

7.2 Der Überdeckungsgraph

Der Algorithmus 7.1 zur Berechnung des Erreichbarkeitsgraphen bricht ohne Ergebnis ab, wenn das Netz unbeschränkt ist. Das ist unbefriedigend, wenn man z.B. zur Fehleranalyse wissen muss, welche Plätze unbeschränkt sind oder wie der Erreichbarkeitsgraph in Teilen aussieht, wo sich die Unbeschränktheit nicht auswirkt.

Mit dem Symbol ω (Omega) wird im Folgenden die Möglichkeit beliebig hoher (aber natürlich endlicher) Markenanzahlen auf einem Platz bezeichnet. Dazu werden die natürlichen Zahlen mit diesem formalen Symbol erweitert, sowie *Pseudomarkierungen* eingeführt, deren Komponenten dieses Omega enthalten.

Definition 7.14 Es sei $\mathbb{N}_\omega := \mathbb{N} \cup \{\omega\}$ zusammen mit folgenden Rechenregeln:

$$\forall n \in \mathbb{N} : \omega > n;$$

$$\forall n \in \mathbb{N}_\omega : \omega + n = \omega - n := \omega;$$

$$\forall n \in \mathbb{N} \setminus \{0\} : n \cdot \omega = \omega \cdot n := \omega; 0 \cdot \omega = \omega \cdot 0 := 0$$

Ein Vektor $\mathbf{m} \in \mathbb{N}_\omega^P$ wird *Pseudomarkierung* genannt, wenn in ihm das Symbol ω vorkommt, und für diese wird die Schaltregel formal übernommen. Eine Pseudomarkierung \mathbf{m} entspricht einer gewöhnlichen (Teil-) Markierung auf den Plätzen s mit $\mathbf{m}(s) \neq \omega$, wobei die mit ω besetzten Komponenten beliebig sind und unberücksichtigt bleiben.

Für Vektoren $\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{Z}^r$ seien die Operatoren $+, -, =$ jeweils komponentenweise erklärt.

Die partielle Ordnung \leq ergibt sich komponentenweise: $\mathbf{m}_1 \leq \mathbf{m}_2$, falls $\forall p \in P : \mathbf{m}_1(p) \leq \mathbf{m}_2(p)$.

Die strikte Ordnung $\mathbf{m}_1 < \mathbf{m}_2$ (besser: $\mathbf{m}_1 \lneq \mathbf{m}_2$) steht für $(\mathbf{m}_1 \leq \mathbf{m}_2 \text{ und } \mathbf{m}_1 \neq \mathbf{m}_2)$.

7.2.1 Konstruktion des Überdeckungsgraphen

Wir definieren im Folgenden den Überdeckungsgraphen. Ein Überdeckungsgraphen ist eine reduzierte Version des Erreichbarkeitsgraphen. Es wird sich zeigen, dass ein *Überdeckungsgraph* stets endlich ist, während dies bei Erreichbarkeitsgraphen bekanntermaßen nicht immer der Fall ist. Somit ist es unvermeidbar, dass ein Überdeckungsgraph weniger Information über die Zustände enthält als der Erreichbarkeitsgraph. Wie wir aber sehen werden, reicht diese eingeschränkte Version bereits aus, um einige (wenn auch nicht alle) Fragen zu Petrinetzen zu entscheiden – insbesondere die Fragestellung, ob ein Netz beschränkt ist.

Sei $\mathcal{N} = (P, T, F, W, \mathbf{m}_0)$ ein P/T-Netz. Wir konstruieren durch Algorithmus 7.4 einen gerichteten, kantenbeschrifteten Graphen $G(\mathcal{N}) := (V, E)$ mit $V \subseteq \mathbb{N}_\omega^P$ und Kanten $E \subseteq V \times T \times V$ und nennen ihn *Überdeckungsgraph*. Dabei schreiben wir $\mathbf{m}_1 \xrightarrow[w]{*} \mathbf{m}_2$, wenn es in $G(\mathcal{N})$ einen Pfad vom Knoten \mathbf{m}_1 zum Knoten \mathbf{m}_2 gibt, der die in der

Algorithmus 7.4 (Berechnung eines Überdeckungsgraphen)

Input - Das P/T-Netz $\mathcal{N} = (P, T, F, W, \mathbf{m}_0)$

Output - Der Überdeckungsgraph $G(\mathcal{N}) = (V, E)$.

1. Initialisiere $G(\mathcal{N}) = (\{\mathbf{m}_0\}, \emptyset)$; \mathbf{m}_0 sei ungefärbt;
2. **while** Es gibt ungefärbte Knoten in V **do**
 - 2.1 Wähle einen ungefärbte Knoten $\mathbf{m} \in V$ und färbe ihn.
 - 2.2 **for** Für jede in \mathbf{m} aktivierte Transition t **do**
 - 2.2.1 Berechne \mathbf{m}' mit $\mathbf{m} \xrightarrow{t} \mathbf{m}'$ und $X(\mathbf{m}') := \{\mathbf{m}''' \in V \mid \mathbf{m}''' \leq \mathbf{m}' \text{ und } \mathbf{m}''' \xrightarrow{*} \mathbf{m}'\}$;
 - 2.2.2 **if** $X(\mathbf{m}') \neq \emptyset$ **then** $\mathbf{m}_1(p) := \begin{cases} \omega, & \exists \mathbf{m}''' \in X(\mathbf{m}') : \mathbf{m}'''(p) < \mathbf{m}'(p) \\ \mathbf{m}'(p), & \text{sonst.} \end{cases}$
 - else** $\mathbf{m}_1 := \mathbf{m}'$;
 - 2.2.3 **if** $\mathbf{m}_1 \notin V$ **then** $V := V \cup \{\mathbf{m}_1\}$, wobei \mathbf{m}_1 ein ungefärbter Knoten sei. ;
 - 2.2.4 $E := E \cup \{(\mathbf{m}, t, \mathbf{m}_1)\}$;
3. Der Algorithmus terminiert mit Ergebnis. ($G(\mathcal{N})$ ist der Überdeckungsgraph.)

Kantenfolge zusammengefügte Beschriftung $w \in T^*$ hat. Wir schreiben $\mathbf{m}_1 \xrightarrow{*} \mathbf{m}_2$, wenn es irgendeinen Pfad gibt, dessen Beschriftung uns nicht wichtig ist.

Beispiel 7.15 Abb. 7.5 zeigt ein P/T-Netz \mathcal{N} mit Überdeckungsgraph.

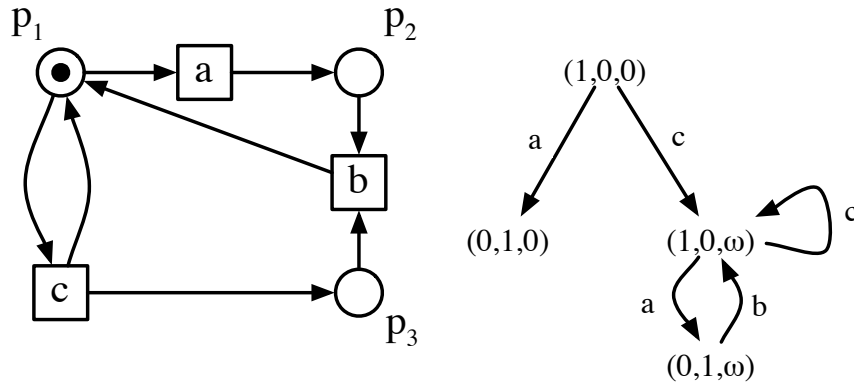


Abbildung 7.5: Ein Netz \mathcal{N} mit Überdeckungsgraph

Die Bedeutung des Überdeckungsgraphen ergibt sich aus folgenden Eigenschaften:

Satz 7.16 Sei $\mathcal{N} = (P, T, F, W, \mathbf{m}_0)$ ein P/T-Netz und $G(\mathcal{N}) = (V, E)$ ein Überdeckungsgraph zu \mathcal{N} , dann ist ein Platz $p \in P$ genau dann beschränkt, wenn es keinen Knoten $\mathbf{m} \in V$ mit $\mathbf{m}(p) = \omega$ gibt.

Gilt $\mathbf{m}_0 \xrightarrow{w} \mathbf{m}$ im Netz \mathcal{N} für ein Wort $w \in T^*$, so gibt es in $G(\mathcal{N})$ einen Knoten \mathbf{m}_1 mit $\mathbf{m}_1 \geq \mathbf{m}$ und $\mathbf{m}_0 \xrightarrow{*w} \mathbf{m}_1$.

Aus dieser letzten Eigenschaft leitet sich der Name „Überdeckungsgraph“ für $G(\mathcal{N})$ ab, denn zu jedem in \mathcal{N} erreichbaren Knoten $\mathbf{m} \in \text{RG}(\mathcal{N})$ gibt es einen, i.A. anderen, in

$G(\mathcal{N})$, der diesen „überdeckt“. Um diesen Sachverhalt zu beweisen, zeigen wir zunächst die Termination dieses Algorithmus.

Satz 7.17 *Der Algorithmus 7.4 zur Konstruktion von $G(\mathcal{N})$ terminiert.*

Beweis: Angenommen, der Algorithmus terminiere *nicht*, dann ist $|V|$ unendlich.

Begründung: Wäre $|V|$ endlich, so auch die Menge $V \times T$ und alle Paare $(m, t) \in (V \times T)$ wären einmal nach Eintritt in die **while**-Schleife ausgewählt worden und die Termination wird erreicht.

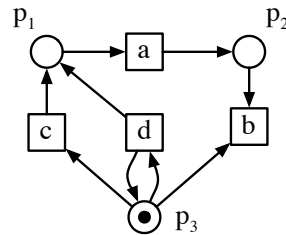
Nun sind nach Konstruktion alle Knoten $\mathbf{m} \in V$ von \mathbf{m}_0 aus erreichbar. Betrachten wir den spannenden Baum $B(\mathcal{N})$ von $G(\mathcal{N})$, der dadurch entsteht, dass diejenigen Kanten weggelassen werden, die im Schritt 2.2.4 zu einem schon existierenden Knoten gezeichnet werden. $B(\mathcal{N})$ entsteht also aus $G(\mathcal{N})$ durch Streichen gewisser Kanten.

