

Prof. Dr. Jürgen Dassow
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik

P E T R I — N E T Z E

Vorlesungsskript

Magdeburg, Oktober 2010 – Januar 2011

Inhaltsverzeichnis

Vorwort	1
1 Einführende Beispiele und Bemerkungen	5
2 Netzgraphen	11
3 Petri-Netze und ihr Verhalten	21
3.1 Grundlegende Definitionen	21
3.2 Beschränktheit und Erreichbarkeit	31
3.3 Lebendigkeit	38
3.4 Reduktionen	45
3.5 Invarianten	53
3.6 Fairness und Synchronie	61
3.7 Deadlocks und Fallen	76
4 Petri-Netze und formale Sprachen	85
4.1 Einiges aus der Theorie formaler Sprachen	85
4.2 Petri-Netz-Sprachen	86
4.3 Petri-Netze und Sprachen mit Auswahlkontext	96
Literaturverzeichnis	100

Kapitel 4

Petri-Netze und formale Sprachen

In diesem Kapitel werden wir auf der einen Seite die Mengen von Schaltfolgen eines Petri-Netzes aus der Sicht der Theorie formaler Sprachen untersuchen, und auf der anderen Seite wollen wir mit Hilfe von Petri-Netzen ein Problem aus der Theorie formaler Grammatiken mit gesteuerten Ableitungen lösen. Zuerst wollen wir aber einige Notationen und Fakten aus der Theorie formaler Sprachen geben, die wir im Folgenden benutzen werden.

4.1 Einiges aus der Theorie formaler Sprachen

Hinsichtlich der Theorie der formalen Sprachen gehen wir davon aus, dass die Leserin / der Leser über die Grundkenntnisse aus einer Vorlesung zur Theoretischen Informatik verfügt (siehe z. B. [12]). Insbesondere erwarten wir die Kenntnis der Chomsky-Normalform für kontextfreie Grammatiken, der Äquivalenz der regulären Sprachen und der von endlichen Automaten akzeptierten Sprachen und der Schleifensätze (Pumpinglemma) für reguläre und kontextfreie Sprachen. Wir geben hier nur kurz einige Bezeichnungen und einige Resultate an, die üblicherweise nicht in den Grundvorlesungen zur theoretischen Informatik vorkommen.

Ein endlicher Automat wird als Quintupel $\mathcal{A} = (X, Z, z_0, F, \delta)$ spezifiziert, wobei X die Menge der Eingabesymbole, Z die Menge der Zustände, $z_0 \in Z$ der Anfangszustand, $F \subseteq Z$ die Menge der akzeptierenden Zustände und $\delta : Z \times X \rightarrow Z$ die Überföhrungsfunktion sind.

Eine Regelgrammatik wird als Quadrupel $G = (N, T, P, S)$ spezifiziert, wobei N die Menge der Nichtterminale, T die Menge der Terminale, P die Menge der Regeln und S das Startsymbol oder Axiom sind. Wir bezeichnen die Mengen der regulären, kontextfreien und kontextabhängigen Sprachen mit $\mathcal{L}(REG)$, $\mathcal{L}(CF)$ und $\mathcal{L}(CS)$.

Lemma 4.1 *i) $R_{a,b} = \{a^n b^m \mid n \geq m \geq 0\} \notin \mathcal{L}(REG)$.*
ii) $R_{a,b,c} = \{a^r b^s c^t \mid r \geq s \geq t \geq 0\} \notin \mathcal{L}(CF)$.

Beweis. i) Wir nehmen an, dass $R_{a,b}$ regulär ist. Dann gibt es einen endlichen Automaten $\mathcal{A} = (\{a, b\}, Z, z_0, F, \delta)$, der $R_{a,b}$ akzeptiert. Die Anzahl der Zustände von Z sei k . Wir betrachten das Wort $a^{k+1} b^{k+1}$. Wegen $a^{k+1} b^{k+1} \in R_{a,b}$ ist $\delta(z_0, a^{k+1} b^{k+1}) \in F$. Wir betrachten die Zustände $z_q = \delta(z_0, a^{k+1} b^q)$, $1 \leq q \leq k+1$. Dies sind $k+1$ Zustände. Da es aber nur k Zustände in Z gibt, existieren Zahlen i und j mit $1 \leq i < j \leq k+1$ derart, dass

$z_i = z_j$ gilt. Damit erhalten wir $\delta(z_i, b^{j-i}) = z_i$ und daraus zuerst $\delta(z_i, b^{n(j-i)}) = z_i$ für alle $n \geq 2$ und dann $\delta(z_0, a^{k+1}b^i b^{n(j-i)} b^{k+1-j}) \in F$. Damit wird $a^{k+1}b^{k+1+(n-1)(j-i)}$ akzeptiert. Also haben wir $a^{k+1}b^{k+1+(n-1)(j-i)} \in R_{a,b}$. Da aber $(n-1)(j-i) > 0$ ist, ist wegen der Struktur der Wörter in $R_{a,b}$ auch $a^{k+1}b^{k+1+(n-1)(j-i)} \notin R_{a,b}$. Der dadurch erhaltene Widerspruch zeigt, dass unsere Annahme falsch, womit die Aussage i) des Lemmas bewiesen ist.

ii) Wir verzichten hier auf einen vollständigen Beweis. Wir nehmen an, dass $R_{a,b,c}$ eine kontextfreie Sprache ist und $G = (N, T, P, S)$ eine kontextfreie Grammatik in Chomsky-Normalform ist, die $R_{a,b,c}$ erzeugt. Die Grammatik G habe k Nichtterminale. Wir betrachten das in $R_{a,b,c}$ liegende Wort $w = a^{2^{k+2}}b^{2^{k+2}}c^{2^{k+2}}$. Dann enthält der Ableitungsbaum für w einen Teilbaum der Tiefe $k+1$, dessen Blätter alle c sind. In diesem Baum gibt es einen Pfad, der ein Nichtterminal doppelt enthält. Mittels dieses Nichtterminals können ein Pumping von zwei Teilwörtern c^q und c^p mit $p+q > 0$ erreichen. Hierdurch entsteht ein Wort mit mehr Vorkommen von c als Vorkommen von a . Folglich wird ein Wort abgeleitet, das nicht in $R_{a,b,c}$ liegt, womit wir einen Widerspruch hergeleitet haben. \square

Es sei $G = (N, T, P, S)$ eine Regelgrammatik. Für eine Ableitung

$$D : S \Longrightarrow w_1 \Longrightarrow w_2 \Longrightarrow \dots \Longrightarrow w_r = w$$

von $w \in T^*$ in G , definieren den *Arbeitsplatz von w bei D* als

$$Ws_G(w, D) = \max\{|w_i| \mid 1 \leq i \leq r\}$$

und den *Arbeitsplatz von w* als

$$Ws_G(w) = \min\{Ws_G(w, D) \mid D \text{ is a derivation of } w \text{ in } G\}.$$

Ohne Beweis geben wir die folgende Aussage an; ein Beweis ist in [13] zu finden.

Satz 4.2 *Wenn es für eine Regelgrammatik $G = (N, T, P, S)$ eine Konstante $k \in \mathbf{N}$ derart gibt, dass $Ws_G(w) \leq k|w|$ für alle $w \in L(G)$ gilt, so ist $L(G)$ eine kontextabhängige Sprache.* \square

4.2 Petri-Netz-Sprachen

Das Verhalten von Petri-Netzen wird in entscheidendem Maße von den im Netz realisierbaren Schaltfolgen bestimmt. Wir haben bereits im Abschnitt 3.1 für jedes Petri-Netz $N = (S, T, F, V, m_0)$ die Menge

$$L(N, m_0) = \{q \mid m_0[q > m \text{ für eine Markierung } m\}$$

definiert. Offensichtlich ist $L(N, m_0)$ eine Sprache über dem Alphabet T . Wir nennen $L(N, m_0)$ die Sprache des Petri-Netzes. Da mit N die Anfangsmarkierung m_0 festgelegt ist, schreiben wir nur einfach $L(N)$ anstelle von $L(N, m_0)$.

