Technische Universität München
Fakultät für Informatik
Lehrstuhl Informatik XIV
Prof. Dr. Ernst W. Mayr
Dr. Werner Meixner

Sommersemester 2015
Lösungsblatt 5
22. Mai 2015

Theoretische Informatik

Hausaufgabe 1 (4 Punkte)

Sei $\Sigma = \{0, 1, (,), +, ^*, \boldsymbol{\emptyset}, \boldsymbol{\epsilon}\}$ die Zeichenmenge, aus der reguläre Ausdrücke über dem Alphabet $\{0, 1\}$ gebildet werden. Wir schreiben hier $+$ anstelle von $|$, um Zeichenverwirrungen zu vermeiden.
Die induktive Definition der regulären Ausdrücke kann man als eine Grammatik $G = (\{R\}, \Sigma, P, R)$ auffassen, deren Produktionenmenge P aus den folgenden Produktionen besteht:

$$R \to 0 \mid 1 \mid \boldsymbol{\epsilon} \mid \boldsymbol{\emptyset} \mid RR \mid R + R \mid R^* \mid (R)$$

1. Zeigen Sie, dass die Grammatik G nicht eindeutig ist.
2. Geben Sie eine eindeutige Grammatik G' für die regulären Ausdrücke an, die die Bindungstärken in regulären Ausdrücken respektiert (also z.B. Konkatenation stärker als $+$ bindet).

   Begründen Sie, dass Ihre Grammatik eindeutig ist.
3. Geben Sie den Syntaxbaum für das Wort 01*0+1 bezüglich Ihrer eindeutigen Grammatik an.

Lösung

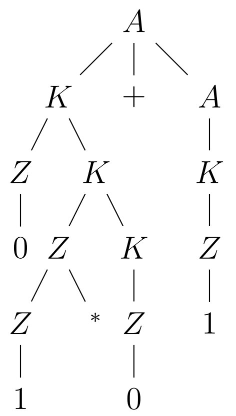
1. Die obige Grammatik ist nicht eindeutig, da z.B. der Ausdruck $0 + 00$ zwei Syntaxbäume hat, die jeweils der Klammerung $0 + (00)$ bzw. $(0 + 0)0$ entsprechen. (1P)
2. Eine eindeutige Grammatik erhält man, indem man mehrere Nichtterminale verwendet wie folgt.

   Wir setzen $G' = (\{A, K, Z\}, \Sigma, P', A)$ mit den Produktionen

$$\begin{aligned} A &\to K \mid K + A \\ K &\to Z \mid ZK \\ Z &\to 0 \mid 1 \mid \boldsymbol{\epsilon} \mid \boldsymbol{\emptyset} \mid (A) \mid Z^* \end{aligned}$$

   Diese Grammatik ist eindeutig, denn jedes Wort w in $L(G')$ ist entweder eine Alternative oder es ist keine Alternative. Es steht also fest, ob die erste angewandte Produktion $A \to K + A$ oder $A \to K$ ist, und nur eine der beiden Produktionen kann als erstes angewandt werden. Entsprechendes gilt für $w \in L(K)$ und die Konkatenation. Bei allen Ableitungen von Wörtern $w \in L(Z)$ aus der Variablen Z ist ebenfalls die erste Produktionsanwendung durch w vorgegeben. (2P)

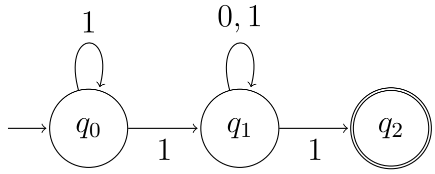
3. Der Syntaxbaum sieht mit G' wie folgt aus:



(1P)

Hausaufgabe 2 (4 Punkte)

Durch die folgende Grafik sei ein nichtdeterministischer endlicher Automat $A = (Q, \Sigma, \delta, \{q_0\}, F)$ gegeben.



1. Bestimmen Sie $\hat{\delta}(\{q_0, q_1, q_2\}, 0)$! Geben Sie einen regulären Ausdruck α an, der die von A akzeptierte Sprache beschreibt, d. h. $L(\alpha) = L(A)$.

2. Konstruieren Sie mit dem Potenzmengenverfahren einen deterministischen endlichen Automaten B, der $L(A)$ akzeptiert.

   Stellen Sie den erhaltenen Automaten B als Übergangsgraph dar.

