ T_1^3 \ T_1^5 \ t_2^1 \ T_1^7 \ t_B^1)$ , die kreisfreie Synchronisationsgraphen darstellen und damit starke Erreichbarkeit nachweisen. Die Transitionen  $b$ ,  $sv_1$  und  $sv_2$  können nachträglich wieder gestrichen werden. Die zweite T-Invariante entspricht dem Ergebnis von Beispiel 6.11.  $\square$

Die bereits durch die Inzidenzmatrix angedeutete, sehr unübersichtliche Struktur von Netz  $\text{PG1}^*$  zeigt sehr deutlich, wie schnell durch Modellierung der Constraints im Netz dessen Größe wächst. Bei Netzen mit einer viel größeren Anzahl elementarer Kreise wird zwar die in Satz 6.1 definierte Gesamtzahl von T-Invarianten auf die kreisfreien reduziert, jedoch wird die Inzidenzmatrix durch die modellierten Constraints zunächst

erheblich erweitert. Außerdem sind die Lösungen hier im Allgemeinen nicht mehr charakteristisch, da die Transitionen  $sv_i$  möglicherweise mehr als einmal schalten müssen. Nur die „ursprünglichen“ Transitionen und die „rechte Seite“  $b$  dürfen höchstens einmal schalten, der gerade erst vereinfachte Algorithmus müsste demnach für diese Bedingungen wieder erweitert werden.

## 6.5 Zusammenfassung und Fazit

Das gesamte in diesem Kapitel 6 vorgestellte Verfahren basiert auf der Annahme, eine T-Invariante  $\vec{r}$  gefunden beziehungsweise berechnet zu haben, die schwache Erreichbarkeit belegt und sich als vielversprechend zum Nachweis starker Erreichbarkeit erweist. Durch kluge Betrachtung der S-Invarianten und Übergänge an Grenzstellen konnte aus minimalen T-Invarianten eine neue Linearkombination  $\vec{r}$  gebildet werden. Anschließend wurde ein Pseudo-Synchronisationsgraph konstruiert, der die Netzdarstellung von  $\vec{r}$  unter Elimination aller Kantengewichte repräsentiert, wobei jede Transition im Netz höchstens einmal feuern darf und damit Schalthäufigkeiten  $\vec{r}(t_i) > 1$  durch Kopien von  $t_i$  im Netz verteilt werden.

Die Berechnung von  $\vec{r}$  bildet den ersten Schwachpunkt des Verfahrens: Bei einer großen Anzahl minimaler T-Invarianten ist das Auffinden einer möglicherweise „passenden“ Linearkombination nicht trivial. Der zweite Schwachpunkt ist eindeutig durch die Größe des Netzes und Anzahl der Eingangs- und Ausgangskanten einer Grenzstelle gegeben: Bei einer großen Menge von Grenzstellen mit vielen Eingangs- und Ausgangskanten wird der entstandene Pseudo-Synchronisationsgraph so groß, dass die Anzahl seiner charakteristischen T-Invarianten den Aufwand einer Berechnung keinesfalls rechtfertigt. Selbst das Hinzufügen von Constraints sowohl direkt ins Netz als auch in ein lineares Programm führt schon bei sehr kleinen Netzen zur Erkenntnis, stattdessen besser durch manuelle oder automatisierte Simulation mit  $\vec{r}$  auf starke Erreichbarkeit zu testen. Der einzige Vorteil einer tatsächlich errechneten kreisfreien Lösung dieses Verfahrens ist die Gewinnung einer eindeutigen Schaltfolge, die ins ursprüngliche Netz übertragbar ist.

Durch gezielte Betrachtung bestimmter (möglicherweise linearkombinierter) T-Invarianten gelangt man letzten Endes demnach immer an den Punkt, an dem Simulationsschritte sinnvoller als rein linear-algebraische Berechnungen sind. Es bleibt nun noch die Frage zu klären, ob ohne Betrachtung bestimmter T-Invarianten, aber bereits nachgewiesener schwacher Erreichbarkeit, trotzdem ein rein rechnerisches Verfahren existiert, um starke Erreichbarkeit nachzuweisen.

## 7 Lösungsansatz II: Entfaltung des Netzes in kreisfreie Schichten

Die Ergebnisse der vorherigen Kapitel zeigen mehr als deutlich, dass die Suche und Betrachtung bestimmter T-Invarianten einen Schwachpunkt jedes vorgestellten Verfahrens darstellt. Darum soll bei diesem Ansatz die Analyse bestimmter T-Invarianten völlig außer Acht gelassen werden. Lediglich schwache Erreichbarkeit muss gegeben und das Netz um die Transitionen  $t_A$  und  $t_B$  erweitert worden sein.

Nun wird ein solches S/T-Netz in kreisfreie Schichten (d.h. miteinander verbundene Kopien seiner selbst) entfaltet, indem seine Kreise hinter geeigneten Transitionen aufgeschnitten werden: Diese von Transitionen ausgehenden Kanten werden so umgeleitet, dass sie nicht in Stellen der *gleichen* Schicht, sondern in Stellen der Schicht *darunter* enden. Die Grundlagen dieser Entfaltung werden in Kapitel 7.1 beschrieben. Auf der untersten Schicht bleibt damit nur die Wahl, die Kreise zu erhalten oder das Netz mit Hilfe des in Abschnitt 5.5.2 vorgestellten Verfahrens in ein zyklensfreies Netz zu transformieren.

In jedem auf diese Art entfalteteten Netz existiert bei gegebener schwacher Erreichbarkeit eine T-Invariante  $\vec{r}$ , die die leere Markierung reproduziert. Auch starke Erreichbarkeit muss, falls gegeben, auf diese Weise nachweisbar sein. Zur Berechnung der T-Invarianten wird jeweils ein ganzzahliges lineares Optimierungsproblem gelöst. Abhängig davon, ob die Kreise auf der untersten Schicht erhalten oder eliminiert wurden, müssen verschiedene Nebenbedingungen definiert werden. In den Kapiteln 7.2 und 7.3 werden die zwei Verfahren und deren Besonderheiten besprochen.

### 7.1 Grundlagen der Entfaltung

Die Entfaltung des Netzes erfolgt nach klar definierten Regeln. Grundsätzlich wird ein Netz  $N_{AB}$  zunächst unverändert als Schicht 0 des entfalteteten Netzes  $N_{AB}|_0$  definiert, Stellen  $i$  und Transitionen  $j$  wollen wir hier für die eindeutige Zuordnung in  $p_{i|0}$  und  $t_{j|0}$  umbenennen. Jede neu hinzukommende Schicht  $n$  mit Stellen  $p_{i|n}$  und Transitionen  $t_{j|n}$  ist eine exakte Kopie von Netz  $N_{AB}$  mit der zusätzlichen Eigenschaft der Kreisfreiheit.

Dazu wird nach dem Verfahren aus Kapitel 5.5.2 einmalig ermittelt, hinter welchen Transitionen welche Kanten „aufgeschnitten“ werden. Auf jeder Schicht  $n > 0$  wird nun jede Kante  $(t_j|_n, p_i|_n)$  dieser Art durch eine Kante  $(t_j|_n, p_i|_{n-1})$  ersetzt, sie wird also auf die darunter liegende Schicht „umgeleitet“. Die Transitionen  $t_A$  und  $t_B$  existieren jeweils nur einmal im Netz, wobei  $t_B$  mit den Stellen auf Schicht 0 und  $t_A$  immer mit den Stellen der aktuell obersten Ebene verbunden ist.

Somit entspricht die Reproduktion der leeren Markierung in diesem Netz dem „Herunterreichen“ der Marken von der obersten in die unterste Schicht durch die Realisierung einer bestimmten T-Invariante, in der  $t_A$  und  $t_B$  gleich oft schalten: Die der T-Invariante entsprechende Schaltsequenz im ursprünglichen Netz  $N_{AB}$  wird auf die einzelnen Schichten verteilt realisiert. Damit keine Marken im Netz „hängen bleiben“, weil sie keine Transition in ihrer Schicht aktivieren, werden gleiche Stellen  $p_i$  auf übereinander liegenden Schichten  $n$  und  $n + 1$  noch durch Hilfstransitionen  $c_{i|n+1,n}$  verbunden.

Eine spezielle Herausforderung stellt die Frage dar, ob auf Schicht 0 besser ein zyklenfreies Netz gemäß Algorithmus 5.5 erzeugt wird oder die Kreise erhalten bleiben sollten. Dies wird, wie bereits erwähnt, in den Abschnitten 7.2 und 7.3 diskutiert. Zunächst beschreibt Algorithmus 7.1 das Hinzufügen einer neuen Schicht zu einem Netz  $N_{AB}|_n$  mit bereits  $n$  Ebenen. Als Eingabeparameter erhält er  $N$  (Vorlage für die Erzeugung der Kopien) und  $N_{AB}|_n$  mit aktuell  $n$  Schichten sowie die Liste  $K$  der aufzuschneidenden Kanten aus Algorithmus 5.4.

**Algorithmus 7.1:** ADD-LAYER — HINZUFÜGEN EINER NEUEN SCHICHT

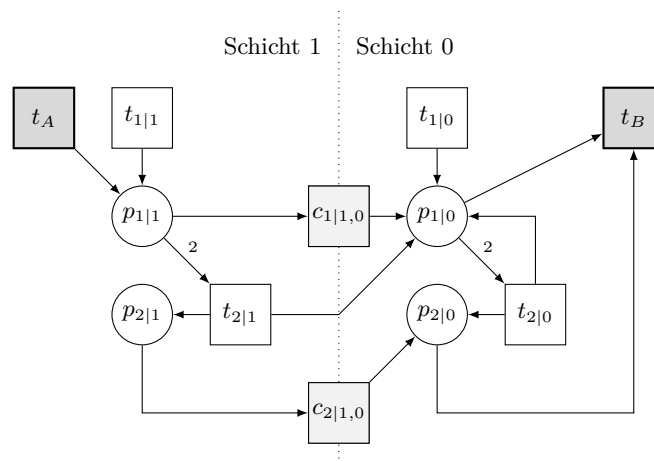
<b>Eingabe:</b> Netz $N = (S, T, F, W)$ , $N_{AB} _n$ , $K$ .	
<b>Ausgabe:</b> Netz $N_{AB} _{n+1} = (S^*, T^*, F^*, W^*)$	
1	<b>for</b> $i := 1$ <b>to</b> $ S $ <b>do</b>
2	$p_i := p_{i n+1}$ ; <span style="float: right;">▶ Umbenennung von Stellen.</span>
3	<b>endfor</b>
4	<b>for</b> $j := 1$ <b>to</b> $ T $ <b>do</b>
5	$t_j := t_{j n+1}$ ; <span style="float: right;">▶ Umbenennung von Transitionen.</span>
6	<b>endfor</b>
7	$N_{AB} _{n+1} := N \cup N_{AB} _n$ ; <span style="float: right;">▶ <math>N</math> als Schicht <math>n + 1</math> hinzufügen.</span>
8	<b>for</b> $i := 1$ <b>to</b> $ S $ <b>do</b> <span style="float: right;">▶ Hilfstransitionen <math>c_i</math> hinzufügen.</span>
9	$T^* := T^* \cup \{c_{i n+1,n}\}$ ;
10	$F^* := F^* \cup \{(p_{i n+1}, c_{i n+1,n}), (c_{i n+1,n}, p_{i n})\}$ ;
11	$W^*((p_{i n+1}, c_{i n+1,n})) := 1$ ;
12	$W^*((c_{i n+1,n}, p_{i n})) := 1$ ;

```

13 endfor
14 for  $i := 1$  to  $|K|$  do                                     ▶ Kreisfreiheit von Schicht  $n + 1$  herstellen.
15   if  $(K[i] = \text{Stelle vom Typ } p_{x \triangleleft x.1})$  then
16     continue;
17   else if  $(K[i] = \text{Stelle vom Typ } p_{x \triangleleft y} \text{ oder } p_x)$  then
18      $t_y := \bullet K[i]$ ;
19      $F^* := F^* \cup \{(t_y|_{n+1}, p_x|_n)\}$ ;                               ▶ Neue Kante hinzufügen.
20      $W^*((t_y|_{n+1}, p_x|_n)) := W^*((t_y|_{n+1}, p_x|_{n+1}))$ ;
21      $F^* := F^* \setminus \{(t_y|_{n+1}, p_x|_{n+1})\}$ ;                       ▶ Alte Kante löschen.
22   endif
23 endfor
24 Sei  $L := t_A^\bullet$  eine Liste der Stellen im Nachbereich von  $t_A$ ;
25 for  $i := 1$  to  $|L|$  do                                       ▶  $t_A$  auf oberste Schicht verschieben.
26    $p_x|_n := L[i]$ ;
27    $F^* := F^* \cup \{(t_A, p_x|_{n+1})\}$ ;
28    $W^*((t_A, p_x|_{n+1})) := W^*((t_A, p_x|_n))$ ;
29    $F^* := F^* \setminus \{(t_A, p_x|_n)\}$ ;
30 endfor

```

**Beispiel 7.1 (Hinzufügen einer weiteren Schicht)** Abbildung 7.1 zeigt das um eine Schicht erweiterte Netz  $N1$  aus Abbildung 2.1 von Seite 9. Hier wurde der einzige Kreis,  $p_1 \rightarrow t_2 \rightarrow p_1$ , in der untersten Ebene noch nicht aufgeschnitten. Der erweiterte Synchronisationsgraph  $ES1$ , über dessen S-Invarianten dieser Kreis ermittelt wurde, ist in Abbildung 2.3 auf Seite 10 zu finden. Auf Ebene 1 verläuft die Kante von  $t_{2|1}$

Abbildung 7.1: Netz  $N1|_1$

nun aber nicht mehr nach  $p_{1|1}$ , sondern wird nach  $p_{1|0}$  umgeleitet. Damit ist diese Ebene kreisfrei. Die Hilfstransitionen  $c_i$  sind grau unterlegt. Das Hinzufügen weiterer Schichten erfolgt nach dem gleichen Prinzip.  $\square$

In Algorithmus 5.4 wurde bereits festgelegt, dass alle ermittelten Kreise immer hinter Transitionen aufgeschnitten werden müssen. Warum die Kreise nicht hinter Stellen unterbrochen werden dürfen, ist in Abbildung 7.2 schnell ersichtlich. Sie entspricht dem Netz aus Abbildung 7.1, wobei der Kreis  $p_1 \rightarrow t_2 \rightarrow p_1$  auf der ersten Schicht nicht hinter  $t_2$ , sondern hinter  $p_1$  aufgeschnitten wurde. Dies führt jedoch zu einer strukturellen Veränderung im Netz, die die Schaltregel für  $t_2$  verändert und darum unzulässig ist:

Transition  $t_{2|1}$  fehlt eine eingehende Kante; sie wird dadurch auf der obersten Schicht zu einer immer aktivierten Randtransition und kann unabhängig von der Markenzahl auf  $p_{1|1}$  schalten. Auf Schicht 0 hat die Kopie dieser Transition,  $t_{2|0}$ , eine zusätzliche eingehende Kante und verletzt damit ebenfalls die ursprüngliche Struktur des Netzes. Solche Unzulässigkeiten treten immer auf der obersten Schicht  $n$  und untersten Schicht 0 in einem entfaltetem Netz auf. Zwischenschichten  $i$  mit  $0 < i < n$  sind davon nicht betroffen.

Im Gegensatz hierzu ändert das Aufschneiden *hinter* Transitionen nichts an Struktur und Verhalten eines Netzes: Die Anzahl der Kanten im Vor- und Nachbereich einer Transition ist auf jeder Schicht identisch, und über die Hilfstransitionen  $c_i$  können übrig gebliebene Marken auf einer Stelle von Schicht zu Schicht weitergereicht werden. Jedes nach dieser Regel entfaltete Netz ist somit also verhaltensgleich zum Original und muss eine T-Invariante besitzen, die schwache und gegebenenfalls starke Erreichbarkeit nachweist.

Nun muss präziser beschrieben werden, wie sich starke Erreichbarkeit für ein mit Hilfe von Algorithmus 7.1 entfaltetes Netz mit einer bestimmten Anzahl von  $n$  Schichten nachweisen lässt. Gesucht ist genau eine nichtnegative, *realisierbare* T-Invariante  $\vec{r}$ , in der  $t_A$  und  $t_B$  genau einmal schalten und deren Netzdarstellung frei von Deadlocks und Traps ist. Die Schalthäufigkeiten der einzelnen Transitionen in  $\vec{r}$  lassen sich in eine ebenso realisierbare T-Invariante  $\vec{r}_{orig}$  des ursprünglichen Netzes umrechnen: In  $\vec{r}_{orig}$  müssen  $t_A$  und  $t_B$  ebenfalls genau einmal schalten; zusätzlich gilt für jede Transition  $t_i$  des Originalnetzes:

$$\vec{r}_{orig}(t_i) = \sum_{k=0}^n (t_{i|k})$$

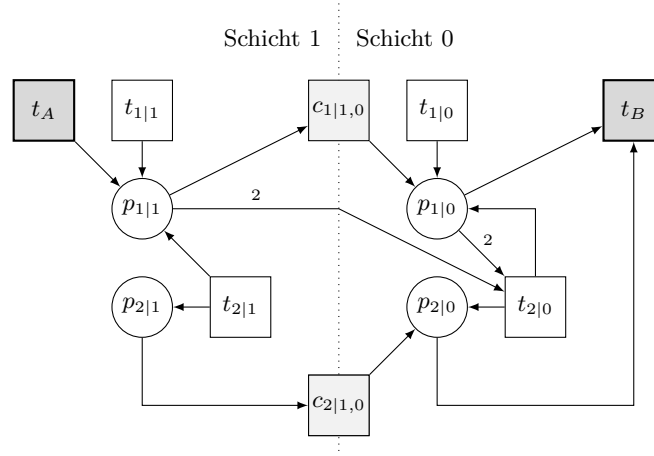


Abbildung 7.2: Netz  $N1|_1$  mit hinter Stellen aufgeschnittenen Kanten

Die Schalthäufigkeiten aller Kopien einer Transition  $t_i$  auf den einzelnen Schichten 0 bis  $n$  werden folglich addiert. Die Hilfstransitionen  $c_i$  existieren im ursprünglichen Netz nicht und werden ignoriert.

**Beispiel 7.2 (Umrechnung der T-Invariante  $\vec{r}$  in  $\vec{r}_{orig}$ )** Eine realisierbare T-Invariante, die starke Erreichbarkeit für das Netz in Abbildung 7.1 nachweist, ist

$$\begin{aligned}\vec{r} &= (t_A \ t_B \ t_{1|0} \ t_{2|0} \ c_{1|1,0} \ c_{2|1,0} \ t_{1|1} \ t_{2|1})^T \\ &= (1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1)^T.\end{aligned}$$

Damit ist  $\vec{r}_{orig}$  also:

$$\begin{aligned}\vec{r}_{orig} &= (t_A \ t_B \ t_1 \ t_2)^T \\ &= \left(1 \ 1 \ \sum_{k=0}^1 (t_{1|k}) \ \sum_{k=0}^1 (t_{2|k})\right)^T \\ &= (1 \ 1 \ t_{1|0} + t_{1|1} \ t_{2|0} + t_{2|1})^T \\ &= (1 \ 1 \ 1 \ 1)^T\end{aligned}$$

□

Da nur genau eine T-Invariante gesucht wird, ist es ratsam, deren Berechnung als ganzzahliges Optimierungsproblem zu formulieren. Das Verhalten des Programms, das das GLOP löst, ist von den Eingabeparametern und insbesondere von der Struktur der untersten Schicht (Erhaltung der Kreise oder Kreisfreiheit) abhängig. Darum werden diese beiden Fälle in den Abschnitten 7.2 und 7.3 einzeln betrachtet.

## 7.2 Entfaltung in ein vollständig kreisfreies Netz

Um Zyklensfreiheit auch auf der untersten Schicht eines entfalteten Netzes zu gewährleisten, kommt der auf Seite 67 vorgestellte Algorithmus 5.5 zur Anwendung: Anhand der Liste  $K$  der aufzuschneidenden Kanten, die schon für das Aufschneiden der Kreise auf höheren Schichten herangezogen wurde, wird nun auch Schicht 0 durch zwei zusätzliche Transitionen und eine Stelle pro Kreis in ein zyklensfreies Äquivalent transformiert:

**Definition 7.1 (kreisfreies Schichtenetz)** Sei  $N_{AB} = (S, T, F, W)$  ein  $S/T$ -Netz, in dem schwache Erreichbarkeit gegeben ist. Ein kreisfreies Schichtenetz  $N_{AB}|_n^{cf} = (S^*, T^*, F^*, W^*)$  erhält man aus  $N$  und der Liste  $K$  der aufzuschneidenden Kanten, indem man gemäß Algorithmus 7.1 nacheinander  $n$  Schichten zu  $N_{AB}$  hinzufügt und schließlich mit Algorithmus 5.5 die Kanten auf Schicht 0 ersetzt. Hierzu wird Algorithmus 5.5 nur auf Schicht 0 angewendet, die Umbenennungen „ $|_0$ “ der Stellen auf der untersten Schicht werden hierbei ignoriert. ■

Eine kreisfreie Version von Netz  $N1|_1^{cf}$  (cf steht hierbei für cycle-free, d.h. zyklensfrei) zeigt Abbildung 7.3. Dem einmaligen Schalten von Transition  $t_{2|0}$  entspricht nun das je einmalige Schalten von  $t_{2|0}$ ,  $t'_{2,1}$  und  $t''_{2,1}$ . Die Netzdarstellung jeder  $T$ -Invariante dieses Netzes ist ebenfalls kreisfrei und die Zustandsgleichung ist in diesem Fall ein hinreichendes Kriterium für den Nachweis starker Erreichbarkeit. Wie lässt sich aber nun starke Erreichbarkeit im Allgemeinen nachweisen? Das ganzzahlige lineare Optimierungsproblem enthält verschiedene Constraints, die im Folgenden aufgeführt sind.

### 7.2.1 Nebenbedingungen des ganzzahligen linearen Programms

Damit eine realisierbare  $T$ -Invariante  $\vec{r}$  überhaupt starke Erreichbarkeit nachweist, muss zum einen  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  gelten. Zum anderen ist das Ziel eine Lösung zu finden, in der die Hilfstransitionen  $t'_i, t''_i$  auf der untersten Schicht *nicht* schalten — denn ein Schalten dieser Transitionen kann dem „Leihen“ ( $t''_i$  liefert Marken, die gebraucht werden) und „Zurückgeben“ ( $t'_i$  gibt die Marken zurück) von Marken entsprechen und simuliert somit das Durchlaufen eines Kreises. Feuert keine dieser Hilfstransitionen, ist das erste Teilziel also erreicht. Alle notwendigen Bedingungen sind:

- $\vec{r}(t'_i) = \vec{r}(t''_i) \forall t_i$

Die Transitionen  $t'_i$  und  $t''_i$  müssen gleich oft schalten, da sie gemeinsam das Feu-



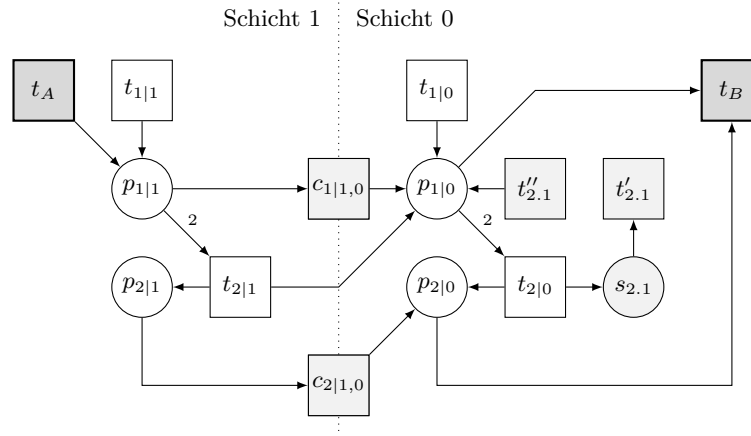


Abbildung 7.3: Zyklenfreies Netz  $N1|1|_1^{cf}$

ern der Transition  $t_i$  vervollständigen. Es dürfen nicht mehr Marken produziert als konsumiert werden und umgekehrt.

- $\vec{r}(t_A) = \vec{r}(t_B) \geq 1$   
 $t_A$  und  $t_B$  müssen ebenfalls gleich oft schalten, jedoch mindestens einmal. Im Fall starker Erreichbarkeit wird eine Lösung gefunden mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$ . Falls nur schwache Erreichbarkeit vorliegt, existiert eine Lösung  $k \cdot \vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = k > 1$ .
- $\sum_i (\vec{r}(t'_i) + \vec{r}(t''_i)) \rightarrow \text{Min}$

Eine mindestens schwache, im besten Fall starke Erreichbarkeit nachweisende Lösung ist genau dann gefunden, wenn keine der Hilfstransitionen  $t'_i, t''_i$  in der T-Invariante  $\vec{r}$  schaltet. Als Zielfunktion des GLOP gilt es also die Summe der Schalthäufigkeiten dieser Transitionen zu minimieren. Bei einem Zielfunktionswert von 0 erhält man das gewünschte Resultat.

### 7.2.2 Nachweis schwacher Erreichbarkeit

Da bei diesem Verfahren bereits davon ausgegangen wird, dass schwache Erreichbarkeit gegeben ist, muss diese auch zwingend als Lösung eines GLOP gefunden werden. Die Suche nach einer solchen Lösung verläuft folgendermaßen: Zunächst wird Netz  $N_{AB}$  als Schicht 0 des zu entfaltenden Netzes  $N_{AB}|_0^{cf}$  festgelegt, Stellen  $p_i$  und Transitionen  $t_j$  wollen wir von nun an  $p_{i|0}$  und  $t_{j|0}$  nennen. Es wird um die nötigen Hilfstransitionen  $t'_i, t''_i$  gemäß Liste  $K$  (vgl. Algorithmus 5.4 auf Seite 64) ergänzt. Anschließend werden

mit Hilfe von Algorithmus 7.1 von Seite 100 nach und nach neue Schichten hinzugefügt. Nach jeder neuen Schicht  $n$  wird  $N_{AB}|_n^{cf}$  mit den Constraints und der Zielfunktion als GLOP formuliert und nach einer Lösung gesucht. Wird keine Lösung mit Zielfunktionswert 0 gefunden, muss eine weitere Schicht hinzugefügt werden. Nach einer bestimmten Anzahl von  $m$  Schichten wird schließlich eine T-Invariante berechnet, in der alle  $t'_i, t''_i$  nicht schalten. Dann ist schwache Erreichbarkeit nachgewiesen.

**Beispiel 7.3 (Nachweis schwacher Erreichbarkeit)** Das in Abbildung 7.4 dargestellte Netz  $R2|_1^{cf}$  ist die um einfache Stellen reduzierte und um  $t_A$  und  $t_B$  ergänzte Variante des Netzes N2 von Seite 14. Eine zusätzliche Schicht wurde bereits hinzugefügt und Kreisfreiheit hergestellt. In diesem Netz ist nur schwache Erreichbarkeit gegeben. Das GLOP sieht für die gesuchte T-Invariante  $\vec{r}$  folgendermaßen aus (Ganzzahligkeit und Nichtnegativität von  $\vec{r}$  sind hier nicht explizit aufgeführt):

$$\begin{aligned} \vec{r}(t'_{4.1}) + \vec{r}(t''_{4.1}) &\rightarrow \text{Min} \\ \vec{r}(t'_{4.1}) &= \vec{r}(t''_{4.1}) \\ \vec{r}(t_A) = \vec{r}(t_B) &\geq 1 \\ [R2|_1^{cf}] \cdot \vec{r} &= \vec{0} \end{aligned}$$

Als Ergebnis bei einer zusätzlichen Schicht liefert das *IntegerProgramming*-Paket aus [Gen05] den Lösungsvektor

$$\vec{r} = (t_A \ t_B \ t_{4|1} \ t_{4|0} \ c_{1|1,0} \ c_{2|1,0} \ t'_{4.1} \ t''_{4.1})^T = (1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1)^T$$

und einen Zielfunktionswert von 2 (da  $\vec{r}(t'_{4.1}) = \vec{r}(t''_{4.1}) = 1$ ). Nachdem  $t_A$  geschaltet hat, verschiebt die Hilfstransition  $c_{1|1,0}$  die erzeugte Marke von  $p_{1|1}$  auf  $p_{1|0}$ . Anschließend muss  $t''_{4.1}$  schalten um zwei Marken zu leihen, die zum Aktivieren von  $t_{4|0}$  benötigt werden. Nach dem Schalten von  $t_{4|0}$  wird dieser Markenüberschuss von  $t'_{4.1}$  getilgt. Dies entspricht dem Durchlaufen der Schlinge  $p_1 \rightarrow t_4 \rightarrow p_1$  in R2. Anschließend kann  $t_B$  schalten und die leere Markierung ist reproduziert.