Da $B(\mathcal{N})$ verzweigungsendlich ist (jeden Knoten verlassen maximal $|T|$ viele Kanten) aber selbst unendlich viele Knoten besitzt, gibt es in $B(\mathcal{N})$ nach dem Satz von König (1936) einen unendlichen Pfad $\mathbf{m}_0 \rightarrow \mathbf{m}_1 \rightarrow \mathbf{m}_2 \rightarrow \dots \rightarrow \mathbf{m}_i \rightarrow \mathbf{m}_{i+1} \rightarrow \dots$ in dem kein Knoten \mathbf{m}_i zweimal vor kommt.

In jeder unendlichen Folge von paarweise verschiedenen Vektoren $\mathbf{m}_i \in \mathbb{N}^P$ gibt es, nach einem Satz von Dickson (1926), eine unendliche Teilfolge $\mathbf{m}_{i_1} \xrightarrow{+} \mathbf{m}_{i_2} \xrightarrow{+} \mathbf{m}_{i_3} \xrightarrow{+} \dots \xrightarrow{+} \mathbf{m}_{i_j} \xrightarrow{+} \mathbf{m}_{i_{j+1}} \xrightarrow{+} \dots$ mit der Eigenschaft $\mathbf{m}_{i_j} < \mathbf{m}_{i_{j+1}}$.

Nach Konstruktion muss $\mathbf{m}_{i_{j+1}}$ eine ω -Komponente mehr als \mathbf{m}_{i_j} haben, denn $\mathbf{m}_{i_j} \xrightarrow{*}_{B(\mathcal{N})} \mathbf{m}_{i_{j+1}}$ impliziert $\mathbf{m}_{i_j} \xrightarrow{*}_{G(\mathcal{N})} \mathbf{m}_{i_{j+1}}$ und wenn $\mathbf{m}_{i_j} < \mathbf{m}_{i_{j+1}}$ ist, wird $\mathbf{m}_{i_{j+1}}$ mindestens eine ω -Komponente mehr enthalten als \mathbf{m}_{i_j} . Dies führt zu dem gewünschten Widerspruch, denn nur endlich viele ω -Komponenten sind überhaupt möglich. Also terminiert die Konstruktion von $G(\mathcal{N})$ nach dem Algorithmus 7.4. \square

Aufgabe 7.18 Konstruieren Sie einen Überdeckungsgraphen $G(\mathcal{N})$ für das folgende P/T-Netz \mathcal{N} .



Satz 7.19 *Für ein P/T-Netz \mathcal{N} ist $\text{RG}(\mathcal{N})$ genau dann endlich, wenn kein Knoten von $G(\mathcal{N})$ eine ω -Komponente besitzt.*

Beweis: (kein $\omega \Rightarrow \text{RG}(\mathcal{N})$ ist endlich)

Kommt in keiner Komponente von Knoten aus V in $G(\mathcal{N}) = (V, E)$ die Bezeichnung ω vor, so ist $G(\mathcal{N})$ identisch mit dem Erreichbarkeitsgraph von \mathcal{N} , und, wegen der Termination des Verfahrens, ist auch die Menge $V = \mathbf{R}(\mathcal{N})$ endlich.

($\omega \Rightarrow \mathbf{RG}(\mathcal{N})$ ist unendlich)

Sei $\mathbf{m}_2 \in V$ mit $\mathbf{m}_2(s) = \omega$ für ein $s \in P$, so gibt es einen Pfad in $G(N)$

$$\mathbf{m}_0 \xrightarrow[\alpha]{*} \mathbf{m}_{3,s} \xrightarrow[\beta]{*} \mathbf{m}_s \xrightarrow[t]{*} \mathbf{m}_{1,s} \xrightarrow[\gamma]{*} \mathbf{m}_{2,s}$$

und $\mathbf{m}_{3,s} < \mathbf{m}_{1,s}$, genauer: $\mathbf{m}_{3,s}(s) < \mathbf{m}_{1,s}(s)$.

Da das Wort w auf dem Pfad von \mathbf{m}_0 über $\mathbf{m}_{3,s}$ nach \mathbf{m}_s eine Schaltfolge im Netz \mathcal{N} ist, gibt es also Wörter $\alpha, \beta, \gamma \in T^*$ und erreichbare Markierungen $\tilde{\mathbf{m}}_3, \tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_4, \dots$ mit $\mathbf{m}_0 \xrightarrow{\alpha} \tilde{\mathbf{m}}_3 \xrightarrow{\beta t} \tilde{\mathbf{m}}_1 \xrightarrow{\gamma} \tilde{\mathbf{m}}_4 \dots$

Da $\tilde{\mathbf{m}}_3(s) < \tilde{\mathbf{m}}_1(s) < \tilde{\mathbf{m}}_4(s) < \dots$ gilt, ist die Platz $s \in P$ also in \mathcal{N} nicht beschränkt und $\mathbf{R}(\mathcal{N})$ ist keine endliche Menge. \square

Wir erhalten aus den vorangegangenen Resultaten insbesondere die folgende Aussage.

Korollar 7.20 *Das Beschränktheitsproblem ist entscheidbar.*

7.2.2 Aussagekraft des Überdeckungsgraphen

Jede Schaltfolge w des Netzes besitzt einen gleichbeschriebenen Pfad w im Überdeckungsgraph. Die Umkehrung gilt im Allgemeinen jedoch nicht, wie das folgende Netz in Abbildung 7.6 zeigt. Das Netz hat den folgenden Erreichbarkeitsgraph:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \xrightleftharpoons[b]{a} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \xrightleftharpoons[b]{a} \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdots \xrightleftharpoons[b]{a} \begin{pmatrix} 1 \\ n \end{pmatrix} \xrightleftharpoons[b]{a} \cdots$$

und den folgenden Überdeckungsgraph:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \xrightarrow{a} \begin{pmatrix} 1 \\ \omega \end{pmatrix} \xrightarrow{b} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Man beachte, dass es keine mit b beschriftete Kante von $\begin{pmatrix} 1 \\ \omega \end{pmatrix}$ nach $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ gibt. Nun ist beispielsweise $w = abb$ ein Kantenzug im Überdeckungsgraph, dieser ist aber nicht in \mathbf{m}_0 aktiviert.



Abbildung 7.6: Beispielnetz

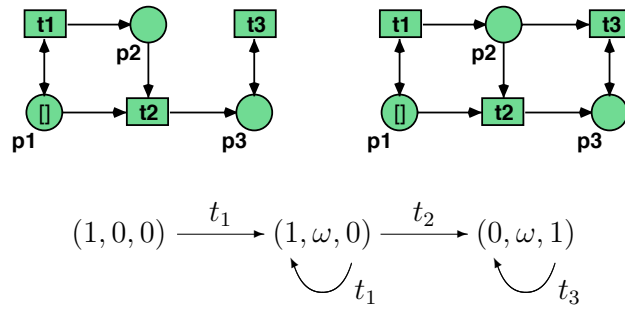
Der Überdeckungsgraph liefert somit die Möglichkeit, Nichterreichbarkeit einer Markierung nachzuweisen, denn Nicht-Überdeckbarkeit impliziert Nicht-Erreichbarkeit. Allerdings ist Erreichbarkeit nicht auf diese Art und Weise nachweisbar und auch nicht Lebendigkeit.

Theorem 7.21 *Es ist mit Hilfe des Überdeckungsgraphen nicht entscheidbar,*

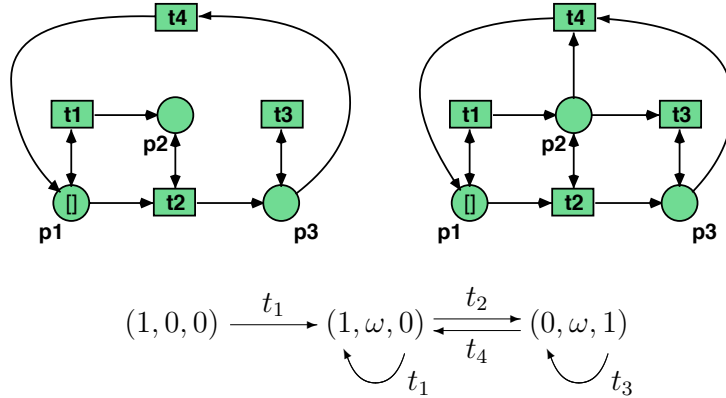
1. *ob eine Transition lebendig ist,*
2. *ob ein Netz lebendig ist, oder*
3. *ob eine Markierung \mathbf{m}' erreichbar ist.*

Beweis: Dies zeigen wir durch die Angabe zweier Netzpaare mit gleichem Überdeckungsgraphen, bei dem ein Netz die Eigenschaft besitzt, das andere aber nicht.

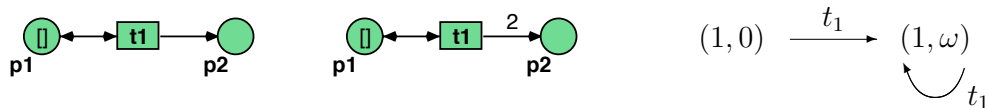
1. Im linken Netz ist t_3 lebendig, im rechten aber nicht.



2. Das linke Netz ist lebendig, das rechte ist nach der Schaltfolge $t_1 t_2 t_3$ tot.



3. Im linken Netz ist die Markierung $(1, 1)^{tr}$ erreichbar, im rechten dagegen nicht.



Also sind die Eigenschaften nicht mittels Überdeckungsgraphen entscheidbar. \square

7.2.3 Komplexität des Beschränktheitsproblem

Wie wir gesehen haben, ist das Beschränktheitsproblem entscheidbar. Wie groß ist der Berechnungsaufwand? Folgende Ergebnisse sind bekannt:

Satz 7.22 (Rackoff, 1978) *Das Beschränktheitsproblem ist mit $O(2^{c \cdot n \cdot \log(n)})$ Platzbedarf entscheidbar.*

Hierbei bezeichnet n die Größe des P/T-Netzes, d.h. z.B. die Länge seiner textuellen Beschreibung (Kodierung).