3. Wie ist der Automat B abzuändern, so dass B das Komplement von $L(A)$ akzeptiert, d. h., so dass $L(B) = \{0, 1\}^* \setminus L(A)$ gilt?

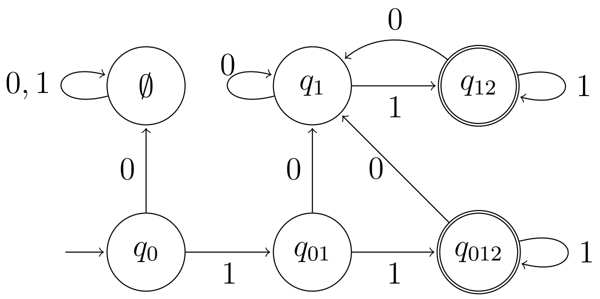
Lösung

1. $\overline{\delta}(\{q_0, q_1, q_2\}, 0) = \{q_1\}$.
   $\alpha = 1^*1(0|1)^*1$. (1P)

2. Anwendung des Potenzmengenverfahrens mit Kurznotation:

q_i	$\delta(q_i, 0)$	$\delta(q_i, 1)$
q_0	$\emptyset$	q_0q_1
q_0q_1	q_1	$q_0q_1q_2$
q_1	q_1	q_1q_2
$q_0q_1q_2$	q_1	$q_0q_1q_2$
q_1q_2	q_1	q_1q_2
$\emptyset$	$\emptyset$	$\emptyset$

Darstellung als Graph:



(2P)

3. Es muss F_B durch $F' = Q \setminus F_B$ ersetzt werden. (1P)

Hausaufgabe 3 (4 Punkte)

Sei Σ eine nichtleere Zeichenmenge.

1. Sei L die Menge aller Wörter über Σ mit geradzahliger Länge. Geben Sie ein Verfahren an, das für einen beliebigen DFA $A = (Q, \Sigma, \delta, q_0, F)$ entscheidet, ob jedes (oder nicht jedes) von A akzeptierte Wort eine ungerade Länge besitzt.

2. Sei $E(x)$ ein Eigenschaft für Wörter x über Σ, so dass $K = \{x\,;\, E(x)\}$ eine reguläre Sprache ist. Verallgemeinern Sie Ihr obiges Verfahren so, dass es für einen beliebigen DFA $A = (Q, \Sigma, \delta, q_0, F)$ entscheidet, ob jedes (oder nicht jedes) von A akzeptierte Wort x die Eigenschaft E besitzt.

Lösung

1. Die Eigenschaft von A, dass jedes akzeptierte Wort ungeradzahlige Länge besitzt, trifft genau dann zu, wenn $L \cap L(A) = \emptyset$ gilt.

   Sei $\Sigma = \{a_1, a_2, \ldots, a_n\}$ n-elementig mit $n \geq 1$. Da $L = L((\Sigma^2)^*)$ gilt, konstruieren wir zu dem regulären Ausdruck $((a_1|a_2|\ldots|a_n)^2)^*$ standardmäßig einen DFA B mit $L(B) = L$.

   Dann konstruieren wir nach Vorlesung einen Produkt-Automat C mit $L(C) = L(A) \cap L(B)$.

   Jedes Verfahren, das $L(C) = \emptyset$ entscheidet, erfüllt die Aufgabenstellung. (2P)

2. Sei $\overline{K}$ das Komplement von K in Σ^*, d. h., es gelte $\overline{K} = \Sigma^* \setminus K$.

   Die einfache Verallgemeinerung besteht nun darin, die Menge L der vorausgegangenen Teilaufgabe durch $\overline{K}$ zu ersetzen und für B einen DFA zu wählen, der $\overline{K}$ akzeptiert. (2P)

Hausaufgabe 4 (4 Punkte)

Stellen Sie die folgenden Sprachen jeweils durch einen regulären Ausdruck dar:

1. Die Sprache L_1 aller Wörter über $\{0,1\}$, die mit 101 beginnen und mit 010 enden.

2. Die Sprache $L_2 = \{a^m b^n\,;\,(m+n) \bmod 3 = 1\}$.

3. Die Sprache L_3 aller Wörter über $\{0,1\}$, in denen kein Paar aufeinanderfolgender Nullen weiter links steht als ein beliebiges Paar benachbarter Einsen.