Petri-Netz-Sprachen haben die folgende einfache Eigenschaft, die oft den Nachweis gestattet, dass eine gegebene Menge keine Petri-Netz-Sprache ist.

Lemma 4.3 i) Es seien $N = (S, T, F, V, m_0)$ ein Petri-Netz und q ein Wort aus $L(N)$. Dann ist auch jedes Anfangsstück q' von q ein Wort aus $L(N)$. Insbesondere gilt $\lambda \in L(N)$.
 ii) Für jedes Petri-Netz $N = (S, T, F, V, m_0)$ gilt $L(N) = \text{Pref}(L(N))$.¹

Beweis. i) folgt aus der Tatsache, dass mit q auch jedes Anfangsstück von q eine Schaltfolge ist. ii) ergibt sich direkt aus i). □

Die folgende Aussage ergibt sich sofort aus der Definition der Lebendigkeit von Petri-Netzen.

Lemma 4.4 Es sei $N = (S, T, F, V, m_0)$ ein Petri-Netz. Dann ist N genau dann lebendig, wenn $L(N)$ die folgende Eigenschaft (*) besitzt.

(*) Für jedes $q \in L(N)$ und jedes $t \in T$ gibt es ein $r \in T^*$ mit $qrt \in L(N)$. □

Folgerung 4.5 Die Sprache eines lebendigen Petri-Netzes ist unendlich. □

Wir definieren nun einige Mengen von Petri-Netz-Sprachen:

$$\begin{aligned} \mathcal{L}(P) &= \{L(N) \mid N \text{ ist ein Petri-Netz}\}, \\ \mathcal{L}(PB) &= \{L(N) \mid N \text{ ist ein beschränktes Petri-Netz}\}, \\ \mathcal{L}(PL) &= \{L(N) \mid N \text{ ist ein lebendiges Petri-Netz}\}. \end{aligned}$$

Wir vergleichen zuerst die gerade definierten Familien von Petri-Netz-Sprachen untereinander und mit den Familien der Chomsky-Hierarchie.

Satz 4.6 Es gilt das folgende Diagramm aus Abbildung 4.1, das wie folgt zu interpretieren ist: $A \rightarrow B$ steht für $A \subset B$; sind zwei Familien nicht durch einen gerichteten Weg verbinden, so sind sie unvergleichbar.

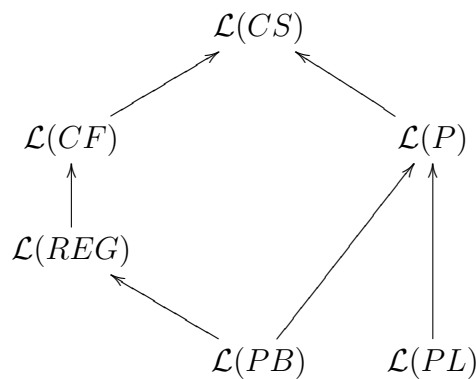


Abbildung 4.1: Hierarchie der Petri-Netz-Sprachen

Beweis. Die Inklusionen $\mathcal{L}(REG) \subset \mathcal{L}(CF) \subset \mathcal{L}(CS)$ und ihre Echtheit sind aus der Vorlesung Theoretische Informatik bekannt (siehe z. B. [12]). Da beschränkte und lebendige Petri-Netze spezielle Petri-Netze sind, folgt aus den Definitionen von $\mathcal{L}(PB)$, $\mathcal{L}(PL)$ und $\mathcal{L}(P)$ sofort $\mathcal{L}(PB) \subseteq \mathcal{L}(P)$ und $\mathcal{L}(PL) \subseteq \mathcal{L}(P)$.

¹Für eine Sprache $L \subseteq V^*$ ist $\text{Pref}(L)$ durch $\text{Pref}(L) = \{u \mid uv \in L \text{ für ein } v \in V^*\}$ definiert.

i) Es gibt eine reguläre Sprache R , die nicht in $\mathcal{L}(P)$ liegt.

Offensichtlich ist die endliche Sprache $R = \{t^5\}$ über dem einelementigen Alphabet $\{t\}$ regulär. Nach Lemma 4.3 i) ist R aber keine Petri-Netz-Sprache. Also gilt $R \notin \mathcal{L}(P)$.

ii) $\mathcal{L}(PB) \subseteq \mathcal{L}(REG)$.

Es sei $L \in \mathcal{L}(PB)$. Dann gilt $L = L(N)$ für ein beschränktes Petri-Netz $N = (S, T, F, V, m_0)$. Dann ist die Menge $R(N, m_0)$ der erreichbaren Markierungen endlich. Wir konstruieren nun den endlichen Automaten

$$\mathcal{A} = (T, R(N, m_0), m_0, R(N, m_0), \delta),$$

wobei δ durch $\delta(m, t) = m'$ mit $m[t > m'$ definiert ist. Da jede erreichbare Markierung zu den akzeptierenden Zuständen gehört, ist jede Schaltfolge in der von \mathcal{A} akzeptierten Menge. Folglich ist $T(\mathcal{A}) = L$ und damit L regulär.

Da jede Sprache aus $\mathcal{L}(PB)$ regulär ist, gilt $\mathcal{L}(PB) \subseteq \mathcal{L}(REG)$.

Die Sprache R aus i) erfüllt offenbar $R \notin \mathcal{L}(PB)$. Folglich ist die Inklusion $\mathcal{L}(PB) \subseteq \mathcal{L}(REG)$ sogar echt.

iii) Es gibt eine Sprache in $\mathcal{L}(PL)$, die nicht kontextfrei ist.

Wir betrachten das gewöhnliche Netz

$$N = (\{s_1, s_2\}, \{t_1, t_2, t_3\}, \{(t_1, s_1), (s_1, t_2), (t_2, s_2), (s_2, t_3)\}, V, (0, 0)).$$

Da in N zu jedem Zeitpunkt die Folge $t_1 t_2 t_3$ geschaltet werden kann, ist N lebendig. Die Petri-Netz-Sprache von N ist

$$L(N) = \{u \mid \text{für jedes Wort } v \in Pref(u) \text{ gilt } \#_{t_1}(v) \geq \#_{t_2}(v) \geq \#_{t_3}(v)\}.$$

Wir zeigen nun, dass $L(N)$ nicht kontextfrei ist. Angenommen, $L(N)$ ist kontextfrei. Dann ist auch $L(N) \cap \{t_1\}^* \{t_2\}^* \{t_3\}^*$ als Durchschnitt einer kontextfreien Sprache mit einer regulären Sprache kontextfrei. Wegen

$$L(N) \cap \{t_1\}^* \{t_2\}^* \{t_3\}^* = \{t_1^r t_2^s t_3^t \mid r \geq s \geq t \geq 0\}$$

erhalten wir einen Widerspruch zu Lemma 4.1 ii).

iv) Aus den Inklusionen $\mathcal{L}(REG) \subset \mathcal{L}(CF)$ und $\mathcal{L}(PL) \subseteq \mathcal{L}(P)$ und den Aussagen in i) und iii) folgt sofort die Unvergleichbarkeit von $\mathcal{L}(REG)$ und $\mathcal{L}(CF)$ einerseits und $\mathcal{L}(PL)$ und $\mathcal{L}(P)$ andererseits.

v) $\mathcal{L}(PL)$ und $\mathcal{L}(PB)$ sind unvergleichbar.

