Skript zur Vorlesung
Petrinetze
SS 95

Prof. Javier Esparza

27. April 2005

Dieses Skript basiert auf meinen eigenen Notizen für die Spezialvorlesung Petrinetze in SS 95. Die Hauptquelle sind die folgenden Bücher (die auch gute Literaturlisten enthalten):

> J. Desel. **Struktur und Analyse von Free-Choice-Petrinetzen**. Deutscher Universitäts Verlag, 1992.
>
> J. Desel und J. Esparza. **Free-choice Petri nets**. Cambridge Tracts in Theoretical Computer Science 40, Cambridge University Press, 1995.

Das Netzmodell von Petersons Algorithmus kommt aus

> E. Best. **Semantik**. Vieweg, 1995.

Die Quelle des Aktion-Reaktion Protokolls ist

> R. Walter. **Petrinetzmodelle verteilter Algorithmen – Intuition und Beweistechnik**. Dieter Bertz Verlag, 1996.

Die Bahnnetz-Beispiele gehören schon zum Petrinetz-Folklore. Die wurden erst von H. Genrich eingeführt.
Vielen Dank an Doris Reisenauer für die Erstellung des Skripts.

Teil I

Das Modell

Kapitel 1

Grundlagen

1.1 Allgemeine Definitionen

1.1.1 Zahlen

$\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ und $\mathbb{R}$ bezeichnen die Mengen der natürlichen, ganzen, rationalen und reellen Zahlen.

Relationen

Sei X eine Menge und sei $R \subseteq X \times X$ eine Relation.
R^* bezeichnet die *transitive und reflexive Hülle* von R.
R^{-1} ist die *inverse* Relation von R : $(x, y) \in R^{-1} \Leftrightarrow (y, x) \in R$.

Sequenzen

Eine *endliche Sequenz* über eine Menge A ist eine Abbildung $\sigma\colon \{1, \ldots, n\} \rightarrow A$, dargestellt durch die Kette $a_1 a_2 \ldots a_n$ mit $a_i = \sigma(i)$ für alle $1 \leq i \leq n$, oder die Abbildung $\epsilon\colon \emptyset \rightarrow A$, die *leere Sequenz*.
Die *Länge* von σ ist n und die Länge von ϵ ist 0.
Eine unendliche Sequenz ist eine Abbildung $\sigma : \mathbb{N} \rightarrow A$. Wir schreiben $\sigma = a_1 a_2 a_3 \ldots$ wobei $a_i = \sigma(i)$.
Die *Konkatenation* zweier endlicher Sequenzen bzw. einer endlichen und einer unendlichen Sequenz wird wie üblich definiert.
σ^ω bezeichnet die *unendliche Konkatenation* $\sigma\sigma\sigma \ldots$ (für σ endlich).
σ ist ein *Präfix* von τ, wenn $\sigma = \tau$ oder $\sigma\sigma' = \tau$ für eine Sequenz σ'.
Das *Alphabet* einer Sequenz σ ist die Menge der in σ auftretenden Elemente.
Für eine Sequenz σ über A und eine Menge $B \subseteq A$ entsteht die *Projektion* oder *Restriktion* $\sigma|_B$ durch Elimination aller Elemente $a \in A \setminus B$ in σ.

Vektoren und Matrizen

Sei $A = \{a_1, \ldots, a_n\}$ eine endliche Menge und sei K eine der Mengen $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$.
Eine Abbildung $X\colon A \to K$ wird durch den Vektor $(X(a_1), \ldots, X(a_n))$ dargestellt.
Wir identifizieren die Abbildung X und ihre Darstellung.
Seien $X = (x_1, \ldots, x_n)$ und $Y = (y_1, \ldots, y_n)$ zwei Vektoren.
Das *Produkt* oder *Skalarprodukt* $X \cdot Y$ ist die Zahl $x_1y_1 + \ldots + x_ny_n$ (wir unterscheiden nicht zwischen Zeichen-und Spaltenvektoren!).
$X \geq Y$ bezeichnet $x_1 \geq y_1 \wedge \ldots \wedge x_n \geq y_n$. $X > Y$ bezeichnet $x_1 > y_1 \wedge \ldots \wedge x_n > y_n$.
Sei $B = \{b_1, \ldots, b_m\}$ eine endliche Menge. Eine Abbildung $C\colon A \times B \to K$ wird durch die $n \times m$ Matrix

$$\begin{pmatrix} C(a_1,b_1) & C(a_1,b_2) & \cdots & C(a_1,b_m) \\ C(a_2,b_1) & C(a_2,b_2) & \cdots & C(a_2,b_m) \\ \cdots & \cdots & \cdots & \cdots \\ C(a_n,b_1) & C(a_n,b_2) & \cdots & C(a_n,b_m) \end{pmatrix}$$

dargestellt. Wir schreiben auch $C = (c_{ij})_{i=1,\ldots,n, j=1,\ldots,m}$, wobei $c_{ij} = C(a_i, b_j)$.
Sei $X = (x_1, \ldots, x_m)$ ein Vektor und sei C eine $n \times m$ Matrix. Das *Produkt* $C \cdot X$ ist der Vektor $Y = (y_1, \ldots, y_n)$ gegeben durch

$$y(i) = c_{i1}x_1 + \ldots + c_{im}x_m$$

Das Produkt $X \cdot C$ ist der Vektor $Y = (y_1, \ldots, y_n)$ gegeben durch

$$y(i) = c_{1i}x_1 + \ldots + c_{mi}x_m$$

1.2 Netze

Definition 1.2.1 (Netz, Vorbereich, Nachbereich)
Ein *Netz* $N = (S, T, F)$ besteht aus einer Menge S von *Stellen* (dargestellt durch Kreise), einer mit S disjunkten Menge T von *Transitionen* (Quadrate) und einer *Fluß-relation* (gerichtete Kanten) $F \subseteq (S \times T) \cup (T \times S)$.
Elemente oder *Knoten* von N sind alle Stellen und alle Transitionen.
Für $x \in S \cup T$ bezeichnet ${}^\bullet x = \{y \mid (y, x) \in F\}$ den *Vorbereich* von x und $x^\bullet = \{y \mid (x, y) \in F\}$ den *Nachbereich* von x. Für eine Menge $X \subseteq S \cup T$ ist ${}^\bullet X = \bigcup_{x \in X} {}^\bullet x$ und $X^\bullet = \bigcup_{x \in X} x^\bullet$

Beispiel. Sei $N = (S, T, F)$ das Netz mit

$$\begin{aligned} S &= \{s_1, \ldots, s_6\} \\ T &= \{t_1, \ldots, t_4\} \\ F &= \{(s_1,t_1), (t_1,s_2), (s_2,t_2), (t_2,s_1), \\ &\quad (s_3,t_2), (t_2,s_4), (s_4,t_3), (t_3,s_3), \\ &\quad (s_5,t_3), (t_3,s_6), (s_6,t_4), (t_4,s_5)\} \end{aligned}$$

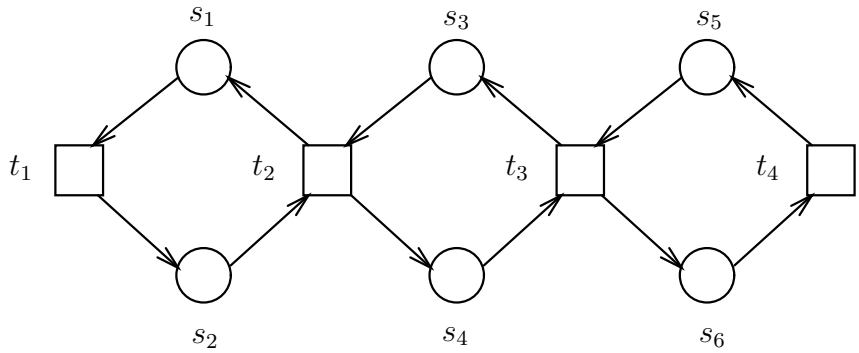


Abbildung 1.1: Graphische Darstellung des Netzes N

Abbildung 1.1 zeigt die graphische Darstellung von N. Wir haben z.B. ${}^\bullet t_2 = \{s_2, s_3\}$ und ${}^\bullet S = S^\bullet = T$

Bemerkung: Netze mit leerem S, T oder F sind erlaubt!

Definition 1.2.2 (Teilnetz)
$N' = (S', T', F')$ ist ein *Teilnetz* von $N = (S, T, F)$ gdw.

- $S' \subseteq S$,
- $T' \subseteq T$, und
- $F' = F \cap ((S' \times T') \cup (T' \times S'))$ (nicht $F' \subseteq F \cap ((S' \times T') \cup (T' \times S'))$!).

Abbildung 1.2 zeigt einige Teilnetze und Nicht-Teilnetze des Netzes aus Abbildung 1.1.

Definition 1.2.3 (Pfad, Kreis)
Sei $N = (S, T, F)$ ein Netz.
Eine endliche nichtleere Sequenz $x_1, \ldots, x_n$ von Elementen von N heißt *Pfad* von N, wenn $(x_1, x_2), \ldots, (x_{n-1}, x_n) \in F$.
Ein Pfad $x_1, \ldots x_n$ *führt von* x_1 *nach* x_n.
Ein Pfad heißt *Kreis*, wenn $(x_n, x_1) \in F$ und $(x_i = x_j) \Rightarrow i = j$ für alle $1 \leq i, j \leq n$.
N ist *zusammenhängend*, wenn $(x, y) \in (F \cup F^{-1})^*$ für alle $x, y \in S \cup T$. N ist *stark zusammenhängend*, wenn $(x, y) \in F^*$ für alle $x, y \in S \cup T$

Bemerkungen:
- Jedes Netz mit 0 oder 1 Element ist stark zusammenhängend!
- N stark zusammenhängend $\Rightarrow$ N zusammenhängend.

Proposition 1.2.4 *Sei $N = (S, T, F)$ ein Netz.*

(1) N ist zusammenhängend, wenn es keine Teilnetze (S_1, T_1, F_1) und (S_2, T_2, F_2) von N gibt, mit

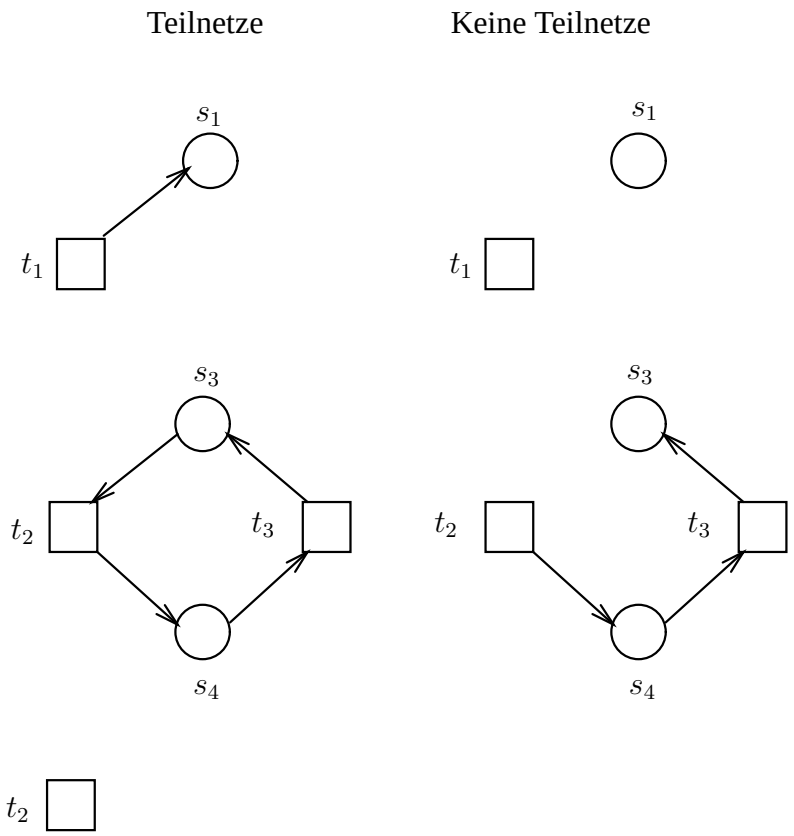


Abbildung 1.2: Teilnetze und Nicht-Teilnetze vom Netz aus Abbildung 1.1

- $S_1 \cup T_1 \neq \emptyset$, $S_2 \cup T_2 \neq \emptyset$;
- $S_1 \cup S_2 = S$, $T_1 \cup T_2 = T$, $F_1 \cup F_2 = F$;
- $S_1 \cap S_2 = \emptyset$, $T_1 \cap T_2 = \emptyset$.

(2) Ein zusammenhängendes Netz ist stark zusammenhängend, wenn es für jede Kante $(x, y) \in F$ einen Pfad gibt, der von y nach x führt.

Beweis. Aufgabe. □

1.3 Verhalten

Definition 1.3.1 (Markierungen)
Sei $N = (S, T, F)$ ein Netz.
Eine *Markierung* von N ist eine Abbildung $M\colon S \to \mathbb{N}$. Für $R \subseteq S$ ist $M(R) = \sum_{s \in R)} M(s)$.
Eine Stelle heißt *markiert* von M (oder unter M), wenn $M(s) > 0$. Eine Menge von Stellen heißt *markiert* von M (oder unter M), wenn $M(R) > 0$, d.h., wenn wenigstens eines ihrer Elemente von M markiert wird.

Um anstelle von Abbildungen $S \to \mathbb{N}$ Vektoren verwenden zu können, betrachten wir stets beliebige aber feste Ordnungen auf den Stellen. Mit dieser Konvention ist jede Markierung $M\colon S \to \mathbb{N}$ als $|S|$-stelliger Vektor darstellbar.
Graphisch werden Markierungen durch schwarze Punkte ("tokens" oder "Marken") auf den Stellen dargestellt.

Definition 1.3.2 (Schaltregel, tote Markierungen)
Eine Transition ist *aktiviert* unter einer Markierung M, wenn $M(s) \geq 1$ für jede Stelle $s \in {}^\bullet t$. Falls t aktiviert ist, kann sie *schalten* und überführt M in die Markierung M' (bezeichnet $M \xrightarrow{t} M'$), die definiert ist durch:

$$M'(s) = \begin{cases} M(s) - 1 & \text{falls } s \in {}^\bullet t \setminus t^\bullet \\ M(s) + 1 & \text{falls } s \in t^\bullet \setminus {}^\bullet t \\ M(s) & \text{sonst} \end{cases}$$

Eine Markierung ist *tot*, wenn sie keine Transition aktiviert.

Beispiel. Sei M die Markierung des Netzes N aus Abbildung 1.1 mit $M(s_1) = M(s_3) = M(s_5) = 1$ und $M(s_2) = M(s_4) = M(s_6) = 0$. Diese Markierung stellen wir als $(1, 0, 1, 0, 1, 0)$ dar.
Die Transitionen t_1 und t_3 sind aktiviert, weil ${}^\bullet t_1 = \{s_1\}$ und ${}^\bullet t_3 = \{s_3, s_5\}$. Die Transition t_2 ist nicht aktiviert, weil $M(s_2) = 0$. Die Transition t_4 ist nicht aktiviert, weil $M(s_6) = 0$.

Wir haben

$$(1,0,1,0,1,0) \xrightarrow{t_1} (0,1,1,0,1,0)$$
$$(1,0,1,0,1,0) \xrightarrow{t_3} (1,0,0,1,0,1)$$

Definition 1.3.3 (Schaltfolgen, Erreichbare Markierungen)
Sei N ein Netz und sei M eine Markierung von N.
Eine endliche Sequenz $\sigma = t_1 \dots t_n$ heißt *von M aktivierte endliche Schaltfolge*, wenn Markierungen $M_1, M_2, \dots, M_n$ existieren, so daß $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} \dots \xrightarrow{t_n} M_n$. Wir schreiben $M \xrightarrow{\sigma} M_n$.
Die leere Sequenz ϵ wird von jeder Markierung aktiviert und es gilt $M \xrightarrow{\epsilon} M$.
Wir schreiben M $\xrightarrow{*} M'$ und sagen, daß M' *von M erreichbar ist*, wenn es eine Sequenz σ existiert mit $M \xrightarrow{\sigma} M'$.
$[M\rangle$ bezeichnet die Menge der von M erreichbaren Markierungen.
Eine unendliche Sequenz $\sigma = t_1 t_2 \dots$ heißt *von M aktivierte unendliche Schaltfolge*, wenn Markierungen $M_1, M_2, \dots$ existieren mit $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \longrightarrow \dots$

Beispiel. Sei N das Netz aus Abbildung 1.1, und sei $M = (1,0,1,0,1,0)$ eine Markierung von N. Wir haben

$$\begin{array}{ccccc} (1,0,1,0,1,0) & \xrightarrow{t_1} & (0,1,1,0,1,0) & \xrightarrow{t_3} & (0,1,0,1,0,1) \\ & & & & \downarrow t_2 \\ & & & & (1,0,1,0,0,1) \end{array} \quad \xrightarrow{t_4} \quad (1,0,1,0,1,0)$$

Also, $t_1\ t_3\ t_2\ t_4$ ist eine von M aktivierte endliche Schaltfolge, und $(t_1\ t_3\ t_2\ t_4)^\omega$ ist eine von M aktivierte unendliche Schaltfolge.

Proposition 1.3.4 *Eine endliche oder unendliche Sequenz σ wird von einer Markierung M aktiviert genau dann, wenn jeder endliche Präfix von σ von M aktiviert wird.*

Beweis. Einfache Aufgabe. □

1.4 Zwei wichtige Eigenschaften

Lemma 1.4.1 *[Das Monotonielemma]*
Seien M und L Markierungen eines Netzes

(1) Wenn $M \xrightarrow{\sigma} M'$ für eine endliche Sequenz σ, dann $(M+L) \xrightarrow{\sigma} (M'+L)$

(2) Wenn $M \xrightarrow{\sigma}$ für eine unendliche Sequenz σ, dann $(M+L) \xrightarrow{\sigma}$

Beweis.
Zu (1): Durch Induktion über die Länge von σ.
Basis: $\sigma = \epsilon$. ϵ wird von jeder Markierung aktiviert.
Schritt: Sei $\sigma = \tau t$ (t Transition) mit $M \xrightarrow{\tau} M'' \xrightarrow{t} M'$. Aus der Induktionsvoraussetzung folgt $(M+L) \xrightarrow{\tau} (M''+L)$. Aus der Schaltregel und $M'' \xrightarrow{t} M'$ folgt

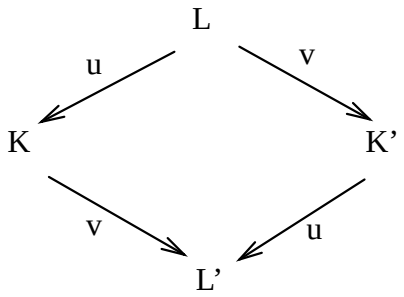


Abbildung 1.3:

$(M'' + L) \xrightarrow{t} (M' + L)$. Damit $(M + L) \xrightarrow{\tau t} (M' + L)$.