Da der Zielfunktionswert noch nicht 0 ist, wird noch eine weitere Schicht hinzugefügt und  $R2|_1^{cf}$  zu  $R2|_2^{cf}$  erweitert (Abbildung 7.5). Für dieses Netz wird nun die Lösung

$$\begin{aligned} \vec{r} &= (t_A \ t_B \ t_{4|2} \ t_{4|1} \ t_{4|0} \ c_{1|1,0} \ c_{2|1,0} \ c_{1|2,1} \ c_{2|2,1} \ t'_{4.1} \ t''_{4.2})^T \\ &= (2 \ 2 \ 1 \ 1 \ 0 \ 0 \ 2 \ 0 \ 1 \ 0 \ 0)^T \end{aligned}$$

mit dem gewünschten Zielfunktionswert 0 berechnet. Die Schaltsequenz, die die leere

Markierung reproduziert, ist:

$$t_A \ t_A \ t_{4|2} \ t_{4|1} \ c_{2|2,1} \ c_{2|1,0} \ c_{2|1,0} \ t_B \ t_B$$

Das Durchlaufen des Kreises  $p_1 \rightarrow t_4 \rightarrow p_1$  wurde auf die Ebenen 2 bis 0 entfaltet:

$$p_{1|2} \rightarrow t_{4|2} \rightarrow p_{1|1} \rightarrow t_{4|1} \rightarrow p_{1|0}$$

Jedoch wird der Kreis insgesamt zweimal durchlaufen:  $t_A$  muss zweimal schalten, damit für  $t_{4|2}$  genügend Marken verfügbar sind. Dieser Überschuss wird durch ein zusätzliches Schalten von  $t_{4|1}$  und zweimaliges Schalten von  $t_B$  ausgeglichen. Es gibt auch keine Möglichkeit, die Schalthäufigkeiten von  $t_A$  und  $t_B$  auf 1 zu reduzieren:  $t_4$  muss auf jeden Fall auf irgendeiner Schicht schalten, um die Marke auf  $p_2$  zu erzeugen, die  $t_B$  benötigt. Stelle  $p_1$  ist auf jeder Schicht immer die einzige Stelle im Vorbereich von  $t_4$ , und auf der obersten Ebene kann nur  $t_A$  als einzige Vortransition von  $p_1$  die zwei Marken bereitstellen, die  $t_4$  aktivieren. Transition  $t_A$  muss also definitiv zweimal schalten, was sich auch durch das Hinzufügen weiterer Schichten nicht ändert. In R2 liegt also nur schwache, aber keine starke Erreichbarkeit vor.  $\square$

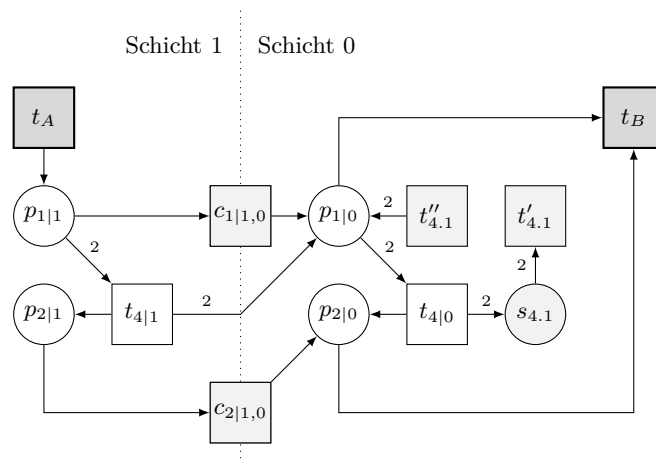


Abbildung 7.4: Erweiterung um eine Schicht: Netz  $R2|_1^{cf}$

Algorithmus 7.2 fasst nun die einzelnen Schritte zum Nachweis der schwachen Erreichbarkeit zusammen. Begonnen wird mit der untersten, kreisfreien Schicht  $N_{AB}|_0^{cf}$  eines Netzes  $N_{AB}$ . Das Programm, das das ganzzahlige Optimierungsproblem löst — im Algorithmus GLOP genannt — wird mit der Inzidenzmatrix  $[N_{AB}|_n^{cf}]$  des aktuellen Netzes, der Zielfunktion ZF und der Liste cons der Nebenbedingungen aufgerufen. Als

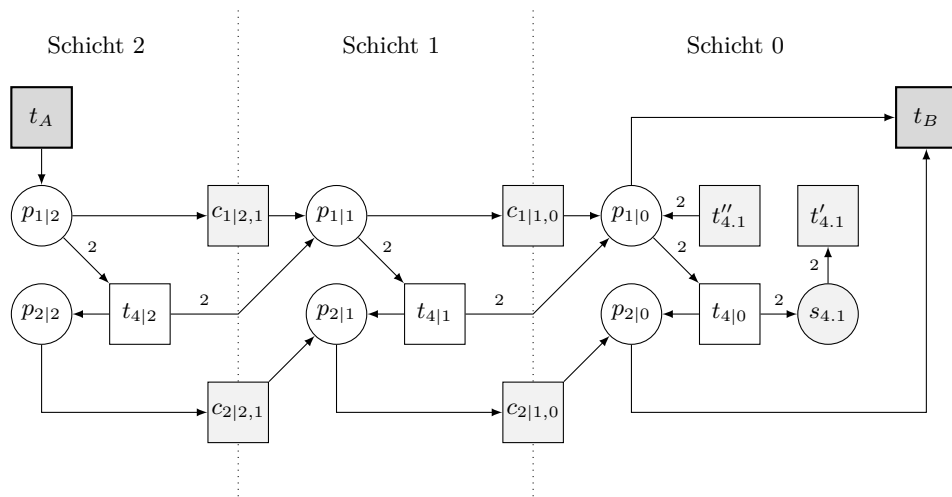


Abbildung 7.5: Erweiterung um 2 Schichten: Netz  $R2|_2^{cf}$

Ergebnis liefert es immer ein Tupel  $(\vec{r}, \text{zfwert})$ , bestehend aus einer T-Invariante  $\vec{r}$  und dem berechneten Zielfunktionswert  $\text{zfwert}$ .

**Algorithmus 7.2:** NACHWEIS SCHWACHER ERREICHBARKEIT IM KREISFREIEN NETZ

**Eingabe:**  $N, N_{AB}|_0^{cf} = (S, T, F, W), K$ .

**Ausgabe:** T-Invariante  $\vec{r}$ , die schwache Erreichbarkeit nachweist, und zugehöriges Netz  $N_{AB}|_m^{cf}$

```

1 ZF := 0;                                     ► Initialisierung.
2 cons := { $\vec{r}(t_A) = \vec{r}(t_B) \geq 1$ };
3 for i := 1 to |T| do                          ► Zielfunktion und Constraints generieren.
4   if (Transition i ist vom Typ  $t'_i$ ) then
5     ZF := ZF +  $\vec{r}(t'_i) + \vec{r}(t''_i)$ ;
6     cons := cons  $\cup$  { $\vec{r}(t'_i) = \vec{r}(t''_i)$ };
7   endif
8 endfor
9  $N_{AB}|_n^{cf} := N_{AB}|_0^{cf}$ ;                    ► n := 0
10  $(\vec{r}, \text{zfwert}) := \text{GLOP}(N_{AB}|_n^{cf}, \text{ZF} \rightarrow \text{Min}, \text{cons})$ ;    ► GLOP für Lage 0 lösen.
11 while (zfwert  $\neq$  0) do                       ► Solange zfwert  $\neq$  0...
12    $N_{AB}|_{n+1}^{cf} := \text{ADD-LAYER}(N, N_{AB}|_n^{cf}, K)$ ;    ► ...weitere Schicht (mit Alg. 7.1) hinzufügen...
13    $(\vec{r}, \text{zfwert}) := \text{GLOP}(N_{AB}|_{n+1}^{cf}, \text{ZF} \rightarrow \text{Min}, \text{cons})$ ;    ► ...und jeweils das GLOP lösen.
14   n := n + 1;                                ► Schichtenanzahl hochzählen.
15 endwhile

```

### 7.2.3 Nachweis starker Erreichbarkeit

Nun stellt sich die Frage, ob starke Erreichbarkeit durch ein ebenso einfaches Verfahren nachweisbar ist. Dies ist jedoch zu verneinen. Die Suche nach einer Lösung  $\vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$ , in der die Hilfstransitionen  $t'_i, t''_i$  nicht feuern, gestaltet sich insofern schwieriger, dass eine solche Lösung gar nicht existieren muss. Im allgemeinen Fall wird möglicherweise — dies muss aber nicht der Fall sein — zunächst eine schwache Erreichbarkeit nachweisende Lösung  $k \cdot \vec{r}$ ,  $k > 1$ , mit Zielfunktionswert 0 gefunden, jedoch ist damit wie bereits erwähnt nur ein Teilziel erreicht: Es gilt nun herauszufinden, ob sich die Schalthäufigkeiten von  $t_A$  und  $t_B$  noch auf 1 reduzieren lassen, und dies geschieht durch das gewohnte Hinzufügen weiterer Schichten und Lösen eines linearen Optimierungsproblems. Eine T-Invariante  $k \cdot \vec{r}$ ,  $k > 1$ , die in einem Netz  $N_{AB}|_m^{cf}$  mit  $m$  Schichten gefunden wird, beschreibt zunächst den folgenden Sachverhalt:

- (1) Hier feuern die Transitionen  $t'_i$  und  $t''_i$  nicht mehr. Es genügen also  $m$  Schichten, um die leere Markierung zu reproduzieren. Sie reichen jedoch nicht aus, um auf allen Stellen derart genügend Marken zu erzeugen, dass jeder Kreis, der zur Reproduktion der leeren Markierung durchlaufen werden *muss*, bei nur einmaligem Schalten von  $t_A$  auch wirklich ausreichend oft durchlaufen werden *kann*. In  $k \cdot \vec{r}$  können die Marken aber durch das  $k$ -malige Schalten von  $t_A$  bereits auf der obersten Schicht  $m$  geliehen werden und Kreise werden früher und öfter durchlaufen. Transition  $t_B$  tilgt diese überschüssigen Marken durch mehrfaches Schalten.
- (2) Liegt starke Erreichbarkeit vor, so können die von  $t_A$  zusätzlich geliehenen Marken nur genau dann vom Netz selbst erzeugt werden, wenn ein oder mehr Kreise im Netz häufiger durchlaufen werden als es bei den momentan  $m$  Schichten möglich ist. Darum müssen weiter Schichten hinzugefügt und Lösungen  $\vec{r}(t_A) = \vec{r}(t_B) < k$  gesucht werden, bis eine T-Invariante mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  errechnet wurde.
- (3) Bei nur schwacher Erreichbarkeit ändert das Hinzufügen weiterer Schichten ab einer bestimmten Schichtenanzahl nichts mehr am Zielfunktionswert: Es wird eine Lösung  $k_{\min} \cdot \vec{r}$  ermittelt und  $k_{\min} > 1$  ist nicht weiter reduzierbar.

Aus den Punkten (2) und (3) ergibt sich unmittelbar die Frage nach einem *passenden Abbruchkriterium* für den Algorithmus.

**Abbruchkriterium bei der Untersuchung auf starke Erreichbarkeit**

Die Frage nach dem Abbruchkriterium ist gleichzusetzen mit der Frage, ab welcher Anzahl von Schichten entschieden werden kann, ob

- (a) eine Lösung  $k_{\min} \cdot \vec{r}$  mit  $k_{\min} > 1$  gefunden ist, die sich nicht mehr auf  $k = 1$  reduzieren lässt, oder
- (b) weitere Schichten hinzugefügt werden müssen, da eine Lösung  $k = 1$  noch gefunden werden kann.

Hierzu wollen wir das GLOP zur Berechnung der Lösung durch eine veränderte Zielfunktion verfeinern: Wie beim Nachweis der schwachen Erreichbarkeit gilt es eine T-Invariante zu finden, in der die Hilfstransitionen  $t'_i, t''_i$  nicht mehr feuern. Zusätzlich muss immer eine Lösung  $\vec{r}$  berechnet werden, in der die Schalthäufigkeiten der Transitionen  $t_A$  und  $t_B$  minimal sind. Existiert also beispielsweise eine Lösung mit  $\vec{r}(t_A) = \vec{r}(t_B) = k$ , so soll der Algorithmus, der das GLOP löst, keine T-Invariante mit  $\vec{r}(t_A) = \vec{r}(t_B) > k$  ausgeben. Nur so kann sichergestellt werden, dass man beim Hinzufügen weiterer Schichten die Änderung des Zielfunktionswertes nachverfolgen kann. Aufgrund dieser neuen Bedingung muss die Zielfunktion folgendermaßen erweitert werden:

$$\left( \sum_i h \cdot (\vec{r}(t'_i) + \vec{r}(t''_i)) \right) + \vec{r}(t_A) + \vec{r}(t_B) \longrightarrow \text{Min}, \quad h \in \mathbb{N}$$

Hierbei ist  $h$  immer eine sehr große natürliche Zahl, die aus folgendem Grund benötigt wird: Durch diese stärkere Gewichtung der  $t'_i$  und  $t''_i$  in der Zielfunktion im Vergleich zu  $t_A$  und  $t_B$  wird der Algorithmus zunächst versuchen, diese Werte zu minimieren — was dem Nicht-Schalten der Hilfstransitionen entspricht. Anschließend bleibt als Zielfunktion nur noch  $\vec{r}(t_A) + \vec{r}(t_B) \longrightarrow \text{Min}$  übrig, so dass auch für diese zwei Transitionen die geringstmöglichen Schalthäufigkeiten berechnet werden. Durch die Konstante  $h$  muss sichergestellt sein, dass die tatsächliche Schalthäufigkeit von  $t_A$  oder  $t_B$  den Wert  $h \cdot \vec{r}(t'_i) = h \cdot \vec{r}(t''_i)$  für keine Hilfstransition übersteigt. Darum muss für  $h$ , *abhängig vom untersuchten S/T-Netz*, eine ausreichend große natürliche Zahl gewählt werden. Für alle Beispiele dieser Arbeit hat sich der konstante Wert  $h = 100$  als ausreichend erwiesen.

Nun verläuft die Berechnung von Lösungsvektoren anders: Es wird nicht mehr nach *irgendeiner* Lösung gesucht, in der die Hilfstransitionen nicht mehr schalten, sondern nach einer Lösung, in der auch  $t_A$  und  $t_B$  so selten wie möglich — im idealen Fall genau einmal — feuern. Mit jeder zusätzlichen Schicht wird also ein für diese Schicht optimales Ergebnis berechnet. Das Resultat ist abhängig von den Kreisen im Netz.

Die minimalen Kreise werden mit jeder zusätzlichen Schicht neu aufgetrennt. Auf jeder Schicht wird ein Kreis begonnen und in der Regel auf der darunter liegenden Schicht beendet. Demnach stellt jede weitere Ebene einen Gewinn beim Vollenden von Kreisen dar, was bereits in Punkt (1) auf Seite 109 erläutert wurde: Kreise können früher durchlaufen werden und enden nicht auf Schicht 0, wo Transitionen  $t'_i, t''_i$  aushelfen müssten. In manchen Fällen — abhängig vom zugrunde liegenden Netz — können durch mehrmaliges Schalten von  $t_A$  so viele Marken geliehen werden, dass manche Kreise häufiger und schneller durchlaufen werden können als es beim einmaligen Schalten von  $t_A$  der Fall wäre. So kann eine schwache Erreichbarkeit nachweisende Lösung  $k \cdot \vec{r}$  bereits bei weniger zusätzlichen Schichten gefunden werden als es für starke Erreichbarkeit möglich wäre.

**Beispiel 7.4 (Durchlaufen von Kreisen und Leihen von Marken)** Abbildung 7.6 zeigt das S/T-Netz  $N6_{AB}$  von Seite 48, in dem starke Erreichbarkeit nachgewiesen werden soll. Die starke Erreichbarkeit nachweisende Schaltfolge einer T-Invariante  $\vec{r}$  ist in diesem Fall:

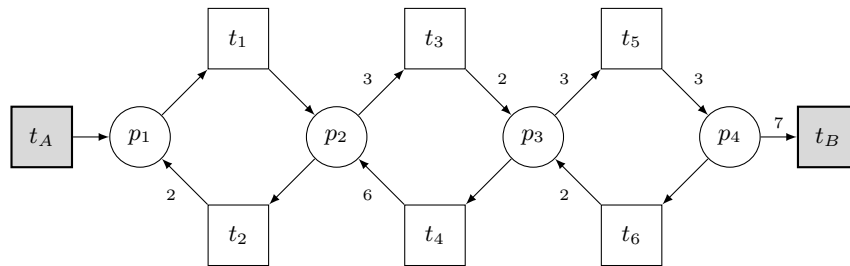
$$t_A t_1 t_2 t_1 t_2 (3 \cdot t_1) t_3 t_4 (2 \cdot t_3) t_5 (2 \cdot t_6) (2 \cdot t_5) t_B$$

$N6_{AB}$  besitzt drei minimale Kreise, die durch die Transitionen  $(t_1, t_2)$ ,  $(t_3, t_4)$  und  $(t_5, t_6)$  verlaufen. Transition  $t_3$  kann nur schalten, wenn  $t_1$  auf Stelle  $p_2$  drei Marken produziert hat. Dafür benötigt  $t_1$  auf  $p_1$  ebenfalls drei Marken, die durch das *zweimalige* Durchlaufen des Kreises  $(t_1, t_2)$  von  $t_2$  erzeugt werden.

In Abbildung 7.7 auf Seite 113 ist das um vier zusätzliche Schichten entfaltete Netz  $N6_{AB}|_4^{\text{cf}}$  dargestellt. Damit  $t_2$  zweimal schalten kann und der durchlaufene Kreis vollständig auf die Schichten entfaltet wurde (d.h. keine der Hilfstransitionen  $t'_{2,1}, t''_{2,1}$  muss feuern), sind in jedem Fall zwei zusätzliche Ebenen notwendig. Jedoch gelangen bei nur einmaligem Schalten von  $t_A$  in  $N6_{AB}|_2^{\text{cf}}$  die erzeugten Marken dann nicht mehr auf  $p_4$ , ohne dass andere Hilfstransitionen auf Schicht 0 schalten müssen.

Für das GLOP wird aber bereits in  $N6_{AB}|_2^{\text{cf}}$  eine Lösung<sup>1</sup>  $3 \cdot \vec{r}$  gefunden. Die drei Marken, die  $t_{1|2}$  auf  $p_{1|2}$  benötigt, werden also von  $t_A$  erzeugt. Sie werden so schneller zu  $p_{4|0}$  propagiert, alle überschüssigen Marken durch häufigeres Schalten der anderen Transitionen und letztlich durch das dreimalige Feuern von  $t_B$  getilgt. Die exakte Schaltfolge für  $N6_{AB}|_2^{\text{cf}}$  ist in Tabelle 7.1 zu finden.  $\square$

<sup>1</sup> Abbildung 7.7 kann das Netz  $N6_{AB}|_2^{\text{cf}}$  repräsentieren, indem man die Schichten 3 und 4 und zugehörige Hilfstransitionen  $c_{i|4,3}$  und  $c_{i|3,2}$  ausblendet und  $t_A$  mit  $p_{1|2}$  verbindet.

Abbildung 7.6: S/T-Netz  $N6_{AB}$ 

Da, wie bereits erwähnt, jede Schicht einen „Gewinn“ für ein entfaltetes Netz darstellt, kann auf jeder Schicht auch mindestens eine Kopie einer Transition des Originalnetzes schalten: Schwache Erreichbarkeit liegt vor und  $t_A$  ist stets auf der obersten Schicht aktiviert. Kann eine Transition in einer Schicht nicht schalten, so fehlen ihr noch Marken, die sie entweder von Transitionen aus der gleichen oder einer darüber liegenden Schicht erhält. Kommt eine Schicht hinzu, werden mehr Marken „auf natürliche Weise“, das heißt durch normale Schaltvorgänge, erzeugt und müssen nicht durch mehrmaliges Schalten von  $t_A$  geliehen werden. Ist wie in Netz  $N6_{AB}|_2^{cf}$  bereits eine schwache Erreichbarkeit nachweisende T-Invariante  $k \cdot \vec{r}$  gefunden, kann eine weitere zusätzliche Schicht zu folgenden verschiedenen Ergebnissen führen:

- (a) Der Zielfunktionswert sinkt nicht  $\Rightarrow$  Die zusätzliche Schicht ist nutzlos.
- (b) Der Zielfunktionswert sinkt  $\Rightarrow$  Mehr Marken wurden ohne Hilfe von  $t_A$  erzeugt.

Tatsächlich sinkt ab  $N6_{AB}|_2^{cf}$  der Zielfunktionswert mit jeder weiteren Schicht streng monoton: in  $N6_{AB}|_3^{cf}$  beträgt er 4 ( $\vec{r}(t_A) = \vec{r}(t_B) = 2$ ) und in  $N6_{AB}|_4^{cf}$  ist eine starke Erreichbarkeit nachweisende T-Invariante gefunden (Zielfunktionswert = 2). Die Ergebnisse für die Netze  $N6_{AB}|_0^{cf}$  bis  $N6_{AB}|_4^{cf}$  sind mit zugehörigen Zielfunktionswerten und Schalthäufigkeiten in Tabelle 7.1 abgebildet. Es liegt also die Vermutung nahe, dass der streng monoton fallende Zielfunktionswert ein vielversprechendes Abbruchkriterium für den Test auf starke Erreichbarkeit liefert: Falls der Wert durch Hinzufügen der nächsten Schicht nicht sinkt, können also keine weiteren Marken *ohne* Zuhilfenahme mehrmaligen Schaltens von  $t_A$  bereitgestellt werden, obwohl eine neue Schicht einen weiteren Durchlauf der Kreise im Netz erlaubt. Auch alle im folgenden Kapitel 8 aufgeführten Anwendungsbeispiele bekräftigen die Theorie der strengen Monotonie: In den Fällen, in denen schwache *vor* (das heißt durch weniger Schichten) starker Erreichbarkeit nachgewiesen wurde, sinkt der Zielfunktionswert mit jeder zusätzlichen Schicht. Falls starke und schwache Erreichbarkeit gleichzeitig durch eine T-Invariante  $\vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  bewiesen werden, liegt selbstverständlich keine strenge Monotonie



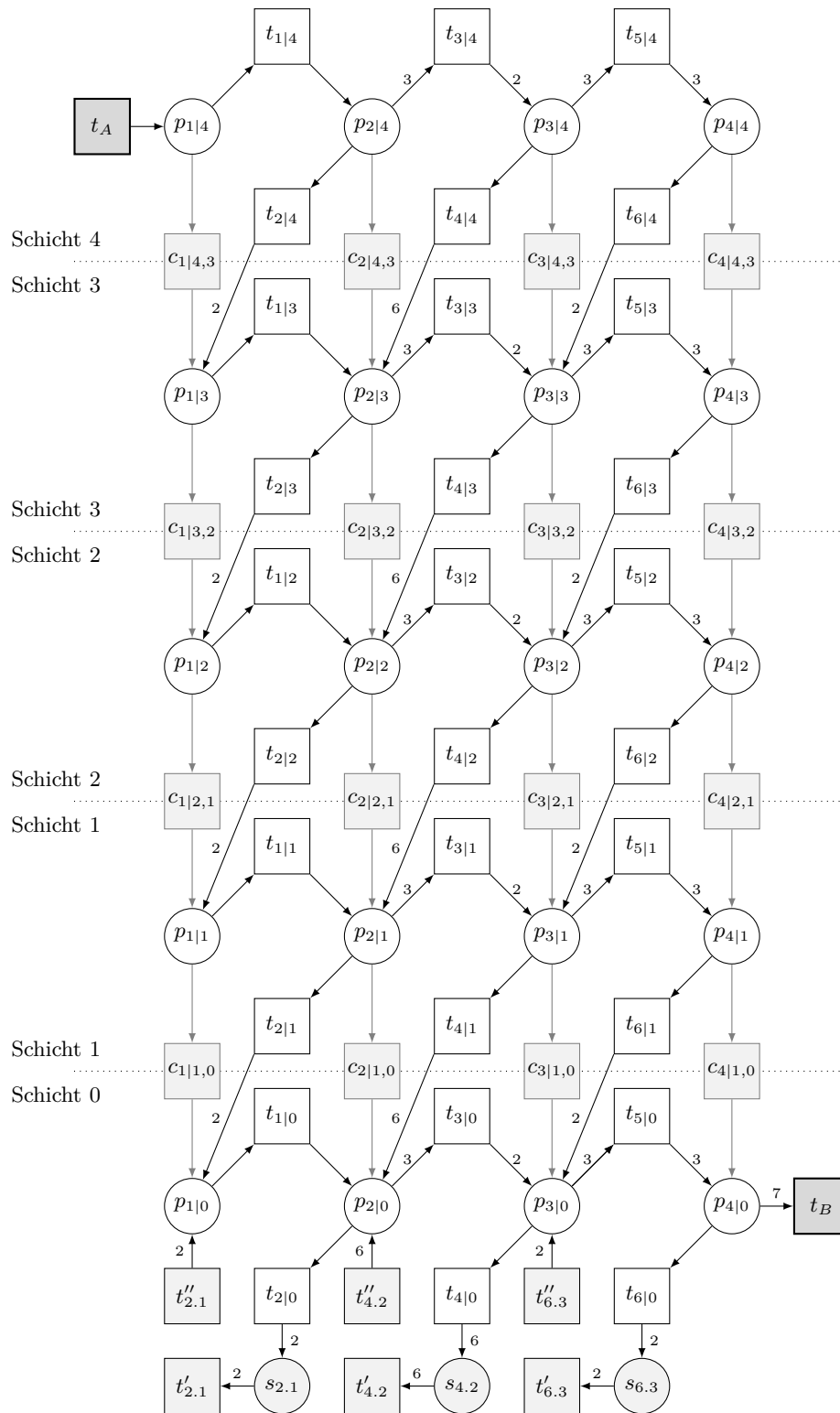


Abbildung 7.7: S/T-Netz  $N_{6|AB|4}^{cf}$

vor, da einen Schritt zuvor noch Hilfstransitionen  $t'_i, t''_i$  schalten mussten. Insgesamt lassen sich die Anwendungsbeispiele aus dieser Arbeit folgendermaßen gliedern:

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$N6_{AB 0}^{cf}$	1002	$t_A, t_{1 0}, t_{2 0}, t'_{2.1}, t''_{2.1}, 2 \cdot t_{1 0}, t_{2 0}, t'_{2.1}, t''_{2.1}, 2 \cdot t_{1 0}, t_{3 0}, t_{4 0}, t'_{4.2}, t''_{4.2}, 2 \cdot t_{3 0}, t_{5 0}, 2 \cdot t_{6 0}, 2 \cdot t'_{6.3}, 2 \cdot t''_{6.3}, 2 \cdot t_{5 0}, t_B$
$N6_{AB 1}^{cf}$	802	$t_A, t_{1 1}, t_{2 1}, 2 \cdot t_{1 0}, t_{2 0}, t'_{2.1}, t''_{2.1}, 2 \cdot t_{1 0}, t_{3 0}, t_{4 0}, t'_{4.2}, t''_{4.2}, 2 \cdot t_{3 0}, t_{5 0}, 2 \cdot t_{6 0}, 2 \cdot t'_{6.3}, 2 \cdot t''_{6.3}, 2 \cdot t_{5 0}, t_B$
$N6_{AB 2}^{cf}$	6	$3 \cdot t_A, 3 \cdot t_{1 2}, t_{3 2}, 2 \cdot t_{4 2}, 6 \cdot t_{2 1}, 2 \cdot t_{3 1}, 3 \cdot t_{4 1}, c_{3 1,0}, 12 \cdot t_{1 0}, 10 \cdot t_{3 0}, 7 \cdot t_{5 0}, 3 \cdot t_B$
$N6_{AB 3}^{cf}$	4	$2 \cdot t_A, 2 \cdot t_{1 3}, 2 \cdot t_{2 3}, 4 \cdot t_{1 2}, t_{2 2}, t_{3 2}, 2 \cdot t_{4 2}, 4 \cdot t_{3 1}, t_{4 1}, 2 \cdot t_{5 1}, t_{1 1}, t_{2 1}, t_{6 1}, c_{1 1,0}, c_{3 1,0}, 5 \cdot c_{4 1,0}, 3 \cdot t_{1 0}, 3 \cdot t_{3 0}, 3 \cdot t_{5 0}, 2 \cdot t_B$
$N6_{AB 4}^{cf}$	2	$t_A, t_{1 4}, t_{2 4}, t_{1 3}, t_{2 3}, c_{1 3,2}, 3 \cdot t_{1 2}, t_{3 2}, t_{4 2}, c_{3 2,1}, 2 \cdot t_{3 1}, t_{5 1}, 2 \cdot t_{6 1}, 2 \cdot c_{3 1,0}, c_{4 1,0}, 2 \cdot t_{5 0}, t_B$

**Tabelle 7.1:** Lösungen des GLOP für das schichtweise erweiterte Netz  $N6_{AB}^{cf}$

1. Für das GLOP wird eine Lösung gefunden, in der die Hilfstransitionen  $t'_i, t''_i$  nicht schalten, d.h. direkt ein Vektor  $\vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$ . Die Lösung ist kreisfrei; damit kann die Zustandsgleichung als hinreichendes Kriterium für Erreichbarkeit herangezogen werden und starke Erreichbarkeit ist nachgewiesen.
2. Für das GLOP wird eine T-Invariante  $k \cdot \vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) > 1$  gefunden, in der die Hilfstransitionen  $t'_i, t''_i$  nicht schalten. Damit ist schwache Erreichbarkeit nachgewiesen, auf starke Erreichbarkeit muss noch getestet werden und weitere Schichten kommen hinzu:
  - Sinkt der Zielfunktionswert nun bei zusätzlichen Schichten streng monoton und bleibt ab einem bestimmten Wert  $\vec{r}(t_A) = \vec{r}(t_B) = k_{min} > 1$  bei neuen Schichten konstant, so haben weitere Schichten offenbar keinen Nutzen und es liegt keine starke Erreichbarkeit vor.
  - Fällt der Zielfunktionswert nun bei zusätzlichen Schichten streng monoton und erreicht den Wert  $\vec{r}(t_A) + \vec{r}(t_B) = 2$ , so weist die zugehörige Lösung  $\vec{r}$  starke Erreichbarkeit nach.