Die Sprache $L(N)$ aus iii) liegt in $\mathcal{L}(PL)$, aber ist nicht kontextfrei. Wegen $\mathcal{L}(PB) \subset \mathcal{L}(REG) \subset \mathcal{L}(CF)$ ist auch $L(N) \notin \mathcal{L}(PB)$.

Andererseits ist für das gewöhnliche Netz $N' = (\{s\}, \{t\}, \{(s, t)\}, V, (1))$ die Erreichbarkeitsmenge $R(N', (1))$ durch $\{(1), (0)\}$ gegeben. Somit ist N' ein beschränktes Petri-Netz. Daher gilt $L(N') = \{\lambda, t\} \in \mathcal{L}(PB)$. Wegen Folgerung 4.5 gilt noch $L(N') \notin \mathcal{L}(PL)$.

vi) $\mathcal{L}(PL) \subset \mathcal{L}(P)$.

Wir haben schon festgestellt, dass $\mathcal{L}(PL) \subseteq \mathcal{L}(P)$ gilt. Die Sprache $L(N')$ aus v) liegt in $\mathcal{L}(PB)$ und folglich in $\mathcal{L}(P)$, aber nach i) nicht in $\mathcal{L}(PL)$. Damit gilt $\mathcal{L}(PL) \subset \mathcal{L}(P)$.

vii) $\mathcal{L}(P) \subset \mathcal{L}(CS)$.

Es sei $N = (S, T, F, V, m_0)$ ein Petri-Netz. Dabei sei $S = \{s_1, s_2, \dots, s_n\}$. Mit jeder Transition t und jeder Stelle s assoziieren wir die Mengen

$$\begin{aligned} M_t &= \{X_t, Y_t, Z_t\}, \\ M_{s,t} &= \{A_{t,s,1}, A_{t,s,2}, \dots, A_{t,s,t^-(s)+1}\}, \\ N_{s,t} &= \{B_{t,s,1}, B_{t,s,2}, \dots, B_{t,s,n(s,t)}\}, \end{aligned}$$

wobei $n(s, t)$ durch

$$n(s, t) = \begin{cases} -\Delta(t)(s) + 1 & \text{für } \Delta(t)(s) < 0 \\ 1 & \text{für } \Delta(t)(s) \geq 1 \end{cases}$$

definiert ist.

Wir konstruieren nun die Regelgrammatik $G = (N, T, P, S)$, wobei

$$N = \{\$, S, X, Y, Y', Z\} \cup \{A_i \mid 1 \leq i \leq n\} \cup \{A'_i \mid 1 \leq i \leq n\} \cup \bigcup_{t \in T} M_t \cup \bigcup_{\substack{t \in T \\ s \in S}} M_{s,t} \cup \bigcup_{\substack{t \in T \\ s \in S}} N_{s,t}$$

gilt und P aus allen Regeln der folgenden Art besteht:

$$S \rightarrow \$YXA_1^{m_0(s_1)}XA_2^{m_0(s_2)} \dots XA_n^{m_0(s_n)}X\$$$

(durch diese Regel wird eine Beschreibung der Anfangsmarkierung

$$m_0 = (m_0(s_1), m_0(s_2), \dots, m_0(s_n))$$

erzeugt),

$$\begin{aligned} \$YX &\rightarrow \$XA_{t,s_1,1} \quad \text{für } t \in T, \\ A_{t,s_i,r}A_i &\rightarrow A'_iA_{t,s_i,r+1} \quad \text{für } t \in T, 1 \leq i \leq n, 1 \leq r \leq t^-(s_i), \\ A_{t,s_i,t^-(s_i)+1}A_i &\rightarrow A_iA_{t,s_i,t^-(s_i)+1} \quad \text{für } t \in T, 1 \leq i \leq n, \\ A_{t,s_i,t^-(s_i)+1}X &\rightarrow XA_{t,s_{i+1},1} \quad \text{für } t \in T, 1 \leq i \leq n-1, \\ A_{t,s_n,t^-(s_n)+1}X &\rightarrow X_tX \quad \text{für } t \in T \end{aligned}$$

(die Symbole $A_{t,s,r}$ wandern von links nach rechts über das Wort und markieren dabei durch Übergang zur gestrichelten Variante jeweils soviel Symbole A_i , wie $t^-(s)$ angibt; sind nicht genügend A_i vorhanden, so wird die Ableitung blockiert, wenn $A_{t,s,r}$ mit $r \leq t^-(s)$ direkt vor einem X steht; sind stets ausreichend viele Symbole A_i vorhanden, so wird X_t eingeführt),

$$\begin{aligned} A_iX_t &\rightarrow X_tA_i \quad \text{für } t \in T, 1 \leq i \leq n, \\ A'_iX_t &\rightarrow X_tA_i \quad \text{für } t \in T, 1 \leq i \leq n, \\ XX_t &\rightarrow X_tX \quad \text{für } t \in T \end{aligned}$$

(das Symbol X_t bewegt sich von rechts nach links über und überführt dabei alle A'_i wieder in A_i),

$$\begin{aligned} \$X_t X &\rightarrow \$X B_1^{\Delta(t)(s_1)} A_{t,s_1,1} \quad \text{für } t \in T, \Delta(t)(s_1) \geq 0, \\ B_{t,s_i,r} A_i &\rightarrow B_{t,s_i,r+1} \quad \text{für } t \in T, 1 \leq i \leq n, 1 \leq r \leq n(t,s), \\ B_{t,s_i,n(t,s)} A_i &\rightarrow A_i B_{t,s_i,n(t,s)} \quad \text{für } t \in T, 1 \leq i \leq n, \\ B_{t,s_i,n(s,t)} X &\rightarrow X A_{i+1}^{\Delta(t)(s_{i+1})} B_{t,s_{i+1},1} \quad \text{für } t \in T, 1 \leq i \leq n-1, \Delta(t)(s_{i+1}) \geq 0, \\ B_{t,s_n,n(s,t)} X &\rightarrow X Y_t \quad \text{für } t \in T \end{aligned}$$

(wenn $\Delta(t)(s_i) < 0$ gilt, so werden $-\Delta(t)(s_i)$ Symbole A_i gestrichen; ist $\Delta(t)(s_i) \geq 0$, so werden $\Delta(t)(s_i)$ Symbole A_i zusätzlich eingeführt; folglich wird die Anzahl der A_i um $\Delta(t)(s_i)$ geändert; nach dieser Änderung wird Y_t erzeugt),

$$\begin{aligned} Y_t t' &\rightarrow t' Y_t \quad \text{für } t, t' \in T, \\ Y_t \$ &\rightarrow Z t \$ \quad \text{für } t \in T \end{aligned}$$

(das Symbol Y_t bewegt sich weiter nach rechts und fügt vor der Endmarkierung $\$$ ein t ein und wird zu Z geändert),

$$\begin{aligned} t Z &\rightarrow Z t \quad \text{für } t \in T, \\ A_i Z &\rightarrow Z A_i \quad \text{für } 1 \leq i \leq n, \\ X Z &\rightarrow Z X, \$ Z \rightarrow \$ Y \end{aligned}$$

(das Symbol Z wandert nach links und wird vor der Anfangsmarkierung in Y geändert; insgesamt wird damit ein Wort $\$Y X A_1^{m_1} X A_{m_2} X \dots X A_n^{m_n} X q \$$ mit $q \in T^*$ ins Wort $\$Y X A_1^{m'_1} X A_{m'_2} X \dots X A_n^{m'_n} X q t \$$ überführt, wobei die Markierung $m = (m_1, m_2, \dots, m_n)$ durch Schalten von t in die Markierung $m' = (m'_1, m'_2, \dots, m'_n)$ übergeht),

$$\begin{aligned} \$Y X &\rightarrow Y', Y' X \rightarrow Y', Y' \$ \rightarrow \lambda, \\ Y' A_i &\rightarrow Y' \quad \text{für } 1 \leq i \leq n, \\ Y' t &\rightarrow t Y' \quad \text{für } t \in T \end{aligned}$$

(das Symbol Y' wird erzeugt, bewegt sich von links nach rechts und löscht dabei alle Symbole, die nicht in T liegen).