Zu (2): Wir zeigen, daß jeder endliche Präfix von σ von der Markierung $M + L$ aktiviert wird. Nach Proposition 1.3.4 wird dann σ auch von M aktiviert. Aus Proposition 1.3.4 folgt, daß jeder endliche Präfix von σ von M aktiviert wird. D.h., für jeden endlichen Präfix τ von σ gibt es eine Markierung M' mit $M \xrightarrow{\tau} M'$. Aus (1) folgt $(M + L) \xrightarrow{\tau} (M' + L)$ und wir sind fertig. □

Lemma 1.4.2 *[Das Vertauschungslemma]*
Sei $N = (S, T, F)$ ein Netz und seien U, V disjunkte Teilmengen von T mit ${}^\bullet U \cap V^\bullet = \emptyset$. Sei σ eine (endliche oder unendliche) Menge von Transitionen mit $\mathcal{A}(\sigma) \subseteq U \cup V$

(1) Ist σ endlich und gilt $M \xrightarrow{\sigma} M'$, dann ist $M \xrightarrow{\sigma|_U \sigma|_V} M'$ eine Schaltfolge.

(2) Ist σ unendlich, ist $\sigma|_U$ endlich und gilt $M \xrightarrow{\sigma}$, dann ist $M \xrightarrow{\sigma|_U \sigma|_V}$ eine unendliche Schaltfolge.

(3) Ist σ unendlich, ist $\sigma|_U$ unendlich und gilt $M \xrightarrow{\sigma}$, dann ist $M \xrightarrow{\sigma|U}$ eine unendliche Schaltfolge.

Beweis. Die folgende Aussage folgt aus den Definitionen: Wenn $L \xrightarrow{u} K \xrightarrow{v} L'$ für beliebige Markierungen L, K, L' und beliebige Transitionen $u \in U, v \in V$, dann gibt es eine Markierung K' mit $L \xrightarrow{v} K' \xrightarrow{u} L'$ (siehe Abbildung 1.3)

(1) Wegen $U \cap V = \emptyset$ und $\mathcal{A}(\sigma) \subseteq U \cup V$ führt das erschöpfende Vertauschen von Paaren u, v mit $u \in U$ und $v \in V$ von $M \xrightarrow{\sigma} M'$ zu $M \xrightarrow{\sigma|_U \sigma|_V} M'$.

(2) Sei $\sigma = \sigma'\sigma''$ derart, daß in σ'' nur Transitionen aus V vorkommen (eine solche Zerlegung existiert, da $\sigma|_U$ endlich ist). Sei $M \xrightarrow{\sigma'} M' \xrightarrow{\sigma''} M'$. Wegen (1) ist $M \xrightarrow{\sigma|_U \sigma'|_V} M'$ eine Schaltfolge. Das Ergebnis folgt wegen $\sigma|_U = \sigma'|_U$ und $\sigma|_V = \sigma'|_V \sigma''$.

(3) Wegen Proposition 1.3.4 reicht es zu zeigen, daß jeder endliche Präfix von $\sigma|_U$ von M aktiviert wird. Sei τ' ein Präfix von $\sigma|_U$ und sei τ ein Präfix von σ mit $\tau' = \tau|_U$. Wegen (1) ist $M \xrightarrow{\tau|_U} M' \xrightarrow{\tau|_V} M'$ eine Schaltfolge für irgendeine Markierung M' und M''. Das Ergebnis folgt wegen $\tau' = \tau|_U$.

□

Kapitel 2

Netzsysteme und Beispiele

Definition 2.0.3 (Netzsystem)
Ein *Netzsystem* oder *System* (N, M_0) besteht aus einem zusammenhängenden Netz $N = (S, T, F)$ mit mindestens einer Stelle und einer Transition und aus einer Anfangsmarkierung $M_0 \colon S \to \mathbb{N}$.
Eine Markierung M ist *erreichbar in* (N, M_0), wenn $M_0 \xrightarrow{*} M$. Wir sagen auch, daß M eine erreichbare Markierung von (N, M_0) ist.

Definition 2.0.4 (Erreichbarkeitsgraph)
Der *Erreichbarkeitsgraph* G eines Systems (N, M_0) mit $N = (S, T, F)$ ist der einzige gerichtete, beschrifte Graph, der die folgende Eigenschaften erfüllt:

- Die Knoten von G sind die erreichbare Markierungen von (N, M_0).
- Die Kanten von G sind mit Transitionen aus T beschriftet.
- Es gibt eine mit t beschriftete Kante von M nach M' genau dann, wenn $M \xrightarrow{t} M$, d.h., genau dann, wenn M die Transition t aktiviert, und das Schalten von t die Markierung M in die Markierung M' überführt.

2.0.1 Ein Puffer mit Kapazität n

Wir modellieren einen Puffer mit Kapazität für n Nachrichten. Abbildung 2.1 zeigt das Netzsystem für $n = 3$. Das System besteht aus n Zellen, jeweils mit Kapazität für eine Nachricht. Die Eingabe einer neuen Nachricht wird durch das Schalten von t_1 modelliert. Das Schalten einer Transition t_i, $1 < i \leq n$, bewirkt, daß die Nachricht von der Zelle $i - 1$ in die Zelle i geht. Das Schalten von t_{i+1} modelliert die Ausgabe einer Nachricht. Das System ist nichtsequentiell: z.B. gibt es erreichbare Markierungen, aus denen die Transitionen t_1 und t_4 völlig unabhängig voneinander schalten können.
Abbildung 2.2 zeigt den Erreichbarkeitsgraphen des Puffers mit Kapazität 3. Mit Hilfe des Erreichbarkeitsgraphen kann man zeigen daß, unter anderen, die folgenden Eigenschaften gelten:

- Konsistenz: keine Zelle ist gleichzeitig leer und voll (d.h., keine erreichbare Markierung markiert gleichzeitig die Stelle s_i und die Stelle s_{i+1} für $i = 1, 2, 3$).

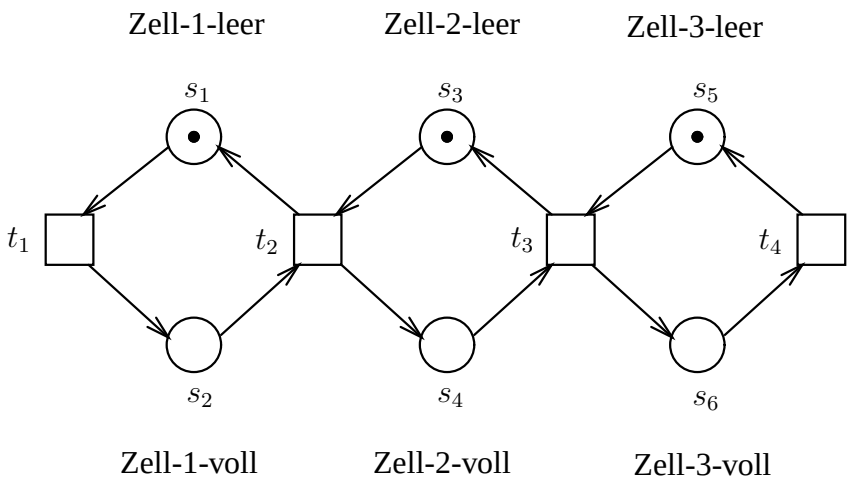


Abbildung 2.1: Ein 3-Puffer

- 1-Beschränkheit: keine erreichbare Markierung stellt mehr als eine Marke auf einer Stelle.
- Verklemmungsfreiheit: jede erreichbare Markierung hat mindestens einen Nachfolger.
  Sogar: jede Zelle kann immer wieder gefüllt und geleert werden (jede Transition kann immer wieder schalten).
- Kapazität n: der Puffer hat tatsächlich Kapazität n, d.h., es gibt eine erreichbare Markierung, in der alle Zellen voll sind (in der die Stellen s_1, s_3, s_5 markiert sind).
- Die Anfangsmarkierung kann immer wieder erreicht werden.
- Zwischen zwei beliebigen erreichbaren Markierungen gibt es einen Weg mit Länge höchstens 6.

2.0.2 Das Bahnnetz (erste Version)

Vier Städte sind durch unidirektionale Bahnstrecken in einem Kreis verbunden. Zwei Züge fahren auf diesen Strecken. Es muß garantiert werden, daß sich niemals beide Züge auf derselben Strecke befinden.
Abbildung 2.3 zeigt eine Netzlösung des Problems. Die vier Strecken werden mit Stellen $s_1, \ldots, s_4$ modelliert. Eine marke auf s_i bedeutet, daß ein Zug auf der i-te Strecke fährt.
Die vier Kontrollstellen $l_1, \ldots, l_4$ garantieren, daß keine erreichbare Markierung mehr als eine Marke auf eine der Stellen s_i stellt. Diese Eigenschaft kann man mit Hilfe des in Abbildung 2.4 dargestellten Erreichbarkeitsgraphen beweisen. Da in jeder erreichbaren Markierung höchstens eine Marke auf einer Stelle liegt, bezeichnen wir eine Markierung mit der Menge der Stellen, die sie markiert. Z.B., die Markierung, die die Stellen l_1, s_2, l_3 und s_4 markiert wird mit $\{l_1, s_2, l_3, s_4\}$ bezeichnet.

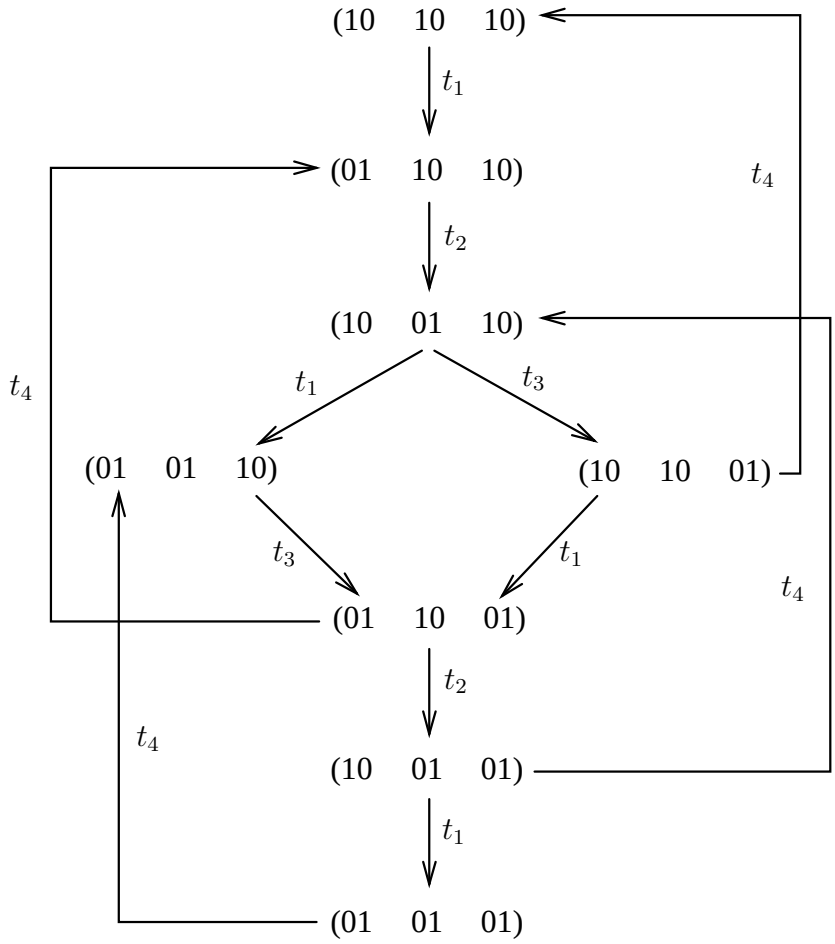


Abbildung 2.2: Erreichbarkeitsgraph des 3-Puffers

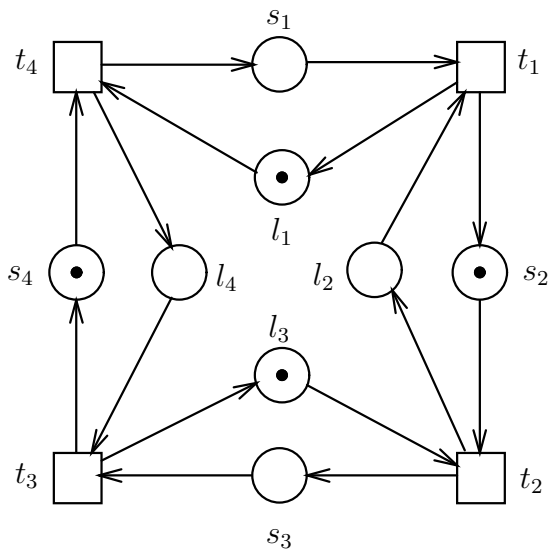


Abbildung 2.3: Bahnnetz (erste Version)

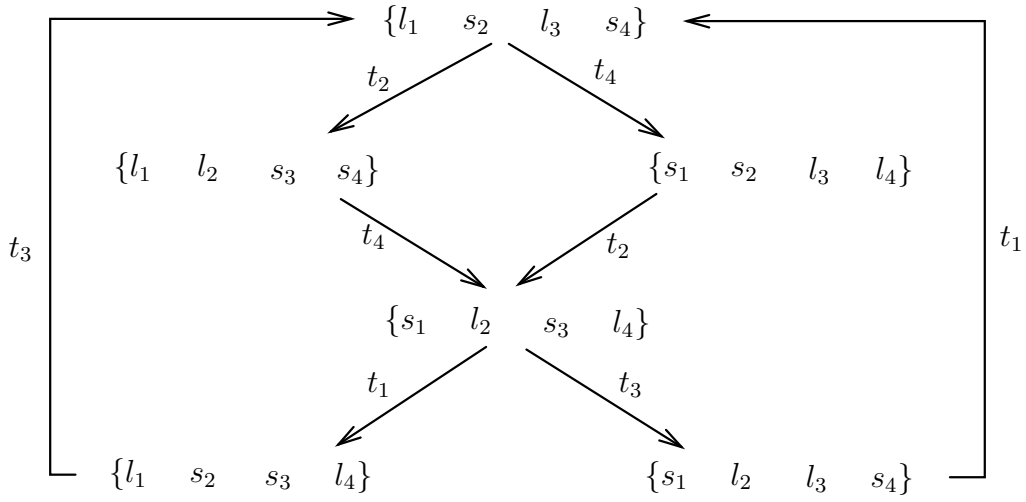


Abbildung 2.4: Erreichbarkeitsgraph des Netzsystems aus Abbildung 2.3

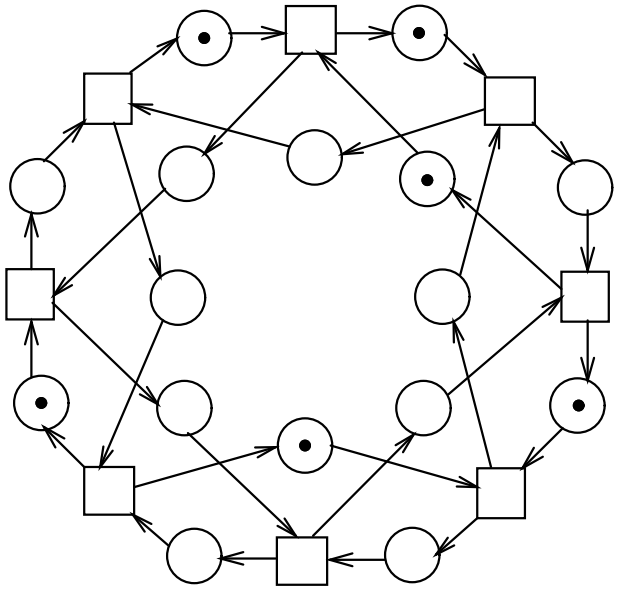


Abbildung 2.5: Bahnnetz (zweite Version)

2.0.3 Das Bahnnetz (zweite Version)

Nun sind es 8 Städte, die durch unidirektionale Bahnstrecken in einem Ring verbunden sind, und vier Züge fahren auf den Strecken. Um die Sicherheit zu erhöhen, muß man garantieren, daß es immer mindestens eine leere Strecke zwischen zwei beliebigen Zügen gibt.
Das System aus Abbildung 2.5 ist eine Lösung des Problems: Der Leser kann den Erreichbarkeitsgraphen konstruieren und zeigen, daß die gewünschte Eigenschaft gilt. Der Erreichbarkeitsgraph ist aber ziemlich groß!

2.0.4 Petersons Algorithmus

Petersons Algorithmus ist eine bekannte Lösung des Problems des gegenseitigen Ausschlusses für zwei Prozesse.