**Algorithmus 7.3:** NACHWEIS STARKER ERREICHBARKEIT FÜR DIE NETZE AUS DIESER ARBEIT (KREISFREI AUF SCHICHT 0)

**Eingabe:**  $N, N_{AB|_0}^{cf} = (S, T, F, W), K, h.$

**Ausgabe:** "stark erreichbar" oder "schwach erreichbar" mit nachweisender T–Invariante  $\vec{r}$  und zugehörigem Netz  $N_{AB|_m}^{cf}$

```

1 ZF :=  $\vec{r}(t_A) + \vec{r}(t_B)$ ; Initialisierung.
2 cons :=  $\{\vec{r}(t_A) = \vec{r}(t_B) \geq 1\}$ ;
3 for  $i := 1$  to  $|T|$  do ► Zielfunktion und Constraints generieren.
4   if (Transition  $i$  ist vom Typ  $t'_i$ ) then
5     ZF :=  $ZF + h \cdot \vec{r}(t'_i) + h \cdot \vec{r}(t''_i)$ ;
6     cons :=  $\text{cons} \cup \{\vec{r}(t'_i) = \vec{r}(t''_i)\}$ ;
7   endif
8 endfor
9  $N_{AB|_n}^{cf} := N_{AB|_0}^{cf}$ ; ►  $n := 0$ 
10  $(\vec{r}, \text{neuer\_zfwert}) := \text{GLOP}(N_{AB|_n}^{cf}, ZF \rightarrow \text{Min}, \text{cons})$ ; ► GLOP für Lage 0 lösen.
11 while  $(\vec{r}(t'_i) = \vec{r}(t''_i) > 0$  für irgendein Transitionspar  $t'_i, t''_i$ ) do
12    $N_{AB|_{n+1}}^{cf} := \text{ADD-LAYER}(N, N_{AB|_n}^{cf}, K)$ ; ► ...weitere Schicht (mit Alg. 7.1) hinzufügen...
13    $(\vec{r}, \text{neuer\_zfwert}) := \text{GLOP}(N_{AB|_{n+1}}^{cf}, ZF \rightarrow \text{Min}, \text{cons})$ ;
14    $n := n + 1$ ; ► Schichtenanzahl hochzählen.
15 endwhile
16 if (neuer_zfwert = 2) then ► Starke Erreichbarkeit wurde nachgewiesen.
17   return ("stark erreichbar",  $N_{AB|_n}^{cf}, \vec{r}$ );
18 endif
19 ► Bisher nur schwache Erreichbarkeit mit neuer_zfwert > 2.
20 alter_zfwert := 0; ► Von nun an ZF-Werte vergleichen.
21 while (alter_zfwert  $\neq$  neuer_zfwert) do ► Solange Zielfunktionswert streng monoton fällt...
22   alter_zfwert := neuer_zfwert;
23    $N_{AB|_{n+1}}^{cf} := \text{ADD-LAYER}(N, N_{AB|_n}^{cf}, K)$ ; ► ...weitere Schicht (mit Alg. 7.1) hinzufügen...
24    $(\vec{r}, \text{neuer\_zfwert}) := \text{GLOP}(N_{AB|_{n+1}}^{cf}, ZF \rightarrow \text{Min}, \text{cons})$ ; ► ...und jeweils das GLOP lösen.
25   if (neuer_zfwert = 2) then
26     return ("stark erreichbar",  $N_{AB|_{n+1}}^{cf}, \vec{r}$ );
27   endif
28    $n := n + 1$ ; ► Schichtenanzahl hochzählen.
29 endwhile
30 return ("schwach erreichbar",  $N_{AB|_n}^{cf}, \vec{r}$ );

```

Ob die strenge Monotonie tatsächlich ein hinreichendes Abbruchkriterium für dieses Verfahren ist, konnte zu diesem Zeitpunkt noch nicht bewiesen werden und stellt einen zukünftigen Forschungsgegenstand dar. Algorithmus 7.3 entscheidet daher das Erreichbarkeitsproblem für die S/T–Netze aus dieser Arbeit unter dem Vorbehalt, dass seine

Korrektheit für beliebige S/T-Netze noch untersucht werden muss. Insgesamt sieben Anwendungsbeispiele für diesen Algorithmus sind in Kapitel 8 zu finden.

### 7.3 Entfaltung mit Erhaltung der Kreise auf der untersten Schicht

Bei dieser Variante des Lösungsansatzes sollen die Kreise auf der untersten Schicht erhalten und das Netz sonst wie im Abschnitt zuvor entfaltet werden. Auf Schicht 0 wird also das Originalnetz  $N = (S, T, F, W)$  zunächst übernommen und um  $t_A$  und  $t_B$  ergänzt.

**Definition 7.2 (Schichtennetz)** Sei  $N_{AB} = (S, T, F, W)$  ein S/T-Netz, in dem schwache Erreichbarkeit gegeben ist. Ein Schichtennetz  $N_{AB}|_n = (S^*, T^*, F^*, W^*)$  erhält man aus  $N$  und der Liste  $K$  der aufzuschneidenden Kanten, indem man gemäß Algorithmus 7.1 nacheinander  $n$  Schichten zu  $N_{AB}$  hinzufügt. Alle auf Schicht 0 existierenden Kreise bleiben erhalten. ■

Um sicherzustellen, dass auch in diesem Fall eine T-Invariante berechnet wird, deren Netzdarstellung kreisfrei ist, dürfen die Originaltransitionen auf der untersten Schicht — außer  $t_B$  — nicht mehr schalten. Nebenbedingungen und Zielfunktion für das ganzzahlige lineare Optimierungsproblem müssen daher anders formuliert werden.

#### 7.3.1 Nebenbedingungen des ganzzahligen linearen Programms

Auch hier kann eine realisierbare T-Invariante  $\vec{r}$  nur starke Erreichbarkeit nachweisen, wenn  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  gilt. Durch die Erhaltung der Kreise auf Schicht 0 kann es beim Lösen des GLOP stets passieren, dass Transitionen dieser Schicht noch feuern und damit Kreise durchlaufen werden. In der Regel ist dann eine solche T-Invariante nicht realisierbar. Ziel muss es also sein, eine Lösung zu finden, in der alle Originaltransitionen  $t_1, \dots, t_n$  des Originalnetzes  $N$  auf Schicht 0 nicht schalten. Alle notwendigen Bedingungen sind damit:

- $\vec{r}(t_A) = \vec{r}(t_B) \geq 1$   
 $t_A$  und  $t_B$  müssen gleich oft schalten, jedoch mindestens einmal. Im Fall starker Erreichbarkeit wird eine Lösung gefunden mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$ . Bei nur schwacher Erreichbarkeit existiert eine Lösung  $k \cdot \vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = k > 1$ .

- $\sum_{i=1}^{|T|} (\vec{r}(t_{i|0})) \rightarrow \text{Min}$

Eine mindestens schwache, im besten Fall starke Erreichbarkeit nachweisende Lösung ist genau dann gefunden, wenn keine der  $n$  Originaltransitionen des Netzes  $N$  auf Schicht 0 in der T-Invariante  $\vec{r}$  schaltet. Als Zielfunktion des GLOP gilt es also die Summe der Schalthäufigkeiten dieser Transitionen zu minimieren. Bei einem Zielfunktionswert von 0 erhält man das gewünschte Resultat.

### 7.3.2 Nachweis schwacher Erreichbarkeit

Da für das Netz  $N_{AB}$  schwache Erreichbarkeit gegeben ist, muss diese auch als Lösung eines GLOP nach einer bestimmten Anzahl von Schichten gefunden werden. Zunächst wird  $N_{AB}$  als Schicht 0 des zu entfaltenden Netzes  $N_{AB}|_0$  festgelegt und die Stellen  $p_i$  und Transitionen  $t_j$  in  $p_{i|0}$  und  $t_{j|0}$  umbenannt. Die Kreise auf dieser Schicht bleiben erhalten, so dass keine weiteren Umformungen erfolgen. Anschließend werden mit Algorithmus 7.1 von Seite 100 neue Schichten hinzugefügt. Nach jeder neuen Schicht  $n$  wird  $N_{AB}|_n$  mit dem einzigen Constraint  $\vec{r}(t_A) = \vec{r}(t_B) \geq 1$  und der zu minimierenden Zielfunktion als GLOP formuliert und nach einer Lösung gesucht. Dies wird so lange wiederholt, bis nach einer bestimmten Anzahl von  $m$  Schichten eine Lösung mit Zielfunktionswert 0 gefunden ist. Dann ist der Nachweis für schwache Erreichbarkeit erbracht.

**Beispiel 7.5 (Nachweis schwacher Erreichbarkeit)** Das Netz  $R2|_2$  in Abbildung 7.8 ist die um einfache Stellen reduzierte und um  $t_A$  und  $t_B$  ergänzte Variante des Netzes  $N2$  von Seite 14. Es wurden bereits zwei zusätzliche Schichten hinzugefügt. In diesem Netz ist nur schwache Erreichbarkeit gegeben. Das GLOP hat folgende Bestandteile (Ganzzahligkeit und Nichtnegativität sind nicht explizit aufgeführt):

$$\begin{aligned} \vec{r}(t_{4|0}) &\rightarrow \text{Min} \\ \vec{r}(t_A) = \vec{r}(t_B) &\geq 1 \\ [R2|_2] \cdot \vec{r} &= \vec{0} \end{aligned}$$

Die Ergebnisse des GLOP, die von Algorithmus 7.4 auf Seite 118 nacheinander für die Netze  $R2|_0$  bis  $R2|_2$  berechnet werden, sind in Tabelle 7.2 zu finden. Bereits mit einer zusätzlichen Schicht ist eine realisierbare Schaltfolge gefunden, die schwache Erreichbarkeit nachweisen würde, wenn das Schalten von  $t_{4|0}$  zulässig wäre. Der Zielfunktionswert ist hier jedoch noch  $> 0$  und Algorithmus 7.4 bricht noch nicht ab.

Erst mit der zweiten zusätzlichen Schicht wird eine T-Invariante  $\vec{r}$  gefunden, für die  $\text{zfwert} = 0$  ist. In der Tat sind aber die für  $R2|_1$  und  $R2|_2$  ermittelten T-Invarianten identische Schaltsequenzen im Originalnetz R2: Beide entsprechen der T-Invariante  $\vec{r}_{orig} = (t_A \ t_A \ t_B)^T = (2 \ 2 \ 2)^T$ .

Damit der Nachweis schwacher Erreichbarkeit *erst* ab dem Nicht-Schalten aller Originaltransitionen erfolgt, sollten wie zuvor auch  $t_A$  und  $t_B$  in die Zielfunktion übernommen und die Originaltransitionen auf Schicht 0 mit einer geeigneten Konstante  $h$  gewichtet werden. Die Ergebnisse für dieses Netz auf Basis dieser neuen Zielfunktion sind in Kapitel 8.2.1 zu finden.  $\square$

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$R2 _0$	1	$t_A, t_{4 0}, t_B$ (nicht realisierbar)
$R2 _1$	1	$2 \cdot t_A, t_{4 1}, c_{2 1,0}, t_{4 0}, 2 \cdot t_B$
$R2 _2$	0	$2 \cdot t_A, t_{4 2}, c_{2 2,1}, t_{4 1}, 2 \cdot c_{2 1,0}, 2 \cdot t_B$

**Tabelle 7.2:** Lösungen des GLOP für das schichtweise erweiterte Netz R2 mit Kreisen auf Schicht 0

#### Algorithmus 7.4: NACHWEIS SCHWACHER ERREICHBARKEIT MIT KREISEN AUF SCHICHT 0

**Eingabe:**  $N = (S, T, F, W)$ ,  $N_{AB}|_0$ ,  $K$ .

**Ausgabe:** T-Invariante  $\vec{r}$ , die schwache Erreichbarkeit nachweist, und zugehöriges Netz  $N_{AB}|_m$

```

1 ZF := 0; ▶ Initialisierung.
2 cons := { $\vec{r}(t_A) = \vec{r}(t_B) \geq 1$ };
3 for  $i := 1$  to  $|T|$  do ▶ Zielfunktion generieren.
4     ZF := ZF +  $t_{i|0}$ ;
5 endfor
6  $N_{AB}|_n := N_{AB}|_0$ ; ▶  $n := 0$ 
7  $(\vec{r}, \text{zfwert}) := \text{GLOP}(N_{AB}|_n, \text{ZF} \rightarrow \text{Min}, \text{cons})$ ; ▶ GLOP für Lage 0 lösen.
8 while ( $\text{zfwert} \neq 0$ ) do ▶ Solange  $\text{zfwert} \neq 0$ ...
9      $N_{AB}|_{n+1} := \text{ADD-LAYER}(N, N_{AB}|_n, K)$ ; ▶ ...weitere Schicht (mit Alg. 7.1) hinzufügen...
10     $(\vec{r}, \text{zfwert}) := \text{GLOP}(N_{AB}|_{n+1}, \text{ZF} \rightarrow \text{Min}, \text{cons})$ ; ▶ ...und jeweils das GLOP lösen.
11     $n := n + 1$ ; ▶ Schichtenanzahl hochzählen.
12 endwhile

```

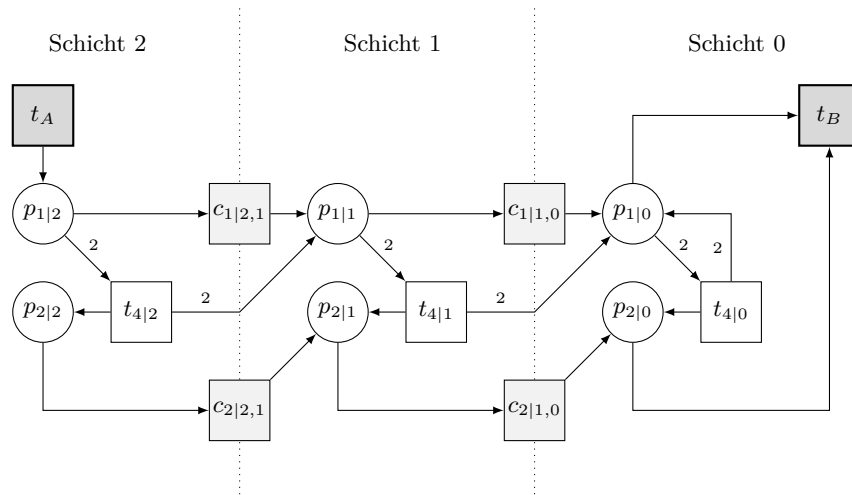


Abbildung 7.8: Netz  $R2|_2$

### 7.3.3 Nachweis starker Erreichbarkeit

Das Verhalten von Algorithmus 7.4 macht deutlich, dass die Originaltransitionen auf Schicht 0 wieder durch eine Konstante  $h$  zusätzlich gewichtet und  $t_A$  und  $t_B$  in die Zielfunktion übernommen werden müssen. Für den Nachweis starker Erreichbarkeit gelten die gleichen Regeln wie in Abschnitt 7.2.3. Eine T-Invariante  $k \cdot \vec{r}$ ,  $k > 1$ , die in einem Netz  $N_{AB}|_m$  mit  $m$  Schichten gefunden wird, weist zunächst nur schwache Erreichbarkeit nach:

- (1) Alle Transitionen  $t_{i|0}$  der untersten Schicht feuern nicht mehr. Die  $m$  Schichten sind ausreichend, um die leere Markierung zu reproduzieren. Sie genügen nicht, um auf allen Stellen derart genügend Marken zu erzeugen, dass jeder Kreis, der zur Reproduktion der leeren Markierung durchlaufen werden muss, bei nur einmaligem Schalten von  $t_A$  auch wirklich ausreichend oft durchlaufen werden kann. Darum feuern  $t_A$  und  $t_B$  insgesamt  $k$ -mal.
- (2) Liegt starke Erreichbarkeit vor, so müsste von nun an der Zielfunktionswert streng monoton fallen, bis eine T-Invariante mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  errechnet wurde.
- (3) Bei nur schwacher Erreichbarkeit kann der Zielfunktionswert mit weiteren Schichten zunächst sinken, bleibt aber ab einer bestimmten Schichtenanzahl konstant: Es wird eine Lösung  $k_{\min} \cdot \vec{r}$  ermittelt und  $k_{\min} > 1$  ist nicht weiter reduzierbar.

Das Abbruchkriterium für Algorithmus 7.5 auf Seite 121, der auf starke Erreichbarkeit für Netze mit Kreisen auf Schicht 0 prüft, entspricht für die Netze in dieser Arbeit dem Abbruchkriterium von Algorithmus 7.3 von Seite 115, der die Erreichbarkeit für kreisfreie Schichtenetze entscheidet. Die beiden Algorithmen unterscheiden sich durch die Zielfunktion und Nebenbedingungen:  $t_A$  und  $t_B$  werden nun in die Zielfunktion übernommen:

$$\left( \sum_{i=1}^{|T|} (h \cdot \vec{r}(t_{i|0})) \right) + \vec{r}(t_A) + \vec{r}(t_B) \longrightarrow \text{Min}, \quad h \in \mathbb{N}$$

Hierbei ist  $h$  wieder eine sehr große natürliche Zahl, die den Originaltransitionen im Verhältnis zu  $t_A$  und  $t_B$  eine größere Gewichtung in der Zielfunktion verleiht, so dass bei der Lösung des GLOP zunächst versucht wird, die Werte der  $t_{i|0}$  zu minimieren. Erst wenn diese minimiert und damit schwache Erreichbarkeit nachgewiesen ist, werden auch die Schalthäufigkeiten von  $t_A$  und  $t_B$  so gering wie möglich gehalten. Bei schwacher Erreichbarkeit wird der Algorithmus also keine Lösung  $\vec{r}(t_A) = \vec{r}(t_B) = k$  liefern, wenn auch eine T-Invariante mit  $\vec{r}(t_A) = \vec{r}(t_B) < k$  existiert. Die einzige Nebenbedingung des Algorithmus ist  $\vec{r}(t_A) = \vec{r}(t_B) \geq 1$ .

Grundsätzlich verhält sich Algorithmus 7.5 nun wie Algorithmus 7.3 — mit dem einzigen Unterschied, dass noch Kreise im Netz vorhanden sind und, wie bereits im letzten Abschnitt erläutert, die berechneten T-Invarianten vor dem Nachweis schwacher Erreichbarkeit nicht realisierbar sein müssen.

Wir wollen nun zum Vergleich der beiden Algorithmen Netz  $\mathbf{N6}_{\text{AB}}$  von Seite 112 mit Erhaltung der Kreise auf Schicht 0 auf starke Erreichbarkeit untersuchen. Die Ergebnisse für die Schichten  $\mathbf{N6}_{\text{AB}}|_0$  bis  $\mathbf{N6}_{\text{AB}}|_5$  sind in Tabelle 7.3 zusammengestellt. Durch die noch vorhandenen Kreise benötigt Algorithmus 7.5 genau eine Schicht mehr um schwache und starke Erreichbarkeit nachzuweisen: Im kreisfreien Netz  $\mathbf{N6}_{\text{AB}}|_2^{\text{cf}}$  wird eine Lösung  $3 \cdot \vec{r}$  gefunden, jedoch schalten einige Originaltransitionen auf der untersten Schicht, die in  $\mathbf{N6}_{\text{AB}}|_2$  nicht feuern sollen und daher hier den Nachweis schwacher Erreichbarkeit verhindern. Erst mit der nächsten Schicht wird eine entsprechende T-Invariante gefunden.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$\mathbf{N6}_{\text{AB}} _0$	1602	$t_A, t_{1 0}, t_{2 0}, 2 \cdot t_{1 0}, t_{2 0}, 2 \cdot t_{1 0}, t_{3 0}, t_{4 0},$ $2 \cdot t_{3 0}, t_{5 0}, 2 \cdot t_{6 0}, 2 \cdot t_{5 0}, t_B$
$\mathbf{N6}_{\text{AB}} _1$	1402	$t_A, t_{1 1}, t_{2 1}, 2 \cdot t_{1 0}, t_{2 0}, 2 \cdot t_{1 0}, t_{3 0}, t_{4 0},$ $2 \cdot t_{3 0}, t_{5 0}, 2 \cdot t_{6 0}, 2 \cdot t_{5 0}, t_B$



Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$N_{6AB} _2$	1102	$t_A, t_{1 2}, t_{2 2}, 2 \cdot t_{1 1}, t_{2 1}, c_{2 1,0}, 2 \cdot t_{1 0}, t_{3 0}, t_{4 0},$ $2 \cdot t_{3 0}, t_{5 0}, 2 \cdot t_{6 0}, 2 \cdot t_{5 0}, t_B$
$N_{6AB} _3$	6	$3 \cdot t_A, 3 \cdot t_{1 3}, t_{3 3}, 2 \cdot t_{4 3}, 6 \cdot t_{2 2}, 2 \cdot t_{3 2}, 3 \cdot t_{4 2}, c_{3 2,1},$ $12 \cdot t_{1 1}, 10 \cdot t_{3 1}, 7 \cdot t_{5 1}, 21 \cdot c_{4 1,0}, 3 \cdot t_B$
$N_{6AB} _4$	4	$2 \cdot t_A, 2 \cdot t_{1 4}, t_{2 4}, c_{2 4,3}, 2 \cdot t_{1 3}, t_{3 3}, 2 \cdot t_{4 3}, 3 \cdot t_{2 2},$ $3 \cdot t_{3 2}, t_{4 2}, t_{5 2}, t_{6 2}, 2 \cdot c_{3 2,1}, 2 \cdot c_{4 2,1}, 6 \cdot t_{1 1}, 4 \cdot t_{3 1},$ $4 \cdot t_{5 1}, 14 \cdot c_{4 1,0}, 2 \cdot t_B$
$N_{6AB} _5$	2	$t_A, t_{1 5}, t_{2 5}, t_{1 4}, t_{2 4}, c_{1 4,3}, 3 \cdot t_{1 3}, t_{3 3}, t_{4 3}, c_{3 3,2},$ $2 \cdot t_{3 2}, t_{5 2}, 2 \cdot t_{6 2}, 2 \cdot c_{3 2,1}, c_{4 2,1}, 2 \cdot t_{5 1}, 7 \cdot c_{4 1,0},$ $t_B$

 Tabelle 7.3: Lösungen des GLOP für das schichtweise erweiterte Netz  $N_{6AB}$ 

**Algorithmus 7.5:** NACHWEIS STARKER ERREICHBARKEIT FÜR DIE NETZE AUS DIESER ARBEIT (MIT KREISEN AUF SCHICHT 0)

**Eingabe:**  $N = (S, T, F, W), N_{AB}|_0, K, h$ .

**Ausgabe:** "stark erreichbar" oder "schwach erreichbar" mit nachweisender T-Invariante  $\vec{r}$  und zugehörigem Netz  $N_{AB}|_m$

```

1 ZF :=  $\vec{r}(t_A) + \vec{r}(t_B)$ ; ► Initialisierung.
2 cons :=  $\{\vec{r}(t_A) = \vec{r}(t_B) \geq 1\}$ ;
3 for  $i := 1$  to  $|T|$  do ► Zielfunktion generieren.
4   ZF := ZF +  $h \cdot \vec{r}(t_{i|0})$ ;
5 endfor
6  $N_{AB}|_n := N_{AB}|_0$ ; ►  $n := 0$ 
7  $(\vec{r}, \text{neuer\_zfwert}) := \text{GLOP}(N_{AB}|_n, \text{ZF} \rightarrow \text{Min}, \text{cons})$ ; ► GLOP für Lage 0 lösen.
8 while (mindestens für eine Originaltranstion  $t_{i|0}$  gilt  $\vec{r}(t_{i|0}) > 0$ ) do
9    $N_{AB}|_{n+1} := \text{ADD-LAYER}(N, N_{AB}|_n, K)$ ; ► ...weitere Schicht (mit Alg. 7.1) hinzufügen...
10   $(\vec{r}, \text{neuer\_zfwert}) := \text{GLOP}(N_{AB}|_{n+1}, \text{ZF} \rightarrow \text{Min}, \text{cons})$ ;
11   $n := n + 1$ ; ► Schichtenanzahl hochzählen.
12 endwhile
13 if (neuer_zfwert = 2) then
14   return ("stark erreichbar",  $N_{AB}|_n, \vec{r}$ );
15 endif
16 alter_zfwert := 0; ► Notwendige Initialisierung.
17 while (alter_zfwert  $\neq$  neuer_zfwert) do ► Solange Zielfunktionswert fällt...
18   alter_zfwert := neuer_zfwert;
    
```

```

19  $N_{AB}|_{n+1} := \text{ADD-LAYER}(N, N_{AB}|_n, K);$     ▶ ...weitere Schicht (mit Alg. 7.1) hinzufügen...
20  $(\vec{r}, \text{neuer\_zfwert}) := \text{GLOP}(N_{AB}|_{n+1}, \text{ZF} \rightarrow \text{Min}, \text{cons});$     ▶ ...und jeweils das GLOP lösen.
21 if (neuer_zfwert = 2) then
22     return ("stark erreichbar",  $N_{AB}|_{n+1}, \vec{r}$ );
23 endif
24  $n := n + 1;$     ▶ Schichtenanzahl hochzählen.
25 endwhile
26 return ("schwach erreichbar",  $N_{AB}|_n, \vec{r}$ );

```

## 7.4 Bewertung des Lösungsansatzes

Insgesamt kann dieses Verfahren als sehr elegant bezeichnet werden: Die Entfaltung der Netze in kreisfreie Schichten ist eine sehr anschauliche Methode, um das mehrmalige Durchlaufen der verschiedenen Zyklen im Netz und die daraus resultierende Markenproduktion und -konsumption nachzuverfolgen. Zudem kann die Zustandsgleichung als hinreichendes Kriterium für Erreichbarkeit herangezogen werden. Für die Performance und die optimale Zielfunktion ist ausschlaggebend, wo die Kreise aufgeschnitten werden und welche Konstante  $h$  in die Zielfunktion übernommen wird:

- Grundsätzlich ist es sinnvoll, einen minimalen Kreis nur einmal aufzuschneiden. Werden mehrere Kanten eines Kreises auf die darunter liegende Schicht umgeleitet, so kann zwar immer noch auf jeder Schicht mindestens eine Transition schalten — die erforderlichen Marken werden schließlich über die Hilfstransitionen  $c_i$  nach unten gereicht —, insgesamt sind aber möglicherweise mehr Schichten notwendig um schwache bzw. starke Erreichbarkeit nachzuweisen. Würde man beispielsweise die Kante  $(t_1, p_6)$  in Netz  $B3_{AB}$  in Abschnitt 8.1.3 aufschneiden, so könnte schwache bzw. starke Erreichbarkeit erst nach vier bzw. fünf Schichten (statt drei und vier) nachgewiesen werden. Genau aus diesem Grund ist es sinnvoll, nur diejenigen Kanten umzuleiten bzw. aufzuschneiden, durch die die meisten Kreise verlaufen. Dieses Kriterium wurde darum bereits in Algorithmus 5.4 von Seite 64 berücksichtigt.
- Die Wahl des Parameters  $h$  für die Zielfunktion ist von entscheidender Bedeutung für das korrekte Verhalten des Programms, welches das GLOP löst: Durch die Gewichtung sollen in der Lösung zuerst die Hilfstransitionen  $t'_i$  und  $t''_i$  (im kreisfreien Netz) bzw. alle Originaltransitionen  $t_{i|0}$  (im Netz mit Kreisen auf Schicht 0) nicht mehr auftreten und dann die Schalthäufigkeiten von  $t_A$  und  $t_B$  mini-

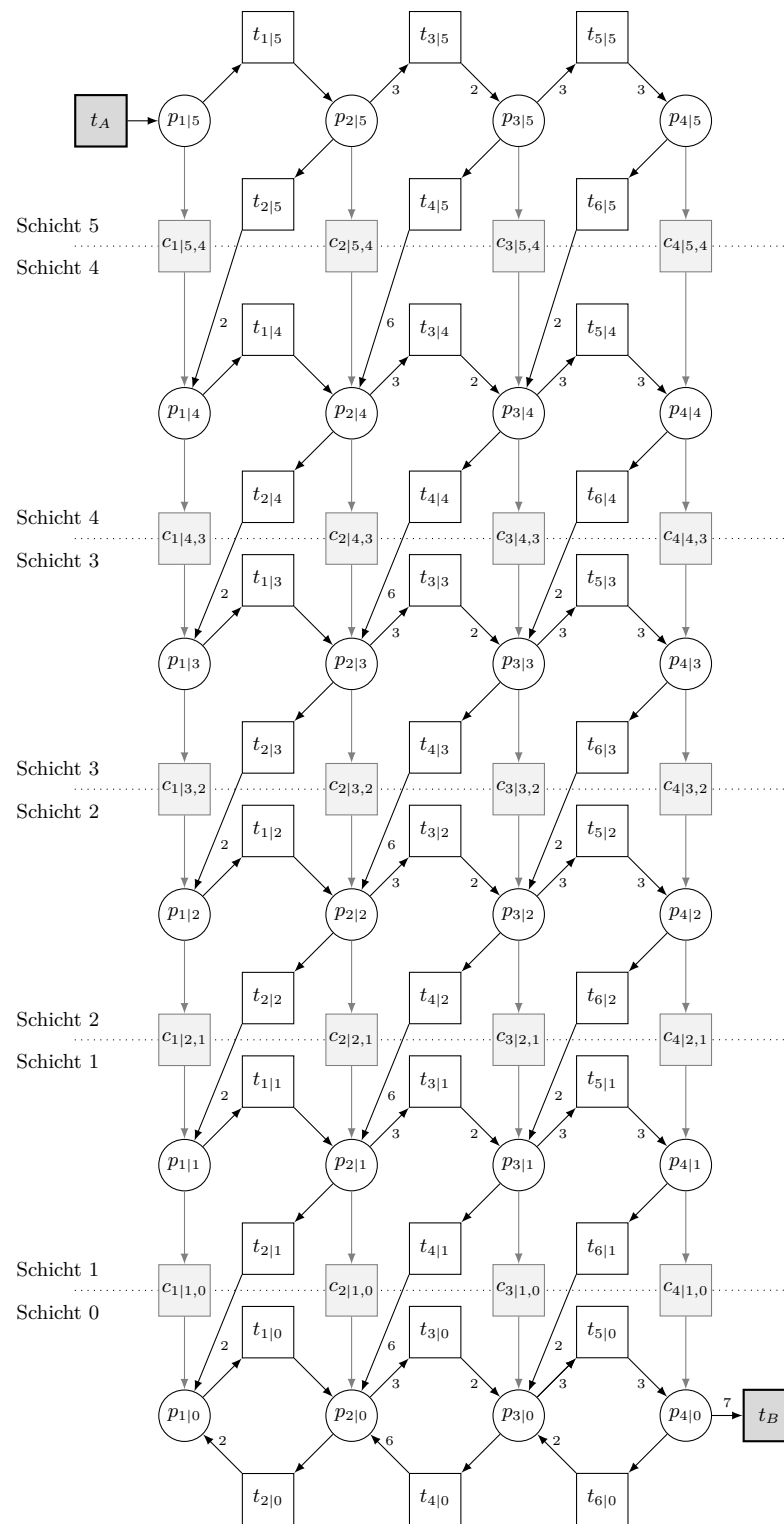


Abbildung 7.9: S/T-Netz  $N_{6AB|5}$

miert werden. Bei der Festlegung der Konstante  $h$  könnte man beispielsweise das Produkt aller Kantengewichte des Netzes verwenden.