Aus den Erklärungen zu den Regeln folgt, dass

$$\$Y X A_1^{m_0(s_1)} X A_2^{m_0(s_2)} \dots X A_n^{m_0(s_n)} X \$ \Longrightarrow^* \$Y X A_1^{u_1} X A_{u_2} X \dots X A_n^{u_n} X p \$ \Longrightarrow p$$

genau dann gilt, wenn $m_0[p > (u_1, u_2, \dots, u_n)$ gültig ist. Hieraus ergibt sich $L(G) = L(N)$.

Wenn $k = \max\{\Delta(t)(s) \mid t \in T, s \in S\}$ ist, so wird bei jedem Hinzufügen einer Transition der vordere Teil aus Nichtterminalen höchstens um $n \times k$ Symbole verlängert. Daher hat jede Satzform einer Ableitung von p höchstens die Länge $kn|p| + n + 4 + |p|p$. Aus Satz 4.2 folgt nun, dass $L(G)$ und damit auch $L(N)$ kontextabhängig ist.

Damit ist $\mathcal{L}(P) \subseteq \mathcal{L}(CS)$. Da die Sprache R aus i) in $\mathcal{L}(CS)$ aber nicht in $\mathcal{L}(P)$ liegt, ist die Inklusion sogar echt. \square

Wir geben nun Abschlusseigenschaften bezüglich der in der Theorie formaler Sprachen üblichen mengentheoretischen Operationen Vereinigung, Durchschnitt und Durchschnitt mit regulären Mengen und der algebraisch motivierten Operationen Konkatenation, Homomorphismen und inversen Homomorphismen.

Satz 4.7 *Die Mengen $\mathcal{L}(P)$, $\mathcal{L}(PB)$ und $\mathcal{L}(PL)$ sind nicht abgeschlossen unter Vereinigung, Konkatenation, Homomorphismen und Durchschnitten mit regulären Mengen.*

Beweis. i) Vereinigung

Wir betrachten die gewöhnlichen Netze

$$N_1 = (\{s_1\}, \{t_1\}, \{(s_1, t_1), (t_1, s_1)\}, V, (1)) \text{ und } N_2 = (\{s_2\}, \{t_2\}, \{(s_2, t_2), (t_2, s_2)\}, V, (1)).$$

Offensichtlich sind die Netze beschränkt und lebendig. Folglich sind die zugehörigen Petri-Netz-Sprachen

$$L(N_1) = \{t_1^n \mid 0 \leq n\} \text{ und } L(N_2) = \{t_2^n \mid 0 \leq n\}$$

sowohl in $\mathcal{L}(P)$ als auch in $\mathcal{L}(PB)$ und $\mathcal{L}(PL)$.

Wir beweisen nun, dass

$$L_1 = L(N_1) \cup L(N_2) = \{t_1^n \mid 0 \leq n\} \cup \{t_2^n \mid 0 \leq n\}$$

nicht in $\mathcal{L}(P)$ liegt.

Wir nehmen an, dass es ein Petri-Netz $N = (S, t_1, t_2, F, V, m_0)$ mit $L(N) = L_1$ gibt. Wir merken zuerst an, dass $\Delta(t_1) \geq 0$ gilt. Dies folgt daraus, dass im anderen Fall eine Stelle s mit $\Delta(t_1)(s) < 0$ existiert, für die nach k -maligem Schalten von t_1 dann $m_0(s) + k \cdot \Delta(t_1)(s)$ Marken auf s liegen und für hinreichend großes k wird dieser Wert kleiner als $t_1^-(s)$, d.h. t_1 wäre nicht mehr aktiviert; aber aus der Struktur von $L(N)$ folgt, dass t_1 nach beliebig oftmaligen Schalten von t_1 immer aktiviert ist. Es sei m_1 durch $m_0[t_1] > m_1$ definiert. Wegen $m_1 = m_0 + \Delta(t_1) \geq m_0$ und der Schaltbarkeit von t_2 bei m_0 ist t_2 auch bei m_1 aktiviert. Somit ist $t_1 t_2$ eine Schaltfolge in N und damit in $L(N)$. Andererseits ist $t_1 t_2$ nicht in L_1 . Dies widerspricht $L(N) = L_1$.

ii) Konkatenation

Wir betrachten erneut die Netze N_1 und N_2 aus i) und die Sprache

$$L_2 = L(N_1) \cdot L(N_2) = \{t_1^n t_2^m \mid n \geq 0, m \geq 0\}.$$

Wir nehmen erneut an, dass es ein Petri-Netz $N' = (S', \{t_1, t_2\}, F', V', m'_0)$ mit $L(N') = L_2$ gibt. Wie in Teil i) kann gezeigt werden, dass $\Delta(t_1) \geq 0$ und $\Delta(t_2) \geq 0$ gelten. Es sei m_1 die Markierung nach Schalten von t_1 . Wegen $m_1 = m'_0 + \Delta(t_1) \geq m'_0$ ist t_2 bei m_1 aktiviert. Die Markierung m_2 entstehe aus m_1 durch Schalten von t_2 . Dann gilt auch $m_2 = m_1 + \Delta(t_2) \geq m_1 \geq m'_0$, womit t_1 bei m_2 aktiviert ist. Damit ist $t_1 t_2 t_1$ eine Schaltfolge in N' , aber $t_1 t_2 t_1$ liegt nicht in L_2 . Der damit erhaltene Widerspruch zeigt, dass L_2 keine Petri-Netz-Sprache ist.

iii) Durchschnitt mit regulären Mengen

Wir betrachten wieder das Netz N_1 aus i) und betrachten die Sprache

$$R = L(N_1) \cap \{t_1^5\} = \{t_1^5\},$$

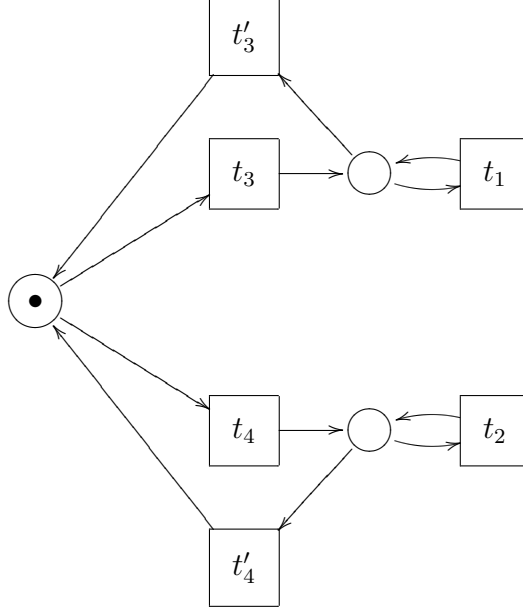


Abbildung 4.2: Netz K

die als Durchschnitt einer Petri-Netz-Sprache mit einer regulären Menge entsteht. Im Teil i) des Beweises von Satz 4.6 haben wir bereits nachgewiesen, dass R keine Petri-Netz-Sprache ist.

iv) Homomorphismen

Wir betrachten das Netz K aus Abbildung 4.2. Es ist leicht zu sehen, dass K beschränkt ist (zu jedem Zeitpunkt ist genau eine Marke im Netz) und die zu K gehörige Petri-Netz-Sprache

$$L(K) = \{q_1 q_2 \dots q_k \mid k \geq 0, q_i \in \{t_3 t_1^n t'_3 \mid n \geq 0\} \cup \{t_4 t_2^n t'_4 \mid n \geq 0\} \text{ für } 1 \leq i \leq k-1, \\ q_k \in \text{Pref}(\{t_3 t_1^n t'_3 \mid n \geq 0\} \cup \{t_4 t_2^n t'_4 \mid n \geq 0\})\}$$

ist. Aufgrund der Struktur von $L(K)$ ist auch sofort zu sehen, dass K lebendig ist (vgl. Lemma 4.4).