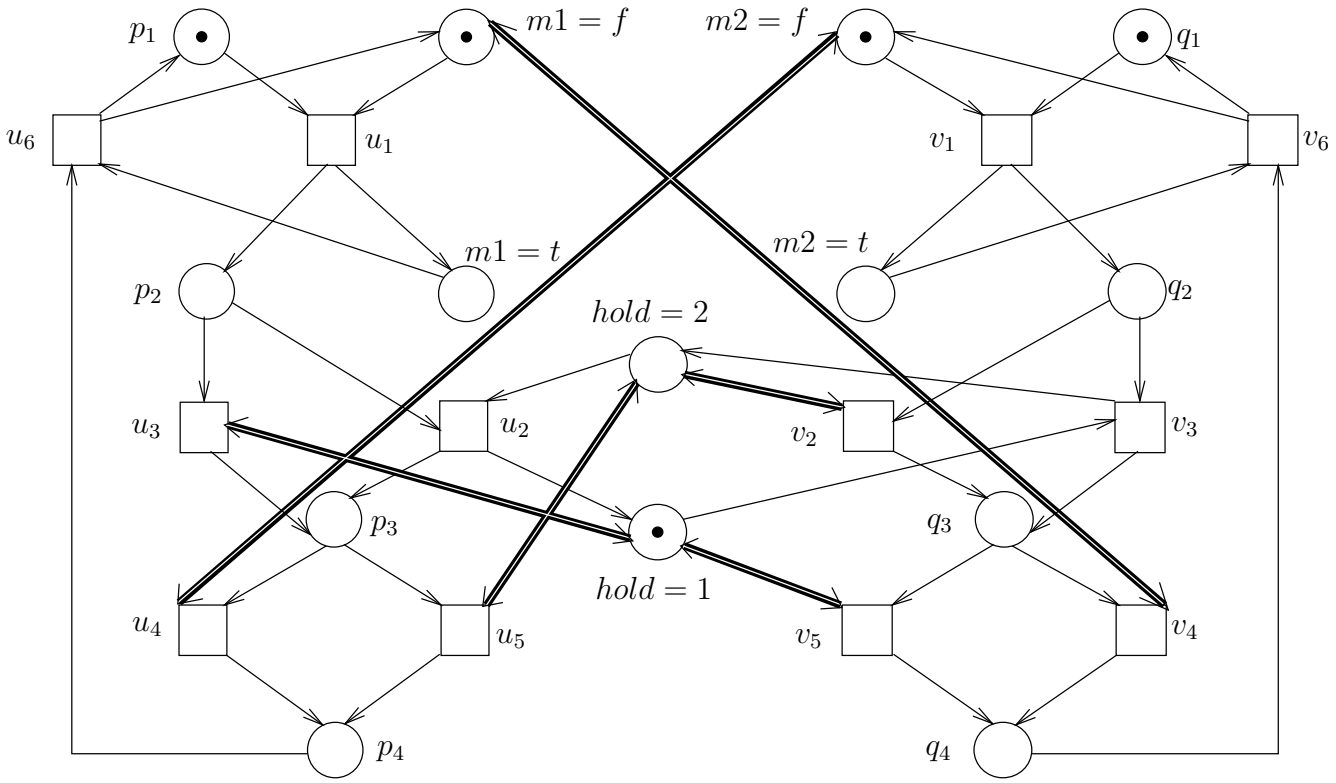


Abbildung 2.6: Netzmodell von Petersons Algorithmus

var $m_1, m_2 : \{false, true\}$ (**init** *false*);
$hold : \{1, 2\}$;

while $true$ **do**
$m_1 := true$;
$hold := 1$;
await$(\neg m_2 \vee hold = 2)$;
(kritischer Abschnitt);
$m_1 := false$;
od

while $true$ **do**
$m_2 := true$;
$hold := 2$;
await$(\neg m_1 \vee hold = 1)$;
(kritischer Abschnitt);
$m_2 := false$;
od

Das System aus Abbildung 2.6 ist ein Netzmodell dieses Algorithmus. Die Variable m_i wird mit Hilfe zweier Stellen modelliert, $m_i = true$ und $m_i = false$. Eine Marke auf $m_i = true$ bedeutet, daß im aktuellen Zustand (Markierung) die Variable m_i den Wert $true$ hat (das System muß also die Eigenschaft erfüllen, daß keine erreichbare Markierung die Stellen $m_i = true$ und $m_i = false$ gleichzeitig markiert). Die Variable k wird analog modelliert.

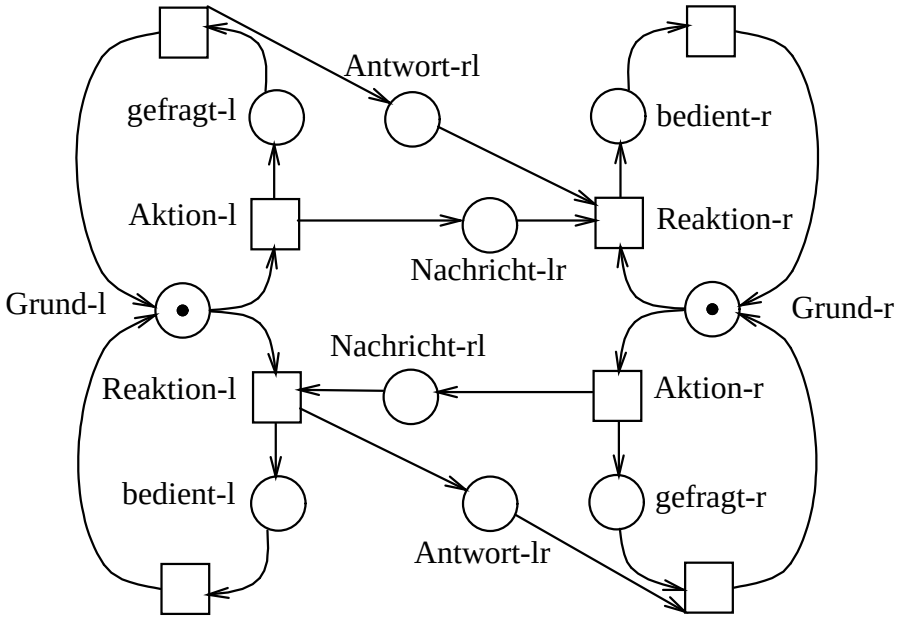


Abbildung 2.7: Der erste Versuch

Eine Marke auf p_4 (q_4) signalisiert, daß der linke (rechte) Prozeß sich in seinem kritischen Abschnitt befindet. Der gegenseitige Auschluß wird also gewährleistet, wenn es keine erreichbare Markierung gibt, die gleichzeitig p_4 und q_4 markiert. Das System hat 18 erreichbare Markierungen.

2.0.5 Das Aktion/Reaktion-Protokoll

Zwei Agenten müssen Informationen austauschen, um eine Aufgabe zu lösen. Wenn ein Agent dem anderen eine Frage stellt, dann muß er warten, bis eine Antwort kommt, bevor er weitermachen kann. Man möchte ein geeignetes Protokoll für den Nachrichtaustausch entwerfen. Insbesondere muß gewährleistet werden, daß es nicht zu einer Situation kommen kann, in der beide Prozesse auf eine Antwort vom anderen Prozeß warten.
Ein erster Lösungsversuch wird in Abbildung 2.7 gezeigt. Die Stellung von Fragen wird durch die Transitionen $Aktion$ modelliert, und die Antworten mit den Transitionen $Reaktion$. Dieses System kann aber eine Verklemmung erreichen: beide Prozeße stellen gleichzeitig eine Frage, und warten vergeblich auf eine Antwort. Eine solche Situation nennen wir ein *crosstalk*. Abbildung 2.8 zeigt einen zweiten Versuch. Wenn ein Prozeß eine crosstalk Situation erkennt, dann beantwortet er die Frage des Partners und wartet weiter auf eine Antwort auf seine eigene Frage. Dieses System kann nicht verklemmen (zeigen Sie es!), aber hat das folgende Problem: ein unkooperativer Prozeß kann immer wieder Antworten auf seine Frage bekommen, während er keine Frage des Partners beantwortet. Die Lösung ist also verklemmungsfrei, aber *unfair*.

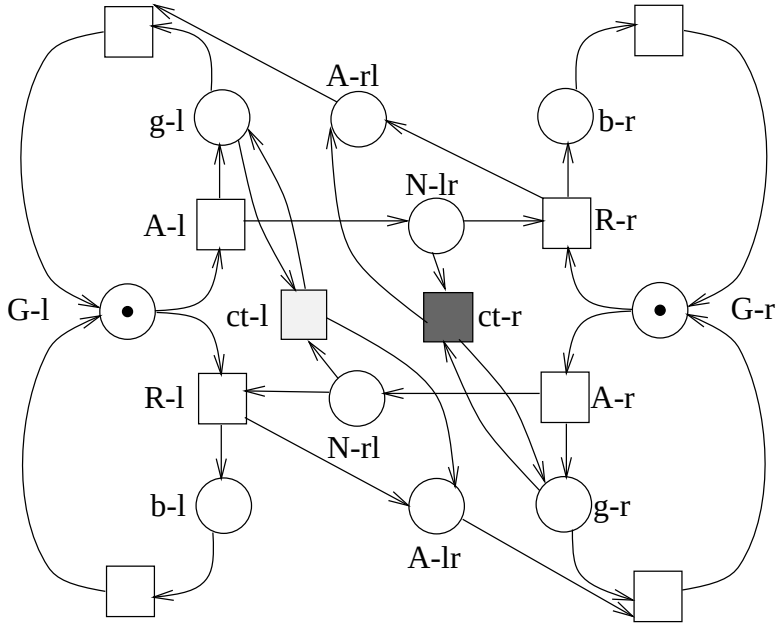


Abbildung 2.8: Der zweite Versuch

Der nächste Versuch (Abbildung 2.9) ist fair. Wenn ein Prozess eine crosstalk Situation erkennt, dann antwortet er auf die Frage des Partners, geht aber in einen Zustand über, in dem er nur bereit ist, eine Antwort auf seine eigene Frage zu bekommen. Leider gibt es wieder eine Verklemmung (können Sie sie finden?). Der letzte Versuch (Abbildung 2.10) ist verklemmungsfrei und fair. Das Protokoll arbeitet nun in Runden. Eine 'gute' Runde besteht aus einer Frage und ihrer Antwort. In einer 'schlechten' Runde stellen beide Prozesse Fragen, und eine crosstalk Situation wird erreicht. Die Runde geht dann so weiter: beide Prozesse erkennen die crosstalk Situation, schicken einander ein Signal 'Rundenende' zu, warten auf das Signal des Partners und gehen dann in ihre Anfangszustände über.
Diese Lösung ist nicht perfekt: im schlimmsten Fall kann es nur schlechte Runden geben, und es werden überhaupt keine Fragen beantwortet.

2.1 Varianten des Modells

Definition 2.1.1 (Netze mit Kapazitäten)
Ein Netz mit Kapazitäten $N = (S, T, F, K)$ besteht aus einem Netz (S, T, F) und einer Abbildung $K : S \to \mathbb{N}$
Eine Transition t ist aktiviert von einer Markierung M von N
– $M(s) \geq 1$ für jede Stelle $s \in {}^\bullet t$ und
– $M(s) < K(s)$ für jede Stelle $s \in t^\bullet \setminus {}^\bullet t$

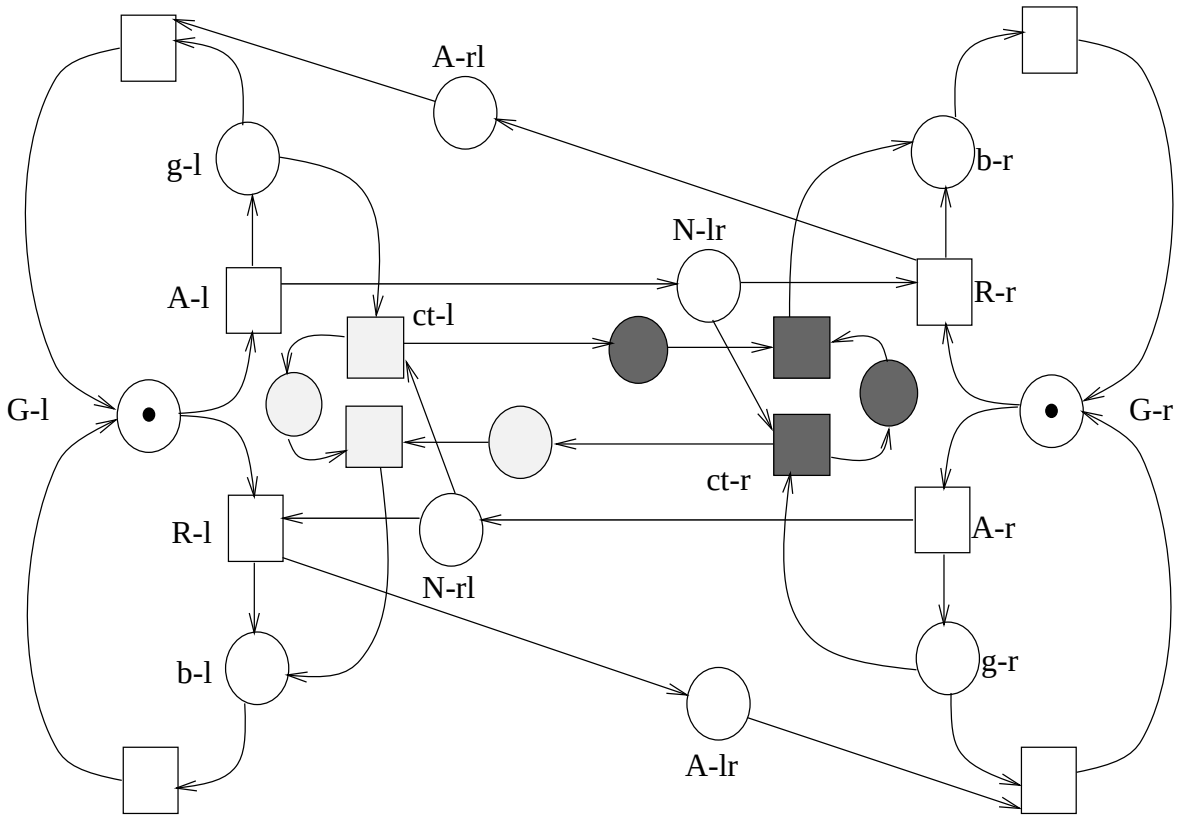


Abbildung 2.9: Der dritte Versuch

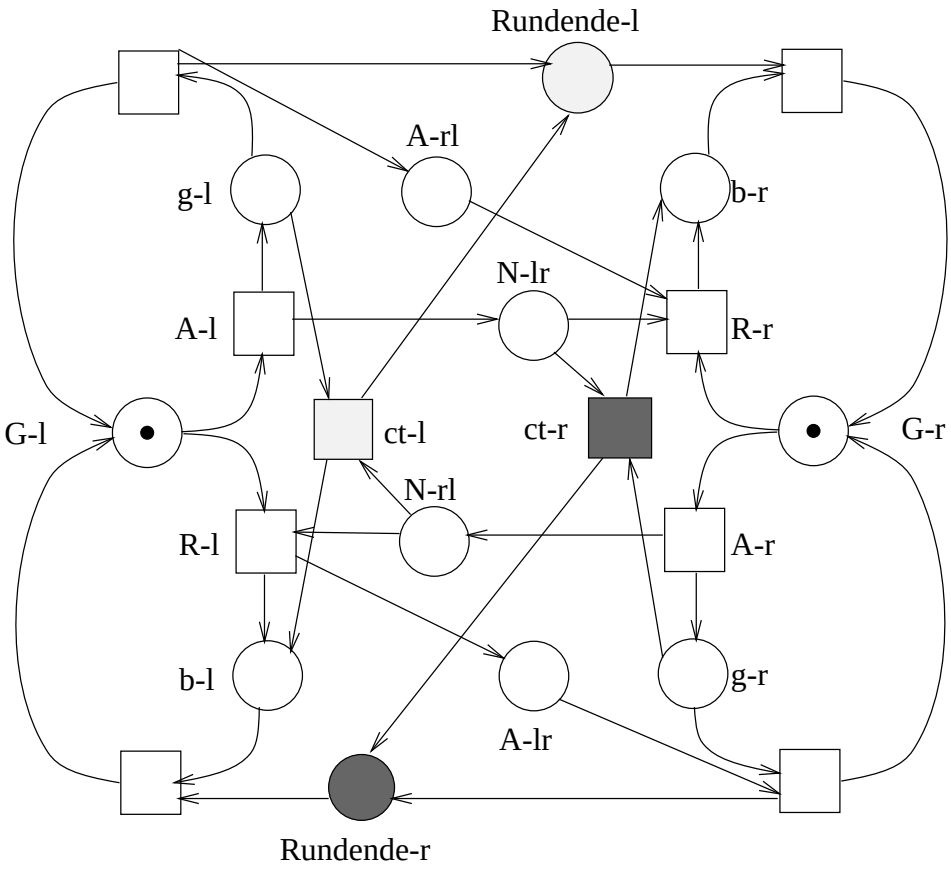


Abbildung 2.10: Der letzte Versuch

Die Begriffe Schaltung, Netzsystem mit Kapazitäten etc. werden wie für herkömmliche Netze definiert.

Definition 2.1.2 (Netze mit Kantengewichten)
Ein Netz mit Kantengewichten $N = (S, T, W)$ besteht aus einer Menge von Stellen und Transitionen und aus einer Gewichtsfunktion $W : (S \times T) \cup (T \times S) \to \mathbb{N}$. Eine Transition t ist aktiviert von einer Markierung M von N, wenn $M(s) \geq W(s,t)$ für alle $s \in S$. Falls t aktiviert ist, kann sie schalten und überführt M in die Markierung M' definiert durch

$$M'(s) = M(s) + W(t,s) - W(s,t)$$

für jede Stelle s. Andere Begriffe werden wie für herkömmliche Netze definiert.

Das Netzsystem mit Kantengewichten auf Abbildung 2.11 modelliert eine Lösung des "'readers and writers" Problem. Eine Menge von Prozessen haben Zugriff zu einer Datenbank. Mehrere Prozesse dürfen simultan Daten lesen. Ein Prozess darf aber nur schreiben, wenn alle anderen Prozesse weder lesen noch schreiben.
Aufgabe: Ändern Sie das System so, daß es für die Leser unmöglich ist, zu verhindern, daß ein Prozeß schreibt.

2.2 Wichtige Systemeigenschaften und Analyseprobleme

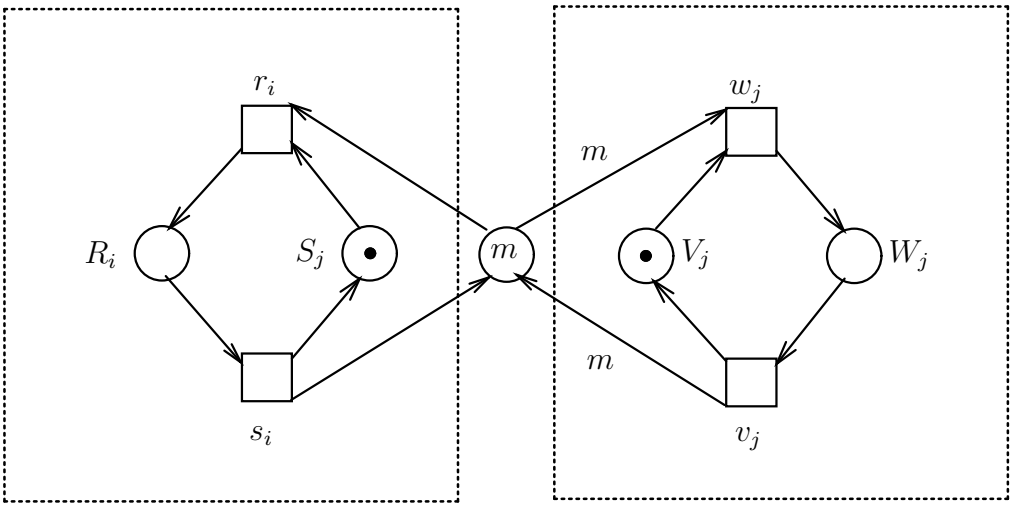
Wir definieren nun die Eigenschaften von Systemen, an deren Verifikation wir interessiert sind. Wir nehmen an, daß Netze mindestens eine Stelle und eine Transition haben.