Ob nun auf der untersten Schicht die Kreise aufgeschnitten oder erhalten bleiben sollten, steht weiterhin zur Diskussion. Der Vorteil eines vollständig kreisfreien Netzes ist jedoch, dass die Lösung des GLOP stets eine *realisierbare* T-Invariante ist, wohingegen beim nicht zyklischen Netz bis zum Nachweis der schwachen Erreichbarkeit auch nicht realisierbare Lösungsvektoren berechnet werden können. In vielen Fällen wird beim vollständig zyklischen Netz auch mit weniger Schichten starke Erreichbarkeit nachgewiesen (siehe z. B. Netze B2 und B4 in Kapitel 8).

Der Rechenaufwand bei diesem Verfahren kann sehr schnell sehr groß werden: Es werden viele zusätzliche Schichten benötigt, wenn ein Netz viele Kreise enthält, die häufig durchlaufen werden und sich gegenseitig mit Marken versorgen müssen. Wenn das Netz selbst schon sehr groß ist, liegt dem GLOP letztendlich ein Gleichungssystem zugrunde, das aus Hunderten, wenn nicht Tausenden von Transitionen besteht. Trotzdem kommt dieses Verfahren aufgrund der Art der Entfaltung in der Regel mit weniger Schichten aus als die in Kapitel 4.2 vorgestellte Methode von Ramachandran und Kamath: Während für das Netz N4 auf Seite 41 insgesamt fünf zusätzliche Schichten bis zum Nachweis starker Erreichbarkeit benötigt werden, kommt Lösungsansatz II bei diesem Netz mit einer (ohne Kreise auf Schicht 0) bzw. zwei (mit Kreisen auf Schicht 0) Ebenen aus (vgl. Abschnitt 8.1.2). Der Grund ist naheliegend: Bei Ramachandran/Kamath wird hinter jeder Transition aufgeschnitten, so dass pro Schicht zwar mindestens eine, insgesamt aber in der Regel weniger Transitionen schalten können.

Die Korrektheit des Lösungsansatzes für beliebige Stellen/Transitions-Netze, in denen schwache Erreichbarkeit gegeben ist, ist zum einen von der Korrektheit des Abbruchkriteriums, zum anderen von der richtigen Wahl der Konstante  $h$  und der Korrektheit des Algorithmus, der das GLOP löst, abhängig. Ein Algorithmus, der für ein gegebenes GLOP nicht die optimale Lösung findet, ist unbrauchbar, da gerade bei diesem Lösungsansatz die sehr spezifische Zielfunktion mit ihrem aussagekräftigen Zielfunktionswert das Abbruchkriterium darstellt. Für alle in dieser Arbeit diskutierten Beispiele haben das Programmpaket von Gensty [Gen05] und das frei verfügbare Programm *lp\_solve* [BEN11] korrekte Ergebnisse geliefert, wodurch das hier verwendete Abbruchkriterium überhaupt erst gefunden wurde. Anwendungsbeispiele für den Nachweis starker und schwacher Erreichbarkeit auf Basis dieses Lösungsansatzes enthält das nächste Kapitel.

## 8 Lösungsansatz II: Anwendungsbeispiele

Für den in Kapitel 7 vorgestellten Lösungsansatz II bietet dieses Kapitel insgesamt sieben ausführliche Anwendungsbeispiele, die in vier für starke (Abschnitt 8.1) und drei für schwache Erreichbarkeit (Abschnitt 8.2) unterteilt sind. In allen Beispielnetzen ist gemäß der Voraussetzung schwache Erreichbarkeit gegeben. Einige Netze wurden bereits in vorherigen Kapiteln zur Erläuterung verschiedener Sachverhalte verwendet. Für eine einheitliche Bezeichnung der  $i$  Beispiele dieses Kapitels wurden sie jedoch in  $B_i$  umbenannt. Einfache Stellen sind in diesen Netzen nur noch vorhanden, wenn sie nicht reduziert werden konnten oder — falls sie durch eine Kante mit  $t_A$  oder  $t_B$  verbunden sind — durften. Die T-Invarianten wurden mit den Algorithmen von Silva und Martínez [MS82, CS91] bzw. Contejean/Devie [CD94] ermittelt. Für die Berechnung der T-Invarianten der in Schichten entfalteten Netze wurde das Mathematica-Paket *IntegerProgramming* von Michael Gensty [Gen05] verwendet.

### 8.1 Beispiele für starke Erreichbarkeit

Alle in diesem Abschnitt behandelten Beispiele B1 bis B4 tauchen bereits in vorherigen Kapiteln der Arbeit auf.

- B1: Dieses Netz wurde erstmalig auf Seite 9 abgebildet und in Kapitel 7 auf starke Erreichbarkeit untersucht.
- B2: Das Netz wurde z. B. in Kapitel 4.2 zur anschaulichen Beschreibung des Verfahrens von Ramachandran herangezogen.
- B3: Hierbei handelt es sich um die um einfache Stellen reduzierte Variante eines Netzes aus [Kos08] und wurde zur Erläuterung von Kostins Verfahren (Abschnitt 4.1) sowie als Beispiel in Kapitel 5.5 verwendet.
- B4: Dieses Netz stammt ebenfalls aus der Veröffentlichung von Kostin [Kos08], wurde bereits um einfache Stellen reduziert und bei der Beschreibung des Lösungsansatzes I in Kapitel 6 zur Erläuterung einiger Sachverhalte herangezogen.

### 8.1.1 Netz B1

Das in Abbildung 8.1 dargestellte Netz besitzt die minimalen T-Invarianten:

$$\begin{array}{c|cccc} & t_1 & t_2 & t_A & t_B \\ \hline \vec{r}^1 & 2 & 1 & 0 & 1 \\ \vec{r}^2 & 0 & 1 & 2 & 1 \end{array}$$

Da noch keine T-Invariante  $\vec{r}$  mit  $\vec{r}(t_A) = \vec{r}(t_B) = 1$  existiert, wird durch eine zusätz-

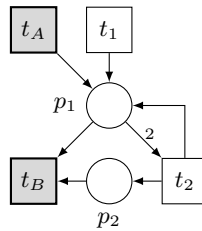


Abbildung 8.1: Netz B1<sub>AB</sub>

liche Stelle  $p_C$ , die  $t_A$  mit  $t_B$  verbindet, eine gleiche Schalthäufigkeit von  $t_A$  und  $t_B$  erzwungen. Die einzige T-Invariante dieses erweiterten Netzes B1<sub>ABC</sub> ist

$$\begin{array}{c|cccc} & t_1 & t_2 & t_A & t_B \\ \hline \vec{r} & 1 & 1 & 1 & 1 \end{array}$$

und entspricht der Linearkombination  $\frac{1}{2} \cdot \vec{r}^1 + \frac{1}{2} \cdot \vec{r}^2$ . Die Netzdarstellung von  $\vec{r}$  hat weder Deadlocks noch Traps; damit ist schwache Erreichbarkeit nachgewiesen.

#### Ermitteln der minimalen Kreise

Im nächsten Schritt wird mit Algorithmus 5.3 (vgl. S. 62) der erweiterte Synchronisationsgraph zu B1<sub>AB</sub> (Abbildung 8.2) erzeugt und dessen S-Invarianten berechnet. Die einzige S-Invariante

$$\begin{array}{c|cccccc} & p_{1 \triangleleft A} & p_{1 \triangleleft 1} & p_{1 \triangleleft 2} & p_{1 \triangleright B} & p_{1 \triangleright 2} & p_2 \\ \hline \vec{i}^1 & 0 & 0 & 2 & 0 & 1 & 0 \end{array}$$

dieses erweiterten Synchronisationsgraphen beschreibt den einzigen minimalen Kreis, die Schlinge  $p_1 \rightarrow t_2 \rightarrow p_1$  in B1<sub>AB</sub>. Algorithmus 5.4 (vgl. S. 64) berechnet auf Basis

der S-Invariante und der Häufigkeitstabelle

	$p_{1\triangleleft A}$	$p_{1\triangleleft 1}$	$p_{1\triangleleft 2}$	$p_{1\triangleright B}$	$p_{1\triangleright 2}$	$p_2$
$\vec{i}^1$	0	0	2	0	1	0
$KH(p)$	0	0	1	–	–	0

die Liste  $K = \{p_{1\triangleleft 2}\}$ , die die aufzuschneidende Kante  $(t_2, p_1)$  in  $B1_{AB}$  repräsentiert. Mit dieser Information kann mit der Schichtenbildung begonnen werden.

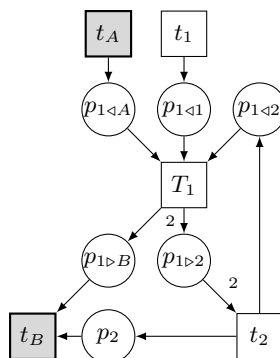


Abbildung 8.2: Erweiterter Synchronisationsgraph zu Netz  $B1_{AB}$

### Nachweis starker Erreichbarkeit im kreisfreien Netz

Die Kante  $(t_2, p_1)$  wird auf Schicht 0 durch Hilfstransitionen  $t'_{2,1}$  und  $t''_{2,1}$  sowie eine Stelle  $s_{2,1}$  ersetzt. Dieses transformierte Netz  $B1_{AB}|_0^{cf}$  bildet zusammen mit Originalnetz  $B1$ ,  $K = \{p_{1\triangleleft 2}\}$  und einer geeigneten Konstante  $h$  für die Constraints im linearen Programm die Eingabe für Algorithmus 7.3 von Seite 115. Als ausreichend hat sich die Konstante  $h = 100$  erwiesen. In Algorithmus 7.3 werden folgende Zielfunktion und Nebenbedingungen erzeugt:

$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}(t'_{2,1}) + 100 \cdot \vec{r}(t''_{2,1}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } & \{\vec{r}(t_A) = \vec{r}(t_B) \geq 1, \vec{r}(t'_{2,1}) = \vec{r}(t''_{2,1})\} \end{aligned}$$

Zunächst berechnet GLOP eine Lösung für  $B1_{AB}|_0^{cf}$ , in der die Hilfstransitionen  $t'_{2,1}$  und  $t''_{2,1}$  noch schalten. Der Zielfunktionswert `neuer_zfwert` beträgt 202. Danach wird mit der Schichtenbildung begonnen und für das neu erzeugte Netz  $B1_{AB}|_1^{cf}$  nach einer Lösung gesucht. Dieses Mal ist bereits starke Erreichbarkeit nachweisbar: `neuer_zfwert` = 2, d.h.  $t_A$  und  $t_B$  schalten je einmal und die Hilfstransitionen  $t'_{2,1}$  und  $t''_{2,1}$  werden nicht mehr benötigt. In Tabelle 8.1 sind die Schaltsequenzen für die berechneten T-Invarianten mit zugehörigen Zielfunktionswerten abgebildet. Die starke bzw. schwache Erreichbarkeit nachweisenden Schaltsequenzen sind ins Originalnetz  $B1_{AB}$  übertragbar,

indem man alle Hilfstransitionen  $c_i$  und für jede Transition  $t_{j|n}$  die Angabe „ $n$ “ der jeweiligen Schicht ignoriert. Im Fall  $B1_{AB}|_1^{cf}$  wäre also  $t_A t_1 t_2 t_B$  die passende Schaltsequenz.

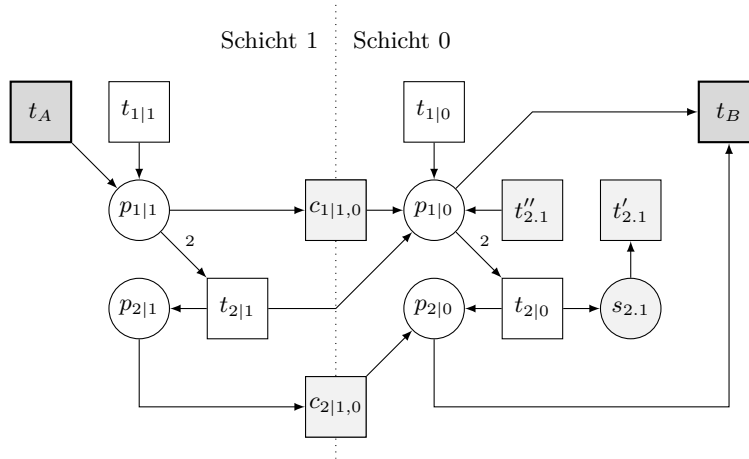


Abbildung 8.3: Netz  $B1_{AB}|_1^{cf}$

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B1_{AB} _0^{cf}$	202	$t_A, t_{1 0}, t_{2 0}, t'_{2,1}, t''_{2,1}, t_B$
$B1_{AB} _1^{cf}$	2	$t_A, t_{1 1}, t_{2 1}, c_{2 1,0}, t_B$

Tabelle 8.1: Lösungen des GLOP für das schichtweise erweiterte Netz  $B1_{AB}^{cf}$

### Nachweis starker Erreichbarkeit mit Kreisen auf Schicht 0

Da die Kante  $(t_2, p_1)$  in diesem Fall auf Schicht 0 erhalten bleiben soll, wird zunächst nichts im Netz ersetzt.  $B1_{AB}|_0$  bildet zusammen mit  $B1$ ,  $K = \{p_{1|2}\}$  und der gleichen Konstante  $h = 100$  die Eingabe für Algorithmus 7.5 von Seite 121. In der Zielfunktion sind die Schalthäufigkeiten aller Originaltransitionen auf der untersten Schicht, d.h.  $t_{1|0}$  und  $t_{2|0}$ , sowie  $t_A$  und  $t_B$  zu minimieren. Es werden nun Zielfunktion und Nebenbedingungen erzeugt:

$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}(t_{1|0}) + 100 \cdot \vec{r}(t_{2|0}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } & \{ \vec{r}(t_A) = \vec{r}(t_B) \geq 1 \} \end{aligned}$$

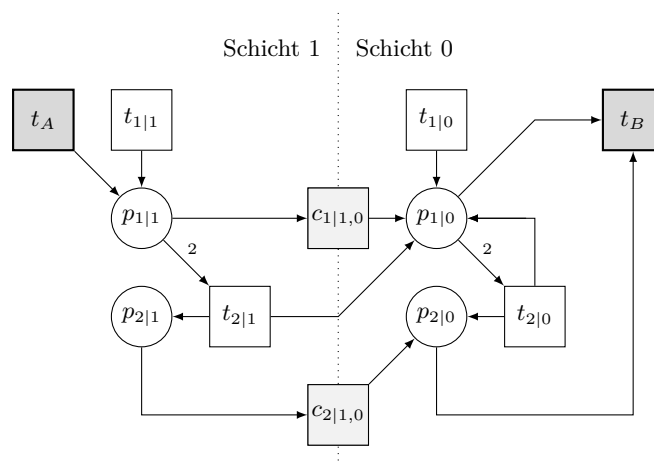
GLOP ermittelt eine Lösung für  $B1_{AB}|_0$ , in der alle Transitionen auf der untersten Schicht schalten. Der Zielfunktionswert `neuer_zfwert` ist 202. Danach wird eine Schicht hinzugefügt und für das neue Netz  $B1_{AB}|_1$  nach einer Lösung gesucht. Dieses Mal ist wie



im kreisfreien Netz bereits starke Erreichbarkeit nachweisbar:  $\text{neuer\_zfwert} = 2$ , d.h.  $t_A$  und  $t_B$  schalten je einmal und auf der untersten Schicht feuert keine Originaltransition. In Tabelle 8.2 sind alle Ergebnisse abgebildet.

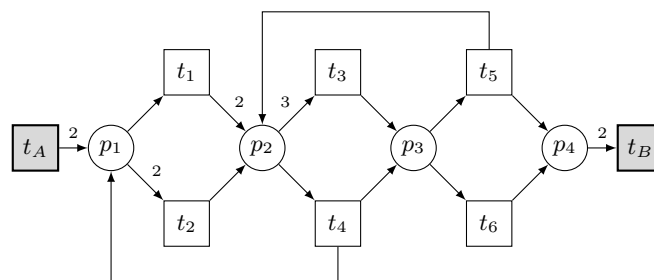
Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B1_{AB 0}$	202	$t_A, t_{1 0}, t_{2 0}, t_B$
$B1_{AB 1}$	2	$t_A, t_{1 1}, t_{2 1}, c_{2 1,0}, t_B$

**Tabelle 8.2:** Lösungen des GLOP für das schichtweise erweiterte Netz  $B1_{AB}$  mit Kreisen auf Schicht 0



**Abbildung 8.4:** Netz  $B1_{AB|1}$

### 8.1.2 Netz B2



**Abbildung 8.5:** Netz  $B2_{AB}$

Das in Abbildung 8.5 dargestellte Netz besitzt zehn minimale T–Invarianten:

	$t_A$	$t_B$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	
$\vec{r}^1$	1	1	2	0	2	0	2	0	$\Leftarrow$
$\vec{r}^2$	4	1	0	4	2	0	2	0	
$\vec{r}^3$	0	1	1	0	1	1	2	0	
$\vec{r}^4$	0	5	0	4	2	8	10	0	
$\vec{r}^5$	3	2	6	0	4	0	0	4	
$\vec{r}^6$	6	1	0	6	2	0	0	2	
$\vec{r}^7$	0	2	3	0	1	3	0	4	
$\vec{r}^8$	1	1	0	2	0	2	0	2	$\Leftarrow$
$\vec{r}^9$	0	1	0	1	0	2	1	1	
$\vec{r}^{10}$	0	3	2	2	0	6	0	6	

Keine Netzdarstellung einer geeigneten T–Invariante, in der  $t_A$  und  $t_B$  nur einmal schalten, ist frei von Deadlocks und Traps.  $B_{2AB}$  wird um die zusätzliche Stelle  $p_C$  erweitert;  $B_{2ABC}$  besitzt die folgenden T–Invarianten:

	$t_A$	$t_B$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	
$\vec{r}_C^1$	1	1	2	0	2	0	2	0	$\Leftarrow$
$\vec{r}_C^2$	1	1	0	2	0	2	0	2	$\Leftarrow$
$\vec{r}_C^3$	5	5	0	8	4	6	10	0	
$\vec{r}_C^4$	2	2	5	0	3	1	0	4	

Nach wie vor ist keine passende T–Invariante gefunden, die schwache Erreichbarkeit nachweist. Für das um  $p_C$  erweiterte Netz werden nun spezielle Generatoren  $\vec{r}^i$  des  $\mathbb{N}$ –Erzeugendensystems (Bedingung:  $\vec{r}^i(t_A) = \vec{r}^i(t_B) \leq 1$ ) ermittelt. Die Vektoren sind:

	$t_A$	$t_B$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	
$\vec{r}_N^1$	1	1	2	0	2	0	2	0	
$\vec{r}_N^2$	1	1	1	1	1	1	1	1	$\Leftarrow$
$\vec{r}_N^3$	1	1	0	2	0	2	0	2	

T–Invariante  $\vec{r}_N^2$ , die Linearkombination  $\frac{1}{2} \cdot \vec{r}^1 + \frac{1}{2} \cdot \vec{r}^8$  der minimalen T–Invarianten von  $B_{2AB}$ , erfüllt schließlich die Bedingung der Deadlock- und Trapfreiheit und weist schwache Erreichbarkeit nach.

### Ermitteln der minimalen Kreise

Nun wird mit Algorithmus 5.3 der erweiterte Synchronisationsgraph zu  $B_{2AB}$  (Abbildung 8.6) konstruiert und dessen S–Invarianten berechnet. Sie beschreiben die Kreise:

$$\begin{aligned}
 p_1 &\rightarrow t_1 \rightarrow p_2 \rightarrow t_4 \rightarrow p_1 \\
 p_1 &\rightarrow t_2 \rightarrow p_2 \rightarrow t_4 \rightarrow p_1 \\
 p_2 &\rightarrow t_3 \rightarrow p_3 \rightarrow t_5 \rightarrow p_2 \\
 p_2 &\rightarrow t_4 \rightarrow p_3 \rightarrow t_5 \rightarrow p_2
 \end{aligned}$$

Die Häufigkeitstabelle  $KH$  für die vier S-Invarianten lautet:

	$p_{1\triangleleft A}$	$p_{1\triangleleft 4}$	$p_{1\triangleright 1}$	$p_{1\triangleright 2}$	$p_{2\triangleleft 5}$	$p_{2\triangleleft 1}$	$p_{2\triangleleft 2}$	$p_{2\triangleright 3}$	$p_{2\triangleright 4}$	$p_{3\triangleleft 3}$	$p_{3\triangleleft 4}$	$p_{3\triangleright 5}$	$p_{3\triangleright 6}$	$p_{4\triangleleft 5}$	$p_{4\triangleleft 6}$	$p_{4\triangleright B}$
$\vec{i}^1$	0	2	2	0	0	1	0	0	2	0	0	0	0	0	0	0
$\vec{i}^2$	0	2	0	1	0	0	2	0	2	0	0	0	0	0	0	0
$\vec{i}^3$	0	0	0	0	3	0	0	1	0	3	0	3	0	0	0	0
$\vec{i}^4$	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0
$KH(p)$	0	2	-	-	2	1	1	-	-	1	1	-	-	0	0	-

Der maximale Wert der Häufigkeitstabelle beträgt 2 und kommt zweimal vor. Die erste zugehörige Stelle, hier  $p_{1\triangleleft 4}$ , wird in  $K$  übernommen. Anschließend wird  $p_{1\triangleleft 4}$  im erweiterten Synchronisationsgraphen gelöscht und dessen S-Invarianten mit Häufigkeitstabelle erneut berechnet:

	$p_{1\triangleleft A}$	$p_{1\triangleright 1}$	$p_{1\triangleright 2}$	$p_{2\triangleleft 5}$	$p_{2\triangleleft 1}$	$p_{2\triangleleft 2}$	$p_{2\triangleright 3}$	$p_{2\triangleright 4}$	$p_{3\triangleleft 3}$	$p_{3\triangleleft 4}$	$p_{3\triangleright 5}$	$p_{3\triangleright 6}$	$p_{4\triangleleft 5}$	$p_{4\triangleleft 6}$	$p_{4\triangleright B}$
$\vec{i}^1$	0	0	0	3	0	0	1	0	3	0	3	0	0	0	0
$\vec{i}^2$	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0
$KH(p)$	0	-	-	2	0	0	-	-	1	1	-	-	0	0	-

Der nun einzige maximale Wert ist Stelle  $p_{2\triangleleft 5}$  zugeordnet. Damit wird  $K$  ergänzt zu  $K = \{p_{1\triangleleft 4}, p_{2\triangleleft 5}\}$ . Wird  $p_{2\triangleleft 5}$  nun im erweiterten Synchronisationsgraphen gelöscht, hat dieser keine S-Invarianten mehr. Damit beschreibt die Liste  $K$  also die aufzuschneidenden Kanten  $(t_4, p_1)$  und  $(t_5, p_2)$  in  $B2_{AB}$ . Nun wird mit der Schichtenbildung begonnen.

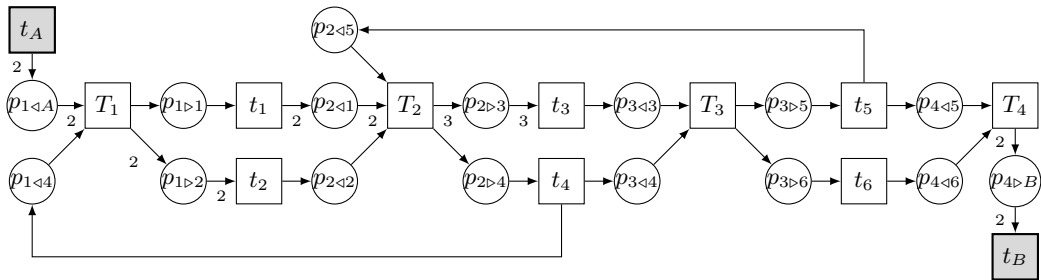


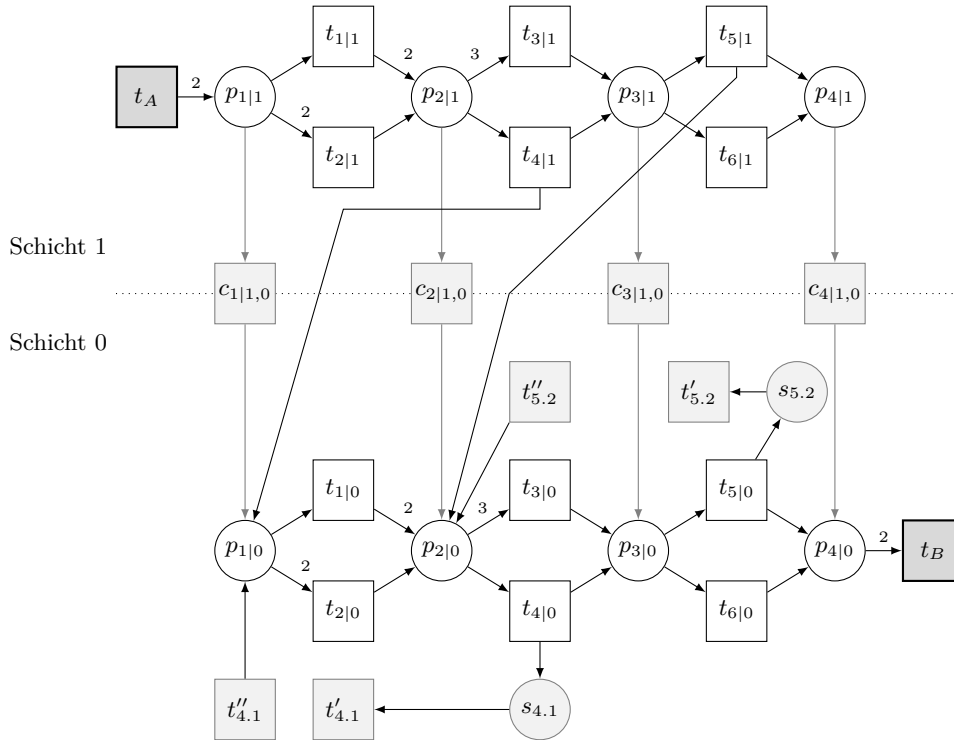
Abbildung 8.6: Erweiterter Synchronisationsgraph zu Netz  $B2_{AB}$

**Nachweis starker Erreichbarkeit im kreisfreien Netz**

Die Kanten  $(t_4, p_1)$  und  $(t_5, p_2)$  werden auf Schicht 0 durch Hilfstransitionen  $t'_{4.1}$ ,  $t''_{4.1}$ ,  $t'_{5.2}$ ,  $t''_{5.2}$  sowie Stellen  $s_{4.1}$  und  $s_{5.2}$  ersetzt. Das transformierte Netz  $B2_{AB}|_0^{cf}$ ,  $B2$ ,  $K = \{p_{1<4}, p_{2<5}\}$  und die Konstante  $h = 100$  werden Algorithmus 7.3 übergeben. Dieser konstruiert folgende Zielfunktion und Nebenbedingungen:

$$\begin{aligned} \text{ZF: } & 100 \cdot \bar{r}(t'_{4.1}) + 100 \cdot \bar{r}(t''_{4.1}) + 100 \cdot \bar{r}(t'_{5.2}) + 100 \cdot \bar{r}(t''_{5.2}) + \bar{r}(t_A) + \bar{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } & \{\bar{r}(t_A) = \bar{r}(t_B) \geq 1, \bar{r}(t'_{4.1}) = \bar{r}(t''_{4.1}), \bar{r}(t'_{5.2}) = \bar{r}(t''_{5.2})\} \end{aligned}$$

Zunächst berechnet GLOP eine Lösung für  $B2_{AB}|_0^{cf}$ , in der die Hilfstransitionen  $t'_{4.1}$  und  $t''_{4.1}$  noch schalten,  $t_A$  und  $t_B$  je zweimal feuern und der Zielfunktionswert `neuer_zfwert` damit 204 beträgt. Danach wird mit der Schichtenbildung begonnen und bereits im nächsten Schritt für  $B2_{AB}|_1^{cf}$  starke Erreichbarkeit nachgewiesen: `neuer_zfwert = 2`. In Tabelle 8.3 sind die Schaltsequenzen für die beiden T-Invarianten mit zugehörigen Zielfunktionswerten abgebildet.