Wir definieren den Homomorphismus h durch

$$h(t_1) = t_1, h(t_2) = t_2, h(t_3) = h(t_4) = \lambda, h(t'_3) = h(t'_4) = t.$$

Dann ergibt sich

$$h(L(K)) = \{t_{i_1}^{n_1} t t_{i_2}^{n_2} t \dots t_{i_k}^{n_k} t \mid k \geq 1, i_j \in \{1, 2\} \text{ für } 1 \leq j \leq k, n_j \geq 0 \text{ für } 1 \leq j \leq k\} \\ \cup \{t_{i_1}^{n_1} t t_{i_2}^{n_2} t \dots t_{i_k}^{n_k} \mid k \geq 1, i_j \in \{1, 2\} \text{ für } 1 \leq j \leq k, n_j \geq 0 \text{ für } 1 \leq j \leq k\}.$$

In $h(L(K))$ steht zwischen zwei verschiedenen t_i und t_j immer ein t . Wir nehmen jetzt wieder an, dass es ein Netz N'' mit $L(N'') = h(L(K))$ gibt. Analog zum Teil i) dieses Beweises können wir zuerst $\Delta(t_1) \geq 0$ und $\Delta(t_2) \geq 0$ für N'' zeigen und dann nachweisen, dass $t_1 t t_2 t_1$ eine Schaltfolge in N'' ist. Wegen $t_1 t t_2 t_1 \notin h(L(K))$ haben wir den gewünschten Widerspruch erhalten. \square

Satz 4.8 Die Menge $\mathcal{L}(P)$ ist abgeschlossen unter Durchschnitten und inversen Homomorphismen.

Beweis. i) Durchschnitt

Es seien L_1 und L_2 zwei Sprachen aus $\mathcal{L}(P)$. Ferner seien $N_1 = (S_1, T_1, F_1, V_1, m_1)$ und $N_2 = (S_2, T_2, F_2, V_2, m_2)$ zwei Petri-Netze mit $L(N_1) = L_1$ und $L(N_2) = L_2$. Ohne Beschränkung der Allgemeinheit können wir annehmen, dass $S_1 \cap S_2 = \emptyset$ gilt.

Der Durchschnitt $L_1 \cap L_2$ ist offenbar eine Sprache über $T_1 \cap T_2$, da eine Transition aus $T_1 \setminus T_2$ bzw. aus $T_2 \setminus T_1$ nicht in L_2 bzw. L_1 vorkommen kann. Folglich können wir die Transitionen aus $(T_1 \setminus T_2) \cup (T_2 \setminus T_1)$ streichen.

Wir konstruieren nun das Petri-Netz

$$N = (S_1 \cup S_2, T_1 \cap T_2, F'_1 \cup F'_2, V, m_0),$$

wobei folgende Beziehungen bestehen:

- $F'_1 = (F_1 \cap (S_1 \times (T_1 \cap T_2))) \cup (F_1 \cap ((T_1 \cap T_2) \times S_1)),$
 $F'_2 = (F_2 \cap (S_2 \times (T_1 \cap T_2))) \cup (F_2 \cap ((T_1 \cap T_2) \times S_2)),$
- $V(s, t) = V_1(s, t)$ und $V(t, s) = V_1(t, s)$ für $s \in S_1,$
 $V(s, t) = V_2(s, t)$ und $V(t, s) = V_2(t, s)$ für $s \in S_2,$
- $m(s) = m_1(s)$ für $s \in S_1$ und
 $m(s) = m_2(s)$ für $s \in S_2.$

Entsprechend der Konstruktion ist eine Transition $t \in T_1 \cap T_2$ in N genau dann aktiviert, wenn sie sowohl im N_1 entsprechenden Teilnetz und im N_2 entsprechenden Teilnetz aktiviert ist. Folglich ergibt sich, dass eine Schaltfolge aus N sowohl Schaltfolge in N_1 als auch Schaltfolge in N_2 ist. Daher gilt $L(N) = L(N_1) \cap L(N_2)$, und $L(N_1) \cap L(N_2) \in \mathcal{L}(P)$ ist gezeigt.

ii) Inverse Homomorphismen.

Es seien L eine Sprache aus $\mathcal{L}(P)$ und $N = (S, T, F, V, m_0)$ ein Petri-Netz mit $L(N) = L$. Ferner sei ein Homomorphismus $h : (T')^* \rightarrow T^*$ gegeben. Dann ist $T' = T_1 \cup T_2$, wobei

$$T_1 = \{A \mid A \in T', h(A) \neq \lambda\} \quad \text{und} \quad T_2 = \{A \mid A \in T', h(A) = \lambda\}.$$

Es sei $q \in T^+$. Wenn m_1 und m_2 zwei Markierungen von N sind, so dass $q \in L(N, m_1)$ und $q \in L(N, m_2)$, so liegt q auch in $L(N, m)$, wobei m durch die Setzungen $m(s) = \min\{m_1(s), m_2(s)\}$ für $s \in S$ definiert ist, da die minimale Anzahl von Marken ausreichend ist, um q schalten zu können. Damit gibt es für jede Schaltfolge q eine minimale Markierung $H(q)$ mit $q \in L(N, H(q))$. Dabei ist $H(q)$ durch

$$H(q)(s) = \min\{m'(s) \mid q \in L(N, m')\}$$

definiert.

Wir betrachten das Netz $N_1 = (S, T_1, F_1, V_1, m_0)$. Dabei sind F_1 und V_1 wie folgt bestimmt. Für A in T_1 mit $h(A) = q$ setzen wir

$$A^- = H(q) \quad \text{und} \quad A^+ = \Delta q + H(q).$$

Durch diese Setzungen sind F_1 und V_1 eindeutig bestimmt. Dann erhalten wir

$$\Delta(A) = A^+ - A^- = \Delta(q) + H(q) - H(q) = \Delta(q).$$

Dadurch ist $A_1 A_2 \dots A_n$ genau dann eine Schaltfolge in N_1 , wenn $h(A_1)h(A_2) \dots h(A_n)$ eine Schaltfolge in N ist. Somit haben wir $L(N_1) = h_1^{-1}(L(N))$ für den Homomorphismus $h_1 : T_1^* \rightarrow T^*$.

Es bleibt nun, eine Erweiterung des Netzes N_1 vorzunehmen, damit auch die Transitionen aus T_2 erfasst werden. Da $h(A) = \lambda$ für $A \in T_2$ gilt, haben wir die Konstruktion so vorzunehmen, dass $A \in T_2$ immer aktiviert ist. Dies geschieht einfach durch Hinzunahme einer Stelle p , die mit jedem $A \in T_2$ in beiden Richtungen jeweils mit der Vielfachheit 1 verbunden wird und in der Anfangsmarkierung eine Marke erhält. Es ist nun dieses Netz mit N_1 so zu verbinden, dass ein zusammenhängendes Netz entsteht, die Schaltungen aber unabhängig voneinander erfolgen können. Diese Verbindung schaffen wir durch eine zusätzliche Transition z , deren Nachbereich aus einer beliebigen Stelle aus S besteht, deren Vorbereich aus p besteht und $V(p, z) = 2$ erfüllt. Da auf p stets nur eine Marke liegt, kann z nie geschaltet werden. Für das so konstruierte Netz N' gilt $L(N') = h^{-1}(L(N)) = h^{-1}(L)$. \square

Die Frage der Abgeschlossenheit unter dem Kleene-Abschluss lassen wir offen.