Definition 2.2.1 (Systemeigenschaften)
Sei (N, M_0) ein Netzsystem.
(N, M_0) heißt *verklemmungsfrei*, wenn jede erreichbare Markierung wenigstens eine Transition aktiviert (d.h. keine erreichbare Markierung ist tot).
(N, M_0) heißt *lebendig*, wenn für jede erreichbare Markierung M und jede Transition t eine Markierung $M' \in [M\rangle$ existiert, die t aktiviert. (Grob gesagt: jede Transition t kann immer wieder schalten).
(N, M_0) heißt *beschränkt*, wenn es für jede Stelle s eine Zahl b gibt, so daß $M(s) \leq b$ für jede erreichbare Markierung M. M_0 ist eine beschränkte Markierung von N, wenn (N, M_0) beschränkt ist. Die Schranke einer Stelle S in einem beschränkten System (N, M_0) ist die Zahl

$$max\{M(s) \mid M \in [M_0\rangle\}$$

(N, M_0) heißt *b-beschränkt*, wenn keine Stelle eine größere Schranke als b hat.

In der Vorlesung werden wir die folgenden Probleme untersuchen:
Verklemmungsfreiheit: ist ein gegebenes Netzsystem (N, M_0) verklemmungsfrei?
Lebendigkeit: ist ein gegebenes Netzsystem (N, M_0) lebendig?
Beschränkheit: ist ein gegebenes Netzsystem (N, M_0) beschränkt?
b-Beschränkheit: gegeben $b \in \mathbb{N}$ und (N, M_0), ist (N, M_0) b-beschränkt?



R_i:	Prozeß i liest	W_i:	Prozeß j schreibt
S_i:	Prozeß i liest nicht	V_j:	Prozeß j schreibt nicht
r_i:	Prozeß i fängt an zu lesen	w_j:	Prozeß j fängt an zu schreiben
s_i:	Prozeß i hört auf zu lesen (stop)	v_j:	Prozeß j hört auf zu schreiben (stop)
m Leser		n Schreiber	

Abbildung 2.11: Readers and writers

Erreichbarkeit: gegeben ein Netsystem (N, M_0) und eine Markierung M von N, ist M erreichbar aus M_0?
Wir zeigen einige einfache Beziehungen zwischen diesen Problemen:

Proposition 2.2.2

(1) (N, M_0) *ist lebendig* $\Rightarrow$ (N, M_0) *ist verklemmungsfrei.*

(2) (N, M_0) *ist beschränkt* $\Rightarrow$ (N, M_0) *ist* b*-beschränkt für eine Zahl* b*.*

(3) (N, M_0) *ist beschränkt* $\Leftrightarrow$ (N, M_0) *hat endlich viele erreichbare Markierungen.*

Beweis. (1) und (2) folgen ummittelbar aus den Definitionen. (3) folgt aus den Definitionen und aus der Tatsache, daß ein Netzsystem nur endlich viele Stellen und Transitionen hat. □

Wir werden auch gelegentlich die folgende Definition benutzen

Definition 2.2.3 (Wohlgeformte Netze)
Ein Netz N ist wohlgeformt, wenn eine lebendige und beschränkte Markierung von N existiert.

und das folgende Problem betrachten:
Wohlgeformheit: ist ein gegebenes Netz N Wohlgeformt?

Teil II

Analysemethoden

Im Kapitel 3 wird gezeigt (meistens ohne Beweise), daß die Probleme **Verklemmungsfreiheit**, **Lebendigkeit**, **Beschränktheit**, b-**Beschränktheit** und **Erreichbarkeit** entscheidbar sind. Die Algorithmen, die diese Probleme lösen, haben aber eine sehr hohe Komplexität, und Ergebnisse der Komplexitätstheorie zeigen, daß keine effizienten Algorithmen für diese Probleme existieren.
Weil Effizienz unentberhlich für die praktische Anwendung ist, muß man sich mit Halbentscheidungsalgorithmen für allgemeine Systeme oder mit Entscheidungsalgorithmen für eingeschränkte Systemklassen zufrieden geben. Unter einem effizienten Halbentscheidungsalgorithmus für die Eigenschaft P verstehen wir einen schnellen positiven (negativen) Test für P: wenn die Eingabe den Test besteht, dann gilt P (nicht P), wenn nicht, dann kann man nichts über P sagen.
Kapitel 4 ist den Halbentscheidungsalgorithmen gewidmet. In Kapitel 5 werden effiziente Entscheidungsalgoritmen für drei Systemklassen entwickelt: S-, T-, und Free-Choice-Systeme.

Kapitel 3

Allgemeine Algorithmen

3.1 Ein Algorithmus für das Beschränktheitsproblem

Das Problem b-**Beschränktheit** ist offensichtlich entscheidbar: wenn die Eingabe (N, M_0) n Stellen hat, dann gibt es n^{b+1} b-beschränkte Markierungen vom Netz N. Um b-**Beschränktheit** zu entscheiden, kann man den Erreichbarkeitsgraphen von (N, M_0) schrittweise konstruieren, bis entweder die Konstruktion terminiert oder bis eine erreichbare Markierung gefunden wird, die nicht b-beschränkt ist.
Beschränktheit ist offensichtlich rekursiv aufzählbar: man konstruiert wieder den Erreichbarkeitsgraph des Systems, Schritt für Schritt. Wenn das System beschränkt ist, dann terminiert die Konstruktion, weil es nur endlich viele erreichbare Markierungen gibt. Für unbeschränkte Systeme terminiert die Konstruktion nicht, und man erhält keine Antwort.
Wir zeigen nun, daß **Beschränktheit** nicht nur rekursiv aufzählbar, sondern auch entscheidbar ist. Dafür brauchen wir das folgende wichtige Lemma:

Lemma 3.1.1 *Jede unendliche Teilmenge $\mathcal{A} \subseteq \mathbb{N}^k$ enthält eine unendliche Kette $A_1 \leq A_2 \leq A_j \leq \ldots$.*

Beweis. Durch Induktion über k
Basis: $k = 1$. Trivial
Schritt: $k > 1$. Für $A \in \mathcal{A}$, $A = (a^1, \ldots, a^k)$ definiere $A' = (a^1, \ldots, a^{k-1})$. Wir schreiben $A = (A' \mid a^k)$.
Aus der Induktionsvoraussetzung folgt:

- Es gibt eine unendliche Sequenz $A_1 A_2 A_3 \ldots$ mit $A'_1 \leq A'_2 \leq A'_3 \ldots$.
- Es gibt eine unendliche Kette $a^k_{i_1} \leq a^k_{i_2} \leq a^k_{i_3} \ldots$.

Damit gilt $A_{i_1} \leq A_{i_2} \leq A_{i_3} \ldots$ und wir sind fertig. □

Nun können wir die Unbeschränktheit eines Systems folgendermaßen charakterisieren:

Satz 3.1.2 *(N, M_0) ist unbeschränkt gdw. es Markierungen M und L gibt, mit $L \neq 0$ und $M_0 \xrightarrow{*} M \xrightarrow{*} (M + L)$*

Beweis.
($\Leftarrow$) : Aus dem Monotonielemma folgt

$$M_1 \xrightarrow{*} (M_1 + L) \xrightarrow{*} (M_1 + 2 \cdot L) \xrightarrow{*} \ldots$$

Damit ist die Menge $[M_0\rangle$ unendlich und (N, M_0) unbeschränkt.

($\Rightarrow$) Wenn (N, M_0) unbeschränkt ist, dann ist die Menge $[M_0\rangle$ unendlich. Es gibt damit eine unendliche Schaltsequenz $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_1} M_2 \ldots$, in der alle vorkommenden Markierungen unterschiedlich sind, denn jede Markierung hat endlich viele Nachfolger (Königs Lemma). Mit Lemma 3.1.1 gibt es M_i und M_j, $i \neq j$, so daß $M_0 \xrightarrow{*} M_i \xrightarrow{*} M_j$ und $M_i \leq M_j$. Definiere $M \equiv M_i$ und $L \equiv M_j - M_i$. □

Satz 3.1.3 *Es gibt einen Algorithmus, der entscheidet, ob ein gegebenes System (N, M_0) beschränkt ist.*

Beweis. Der Algorithmus konstruiert breadth-first immer größere endliche Teile des Erreichbarkeitsgraphen, und testet nach jeder neuen Kante, ob es eine Schaltsequenz $M_0 \xrightarrow{*} M \xrightarrow{*} (M + L)$ mit $L \neq 0$ gibt. Der Algorithmus terminiert entweder wenn eine solche Sequenz gefunden wird, oder wenn keine neue Kante hinzugefügt werden kann. Im ersten Fall antwortet der Algorithmus "unbeschränkt", und im zweiten Fall "beschränkt".
Wenn (N, M_0) beschränkt ist, dann ist der erste Terminierungsfall unmöglich wegen Satz 3.1.2. Weil es nur endlich viele erreichbare Markierungen gibt, kommt aber der zweite Fall irgendwann einmal vor. Also, der Algorithmus terminiert und gibt die korrekte Antwort aus.
Wenn (N, M_0) unbeschränkt ist, dann gibt es unendlich viele erreichbare Markierungen, und der zweite Terminierungsfall ist unmöglich. Wegen Satz 3.1.2 kommt aber der erste Terminierungsfall irgendwann einmal vor. Also, der Algorithmus terminiert und gibt die korrekte Antwort aus. □

3.2 Algorithmen für die restlichen Probleme

Die Entscheidbarkeit des Erreichbarkeitsproblems war ca. 10 Jahre lang offen, und wurde von Prof. E.W. Mayr (ja, derselbe Prof. Mayr, der die Vorlesungen "Effiziente Algorithmen" und "Komplexität" hält) im Jahre 1980 in seiner Dissertation bewiesen. Leider ist der Algorithmus zu kompliziert für diese Vorlesung.
Wir werden nun zeigen, daß **Verklemmungsfreiheit** sich auf **Erreichbarkeit** reduzieren laßt. D.h., wir zeigen, daß wenn es einen Algorithmus für **Erreichbarkeit** gibt,

dann gibt es auch einen Algorithmus für **Verklemmungsfreiheit**. Zusammen mit dem Ergebniss von Prof. Mayr wird so gezeigt, daß **Verklemmungsfreiheit** entscheidbar ist.
Der Beweis erfolgt in zwei Schritten. Wir betrachten das folgende Hilfsproblem

> **P**: Gegeben ist ein Netzsystem (N, M_0) und eine Untermenge R von Stellen von N. Gibt es eine erreichbare Markierung M mit $M(s) = 0$ für alle $s \in R$?

und zeigen, daß **Verklemmungsfreiheit** auf **P**, und **P** auf **Erreichbarkeit** reduzierbar ist.

Satz 3.2.1 Verklemmungsfreiheit *ist auf* **P** *reduzierbar.*

Beweis. Sei (N, M_0) ein Netzsystem, und sei $N = (S, T, F)$. Definiere

$$\mathcal{S} = \{R \subseteq S \mid \forall t \in T : {}^\bullet t \cap R \neq \emptyset\}$$

d.h., ein Element von $\mathcal{S}$ enthält für jede Transition t mindestens eine Stelle im Vorbereich von t. Die folgenden zwei Aussagen sind ummittelbare Konsequenzen der Definition von $\mathcal{S}$:

(1) $\mathcal{S}$ ist endlich.

(2) Eine Markierung M von N ist tot, gdw. die Menge der Stellen, die von M nicht markiert werden, Element von $\mathcal{S}$ sind.

Nehmen wir nun an, daß ein Algorithmus existiert, der **P** entscheidet. Dann können wir für jedes Element R von S die Frage beantworten, ob es eine aus M_0 erreichbare Markierung M gibt mit $M(s) = 0$ für alle $s \in R$. Aus (2) folgt: (N, M_0) ist verklemmungsfrei, gdw. die Antwort auf *alle* diese Fragen "nein" lautet. Wegen (1) gibt es nur endlich viele solcher Fragen. Also, **Verklemmungsfreiheit** ist entscheidbar. □

Satz 3.2.2 P *ist auf* **Erreichbarkeit** *reduzierbar.*

Beweis. Sei (N, M_0) ein Netzsystem mit $N = (S, T, F)$, und sei R eine Menge von Stellen von N. Wir konstruieren ein neues System (N', M_0'), indem wir neuen Stellen, Transitionen, Kanten und Marken zu (N, M_0) hinzufügen. Das machen wir in zwei Etappen (siehe Abbildung 3.1):

- Addiere neue Stellen s_0 und r_0. Stelle eine Marke auf s_0.
- Addiere eine Transition t_0 und Kanten (s_0, t_0) und (t_0, r_0).
- Für jede Transition $t \in T$ addiere zwei Kanten (s_0, t) und (t, s_0).

Solange s_0 markiert bleibt, können die Transitionen von T ungehindert schalten. Die Transition t_0 kann aber jederzeit schalten, und ab diesem Punkt werden alle Transitionen von T "tot", d.h., das System (N, M_0) wird "gefroren".

- Für jede Stelle $s \in S \backslash R$ addiere eine neue Transition t_s und Kanten $(s, t_s), (r_0, t_s), (t_s, r_0)$.

Wenn eine Marke auf r_0 liegt, dann können die Transitionen t_s schalten. Diese Transitionen "entleeren" die Stellen in $S \setminus R$.
Damit ist die Definition von (N', M_0') beendet.
Sei M_{r_0} die Markierung von N', die eine Marke auf r_0 legt, und sonst keine. Wir haben

(1) Wenn es eine erreichbare Markierung M von (N, M_0) gibt, die keine Stelle von R markiert, dann ist M_{r_0} erreichbar in (N', M_0').

Um M_{r_0} zu erreichen, schalte erst Transitionen von T, bis M erreicht wird. Schalte dann die Transition t_0, und anschließend Transitionen t_s bis keine Marken mehr auf den Stellen von S liegen.

(2) Wenn M_{r_0} erreichbar in (N', M_0') ist, dann gibt es eine erreichbare Markierung M von (N, M_0), die keine Stelle von R markiert.

M_{r_0} kann nur erreicht werden, wenn die Transition t_0 schaltet und vor dem Schalten dieser Transition keine Marken auf Stellen von R liegen (die Stellen von R können später nicht entleert werden). M ist die Markierung von N vor dem Schalten von t_0.

Aus (1) und (2) folgt: um zu entscheiden, ob es eine erreichbare Markierung M von (N, M_0) gibt, die keine Stelle von R markiert, reicht es, das System (N', M_0') zu konstruieren, und dann zu entscheiden, ob die Markierung M_{r_0} erreichbar ist. Also, wenn es einen Algorithmus für **Erreichbarkeit** gibt, dann gibt es auch einen für **P**. □

Mit Hilfe derselben Technik kann man zeigen, daß auch **Lebendigkeit** auf **Erreichbarkeit** reduzierbar ist. Der Beweis ist aber etwas zu kompliziert für diese Vorlesung.

3.3 Komplexität

Wir stellen uns die Frage, wieviel Zeit und oder Platz brauchen die Algorithmen der vorigen Sektion. Die Antwort ist: leider zu viel. Der Algorithmus für **Beschränkheit** braucht exponentiell viel Platz in der Größe des Netzsystems, und man weiß, daß keine wesentlich besseren Algorithmen existieren. Der Algorithmus für **Erreichbarkeit** ist noch aufwendiger: keine primitiv rekursive Funktion ist eine obere Schranke seiner Platzsverbrauchs. Um zu verstehen, was das bedeutet, definieren wir induktiv die Funktionen $exp_k(x)$ durch

- $exp_0(x) = x$;
- $exp_{k+1}(x) = 2^{exp_k(x)}$.

Der worst-case Platzverbrauch (und Zeitverbrauch) des Erreichbarkeitsalgorithmus wächst schneller als $\exp_k$ *für alle* $k \geq 0$!!

3.4 Algorithmen für beschränkte Netzsysteme

In vielen Fällen aus der Praxis ist es leicht zu zeigen, daß ein Netzmodell beschränkt ist. In diesem Fall ist die Menge der erreichbaren Markierungen endlich, und der Erreichbarkeitsgraph kann prinzipiell berechnet und gespeichert werden. Wenn der Erreichbarkeitsgraph vorhanden ist, dann sind **Beschränkheit**, b-**Beschränkheit** und **Erreichbarkeit** trivial. Wir zeigen nun, daß **Verklemmungsfreiheit** und **Lebendigkeit** sich auch leicht lösen lassen.
Sei $G = (V, E)$ der Erreichbarkeitsgraph eines Systems (N, M_0). Wir definieren die Relation $\overset{*}{\longleftrightarrow} \subseteq V \times V$ [1] auf die folgende Weise: $M \overset{*}{\longleftrightarrow} M'$ gdw. $M \overset{*}{\longrightarrow} M'$ und $M' \overset{*}{\longrightarrow} M$.
$\overset{*}{\longleftrightarrow}$ ist offensichtlich eine Äquivalenzrelation auf V. Jede Äquivalenzklasse $\emptyset \neq V' \subseteq V$ von $\overset{*}{\longleftrightarrow}$ definiert mit $E' = E \cap (V' \times V)$ eine starke Zusammenhangskomponente (V', E') von G.
Starke Zusammenhangskomponenten sind durch die Relation $<$ geordnet: $(V', E') < (V'', E'')$, wenn $V' \neq V''$ und $\forall M' \in V',\ M'' \in V'' : M'' \in [M'\rangle$.

Definition 3.4.1 (Maximale starke Zusammenhangskomponente)
Eine starke Zusammenhangskomponente heißt maximal, wenn sie maximal bezüglich der Ordnung $<$ ist.

Proposition 3.4.2 *Sei (N, M_0) ein System*

1. *(N, M_0) ist verklemmungsfrei gdw. jeder Knoten seines Erreichbarkeitsgraphen einen Nachfolger hat.*
2. *Falls das System (N, M_0) beschränkt ist, ist es lebendig gdw. in jeder maximalen starken Zusammenhangskomponente des Erreichbarkeitsgraphen für jede Transition t von N eine Markierung existiert, die t aktiviert.*

Beweis. Folgt ummittelbar aus den Definitionen von Verklemmungsfreiheit und Lebendigkeit. □

Mit Hilfe dieser Charakterisierung lassen sich Algorithmen für **Verklemmungsfreiheit** und **Lebendigkeit** in beschränkten Petrinetzen ableiten, die linear in der Größe des Erreichbarkeitsgraphen sind (Aufgabe: geben Sie einen solchen Algorithmus für **Lebendigkeit**.). Leider kann die Anzahl der erreichbaren Markierungen exponentiell in der Größe des Netzes wachsen. Deswegen sind die Algorithmen, die den Erreichbarkeitsgraphen konstruieren, zwar sehr nutzlich, aber keine endgültige Lösung der Verifikationsprobleme.