**Abbildung 8.7:** Netz  $B2_{AB}|_1^{cf}$

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B2_{AB} _0^{cf}$	204	$2 \cdot t_A, 4 \cdot t_{1 0}, t_{4 0}, t'_{4.1}, t''_{4.1}, t_{1 0}, 3 \cdot t_{3 0}, 4 \cdot t_{6 0}, 2 \cdot t_B$
$B2_{AB} _1^{cf}$	2	$t_A, t_{2 1}, t_{4 1}, t_{5 1}, c_{4 1,0}, t_{1 0}, t_{3 0}, t_{6 0}, t_B$

Tabelle 8.3: Lösungen des GLOP für das schichtweise erweiterte Netz  $B2_{AB}^{cf}$ **Nachweis starker Erreichbarkeit mit Kreisen auf Schicht 0**

Netz  $B2_{AB}|_0$  bildet zusammen mit  $B2$ ,  $K = \{p_{1 \leftarrow 4}, p_{2 \leftarrow 5}\}$  und Konstante  $h = 100$  die Eingabe für Algorithmus 7.5. Es werden nun Zielfunktion und Nebenbedingungen erzeugt:

$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}(t_{1|0}) + 100 \cdot \vec{r}(t_{2|0}) + 100 \cdot \vec{r}(t_{3|0}) + 100 \cdot \vec{r}(t_{4|0}) + 100 \cdot \vec{r}(t_{5|0}) \\ & + 100 \cdot \vec{r}(t_{6|0}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } & \{\vec{r}(t_A) = \vec{r}(t_B) \geq 1\} \end{aligned}$$

Für  $B2_{AB}$  werden bis zum Nachweis der starken Erreichbarkeit insgesamt zwei zusätzliche Schichten benötigt. Der Zielfunktionswert sinkt, bis letztendlich in  $B2_{AB}|_2$  die leere Markierung ohne Schalten einer Transition der untersten Schicht (bis auf  $t_B$ ) reproduziert wird. In Tabelle 8.4 sind die Ergebnisse abgebildet. Die Schaltfolgen für  $B2_{AB}|_0$  und  $B2_{AB}|_1$  sind nicht realisierbar, was auf die Kreise der untersten Schicht zurückzuführen ist: In  $B2_{AB}|_1$  können beispielsweise alle Transitionen bis zu  $t_{3|0}$  schalten. Die Transitionen  $t_{3|0}$  und  $t_{5|0}$  liegen auf einem Kreis. Das Fehlen der Marke für  $t_{3|0}$ , die  $t_{5|0}$  für sie produzieren müsste, wird aufgrund dieses Kreises nicht bemerkt.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B2_{AB} _0$	602	$t_A, 2 \cdot t_{2 0}, 2 \cdot t_{4 0}, 2 \cdot t_{6 0}, t_B$ (nicht realisierbar)
$B2_{AB} _1$	202	$t_A, 2 \cdot t_{1 1}, t_{3 1}, t_{5 1}, t_{2 1,0}, c_{4 1,0}, t_{3 0}, t_{5 0}, t_B$ (nicht realisierbar)
$B2_{AB} _2$	2	$t_A, t_{2 2}, t_{4 2}, t_{5 2}, c_{4 2,1}, t_{1 1}, t_{3 1}, t_{6 1}, 2 \cdot c_{4 1,0}, t_B$

Tabelle 8.4: Lösungen des GLOP für das schichtweise erweiterte Netz  $B2_{AB}$  mit Kreisen auf Schicht 0

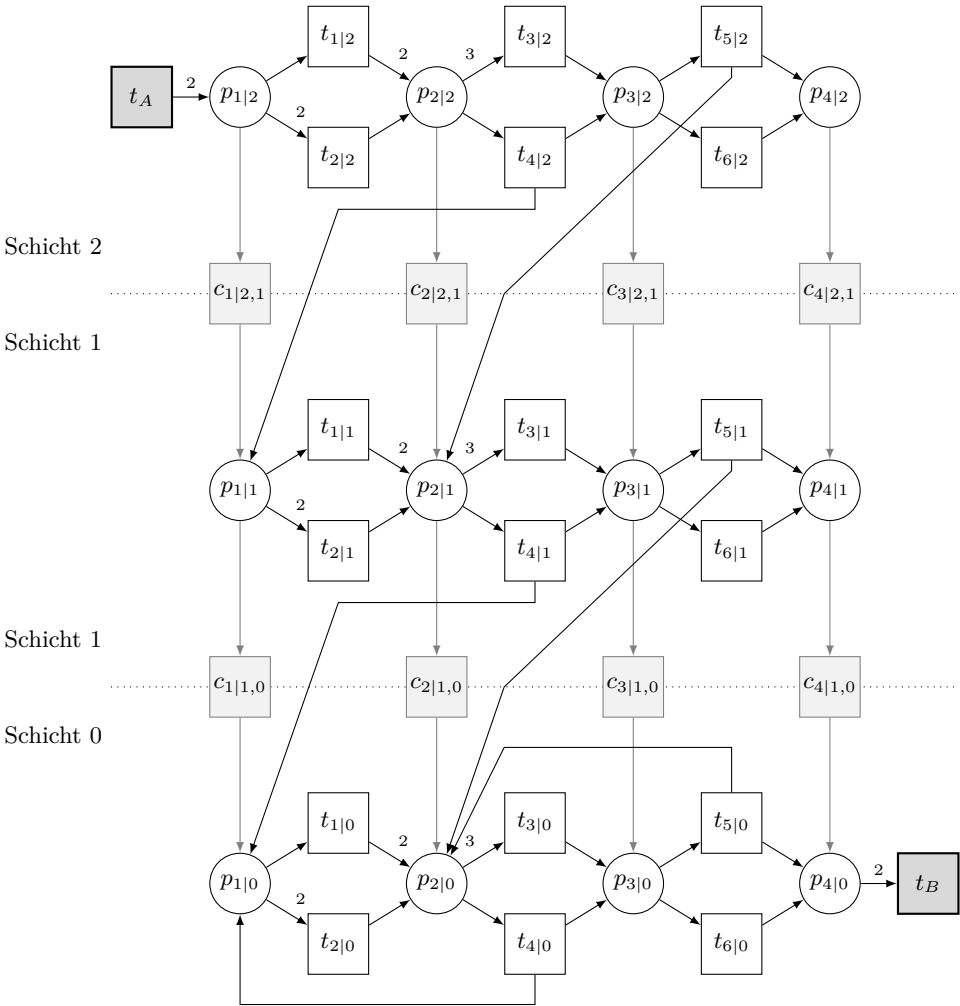


Abbildung 8.8: Netz  $B_{2|AB|2}$

8.1.3 Netz B3

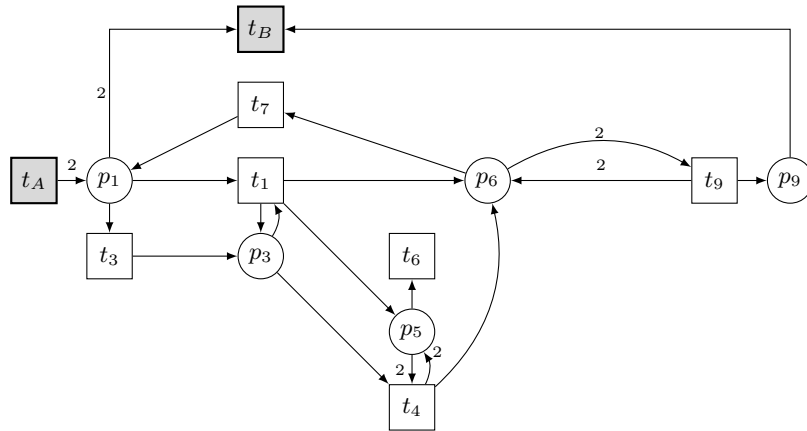


Abbildung 8.9: Netz B3<sub>AB</sub>

Das Netz in Abbildung 8.9 hat folgende drei minimale T-Invarianten, deren Summe eine schwache Erreichbarkeit nachweisende Linearkombination ist:

	$t_1$	$t_3$	$t_4$	$t_6$	$t_7$	$t_9$	$t_A$	$t_B$
$\vec{r}^1$	0	1	1	0	1	0	0	0
$\vec{r}^2$	1	0	0	1	1	0	0	0
$\vec{r}^3$	0	0	0	0	0	1	1	1

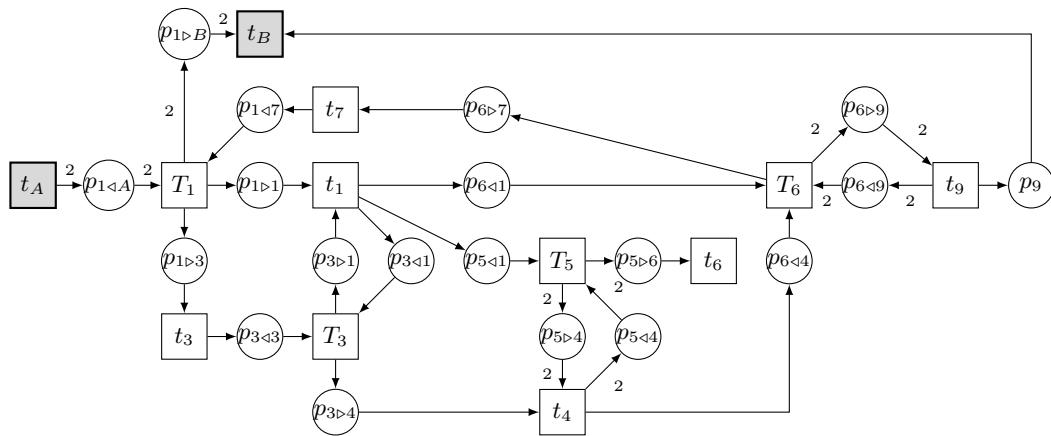


Abbildung 8.10: Erweiterter Synchronisationsgraph zu Netz B3<sub>AB</sub>

**Ermitteln der minimalen Kreise**

In Abbildung 8.10 ist der zugehörige erweiterte Synchronisationsgraph dargestellt. Dieser besitzt die folgenden insgesamt neun S-Invarianten:

$p$	9	1	◀A	1▷	3	1▷	1	◀7	1▷	B	3	◀3	3▷	1	3	◀4	5	◀1	5▷	4	5	◀4	5▷	6	6▷	7	6	◀1	6	◀4	6	◀9	6▷	9
$\vec{i}^1$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	
$\vec{i}^2$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
$\vec{i}^3$	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
$\vec{i}^4$	0	0	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	
$\vec{i}^5$	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	
$\vec{i}^6$	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	2	0	2	0	2	0	0	0	0	0	0	
$\vec{i}^7$	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	
$\vec{i}^8$	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	
$\vec{i}^9$	0	0	2	0	2	0	2	2	0	0	2	2	0	0	2	1	0	0	0	0	0	2	0	2	0	2	0	2	0	0	0	0	0	
$KH(p)$	0	0	-	-	6	-	3	-	2	-	2	-	2	-	2	-	1	-	-	2	4	1	-											

Alle aufzuschneidenden Kanten wurden bereits in Abschnitt 5.5.2 ab Seite 64 ermittelt. Algorithmus 5.4 findet zunächst drei Schlingen beschreibende Kreise in den S-Invarianten  $\vec{i}^1$  bis  $\vec{i}^3$  und  $p_{6<9}$ ,  $p_{5<4}$  und  $p_{3<1}$  werden in  $K$  übernommen. Nach Löschung der Stellen verbleiben die folgenden S-Invarianten mit aktualisierter Häufigkeitstabelle:

$p$	9	1	◀A	1▷	3	1▷	1	◀7	1▷	B	3	◀3	3▷	1	3	◀4	5	◀1	5▷	4	5	◀6	6▷	7	6	◀1	6	◀4	6▷	9		
$\vec{i}^1$	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0					
$\vec{i}^2$	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0							
$\vec{i}^3$	0	0	0	2	2	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	2	0	2	0							
$\vec{i}^4$	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0							
$\vec{i}^5$	0	0	2	0	2	0	2	2	0	2	2	0	2	1	0	2	1	0	0	0	0	2	0	2	0							
$KH(p)$	0	0	-	-	5	-	3	-	-	2	-	-	2	-	-	-	-	-	-	-	-	2	3	-								

Der maximale Wert in  $KH$  steht an Stelle  $p_{1<7}$ . Nach der Löschung dieser Stelle ist der erweiterte Synchronisationsgraph kreisfrei. Damit ist  $K = \{p_{6<9}, p_{5<4}, p_{3<1}, p_{1<7}\}$  und repräsentiert die in  $B_{3AB}$  aufzuschneidenden Kanten  $(t_9, p_6)$ ,  $(t_4, p_5)$ ,  $(t_1, p_3)$  und  $(t_7, p_1)$ . Nun kann auf starke Erreichbarkeit untersucht werden.

**Nachweis starker Erreichbarkeit im kreisfreien Netz**

Die vier Kanten werden auf Schicht 0 durch acht Hilfstransitionen sowie vier Stellen ersetzt. Algorithmus 7.3 erhält das nun kreisfreie Netz  $B_{3AB}|_0^{cf}$ ,  $B_3$ ,  $K$  und die Konstante  $h$  als Eingabe. Zunächst werden Zielfunktion und Nebenbedingungen konstruiert:

$$\begin{aligned}
 \text{ZF: } & 100 \cdot \vec{r}(t'_{9.6}) + 100 \cdot \vec{r}(t''_{9.6}) + 100 \cdot \vec{r}(t'_{4.5}) + 100 \cdot \vec{r}(t''_{4.5}) \\
 & + 100 \cdot \vec{r}(t'_{1.3}) + 100 \cdot \vec{r}(t''_{1.3}) + 100 \cdot \vec{r}(t'_{7.1}) + 100 \cdot \vec{r}(t''_{7.1}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min}
 \end{aligned}$$



$$\text{cons: } \{\bar{r}(t_A) = \bar{r}(t_B) \geq 1, \bar{r}(t'_{9.6}) = \bar{r}(t''_{9.6}), \bar{r}(t'_{4.5}) = \bar{r}(t''_{4.5}), \bar{r}(t'_{1.3}) = \bar{r}(t''_{1.3}), \\ \bar{r}(t'_{7.1}) = \bar{r}(t''_{7.1})\}$$

GLOP ermittelt danach eine Lösung für  $B3_{AB}|_0^{\text{cf}}$ , in der die Hilfstransitionen  $t'_{9.6}$  und  $t''_{9.6}$  schalten, was einem Zielfunktionswert von 202 entspricht. Der Wert ändert sich erst mit der dritten zusätzlichen Schicht, wenn schwache Erreichbarkeit nachgewiesen wird. Nun muss beobachtet werden, ob sich die Schalthäufigkeiten von  $t_A$  und  $t_B$  noch reduzieren lassen, d.h. der Zielfunktionswert nun streng monoton fällt. In der Tat ist dies mit einer weiteren Schicht der Fall: In  $B3_{AB}|_4^{\text{cf}}$  schalten beide Transitionen nur einmal und starke Erreichbarkeit ist bewiesen. In Tabelle 8.5 sind die Schaltsequenzen mit zugehörigen Zielfunktionswerten abgebildet.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\bar{r}$ )
$B3_{AB} _0^{\text{cf}}$	202	$t_A, t''_{9.6}, t_{9 0}, t'_{9.6}, t_B$
$B3_{AB} _1^{\text{cf}}$	202	$t_A, 2 \cdot c_{1 1,0}, t''_{9.6}, t_{9 0}, t'_{9.6}, t_B$
$B3_{AB} _2^{\text{cf}}$	202	$t_A, 2 \cdot c_{1 2,1}, 2 \cdot c_{1 1,0}, t''_{9.6}, t_{9 0}, t'_{9.6}, t_B$
$B3_{AB} _3^{\text{cf}}$	4	$2 \cdot t_A, 2 \cdot t_{3 3}, 2 \cdot t_{1 3}, t_{9 3}, 2 \cdot c_{5 3,2}, c_{9 3,2}, t_{4 2}, t_{7 2}, \\ t_{9 2}, c_{3 2,1}, 2 \cdot c_{9 2,1}, t_{4 1}, 3 \cdot t_{7 1}, c_{1 1,0}, 2 \cdot c_{9 1,0}, 2 \cdot t_{6 0}, \\ 2 \cdot t_B$
$B3_{AB} _4^{\text{cf}}$	2	$t_A, t_{3 4}, t_{1 4}, t_{7 4}, c_{5 4,3}, t_{1 3}, 2 \cdot c_{5 3,2}, c_{6 3,2}, t_{4 2}, t_{9 2}, \\ c_{9 2,1}, 2 \cdot t_{6 1}, 2 \cdot t_{7 1}, c_{9 1,0}, t_B$

**Tabelle 8.5:** Lösungen des GLOP für das schichtweise erweiterte Netz  $B3_{AB}^{\text{cf}}$

### Nachweis starker Erreichbarkeit mit Kreisen auf Schicht 0

Netz  $B3_{AB}|_0$  bildet zusammen mit  $B3$ ,  $K$  und Konstante  $h = 100$  die Eingabe für Algorithmus 7.5. Es werden folgende Zielfunktion und Nebenbedingungen erzeugt:

$$\text{ZF: } 100 \cdot \bar{r}(t_{1|0}) + 100 \cdot \bar{r}(t_{3|0}) + 100 \cdot \bar{r}(t_{4|0}) + 100 \cdot \bar{r}(t_{6|0}) + 100 \cdot \bar{r}(t_{7|0}) \\ + 100 \cdot \bar{r}(t_{9|0}) + \bar{r}(t_A) + \bar{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } \{\bar{r}(t_A) = \bar{r}(t_B) \geq 1\}$$

Da für  $B3_{AB}|_0$  keine schwache Erreichbarkeit nachweisbar war, fügt der Algorithmus zunächst drei zusätzliche Schichten hinzu und berechnet T-Invarianten und Zielfunktionswerte der erweiterten Netze. Der Zielfunktionswert ändert sich zunächst nicht und bleibt konstant auf 102 stehen. Die Lösung, die in diesen Fällen einen minimalen Wert liefert, entspricht mit dem Schalten von  $t_A$ ,  $t_{9|0}$  und  $t_B$  einer minimalen T-Invariante

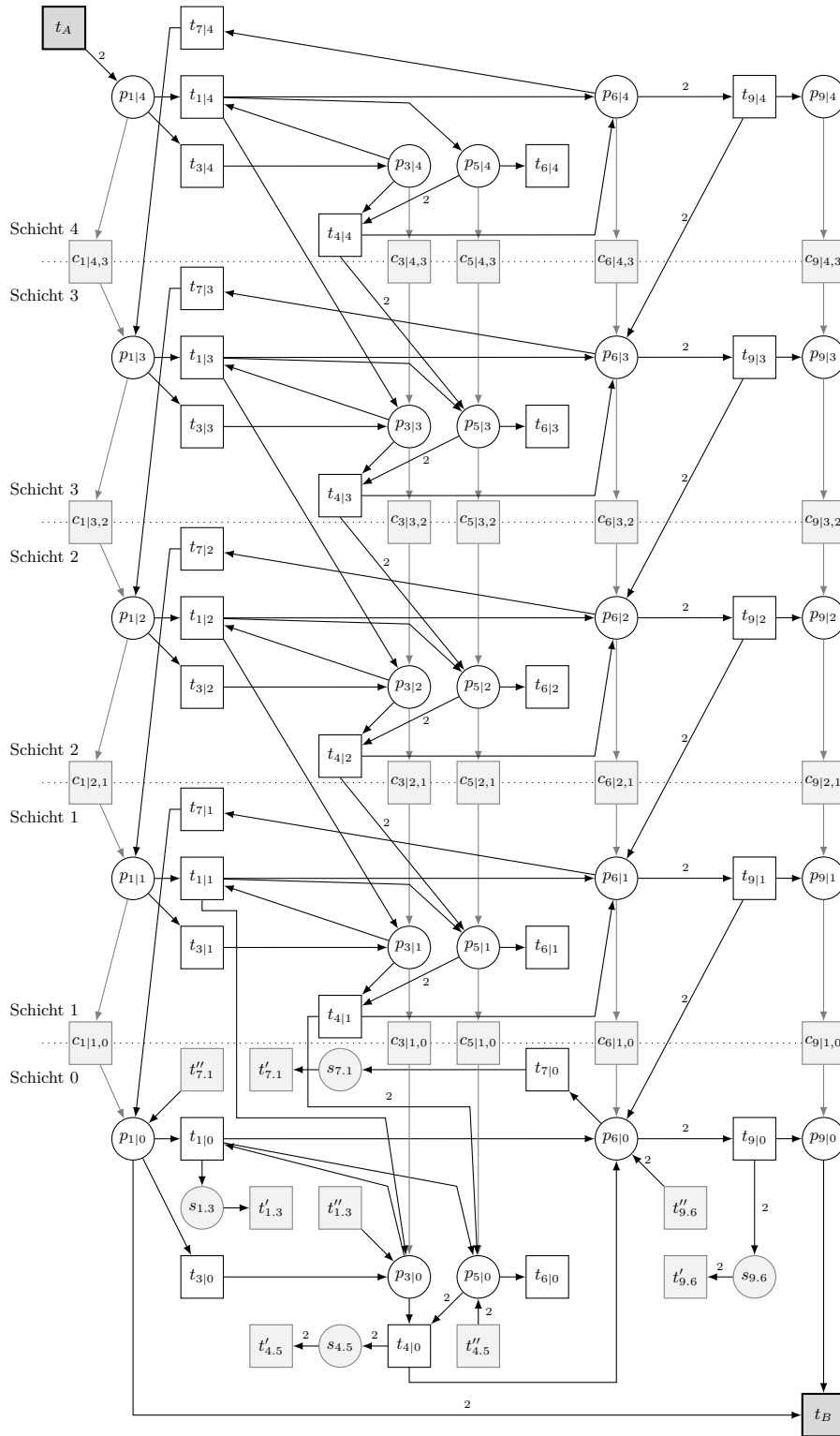


Abbildung 8.11: Netz  $B3_{AB|4}^{cf}$

des Originalnetzes  $B_{3AB}$  (vgl.  $\vec{r}^3$  auf S. 135), die nicht realisierbar ist. Erst mit der vierten zusätzlichen Schicht wird in  $B_{3AB}|_4$  eine T-Invariante gefunden, deren Netzdarstellung kreisfrei ist und die sofort starke Erreichbarkeit nachweist.

Die T-Invariante, die in  $B_{3AB}|_3^{\text{cf}}$  den Nachweis für schwache Erreichbarkeit lieferte, wurde in  $B_{3AB}|_3$  nicht gefunden, da  $t_{6|0}$  zweimal schalten und damit zu einem Zielfunktionswert  $2 \cdot (100 \cdot t_{6|0} + t_A + t_B) = 204 > 102$  geführt hätte. Dass diese Lösung tatsächlich realisierbar wäre und zudem eine kreisfreie Netzdarstellung gehabt hätte, wird aufgrund der Gestalt der Zielfunktion (Originaltransitionen auf der untersten Schicht sollen so selten wie möglich schalten) nicht erkannt. Die Ausgaben des Algorithmus für jede Schicht sind in Tabelle 8.6 zusammengestellt.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B_{3AB} _0$	102	$t_A$ , $t_{9 0}$ , $t_B$ (nicht realisierbar)
$B_{3AB} _1$	102	$t_A$ , $2 \cdot c_{1 1,0}$ , $t_{9 0}$ , $t_B$ (nicht realisierbar)
$B_{3AB} _2$	102	$t_A$ , $2 \cdot c_{1 2,1}$ , $2 \cdot c_{1 1,0}$ , $t_{9 0}$ , $t_B$ (nicht realisierbar)
$B_{3AB} _3$	102	$t_A$ , $2 \cdot c_{1 3,2}$ , $2 \cdot c_{1 2,1}$ , $2 \cdot c_{1 1,0}$ , $t_{9 0}$ , $t_B$ (nicht realisierbar)
$B_{3AB} _4$	2	$t_A$ , $t_{3 4}$ , $t_{1 4}$ , $t_{7 4}$ , $c_{5 4,3}$ , $t_{1 3}$ , $2 \cdot c_{5 3,2}$ , $c_{6 3,2}$ , $t_{4 2}$ , $t_{9 2}$ , $c_{9 2,1}$ , $2 \cdot t_{6 1}$ , $2 \cdot t_{7 1}$ , $c_{9 1,0}$ , $t_B$

**Tabelle 8.6:** Lösungen des GLOP für das schichtweise erweiterte Netz  $B_{3AB}$  mit Kreisen auf Schicht 0

#### 8.1.4 Netz $B_4$

Bei  $B_{4AB}$  handelt es sich um ein (leicht vereinfachtes) Netz aus [Kos08], das bereits in Kapitel 6 vorgestellt wurde. Gemäß Algorithmus 5.1 von Seite 50 wird das Netz zunächst um  $p_C$  zu  $B_{4ABC}$  erweitert und dessen minimale T-Invarianten berechnet:

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_A$	$t_B$
$\vec{r}^1$	0	0	0	0	0	0	0	0	1	1	1
$\vec{r}^2$	2	2	1	0	3	0	0	0	0	0	0
$\vec{r}^3$	0	0	0	0	3	1	0	2	0	0	0
$\vec{r}^4$	1	1	2	3	0	0	0	0	0	0	0
$\vec{r}^5$	0	0	3	6	0	1	0	2	0	0	0
$\vec{r}^6$	2	2	1	0	0	1	2	0	0	0	0
$\vec{r}^7$	0	0	0	0	0	1	1	1	0	0	0

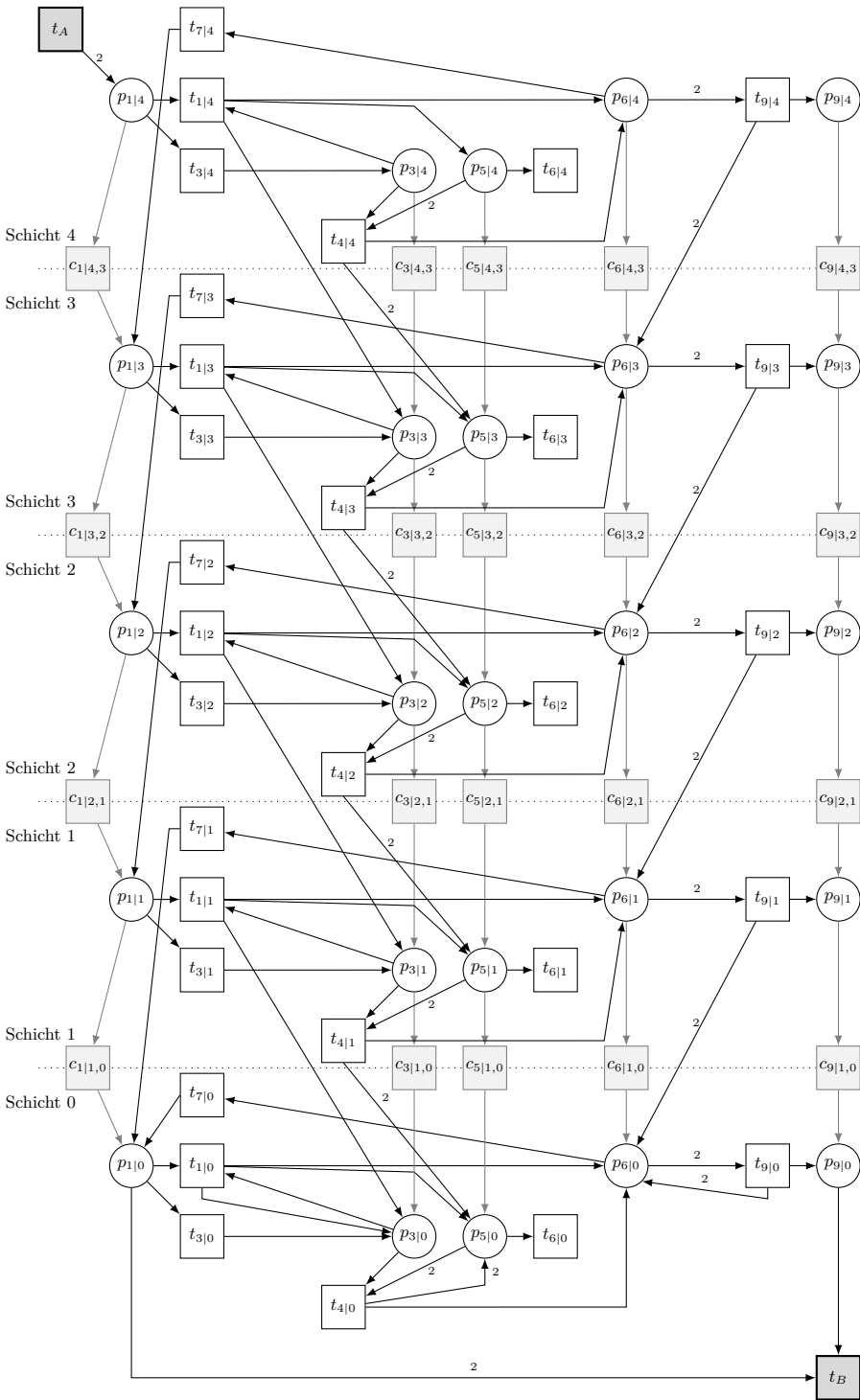


Abbildung 8.12: Netz  $B_{3AB|4}$

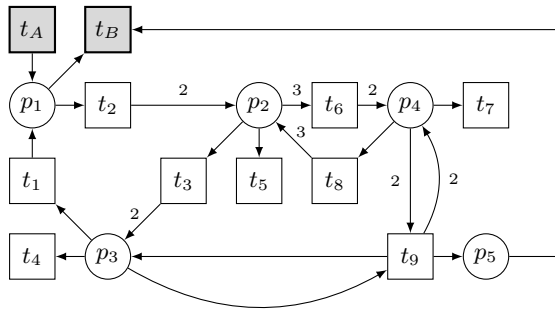


Abbildung 8.13: Netz  $B_{4AB}$

Da die Netzdarstellung von  $\vec{r}^1$  allein nicht frei von Deadlocks und Traps ist, müssen Vektoren addiert werden. Die Summe  $\vec{r}^1 + \vec{r}^2 + \vec{r}^3$  der ersten drei minimalen T-Invarianten bildet eine Linearkombination, in der  $t_A$  und  $t_B$  genau einmal schalten und deren Netzdarstellung keine Deadlocks und Traps enthält. Schwache Erreichbarkeit ist damit nachgewiesen.

**Ermitteln der minimalen Kreise**

Der zu  $B_{4AB}$  gehörende erweiterte Synchronisationsgraph ist in Abbildung 8.14 dar-

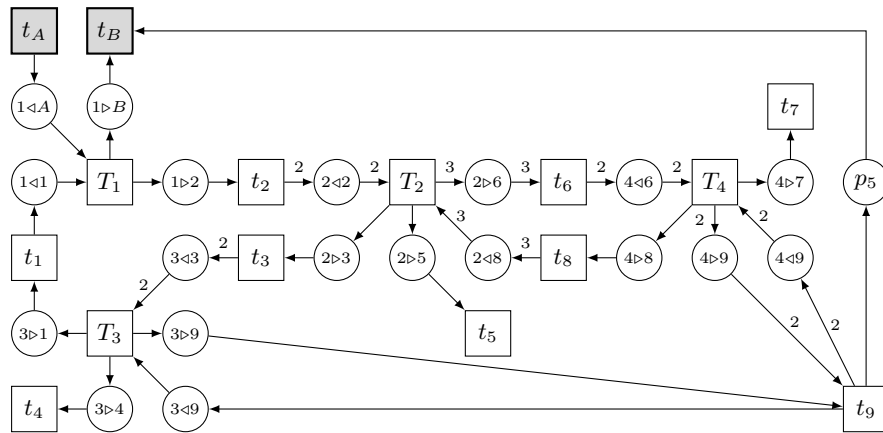


Abbildung 8.14: Erweiterter Synchronisationsgraph zu Netz  $B_{4AB}$

gestellt. Er besitzt sechs minimale S-Invarianten. Bevor die Häufigkeitstabelle genauer betrachtet wird, findet Algorithmus 5.4 zwei Schlingen beschreibende Kreise in den S-Invarianten  $\vec{i}^1$  und  $\vec{i}^2$ ,  $p_{3 \triangleleft 9}$  und  $p_{4 \triangleleft 9}$  werden somit in  $K$  übernommen und entsprechen den aufzuschneidenden Kanten  $(t_9, p_3)$  und  $(t_9, p_4)$ .