   Hinweis: Sei $w \in L_3$ mit $w = x_1x_2\ldots x_n$ und $x_i \in \{0,1\}$. Sei i der größte Index (größte Zahl), so dass $x_ix_{i+1} = 11$. i gibt also die Position des am weitesten rechts stehenden Einserpaares. Falls kein Einserpaar existiert, dann setzen wir $i = 0$.
   Dann ist klar, dass das Wort $v = x_{i+1}x_{i+2}\ldots x_n$ kein Einserpaar enthält. Andererseits darf nach Definition von L_3 das Wort $u = x_1x_2\ldots x_i$ kein Paar von Nullen enthalten.

Lösung

1. $L_1 = 101\,(0\,|\,(1|0)^*010)$. (1P)

2. $\alpha = (aaa)^*(a|b|aabb)(bbb)^*$. (1P)

3. Seien L_l bzw. L_r die Sprachen über Σ, die kein Paar von benachbarten Nullen bzw. Einsen enthalten. Offensichtlich gilt dann

$$L_lL_r \subseteq L_3\,.$$

   Gemäß obigem Hinweis gilt auch die Umkehrung, d. h. es gilt $L_3 = L_lL_r$.

   Nun werden L_l bzw. L_r als Sprache der regulären Ausdrücke

$$l = (01|1)^*(\epsilon|0) \quad \text{bzw.} \quad r = (10|0)^*(\epsilon|1)$$

   dargestellt und wir erhalten die Lösung

$$(01|1)^*(\epsilon|0)(10|0)^*(\epsilon|1)\,.$$

   Diese läßt sich noch vereinfachen zu $(01|1)^*(10|0)^*(\epsilon|1)$. (2P)

Hausaufgabe 5 (4 Punkte)

Das Reißverschlussprodukt von zwei Sprachen L_1 und L_2 ist wie folgt definiert:

$$L_1 \,\%\, L_2 = \{a_1b_1a_2b_2\ldots a_nb_n\,;\, a_i \in \Sigma, b_i \in \Sigma, a_1\ldots a_n \in L_1, b_1\ldots b_n \in L_2\}$$

1. Geben Sie einen regulären Ausdruck an, der $L(aba^*) \,\%\, L(bc^*)$ beschreibt.

2. Zeigen Sie durch eine geeignete Automatenkonstruktion: Wenn L_1 und L_2 regulär sind, dann ist auch $L_1 \,\%\, L_2$ regulär.

   Beschreiben Sie Ihre Konstruktion zunächst informell. Für die Konstruktion bildet man das Produkt der beiden Automaten mit einem zusätzlichen Flag, welches angibt, welcher der beiden Automaten gerade „an der Reihe“ ist.

Lösung

1. $abbc(ac)^*$ (1P)

2. Für die Konstruktion bildet man das Produkt der beiden Automaten mit einem zusätzlichen Flag, welches angibt, welcher der beiden Automaten gerade "an der Reihe" ist. Wir nehmen also DFA's $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ und $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ an und konstruieren daraus den DFA

$$A' = (Q_1 \times Q_2 \times \{1, 2\}, \Sigma, \delta', (q_{01}, q_{02}, 1), F_1 \times F_2 \times \{1\})$$

   Die Übergangsfunktion δ' ist definiert mit

$$\begin{aligned} \delta'((q, q', 1), a) &= (\delta(q, a), q', 2) && \forall q \in Q_1, q' \in Q_2, a \in \Sigma \\ \delta'((q, q', 2), a) &= (q, \delta(q', a), 1) && \forall q \in Q_1, q' \in Q_2, a \in \Sigma \end{aligned}$$

   (3P)

Zusatzaufgabe 3 (wird nicht korrigiert)

Das sogenannte Shuffle-Produkt spielt in der Theorie der nebenläufigen Systeme eine wichtige Rolle. Für zwei Sprachen L_1 und L_2 bezeichnet $L_1 \parallel L_2$ die Menge der Wörter, die man erhält, indem man zwei Wörter $v \in L_1$ und $w \in L_2$ beliebig miteinander verschränkt. Dabei können sich Teile aus v und w beliebig abwechseln, wobei die Reihenfolge der Zeichen aus v und w jedoch erhalten bleibt. Das kann man sich gut als das Ineinanderschieben zweier Kartenstapel veranschaulichen. Formal definieren wir $L_1 \parallel L_2$ wie folgt:

$$\begin{aligned} L_1 \parallel L_2 = \{v_1 w_1 v_2 w_2 \cdots v_n w_n \,;\ & v_1, \ldots, v_n, w_1, \ldots, w_n \in \Sigma^*, \\ & v_1 v_2 \cdots v_n \in L_1 \text{ und } w_1 w_2 \cdots w_n \in L_2\} \end{aligned}$$