[1] Zur Erinnerung: V ist die Menge der erreichbaren Markierungen von (N, M_0)

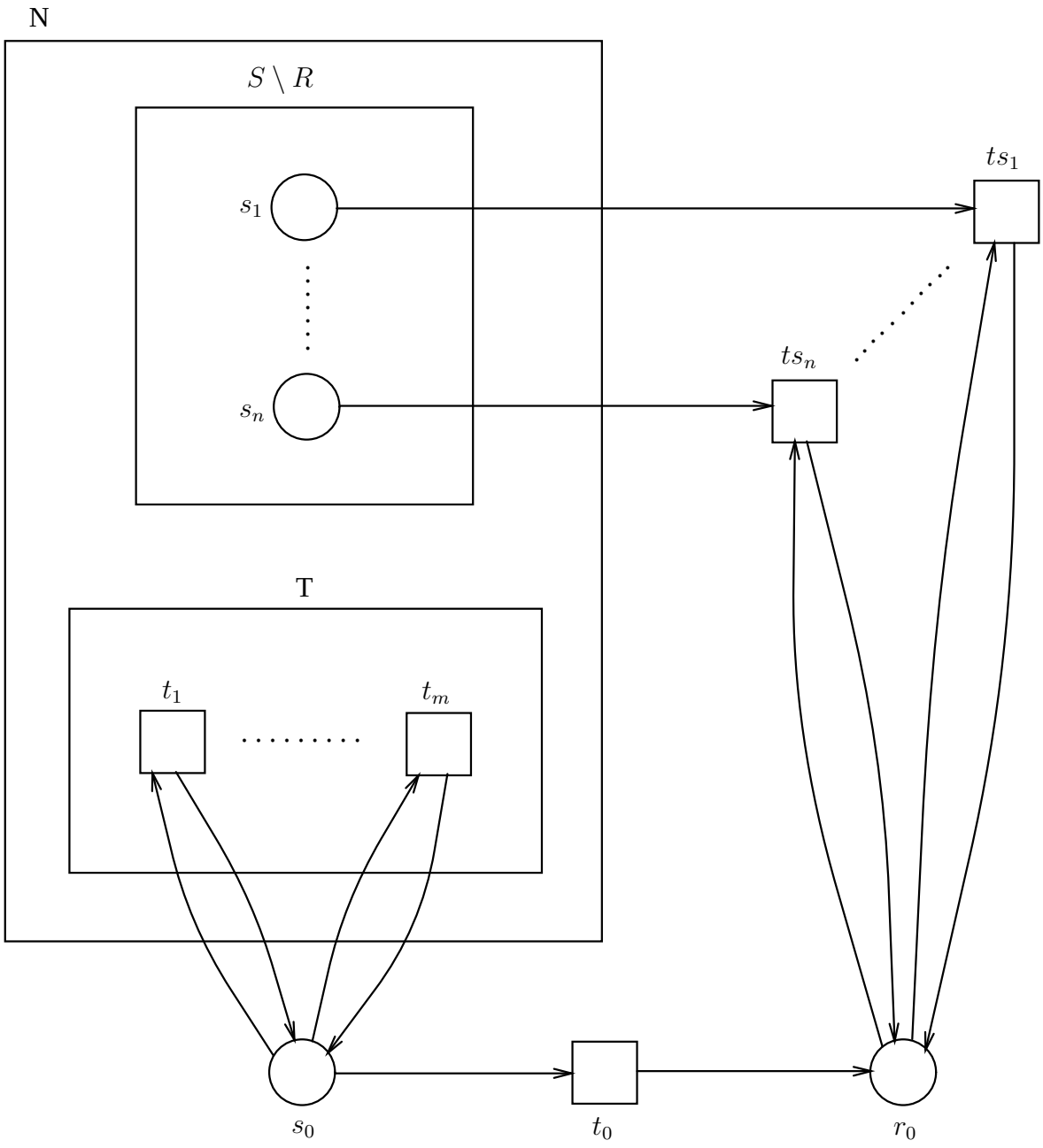


Abbildung 3.1: Konstruktion vom Satz 3.2.2

Kapitel 4

Halbentscheidungsmethoden

4.1 Lineare Algebra und lineare Programmierung

In den nächsten zwei Sektionen werden wir Systeme von linearen Gleichungen und Ungleichungen mit integer Koeffizienten konstruieren, die partielle Informationen über unsere Verifikationsprobleme liefern. Wir werden Aussagen der Art "wenn das Gleichungssystem $A \cdot X \leq b$ eine rationale Lösung hat, dann ist das Netzsystem (N, M_0) beschränkt" (hinreichende Bedingung), oder "wenn die Markierung M erreichbar ist, dann hat das Gleichungssystem $A \cdot X = b$ eine natürliche Lösung" (notwendige Bedingung) beweisen. Diese Aussagen führen ummittelbar zu Halbentscheidungsalgorithmen für **Beschränkheit** und die restlichen Verifikationsprobleme. Die Kömplexität dieser Algorithmen hängt von der Komplexität ab, Lösungen für die verschiedenen Gleichungssysteme zu finden.
Die *Größe* eines Gleichungssystems $A \cdot X = b$ oder $A \cdot X \leq b$ mit $A = (a_{ij})_{i=1,\ldots n, j=1,\ldots,m}$ und $b = (b_j)_{j=1,\ldots,m}$ definieren wir als

$$\sum\{log_2|a_{ij}| \mid 1 \leq i \leq n, 1 \leq j \leq m\} + \sum\{log_2|b_j| \mid 1 \leq j \leq m\}$$

Das Problem, zu entscheiden, ob $A \cdot X = b$ eine

- rationale Lösung hat, ist in polynomieller Zeit lösbar.
- ganzzählige Lösung hat, ist in polynomieller Zeit lösbar.
- natürliche Lösung hat ist NP-vollständig.

Das Problem, zu entscheiden, ob $A \cdot X \leq b$ eine

- rationale Lösung hat, ist in polynomieller Zeit lösbar.[1]
- ganzzählige Lösung hat ist NP-vollständig.

[1] In der Praxis werden diese Problem mit dem Simplex-Algorithmus gelöst, der zwar exponentielle worst-case Zeitkomplexität hat, aber sehr effizient für "normale" Instanzen.

- natürliche Lösung hat ist NP-vollständig.

Gegeben eine lineare Funktion $f(X) = c_1x_1 + \ldots c_m$ kann man mit derselben Komplexität auch bestimmen, ob es eine Lösung X_{op} gibt, die $f(X)$ maximiert und, wenn ja, den Wert $f(X_{op})$.

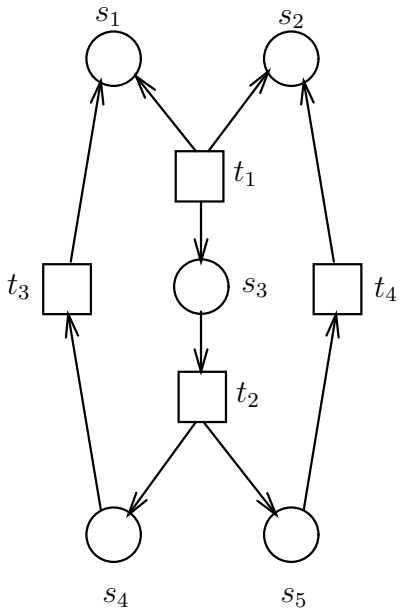
4.2 Die Markierungsgleichung

Definition 4.2.1 (Die Inzidenzmatrix)
Sei $N = (S, T, F)$ ein Netz. Die Inzidenzmatrix $\mathbf{N} : (S \times T) \to \{-1, 0, 1\}$ ist definiert durch

$$\mathbf{N}(s,t) = \begin{cases} 0 & \text{falls} \quad (s,t) \notin F \text{ und } (t,s) \notin F \text{ oder} \\ & \qquad\quad (s,t) \in F \text{ und } (t,s) \in F \\ -1 & \text{falls} \quad (s,t) \in F \text{ und } (t,s) \notin F \\ 1 & \text{falls} \quad (s,t) \notin F \text{ und } (t,s) \in F \end{cases}$$

Die Spalte $\mathbf{N}(-,t)$ wird mit $\mathbf{t}$ bezeichnet. Die Zeile $\mathbf{N}(s,-)$ wird mit $\mathbf{s}$ bezeichnet.

Beispiel:



	t_1	t_2	t_3	t_4
s_1	-1	0	1	0
s_2	-1	0	0	1
s_3	1	-1	0	0
s_4	0	1	-1	0
s_5	0	1	0	-1

Definition 4.2.2 (Parikh-Vektor einer Transitionssequenz)
Sei $N = (S, T, F)$ ein Netz und sei σ eine endliche Sequenz von Transitionen. Der Parikh-Vektor $\vec{\sigma} : T \to \mathbb{N}$ von σ wird definiert durch

$$\vec{\sigma}(t) = \text{ Anzahl der Vorkommnisse von } t \text{ in } \sigma$$

Lemma 4.2.3 (Das Markierungsgleichungslemma)
Sei N ein Netz und sei $M \xrightarrow{\sigma} M'$ eine Schaltfolge von N. Dann $M' = M + \mathbf{N} \cdot \vec{\sigma}$.

Beweis. Durch Induktion über die Länge von σ.
Basis: $\sigma = \epsilon$. Dann $M = M'$ und $\vec{\sigma} = 0$
Schritt: $\sigma = \tau t$ für eine Sequenz τ und Transition t. Sei $M \xrightarrow{\tau} L \xrightarrow{t} M'$. Wir haben

$$\begin{array}{lll} M' & = L + \mathbf{t} & \text{(Definition von } \mathbf{t}) \\ & = L + \mathbf{N} \cdot \vec{t} & \text{(Definition von } \vec{t}) \\ & = M + \mathbf{N} \cdot \vec{\tau} + \mathbf{N} \cdot \vec{t} & \text{(Induktionsvoraussetzung)} \\ & = M + \mathbf{N} \cdot (\vec{\tau} + \vec{t}) & \\ & = M + \mathbf{N} \cdot \vec{\tau t} & \text{(Definition von Parikh-Vektor)} \\ & = M + \mathbf{N} \cdot \vec{\sigma} & (\sigma = \tau t) \end{array}$$

□

Beispiel. Im vorigen Netz haben wir $(11000) \xrightarrow{t_1 t_2 t_3} (10001)$, und es gilt

$$\begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{array} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Die Markierung, die erreicht wird, wenn eine Sequenz σ aus einer Markierung M schaltet, ist *nur vom Parikh-Vektor* $\vec{\sigma}$ abhängig. D.h., wenn M zwei Schaltsequenzen σ und τ aktiviert mit $\vec{\sigma} = \vec{\tau}$, dann führen beide σ und τ zur selben Markierung.

Definition 4.2.4 (Die Markierungsgleichung)
Die Markierungsgleichung eines Netzsystems (N, M_0) mit $N = (S, T, F)$ ist $M = M_0 + \mathbf{N} \cdot X$ mit Variablen M und X.

Die Markierungsgleichung führt zu den folgenden Halbentscheidungsalgorithmen für **Beschränkheit**, b-**Beschränkheit**, (Nicht)-**Erreichbarkeit**, und **Verklemmungsfreiheit**:

Proposition 4.2.5 (Eine hinreichende bedingung für Beschränkheit)
Sei (N, M_0) ein System. Wenn das Optimierungsproblem

$$\begin{array}{ll} \textit{maximize} & \sum_{s \in S} M(s) \\ \textit{subject to} & M = M_0 + \mathbf{N} \cdot X \end{array}$$

eine optimale Lösung hat, dann ist (N, M_0) beschränkt.

Beweis. Sei n die optimale Lösung des Problems. Dann gilt $n \geq \sum_{s \in S} M(s)$ für alle Markierungen M, für die es einen Vektor X gibt mit $M = M_0 + \mathbf{N} \cdot X$. Aus Lemma 4.2.3 folgt $n \geq \sum_{s \in S} M(s)$ für alle *erreichbaren* Markierungen M, und somit $n \geq M(s)$ für jede erreichbare Markierung M und jede Stelle s. □

Aufgabe: Ändern Sie den obigen Algorithmus um zu testen, ob eine Stelle n-beschränkt ist.

Proposition 4.2.6 (Eine hinreichende bedingung für Nichterreichbarkeit)
Sei (N, M_0) ein System und sei L eine Markierung von N. Wenn die Gleichung

$$L = M_0 + \mathbf{N} \cdot X \qquad (\textit{nur } X \textit{ als Variable})$$

keine Lösung hat, dann ist L nicht von M_0 erreichbar.

Beweis. Folgt ummittelbar aus Lemma 4.2.3. □

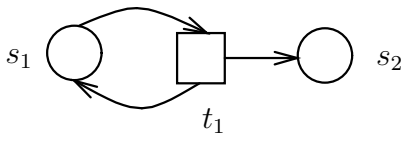
Proposition 4.2.7 (Eine hinreichende bedingung für Verklemmungsfreiheit)
Sei (N, M_0) ein 1-beschränktes System mit $N = (S, T, F)$. Wenn das folgende System von Gleichungen und Ungleichungen keine Lösung hat, dann ist (N, M_0) verklemmungsfrei.

$$M = M_0 + \mathbf{N} \cdot X$$
$$\sum_{s \in \cdot t} M(s) < |{}^\bullet t| \textit{ für jede Transition } t.$$

Beweis. Wir zeigen: wenn es eine tote erreichbare Markierung M gibt, dann ist M eine Lösung des Systems von Gleichungen und Ungleichungen. Aus Lemma 4.2.3 und der Erreichbarkeit von M folgt, daß es einen Vektor X gibt mit $M = M_0 + \mathbf{N} \cdot X$. Da (N, M_0) 1-beschränkt ist, gilt $M(s) \leq 1$ für jede Stelle s. Sei t eine beliebige Transition. Da M die Transition t nicht aktiviert, gilt $M(s) = 0$ für mindestens eine Stelle $s \in {}^\bullet t$. Da M keine Transition aktiviert, gilt $\sum_{s \in \cdot t} M(s) < |{}^\bullet t|$. □

Bemerkung: Die Umkehrung dieser Propositionen gilt **nicht** (und deswegen handelt es sich um Halbentscheidungsalgorithmen). Gegenbeispiele sind:

- Zu Proposition 4.2.5:



	t_1
s_1	0
s_2	1

  (N, M_0) ist beschränkt aber

$$\begin{pmatrix} 0 \\ n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot n$$

  für alle n (d.h., die Markierungsgleichung hat eine Lösung für alle Markierungen der Form $(0, n)$).

- Zu Proposition 4.2.6:

  Petersons Algorithmus: die Markierung $(p_2, m1 = true, m2 = true, q_3)$ ist nicht erreichbar, aber es gibt eine Lösung der Markierungsgleichung (Aufgabe: geben Sie ein kleineres Gegenbeispiel an).

- Zu Proposition 4.2.7:

  Petersons Algorithmus plus eine zusätzliche Transition t mit ${}^\bullet t = \{p_3, q_3\}$ und $t^\bullet = \emptyset$. Obwohl das System verklemmungsfrei ist, gibt es eine Lösung der Markierungsgleichung für die Markierung $(m1 = true, m2 = true)$. Diese Lösung erfüllt auch die Ungleichungen von Proposition 4.2.7 (Aufgabe: geben Sie ein kleineres Gegenbeispiel an).

4.3 S- und T-Invarianten

4.3.1 S-Invarianten

Definition 4.3.1 (S-Invariante)
Sei $N = (S, T, F)$ ein Netz. Ein Vektor $I : S \to \mathbb{Q}$ heißt S-invariante, wenn $I \cdot \mathbf{N} = 0$.

Proposition 4.3.2 (Fundamentale Eigenschaft von S-Invarianten)
Sei (N, M_0) ein System und I eine S-Invariante von N. Wenn $M_0 \xrightarrow{*} M$, dann $I \cdot M = I \cdot M_0$.

Beweis. Wir haben $M_0 \xrightarrow{\sigma} M$ für eine bestimmte Schaltsequenz σ. Aus dem Markierungsgleichungslemma folgt

$$M = M_0 + \mathbf{N} \cdot \vec{\sigma}$$

Damit

$$\begin{array}{lll} I \cdot M & = I \cdot M_0 + I \cdot \mathbf{N} \cdot \vec{\sigma} & \text{(Markierungsgleichung)} \\ & = I \cdot M_0 & (I \cdot \mathbf{N} = 0) \end{array}$$

□

$I \cdot M$ bleibt also invariant gegenüber Transitionschaltungen.
Beispiel. Wir berechnen die S-invarianten des Netzes aus Abbildung 4.1
Die Inzidenzmatrix ist:

	t_1	t_2	t_3
s_1	1	−1	0
s_2	0	−1	1
s_3	−1	1	0
s_4	0	1	−1

Wir müssen die Lösungen des folgenden Gleichungssystems bestimmen

$$(i_1, i_2, i_3, i_4) \cdot \begin{pmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \\ -1 & 1 & 0 \\ 0 & 1 & -1 \end{pmatrix} = 0$$

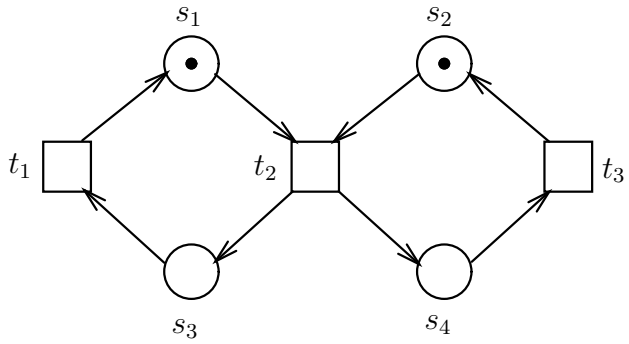


Abbildung 4.1:

Die allgemeine Form der S-Invarianten ist also (x, y, x, y) mit $x, y \in \mathbb{Q}$
Die folgende Proposition folgt unmittelbar aus der Definition von S-Invariante:

Proposition 4.3.3 *Die S-Invarianten eines Netzes bilden einen Vektorraum (mit dem Körper der natürlichen Zahlen, der üblichen komponentenweisen Addition von Vektoren und der Multiplikation mit rationalen Zahlen).*

Die Definition von S-Invariante ist sehr geeignet für maschinelle Bearbeitung aber nicht für Menschen, denn Menschen können nur sehr kleine Gleichungssysteme ohne Hilfe (und fehlerfrei!) lösen. Es gibt eine äquivalente Definition, mit der Menschen entscheiden können, sogar für Netze mit einigen Dutzend Knoten, ob ein gegebener Vektor eine S-Invariante ist.