Eine untere Schranke wurde schon zuvor von Lipton gefunden:

Satz 7.23 (Lipton, 1976) *Das Beschränktheitsproblem benötigt für seine Entscheidung mindestens $O(2^{c \cdot \sqrt{n}})$ Platzbedarf.*

Ein besseres Ergebnis ist von Rosier und Yen bewiesen worden:

Satz 7.24 (Rosier und Yen, 1986) *Das Beschränktheitsproblem kann für ein gegebenes P/T-Netz $\mathcal{N} = (P, T, F, W, \mathbf{m}_0)$ mit*

$$O\left(2^{c \cdot |P| \cdot \log(|P|)} \cdot (\log(|T|) + \max_{x,y \in P \cup T} (|W(x,y)|))\right)$$

Platzbedarf entschieden werden.

Für festes $|P|$ ist das Problem PSPACE-vollständig.

Erst für spezielle Teilklassen der Petrinetze kann man eine niedrigere Komplexität für dieses Entscheidungsproblem erwarten.

Definition 7.25 *Ein P/T-Netz $\mathcal{N} = (P, T, F, W, \mathbf{m}_0)$ heißt konfliktfrei, falls es keinen Konfliktplatz enthält, d.h. einen Platz p mit $|p^\bullet| > 1$.*

Satz 7.26 (Rosier et. al., 1987) *Das Beschränktheitsproblem kann für konfliktfreie Petrinetze mit $O(n^{1,5})$ Platzbedarf entschieden werden.*

7.3 Zustandsraumexplosion bei nebenläufigen Systemen

Es gibt Petrinetze mit im Verhältnis zu ihrer Größe sehr großer Erreichbarkeitsmenge, wie das folgende Ergebnis zeigt. Diese Erscheinung ist unter dem Begriff *Zustandsraumexplosion* bekannt.

Satz 7.27 *Es gibt eine unendliche Folge von beschränkten Petri-Netzen $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \dots$, deren Größe – definiert durch $(|P| + |T| + |F|)$ – linear wächst, während die Größe ihrer Erreichbarkeitsmengen $|\mathbf{R}(\mathcal{N}_1)|, |\mathbf{R}(\mathcal{N}_2)|, |\mathbf{R}(\mathcal{N}_3)|, \dots$ schneller als jede primitiv rekursive Funktion wächst.*

Eine nicht primitiv rekursiv berechenbare Funktion ist die folgende Variante der Ackermann-Funktion:

$$A(0, n) := 2n + 1, A(m + 1, 0) := 1, A(m + 1, n + 1) := A(m, A(m + 1, n))$$

Die Netze \mathcal{N}_i berechnen gerade immer Markenzahlen auf einem festgelegten Platz, die bis an den Wert von $A(i, 2)$ heranreichen.

Aus diesem Ergebnis folgt, dass das eben angegebene Entscheidungsverfahren (Konstruktion 7.4) für die Endlichkeit der Erreichbarkeitsmenge eines P/T-Netzes mit dem Überdeckungsgraphen nicht primitiv-rekursiv ist!

```
function hexp( $n$ ) is
  val := 1
  for  $i := 1$  to  $n$  do
    cnt := val
    while  $cnt > 0$  do
      val :=  $2 * val$ 
      cnt :=  $cnt - 1$ 
    endwhile
  endfor
  return val
```

Abbildung 7.7: Hyper-exponentielles Wachstum

Wir illustrieren im Folgenden die *Zustandsraumexplosion* an einem Beispiel. Dazu betrachten wir zunächst den Algorithmus in Abbildung 7.7. Wir indizieren die Werte der Variablen mit dem Schleifenindex i . Der Initialwert für $i = 0$ ist $val_0 = 1$. Für $i = 1$ ergibt sich $cnt_1 = 1$ und nach Durchlauf der While-Schleife $val_1 = 2$. Für $i = 2$ wird $cnt_2 = val_1 = 2$ und nach Durchlauf der Schleife ist:

$$val_2 = \underbrace{2 \cdots 2}_{cnt_2\text{-mal}} val_1 = 2^{val_1} val_1$$

Allgemein gilt die Rekursionsgleichung:

$$val_{i+1} = val_i \cdot 2^{val_i} \quad \text{und} \quad val_0 = 1$$

Wir wollen das Wachstum von val_i studieren und schätzen daher $val_i \cdot 2^{val_i}$ durch $v_i := 2^{v_i}$, $v_0 = 1$ von unten ab. Es ist $v_0 = 1$, $v_1 = 2$, $v_2 = 2^2$, $v_3 = 2^{2^2}$, $v_4 = 2^{2^{2^2}}$ usw. Abwickeln der Rekursion ergibt also:

$$v_n = 2^{v_{n-1}} = 2^{2^{v_{n-2}}} = 2^{2^{2^{v_{n-3}}}} = 2^{2^{2^{2^{v_{n-4}}}}} \dots \}^{n\text{-mal}}$$

Also bestimmt das Argument n die Höhe des Exponentenstapels. Ein solches Wachstum nennt man hyper-exponentiell.

Wir können natürlich auch einen anderen Faktor als 2 wählen. Dazu müssen wir die Programmzeile $val := 2 * val$ nur gegen $val := k * val$ für ein beliebiges $k \geq 2$ austauschen. Allgemein gilt dann die Rekursionsgleichung:

$$f(i+1) = f(i) \cdot k^{f(i)} \quad \text{und} \quad f(0) = 1$$

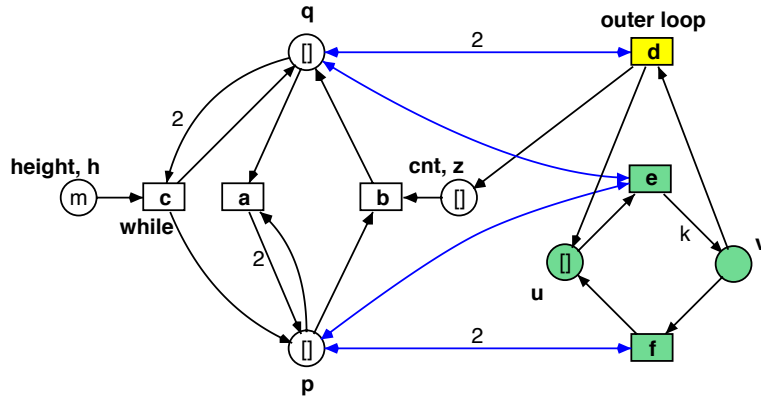


Abbildung 7.8: P/T Netz \mathcal{N}_7 mit hyper-exponentieller Beschränkung

Wir wollen jetzt ein Netz betrachten, das im Wesentlichen den obigen Algorithmus zum hyper-exponentiellen Wachstum nachbildet. Das Netz \mathcal{N}_7 aus Abb. 7.8 besitzt zwei Stellen u und v , die das mit Hilfe der Transitionen e und f die Anweisung $val := k * val$ simulieren. Die Anweisung wird dabei in zwei Teile aufgespaltet:

$$tmp := k * val; val := tmp$$

Die Zählerstelle z simuliert die Anzahl der Schleifendurchläufe. Das Netz hat Operationsmodi: p^2 , pq und q^2 . Im Modus pq wird die erste Zuweisung dieser Anweisung der While-Schleife simuliert (Transition e), im Modus p^2 wird die zweite Zuweisung simuliert. Im Modus q^2 wird durch die Transition d die Anweisung $cnt := val$ simuliert.

Dieses Netz zeigt, dass eine alleinige Beschränkung der Kapazität der Plätzen in der Regel nicht ausreicht, um niedrige Komplexitätseigenschaften zu erreichen, denn das Netz \mathcal{N}_7 hat stets eine endliche Erreichbarkeitsmenge, diese ist jedoch so groß, dass alle Petrinetz-Analysewerkzeuge auf ihre Grenzen hin getestet werden können.

7.4 Inhibitor-Netze, Zählerprogramme und Turing-Mächtigkeit

Wir diskutieren die Frage: Sind Petrinetze genauso mächtig wie Turing-Maschinen? Im affirmativen Fall hätte dies den **Vorteil**, dass Petrinetze alle berechenbaren Prozeduren ausführen könnten und der Modellierer in dieser Hinsicht nicht eingeschränkt wird. Andererseits hätte dies den **Nachteil**, dass viele Eigenschaften wie bei Turing-Maschinen nicht entscheidbar sind, wie z.B. Beschränktheit, die Erreichbarkeit eines Zustandes (einer Markierung) oder Lebendigkeit.

Beginnen wir, diese Frage bei P/T-Netzen zu diskutieren. Diese sind leichter mit *Zählerautomaten* zu vergleichen als direkt mit Turing-Maschinen. Bekanntlich sind schon Zählerautomaten mit nur zwei Zählern so mächtig wie Turing-Maschinen¹. In der Abbildung 7.9 a) ist als Beispiel ein Zählerautomat mit drei Zählern x , y und z gegeben. Die

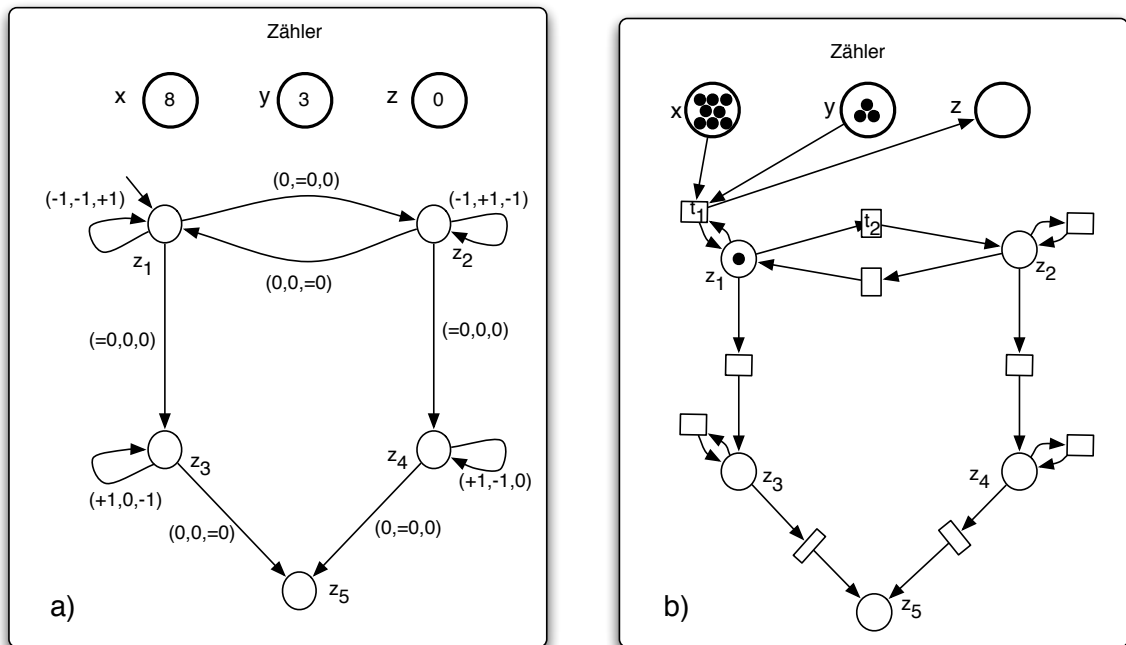


Abbildung 7.9: 3-Zählerautomat und P/T-Netz für $x := x \bmod y$

Zähler haben Werte aus den natürlichen Zahlen \mathbb{N} . Gesteuert wird er durch ein Transitionssystem mit endlich vielen (hier fünf) Zuständen, wobei z_1 der Anfangszustand ist. Die Transitionen haben die Form (a, b, c) , wobei sich a , b und c jeweils auf die Zähler x , y und z beziehen. Die Einträge a , b und c haben Werte aus $\{+1, -1, 0\}$.