	1	<1	1	<A	1	▷B	1	▷2	2	<2	2	<8	2	▷3	2	▷5	2	▷6	3	<3	3	<9	3	▷1	3	▷4	3	▷9	4	<6	4	<9	4	▷8	4	▷9	4	▷7	p <sub>5</sub>
$\vec{i}^1$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
$\vec{i}^2$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	
$\vec{i}^3$	0	0	0	0	0	0	2	0	0	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	3	0	6	0	6	0	0	0	0	0	0	0	
$\vec{i}^4$	2	0	0	2	1	0	2	0	0	1	0	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
$\vec{i}^5$	0	0	0	0	0	2	6	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	3	6	0	3	6	0	0	0	0	0	0	0	
$\vec{i}^6$	6	0	0	6	3	0	0	0	0	2	0	6	6	0	0	0	0	0	0	0	6	6	0	0	0	3	0	0	0	3	0	0	3	0	0	0	0	0	
$KH(p)$	2	0	-	-	2	2	-	-	-	2	2	-	-	-	-	2	2	-	-	-	-	-	-	-	2	2	-	-	-	-	-	-	-	-	-	-	0		

Nach Löschung dieser zwei Stellen im erweiterten Synchronisationsgraphen verbleiben folgende S-Invarianten mit aktualisierter Häufigkeitstabelle:

	1	<1	1	<A	1	▷B	1	▷2	2	<2	2	<8	2	▷3	2	▷5	2	▷6	3	<3	3	<9	3	▷1	3	▷4	3	▷9	4	<6	4	<8	4	▷8	4	▷9	4	▷7	p <sub>5</sub>
$\vec{i}^1$	2	0	0	2	1	0	2	0	0	1	2	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
$\vec{i}^2$	0	0	0	0	0	2	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	6	0	0	0	0	0	0	0	0	0	0	0
$KH(p)$	1	0	-	-	1	1	-	-	-	1	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	0	

Der maximale Wert in  $KH$  ist 1 und kommt mehrmals vor. Es wird darum die erste Stelle  $p$  mit  $KH(p) = 1$  in  $K$  übernommen; in diesem Fall also  $p_{1<1}$ . Nach der Löschung dieser Stelle besitzt der erweiterte Synchronisationsgraph noch genau eine S-Invariante:

	1	<A	1	▷B	1	▷2	2	<2	2	<8	2	▷3	2	▷5	2	▷6	3	<3	3	<9	3	▷1	3	▷4	3	▷9	4	<6	4	<8	4	▷8	4	▷9	4	▷7	p <sub>5</sub>
$\vec{i}^1$	0	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	6	0	0	0	0	0	0	0	0	0	0
$KH(p)$	0	-	-	0	1	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	0

Die letzte zu löschende Stelle ist erneut die erste Stelle mit maximaler Auftrittshäufigkeit, hier  $p_{2<8}$ . Damit erhält man insgesamt  $K = (p_{3<9}, p_{4<9}, p_{1<1}, p_{2<8})$ , was den Kanten  $(t_9, p_3)$ ,  $(t_9, p_4)$ ,  $(t_1, p_1)$  und  $(t_8, p_2)$  entspricht.

### Nachweis starker Erreichbarkeit im kreisfreien Netz

Die vier aufzuschneidenden Kanten werden nun zunächst auf Schicht 0 durch insgesamt acht Hilfstransitionen sowie vier Stellen ersetzt. Algorithmus 7.3 erwartet als Eingabe die Netze  $B_{4AB}|_0^{cf}$  und  $B_4$  sowie  $K$  und die Konstante  $h$ . Zielfunktion und Nebenbedingungen werden konstruiert:

$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}(t'_{9,3}) + 100 \cdot \vec{r}(t''_{9,3}) + 100 \cdot \vec{r}(t'_{9,4}) + 100 \cdot \vec{r}(t''_{9,4}) \\ & + 100 \cdot \vec{r}(t'_{1,1}) + 100 \cdot \vec{r}(t''_{1,1}) + 100 \cdot \vec{r}(t'_{8,2}) + 100 \cdot \vec{r}(t''_{8,2}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min} \end{aligned}$$

cons:  $\{\bar{r}(t_A) = \bar{r}(t_B) \geq 1, \bar{r}(t'_{9.3}) = \bar{r}(t''_{9.3}), \bar{r}(t'_{9.4}) = \bar{r}(t''_{9.4}), \bar{r}(t'_{1.1}) = \bar{r}(t''_{1.1}), \bar{r}(t'_{8.2}) = \bar{r}(t''_{8.2})\}$

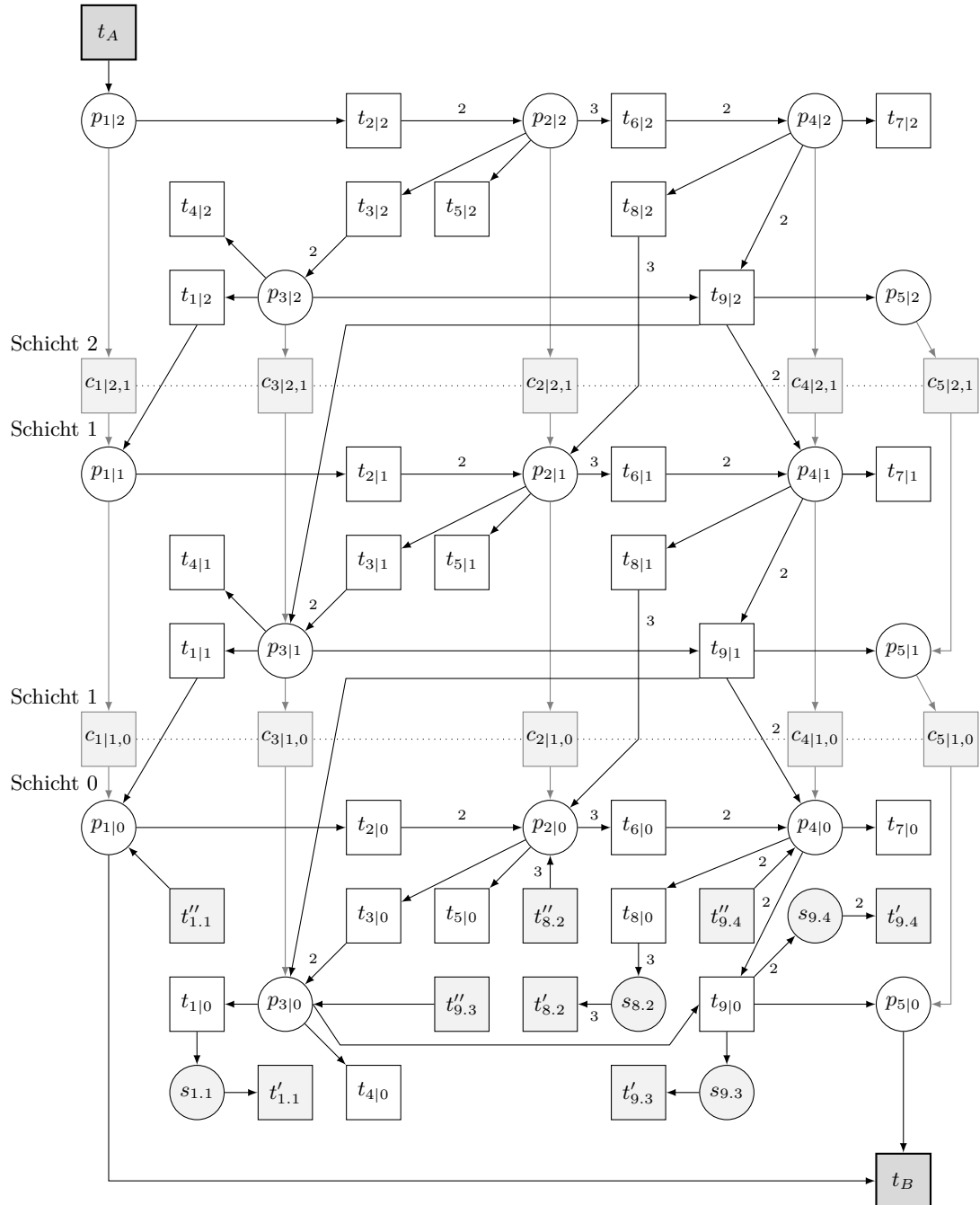


Abbildung 8.15: Netz  $B_{4AB|2}^{cf}$

GLOP ermittelt danach eine Lösung für  $B4_{AB}|_0^{cf}$ , in der insgesamt vier Hilfstransitionen schalten, was einem Zielfunktionswert von 402 entspricht. Dieser Wert ändert sich erst mit der dritten Schicht, wobei sofort starke Erreichbarkeit nachgewiesen wird. Tabelle 8.7 fasst die Ergebnisse zusammen.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B4_{AB} _0^{cf}$	402	$t_A, t''_{9.3}, t''_{9.4}, t_{9 0}, t'_{9.3}, t'_{9.4}, t_B$
$B4_{AB} _1^{cf}$	402	$t_A, c_{1 1,0}, t''_{9.3}, t''_{9.4}, t_{9 0}, t'_{9.3}, t'_{9.4}, t_B$
$B4_{AB} _2^{cf}$	2	$t_A, t_{2 2}, t_{3 2}, t_{5 2}, 2 \cdot t_{1 2}, 2 \cdot t_{2 1}, t_{3 1}, t_{6 1}, t_{9 1}, t_{1 1}, c_{5 1,0}, t_{4 0}, 2 \cdot t_{7 0}, t_B$

**Tabelle 8.7:** Lösungen des GLOP für das schichtweise erweiterte Netz  $B4_{AB}^{cf}$

**Nachweis starker Erreichbarkeit mit Kreisen auf Schicht 0**

Netz  $B4_{AB}|_0$  wird zusammen mit  $B4$ ,  $K$  und Konstante  $h = 100$  Algorithmus 7.5 übergeben. Zunächst werden folgende Zielfunktion und Nebenbedingungen erzeugt:

$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}(t_{1|0}) + 100 \cdot \vec{r}(t_{2|0}) + 100 \cdot \vec{r}(t_{3|0}) + 100 \cdot \vec{r}(t_{4|0}) + 100 \cdot \vec{r}(t_{5|0}) \\ & + 100 \cdot \vec{r}(t_{6|0}) + 100 \cdot \vec{r}(t_{7|0}) + 100 \cdot \vec{r}(t_{8|0}) + 100 \cdot \vec{r}(t_{9|0}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } & \{ \vec{r}(t_A) = \vec{r}(t_B) \geq 1 \} \end{aligned}$$

Für die Netze  $B4_{AB}|_0$  bis  $B4_{AB}|_2$  ist keine schwache Erreichbarkeit nachweisbar und der Zielfunktionswert ändert sich nicht. Die ermittelten T-Invarianten sind nicht realisierbar. Mit der dritten Schicht wird eine starke Erreichbarkeit nachweisende T-Invariante gefunden. Die Ausgaben des Algorithmus sind in Tabelle 8.8 abgebildet.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B4_{AB} _0$	102	$t_A, t_{9 0}, t_B$ (nicht realisierbar)
$B4_{AB} _1$	102	$t_A, c_{1 1,0}, t_{9 0}, t_B$ (nicht realisierbar)
$B4_{AB} _2$	102	$t_A, t_{2 2}, 2 \cdot c_{2 2,1}, t_{3 1}, t_{4 1}, t_{5 1}, t_{1 1}, t_{9 0}, t_B$ (nicht realisierbar)
$B4_{AB} _3$	2	$t_A, t_{2 3}, t_{3 3}, t_{1 3}, c_{2 3,2}, c_{3 3,2}, t_{2 2}, t_{6 2}, t_{9 2}, c_{5 2,1}, t_{1 1}, 2 \cdot t_{7 1}, c_{5 1,0}, t_B$

**Tabelle 8.8:** Lösungen des GLOP für das schichtweise erweiterte Netz  $B4_{AB}$  mit Kreisen auf Schicht 0



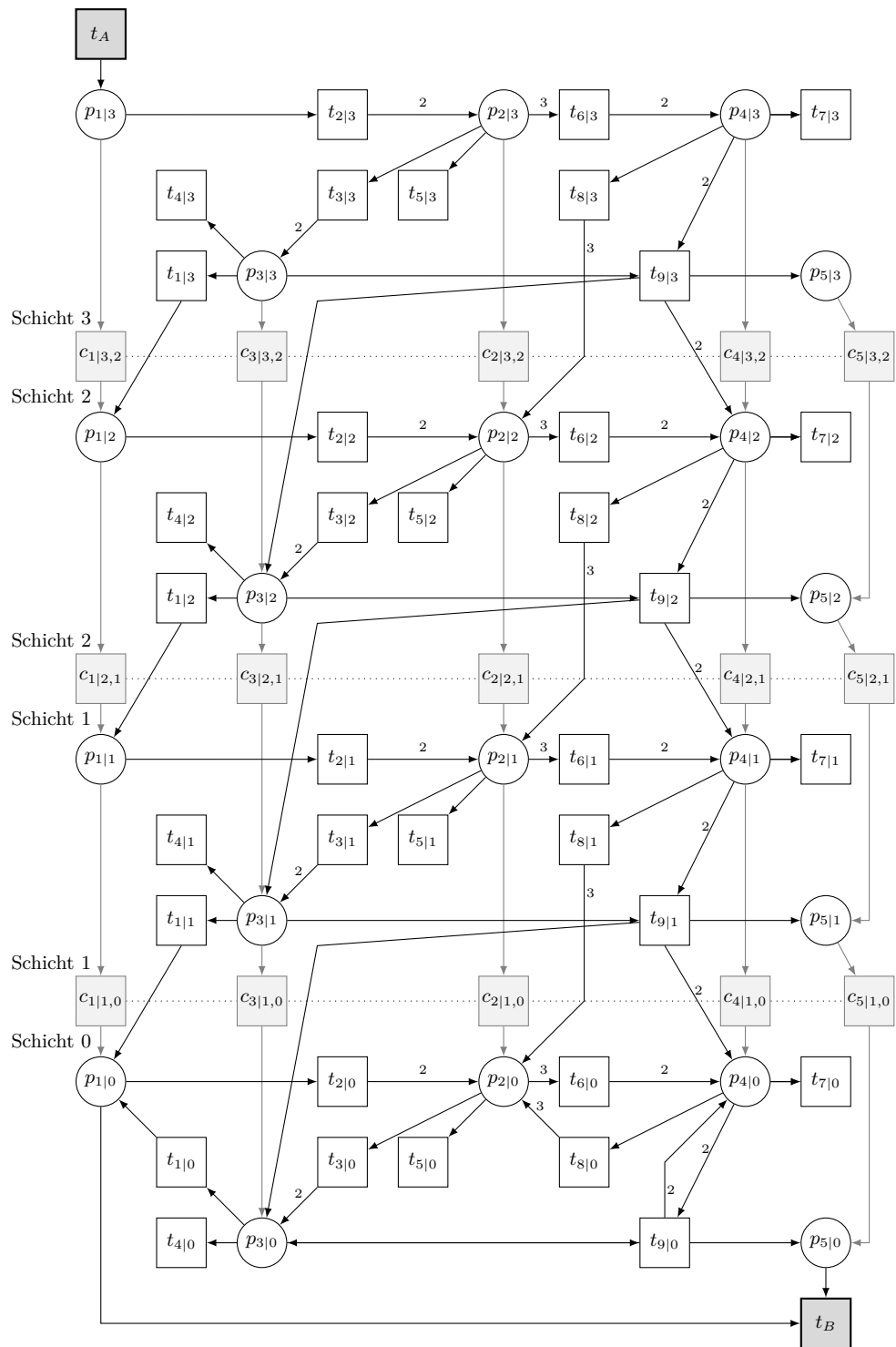


Abbildung 8.16: Netz  $B_{4AB|3}$

## 8.2 Beispiele für schwache Erreichbarkeit

Zum besseren Vergleich mit den Beispielen aus dem vorherigen Abschnitt handelt es sich bei den Netzen in diesem Kapitel um geringfügig veränderte Kopien der Netze B1, B3 und B4:

B5: Dieses Netz wurde in Kapitel 7 zur Erklärung des Lösungsansatzes für schwach erreichbare Fälle verwendet.

B6: Das Netz  $B6_{AB}$  unterscheidet sich von Netz  $B3_{AB}$  nur durch eine andere, durch  $t_A$  und  $t_B$  repräsentierte, Start- und Zielmarkierung.

B7: Hierbei handelt es sich um eine Kopie des Netzes B4, in der lediglich das Gewicht der Kante  $(p_1, t_2)$  verändert wurde.

Insbesondere in den Netzen  $B6_{AB}$  und  $B7_{AB}$  wurden folglich nur Kantengewichte verändert, so dass die Listen der aufzuschneidenden Kanten den jeweils ermittelten Listen für die Netze  $B3_{AB}$  und  $B4_{AB}$  aus dem vorherigen Abschnitt entsprechen, da die minimalen Kreise im Netz identisch sind. Die einzelnen Rechenschritte sind aus diesem Grund nicht erneut aufgeführt.

### 8.2.1 Netz B5

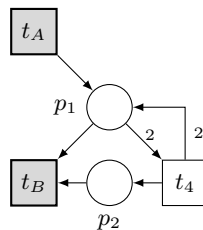


Abbildung 8.17: Netz  $B5_{AB}$

Das Netz in Abbildung 8.17 besitzt die minimale T-Invariante:

$$\begin{array}{c|cccc} & t_1 & t_4 & t_A & t_B \\ \hline \vec{r} & 1 & 1 & 1 & 1 \end{array}$$

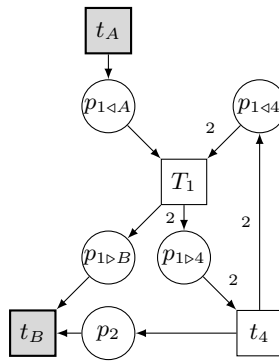
Die Netzdarstellung von  $\vec{r}$  hat keine Deadlocks und Traps und schwache Erreichbarkeit ist damit nachgewiesen.

**Ermitteln der minimalen Kreise**

Im nächsten Schritt wird mit Algorithmus 5.3 der erweiterte Synchronisationsgraph zu  $B5_{AB}$ , dargestellt in Abbildung 8.18, erzeugt und dessen S-Invarianten berechnet. Das Netz besitzt genau eine S-Invariante:

$$\vec{i} \begin{array}{c|ccccc} & p_{1\triangleleft A} & p_{1\triangleleft 4} & p_{1\triangleright B} & p_{1\triangleright 4} & p_2 \\ \hline & 0 & 1 & 0 & 1 & 0 \end{array}$$

Sie beschreibt den einzigen minimalen Kreis, die Schlinge  $p_1 \rightarrow t_4 \rightarrow p_1$  in  $B5_{AB}$ . Algorithmus 5.4 ermittelt für diese Schlinge  $K = \{p_{1\triangleleft 4}\}$  und damit die aufzuschneidende Kante  $(t_4, p_1)$ . Ohne die Stelle  $p_{1\triangleleft 4}$  ist der erweiterte Synchronisationsgraph kreisfrei und es kann mit der Schichtenbildung begonnen werden.



**Abbildung 8.18:** Erweiterter Synchronisationsgraph zu Netz  $B5_{AB}$

**Nachweis schwacher Erreichbarkeit im kreisfreien Netz**

Durch Hilfstransitionen  $t'_{4,1}$  und  $t''_{4,1}$  sowie eine Stelle  $s_{4,1}$  wird die Kante  $(t_4, p_1)$  auf Schicht 0 ersetzt. Das transformierte Netz  $B5_{AB}|_0^{cf}$  bildet zusammen mit Netz  $B5$ ,  $K = \{p_{1\triangleleft 4}\}$  und  $h = 100$  die Eingabe für Algorithmus 7.3 von Seite 115. Es werden zunächst folgende Zielfunktion und Nebenbedingungen erzeugt:

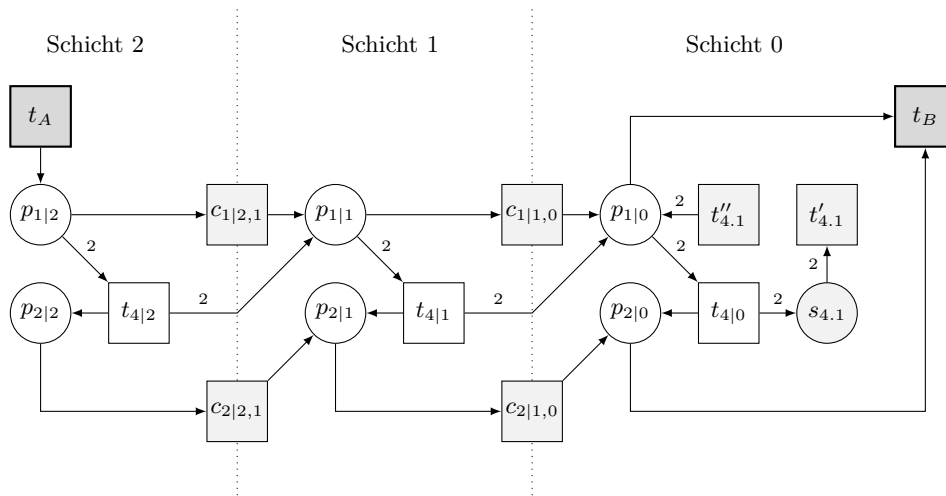
$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}(t'_{4,1}) + 100 \cdot \vec{r}(t''_{4,1}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } & \{ \vec{r}(t_A) = \vec{r}(t_B) \geq 1, \vec{r}(t'_{4,1}) = \vec{r}(t''_{4,1}) \} \end{aligned}$$

GLOP berechnet eine Lösung für  $B5_{AB}|_0^{cf}$ , in der die Hilfstransitionen  $t'_{4,1}$  und  $t''_{4,1}$  noch schalten. Der Zielfunktionswert `neuer_zfwert` beträgt 202. Danach wird mit der Schichtenbildung begonnen und für das neu erzeugte Netz  $B5_{AB}|_1^{cf}$  das gleiche Ergebnis erzielt. Nach Hinzufügen der zweiten Schicht ist in Netz  $B5_{AB}|_2^{cf}$  schwache Erreichbarkeit nachgewiesen: keine Hilfstransition schaltet und  $t_A$  und  $t_B$  feuern je zweimal mit

neuer\_zfwert = 4. Nun muss, wenn starke Erreichbarkeit vorliegen soll, durch eine weitere Schicht auch ein geringerer Zielfunktionswert erzielt werden, da das Netz durch die zusätzliche Ebene den erneuten Durchlauf eines Kreises „dazugewinnt“ und damit die Chance hat, Marken selbst zu produzieren. Jedoch ist dies für  $B5_{AB}|_3^{cf}$  nicht der Fall: Die realisierbare T-Invariante enthält nach wie vor zweimaliges Schalten von  $t_A$  und  $t_B$ . Eine zusätzliche Schicht war also nutzlos, da keine weiteren Marken ohne Hilfe von  $t_A$  erzeugt werden können. Es liegt also keine starke Erreichbarkeit vor. In Tabelle 8.9 sind die Ergebnisse zusammengefasst.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B5_{AB} _0^{cf}$	202	$t_A, t''_{4.1}, t_{4 0}, t'_{4.1}, t_B$
$B5_{AB} _1^{cf}$	202	$t_A, c_{1 1,0}, t''_{4.1}, t_{4 0}, t'_{4.1}, t_B$
$B5_{AB} _2^{cf}$	4	$2 \cdot t_A, t_{4 2}, c_{2 2,1}, t_{4 1}, 2 \cdot c_{2 1,0}, 2 \cdot t_B$
$B5_{AB} _3^{cf}$	4	$2 \cdot t_A, t_{4 3}, c_{2 3,2}, t_{4 2}, 2 \cdot c_{2 2,1}, 2 \cdot c_{1 1,0}, 2 \cdot c_{2 1,0}, 2 \cdot t_B$

**Tabelle 8.9:** Lösungen des GLOP für das schichtweise erweiterte Netz  $B5_{AB}^{cf}$



**Abbildung 8.19:** Netz  $B5_{AB}|_2^{cf}$

### Nachweis schwacher Erreichbarkeit mit Kreisen auf Schicht 0

Die Kante  $(t_4, p_1)$  bleibt in diesem Fall auf Schicht 0 erhalten.  $B5_{AB}|_0$  bildet zusammen mit  $B5$ ,  $K = \{p_{1..4}\}$  und der gleichen Konstante  $h = 100$  die Eingabe für Algorithmus

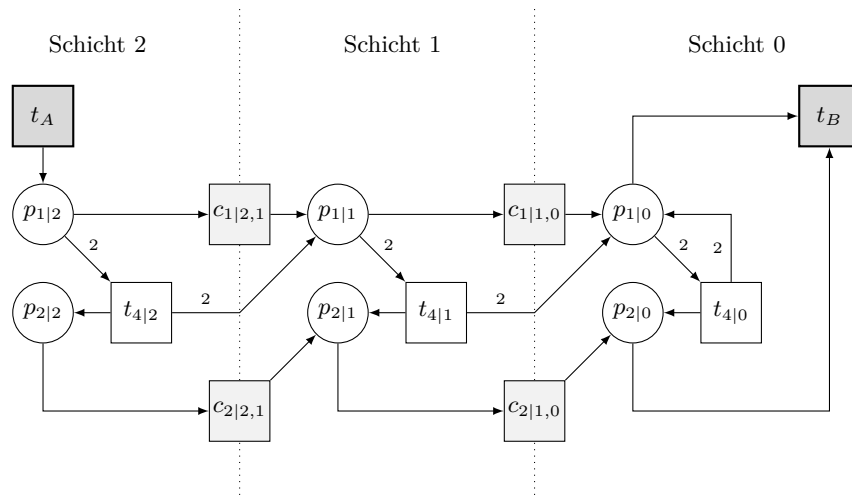
7.5 von Seite 121. Es werden zunächst wieder Zielfunktion und Nebenbedingungen erzeugt:

$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}(t_{4|0}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } & \{ \vec{r}(t_A) = \vec{r}(t_B) \geq 1 \} \end{aligned}$$

GLOP ermittelt eine Lösung für  $B5_{AB|0}$ , in der  $t_{4|0}$  schaltet und der Zielfunktionswert 102 beträgt. Die T-Invariante ist jedoch nicht realisierbar. Erst nach zwei weiteren Schichten wird in  $B5_{AB|2}$  schwache Erreichbarkeit nachgewiesen. Noch eine weitere Schicht ändert — wie im kreisfreien Netz — nichts am Zielfunktionswert: Die zusätzliche Ebene ist nutzlos und es liegt keine starke Erreichbarkeit vor. Die Ergebnisse sind in Tabelle 8.10 zu finden.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B5_{AB 0}$	102	$t_A, t_{4 0}, t_B$ (nicht realisierbar)
$B5_{AB 1}$	102	$t_A, c_{1 1,0}, t_{4 0}, t_B$ (nicht realisierbar)
$B5_{AB 2}$	4	$2 \cdot t_A, t_{4 2}, c_{2 2,1}, t_{4 1}, 2 \cdot c_{2 1,0}, 2 \cdot t_B$
$B5_{AB 3}$	4	$2 \cdot t_A, t_{4 3}, c_{2 3,2}, t_{4 2}, 2 \cdot c_{2 2,1}, 2 \cdot c_{1 1,0}, 2 \cdot c_{2 1,0}, 2 \cdot t_B$

**Tabelle 8.10:** Lösungen des GLOP für das schichtweise erweiterte Netz  $B5_{AB}$  mit Kreisen auf Schicht 0



**Abbildung 8.20:** Netz  $B5_{AB|2}$

### 8.2.2 Netz B6

Das in Abbildung 8.21 dargestellte Netz  $B6_{AB}$  unterscheidet sich von dem S/T-Netz  $B3_{AB}$  von Seite 135 nur durch eine andere, durch  $t_A$  und  $t_B$  repräsentierte, Start- und Zielmarkierung  $M6_A$  und  $M6_B$ . Warum für dieses Netz  $B6_{AB}$  unter der gegebenen Markierung keine starke Erreichbarkeit vorliegen kann, ist bereits nach ersten Simulationsschritten erkennbar: Die Transitionen  $t_1$  und  $t_3$  konkurrieren um Marken auf  $p_1$ . Darf  $t_A$  nur *einmal* schalten, so wird diese Marke beispielsweise von  $t_3$  konsumiert und auf  $p_3$  gelegt. Dann ist das Netz tot, da  $t_1$  zum Schalten wiederum eine Marke auf  $p_1$  fehlt. Nur durch zweimaliges Feuern von  $t_A$  kann diese Konfliktsituation verhindert werden. Die minimalen T-Invarianten von Netz  $B6_{AB}$  entsprechen den T-Invarianten in Netz  $B3_{AB}$  auf Seite 135, schwache Erreichbarkeit ist demnach gegeben.