Proposition 4.3.4 *I ist eine S-Invariante von $N = (S, T, F)$ gdw. $\forall t \in T : \sum_{s \in {}^\bullet t} I(s) = \sum_{s \in t^\bullet} I(s)$.*

Beweis. $I \cdot \mathbf{N} = 0$ ist äquivalent zu $I \cdot \mathbf{t} = 0$ für jede Transition t. Für jede Transition t gilt: $I \cdot \mathbf{t} = \sum_{s \in t^\bullet} I(s) - \sum_{s \in {}^\bullet t} I(s)$. □

Beispiel. Wir zeigen, daß $I = (1, 1, 2, 1)$ eine S-Invariante des Netzes aus Abbildung 4.2 ist. Die Bedingung aus Proposition 4.3.4 muß für die Transitionen t_1, t_2 und t_3 gelten.

- Transition t_1: $I(s_1) + I(s_2) = I(s_3) = 2$.
- Transition t_2: $I(s_3) = I(s_1) + I(s_4) = 2$.
- Transition t_3: $I(s_3) = I(s_4) + I(s_2) = 2$.

Mit Hilfe von S-Invarianten kann man eine hinreichende Bedingungen für die Beschränktheit eines Netzsystems schaffen, sowie notwendige Bedingungen für die Lebendigkeit eines Systems und für die Erreichbarkeit einer Markierung:

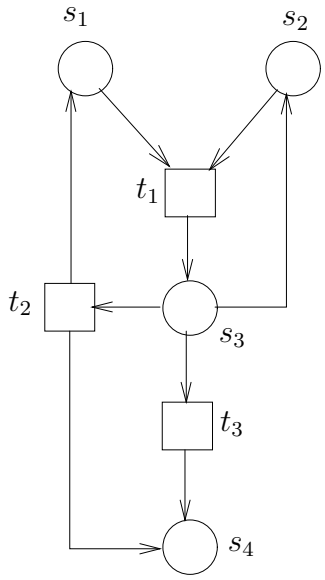


Abbildung 4.2:

Definition 4.3.5 (Semi-positive und positive S-Invarianten)
Sei I eine S-Invariante von $N = (S, T, F)$. I ist semi-positiv, wenn $I \geq 0$ und $I \neq 0$ und positiv, wenn $i > 0$ (d.h. $I(s) > 0$ für jedes $s \in S$). Die *Trägermenge* einer positiven S-Invariante ist ide Menge $\langle I \rangle = \{s \in S \mid I(s) > 0\}$.

Proposition 4.3.6 *[Eine hinreichende Bedingung für Beschränktheit]*
Sei (N, M_0) ein System. Wenn N eine positive S-Invariante I besitzt, dann ist (N, M_0) beschränkt. Genauer: (N, M_0) ist n-beschränkt mit

$$n = max \left\{ \frac{I \cdot M_0}{I(s)} \mid s \text{ ist eine Stelle von } N \right\}$$

Beweis. Sei M eine beliebige erreichbare Markierung. Aus der fundamentalen Eigenschaft der S-Invarianten folgt $I \cdot M = I \cdot M_0$.
Sei s eine beliebige Stelle von N. Da $I > 0$ ist $I(s) \cdot M(s) \leq I \cdot M = I \cdot M_0$ und $M(s) \leq \frac{I \cdot M_0}{I(s)}$. □

Proposition 4.3.7 *[Eine notwendige Bedingung für Lebendigkeit]*
Wenn (N, M_0) lebendig ist, dann gilt $I \cdot M_0 > 0$ für jede semi-positive S-Invariante von N.

Beweis. Sei I eine semi-positive S-Invariante und sei s eine Stelle von $\langle I \rangle$. Da (N, M_0) lebendig ist, existiert eine erreichbare Markierung M, die s markiert, d.h. $M(s) > 0$. Da I semi-positiv ist, gilt $I \cdot M \geq I(s) \cdot M(s) > 0$. Da I eine S-Invariante ist, gilt

$I \cdot M_0 = I \cdot M > 0$ □

Diese zwei Propositionen führen ummittelbar zu zwei effizienten Halbentscheidungsalgorithmen für **Beschränkheit** bzw. für **Lebendigkeit**.

Definition 4.3.8 (Die Relation $\sim$)
Seien M und L Markierungen und sei I eine S-Invariante eines Netzes N. M und L *stimmen überein bezüglich* I, wenn $I \cdot M = I \cdot L$. Wir schreiben $M \sim L$, wenn M und L bezüglich aller S-Invarianten von N übereinstimmen.

Proposition 4.3.9 *[Eine notwendige Bedingung für Erreichbarkeit]*
Sei (N, M_0) ein System. Für $M \in [M_0\rangle$ gilt $M \sim M_0$.

Beweis. Folgt aus der fundamentalen Eigenschaft von S-Invarianten. □

Der folgende Satz erlaubt, effizient zu entscheiden, ob $M \sim L$ für zwei gegebene Markierungen M und L gilt.

Satz 4.3.10 *Sei N ein Netz und seien M, L zwei Markierungen von N.*
$M \sim L$ gdw. die Gleichung $M = L + \mathbf{N} \cdot X$ eine rationale Lösung hat.

Beweis. $(\Rightarrow)$: M und L stimmen auf den Elementen einer Basis $\{I_1, \ldots, I_k\}$ des Vektorraums aller S-Invarianten überein. Für jeden Vektor I_j dieser Basis gilt $I_j \cdot (L - M) = 0$. Wegen eines bekannten Satzes der linearen Algebra gilt: die Spalten von $\mathbf{N}$ enthalten eine Basis des Lösungsraums des Systems $I_j \cdot X = 0$, $(1 \leq j \leq k)$. Damit ist $(L - M)$ eine lineare Kombination in Q dieser Spalten, d.h. die Gleichung $\mathbf{N} \cdot X = (L - M)$ hat eine rationale Lösung.

$(\Leftarrow)$: Sei I eine S-Invariante von N. Mit $I \cdot \mathbf{N} = 0$ gilt $I \cdot L = I \cdot M + I \cdot \mathbf{N} \cdot X = I \cdot M$. □

Wir haben also die folgenden Implikationen:

$$
\begin{array}{c}
M \text{ ist erreichbar aus } L \\
\not\Uparrow \qquad \Downarrow \\
M = L + \mathbf{N} \cdot X \text{ hat eine Lösung } X \in \mathbb{N}^{|T|} \\
\not\Uparrow \qquad \Downarrow \\
M = L + \mathbf{N} \cdot X \text{ hat eine Lösung } X \in \mathbb{Q}^{|T|} \\
\Updownarrow \\
M \sim L
\end{array}
$$

4.4 T-Invarianten

Definition 4.4.1 (T-Invariante)
Sei $N = (S, T, F)$ ein Netz. Ein Vektor $J : T \to Q$ heißt T-Invariante, wenn $\mathbf{N} \cdot J = 0$.

Proposition 4.4.2 *J ist eine T-Invariante von $N = (S, T, F)$ gdw. $\forall s \in S : \sum_{t \in {}^\bullet s} J(t) = \sum_{t \in s^\bullet} J(t)$.*

Proposition 4.4.3 *[Fundamentale Eigenschaft von T-Invarianten]*
Sei N ein Netz und sei σ eine Sequenz von Transitionen von N, die von einer Markierung M aktiviert wird. Der Vektor $\vec{\sigma}$ ist eine T-Invariante von N gdw. $M \xrightarrow{\sigma} M$.

Beweis. $(\Rightarrow)$: Sei M' die Markierung mit $M \xrightarrow{\sigma} M'$. Aus der Markierungsgleichung folgt $M' = M + \mathbf{N} \cdot \vec{\sigma}$. Mit $\mathbf{N} \cdot \vec{\sigma} = 0$ gilt $M' = M$

$(\Leftarrow)$: Aus der Markierungsgleichung folgt $M = N + \mathbf{N} \cdot \vec{\sigma}$ und damit $\mathbf{N} \cdot \vec{\sigma} = 0$. □

Beispiel. Wir berechnen die T-Invarianten des Netzes aus Abbildung 4.1. Dafür müssen wir die Lösungen des folgenden Gleichungssystems bestimmen

$$\begin{pmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \\ -1 & 1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} j_1 \\ j_2 \\ j_3 \end{pmatrix} = 0$$

Die allgemeine Form der T-Invarianten ist (x, x, x), mit $x \in \mathbb{Q}$.
T-Invarianten führen zu einer notwendigen Bedingung für die Wohlgeformtheit eines Netzes:

Satz 4.4.4 *[Notwendige Bedingung für Wohlgeformheit]*
Jedes wohlgeformte Netz besitzt eine positive T-Invariante.

Beweis. Sei N ein wohlgeformtes Netz und sei M_0 eine lebendige und beschränkte Markierung von N. Aufgrund der Lebendigkeit existiert eine unendliche Folge $\sigma_1, \sigma_2, \sigma_3, \ldots$ endlicher Transitionen-Sequenzen, in denen jeweils *alle* Transitionen von N vorkommen und mit denen gilt

$$M_0 \xrightarrow{\sigma_1} M_1 \xrightarrow{\sigma_2} M_2 \xrightarrow{\sigma_3} \ldots$$

Wegen der Beschränktheit können die Markierungen $M_0, M_1, M_2, \ldots$ nicht alle unterschiedlich sein und wir finden Indizes i und j mit $i < j$ und $M_i = M_j$. Die Teilfolge $\sigma_{i+1} \ldots \sigma_j$ erfüllt damit

$$M_i \xrightarrow{\sigma_{i+1} \ldots \sigma_j} M_i$$

und $J = \vec{\sigma}_{i+1} + \ldots + \vec{\sigma}_j$ ist eine T-Invariante. J ist positiv, weil jede Transition mindestens einmal in der Sequenz $\sigma_{i+1} \ldots \sigma_j$ vorkommt. □

4.5 Siphons und Traps

4.5.1 Siphons

Definition 4.5.1 (Siphon)
Sei $N = (S, T, F)$ ein Netz. Eine Menge $R \subseteq S$ von Stellen heißt Siphon, wenn

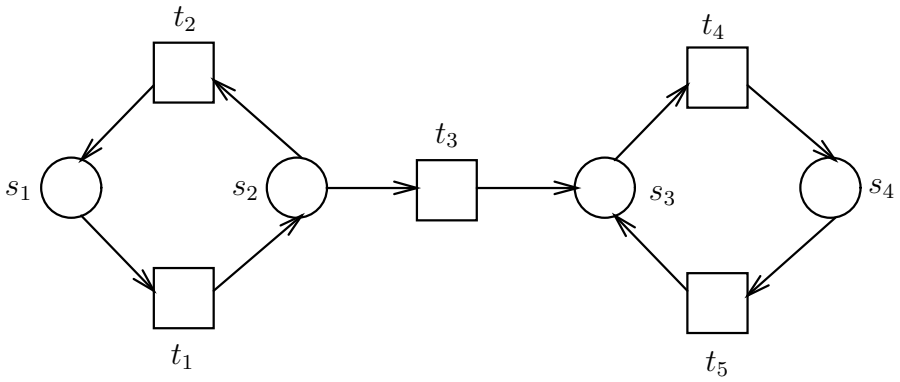


Abbildung 4.3:

${}^\bullet R \subseteq R^\bullet$. Ein Siphon R ist echt, wenn $R \neq \emptyset$.

$\{s_1, s_2\}$ ist ein Siphon vom Netz aus Abbildung 4.3, weil

$$ {}^\bullet\{s_1, s_2\} = {}^\bullet s_1 \cup {}^\bullet s_2 = \{t_2\} \cup \{t_1\} = \{t_1, t_2\} $$

und

$$ \{s_1, s_2\}^\bullet = s_1^\bullet \cup s_2^\bullet = \{t_1\} \cup \{t_2, t_3\} = \{t_1, t_2, t_3\} $$

Proposition 4.5.2 *[Fundamentale Eigenschaft von Siphons]*
Sei R ein Siphon eines Netzes N, und sei $M \xrightarrow{\sigma} M'$ eine Schaltsequenz von N. Wenn $M(R) = 0$, dann $M'(R) = 0$.

Beweis. ${}^\bullet R \subseteq R^\bullet$ impliziert: die Transitionen, die R markieren können, können nur schalten, wenn R schon markiert ist. □

D.h. wird ein Siphon von einer Markierung M nicht markiert, so wird er von keiner Folgemarkierung von M markiert. Oder, in knapper Form: nicht markierte Siphons bleiben nicht markiert.

Proposition 4.5.3 *[Eine notwendige Bedingung für Lebendigkeit]*
Ist ein System (N, M_0) lebendig, so markiert M_0 jeden echten Siphon von N.

Beweis. Sei R ein echter Siphon von N und sei $s \in R$. Wegen der Lebendigkeit gibt es eine erreichbare Markierung M, die s markiert. Damit markiert M auch den Siphon R. Aus der Proposition 4.5.3 folgt, daß M_0 den Siphon R auch markiert. □

Die Existenz eines unter M_0 nicht markierten Siphon kann in polynomieller Zeit mit Hilfe des folgenden Algorithmus getestet werden. Der Algorithmus berechnet den größten Siphon, der in einer gegebenen Menge R von Stellen enthalten ist. Es reicht also R zu wählen als die Menge der Stellen s mit $M_0(s) = 0$.

Input: Ein Netz $N = (S, T, F)$ und $R \subseteq S$.
Output: $Q \subseteq R$.
Initialisierung: $Q = R$.

begin
  while es gibt $s \in Q$ und $t \in {}^\bullet s$ mit $t \notin Q^\bullet$ **do**
    $Q := Q \setminus \{s\}$
  endwhile
end

Aufgaben:

(1) Zeigen Sie, daß der Algorithmus korrekt ist. D.h. beweisen Sie, daß der Algorithmus terminiert, und daß nach Terminierung Q der größte in R enthaltenen Siphon ist.

(2) Modifizieren Sie den Algorithmus, so daß er den größten in R enthaltenen Trap berechnet.

Proposition 4.5.4 *Ist ein System (N, M_0) verklemmt (d.h., M_0 aktiviert keine Transition von N), so ist die Menge der Stellen, die von M_0 nicht aktiviert werden, ein echter Siphon von N.*

Beweis. Sei $R = \{s \mid M_0(s) = 0\}$. Für jede Transition t gibt es eine Stelle $s \in {}^\bullet t$ mit $M_0(s) = 0$ (sonst wäre t aktiviert). Also, $R^\bullet$ enthält alle Transitionen von N. Es folgt ${}^\bullet R \subseteq R^\bullet$. □

Korollar 4.5.5 *[Eine hinreichende Bedingung für Verklemmungsfreiheit] Sei (N, M_0) ein System. Werden alle Siphons von N von allen erreichbaren Markierungen markiert, so ist (N, M_0) verklemmungsfrei.*

4.6 Traps

Definition 4.6.1 (Trap)
Sei $N = (S, T, F)$ ein Netz. Eine Menge $R \subseteq S$ von Stellen heißt Trap, wenn $R^\bullet \subseteq {}^\bullet R$. Ein Trap R ist echt, wenn $R \neq \emptyset$.

$\{s_3, s_4\}$ ist ein Trap vom Netz aus Abbildung 4.3.

Proposition 4.6.2 *[Fundamentale Eigenschaft von Traps]*
Sei R ein Trap eines Netzes N, und sei $M \xrightarrow{\sigma} M'$ eine Schaltsequenz von N. Wenn $M(R) > 0$, dann $M'(R) > 0$.

Beweis. ${}^\bullet R \subseteq {}^\bullet R$ impliziert: die Transitionen, die Marken von R nehmen, geben R wieder Marken. □

D.h. wird ein Trap von einer Markierung M markiert, so wird er von jeder Folgemarkierung von M markiert. Oder: markierte Traps bleiben markiert. Zu merken ist, daß die Gesamtzahl der Marken auf einem markierten Trap kleiner werden kann, nur nicht 0.

Proposition 4.6.3 *[Eine hinreichende Bedingung für Verklemmungsfreiheit]*
Sei (N, M_0) ein System. Enhält jeder echte Siphon von N einen von M_0 markierten Trap, so ist (N, M_0) verklemmungsfrei.

Beweis. Folgt aus Korollar 4.5.5 und Proposition 4.6.2. □

Diese Bedingung ist leider nicht leicht so verifizieren. Ob jeder echte Siphon von N einen von M_0 markierten Trap enthält, ist ein NP-vollständiges Problem.
Zuletzt zeigen wir wie S-Invarianten und Traps sich ergänzen, um zu beweisen, däs Petersons Algorithmus den gegenseitigen Auschluß gewährleistet. Im Netzmodell aus Abbildung 2.6 bedeutet das, daß keine erreichbare Markierung M die Bedingung $M(p_4) \geq 1 \wedge M(q_4) \geq 1$ erfüllt. Wir berechnen drei S-Invarianten

(1) $M(hold = 1) + M(hold = 2) = 1$

(2) $M(p_2) + M(p_3) + M(p_4) + M(m_1 = f) = 1$

(3) $M(q_2) + M(q_3) + M(q_4) + M(m_1 = f) = 1$

und zwei Traps

(4) $M(m_1 = f) + M(p_2) + M(hold = 1) + M(q_3) > 0$

(5) $M(m_2 = f) + M(q_2) + M(hold = 2) + M(p_3) > 0$

Aus der Annahme $M(p_4) \geq 1 \wedge M(q_4) \geq 1$ leiten wir nun einen Widerspruch ab:

$$M(p_4) \geq 1 \wedge M(q_4) \geq 1$$

$$\Rightarrow \{(2), (3)\}$$

$$M(p_2) + M(p_3) + M(m_1 = f) = 0 \quad \wedge \quad M(q_2) + M(q_3) + M(m_2 = f) = 0$$

$$\Rightarrow \{(1)\}$$

$$\begin{array}{lcl} M(m_1 = f) + M(p_2) + & & M(m_2 = f) + M(q_2) + \\ M(hold = 1) + M(q_3) = 0 & \vee & M(hold = 2) + M(p_3)0 \end{array}$$

Widerspruch zu (4) Widerspruch zu (5)

Kapitel 5

Systemklassen mit effizienten Verifikationsalgorithmen

In den drei Sektionen des Kapitels untersuchen wir drei Systemklassen: S-Systeme, T-Systeme und Free-Choice-Systeme. Alle Sektionen haben eine ähnliche Struktur. Nach der Definition der Klasse, werden drei Sätze vorgestellt: der Lebendigkeitssatz, der Beschränkheitssatz, und der Erreichbarkeitssatz. Der Lebendigkeissatz charakterisiert die Systeme der Klasse, die lebendig sind. Der Beschränkheitssatz charakterisiert die lebendigen Systemen, die dazu beschränkt bzw. b-beschränkt sind. Der Erreichbarkeitssatz charakterisiert die erreichbaren Markierungen der lebendigen und beschränkten Systeme. Der Beweis dieser drei Sätze benötigt oft einige Ergebnisse über die Struktur der S- und T-Invarianten der Systeme, die auch vorgestellt werden.
Aus diesen Sätzen folgen effiziente Algorithmen für **Lebendigkeit**, **Beschränktheit** und **Erreichbarkeit** (mindestens effizienter als die sehr aufwendigen allgemeinen Algorithmen).
Zuletzt wird in jeder Sektion ein Satz des kürzesten Pfades vorgestellt, der die Länge der kürzesten Schaltsequenz, die zu einer gegebenen Markierung führt angibt.
Man stellt sich die Frage, warum die Beschränktheit nur für lebendige Systeme und die Erreichbarkeit nur für lebendige und beschränkte Systeme charakterisiert wird. Der erste Grund ist, daß die beschränkten aber nicht lebendigen Systeme relativ uninteressant sind: ein Netzmodell eines korrekten Systems muß in der Regel lebendig *und* beschränkt sein. Aus demselben Grund untersuchen wir nur die Erreichbarkeit in diesen Systemen. Der zweite Grund ist, daß die allgemeine Charakterisierungen komplizierter sind.
Die S-Systeme dienen als Einstieg: manche Beweise werden weggelassen oder nur skizziert, weil sie sehr leicht sind. Die T-Systeme haben den “richtigen” Schwierigkeitsgrad für diese Vorlesung. Die Free-Choice Systeme enthalten sowohl die S-Systeme als auch die T-Systeme. Die Beweise werden oft weggelassen, weil sie zu lang bzw. zu schwer sind.