¹Zählerautomaten mit nur *einem* Zähler sind mächtiger als endliche Automaten, aber weniger mächtig als Kellerautomaten, die alle kontextfreien Sprachen akzeptieren können.

Diese Werte bedeuten folgendes:

- Falls der Wert $+1$ ist, wird der jeweilige Zähler um eins erhöht.
- Falls der Wert -1 ist, ist die Transition nur möglich, wenn der jeweilige Zählerwert positiv ist. In diesem Fall wird der Wert um eins verringert.
- Falls der Wert 0 ist, wird der jeweilige Zählerwert nicht verändert.
- Falls der Wert $= 0$ ist, ist die Transition nur möglich, wenn der jeweilige Zählerwert Null ist. In diesem Fall wird der Wert ebenfalls nicht verändert.

Zum Beispiel bewirkt die Transition $z_1 \xrightarrow{(-1, -1, +1)} z_1$, dass die Zähler x und y um eins verringert werden, falls ihr Wert positiv ist und im selben Schritt der Zähler z um eins erhöht wird. Dies ist bei der gegebenen Anfangskonfiguration drei mal möglich, wonach nur die Transition $z_1 \xrightarrow{(0, = 0, 0)} z_2$ erfolgen kann und danach entsprechendes im Zustand z_2 passiert. Offensichtlich wird insgesamt der Anfangswert von x so oft wie möglich um den Anfangswert von y verringert und der verbleibende Rest nach x gebracht, wonach der Automat in den Endzustand z_5 übergeht. Dadurch wird die Zuweisung $x := x \bmod y$ berechnet. Da bei der Reduktion von x durch y der Wert von y verloren geht, wird er in z wieder hergestellt, was in der folgenden Phase in umgekehrter Weise geschieht.

Kann jeder Zählerautomat durch ein P/T-Netz simuliert werden? Die Abbildung 7.9 b) zeigt den Anfang einer solchen Konstruktion. Die Zähler werden durch entsprechende Plätze x , y und z dargestellt, deren Markenzahl dem Zählerinhalt entspricht. Die Plätze z_i stellen das Transitionssystem zur Steuerung dar. Sie können insgesamt nur eine Marke enthalten². Der oben beschriebenen Transition $z_1 \xrightarrow{(-1, -1, +1)} z_1$ entspricht die Netztransition t_1 . Die Wirkung der anderen Transitionen auf die Zähler müsste entsprechend ergänzt werden.

Wäre diese Konstruktion erfolgreich durchführbar, so bestünde ein gravierender logischer Widerspruch. Das Beschränktheitsproblem bei Zählerautomaten ist nämlich (wie bei Turing-Maschinen) unentscheidbar, während in diesem Kapitel gezeigt wurde, dass es für P/T-Netze entscheidbar ist. P/T-Netze können also nicht Turing-Maschinen simulieren. Wo scheitert der obige Konstruktionsversuch? Um der Transition t_2 in Abbildung 7.9 b) das Verhalten der entsprechenden Transition $z_1 \xrightarrow{(0, = 0, 0)} z_2$ in a) zu geben, müsste geprüft werden, ob der Platz y keine Marke enthält. Ist ein solcher Test bei P/T-Netzen möglich?

Die Abbildung 7.10 a) zeigt die Situation *in nuce*. Der Ausschnitt des steuernden Transitionssystems ist durch die Plätze z_1 , z_2 und z_3 gegeben. Dabei soll die Marke in z_1 genau dann nach z_2 gebracht werden, wenn der Platz („der Zähler“) y mindestens eine Marke enthält. Beim Schalten der Transition **not null** funktioniert dies, jedoch nicht bei der Transition **null**. Eine Lösung des Problems erfolgt üblicherweise mit Hilfe der Hinzufügung eines *komplementären Platzes* y' wie in Abbildung 7.10 b). Dies ist ein Platz der die P-Invarianten-Gleichung $y + y' = k$ erfüllt, wobei k wie immer aus der Anfangsmarkierung zu berechnen ist (hier $k = 2$). Folglich kann die Transition **null** genau dann schalten, wenn $\mathbf{m}(y') \geq 2$, d.h. – wie gewünscht – $\mathbf{m}(y) = 0$ ist. Diese Konstruktion ist wegen der P-Invarianten-Gleichung jedoch nur möglich, wenn der Platz y beschränkt

²Sie erfüllen die P-Invariante $z_1 + z_2 + z_3 + z_4 + z_5 = 1$.

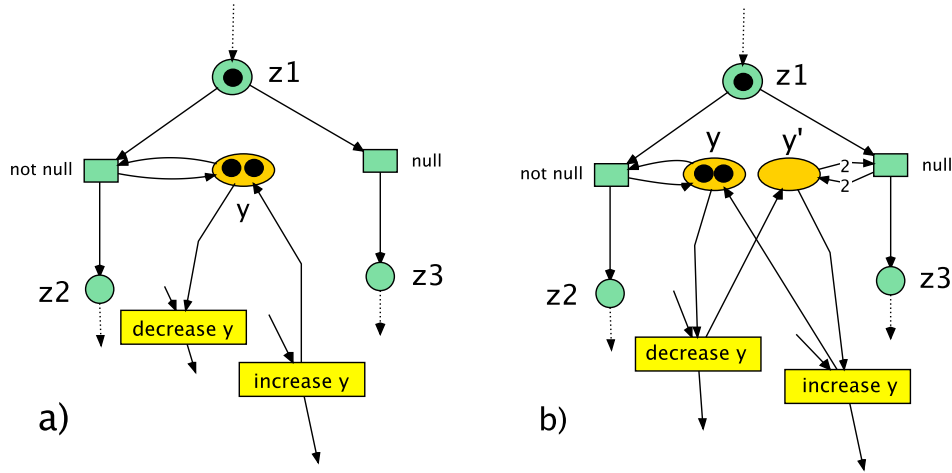


Abbildung 7.10: Nulltest-Situation für P/T-Netze in a) und Lösung in b)

ist, eine Forderung, die bei der Simulation von Turing-mächtigen Zählerautomaten nicht zu erfüllen ist³. Generell ist ein solcher *Nulltest* bei P/T-Netzen nicht möglich, d.h. sie sind weniger mächtig als Turing-Maschinen.

Um Nulltests zu ermöglichen, wurden für P/T-Netze *Nulltest-Kanten* oder *Inhibitor-Kanten* wie in Abbildung 7.11 a) eingeführt. Hier kann *per definitionem* die Transition t nur schalten, wenn der mit ihr über die Inhibitor-Kante verbundene Platz y leer ist. P/T-Netze, bei denen Inhibitor-Kanten erlaubt sind heißen *Inhibitor-Netze*. Eine Lösung des Nulltest-Problems für Inhibitor-Netze ist in Abbildung 7.11 b) dargestellt.

Inhibitor-Netze haben neben diesem Vorteil natürlich den Nachteil, dass viele Probleme unentscheidbar werden oder die Analysekomplexität erheblich steigt. Dies gilt auch für gefärbte Netze, für die im Vorgriff auf ein späteres Kapitel eine Lösung des Nulltest-Problems in Abbildung 7.11 c) gegeben ist. Eine solche Lösung setzt die Existenz einer nicht endlichen Farbmeng voraus, wie hier die Farbe **integer** für den Platz y . Gefärbte Netze mit nur endlichen Farbmengen sind gleichmächtig zu P/T-Netzen, da sie immer zu solchen aufgefaltet werden können. Eine Übersicht über die Turing-Mächtigkeit bei Petrinetzen gibt die Tabelle 7.1.

³Wenn ein Zähler beschränkt ist, kann man ihn eliminieren, indem seine Funktion in das endliche steuernde Transitionssystem integriert wird. Wenn alle Zähler beschränkt sind, ist der Zählerautomat nur so mächtig wie ein endlicher Automat.

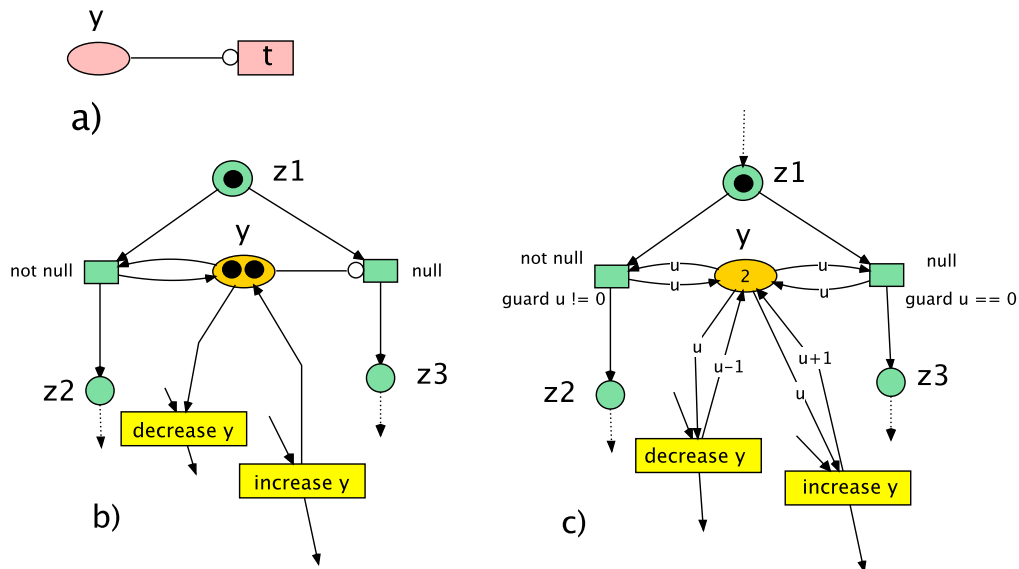


Abbildung 7.11: Nulltest in Inhibitor- und gefärbten Netzen

	Turing-mächtig?
P/T-Netze	nein
Inhibitor-Netze	ja
gefärbte Netze mit endlichen Farbmengen	nein
gefärbte Netze mit beliebigen Farbmengen	ja

Tabelle 7.1: Turing-Mächtigkeit von Petrinetzen

7.5 Strukturelle Eigenschaften

Netzinvarianten erlauben die Verifikation von Markierungsinvarianz-Eigenschaften ohne die oft exponentielle Konstruktion des Erreichbarkeitsgraphen.