1. Versuchen Sie, eine einfache Beschreibung von $L((01)^*) \parallel L((10)^*)$ zu finden.

2. Begründen Sie: Wenn L_1 und L_2 regulär sind, dann ist auch $L_1 \| L_2$ regulär.
   Hinweis: Konstruieren Sie einen NFA für $L_1 \parallel L_2$.

3. Führen Sie die Konstruktion konkret für die Sprachen $L((01)^*)$ und $L((10)^*)$ durch und geben Sie einen regulären Ausdruck an, der die Sprache beschreibt.

4. Vergleichen Sie dass Reißverschlussprodukt mit dem Shuffle-Produkt und finden Sie zwei Sprachen A und B und ein Wort, das in $(A \parallel B)$, aber nicht in $A \,\%\, B$ liegt.

Lösung

1. Es ist auf Anhieb nicht klar, wie die Sprache $L((01)^*) \parallel L((10)^*)$ aussieht. Die erste Hypothese, dass es sich um $\{0, 1\}^*$ handeln könnte, ist falsch, denn in beiden Teilsprachen ist die Anzahl der Einsen und der Nullen stets gleich, also muss das auch für das Shuffle-Produkt gelten. Andererseits ist z.B. das Wort 000111 nicht möglich, da drei Nullen hintereinander nicht vorkommen können. Man sieht also, dass eine kompakte Beschreibung des Shuffle-Produkts unter Umständen schwierig sein kann. Die Automatenkonstruktion im nächsten Aufgabenteil hilft uns aber weiter.

2. Aus zwei NFAs $N_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ und $N_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ für L_1 bzw. L_2 konstruieren wir einen NFA, der das Shuffle-Produkt akzeptiert. Wir verwenden einen Produktautomaten, dessen Zustände genau Paare von Zuständen der ursprünglichen Automaten sind:

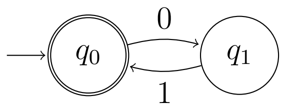
$$N_\| = (Q_1 \times Q_2, \Sigma, \delta_\|, (q_{01}, q_{02}), F_1 \times F_2)$$

Ein Schritt im neuen Automaten besteht darin, dass einer der beiden "Teilautomaten" einen Schritt macht:

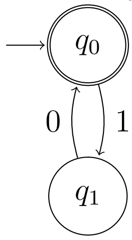
$$\delta_\|((q_1, q_2), a) = \{(q', q_2) \mid q' \in \delta_1(q_1, a)\} \cup \{(q_1, q') \mid q' \in \delta_2(q_2, a)\}$$

3. Zu den regulären Ausdrücken $(01)^*$ und $(10)^*$ kann man leicht je einen NFA angeben:

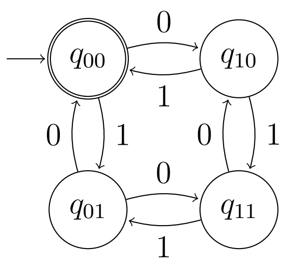
N_1:



N_2:



Der Automat $N_\|$ für das Shuffle-Produkt hat dementsprechend vier Zustände:



Hier kann man jetzt auch etwas leichter ablesen, wie die Produktsprache aussieht: Sie kann z.B. durch den regulären Ausdruck $(01|10)^*$ beschrieben werden.

4. Ein einfaches Beispiel ist $A = \{a, aa\}$ und $B = \{b\}$. Dann ist $A \| B = \{ba, ab, baa, aba, aab\}$, dagegen ist $A \% B = \{ab\}$. Im Allgemeinen gilt stets $A \% B \subseteq A \| B$.

Hinweis: *Die Vorbereitungsaufgaben bereiten die Tutoraufgaben vor und werden in der Zentralübung unterstützt. Tutoraufgaben werden in den Übungsgruppen bearbeitet. Hausaufgaben sollen selbstständig bearbeitet und zur Korrektur und Bewertung abgegeben werden.*

Vorbereitung 1

1. Gegeben seien $\Sigma = \{d, e, f\}$ und $L = \{d^k e^m f^n \;;\; k, m, n \in \mathbb{N}, m \neq n\}$.

   Zeigen Sie, dass die Sprache L kontextfrei ist.