5.1 S-Systeme

Definition 5.1.1 (S-Netze, S-Systeme) Ein Netz heißt S-Netz, wenn $|{}^\bullet t| = 1 = |t^\bullet|$ für jede Transition t. Ein System (N, M_0) heißt S-System, wenn N ein S-Netz ist.

Proposition 5.1.2 (Fundamentale Eigenschaft von S-Systemen)
Sei (N, M_0) ein S-System, S die Menge der Stellen in N_0 und M eine erreichbare Markierung. Dann gilt $M_0(S) = M(S)$.

Beweis. Wenn eine Transition schaltet, dann konsumiert sie genau eine Marke, und kreiert auch eine marke. Die Gesamtzahl von Marken bleibt also invariant. □

Die Transitionen von S-Systemen "erzeugen" oder "vernichten" keine Marken. Wir können uns dann vorstellen, daß die Marken eine Identität haben und sich durchs Netz bewegen.

Satz 5.1.3 (Lebendigkeitssatz) *Ein S-System (N, M_0) ist lebendig, gdw. N stark zusammenhängend ist und M_0 mindestens eine Stelle markiert.*

Beweis. (Skizze.)
($\Rightarrow$): Wir beweisen die Kontraposition.
(1) Wenn N nicht stark zusammenhängend ist, dann gibt es eine Kante (s, t), so daß es keinen Pfad von t nach s gibt. Für alle Stellen s', für die es einen Pfad von s' nach s gibt, machen wir folgendes: durch geeignetes Schalten von Transitionen führen wir alle Marken auf diesen Stellen nach s, und dann schalten wir t bis es keine Marken mehr auf s gibt. Aus der Markierung, die wir so erreichen, ist es nicht mehr möglich t zu schalten.
(2) Wenn M_0 keine Stelle markiert, dann aktiviert M_0 keine Transition, weil jede Transition eines S-Systems eine Stelle im Vorbereich hat.
($\Leftarrow$): Wenn N stark zusammenhängend ist, dann kann jede Marke (und es gibt mindestens eine) sich frei im Netz bewegen und jede Stelle aus jeder anderen erreichen. Es ist also immer möglich, eine beliebige Transition erneut zu aktivieren. □

Satz 5.1.4 (Beschränktheitssatz) *Ein lebendiges System (N, M_0) ist b-beschränkt, gdw. $M_0(S) \leq b$.*

Beweis. Trivial. □

Aufgabe: Gegenbeispiel angeben für nicht lebendige S-Systeme.

Satz 5.1.5 (Erreichbarkeitssatz) *Sei (N, M_0) ein lebendiges S-System und sei M eine Markierung von M. M ist erreichbar, gdw. $M_0(S) = M(S)$.*

Beweis. Aus Proposition 5.1.3 folgt, daß N stark zusammenhängend ist. Die Marken können sich dann also frei im Netz bewegen, und von jeder Stelle jede andere erreichen. Ob eine Markierung erreichar ist oder nicht kommt dann darauf an, wie viele Marken sie hat und nicht auf ihre Positionierung. □

Proposition 5.1.6 (S-Invarianten von S-Netzen) *Sei $N = (S, T, F)$ ein zusammenhangendes S-Netz. Ein Vektor $I : S \to Q$ ist eine S-Invariante von N, gdw. $I = (x, \ldots, x)$ für eine Zahl $x \in Q$.*

Beweis.
($\Rightarrow$) Folgt aus Proposition 4.3.4 (alternative Definition von S-Invarianten).

($\Leftarrow$) Jede Transition t von N hat genau eine Stelle s_t im Vorbereich und eine Stelle s'_t im Nachbereich. Es folgt

$$\sum_{s \in {}^\bullet t} I(s) = I(s_t) \quad \text{und} \quad \sum_{s \in t^\bullet} I(s) = I(s'_t)$$

Damit

$$\begin{array}{ll} & I \text{ ist eine S-Invariante} \\ \Leftrightarrow & \{\text{Proposition 4.3.4}\} \\ & \forall t \in T : I(s_t) = I(s'_t) \\ \Leftrightarrow & \{N \text{ ist zusammenhängend}\} \\ & \forall s_1, s_2 \in S : I(s_1) = I(s_2) \\ \Leftrightarrow & \{\text{Logik}\} \\ & \exists x \in \mathbb{Q} \forall s \in S : I(s) = x. \end{array}$$

□

5.2 T-Systeme

Definition 5.2.1 (T-Netze, T-Systeme) Ein Netz heißt T-Netz, wenn $|{}^\bullet s| = 1 = |s^\bullet|$ für jede Stelle s. Ein System (N, M_0) heißt T-System, wenn N ein T-Netz ist.

Notation: Sei γ ein Kreis eines Netzes und sei M eine Markierung. Sei R die Menge der Stellen von γ. $M(\gamma)$ bezeichnet die Anzahl der Marken von γ unter M, d.h., $M(\gamma) = \sum_{s \in R} M(s)$.

Proposition 5.2.2 (Fundamentale Eigenschaft von T-Systemen) *Sei γ ein Kreis eines T-Systems (N, M_0) und sei M eine erreichbare Markierung. Dann gilt $M(\gamma) = M_0(\gamma)$.*

Beweis. Das Schalten einer Transition t berührt die Marken eines Kreises gar nicht oder nimmt eine Marke von der einzigen Stelle des Kreises in ${}^\bullet t$, und legt eine Marke auf die einzige Stelle des Kreises in $t^\bullet$. □

5.2.1 Lebendigkeit

Satz 5.2.3 (Lebendigkeitssatz) *Ein T-System (N, M_0) ist lebendig, gdw. $M_0(\gamma) > 0$ für jeden Kreis γ von N.*

Beweis.
($\Rightarrow$) Sei γ ein Kreis mit $M_0(\gamma) = 0$. Aus Proposition 5.2.2 folgt $M(\gamma) = 0$ für jede erreichbare Markierung M. Damit kann keine Transition von γ irgendwann einmal schalten.
($\Leftarrow$) Sei t eine beliebige Transition und M eine beliebige erreichbare Markierung. Wir zeigen, daß es eine aus M erreichbare Markierung gibt, die t aktiviert. Sei S_M die Untermenge der Stellen s von N mit der folgenden Eigenschaft: es gibt einen Pfad von s nach t, der keine unter M markierten Stellen enthält.
Wir führen Induktion über $|S_M|$.
Basis: $|S_M| = 0$. Mit diesem Fall gilt $M(s) > 0$ für jede Stelle $s \in {}^\bullet t$. Also M aktiviert t.
Schritt: $|S_M| > 0$. Mit der fundamentalen Eigenschaft von T-Systemen ist jeder Kreis von N unter M markiert. Also gibt es einen Pfad Π mit:

(1) Π führt nach t;

(2) M markiert keine Stelle von Π;

(3) Π hat maximale Länge (d.h., kein Kreis länger als Π erfüllt (1) und (2)).

Sei u das erste Element von Π. Mit (3) ist u eine Transition und M markiert alle Stellen in ${}^\bullet u$. Also u wird von M aktiviert. Wir haben $u \neq t$, weil t von M nicht aktiviert wird. Definiere $M \xrightarrow{u} M'$. Wir zeigen $S_{M'} \subset S_M$ und damit $|S_{M'}| < |S_M|$.

1. $S_{M'} \subseteq S_M$
   Sei $s \in S_{M'}$. Wir zeigen $s \in S_M$. Es gibt einen Pfad $\Pi' = s \ldots t$, der keine von M' markierten Stellen enthält. Nehmen wir an, daß Π' eine von M markierte Stelle r enthält. Aus $M'(r) = 0$ und $M \xrightarrow{u} M'$ folgt $u \in r^\bullet$ und damit $\{u\} = r^\bullet$. Also u ist der Nachfolger von r in Π'. Mit $u \neq t$ gilt, daß der Nachfolger von u in Π' von M' markiert wird. Das widerspricht der Definition von Π'.

2. $S_{M'} \neq S_M$. Sei s der Nachfolger von u in Π. Wir haben $s \in S_M$ aber $s \notin S_{M'}$, weil $M'(s) > 0$.

Die Anwendung der Induktionshypothese liefert eine Schaltsequenz $M \xrightarrow{\sigma} M''$ und M' aktiviert t. Damit $M \xrightarrow{u} M' \xrightarrow{\sigma} M''$. Also M'' ist eine aus M erreichbare Markierung, die t aktiviert. □

5.2.2 Beschränkheit

Satz 5.2.4 (Beschränktheitssatz) *Ein lebendiges T-System (N, M_0) ist b-beschränkt gdw. jede Stelle s in einem Kreis γ mit $M_0(\gamma) \leq b$ enthalten ist.*

Beweis. ($\Leftarrow$) Folgt aus der fundamentalen Eigenschaft von T-Systemen (Proposition 5.2.2).
($\Rightarrow$) Sei s eine Stelle und sei M eine erreichbare Markierung für die $M(s)$ maximal ist. Es gilt $M(s) \leq b$. Definiere die Markierung L

$$L(r) = \begin{cases} M(r) & \text{falls } r \neq s \\ 0 & \text{falls } r = s \end{cases}$$

Das System (N, L) ist nicht lebendig. Sonst gäbe es eine Schaltsequenz $L \xrightarrow{\sigma} L'$ mit $L'(s) > 0$ und wegen der Monotonie gälte auch $M \xrightarrow{\sigma} M'$ für eine Markierung M' mit $M'(s) = L'(s) + M(s) > M(s)$, was der Maximalität von $M(s)$ widerspricht. Mit dem Lebendigkeitssatz gibt es einen unter L unmarkierten Kreis γ, der unter M doch markiert ist. L und M unterscheiden sich nur in s und daher enthält γ die Stelle s. Außerdem ist s die einzige unter L markierte Stelle von γ. Also $M(\gamma) = M(s)$ und mit $M(s) \leq b$ gilt $M(\gamma) \leq b$. □

Korollar 5.2.5 *Sei (N, M_0) ein lebendiges T-System*

1. *Eine Stelle ist beschränkt gdw. sie zu einem Kreis gehört.*
2. *Sei s eine beschränkte Stelle*

$$max\{M(s) \mid M_0 \xrightarrow{*} M\} = min\{M_0(\gamma) \mid \gamma \text{ enthält } s\}$$

3. *(N, M_0) ist beschränkt gdw. N stark zusammenhängend ist.*

Beweis. Aufgabe □

5.2.3 Erreichbarkeit

Für den Erreichbarkeitssatz müssen wir erst die T-Invarianten von T-Netzen näher betrachten:

Proposition 5.2.6 (T-Invarianten von T-Netzen) *Sei $N = (S, T, F)$ ein zusammenhängendes T-Netz. Ein Vektor $J : T \to Q$ ist eine T-Invariante gdw. $J = (x \dots x)$ für eine Zahl $x \in Q$.*

Beweis. Dual vom Beweis von Proposition 5.1.6. □

Satz 5.2.7 (Erreichbarkeitssatz) *Sei (N, M_0) ein lebendiges T-System. Eine Markierung M ist erreichbar gdw. $M_0 \sim M$.*

Beweis. ($\Rightarrow$) Proposition 4.4.2
($\Leftarrow$) Mit Satz 4.3.10 gibt es einen rationalen Vektor X mit

$$M = M_0 + \mathbf{N}.X \tag{5.1}$$

Der Vektor $J = (1, 1, \ldots, 1)$ ist eine T-Invariante von N (Proposition 5.2.6). Damit gilt

$$\mathbf{N} \cdot (X + \lambda J) = \mathbf{N} \cdot X$$

für jede Zahl λ. Daraus folgt, daß man oBdA. $X \geq 0$ annehmen darf.
Sei T die Menge der Transitionen von N. Wir zeigen:

(1) Es gibt einen Vektor $Y : T \to \mathbb{N}$ mit $M = M_0 + \mathbf{N} \cdot Y$. Sei Y der Vektor mit $Y(t) = \lceil X(t) \rceil$ für jede Transition t ($\lceil x \rceil$ bezeichnet die größte natürliche Zahl, die kleiner oder gleich x ist). Aus (5.1) folgt

$$M(s) = M_0(s) + X(t_1) - X(t_2)$$

für jede Stelle s, wobei $\{t_1\} = {}^\bullet s$ und $\{t_2\} = s^\bullet$. Sowohl $M(s)$ als auch $M_0(s)$ sind ganzzahlig. Aus der Definition von Y folgt

$$X(t_1) - X(t_2) = Y(t_1) - Y(t_2)$$

Also $M(s) = M_0 + Y(t_1) - Y(t_2)$ und damit $M = M_0 + \mathbf{N} \cdot Y$

(2) $M_0 \xrightarrow{*} M$
Wir führen Induktion über $|Y| = \sum_{t \in T} Y(t)$.
Basis: $|Y| = 0$. Dann $Y = 0$ und $M = M_0$.
Schritt: $|Y| > 0$.
Wir zeigen, daß M_0 mindestens eine Transition aus $\langle Y \rangle$ aktiviert. Sei

$$Sy = \{s \in \cdot\langle Y \rangle \mid M_0(s) = 0\}$$

Sei $s \in Sy$. Aus $M_0(s)$ und $M_0 + \mathbf{N} \cdot Y = M \geq 0$ folgt:

> liegt eine Transition aus $s^\bullet$ in $\langle Y \rangle$, so liegt eine Transition aus ${}^\bullet s$ in $\langle Y \rangle$. (*)

Es existiert einen Pfad mit maximaler Länge, der nur Stellen aus *Sy* und Transitionen aus $\langle Y \rangle$ enthält (sonst gäbe es einen unter M_0 unmarkierten Kreis). Der Pfad beginnt mit einer Transition $t \in \langle Y \rangle$ wegen (*), und keine Stelle von ${}^\bullet t$ gehört zu *Sy*. Also, jede Stelle in ${}^\bullet t$ wird von M_0 markiert, d.h. t wird von M_0 aktiviert.
Sei M_1 die Markierung mit $M_0 \xrightarrow{t} M_1$. Dann gilt

$$M_1 + \mathbf{N}(Y - \vec{t}) = M$$

mit

$$|Y - \vec{t}| = |Y| - 1 < |Y|$$

Aus der Induktionsvoraussetzung folgt $M_1 \xrightarrow{*} M$. Mit $M_0 \xrightarrow{t} M_1 \xrightarrow{*} M$ gilt $M_0 \xrightarrow{*} M$.

□

5.2.4 Weitere Eigenschaften

Die Sätze, die wir betrachten haben, führen zu vielen anderen interessanten Eigeschaften. Hier sind zwei davon:

Satz 5.2.8 *Sei (N, M_0) ein stark zusammenhängendes T-Netz. Die folgenden Aussagen sind äquivalent:*

(1) (N, M_0) *ist lebendig.*

(2) (N, M_0) *ist verklemmungsfrei.*

(3) (N, M_0) *hat eine unendliche Schaltsequenz.*

Beweis. (1) $\Rightarrow$ (2) $\Rightarrow$ (3) folgt unmittelbar aus den Definitionen. Wir zeigen (3) $\Rightarrow$ (1). Sei $M_0 \xrightarrow{\sigma}$ eine unendliche Schaltsequenz. Wir zeigen zuerst, daß alle Transitionen von N in σ vorkommen. Da N stark zusammenhängend ist, ist (N, M_0) beschränkt (Satz 5.2.4). Sei $\sigma = t_1\ t_2\ t_3 \ldots$, und sei $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} \ldots$. Da (N, M_0) beschränkt ist, gibt es Indices i und j, $i < j$, mit $M_i = M_j$. Sei σ_{ij} die Teilsequenz von σ, die die Transitionen zwischen M_i und M_j enthält. Aus der fundamentalen Eigenschaft der T-invarianten (Proposition 4.4.3) folgt, daß $\vec{\sigma_{ij}}$ eine T-Invariante ist, und aus Proposition 5.2.6, daß es $n \in \mathbb{N}$ gibt mit $\vec{\sigma_{ij}} = (n \ldots n)$. Also, jede Transition von N kommt in σ_{ij} (und damit auch in σ) vor.
Weil alle Transitionen von N in σ vorkommen, wird jede Stelle von N (und damit jeder Kreis) während der Ausführung von σ markiert. Aus der fundamentalen Eigenschaft der T-Systeme folgt, daß alle Kreise von N anfänglich markiert sind. Mit dem Lebendigkeitssatz (Satz 5.2.3) ist (N, M_0) lebendig. □

Satz 5.2.9 (Genrich'sche Satz) *Sei N ein stark zusammenhängendes T-Netz mit mindestens einer Stelle und einer Transition. Es gibt eine Markierung M_0 von N, so daß (N, M_0) lebendig und 1-beschränkt ist.*

Beweis. Die Markierung von N, in der eine Marke auf jede Stelle liegt, ist lebendig, weil sie alle Kreise markiert. Es gibt also lebendige Markierungen von N.
Sei (N, M) lebendig aber nicht 1-beschränkt. Wir konstruieren eine neue lebendige Markierung L von N, die die folgenden zwei Bedingungen erfüllt:

(1) $L(\gamma) \leq M(\gamma)$ für jeden Kreis γ, und

(2) $L(\gamma) < M(\gamma)$ für mindestens einen Kreis γ.