7.5.1 Stelleninvarianten

Beziehungen zwischen Programmvariablen, die bei der Ausführung eines Programms erhalten bleiben, heißen Programminvarianten. Ihre Nützlichkeit zum Nachweis von Programmeigenschaften ist aus der sequentiellen Programmierung bereits hinreichend bekannt. Da das Verhalten von nebenläufigen Programmen weitaus komplexer und unübersichtlicher ist, sind Invarianten von entsprechend größerer Bedeutung, vorausgesetzt, es gelingt solche Gesetzmäßigkeiten zu erkennen. Eine Besonderheit von P -Invarianten (bzw. S -Invarianten) liegt darin, dass sie berechenbar sind (im Gegensatz zu allgemeinen Programminvarianten). Die folgenden Abschnitte stammen aus [JV87].

Als Beispiel betrachten wir das Leser/Schreiber-Problem. Die folgende Abb. 7.12 a) zeigt ein entsprechendes Modell als P/T-Netz für n Aufträge ($n \in \mathbb{N}, n > 0$), die Anfangs in der Stelle (dem Platz) lok liegen, was ausdrückt, dass sie noch nichts mit dem kritischen Abschnitt zu tun haben, auf den sie später lesend oder schreibend zugreifen. Durch das Schalten der Transition a bzw. d melden sie sich als Lese- bzw. Schreibaufträge an, d.h. die entsprechende Marke liegt in la bzw. sa , was „zum Lesen angemeldet“ bzw. „zum Schreiben angemeldet“ heißen soll. Dann folgt der Zugriff auf die kritischen Daten mit den bekannten Spezifikationen:

- a) Wenn ein Schreiber schreibt, darf kein Leser lesen
- b) Es darf höchstens ein Schreiber schreiben
- c) Wenn ein Leser liest, darf kein Schreiber schreiben

Für alle erreichbaren Markierungen $\mathbf{m} \in \mathbf{R}(\mathcal{N})$ sollen also folgende Spezifikationen gelten:

$$\mathbf{m}(s) > 0 \Rightarrow \mathbf{m}(l) = 0 \quad (7.1)$$

$$\mathbf{m}(s) \leq 1 \quad (7.2)$$

$$\mathbf{m}(l) > 0 \Rightarrow \mathbf{m}(s) = 0 \quad (7.3)$$

Wir werden die Gültigkeit dieser Bedingungen im nächsten Abschnitt beweisen.

Es gibt verschiedene Varianten des Leser/Schreiber-Problems. Um eine möglichst schnelle Aktualisierung der Daten zu gewährleisten, kann man z.B. den Schreiber priorisierten Zugriff einräumen. Abbildungen 7.12 b) zeigt die entsprechende Erweiterung: Sobald mindestens ein Schreiber angemeldet ist oder schreibt, darf kein neuer Leser anfangen zu lesen (denn b ist wegen $m(p) < n$ gesperrt).

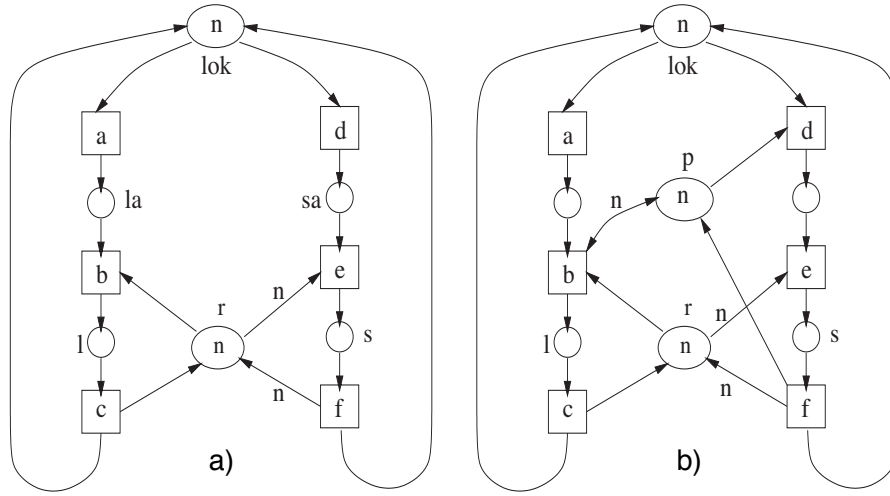


Abbildung 7.12: Leser/Schreiber-Problem in a) und mit Priorität für Schreibaufträge in b)

Für das P/T-Netz des Leser/Schreiber-Problems in Abb. 7.12 a) gelten die Gleichungen ($n \geq 1$):

$$i_1 : \quad lok + la + sa + l + s = n \quad (7.4)$$

$$i_2 : \quad l + r + n \cdot s = n \quad (7.5)$$

lok, la, \dots stehen hier abkürzend für $\mathbf{m}(lok), \mathbf{m}(la), \dots$. Gleichungen dieser Form heißen *P-Invarianten-Gleichungen*⁴.

Definition 7.28 Es sei $\mathcal{N} = (P, T, F, W, \mathbf{m}_0)$ ein P/T-Netz mit $P = \{p_1, \dots, p_n\}$. Eine Gleichung der Form $\sum_{i=1}^n k_i \cdot \mathbf{m}(p_i) = k$ mit $k_i, k \in \mathbb{Z}$ heißt *P-Invarianten-Gleichung*, wenn sie für alle erreichbaren Markierungen $\mathbf{m} \in \mathbf{R}(\mathcal{N})$ gilt. Abkürzend wird eine P-Invarianten-Gleichung auch als $\sum_{i=1}^n k_i \cdot p_i = k$ geschrieben.

Eine P-Invarianten-Gleichung ist also eine spezielle Markierungs-Invarianzeigenschaft (Definition 7.4 auf Seite 124). Der Beweis, dass obige P-Invarianten-Gleichungen für alle erreichbaren Markierungen gelten, stellen wir zurück, leiten aber aus ihnen die Synchronisationsspezifikationen ab.

- Spezifikation 7.1:⁵ $s > 0 \rightarrow l = 0$ folgt aus i_2 .
- Spezifikation 7.2: $s \leq 1$ folgt aus i_2 .
- Spezifikation 7.3: $l > 0 \rightarrow s = 0$ folgt aus i_2 .

⁴Zur Unterscheidung zu dem bald definierten P-Invarianten-Vektor.

⁵Wieder steht $s > 0$ für $\mathbf{m}(s) > 0$ usw.

Damit ist nachgewiesen, dass das Netz von Abb. 7.12 a) die Synchronisationsspezifikationen des Leser/Schreiber-Problems erfüllen. Wir zeigen nun, wie man die Gültigkeit der P-Invarianten-Gleichungen nachweist. Dazu betrachten wir das P/T-Netz von Abb. 7.12 a) und die P-Invarianten-Gleichung i_2 :

$$i_2 : \mathbf{m}(l) + \mathbf{m}(r) + n \cdot \mathbf{m}(s) = n \quad (7.6)$$

Sie gilt für die Anfangsmarkierung \mathbf{m}_0 .

Als Induktionsbeweis zeigt man dann: Gilt 7.6 für eine Markierung $\mathbf{m}_1 \in \mathbf{R}(\mathcal{N})$ und ändert eine Transition t die Markierung \mathbf{m}_1 durch Schalten zu \mathbf{m}_2 (also $\mathbf{m}_1 \xrightarrow{t} \mathbf{m}_2$), dann gilt 7.6 auch für \mathbf{m}_2 .

- Für $t = e$ gilt:

$$\begin{aligned} \mathbf{m}_2(sa) &= \mathbf{m}_1(sa) - 1 \\ \mathbf{m}_2(r) &= \mathbf{m}_1(r) - n \\ \mathbf{m}_2(s) &= \mathbf{m}_1(s) + 1, \end{aligned} \quad (7.7)$$

während alle anderen Stellen unverändert bleiben.

Also: Gilt 7.6 für \mathbf{m}_1 , dann auch für \mathbf{m}_2 .

- Dies ist analog für alle Transitionen durchzuführen.

Um dies systematischer zu behandeln, definieren wir die *Wirkung* einer Transition.

Definition 7.29 Es sei $\mathcal{N} = (P, T, F, W, \mathbf{m}_0)$ ein P/T-Netz.

Der Vektor $\Delta_{\mathcal{N}}(t) \in \mathbb{Z}^{|P|}$ heißt *Wirkung der Transition* $t \in T$ und ist definiert durch

$$\Delta_{\mathcal{N}}(t)(p) = -\widetilde{W}(p, t) + \widetilde{W}(t, p)$$

Die durch Aneinanderreihung der Vektoren $\Delta_{\mathcal{N}}(t_1) \dots \Delta_{\mathcal{N}}(t_{|T|})$ gebildete $(|P| \times |T|)$ -Matrix $\Delta_{\mathcal{N}}$ heißt *Wirkungsmatrix* oder *Inzidenzmatrix*. $\Delta_{\mathcal{N}}(t)$ ist dann die t -Spalte von $\Delta_{\mathcal{N}}$.