2. Gegeben seien $\Sigma = \{d, e, f\}$ und $L = \{d^m e^m f^n \;;\; m, n \in \mathbb{N}, m \neq n\}$.

   Zeigen Sie, dass die Sprache L ein Durchschnitt kontextfreier Sprachen ist.

Lösung

1. Es wäre ungeschickt, für L eine kontextfreie Grammatik zu konstruieren. Besser ist es, die Abgeschlossenheit der Klasse der kontextfreien Sprachen zu benutzen.

   Wir stellen L dar durch $L = L_1(L_2 \bigcup L_3)$, wobei wir definieren

$$\begin{aligned} L_1 &= \{d^k \,;\, k \in \mathbb{N}\}\,, \\ L_2 &= \{e^m f^n \,;\, m, n \in \mathbb{N}, m < n\}\,, \\ L_3 &= \{e^m f^n \,;\, m, n \in \mathbb{N}, m > n\}\,. \end{aligned}$$

   L_1 ist wegen $L_1 = L(d^*)$ regulär, also auch kontextfrei.

   L_2 wird erzeugt durch die kontextfreie Grammatik G_2 mit den Produktionen

$$\begin{aligned} S &\rightarrow eSf \mid T, \\ T &\rightarrow Tf \mid f. \end{aligned}$$

   L_2 ist also kontextfrei. Analog wird mit Produktionen $T \rightarrow eT \mid e$ gezeigt, dass L_3 kontextfrei ist. Da alle L_i kontextfrei sind, ist auch L kontextfrei.

2. Mit

$$\begin{aligned} L_1 &= \{d^k e^m f^n \,;\, k, m, n \in \mathbb{N}, m \neq n\}\,, \\ L_2 &= \{d^m e^m f^k \,;\, k, m \in \mathbb{N}\} \end{aligned}$$

   gilt

$$L = L_1 \cap L_2\,.$$

   Nach Teilaufgabe 1 ist L_1 kontextfrei. Für L_2 gilt $L_2 = L_3 L_4$ mit

$$\begin{aligned} L_3 &= \{d^m e^m \,;\, m \in \mathbb{N}\}\,, \\ L_4 &= \{f^k \,;\, k \in \mathbb{N}\}\,. \end{aligned}$$

   L_3 ist bekanntlich kontextfrei. $L_4 = L(f^*)$ ist ebenfalls kontextfrei. Also ist auch die Konkatenation $L_2 = L_3 L_4$ kontextfrei.

Vorbereitung 2

(Wiederholungsaufgabe)
Wandeln Sie die durch folgende Produktionen gegebene Grammatik mit Startsymbol S in Chomsky-Normalform um:

$$\begin{array}{lllllll} S & \to & 0A0 \mid 1B1 \mid BB\,, & & & & \\ A & \to & D\,, & B & \to & S \mid A\,, \\ C & \to & S \mid \epsilon\,, & D & \to & C\,. \end{array}$$

Lösung

Wir können Variablen mit nur einer einzigen Produktion durch "Anwendung" spracherhaltend eliminieren. Somit vereinfachen wir die Grammatik zunächst zu

$$S \to 0A0 \mid 1B1 \mid BB\,, \quad A \to C\,, \quad B \to S \mid A\,, \quad C \to S \mid \epsilon\,.$$

und schließlich zu

$$S \to 0C0 \mid 1B1 \mid BB\,, \quad B \to S \mid C\,, \quad C \to S \mid \epsilon\,.$$

Nun eliminieren wir die ϵ-Produktionen. Dazu bauen wir die Menge P' wie in Lemma 12 auf. Wir müssen die Produktionen $S \to 00$, $S \to 11$, $S \to B$ zu P (ohne ϵ-Produktionen) hinzufügen und erhalten

$$S \to 0C0 \mid 00 \mid 1B1 \mid 11 \mid BB \mid B\,, \quad B \to S \mid C\,, \quad C \to S\,.$$

Die Produktionen $T \to \epsilon \mid S$ fügt man hinzu, wenn die Sprache ϵ enthält, was hier der Fall ist, und erklärt in diesem Fall T zum Axiom.