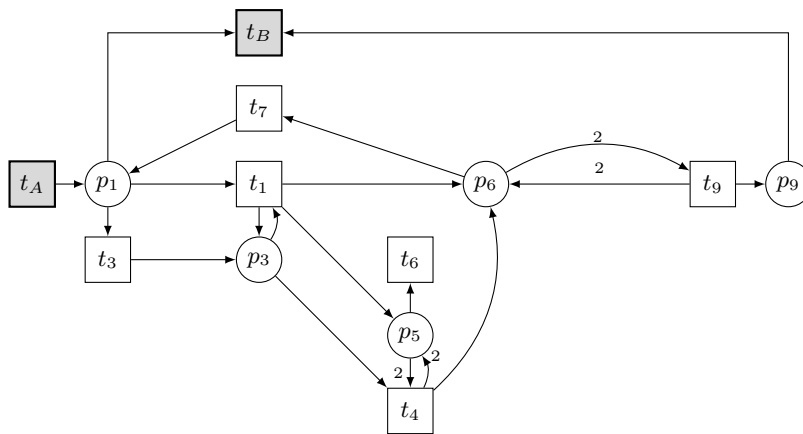


Abbildung 8.21: Netz  $B6_{AB}$

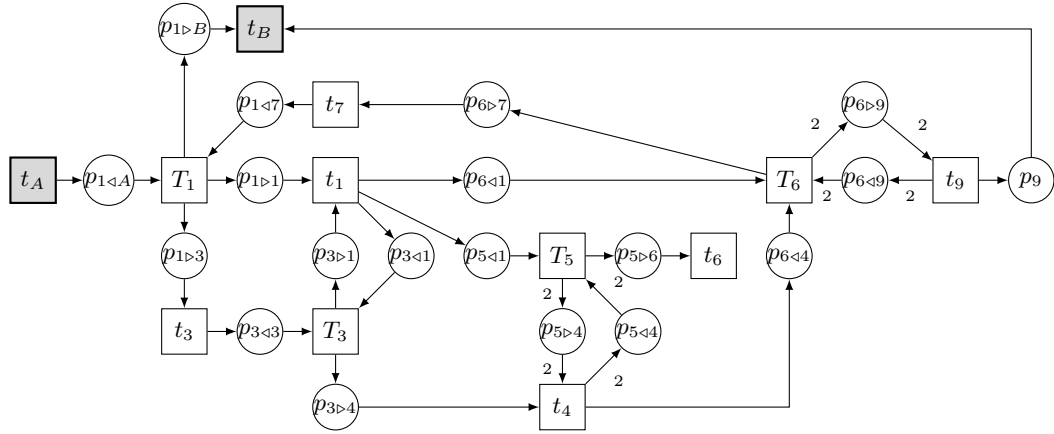
#### Ermitteln der minimalen Kreise

Da bis auf die modellierten Markierungen nichts an der Struktur des Netzes geändert wurde, können auch die aufzuschneidenden Kanten, repräsentiert durch die Liste  $K = \{p6 \prec 9, p5 \prec 4, p3 \prec 1, p1 \prec 7\}$ , übernommen werden.

#### Nachweis schwacher Erreichbarkeit im kreisfreien Netz

Die vier aufzuschneidenden Kanten werden auf Schicht 0 durch acht Hilfstransitionen sowie vier Stellen ersetzt. Algorithmus 7.3 erhält Netz  $B6_{AB}|_0^{cf}$ ,  $B6$ ,  $K$  und die Konstante  $h = 100$  als Eingabe. Zielfunktion und Nebenbedingungen werden konstruiert:

$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}'(t'_{9,6}) + 100 \cdot \vec{r}''(t''_{9,6}) + 100 \cdot \vec{r}'(t'_{4,5}) + 100 \cdot \vec{r}''(t''_{4,5}) \\ & + 100 \cdot \vec{r}'(t'_{1,3}) + 100 \cdot \vec{r}''(t''_{1,3}) + 100 \cdot \vec{r}'(t'_{7,1}) + 100 \cdot \vec{r}''(t''_{7,1}) + \vec{r}'(t_A) + \vec{r}'(t_B) \rightarrow \text{Min} \end{aligned}$$


 Abbildung 8.22: Erweiterter Synchronisationsgraph zu Netz  $B6_{AB}$ 

$$\text{cons: } \{\bar{r}(t_A) = \bar{r}(t_B) \geq 1, \bar{r}(t'_{9,6}) = \bar{r}(t''_{9,6}), \bar{r}(t'_{4,5}) = \bar{r}(t''_{4,5}), \bar{r}(t'_{1,3}) = \bar{r}(t''_{1,3}), \\ \bar{r}(t'_{7,1}) = \bar{r}(t''_{7,1})\}$$

GLOP ermittelt danach eine Lösung für  $B6_{AB}|_0^{cf}$ , in der die Hilfstransitionen  $t'_{9,6}$  und  $t''_{9,6}$  schalten, was einem Zielfunktionswert von 202 entspricht. Anschließend wird auf schwache Erreichbarkeit untersucht. Erst mit der vierten zusätzlichen Schicht ist diese nachweisbar;  $t_A$  und  $t_B$  müssen je viermal feuern. Nun muss kontrolliert werden, ob der Zielfunktionswert von nun an streng monoton fällt. In der Tat ist dies mit einer weiteren Schicht der Fall: In  $B6_{AB}|_5^{cf}$  schalten beide Transitionen nur noch zweimal. Nun folgt mit der nächsten Schicht der direkte Test auf starke Erreichbarkeit, da der nächste Zielfunktionswert  $< 4$  nur einmaliges Schalten von  $t_A$  und  $t_B$  repräsentieren würde. In  $B6_{AB}|_6^{cf}$  wird jedoch nur eine Lösung mit Zielfunktionswert 4 gefunden, so dass die zusätzliche Schicht keinen weiteren Nutzen gebracht hat. Es liegt daher nur schwache, aber keine starke Erreichbarkeit vor. In Tabelle 8.11 sind alle Ergebnisse abgebildet.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\bar{r}$ )
$B6_{AB} _0^{cf}$	202	$t_A, t''_{9,6}, t_{9 0}, t'_{9,6}, t_B$
$B6_{AB} _1^{cf}$	202	$t_A, c_{1 1,0}, t''_{9,6}, t_{9 0}, t'_{9,6}, t_B$
$B6_{AB} _2^{cf}$	202	$t_A, c_{1 2,1}, c_{1 1,0}, t''_{9,6}, t_{9 0}, t'_{9,6}, t_B$
$B6_{AB} _3^{cf}$	202	$t_A, c_{1 3,2}, c_{1 2,1}, c_{1 1,0}, t''_{9,6}, t_{9 0}, t'_{9,6}, t_B$
$B6_{AB} _4^{cf}$	8	$4 \cdot t_A, 2 \cdot t_{3 4}, 2 \cdot t_{1 4}, t_{9 4}, 2 \cdot c_{5 4,3}, c_{9 4,3}, t_{4 3}, t_{9 3}, \\ c_{3 3,2}, c_{6 3,2}, 2 \cdot c_{9 3,2}, t_{4 2}, 2 \cdot t_{9 2}, 4 \cdot c_{9 2,1}, t_{6 1}, 4 \cdot t_{7 1}, \\ c_{5 1,0}, 4 \cdot c_{9 1,0}, t_{6 0}, 4 \cdot t_B$

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B6_{AB} _5^{cf}$	4	$2 \cdot t_A, t_{3 5}, t_{1 5}, t_{7 5}, c_{5 5,4}, t_{1 4}, 2 \cdot c_{5 4,3}, c_{6 4,3}, t_{4 3}, t_{9 3}, c_{9 3,2}, 2 \cdot t_{6 2}, t_{9 2}, 2 \cdot c_{9 2,1}, 2 \cdot t_{7 1}, 2 \cdot c_{9 1,0}, 2 \cdot t_B$
$B6_{AB} _6^{cf}$	4	$2 \cdot t_A, t_{3 6}, t_{1 6}, t_{7 6}, c_{5 6,5}, t_{1 5}, 2 \cdot c_{5 5,4}, c_{6 5,4}, t_{4 4}, t_{9 4}, c_{9 4,3}, 2 \cdot t_{6 3}, t_{9 3}, 2 \cdot c_{9 3,2}, 2 \cdot t_{7 2}, 2 \cdot c_{9 2,1}, 2 \cdot c_{1 1,0}, 2 \cdot c_{9 1,0}, 2 \cdot t_B$

Tabelle 8.11: Lösungen des GLOP für das schichtweise erweiterte Netz  $B6_{AB}^{cf}$ **Nachweis schwacher Erreichbarkeit mit Kreisen auf Schicht 0**

Netz  $B6_{AB}|_0$  bildet zusammen mit  $B6$ ,  $K$  und Konstante  $h = 100$  die Eingabe für Algorithmus 7.5. Es werden folgende Zielfunktion und Nebenbedingungen erzeugt:

$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}(t_{1|0}) + 100 \cdot \vec{r}(t_{3|0}) + 100 \cdot \vec{r}(t_{4|0}) + 100 \cdot \vec{r}(t_{6|0}) + 100 \cdot \vec{r}(t_{7|0}) \\ & + 100 \cdot \vec{r}(t_{9|0}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } & \{\vec{r}(t_A) = \vec{r}(t_B) \geq 1\} \end{aligned}$$

Für  $B6_{AB}|_0$  ist keine schwache Erreichbarkeit nachweisbar. Der Algorithmus fügt zunächst drei zusätzliche Schichten bei einem konstant bleibenden Zielfunktionswert von 102 hinzu. Die Lösung, die in diesen Fällen einen minimalen Wert liefert, entspricht mit dem Schalten von  $t_A$ ,  $t_{9|0}$  und  $t_B$  (und gegebenenfalls Hilfstransitionen  $c_i$ ) einer der drei minimalen T-Invarianten des Originalnetzes  $B6_{AB}$ , die allein nicht realisierbar ist. Erst mit der vierten zusätzlichen Schicht wird in  $B6_{AB}|_4$  eine T-Invariante  $\vec{r}$  gefunden, deren Netzdarstellung kreisfrei ist und die schwache Erreichbarkeit nachweist:  $\vec{r}(t_A) = \vec{r}(t_B) = 4$ .

Nun wird auf starke Erreichbarkeit untersucht: Mit der nächsten Schicht sinkt der Zielfunktionswert streng monoton,  $t_A$  und  $t_B$  schalten in der Lösung für  $B6_{AB}|_5$  je zweimal. Allerdings ist damit auch das Ziel erreicht: Die sechste Ebene bringt keinen weiteren Nutzen. Der Lösungsvektor für  $B6_{AB}|_6$  zeigt, dass ohne zweimaliges Schalten von  $t_A$  die leere Markierung nicht reproduzierbar ist. Die Ausgaben des Algorithmus für jede Schicht sind in Tabelle 8.12 zusammengestellt.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B6_{AB} _0$	102	$t_A, t_{9 0}, t_B$ (nicht realisierbar)
$B6_{AB} _1$	102	$t_A, c_{1 1,0}, t_{9 0}, t_B$ (nicht realisierbar)
$B6_{AB} _2$	102	$t_A, c_{1 2,1}, c_{1 1,0}, t_{9 0}, t_B$ (nicht realisierbar)



Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B6_{AB} _3$	102	$t_A, c_{1 3,2}, c_{1 2,1}, c_{1 1,0}, t_{9 0}, t_B$ (nicht realisierbar)
$B6_{AB} _4$	8	$4 \cdot t_A, 2 \cdot t_{3 4}, 2 \cdot t_{1 4}, t_{9 4}, 2 \cdot c_{5 4,3}, c_{9 4,3}, t_{4 3}, t_{9 3}, c_{3 3,2}, c_{6 3,2}, 2 \cdot c_{9 3,2}, t_{4 2}, 2 \cdot t_{9 2}, 4 \cdot c_{9 2,1}, 2 \cdot t_{6 1}, 4 \cdot t_{7 1}, 4 \cdot c_{9 1,0}, 4 \cdot t_B$
$B6_{AB} _5$	4	$2 \cdot t_A, t_{3 5}, t_{1 5}, t_{7 5}, c_{5 5,4}, t_{1 4}, 2 \cdot c_{5 4,3}, c_{6 4,3}, t_{4 3}, t_{9 3}, c_{9 3,2}, 2 \cdot t_{6 2}, t_{9 2}, 2 \cdot c_{9 2,1}, 2 \cdot t_{7 1}, 2 \cdot c_{9 1,0}, 2 \cdot t_B$
$B6_{AB} _6$	4	$2 \cdot t_A, t_{3 6}, t_{1 6}, t_{7 6}, c_{5 6,5}, t_{1 5}, 2 \cdot c_{5 5,4}, c_{6 5,4}, t_{4 4}, t_{9 4}, c_{9 4,3}, 2 \cdot t_{6 3}, 2 \cdot c_{6 3,2}, c_{9 3,2}, t_{9 2}, 2 \cdot c_{9 2,1}, 2 \cdot t_{7 1}, 2 \cdot c_{9 1,0}, 2 \cdot t_B$

**Tabelle 8.12:** Lösungen des GLOP für das schichtweise erweiterte Netz  $B6_{AB}$  mit Kreisen auf Schicht 0

### 8.2.3 Netz B7

Bei  $B7_{AB}$  handelt es sich um Netz  $B4_{AB}$  mit einer einzigen kleinen Veränderung: Das Kantengewicht  $W((p_1, t_2))$  wurde von 1 auf 2 erhöht. Es ist schnell ersichtlich, dass in keinem Fall starke Erreichbarkeit gegeben ist: Darf  $t_A$  nur einmal schalten, so ist das Netz danach bereits tot. Erweitert zu  $B7_{ABC}$  besitzt es folgende minimale T-Invarianten:

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_A$	$t_B$
$\vec{r}^1$	0	0	0	0	0	0	0	0	1	1	1
$\vec{r}^2$	2	1	1	0	1	0	0	0	0	0	0
$\vec{r}^3$	0	0	0	0	3	1	0	2	0	0	0
$\vec{r}^4$	2	1	2	2	0	0	0	0	0	0	0
$\vec{r}^5$	0	0	3	6	0	1	0	2	0	0	0
$\vec{r}^6$	6	3	3	0	0	1	2	0	0	0	0
$\vec{r}^7$	0	0	0	0	0	1	1	1	0	0	0

Die Netzdarstellung von  $\vec{r}^1$  ist nicht frei von Deadlocks und Traps ist, somit müssen andere T-Invarianten hinzuaddiert werden. Die Summe  $\vec{r}^1 + \vec{r}^2 + \vec{r}^3$  bildet wie in  $B4_{AB}$  eine Linearkombination, die schwache Erreichbarkeit nachweist.

#### Ermitteln der minimalen Kreise

Die für  $B7_{AB}$  zu ermittelnde Liste  $K$  entspricht der Liste für  $B4_{AB}$ . Es ist somit  $K = (p_{3 \leftarrow 9}, p_{4 \leftarrow 9}, p_{1 \leftarrow 1}, p_{2 \leftarrow 8})$ , was den Kanten  $(t_9, p_3)$ ,  $(t_9, p_4)$ ,  $(t_1, p_1)$  und  $(t_8, p_2)$  entspricht.

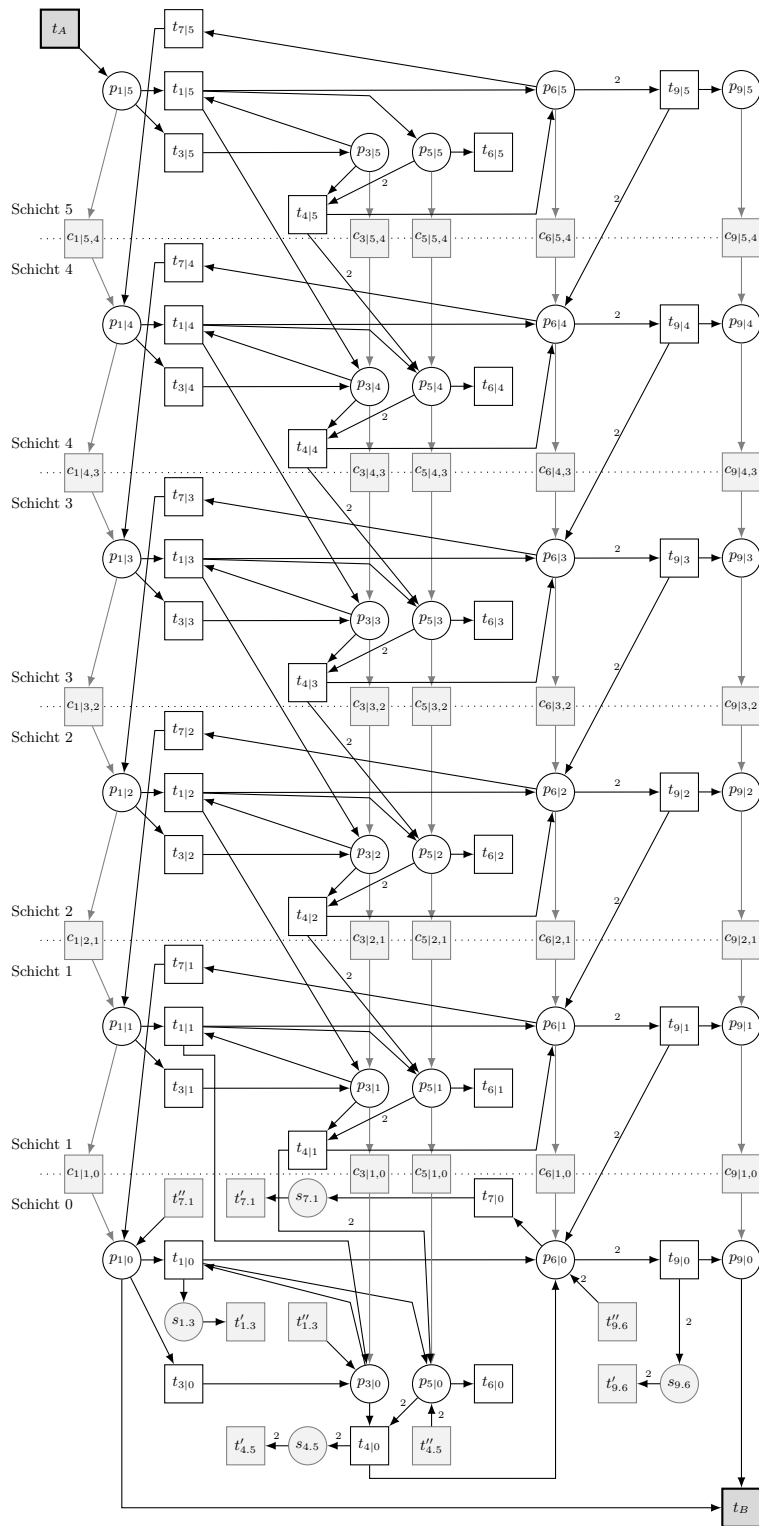


Abbildung 8.23: Netz  $B_{6AB|5}^{cf}$

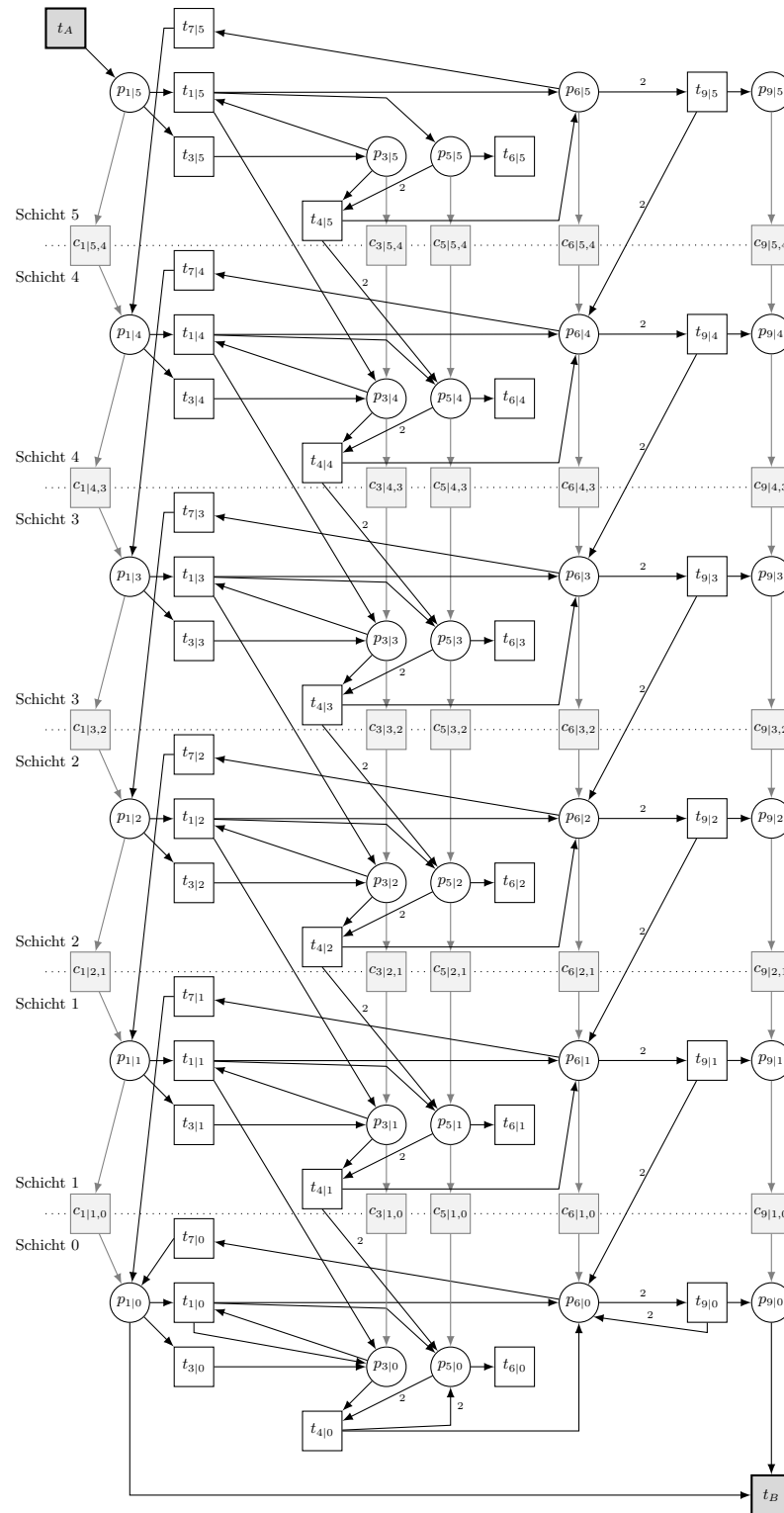


Abbildung 8.24: Netz  $B6_{AB|5}$

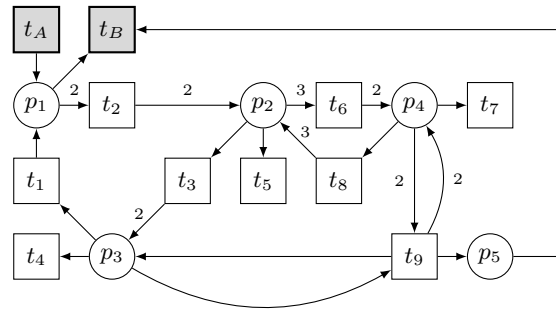


Abbildung 8.25: Netz B7<sub>AB</sub>

**Nachweis schwacher Erreichbarkeit im kreisfreien Netz**

Die vier aufzuschneidenden Kanten werden durch die acht Hilfstransitionen und vier Stellen ersetzt. Algorithmus 7.3 erwartet als Eingabe die Netze  $B7_{AB}|_0^{cf}$  und  $B7$  sowie die Liste  $K$  und die Konstante  $h = 100$ . Zielfunktion und Nebenbedingungen werden wie gewohnt konstruiert:

$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}(t'_{9,3}) + 100 \cdot \vec{r}(t''_{9,3}) + 100 \cdot \vec{r}(t'_{9,4}) + 100 \cdot \vec{r}(t''_{9,4}) \\ & + 100 \cdot \vec{r}(t'_{1,1}) + 100 \cdot \vec{r}(t''_{1,1}) + 100 \cdot \vec{r}(t'_{8,2}) + 100 \cdot \vec{r}(t''_{8,2}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } & \{ \vec{r}(t_A) = \vec{r}(t_B) \geq 1, \vec{r}(t'_{9,3}) = \vec{r}(t''_{9,3}), \vec{r}(t'_{9,4}) = \vec{r}(t''_{9,4}), \vec{r}(t'_{1,1}) = \vec{r}(t''_{1,1}), \\ & \vec{r}(t'_{8,2}) = \vec{r}(t''_{8,2}) \} \end{aligned}$$

GLOP ermittelt zunächst eine Lösung für  $B7_{AB}|_0^{cf}$ , in der vier Hilfstransitionen schalten, was einem Zielfunktionswert von 402 entspricht. Der Zielfunktionswert ändert sich erst mit der dritten zusätzlichen Schicht: Die T-Invariante  $\vec{r}$  in  $B7_{AB}|_3^{cf}$ , in der  $t_A$  und  $t_B$  zweimal schalten, weist schwache Erreichbarkeit nach. Nun müsste mit der nächsten zusätzlich Schicht der Zielfunktionswert auf 2 fallen, wenn starke Erreichbarkeit gegeben ist. Dies ist aber nicht der Fall: Der Zielfunktionswert bleibt identisch, die neue Schicht war also nutzlos und es liegt nur schwache Erreichbarkeit vor. In Tabelle 8.13 sind alle Ergebnisse zu finden.

Netz	ZF-Wert	Schaltfolge der Lösung (T-Invariante $\vec{r}$ )
$B7_{AB} _0^{cf}$	402	$t_A, t''_{9,3}, t'_{9,4}, t_{9 0}, t'_{9,3}, t'_{9,4}, t_B$
$B7_{AB} _1^{cf}$	402	$t_A, c_{1 1,0}, t''_{9,3}, t'_{9,4}, t_{9 0}, t'_{9,3}, t'_{9,4}, t_B$
$B7_{AB} _2^{cf}$	402	$t_A, c_{1 2,1}, c_{1 1,0}, t''_{9,3}, t'_{9,4}, t_{9 0}, t'_{9,3}, t'_{9,4}, t_B$
$B7_{AB} _3^{cf}$	4	$2 \cdot t_A, t_{2 3}, 2 \cdot t_{3 3}, 4 \cdot t_{1 3}, 2 \cdot t_{2 2}, 4 \cdot t_{3 2}, 8 \cdot t_{1 2}, 4 \cdot t_{2 1}, 2 \cdot t_{3 1}, 2 \cdot t_{6 1}, 2 \cdot t_{9 1}, 2 \cdot t_{1 1}, 2 \cdot c_{5 1,0}, 2 \cdot t_{4 0}, 4 \cdot t_{7 0}, 2 \cdot t_B$

Netz	ZF-Wert	Schaltfolge der Lösung (T–Invariante $\vec{r}$ )
$B7_{AB} _4^{cf}$	4	$2 \cdot t_A, t_{2 4}, t_{3 4}, 2 \cdot t_{1 4}, c_{2 4,3}, t_{2 3}, t_{6 3}, t_{8 3}, c_{4 3,2}, t_{6 2}, 3 \cdot t_{8 2}, 2 \cdot t_{3 1}, t_{5 1}, 2 \cdot t_{6 1}, 2 \cdot t_{9 1}, 2 \cdot t_{1 1}, 2 \cdot c_{5 1,0}, 2 \cdot t_{4 0}, 4 \cdot t_{7 0}, 2 \cdot t_B$

Tabelle 8.13: Lösungen des GLOP für das schichtweise erweiterte Netz  $B7_{AB}^{cf}$ **Nachweis schwacher Erreichbarkeit mit Kreisen auf Schicht 0**

Netz  $B7_{AB}|_0$  wird zusammen mit  $B7$ ,  $K$  und Konstante  $h = 100$  Algorithmus 7.5 übergeben. Zunächst werden folgende Zielfunktion und Nebenbedingungen erzeugt:

$$\begin{aligned} \text{ZF: } & 100 \cdot \vec{r}(t_{1|0}) + 100 \cdot \vec{r}(t_{2|0}) + 100 \cdot \vec{r}(t_{3|0}) + 100 \cdot \vec{r}(t_{4|0}) + 100 \cdot \vec{r}(t_{5|0}) \\ & + 100 \cdot \vec{r}(t_{6|0}) + 100 \cdot \vec{r}(t_{7|0}) + 100 \cdot \vec{r}(t_{8|0}) + 100 \cdot \vec{r}(t_{9|0}) + \vec{r}(t_A) + \vec{r}(t_B) \rightarrow \text{Min} \\ \text{cons: } & \{\vec{r}(t_A) = \vec{r}(t_B) \geq 1\} \end{aligned}$$

Der Algorithmus fügt drei zusätzliche Schichten hinzu, mit denen sich der ermittelte Zielfunktionswert von 102 nicht ändert. Der Lösungsvektor ist nicht realisierbar und entspricht einer minimalen T–Invariante von  $B7_{AB}$ , in der die Transition  $t_{9|0}$  schalten müsste, dies aber nicht kann, da ihr die Marken auf  $p_{3|0}$  und  $p_{4|0}$  fehlen. Hinzufügen der vierten Schicht führt zu einer realisierbaren T–Invariante, die schwache Erreichbarkeit nachweist. Mit der fünften Schicht ändert sich der Zielfunktionswert nicht mehr, es liegt also keine starke Erreichbarkeit vor. Die Ergebnisse zeigt Tabelle 8.14.