Der Name *Inzidenzmatrix* ist durch die Darstellung von \mathcal{N} als Graph zu erklären, während die Bezeichnung *Wirkung* durch den folgenden Satz deutlich wird.

Satz 7.30 Wenn t die Markierung \mathbf{m}_1 durch Schalten in \mathbf{m}_2 überführt ($\mathbf{m}_1 \xrightarrow{t} \mathbf{m}_2$), dann gilt:

$$\mathbf{m}_2 = \mathbf{m}_1 + \Delta_{\mathcal{N}}(t)$$

Beweis: Der Beweis folgt direkt durch Vergleich der Definitionen 6.17 und 7.29. \square

Von Invarianten bei Programmen ist bekannt, dass sie nicht algorithmisch aus dem Programm gewonnen werden können. Einer der Vorteile der Darstellung von Synchronisations-Problemen durch P/T-Netze beruht darauf, dass Netz-Invarianten berechnet werden können. Die Grundlage dazu liefert der folgende *Satz von Lautenbach*.

Definition 7.31 Es sei M^{tr} die Transponierte einer Matrix M und $\underline{0} \in \mathbb{Z}^{|T|}$ der Nullvektor.

Jede ganzzahlige Lösung $i \in \mathbb{Z}^{|P|} \setminus \{\underline{0}\}$ des linearen Gleichungssystems $\Delta_{\mathcal{N}}^{tr} \cdot i = \underline{0}$ heißt P -Invarianten-Vektor⁶ des P/T -Netzes \mathcal{N} .

Satz 7.32 (Lautenbach) Wenn $i \in \mathbb{Z}^{|P|} \setminus \{\underline{0}\}$ ein P -Invarianten-Vektor ist, dann gilt $i^{tr} \cdot \mathbf{m} = i^{tr} \cdot \mathbf{m}_0$ für alle erreichbaren Markierungen $\mathbf{m} \in \mathbf{R}(\mathcal{N})$.

Beweis: Induktion über $\mathbf{R}(\mathcal{N})$.

Anfang: Die Behauptung ist trivial für $\mathbf{m} = \mathbf{m}_0$.

Annahme: Es gelte die Behauptung für $\mathbf{m}_1 \in \mathbf{R}(\mathcal{N})$, also $i^{tr} \cdot \mathbf{m}_1 = i^{tr} \cdot \mathbf{m}_0$, und es gelte $\mathbf{m}_1 \xrightarrow{t} \mathbf{m}_2$ für eine Transition $t \in T$.

Schritt: Aus der Voraussetzung $\Delta_{\mathcal{N}}^{tr} \cdot i = \underline{0}$ folgt dann $\underline{0}^{tr} = (\Delta_{\mathcal{N}}^{tr} \cdot i)^{tr} = i^{tr} \cdot (\Delta_{\mathcal{N}}^{tr})^{tr} = i^{tr} \cdot \Delta_{\mathcal{N}}$ und damit $i^{tr} \cdot \Delta_{\mathcal{N}}(t) = 0$.

Also gilt mit Satz 7.30 die Induktionsbehauptung:

$$\begin{aligned} i^{tr} \cdot \mathbf{m}_2 &= i^{tr} \cdot (\mathbf{m}_1 + \Delta_{\mathcal{N}}(t)) \\ &= i^{tr} \cdot \mathbf{m}_1 + i^{tr} \cdot \Delta_{\mathcal{N}}(t) \\ &= i^{tr} \cdot \mathbf{m}_1 \\ &= i^{tr} \cdot \mathbf{m}_0 \end{aligned}$$

□

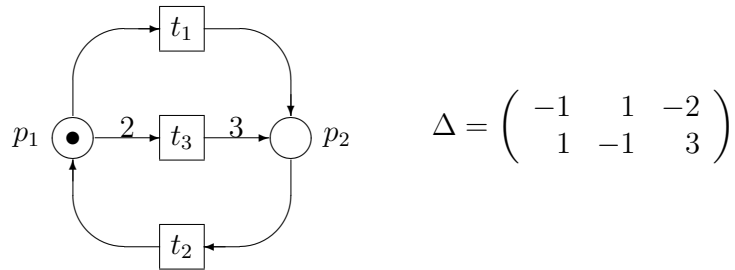


Abbildung 7.13: Beispielnetz

Aus einer P -Invarianten-Gleichung $\sum_{i=1}^n k_i \cdot \mathbf{m}(p_i) = k$ kann man aber nicht schließen, dass der Vektor (k_1, \dots, k_n) eine P -Invariante ist. Dies kann daran liegen, dass in den Schaltfolgen Transitionen, die die Eigenschaft zerstören könnten, nie aktiviert sind. Das Netz in Abbildung 7.13 erfüllt für alle aus der Anfangsmarkierung $\mathbf{m}_0 = (1, 0)^{tr}$ erreichbaren Markierungen die Invariantengleichung:

$$1 \cdot \mathbf{m}(p_1) + 1 \cdot \mathbf{m}(p_2) = 1 \cdot \mathbf{m}_0(p_1) + 1 \cdot \mathbf{m}_0(p_2) = 1$$

⁶Im Gegensatz zur P -Invarianten-Gleichung in Def. 7.28.

Der zugehörige Vektor $i = (1, 1)^{tr}$ ist jedoch kein Invariantenvektor, denn

$$\Delta^{tr} i = \begin{pmatrix} -1 & 1 \\ 1 & -1 \\ -2 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Man beachte, dass für dieses Beispiel die Anfangsmarkierung gerade so gewählt ist, dass in keiner erreichbaren Markierung t_3 aktiviert ist. Für $\mathbf{m} = (2, 0)$ ist t_3 aktiviert, und die Invariantengleichung ist ungültig. Eine Invariantengleichung $\sum_{i=1}^n k_i \cdot \mathbf{m}(p_i) = k$, die aus einer P -Invarianten abgeleitet wurde, gilt dagegen für jede Wahl der Initialmarkierung – für eine jeweils angepasste Konstante k .

Beispiel 7.33 Wir interpretieren Satz 7.32 anhand unseres Beispiels. Abbildung 7.14 zeigt die Inzidenzmatrix $\Delta_{\mathcal{N}}$ des Leser/Schreiber Netzes von Abb. 7.12 a) und zwei Invarianten-Vektoren i_1 und i_2 .

$\Delta_{\mathcal{N}}$	a	b	c	d	e	f		i_1	i_2
lok	-1	0	1	-1	0	1		1	0
la	1	-1	0	0	0	0		1	0
sa	0	0	0	1	-1	0		1	0
l	0	1	-1	0	0	0		1	1
s	0	0	0	0	1	-1		1	n
r	0	-1	1	0	- n	n		0	1
j	3	3	3	2	2	2			

Abbildung 7.14: Inzidenzmatrix $\Delta_{\mathcal{N}}$ mit P -Invariantenvektor i_1, i_2 und T -Invariantenvektor j

Man rechne nach: $\Delta_{\mathcal{N}}^{tr} \cdot i_1 = \underline{0}$ und $\Delta_{\mathcal{N}}^{tr} \cdot i_2 = \underline{0}$.

Folglich gilt nach Satz 7.32 für jede von der Anfangsmarkierung $\mathbf{m}_0^{tr} = (n, 0, 0, 0, 0, n)$ aus erreichbare Markierung \mathbf{m} :

$$\begin{aligned} i_2^{tr} \cdot \mathbf{m} &= 1 \cdot \mathbf{m}(l) + n \cdot \mathbf{m}(s) + 1 \cdot \mathbf{m}(r) = \\ i_2^{tr} \cdot \mathbf{m}_0 &= 1 \cdot \mathbf{m}_0(l) + n \cdot \mathbf{m}_0(s) + 1 \cdot \mathbf{m}_0(r) \\ &= 1 \cdot 0 + n \cdot 0 + 1 \cdot n = n \end{aligned}$$

Dies ist genau die P -Invarianten-Gleichung 7.6.

Wir fassen zusammen: Aus einem gegebenen Netz \mathcal{N} können durch Berechnung aller ganzzahligen Lösungen i in $\Delta_{\mathcal{N}}^{tr} \cdot i = \underline{0}$ alle P -Invarianten-Vektoren gefunden werden.

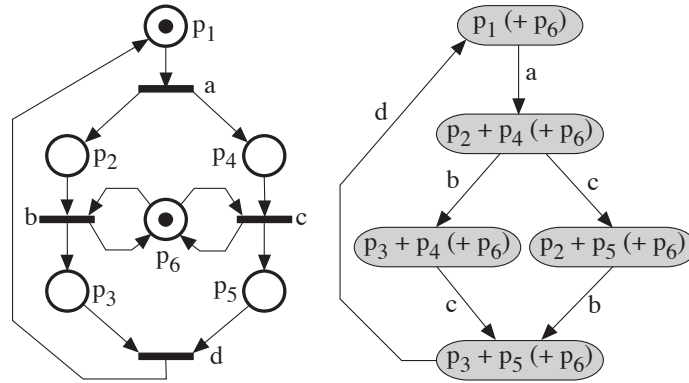


Abbildung 7.15: Beschränktes, lebendiges und reversibles Netz mit Erreichbarkeitsgraph

Für die Anfangsmarkierung \mathbf{m}_0 werden durch $i^{tr} \cdot \mathbf{m} = i^{tr} \cdot \mathbf{m}_0$ die entsprechenden P-Invarianten-Gleichung aufgestellt, die für alle $\mathbf{m} \in \mathbf{R}(\mathcal{N})$ gelten. Mit ihnen können, wie oben gezeigt, Netzeigenschaften nachgewiesen werden.

Die Umkehrung gilt aber i.a. nicht. Die P-Invarianten-Vektoren können u.a. herangezogen werden, um *strukturelle* Eigenschaften zu beweisen.