Wir eliminieren jetzt die Kettenproduktion $C \to S$, da die Variable C nur in einer einzigen Produktion links vorkommt, und zwar zunächst durch Ersetzen von C, d.h. Anwendung von $C \to S$.

$$S \to 0S0 \mid 00 \mid 1B1 \mid 11 \mid BB \mid B\,, \quad B \to S\,.$$

Die Kettenproduktion $B \to S$ tritt nun ebenfalls links nur in einer einzigen Produktion auf und kann durch Anwendung eliminiert werden, wie folgt.

$$S \to 0S0 \mid 00 \mid 1S1 \mid 11 \mid SS \mid S\,.$$

$S \to S$ kann entfernt werden:

$$S \to 0S0 \mid 00 \mid 1S1 \mid 11 \mid SS\,.$$

Nun erst wenden wir den Algorithmus der Vorlesung im Abschnitt 4.2 an.

Im ersten Schritt führen wir neue Variablen N und E für 0 bzw. 1 ein:

$$S \to NSN \mid NN \mid ESE \mid EE \mid SS\,, \quad N \to 0\,, \quad E \to 1\,.$$

Die Dreier-Produktionen eliminieren wir, indem wir Zwischensymbole T_1 und T_2 einfügen:

$$S \to NT_1 \mid NN \mid ET_2 \mid EE \mid SS\,, \quad T_1 \to SN\,, \quad T_2 \to SE\,, \quad N \to 0\,, \quad E \to 1\,.$$

Diese Grammatik ist nun in Chomsky-Normalform und beschreibt dieselbe Sprache, ausgenommen dem ϵ.

Da das leere Wort in der Sprache ist, fügen wir die Produktionen $T \to \epsilon \mid \alpha$ für jede rechte Seite α einer Produktion $S \to \alpha$ hinzu und erklären T zum Axiom, ohne die Eigenschaft der Grammatik, in Chomsky-Normalform zu sein, zu verletzen.

Bemerkung: Durch die Eliminationsschritte zur Beseitigung der Kettenproduktionen entfällt der aufwendige Schritt 3 im Algorithmus der Vorlesung zur Konstruktion einer Grammatik in Chomsky-Normalform.

Tutoraufgabe 1 (CYK)

Wir betrachten die kontextfreie Grammatik $G = (V, \{a, b\}, P, S)$ mit Produktionen in Chomsky-Normalform

$$
\begin{aligned}
S &\to YC_3 \mid YX \mid AC_3 \mid AX \mid YY \mid YB \mid YA \mid AY \mid AB \mid AA\,,\\
X &\to ZC_2 \mid ZB \mid BZ \mid BB\,,\\
Y &\to AC_1 \mid AY \mid AA\,,\\
C_1 &\to YZ \mid AZ\,,\\
C_2 &\to BZ\,,\\
Z &\to YC_3 \mid YX \mid AC_3 \mid AX \mid YY \mid YB \mid YA \mid AY \mid AB \mid AA\,,\\
C_3 &\to XZ \mid YZ \mid BZ \mid AZ\,,\\
A &\to a\,,\\
B &\to b\,.
\end{aligned}
$$

Entscheiden Sie mit Hilfe des Cocke-Younger-Kasami-Algorithmus, ob die Wörter $aabaa$ und $abab$ in der Sprache $L(G)$ enthalten sind! Geben Sie gegebenenfalls Ableitungen an!

Lösung

Zur Entscheidung der Ableitbarkeit eines Wortes $w = a_1 a_2 \ldots a_n \in \{a, b\}^+$ berechnet der CYK-Algorithmus die Mengen $V_{i,j} = \{A \in V\,;\ A \to^* a_i \ldots a_j\}$ wie folgt.

$$
\begin{aligned}
V_{i,i} &= \{A \in V\,;\ A \to a_i \in P\}\\
\text{und für } i < j \quad V_{i,j} &= \bigcup_{i \le k < j} \{A \in V\,;\ (A \to BC) \in P \ \wedge\ B \in V_{i,k} \ \wedge\ C \in V_{k+1,j}\}\,.
\end{aligned}
$$

Die Frage ist dann $S \in V_{1,n}$?
Berechnungstabelle (in üblicher Matrixdarstellung) für die Mengen V_{ij} mit $w = aabaa$:

	1	2	3	4	5
1	A	S, Y, Z	S, Z, X, C_1, C_3	$\emptyset$	$S, \ldots$
2		A	S, Z	$\emptyset$	S, Z
3			B	$\emptyset$	X, C_2, C_3
4				A	S, Y, Z
5					A

Das Wort `aabaa` ist also in der Sprache enthalten.