Nach den Abschlusseigenschaften wenden wir uns nun den Entscheidungsproblemen zu. Dabei beschränken wir uns auf Mitglieds-, Leerheits-, Endlichkeits-, Universalitäts- und Äquivalenzproblem, die wie folgt spezifiziert sind:

Mitgliedsproblem:

Gegeben: Petri-Netz $N = (S, T, F, V, m_0)$, $q \in T^*$

Frage: Gilt $q \in L(N)$?

Leerheitsproblem:

Gegeben: Petri-Netz N ,

Frage: Enthält $L(N)$ ein nichtleeres Wort?

Endlichkeitsproblem:

Gegeben: Petri-Netz N ,

Frage: Ist $L(N)$ endlich?

Universalitätsproblem:

Gegeben: Petri-Netz $N = (S, T, F, V, m_0)$,

Frage: Gilt $L(N) = T^*$?

Äquivalenzproblem:

Gegeben: zwei Petri-Netz N_1 und N_2

Frage: Gilt $L(N_1) = L(N_2)$?

Wir bemerken, dass wir das Leerheitsproblem gegenüber der üblichen Variante aus der Theorie formaler Sprachen leicht modifiziert haben. Das liegt daran, dass jede Petri-Netz-Sprache das leere Wort enthält. Folglich ist $L(N)$ nie leer. Es bleibt daher die Frage, ob $L(N) \setminus \{\lambda\}$ leer ist. Dies ist offenbar unsere obige Frage.

Offenbar gilt $q \in L(N)$ genau dann, wenn q eine Schaltfolge in N (mit Start in m_0) ist, d.h. wenn wir in N der Reihe nach die Transitionen von q schalten können. Dies ist einfach zu überprüfen. Daher gilt die folgende Aussage.

Satz 4.9 *Das Mitgliedsproblem ist für beliebige Petri-Netze $N = (S, T, F, V, m_0)$ und beliebige Wörter $q \in T^*$ entscheidbar.* \square

Wenn es ein nichtleeres Wort q in $L(N)$ gibt, so ist $q = tq'$ für ein $t \in T$ und ein $q' \in T^*$ und t liegt auch in $L(N)$. Folglich enthält $L(N)$ genau dann ein nichtleeres Wort, wenn es ein $t \in T$ mit $t \in L(N)$ gibt. Da für jedes $t \in T$ die Frage „Ist t in $L(N)$?“ entscheidbar ist, erhalten wir den folgenden Satz.

Satz 4.10 *Das Leerheitsproblem ist für beliebige Petri-Netze entscheidbar.* \square

Satz 4.11 *Das Endlichkeitsproblem ist für beliebige Petri-Netze entscheidbar.* \square

Beweis. Wir entscheiden zuerst, ob das Petri-Netz $N = (S, T, F, V, m_0)$ beschränkt ist. Falls es beschränkt ist, so können wir aus dem endlichen Erreichbarkeitsgraphen einen endlichen Automaten \mathcal{A} konstruieren, der $L(N)$ akzeptiert (siehe Beweis von Satz 4.6, Teil ii)). Da für endliche Automaten (oder für die dazu äquivalenten regulären Grammatiken) entscheidbar ist, ob $T(\mathcal{A})$ endlich ist oder nicht, können wir die Endlichkeit von $L(N)$ entscheiden.

Wenn N nicht beschränkt ist, so gibt es eine Stelle s , auf die beliebig viele Marken geschaltet werden können. Es sei $r = \max\{\Delta(t)(s) \mid t \in T\}$. Bei einem Schalten einer Transition kommen also höchstens r zusätzliche Marken auf s geschaltet werden. Um k Marken auf s zu erreichen, sind somit mindestens n Schaltungen erforderlich, wobei n die kleinste Zahl mit $k - m_0(s) \geq n \cdot r$ ist. Offenbar wächst mit k auch n und wir haben daher in $L(N)$ beliebig lange Schaltfolgen, womit $L(N)$ als unendlich nachgewiesen ist. \square

Satz 4.12 *Das Universalitätsproblem ist für beliebige Petri-Netze entscheidbar.* \square

Beweis. Wir beweisen die folgende Aussage: *Für ein Petri-Netz $N(S, T, F, V, m_0)$ gilt $L(N) = T^*$ genau dann, wenn $\Delta(t) \geq 0$ für alle $t \in T$ gilt und jedes $t \in T$ bei m_0 aktiviert ist.* Da diese Eigenschaft offensichtlich entscheidbar ist, erhalten wir sofort die Aussage des Satzes.

Sei zuerst $L(N) = T^*$. Dann ist $t \in L(N)$ für jede Transition t . Damit ist jede Transition t aktiviert bei m_0 . Außerdem haben wir $t^n \in L(N)$ für jede Transition t und jede natürliche Zahl n . Wie im Beweis von Satz 4.8 (Teil i)) können wir nun nachweisen, dass $\Delta(t) \geq 0$ gilt.

Ist umgekehrt jede Transition t bei m_0 aktiviert, so kann jede Transition bei m_0 geschaltet werden. Somit ist jedes Wort der Länge 1 in $L(N)$. Sei nun schon bewiesen, dass alle Wörter der Länge n in $L(N)$ liegen. Wir betrachten ein beliebiges Wort q der Länge $n+1$. Dann gilt $q = q't$ für ein $q' \in T^n$ und $t \in T$. Wegen $\Delta(t) \geq 0$ für jedes $t \in T$ erhalten wir auch

$$\Delta(t_1 t_2 \dots t_k) = \Delta(t_1) + \Delta(t_2) + \dots + \Delta(t_k) \geq 0$$

für jedes Wort $t_1 t_2 \dots t_k \in T^*$. Damit ist $\Delta(q') \geq 0$. Somit gilt für die nach Schalten von q' erhaltene Markierung m die Beziehung $m = m_0 + \Delta(q') \geq m_0$. Daher ist t aktiviert bei m und $q = q't$ eine Schaltfolge in N . Folglich gilt $q \in L(N)$. Deshalb liegt jedes Wort der Länge $n+1$ in $L(N)$. Damit haben wir $L(N) = T^*$. \square

Satz 4.13 *Das Äquivalenzproblem für Petri-Netze ist unentscheidbar.* \square

Wir verzichten auf den langen Beweis und verweisen auf [11].

4.3 Petri-Netze und Sprachen mit Auswahlkontext

Es ist bekannt, dass nicht alle Aspekte und Phenomena, die bei natürlichen und Programmiersprachen auftreten, durch kontextfreie Sprachen beschrieben werden können. Wir geben ein Beispiel.

Wir betrachten die Programmiersprache ALGOL60, einer der klassische deklarativen Programmiersprachen (ähnliche Konstruktionen lassen sich aber auch für andere Programmiersprachen durchführen). Bei ALGOL60 ist es erforderlich, jede auftretende Variable zu deklarieren. Wir betrachten das Programm

```
begin integer x;  
      y := 1  
end
```

wobei die beiden auftretenden Variablen x und y jeweils durch ein Wort über $\{a, b\}$ gegeben seien. Damit das Programm richtig ist, muss an beiden Stellen des Auftretens einer Variablen jeweils das gleiche Wort stehen. Wenn wir den Durchschnitt von ALGOL60 mit der regulären Menge S aller obigen Programme bilden und durch einen Homomorphismus h dann alle von a und b verschiedenen Symbole löschen ergibt sich

$$R = h''(\text{ALGOL} \cap S) = \{xx \mid x \in \{a, b\}^*\}$$

Wenn ALGOL60 kontextfrei wäre, so würde auch R kontextfrei sein, was aber nicht der Fall ist. Damit ist ALGOL60 keine kontextfreie Sprache.