Beispiel 7.34 Für *jede* Anfangsmarkierung gelten für das Netz in Abb. 7.15 die folgenden Markenerhaltungsgesetze (P-Invariantengleichungen):

$$\begin{aligned} \mathbf{m}(p_1) + \mathbf{m}(p_2) + \mathbf{m}(p_3) &= \mathbf{m}_0(p_1) + \mathbf{m}_0(p_2) + \mathbf{m}_0(p_3) = k_1(\mathbf{m}_0) \\ \mathbf{m}(p_1) + \mathbf{m}(p_4) + \mathbf{m}(p_5) &= \mathbf{m}_0(p_1) + \mathbf{m}_0(p_4) + \mathbf{m}_0(p_5) = k_2(\mathbf{m}_0) \\ \mathbf{m}(p_6) &= \mathbf{m}_0(p_6) = k_3(\mathbf{m}_0) \end{aligned}$$

Hierbei ist \mathbf{m}_0 die Anfangsmarkierung und \mathbf{m} eine beliebige erreichbare Markierung. Die $k_i(\mathbf{m}_0) \in \mathbb{Z}$ sind jeweils eine von \mathbf{m}_0 abhängige Konstante. Daher gelten die Ungleichungen:

$$\begin{aligned} \mathbf{m}(p_1) &\leq \min(k_1(\mathbf{m}_0), k_2(\mathbf{m}_0)) \\ \mathbf{m}(p_i) &\leq k_1(\mathbf{m}_0); i = 2, 3 \\ \mathbf{m}(p_j) &\leq k_2(\mathbf{m}_0); j = 4, 5 \\ \mathbf{m}(p_6) &= k_3(\mathbf{m}_0) \end{aligned}$$

Diese Ungleichungen beweisen, dass das Netz für *jede* Anfangsmarkierung beschränkt ist. Dies ist natürlich eine stärkere Eigenschaft als Beschränktheit. Da sie nur von der Struktur des Netzes und nicht von der jeweils gewählten Anfangsmarkierung abhängt, heißt sie *strukturelle Beschränktheit* (structural boundedness).

Ein Netz heißt *strukturell beschränkt* (engl. structurally bounded), wenn es für jede Initialmarkierung beschränkt ist. Eine echt positive P-Invariante \mathbf{i} kann strukturelle Beschränktheit zeigen. Eine P-Invariante $\mathbf{i} \geq \mathbf{0}$ heißt überdeckend, wenn ihr Träger alle Stellen enthält.

Theorem 7.35 *Besitzt das P/T Netz \mathcal{N} eine überdeckende, positive P -Invariante \mathbf{i} , d.h. $\mathbf{i}(p) > 0$ für alle p , dann ist \mathcal{N} strukturell beschränkt.*

Beweis: Sei \mathbf{m}_0 eine beliebige Initialmarkierung. Mit dem Satz von Lautenbach gilt für jede erreichbare Markierung \mathbf{m} :

$$\mathbf{i} \cdot \mathbf{m} = \sum_{p \in P} \mathbf{i}(p) \cdot \mathbf{m}(p) = \sum_{p \in P} \mathbf{i}(p) \cdot \mathbf{m}_0(p)$$

Da $\mathbf{i}(p) > 0$ für jede Stelle gilt, erhalten wir:

$$\mathbf{i}(p) \cdot \mathbf{m}(p) \leq \sum_{p \in P} \mathbf{i}(p) \cdot \mathbf{m}(p)$$

Also gilt für p die Beschränkung (wieder mit $\mathbf{i}(p) > 0$):

$$\mathbf{m}(p) \leq \frac{(\mathbf{i} \cdot \mathbf{m}_0)}{\mathbf{i}(p)} \leq \mathbf{i} \cdot \mathbf{m}_0$$

Da für jedes Netz die Anfangsmarkierung fest ist, existiert also für jede Anfangsmarkierung und jeden Platz eine obere Schranke für die Markenzahl. \square

Satz 7.36 *Sei \mathcal{N} konservatives P/T -Netz, dann gilt: Aus $f(\mathbf{m}) := \sum_{p \in P} \mathbf{m}(p) \cdot f(p)$ folgt für jede erreichbare Markierung $\mathbf{m} \in \mathbf{R}(\mathcal{N})$ stets $f(\mathbf{m}_0) = f(\mathbf{m})$.*

Beweis: Konservative Netze besitzen mit f eine echt positive Stellen-Invariante. \square

Satz 7.36 impliziert die Endlichkeit der Erreichbarkeitsmenge $\mathbf{R}(\mathcal{N})$.

7.5.2 Transitionsinvarianten

Für die Wirkung einer Schaltfolge ist die Reihenfolge der Transitionen unerheblich, nur die Anzahl ist entscheidend. Die *Parikh-Abbildung* liefert zu jeder Sequenz ein Tupel mit den Häufigkeiten.

Definition 7.37 *Sei $\Sigma = \{x_1, x_2, \dots, x_k\}$ ein endliches, geordnetes Alphabet (mit $x_1 < x_2 < \dots < x_k$). Die Häufigkeit, mit der das Zeichen $x \in \Sigma$ in dem Wort $w \in \Sigma^*$ vorkommt, wird mit $|w|_x$ notiert.*

Die Parikh-Abbildung ist der Homomorphismus $\psi : \Sigma^ \rightarrow \mathbb{N}^k$, definiert durch:*

$$\psi(w) := (|w|_{x_1}, |w|_{x_2}, \dots, |w|_{x_k})$$

Der *Parikh-Vektor* $\psi(w)$ gibt in der i -ten Komponente an, wie oft das Zeichen x_i in dem Wort $w \in \Sigma^*$ vorkommt.⁷

Auch die Lösungen j von $\Delta_{\mathcal{N}} \cdot j = \underline{0}$ haben eine wichtige Interpretation für die Analyse von Systemen. (Hier $\Delta_{\mathcal{N}}$ wird *nicht* transponiert.)

⁷In manchen Büchern wird der Parikh-Vektor durch: $\#(w) := (\#_1(w), \#_2(w), \dots, \#_{|T|}(w))$ dargestellt, wobei dann natürlich $\#_i(w) := |w|_{t_i}$ gilt.

Definition 7.38 Jede Lösung $j \in \mathbb{N}^{|T|} \setminus \{\underline{0}\}$ von $\Delta_{\mathcal{N}} \cdot j = \underline{0}$ heißt T -Invarianten-Vektor des P/T -Netzes \mathcal{N} .

T -Invarianten liefern notwendige Bedingungen für die Reproduzierbarkeit von Systemen. Dabei heie eine Markierung \mathbf{m} *reproduzierbar*, wenn sie durch eine (nicht leere) Schaltfolge w von Transitionen wieder erreichbar ist. Kommt eine Transition $t_i \in T$ in w gerade $|w|_{t_i}$ mal vor, dann ist der Parikh-Vektor $\psi(w) := (|w|_{x_1}, |w|_{x_2}, \dots, |w|_{x_{|T|}})$ von dem Wort w ein T -Invarianten-Vektor. T -Invarianten beschreiben also die Hufigkeit des Schaltens jeder Transition bei reproduzierendem Verhalten.

Satz 7.39 Es seien $\mathbf{m}_1, \mathbf{m}_2$ Markierungen und $w = t_{i_1} \dots t_{i_k}$ eine Schaltfolge, die \mathbf{m}_1 in \mathbf{m}_2 berfhrt: $\mathbf{m}_1 \xrightarrow{w} \mathbf{m}_2$.

Die Markierungen \mathbf{m}_1 und \mathbf{m}_2 sind genau dann gleich, wenn es einen T -Invarianten-Vektor $j \in \mathbb{N}^{|T|}$ derart gibt, dass jede Transition $t \in T$ genau $j(t)$ mal in w vorkommt, d.h.: $j = \psi(w)$.

Beweis: Es gilt nach Voraussetzung

$$\mathbf{m}_1 = \mathbf{m}_2 = \mathbf{m}_1 + \Delta_{\mathcal{N}}(t_{i_1}) + \Delta_{\mathcal{N}}(t_{i_2}) + \dots + \Delta_{\mathcal{N}}(t_{i_k})$$

genau dann, wenn gilt:

$$\begin{aligned} \underline{0} &= \Delta_{\mathcal{N}}(t_{i_1}) + \dots + \Delta_{\mathcal{N}}(t_{i_k}) \\ &= \Delta_{\mathcal{N}} \cdot e_{t_{i_1}} + \dots + \Delta_{\mathcal{N}} \cdot e_{t_{i_k}} \\ &= \Delta_{\mathcal{N}} \cdot (|w|_{t_1} e_{t_1}) + \dots + \Delta_{\mathcal{N}} \cdot (|w|_{t_{|T|}} e_{t_{|T|}}) \\ &= \Delta_{\mathcal{N}} \cdot \psi(w) \end{aligned}$$

Mit anderen Worten: $\psi(w)$ ist eine T -Invariante. □

Fr $\mathbf{m} \xrightarrow{w} \mathbf{m}$ gilt dann also $\Delta_{\mathcal{N}} \cdot \psi(w) = \underline{0}$.

Der T -Invarianten-Vektor j in Abb. 7.14 besagt also, dass die Anfangsmarkierungen \mathbf{m}_0 dann wieder erreicht (reproduziert) wird, wenn drei Lese- und zwei Schreibauftrge ihren Zyklus durchlaufen haben.

Aufgabe 7.40 (Invarianten) Gegeben seien die P/T -Netze \mathcal{N} und \mathcal{N}' , wobei \mathcal{N} das Gesamtnetz aus der Abb. 7.16 ist und \mathcal{N}' das Teilnetz ist, dessen Elemente mit durchgezogenen Linien gezeichnet sind.

Berechnen Sie mit Hilfe der Stze 7.32 (Satz von Lautenbach) und 7.39 P -Invarianten und T -Invarianten fr das Gesamtnetz \mathcal{N} und das Teilnetz \mathcal{N}' (falls diese existieren). Untersuchen Sie, ob es in den Netzen Schaltfolgen gibt, die die Anfangsmarkierung wiederherstellen.

Theorem 7.41 Sei \mathcal{N} ein beschrnktes P/T -Netz, das in \mathbf{m} die unendliche Schaltfolge $w \in T^\omega$ aktiviert. Dann existiert eine T -Invariante j mit $j(t) = 0$, falls t nicht oder nur endlich oft in w vorkommt $j(t) > 0$, falls t unendlich oft in w vorkommt.