Die Iteration dieser Konstruktion liefert am Ende eine 1-beschränkte Markierung von N.
Sei s eine nicht-1-beschränkte Stelle von (N, M). Es gibt eine erreichbare Markierung M' mit $M'(s) \geq 2$. Sei L die Markierung, die mit L auf allen Stellen außer s übereinstimmt, und genau eine Marke auf s legt.
Da M lebendig ist, markiert sie alle Kreise von N. Aus der Konstruktion von L folgt, daß L auch alle Kreise markiert. Also, L ist lebendig. (1) folgt aus der Konstruktion von L. (2) gilt für alle Kreise, die s enthalten (es gibt mindestens einen, weil N start zusammenhängend ist). □

5.3 Free-Choice-Systeme

Definition 5.3.1 (Free-Choice-Netze, Free-Choice-Systeme) Ein Netz $N = (S, T, F)$ heißt *free-choice*, wenn $s^\bullet \times {}^\bullet t \subseteq F$ für jedes $s \in S$ und $t \in T$ mit $(s, t) \in F$. Ein System (N, M_0) heißt *free-choice*, wenn N ein Free-Choice-Netz ist.

Diese Definition ist sehr knapp und außerdem symmetrisch bezüglich Stellen und Transitionen. Wenn Sie sie etwas kryptisch finden, dann können Ihnen die folgenden äquivalenten Definitionen helfen:

Proposition 5.3.2 (Alternative Definitionen von Free-Choice-Netzen)

(1) Ein Netz ist free-choice, wenn für alle Transitionen t_1, t_2 gilt:

$$(t_1 \neq t_2 \wedge {}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset) \Rightarrow {}^\bullet t_1 = {}^\bullet t_2$$

(2) Ein Netz ist free-choice, wenn für alle Stellen s_1, s_2 gilt:

$$(s_1 \neq s_2 \wedge s_1^\bullet \cap s_2^\bullet \neq \emptyset) \Rightarrow s_1^\bullet = s_2^\bullet$$

Beweis. Aufgabe. □

Abbildung 5.1 erläutert diese Definitionen.
Es folgt aus diesen Definitionen, daß alle S- und T-Systeme Free-Choice sind aber nicht umgekehrt (siehe Abbildung 5.2).

5.3.1 Lebendigkeit

Wir haben im letzten Kapitel gezeigt, daß wenn jeder Siphon eines Systems einen unter der Anfangsmarkierung markierten Trap enthält, das System verklemmungsfrei ist (aber nicht umgekehrt). Wenn wir uns auf Free-Choice-Systeme einschränken, dann erhalten wir eine viel stärkere Aussage, den Satz von Commoner, der die Lebendigkeit von Free-Choice-Netzen charakterisiert:

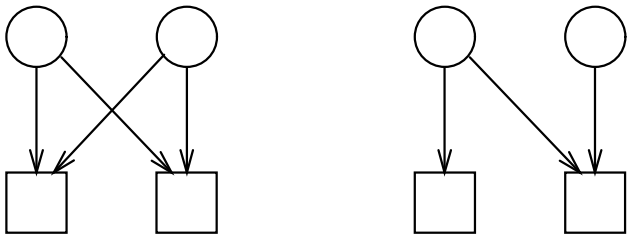


Abbildung 5.1:

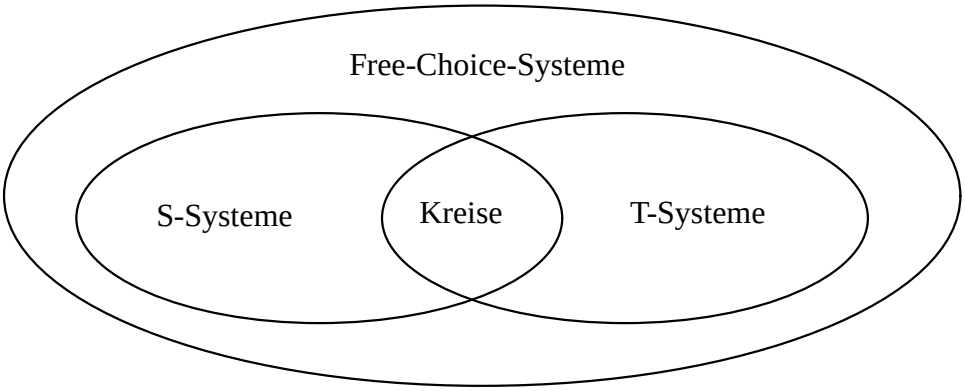


Abbildung 5.2:

Satz 5.3.3 (Commoner'scher Lebendigkeitssatz) *Ein Free-Choice-System (N, M_0) ist lebendig gdw. jeder Siphon von N einen unter M_0 markierten Trap enthält.*

Beweis. Wir skizzieren eine Richtung: wenn jeder Siphon von N einen unter M_0 markierten Trap enthält, dann ist (N, M_0) lebendig. Wir benutzen die folgenden Definitionen. Sei M eine Markierung von N. Eine Transition t ist *tot unter* M, wenn sie keine Markierung aus $[M\rangle$ aktiviert. Die Menge der unter M toten Transitionen wird mit D_M bezeichnet. Eine Transition t ist *lebendig unter* M, wenn $t \notin D_{M'}$ für alle Markierungen $M' \in [M\rangle$. Die Menge der unter M lebendigen Transitionen wird mit L_M bezeichnet. Zu merken ist, daß eine Transition weder lebendig noch tot unter einer Markierung sein kann. Außerdem:

- wenn $t \in L_M$ und $M' \in [M\rangle$, dann $t \in L_{M'}$, d.h., lebendige Transitionen bleiben lebendig.
- wenn $t \in D_M$ und $M' \in [M\rangle$, dann $t \in L_{M'}$, d.h., tote Transitionen bleiben tot.
- wenn $t \notin L_M \cup D_M$, dann $t \notin L_M$ für alle Markierungen M' erreichbar von M, und es gibt eine Markierung M' erreichbar von M so daß $t \in D_{M'}$. D.h. Transitionen, die weder lebendig not tot sind, bleiben nicht lebendig und können (müssen aber nicht!) tot werden.

Sei T die Menge der Transitionen von N. Aus den obigen Definitionen folgt (Aufgabe: zeigen Sie es!): es gibt eine aus M_0 erreichbaren Markierung M, für die $T = D_M \cup L_M$ gilt, d.h. jede Transition ist entweder tot oder lebendig unter M. Für jede $t \in D_M$ gibt es eine Stelle $s_t \in {}^\bullet t$ mit $M(s_t) = 0$. Weil N free-choice ist, gilt (Aufgabe: zeigen Sie es!): keine Transition im Vorbereich von s_t ist unter M lebendig, d.h. ${}^\bullet s_t \subseteq D_M$. Es folgt: die Menge $R = \{s_t \mid t \in T\}$ ist ein Siphon, der von M nicht markiert wird. R kann dann keinen unter M_0 markierten Trap enthalten, weil markierte Traps markiert bleiben.

□

Der Lebendigkeitssatz gilt offensichtlich noch wenn man 'Siphon' durch 'minimaler Siphon' ersetzt: ein Siphon heißt *minimal*, wenn er echt ist und keinen kleineren echten Siphon enthält. Das Netz aus Abbildung 5.3 enthält zwei minimalen Siphons: $R_1 = \{s_1, s_3, s_5, s_7\}$ und $R_2 = \{s_2, s_4, s_6, s_8\}$. R_1 und R_2 sind auch Traps; insbesondere, sie enthalten Traps. Es folgt aus Satz 5.3.3, daß jede Markierung, die R_1 und R_2 markiert, lebendig ist.
Der Satz von Commoner führt leider zu keinem effizienten Algorithmus für **Lebendigkeit** in Free-Choice-Systemen. Das ist nicht überraschend: das Problem ist nämlich NP-vollständig.

Satz 5.3.4 (Komplexität) *Das Problem*
Gegeben: ein free-choice System (N, M_0)
Frage: Ist (N, M_0) nicht lebendig?
Ist NP-vollständig.

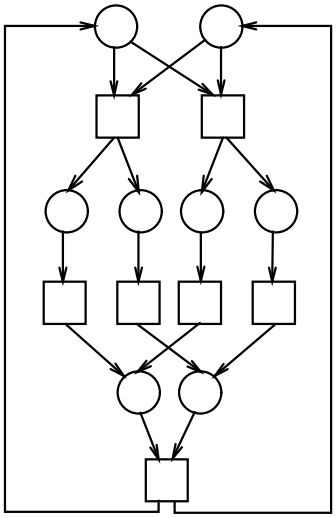


Abbildung 5.3: Ein Free-Choice-System

Beweis. Der Beweis erfolgt durch Reduktion vom Erfüllbarkeitsproblem der Aussagenlogik (SAT). Die Reduktion wird in der Abbildung 5.4 and dem folgenden Beispiel erläutert:

$$\Phi = (x_1 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee \overline{x_3})$$

□

5.3.2 Beschränkheit

Definition 5.3.5 (S-Komponente) Sei $N = (S, T, F)$ ein Netz und sei $N' = (S', T', F')$ ein Teilnetz von N. N' heißt S-Komponente von N, wenn es die folgenden Bedingungen erfüllt:

1. $T' = {}^\bullet S' \cup S'^\bullet$ (wobei $s^\bullet = \{t \in T \mid (t, s) \in F\}$ und analog für $s^\bullet$)
2. N' ist stark zusammenhängend

Abbildung 5.5 zeigt zwei S-Komponenten vom Netz aus Abbildung 5.3.
S-Komponenten sind für Free-Choice-System das, was Kreise für T-Systeme sind, wobei die Gesamtzahl der Marken auf einer S-Komponente invariant bleibt.

Proposition 5.3.6 *Sei (N, M_0) ein System und sei $N' = (S', T', F')$ eine S-Komponente von N. Für jede erreichbare Markierung M gilt $M_0(S') = M(S')$.*

Beweis. Wir bezeichnen mit M' die Projektion einer Markierung M von N auf die Menge S'. Die folgende Aussage ist leicht zu zeigen: wenn M erreichbar in (N, M_0)

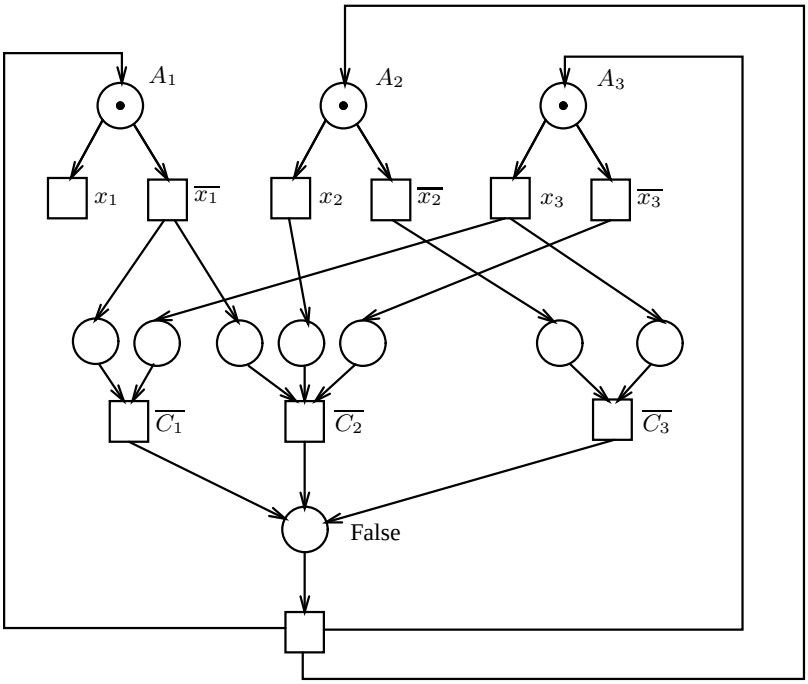


Abbildung 5.4: Das Free-Choice-System für die Formel Φ

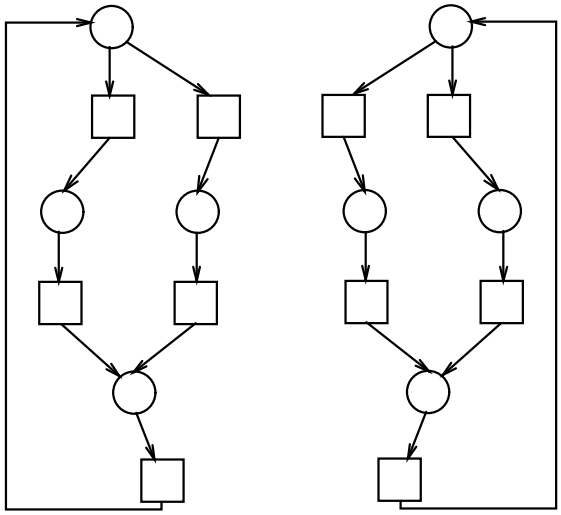


Abbildung 5.5: S-Komponenten vom Netz aus Abbildung 5.3

ist, dann ist M' in (N', M'_0) auch erreichbar. Der Satz folgt nun aus der fundamentalen Eigenschaft der S-Systeme. □

Satz 5.3.7 (Hack'scher Beschränktheitssatz) *Sei (N, M_0) ein lebendiges free-choice System. (N, M_0) ist beschränkt gdw. jede Stelle von N in einer S-Komponente enthalten ist.*

Beweis. ($\Leftarrow$) Aufgabe

($\Rightarrow$) (Skizze). Man zeigt zuerst, daß jeder minimale Graph ein Siphon der Menge der Stellen einer S-Komponente ist. Dann zeigt man, daß jede Stelle in einem minimalen Siphon enthalten ist. □

Proposition 5.3.8 (Schranken von Stellen) *Sei (N, M_0) ein lebendiges und beschränktes free-choice System und sei s eine Stelle von N. Es gilt*

$$max\{M(s) \mid M_0 \xrightarrow{*} M\} =$$
$$min\{M_0(S') \mid S' \textit{ ist die Menge der Stellen einer S-Komponente von } N\}$$

Beweis. Analog zum Beschränktheitssatz für T-Systeme. □

Satz 5.3.4 besagt, daß es keinen polynomiellen Algorithmus gibt, der **Lebendigkeit** für Free-Choice-Systeme löst (wenn $P \neq NP$). Wir fragen uns nun, wie schwer ist es zu entscheiden, ob ein Free-Choice-System lebendig *und* beschränkt ist. Natürlich kann man das lösen, indem man erst entscheidet, ob das System lebendig ist und, wenn ja, ob es beschränkt ist. Es gibt aber effizientere Methoden.[1]. Der schnellste Algorithmus braucht $O(n \cdot m)$ Zeit für ein Netz mit n Stellen und m Transitionen. Ein nicht so effizienter aber einfacherer Algorithmus folgt aus dem nächsten Satz:

Definition 5.3.9 (Cluster) Sei $N = (S, T, F)$ ein Netz. Ein Cluster ist eine Äquivalenzklasse der Relation $((F \cap (S \times T)) \cup (F \cap (S \times T))^{-1})^*$.

Abbildung 5.6 zeigt die Cluster des Netzes aus Abbildung 5.3.

Satz 5.3.10 (Rangssatz) *Ein Free-Choice-System (N, M_0) ist lebendig und beschränkt gdw. die folgenden Bedingungen gelten:*

1. *N hat eine positive S-Invariante.*
2. *N hat eine positive T-Invariante.*

[1] Vergleiche mit: um zu entscheiden, ob eine Zahl durch 100.000 teilbar ist, kann man erst überprüfen, ob sie durch 3125 teilbar ist, und, wenn ja, durch 32. Es gibt aber eine effizientere Methode: sind die letzten fünf Stellen Nullen?

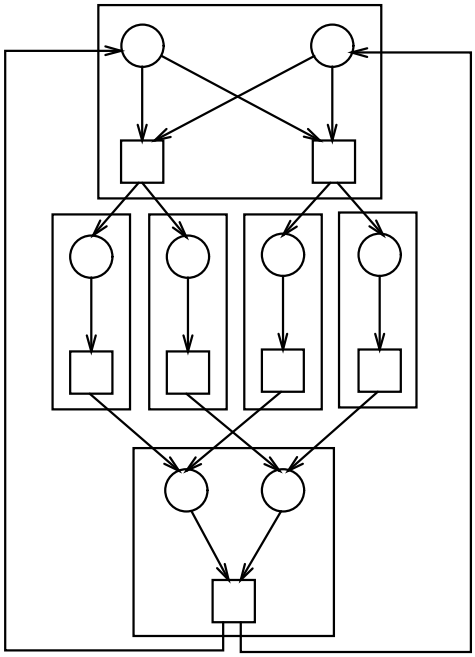


Abbildung 5.6: Clusters des Netzes aus Abbildung 5.3

3. *Rang* $(\mathbf{N}) = c - 1$, *wobei* c *die Anzahl der Cluster von* N *ist.*

4. *Jeder Siphon von* N *ist unter* M_0 *markiert.*

Beweis. Weggelassen (sehr schwer). □

Bedingungen (1) und (2) können mit linearer Programmierung verifiziert werden, Bedingung (3) mit bekannten Algorithmen der linearer Algebra und Bedingung (4) mit dem Algorithmus aus Abschnitt 4.5.1.

5.4 Erreichbarkeit

Wenn $P \neq NP$, dann kann es leider keinen polynomiellen Algorithmus für **Erreichbarkeit** in Free-Choice-Netzen geben, weil:

Satz 5.4.1 Erreichbarkeit *für lebendige und beschränkte free-choice Systeme ist NP-vollständig.*

Beweis. Eine nicht sehr schwierige, aber etwas aufwendige, Reduktion von SAT. □

Es gibt aber polynomielle Algorithmen, wenn man eine zusätzliche Einschränkung in Kauf nimmt. Ein System (N, M_0) heißt *zyklisch*, wenn

$$\forall M \in [M_0\rangle : M_0 \in [M\rangle$$

D.h. ein System ist zyklisch, wenn es immer möglich ist, zurück zur Anfangsmarkierung zu kommen. Wir haben:

Satz 5.4.2 (Erreichbarkeitssatz) *Sei (N, M_0) ein lebendiges, beschränktes und zyklisches Free-Choice-System. Eine Markierung M von N ist erreichbar aus M_0 gdw. $M_0 \sim M$.*

Beweis. Weggelassen (sehr schwer). □

Korollar 5.4.3 *Das Problem*
Gegeben: ein lebendiges, beschränktes und zyklisches Free-Choice-System (N, M_0) und eine Markierung M
Frage: Ist M erreichbar?
Ist in polynomieller Zeit lösbar.

Dieses Ergebnis ist nur nutzlich, wenn man effizient entscheiden kann, ob ein lebendiges und beschränktes Free-Choice-System zyklisch ist. Der folgende Satz sagt, daß das der Fall ist:

Satz 5.4.4 *Ein lebendiges und beschränktes Free-Choice-System (N, M_0) ist zyklisch gdw. jeder Trap von N unter M_0 markiert ist.*

Beweis. Weggelassen (ziemlich schwer). □