Zur Beschreibung von natürlichen Sprachen und Programmiersprachen bedarf es also Mechanismen, die auch gewisse nicht-kontextfreie Sprachen erzeugen können. Dies ist sicher für die kontextsensitiven Grammatiken der Fall, die aber zwei negative Aspekte haben: Einige Entscheidbarkeitsprobleme sind für kontextsensitive Grammatiken unentscheidbar bzw. sehr schwer; so sind z.B. das Endlichkeits- und das Leerheitsproblem unentscheidbar. Zum anderen gibt es für kontextsensitive Grammatiken keine vernünftigen Ableitungsbäume, da mehrere Nichtterminale aus verschiedenen Schichten des Baumes von einer Regel betroffen sein können. Ableitungsbäume haben sich aber als sehr gutes Instrument für die Analyse von Sätzen in natürlichen Sprachen bzw. von Programmen erwiesen.

Daher ist man an Erweiterungen der kontextfreien Grammatiken interessiert, die zum einen Ableitungsbäume ermöglichen (d.h. jede angewendete Regel muss kontextfrei sein) und zum anderen möglichst einfach zu entscheidende Probleme haben. Wir präsentieren eine derartige Grammatik.

Definition 4.14 *i) Eine Grammatik mit Auswahlkontext ist ein Quadrupel $G = (N, T, P, S)$, wobei*

- N, T, S die Menge der Nichtterminale, die Menge der Terminale bzw. das Startwort (Axiom) sind,
- P eine endliche Menge von Paaren $p = (r_p, E_p)$ ist, wobei jeweils $r_p = A_p \rightarrow w_p$ eine kontextfreie Regel ist, und E_p eine Teilmenge von N ist.

ii) Für zwei nichtleere Wörter x und y über $N \cup T$ sagen wir, dass y aus x durch Anwendung von $p = (A \rightarrow w, E)$ erzeugt wird (geschrieben als $x \Longrightarrow_p y$), wenn folgende Bedingungen erfüllt sind:

- $x = uAv$, $y = uwv$ (kontextfreie Ersetzung)
- jedes Symbol aus E kommt in uv vor.

Wir sagen, dass y aus x in G durch einen Ableitungsschritt entsteht (geschrieben als $x \Longrightarrow_G y$), wenn es eine Regel p mit $x \Longrightarrow_p y$ gibt.

iii) Die von einer Grammatik $G = (N, T, P, S)$ mit Auswahlkontext erzeugte Sprache $L(G)$ ist

$$L(G) = \{w \mid S \Longrightarrow_G^* w, w \in T^*\},$$

wobei \Longrightarrow_G^* der reflexive und transitive Abschluss von \Longrightarrow_G ist.

Bei einer Regel $p = (A \rightarrow w, E)$ heißt E der geforderte Kontext der Regel $A \rightarrow w$.

Die Definition der Grammatiken mit Auswahlkontext ist speziell darauf abgestellt, einen Zusammenhang zwischen Deklaration und Benutzung von Variablen in Programmiersprachen zu ermöglichen, d.h. den Grund dafür, dass oben im dritten Beispiel eine nicht kontextfreie Sprache auftritt, zu eliminieren. Dies geschieht dadurch, dass eine Benutzung einer Variablen entsprechend einer Regel nur möglich ist, wenn diese schon in der Satzform vorkommt.

Wir bemerken weiterhin, dass Grammatiken mit Auswahlkontext eine Erweiterung der kontextfreien Grammatiken sind, denn bei der Wahl von $E = \emptyset$ für jede Regel, ergibt sich eine kontextfreie Grammatik, da nun keine Bedingungen mehr für die Anwendung von Regeln vorliegen (also jede Regel jederzeit anwendbar ist); und umgekehrt kann jede kontextfreie Grammatik als eine Grammatik mit Auswahlkontext aufgefasst werden, bei der alle geforderten Kontexte leer sind.

Wir zeigen nun, dass Grammatiken mit Auswahlkontext Sprachen erzeugen können, die nicht kontextfrei sind.

Beispiel 4.15 Wir betrachten die Grammatik mit Auswahlkontext

$$G_1 = (\{S, A, A', A_a, A_b, B, B'\}, \{a, b, c\}, \{p_0, p_1, \dots, p_{10}\}, S)$$

mit den Regeln

$$\begin{array}{lll} p_0 = (S \rightarrow AB, \emptyset), & p_1 = (A \rightarrow aA_a, \{B\}), & p_2 = (A \rightarrow bA_b, \{B\}), \\ p_3 = (B \rightarrow aB', \{A_a\}), & p_4 = (B \rightarrow bB', \{A_b\}), & p_5 = (A_a \rightarrow A, \{B'\}), \\ p_6 = (A_b \rightarrow A, \{B'\}), & p_7 = (B' \rightarrow B, \{A\}) & p_8 = (A \rightarrow A', \{B\}), \\ p_9 = (B \rightarrow \lambda, \{A'\}), & p_{10} = (A' \rightarrow \lambda, \emptyset). \end{array}$$

Offensichtlich beginnt jede Ableitung mit der einzigen Regel für S , d.h. wir erhalten $S \Longrightarrow AB$. Wir wollen etwas allgemeiner von einer Satzform $wAwB$ ausgehen (das in einem Schritt erhaltene Wort AB wird gerade bei $w = \lambda$ erhalten). Wir können keine der Regeln mit linker Seite B anwenden, da der jeweilige geforderte Kontext A_a bzw. A_b bzw. A' in der Satzform nicht vorhanden ist. Folglich muss eine Regel mit linker Seite A verwendet werden. Wir unterscheiden drei Fälle.

Fall 1. Es wird die Regel p_1 angewendet. Dadurch erhalten wir $waA_a wB$. Da die einzige Regel für A_a den geforderten Kontext B' hat, ist eine Regel für B anzuwenden. Es kommt

nur Regel p_3 in Frage, da bei den anderen Regeln wieder der geforderte Kontext fehlt. Daher ergibt sich $waA_a waB'$. Jetzt ist nur Regel p_5 anwendbar, wodurch $waAwaB'$ entsteht. Nun ist Regel p_7 anzuwenden, und wir erhalten $waAwaB$. Damit haben wir wieder die Ausgangssituation erhalten, nur dass das Wort an beiden Stellen um den Buchstaben a verlängert wurde.

Fall 2. Es wird die Regel p_2 angewendet. Dann ergibt sich analog zu Fall 1 als einzig mögliche Ableitung

$$wAwB \xRightarrow{p_2} wbA_b wB \xRightarrow{p_4} wbA_b wbB' \xRightarrow{p_6} wbAwbB' \xRightarrow{p_7} wbAwbB,$$

d.h. wir haben das Wort an beiden Stellen um den Buchstaben b verlängert.

Fall 3. Es wird die Regel p_8 angewendet. Dann ergibt sich die eindeutige Ableitung

$$wAwB \xRightarrow{p_8} wA'wB \xRightarrow{p_9} wA'w \xRightarrow{p_{10}} ww$$

(man beachte, dass die Anwendung von p_{10} auf $wA'wB$ zwar möglich ist, aber wwB liefert, worauf keine Regel mehr anwendbar ist und somit kein terminales Wort erreicht werden kann).

Aufgrund der drei Fälle ist sofort zu sehen, dass

$$L(G_1) = \{ww \mid w \in \{a, b\}^*\}$$

gilt.