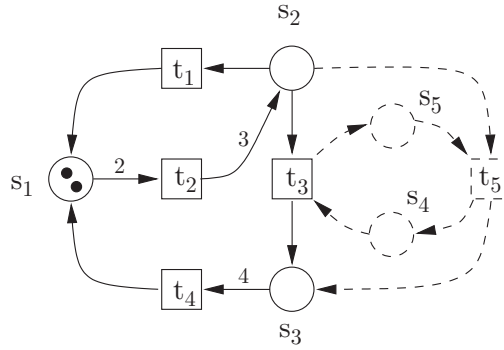


Abbildung 7.16: Netz und Teilnetz

Beweis: Für die Schaltfolge $w \in T^\omega$ gelte:

$$\mathbf{m} = \mathbf{m}_1 \xrightarrow{t_1} \mathbf{m}_2 \xrightarrow{t_2} \mathbf{m}_3 \xrightarrow{t_3} \dots$$

Sei T_ω die Menge der in w unendlich oft vorkommenden Transitionen. Sei $w = t_1 t_2 t_3 \dots$ und sei k_1 der Index, ab dem $w' = t_{k_1} t_{k_1+1} t_{k_1+2} \dots$ nur noch Transitionen aus T_ω enthält. Wir können nun w' weiter zerlegen. Wir definieren die Indexfolge k_1, k_2, k_3, \dots mit $k_1 < k_2 < k_3 < \dots$ so, dass die Schaltfolge

$$w_{k_i} = t_{k_i} t_{k_i+1} \dots t_{k_{i+1}-1}$$

alle Transitionen aus T_ω enthält.

Nun können die Markierungen \mathbf{m}_{k_i} in der Folge

$$\mathbf{m}_1 \xrightarrow{w_{k_1}} \mathbf{m}_{k_2} \xrightarrow{w_{k_2}} \mathbf{m}_{k_3} \dots$$

nicht alle verschieden sein, denn da \mathcal{N} ein beschränktes P/T-Netz ist, existieren nur endlich viele erreichbare Markierungen, so dass sich mindestens eine Markierung wiederholen muss:

$$\mathbf{m} = \mathbf{m}_{k_i} = \mathbf{m}_{k_j} \quad \text{für} \quad i < j$$

Also ist

$$\mathbf{m}_{k_i} \xrightarrow{w_{k_i} \dots w_{k_j-1}} \mathbf{m}_{k_j}$$

Damit ist das Parikh-Bild $\psi(w_{ij})$ der Schaltfolge $w_{ij} := w_{k_i} \dots w_{k_j-1}$ nach Satz 7.39 eine T -Invariante. \square

Folgendes Theorem erlaubt es, für beschränkte Netze durch Invariantenanalyse Lebendigkeit zu verwerfen: Besitzt ein beschränktes Netz keine T -Invariante, dann kann es nicht lebendig sein.

Theorem 7.42 *Sei \mathcal{N} ein beschränktes und lebendiges P/T-Netz. Dann existiert eine echt positive T -Invariante j , d.h. $j(t) > 0$ für alle $t \in T$.*

Beweis: Da das Netz lebendig ist, existiert eine unendliche Schaltfolge w , in der alle Transitionen unendlich oft vorkommen. Mit Theorem 7.41 existiert dann eine echt positive T -Invariante.

Man kann dies auch direkt beweisen: Wenn \mathcal{N} lebendig ist, dann gibt es zur Initialmarkierung \mathbf{m}_0 eine Schaltfolge, in der alle Transitionen unendlich oft vorkommen. Da das Netz beschränkt ist, kommt in dieser Folge $\mathbf{m}_0 \xrightarrow{t_{i_1}} \mathbf{m}_1 \xrightarrow{t_{i_2}} \mathbf{m}_2 \cdots$ mindestens eine Markierung doppelt vor: $\mathbf{m}_i \xrightarrow{w} \mathbf{m}_j$ mit $\mathbf{m}_i = \mathbf{m}_j$. Hierbei können wir w so wählen, dass alle Transitionen vorkommen. Also ist nach Satz 7.39 $\psi(w)$ eine T -Invariante. \square

Die Umkehrung des Satzes ist praktisch. Sei \mathcal{N} ein beschränktes P/T-Netz, für das keine echt positive T -Invariante j existiert, dann ist \mathcal{N} auch nicht lebendig.

7.5.3 Fallen und Siphons

Wir betrachten im Folgenden nur einfache P/T-Netze, d.h. $W(x, y) \leq 1$.

Fallen Wir betrachten Stellenmengen $A \subseteq P$ mit der Eigenschaft, dass alle Transition t , die aus A Marken entfernen, auch wieder Marken in A generieren.

Definition 7.43 Sei $\mathcal{N} = (P, T, F, W, \mathbf{m}_0)$ ein einfaches P/T-Netz.

Eine Stellenmenge $A \subseteq P$ heißt Falle (engl. trap), wenn jede Transition t , die aus A Marken entfernt, auch wieder Marken in A generiert:

$$\forall t \in T : t \in A^\bullet \Rightarrow t \in {}^\bullet A \quad \text{bzw. kurz} \quad A^\bullet \subseteq {}^\bullet A$$

Eine Falle $A \subseteq P$ heißt in \mathbf{m} markiert, wenn mindestens eine Stelle in A markiert ist:

$$\sum_{p \in A} \mathbf{m}(p) > 0$$

Da jede Transition t , die Marken aus der Falle entfernt, auch wieder welche in ihr erzeugt, ist es nicht möglich, eine markierte Falle komplett zu leeren. Es ist also mindestens eine Marke in der Falle gefangen.

Lemma 7.44 Sei A eine Falle eines einfachen P/T-Netzes \mathcal{N} . Ist die Falle A in einer Markierung \mathbf{m}_0 markiert, dann bleibt sie dies auch in allen von \mathbf{m}_0 aus erreichbaren Markierungen.

$$\forall \mathbf{m}_0 : \sum_{p \in A} \mathbf{m}_0(p) > 0 \Rightarrow \forall \mathbf{m} \in R(\mathcal{N}, \mathbf{m}_0) : \sum_{p \in A} \mathbf{m}(p) > 0$$

Beweis: Als Übung. \square

Siphons/Co-Fallen Die Umkehrung des Konzeptes *Falle* existiert auch. Es wird als *Siphon* (engl. Abfluss) oder auch als *co-Falle* bezeichnet.

Definition 7.45 Sei $\mathcal{N} = (P, T, F, W, \mathbf{m}_0)$ ein einfaches P/T -Netz.

Eine Stellenmenge $A \subseteq P$ heißt Siphon, wenn jede Transition t , die Marken in A generiert, dazu welche aus A benötigt:

$$\forall t \in T : t \in \bullet A \Rightarrow t \in A^\bullet \quad \text{bzw. kurz} \quad \bullet A \subseteq A^\bullet$$

Ein Siphon $A \subseteq P$ heißt markiert, wenn mindestens eine Stelle in A markiert ist.

$$\sum_{p \in A} \mathbf{m}(p) > 0$$

Bemerkung: Nach Definition ist die leere Menge sowohl ein Siphon als auch eine Falle. Ebenso ist die Stellenmenge P selbst sowohl ein Siphon als auch eine Falle.

Da ein Siphon nur durch Transition t gefüllt werden, die zum Schalten Marken aus dem Siphon benötigen, muss ein unmarkierter Siphon dies auch in allen erreichbaren Markierungen bleiben.

Lemma 7.46 Sei A ein Siphon eines einfachen P/T -Netzes \mathcal{N} . Ist der Siphon A in einer Markierung \mathbf{m}_0 unmarkiert, dann bleibt er dies auch in allen von \mathbf{m}_0 aus erreichbaren Markierungen.

$$\forall \mathbf{m}_0 : \sum_{p \in A} \mathbf{m}_0(p) = 0 \Rightarrow \forall \mathbf{m} \in R(\mathcal{N}, \mathbf{m}_0) : \sum_{p \in A} \mathbf{m}(p) = 0$$

Beweis: Als Übung. □

Aus strukturellen Eigenschaften lassen sich dynamische folgern. Mit dem folgenden Lemma können wir insbesondere die nachweisen, dass ein Netz nicht lebendig sein kann.

Lemma 7.47 Sei \mathcal{N} ein einfaches P/T -Netz ohne isolierte Plätze. Wenn \mathcal{N} ein lebendiges Netz ist, dann muss jeder nichtleere Siphon A in der Initialmarkierung markiert sein.

Beweis: Wäre der Siphon A in der Initialmarkierung unmarkiert, dann wären alle Transitionen aus A^\bullet tot und es gilt $A^\bullet \neq \emptyset$, da \mathcal{N} keine isolierte Plätze besitzt. Also existiert mindestens eine tote Transition. Widerspruch. □

Siphon/Trap-Eigenschaft

Wir können beide Eigenschaften – Falle und Co-Falle – sinnvoll kombinieren: Eine markierte Falle A wird nie leer. Ist A Teilmenge eines Siphons B , dann kann auch B nie leer werden.

Definition 7.48 *Ein einfaches P/T-Netz \mathcal{N} hat die Siphon/Trap-Eigenschaft, wenn folgendes gilt: Jeder Siphon B von \mathcal{N} enthält eine anfangsmarkierte Falle A als Teilmenge.*

Angenommen \mathcal{N} besitzt die Siphon/Trap-Eigenschaft, dann aktiviert jede erreichbare Markierung mindestens eine Transition.

Satz 7.49 *Wenn \mathcal{N} die Siphon/Trap-Eigenschaft besitzt, dann aktiviert jede erreichbare Markierung von \mathcal{N} mindestens eine Transition: \mathcal{N} ist verklemmungsfrei.*

Beweis: Wir zeigen zunächst: Wenn \mathbf{m} eine Verklemmung ist, dann ist die Menge $B := \{p \mid \mathbf{m}(p) = 0\}$ ein Siphon.

Sei $t \in {}^\bullet B$, dann muss es mindestens einen unmarkierten Platz $p \in {}^\bullet t$ geben, der daher auch in B liegt, denn andernfalls wäre t ja aktiviert (wegen $W(x, y) \leq 1$) und \mathbf{m} wäre keine Verklemmung. Also gilt $t \in p^\bullet \subseteq B^\bullet$, also gilt ${}^\bullet B \subseteq B^\bullet$, d.h. B ist ein Siphon.

Angenommen \mathcal{N} wäre nicht verklemmungsfrei. Da der Siphon $B := \{p \mid \mathbf{m}(p) = 0\}$ dann unmarkiert ist, enthält er erst recht keine anfangsmarkierte Falle als Teilmenge. Also besitzt \mathcal{N} nicht die Siphon/Trap-Eigenschaft. Widerspruch. \square