Netz	ZF-Wert	Schaltfolge der Lösung (T–Invariante $\vec{r}$ )
$B7_{AB} _0$	102	$t_A, t_{9 0}, t_B$ (nicht realisierbar)
$B7_{AB} _1$	102	$t_A, c_{1 1,0}, t_{9 0}, t_B$ (nicht realisierbar)
$B7_{AB} _2$	102	$t_A, c_{1 2,1}, c_{1 1,0}, t_{9 0}, t_B$ (nicht realisierbar)
$B7_{AB} _3$	102	$t_A, c_{1 3,2}, c_{1 2,1}, c_{1 1,0}, t_{9 0}, t_B$ (nicht realisierbar)
$B7_{AB} _4$	4	$2 \cdot t_A, t_{2 4}, 2 \cdot t_{3 4}, 4 \cdot t_{1 4}, 2 \cdot t_{2 3}, t_{3 3}, t_{6 3}, 2 \cdot t_{8 3}, 2 \cdot c_{3 3,2}, 2 \cdot t_{6 2}, 2 \cdot t_{9 2}, 2 \cdot c_{5 2,1}, 2 \cdot t_{1 1}, 4 \cdot t_{7 1}, 2 \cdot c_{5 1,0}, 2 \cdot t_B$
$B7_{AB} _5$	4	$2 \cdot t_A, t_{2 5}, t_{3 5}, 2 \cdot t_{1 5}, c_{2 5,4}, t_{2 4}, t_{6 4}, t_{8 4}, c_{4 4,3}, t_{6 3}, 3 \cdot t_{8 3}, t_{3 2}, 2 \cdot t_{6 2}, 2 \cdot t_{9 2}, 2 \cdot c_{2 2,1}, 2 \cdot c_{5 2,1}, 2 \cdot t_{5 1}, 4 \cdot t_{7 1}, 2 \cdot t_{1 1}, 2 \cdot c_{5 1,0}, 2 \cdot t_B$

Tabelle 8.14: Lösungen des GLOP für das schichtweise erweiterte Netz  $B7_{AB}$  mit Kreisen auf Schicht 0

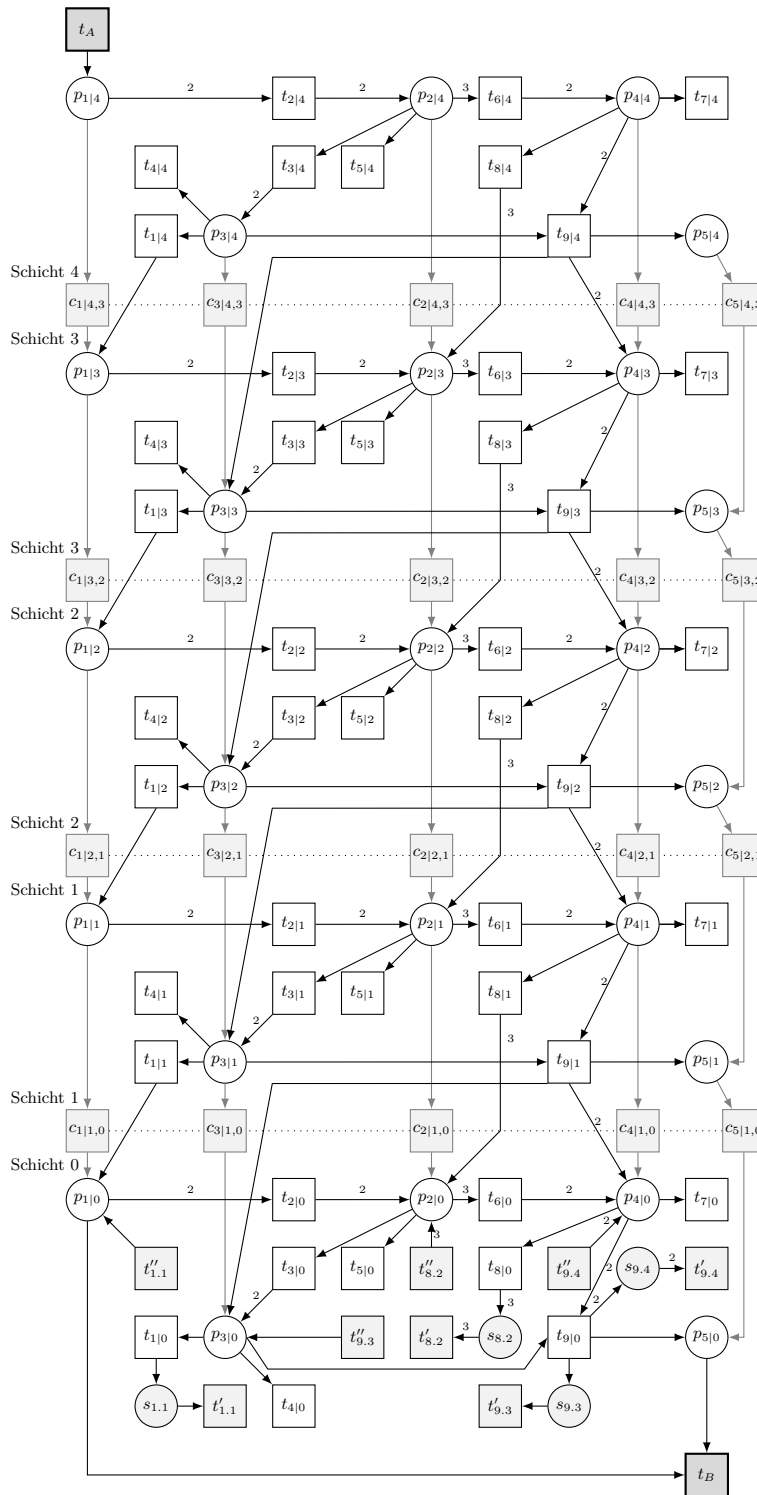


Abbildung 8.26: Netz  $B7_{AB}|_4^{cf}$

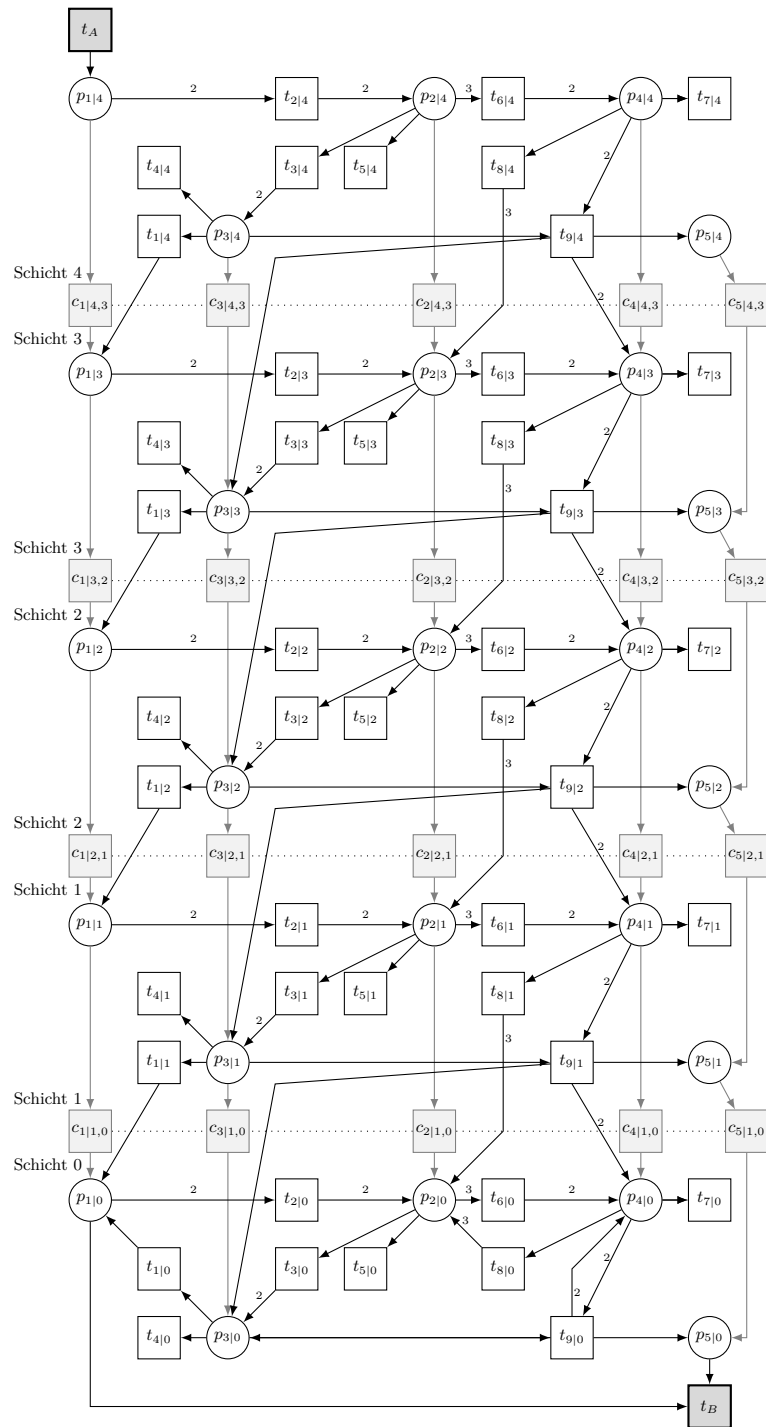


Abbildung 8.27: Netz  $B7_{AB|4}$





## 9 Fazit und Ausblick

Im Rahmen der vorliegenden Arbeit wurde untersucht, ob es rein linear-algebraische Verfahren zum Nachweis der Erreichbarkeit einer Markierung  $B$  aus einer Markierung  $A$  in einem beliebigen Stellen/Transitions-Netz gibt. Obwohl die lineare Algebra ein sehr übersichtliches und vielseitig einsetzbares Handwerkszeug ist, stellen Kreise in Netzen ein Problem für sie dar: In einem Schalthäufigkeitsvektor ist nicht erkennbar, in welcher Reihenfolge dessen Transitionen schalten müssen, um  $B$  aus  $A$  zu erreichen. Aus diesem Grund kann ein solcher Vektor in einem Netz mit Kreisen zwar die Zustandsgleichung erfüllen, jedoch ist nicht erkennbar, ob für bestimmte Transitionen erforderliche Marken sofort, erst nach dem Durchlaufen bestimmter Kreise oder gar nicht zur Verfügung stehen. Es gibt verschiedene Möglichkeiten, dieses Problem anzugehen. Zwei in den letzten Jahren entwickelte Verfahren wurden in dieser Arbeit beispielhaft vorgestellt und deren Schwächen aufgezeigt:

*Kostin* (Kapitel 4.1) erweitert das gegebene Netz um eine zusätzliche Transition  $t_c$ , die durch einmaliges Schalten die Startmarkierung  $A$  aus der Zielmarkierung  $B$  erzeugt. Anhand von  $t_c$  klassifiziert Kostin die minimalen T-Invarianten dieses erweiterten Netzes und versucht durch intelligente Bildung von Linearkombinationen von T-Invarianten aus diesen insgesamt drei Klassen einen passenden Schalthäufigkeitsvektor zu konstruieren. Intelligent heißt hierbei, dass auch der notwendige Markenbedarf jeder T-Invariante für deren Realisierbarkeit ermittelt wird und darauf aufbauend abgeschätzt wird, wie oft andere T-Invarianten, die diese Marken zur Verfügung stellen können, in die Linearkombination einfließen müssen. Letztlich versucht Kostin mit dieser Linearkombination durch Simulationsschritte nachzuweisen, dass eine Schaltsequenz von  $A$  nach  $B$  und durch einmaliges Schalten von  $t_c$  wieder zurück nach  $A$  existiert. Der Schwachpunkt von Kostins Verfahren liegt eindeutig in der Vielzahl der Kombinationsmöglichkeiten bei der Bildung von Linearkombinationen und der Tatsache, schlussendlich *doch* auf Simulationsschritte zurückgreifen zu müssen.

*Ramachandran und Kamath* (Kapitel 4.2) wählten einen anderen Ansatz: Bei einem kreisfreien Netz ist die erfüllte Zustandsgleichung kein notwendiges, sondern ein hin-

reichendes Kriterium. Ein gegebenes S/T-Netz wird darum so in kreisfreie Schichten entfaltet, dass die Zustandsgleichung als Nachweis für Erreichbarkeit herangezogen werden kann. Jedoch basiert dieses Verfahren darauf, dass bereits ein Schalthäufigkeitsvektor bekannt ist, der die Gleichung erfüllt. Nur mit dem Wissen, dass maximal  $d$  (als Summe aller Komponenten des Schalthäufigkeitsvektors) Schaltvorgänge ausgeführt werden müssen, besitzt das Verfahren ein Abbruchkriterium: Nach maximal  $d$  zusätzlichen Schichten — da auf jeder Schicht mindestens eine Transition feuern kann — wird die Erreichbarkeit für diesen bestimmten Schalthäufigkeitsvektor entschieden. Falls keine Erreichbarkeit vorliegt, gilt dies aber nur für diesen konkreten Schalthäufigkeitsvektor. Es kann trotzdem noch andere, realisierbare Schaltsequenzen geben, die erneut getestet werden müssen.

Beide Verfahren zeigten deutlich, dass erforderliche Simulationsschritte und fehlende Abbruchkriterien ein Problem darstellen. Es bestand also die Notwendigkeit, neue Methoden zum Umgang mit diesen Defiziten zu entwickeln. Auf Basis der Reproduzierbarkeit der leeren Markierung wurde darum das Erreichbarkeitsproblem in Kapitel 5 nach einer Idee von Kurt Lautenbach in zwei Problemklassen, starke und schwache Erreichbarkeit, verfeinert. Ein gegebenes S/T-Netz wird nun um zwei Transitionen  $t_A$  und  $t_B$  zu einem Netz  $N_{AB}$  ergänzt; durch Schalten von  $t_A$  wird die Markierung  $A$  auf das Netz gelegt und durch Feuern von  $t_B$  die Markierung  $B$  vom Netz genommen. In diesem erweiterten Netz ist die leere Markierung die Startmarkierung. Der Nachweis starker und schwacher Erreichbarkeit wurde folgendermaßen auf die Reproduktion der leeren Markierung projiziert:

Im Fall schwacher Erreichbarkeit existiert eine nichtnegative T-Invariante  $\vec{r}$ , in der  $t_A$  und  $t_B$  einmal schalten und deren Netzdarstellung weder Deadlocks noch Traps besitzt — das S/T-System ist damit lebendig in beide Richtungen. Im Fall starker Erreichbarkeit muss es eine solche T-Invariante geben, die zudem realisierbar ist. Alle weiteren in dieser Arbeit diskutierten Ansätze unterschieden nun zwischen schwacher und starker Erreichbarkeit:

In den Kapiteln 5.3 und 5.4 wurde anschließend gezeigt, warum die Berechnung oder Konstruktion einer geeigneten Linearkombination  $\vec{r}$  aus den minimalen T-Invarianten von  $N_{AB}$  alles andere als trivial ist: Bei großen Netzen mit Hunderten von T-Invarianten gibt es eine Vielzahl von Kombinationsmöglichkeiten, die im schlimmsten Fall alle berechnet und getestet werden müssen. Auch über die Lösung eines GLOP eine einzelne T-Invariante zu gewinnen führt in der Regel nicht schneller zum Ziel, da die Netzdarstellung dieser T-Invariante meistens noch Deadlocks oder Traps enthält. Aus

---

diesem Grund wurde nach weiteren Möglichkeiten gesucht, die Anzahl oder Gestalt der T-Invarianten durch eine geeignete Klassifizierung zu beschränken.

Da die Kreise im Netz das eigentliche Problem der Erreichbarkeitsuntersuchungen darstellen, musste zunächst ausreichend Kenntnis über deren Gestalt gewonnen werden. Indem  $N_{AB}$  in einen erweiterten Synchronisationsgraphen transformiert wurde, konnten in diesem S-Invarianten und damit alle minimalen Kreise von  $N_{AB}$  berechnet werden. Zudem wurde ein Verfahren vorgestellt, wie sich aus der Gestalt der Kreise — d.h. durch welche Stellen und Transitionen diese verlaufen — eine minimale Anzahl von Kanten ermitteln lässt, die man in  $N_{AB}$  löschen muss, um ein kreisfreies Netz zu erhalten. Um nichts am dynamischen Verhalten des Netzes zu ändern, wurden diese Kanten durch eine geeignete Struktur aus zusätzlichen Stellen und Transitionen ersetzt. Auf Basis dieses neuen Wissens wurden anschließend in den Kapiteln 6 und 7 zwei neue Lösungsansätze vorgestellt:

In *Lösungsansatz I* wurde zunächst eine T-Invariante  $\vec{r}$  berechnet, die schwache Erreichbarkeit nachweist und demnach die Basis für die Untersuchung auf starke Erreichbarkeit bildet. Ähnlich wie bei Kostins Verfahren wurden die T-Invarianten in Klassen eingeteilt. Mit dem neuen Wissen aus dem vorherigen Kapitel — die minimalen S-Invarianten des erweiterten Synchronisationsgraphen — konnte für Grenzstellen im Netz ermittelt werden, wie häufig Transitionen in deren Vorbereich schalten müssen, um genügend Marken für die Nachtransitionen bereitzustellen. Mittels dieser Gewichtung der verschiedenen Transitionen wurden geeignete T-Invarianten, in denen diese Transitionen feuern, ausgewählt und gewichtet zu einer Linearkombination  $\vec{r}$  addiert. Die Linearkombination selbst wurde (kurz gefasst) in einen sogenannten Pseudo-Synchronisationsgraphen entfaltet, in dem jede Transition im Netz nur einmal feuern darf. Transitionen mit Schalthäufigkeit  $h > 1$  wurden darum  $(h - 1)$ -mal als Kopie hinzugefügt und alle Kantengewichte durch Vervielfachung der Stellen eliminiert. Die Übergänge an Grenzstellen wurden durch sogenannte Transitionsgruppen modelliert, so dass alle denkbaren Wege von Eingangs- zu Ausgangstransitionen möglich sind. Die Existenz einer T-Invariante im erweiterten Synchronisationsgraphen, deren Netzdarstellung ein kreisfreier Synchronisationsgraph und damit automatisch lebendig ist, entsprach dem Nachweis starker Erreichbarkeit.

Bereits diese kurze Zusammenfassung zeigt aber, wie viele einzelne Schritte ausgeführt müssen, um letztendlich eine möglicherweise passende Linearkombination zu finden und zu testen. Zudem kann der Pseudo-Synchronisationsgraph schon bei sehr kleinen Netzen mit vielen Kanten oder großen Kantengewichten so komplex werden, dass eine Berechnung aller darin enthaltenen kreisfreien Synchronisationsgraphen in

der Praxis nicht ausführbar ist. Die große Anzahl von T-Invarianten entsteht aufgrund der vielen möglichen Wege durch die Transitionsgruppen. Auch die Berechnung einer einzigen T-Invariante durch Lösung eines GLOP kam in Frage, wurde jedoch verworfen, da in diesem Fall zwar nur genau eine T-Invariante berechnet wird, diese aber durch Constraints („Verboten der Kreise“) so genau spezifiziert werden muss, dass die Anzahl der erforderlichen Nebenbedingungen auch bei sehr kleinen Netzen sehr groß sein kann. Von der Überlegung, durch intelligente Kombination von T-Invarianten starke Erreichbarkeit nachzuweisen, ist also grundsätzlich abzuraten.

*Lösungsansatz II* ist als das vielversprechendste aller Verfahren anzusehen. Hierbei wird ein Netz ähnlich wie bei der Methode von Ramanchandran und Kamath in kreisfreie Schichten entfaltet. Alle Kreise im Netz werden aufgetrennt, indem sie nicht in der gleichen Schicht, sondern der Schicht darunter enden. Schwache Erreichbarkeit muss bereits gegeben sein. Starke Erreichbarkeit ist genau dann bewiesen, wenn nach einer gewissen Anzahl zusätzlicher Schichten eine T-Invariante  $\vec{r}$  mit kreisfreier Netzdarstellung gefunden wird, in der  $t_A$  und  $t_B$  nur einmal feuern. Die Berechnung der T-Invariante erfolgt durch das Lösen einer ganzzahligen linearen Extremwertaufgabe mit zusätzlichen Nebenbedingungen. Das Abbruchkriterium, das sich für die Netze in dieser Arbeit als brauchbar erwiesen hat, aber für beliebige Netze noch bewiesen werden muss, setzt sich aus zwei möglichen Fällen für die Unterscheidung starker und schwacher Erreichbarkeit zusammen: Starke Erreichbarkeit liegt im oben beschriebenen Fall vor. Schwache Erreichbarkeit ist nachgewiesen, wenn eine T-Invariante  $k_{min} \cdot \vec{r}$ ,  $k_{min} > 1$  gefunden wird und der Faktor  $k_{min}$  mit dem Hinzufügen einer neuen Schicht nicht mehr reduzierbar ist: Fällt der Zielfunktionswert nach dem Nachweis schwacher Erreichbarkeit nicht streng monoton mit jeder nächsten Schicht, so ist die neue Schicht „nutzlos“ und starke Erreichbarkeit kann nicht vorliegen.

Dieser Lösungsansatz ist zudem als der eleganteste und intuitiv einfachste anzusehen, was durch die in Kapitel 8 aufgeführten Beispiele deutlich geworden ist. In den Schichtennetzen kann nachverfolgt werden, wann welche Kreise durchlaufen und Marken geliehen oder zurückgegeben werden. Insbesondere, wenn eine Lösung gefunden ist, kann man deren Schaltfolge in Schichtennetzen mit niedrigerer Anzahl von Ebenen simulieren und erhält Informationen darüber, wo Marken beim Durchlaufen eines Kreises nicht rechtzeitig zurückgegeben werden konnten.

Als zukünftiger Forschungsgegenstand ist der Beweis der Korrektheit des Abbruchkriteriums für beliebige S/T-Netze anzusehen. Ist dieses Abbruchkriterium korrekt, so wurde tatsächlich eine rein linear-algebraische Methode für den Test auf starke Erreichbarkeit entwickelt. Weitere Anregungen für zukünftige Forschungsarbeiten sind:

---

Falls dieses Abbruchkriterium nicht für beliebige S/T-Netze gilt, kann selbstverständlich nach anderen oberen Schranken gesucht werden, beispielsweise nach einer maximalen Anzahl von Schichten, ab welcher starke Erreichbarkeit nicht mehr vorliegen kann. Einige zu untersuchende Kriterien können sein:

- die Anzahl der minimalen oder elementaren Kreise im Netz,
- erforderliche Schalthäufigkeiten bestimmter Transitionen unter Betrachtung der Kantengewichte
- oder eine Kombination aller Kriterien.

In der vorliegenden Arbeit wurde zudem deutlich, wie sich die Performance von Algorithmen zur Berechnung von Erzeugendensystemen steigern lässt, indem man nach T-Invarianten mit bestimmten Einschränkungen sucht, wie beispielsweise charakteristischen Vektoren oder festen Werten einzelner Komponenten, wodurch nur ein Bruchteil von Lösungen errechnet werden muss. Auch hier kann untersucht werden, ob sich ein S/T-Netz in eine Form transformieren lässt, durch die darauf arbeitende Algorithmen modifiziert werden und leistungsfähiger arbeiten können.

Zuletzt kann untersucht werden, ob sich Verfahren zur Lösung linearer ganzzahliger Extremwertaufgaben derart optimieren lassen, dass sie speziell auf Stellen/Transitions-Netze und deren Eigenschaften abgestimmt sind.



# Literaturverzeichnis

- [Bau96] BAUMGARTEN, BERND: *Petri-Netze. Grundlagen und Anwendungen*. Spektrum Akademischer Verlag, Heidelberg, 1996.
- [BEN11] BERKELAAR, M., K. EIKLAND und P. NOTEBAERT: *Lp\_solve Reference Guide*. <http://lpsolve.sourceforge.net/5.5/>, Abruf: 27.02.2011.
- [CD94] CONTEJEAN, EVELYNE und HERVÉ DEVIE: *An efficient incremental algorithm for solving systems of linear Diophantine equations*. Inf. Comput., 113(1):143–172, 1994.
- [CS91] COLOM, J. und M. SILVA: *Convex geometry and semiflows in P/T nets. A comparative study of algorithms for computation of minimal p-semiflows*. In: *Advances in Petri Nets 1990*, Band 483 der Reihe *Lecture Notes in Computer Science*, Seiten 79–112. Springer, Berlin/Heidelberg, 1991.
- [ECS93] EZPELETA, J., J. COUVREUR und M. SILVA: *A new technique for finding a generating family of siphons, traps and st-components. Application to colored Petri nets*. In: *Advances in Petri Nets 1993*, Band 674 der Reihe *Lecture Notes in Computer Science*, Seiten 126–147. Springer, Berlin/Heidelberg, 1993.
- [Esp98] ESPARZA, JAVIER: *Decidability and complexity of Petri net problems — An introduction*. In: *Lectures on Petri Nets I: Basic Models*, Band 1491 der Reihe *Lecture Notes in Computer Science*, Seiten 374–428. Springer, Berlin/Heidelberg, 1998.
- [Gen05] GENSTY, MICHAEL: *Bestimmung bedingter S- und T-Invarianten mit Mathematica im Petri-Netz-Tool Poseidon*. Studienarbeit, Universität Koblenz-Landau, Institut für Informatik, 2005.
- [JL<sup>+</sup>10] JÜNGER, M., THOMAS M. LIEBLING et al. (Herausgeber): *50 Years of Integer Programming 1958-2008*. Springer, Berlin/Heidelberg, 2010.
- [Joh75] JOHNSON, DONALD B.: *Finding All the Elementary Circuits of a Directed Graph*. SIAM J. Comput., 4(1):77–84, 1975.

- [KJ87] KRÜCKEBERG, FRITZ und MICHAEL JAXY: *Mathematical methods for calculating invariants in Petri nets*. In: *Advances in Petri Nets 1987*, Band 266 der Reihe *Lecture Notes in Computer Science*, Seiten 104–131. Springer, Berlin/Heidelberg, 1987.
- [Kos06] KOSTIN, ALEXANDER: *A Reachability Algorithm for General Petri Nets Based on Transition Invariants*. In: *Mathematical Foundations of Computer Science 2006*, Band 4162 der Reihe *Lecture Notes in Computer Science*, Seiten 608–621. Springer, Berlin/Heidelberg, 2006.
- [Kos08] KOSTIN, ALEXANDER E.: *Using Transition Invariants for Reachability Analysis of Petri Nets*. In: *Petri Net, Theory and Applications*. I-Tech Education and Publishing, 2008.
- [Lau73] LAUTENBACH, KURT: *Exakte Bedingungen der Lebendigkeit für eine Klasse von Petri-Netzen*. Berichte der GMD Nr. 82, Bonn, 1973.
- [Lau87] LAUTENBACH, KURT: *Linear algebraic techniques for place/transition nets*. In: *Petri Nets: Central Models and Their Properties*, Band 254 der Reihe *Lecture Notes in Computer Science*, Seiten 142–167. Springer, Berlin/Heidelberg, 1987.
- [Lau02] LAUTENBACH, KURT: *Reproducibility of the Empty Marking*. In: *Application and Theory of Petri Nets 2002*, Band 2360 der Reihe *Lecture Notes in Computer Science*, Seiten 281–300. Springer, Berlin/Heidelberg, 2002.
- [Lau04] LAU, DIETLINDE: *Algebra und Diskrete Mathematik 2*. Springer, Berlin/Heidelberg, 2004.
- [Lau05] LAUTENBACH, KURT: *Duality and Diagnosis in Higher Level Petri Nets*. Skript zur Vorlesung ‘Duality and Diagnosis in Higher Level Petri Nets’ im SS 2005, Universität Koblenz-Landau, Institut für Softwaretechnik, Arbeitsgruppe Petri-Netze, 2005.
- [Lip76] LIPTON, R.: *The reachability problem requires exponential space*. Technischer Bericht, Department of Computer Science, New Haven, CT, 1976.
- [May81] MAYR, E. W.: *An Algorithm for the General Petri Net Reachability Problem*. In: *Proc. of the 13th Annual ACM Symp. on Theory of Computing*, Seiten 238–246, 1981.



- [Mey11] MEYER, FRANK: *ElementaryCycles – This is an implementation of an algorithm by Donald B. Johnson to find all elementary cycles in a directed graph.* <http://normalisiert.de/code/java/elementaryCycles.zip> (.zip-Datei, bestehend aus 5 Java-Klassen), Abruf: 27.02.2011.
- [MS82] MARTÍNEZ, J. und MANUEL SILVA: *A Simple and Fast Algorithm to Obtain All Invariants of a Generalized Petri Net.* In: *Selected Papers from the First and the Second European Workshop on Application and Theory of Petri Nets*, Band 52 der Reihe *Informatik-Fachberichte*, Seiten 301–310, Springer, London, UK, 1982.
- [Mül85] MÜLLER, HORST: *The reachability problem for VAS.* In: *Advances in Petri Nets 1984*, Band 188 der Reihe *Lecture Notes in Computer Science*, Seiten 376–391. Springer, Berlin/Heidelberg, 1985.
- [Mur89] MURATA, TADAO: *Petri Nets: Properties, Analysis and Applications.* In: *Proceedings of the IEEE*, Seiten 541–580, 1989.
- [NW88] NEMHAUSER, GEORGE L. und LAURENCE A. WOLSEY: *Integer and combinatorial optimization.* Wiley-Interscience, New York, NY, USA, 1988.
- [Pas86] PASCOLETTI, K.H.: *Diophantische Systeme und Lösungsmethoden zur Bestimmung aller Invarianten in Petri-Netzen.* Berichte der GMD Nr. 160, Bonn, 1986.
- [RK04] RAMACHANDRAN, P. und M. KAMATH: *A Sufficient Condition for Reachability in a General Petri Net.* *Discrete Event Dynamic Systems*, 14:251–266, 2004.
- [Row11] ROWLAND, TODD: *Vector Space Basis.* From MathWorld — A Wolfram Web Resource, created by Eric W. Weisstein. <http://mathworld.wolfram.com/VectorSpaceBasis.html>, Abruf: 27.02.2011.
- [Sch99] SCHRIJVER, ALEXANDER: *Theory of linear and integer programming.* Wiley-Interscience, Chichester [u.a.], 1999.
- [Sta90] STARKE, PETER H.: *Analyse von Petri-Netz-Modellen.* Teubner, Stuttgart, 1990.
- [STC98] SILVA, MANUEL, ENRIQUE TERUE und JOSÉ COLOM: *Linear algebraic and linear programming techniques for the analysis of place/transition net systems.*

- In: *Lectures on Petri Nets I: Basic Models*, Band 1491 der Reihe *Lecture Notes in Computer Science*, Seiten 309–373. Springer, Berlin/Heidelberg, 1998.
- [Sus08] SUSEWIND, KERSTIN: *Berechnungsalgorithmen für Hermite-Normalformen und deren Anwendung zur Bestimmung ganzzahliger Lösungen linearer Gleichungssysteme*. Studienarbeit, Universität Koblenz-Landau, Institut für Informatik, 2008.
- [Tar72] TARJAN, ROBERT E.: *Enumeration of the Elementary Circuits of a Directed Graph*. Technischer Bericht, Department of Computer Science, Ithaca, NY, USA, 1972.
- [Wei11] WEISSTEIN, ERIC W.: *Gaussian Elimination*. From MathWorld — A Wolfram Web Resource. <http://mathworld.wolfram.com/GaussianElimination.html>, Abruf: 27.02.2011.
- [Wol03] WOLFRAM, STEPHEN: *The Mathematica Book*. Wolfram Media, Inc., 2003.
- [WW10] WIMMEL, HARRO und KARSTEN WOLF: *Reachability Analysis via Net Structure*. In: *Proceedings of the 17th German Workshop on Algorithms and Tools for Petri Nets (AWPN 2010)*, Band 643 der Reihe *CEUR Workshop Proceedings*, Seiten 118–123, Cottbus, 2010.