Beispiel 4.16 Wir betrachten die Grammatik

$$G_2 = (\{S, A, B, A', B'\}, \{a, b, c\}, \{q_0, q_1, \dots, q_7\}, S),$$

wobei die Regeln durch

$$\begin{aligned} q_0 &= (S \rightarrow AB, \emptyset), & q_1 &= (A \rightarrow aA'b, \{B\}), & q_2 &= (B \rightarrow B'c, \{A'\}), \\ q_3 &= (A' \rightarrow A, \{B'\}), & q_4 &= (B' \rightarrow B, \{A\}), & q_5 &= (A \rightarrow A'', \{B\}), \\ q_6 &= (B \rightarrow c, \{A''\}), & q_7 &= (A'' \rightarrow ab, \emptyset) \end{aligned}$$

gegeben sind. Es ist leicht – in Analogie zum vorhergehenden Beispiel – zu sehen, dass ausgehend von einem Wort der Form $a^n Ab^n Bc^n$ mit $n \geq 0$ (durch Anwendung der einzigen Regel für das Axiom entsteht ein Wort dieser Form mit $n = 0$) nur die beiden folgenden Ableitungen möglich sind:

$$\begin{aligned} a^n Ab^n Bc^n &\xRightarrow{} a^n aA'bb^n Bc^n \xRightarrow{} a^n aA'bb^n B'cc^n \xRightarrow{} a^{n+1} Ab^{n+1} B'c^{n+1} \xRightarrow{} a^{n+1} Ab^{n+1} Bc^{n+1}, \\ a^n Ab^n Bc^n &\xRightarrow{} a^n A''b^n Bc^n \xRightarrow{} a^n A''b^n cc^n \xRightarrow{} a^n abb^n c^{n+1} = a^{n+1} b^{n+1} c^{n+1}. \end{aligned}$$

Hieraus folgt sofort

$$L(G_2) = \{a^n b^n c^n \mid n \geq 1\}.$$

Es sei eine Grammatik $G = (N, T, P, S)$ mit Auswahlkontext gegeben. Dabei können wir ohne Beschränkung der Allgemeinheit annehmen, dass bei jeder Regel $p = (A_p \rightarrow w_p, E_p)$ die Beziehung $A_p \notin E_p$ gilt (die Forderung nach der Existenz von A_p in der Satzform, auf die die Regel angewendet werden soll, ist nicht notwendig, da nur bei Existenz von A_p die Regel $A_p \rightarrow w_p$ anwendbar ist). Wir definieren den Homomorphismus

$h : (N \cup T)^* \rightarrow N^*$ durch $h(A) = A$ für $A \in N$ und $h(a) = \lambda$ für $a \in T$. Für ein Wort $w \in (N \cup T)^*$ wird $h(w)$ offenbar dadurch gebildet, dass alle Terminale in w gestrichen werden.

Wir konstruieren nun das Petri-Netz $H = (N, P, F, V, m_0)$, bei dem die Menge der Stellen durch die Menge der Nichtterminale von G und die Menge der Transitionen durch die Regeln von G gegeben ist und F, V und m_0 durch folgende Setzungen definiert sind. Für eine Regel $p = (A_p \rightarrow w_p, E_p)$

$$\begin{aligned} p^-(A) &= \begin{cases} 1 & \text{for } A \in E_p \\ 1 & \text{for } A = A_p \\ 0 & \text{otherwise} \end{cases} , \\ p^+(A) &= \begin{cases} 1 + \#_A(h(w_p)) & \text{for } A \in E_p \\ \#_A(h(w_p)) & \text{otherwise} \end{cases} , \\ m_0(A) &= \begin{cases} 1 & \text{for } S \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Bei der Anfangsmarkierung ist genau eine Marke im Netz, die auf der Stelle S liegt. Offensichtlich ist die Transition p aktiviert, wenn auf jeder Stelle A aus E_p und auf der Stelle A_p jeweils mindestens eine Marke liegt, d.h. wenn p als Regel anwendbar ist. Beim Schalten von p wird die Marke von A_p abgezogen, jeweils eine Marke auf den A aus E_p wird abgezogen und wieder hingelegt und zusätzlich werden auf jede Stelle A soviel Marken abgelegt, wie A in w_p vorkommen. Aus diesen Bemerkungen folgt, dass jeder Ableitung

$$S = w_0 \xRightarrow{p_1} w_1 \xRightarrow{p_2} w_2 \xRightarrow{p_3} w_3 \xRightarrow{p_4} \dots \xRightarrow{p_{k-1}} w_{k-1} \xRightarrow{p_k} w_k \in T^*$$

eine Folge

$$m_0[p_1 > m_1[p_2 > m_2[p_3 > m_3[p_4 > \dots [p_{k-1} > m_{k-1}[p_k > m_k$$

mit

$$m_i(A) = \#_A(h(w_i))$$

entspricht und umgekehrt. Da w_k in T^* liegt, gilt $\#_A(w_k) = 0$ für alle $A \in N$. Daher wird genau dann ein Wort über dem Terminalalphabet erzeugt, wenn die 0-Markierung in H erreichbar ist. Somit ist die von G erzeugte Sprache $L(G)$ genau dann nicht leer, wenn die 0-Markierung in H erreichbar ist. Da die Erreichbarkeit einer Markierung in einem Netz nach Satz 3.31 entscheidbar ist, erhalten wir den folgenden Satz.

Satz 4.17 *Das Leerheitsproblem für Grammatiken mit Auswahlkontext ist entscheidbar.*

Da die Regeln einer Grammatik mit Auswahlkontext kontextfrei sind, können wir für diese Grammatiken Ableitungsbäume definieren. Außerdem sind wir in der Lage auch einige nicht-kontextfreie Sprache zu erzeugen (darunter die Sprachen, die man aus der Sicht natürlicher Sprachen für erforderlich hält). Während das Leerheitsproblem für kontextsensitive Grammatiken unentscheidbar ist, ist es für Grammatiken mit Auswahlkontext entscheidbar. Damit sind Grammatiken mit Auswahlkontext ein möglicher Kandidat zur Beschreibung von natürlichen und Programmiersprachen.

Literaturverzeichnis

- [1] B. BAUMGARTEN, *Petri-Netze. Grundlagen und Anwendungen.* BI-Wissenschaftsverlag, Mannheim, 1990.
- [2] M. JANTZEN, The large marking problem. *Petri Net Newsletter* **14** (1983) 24–25.
- [3] S. R. KOSARAJU, Decidability of reachability in vector addition systems. *Proc. 14th Ann. ACM STOC* (1982) 267–281.
- [4] E. W. MAYR, An algorithm for the general petri net reachability problem. *SIAM J. Comput.* **13** (1984) 441–460.
- [5] H. MÜLLER, The reachability problem for VAS. In: *Lecture Notes in Computer Science* **188** (1984) 376–391.
- [6] W. REISIG, *Petrinetze. Eine Einführung.* Springer-Verlag, Berlin, 1991.
- [7] A. SCHRIJVER, *Theory of Linear and Integer Programming.* Wiley, 1986.
- [8] P. H. STARKE, *Petri-Netze.* VEB Deutscher Verlag der Wissenschaften, Berlin, 1980.
- [9] P. H. STARKE, *Analyse von Petri-Netz-Modellen.* B. G. Teubner, Stuttgart, 1990.
- [10] J. WEHLER, *Petri Nets.* Vorlesungsskript, Ludwig-Maximilians-Universität München, 1999.
- [11] L. PRIESE und A. WIMMEL, *Petri-Netze.* Springer-Verlag, Berlin, 2008.
- [12] J. DASSOW, *Theoretische Informatik.* Vorlesungsskript, Otto-von-Guericke-Universität Magdeburg, 2005 bzw. 2007.
- [13] A. SALOMAA, *Formale Sprachen.* Springer-Verlag, Berlin, 1978.