Ableitung:

$S(1,5) \to Y(1,2)X(3,5) \to A(1,1)A(2,2)X(3,5) \to AAB(3,3)Z(4,5)$
$\to AABA(4,4)A(5,5) \to_G{}^* aabaa\,.$
Die Indizes bezeichnen die Menge $V_{i,j}$, in der die betreffende Variable berechnet wurde, und sind natürlich nicht Teil der Sprache.

Berechnungstabelle für $w = abab$:

	1	2	3	4
1	A	S, Z	$\emptyset$	$S, \ldots$
2		B	$\emptyset$	X, C_2, C_3
3			A	S, Z
4				B

Das Wort `abab` ist also auch in der Sprache enthalten.

Ableitung: $S \to A(1,1)X(2,4) \to AB(2,2)Z(3,4) \to ABA(3,3)B(4,4) \to_G{}^* abab\,.$

Tutoraufgabe 2 (Produktive Regeln)

Sei G eine kontextfreie Grammatik in CNF, die nur nützliche Symbole enthält. Man zeige: $L(G)$ ist genau dann unendlich, wenn G einen Zyklus enthält, d. h. ein Nichtterminal A, so dass $A \underset{G}{\rightarrow}{}^n \alpha A \beta$ mit $n > 0$ gilt.

Lösung

$\Leftarrow$:

Sei $A \underset{G}{\rightarrow}{}^* \alpha A \beta$ ein Zyklus.
Da alle Variablen von G nützlich sind, gibt es $s, t \in \Sigma^*$, so dass $\alpha A \beta \underset{G}{\rightarrow}{}^* sAt$, mithin gilt $A \underset{G}{\rightarrow}{}^* sAt$.
n-fache Anwendung des letzten Zyklus ergibt $A \underset{G}{\rightarrow}{}^* s^n A t^n$.
Da A nützlich ist, gilt $S \underset{G}{\rightarrow}{}^* uAv$ mit $u, v \in \Sigma^*$.
Nun folgt für beliebiges n

$$S \underset{G}{\rightarrow}{}^* uAv \underset{G}{\rightarrow}{}^* us^n A t^n v$$

Da A nützlich ist, gibt es ein $w \in \Sigma^*$, so dass $A \underset{G}{\rightarrow}{}^* w$. Daraus folgt für alle n

$$S \underset{G}{\rightarrow}{}^* us^n w t^n v \in L(G)\,.$$

$\Rightarrow$:

Ohne Zyklus hat ein Ableitungsbaum endliche Tiefe $\leq |V|$, da auf einem Pfad jede Variable nur einmal vorkommen kann. Entsprechend sind nur endlich viele Wörter ableitbar.

Tutoraufgabe 3 (Pumping-Lemma für CFL)

1. Zeigen Sie, dass $L = \{a^n b^m c^k\,;\, 0 < n < m < k\}$ keine kontextfreie Sprache ist.
2. Sei $f : \mathbb{N} \to \mathbb{N}$ eine beliebige Funktion und es sei $L = \{a^n b^{f(n)} c^n\,;\, n \in \mathbb{N}\}$.

   Beweisen Sie:

   Falls L kontextfrei ist, dann ist f beschränkt, d. h., dann gilt

   $$(\exists k \in \mathbb{N}\ \forall n \in \mathbb{N})\,[f(n) \leq k]\,.$$

Lösung

1. Wir nehmen an, dass L kontextfrei ist und führen diese Annahme zum Widerspruch.

   Sei $n \geq 1$ eine Pumping-Lemma-Zahl für die kontextfreie Sprache L. Sei zusätzlich $z = a^n b^{n+1} c^{n+2}$, d.h., $z \in L$ und $|z| \geq n$. Gemäß Pumping-Lemma gibt es dann also eine Zerlegung $z = uvwxy$ mit Wörtern $u, v, w, x, y \in \{a, b, c\}^*$ und den folgenden Eigenschaften:

   $$(1)\ vx \neq \epsilon \qquad (2)\ |vwx| \leq n \qquad (3)\ \forall i \in \mathbb{N}_0.\ uv^i wx^i y \in L$$

   Aufgrund von (1) unterscheiden wir die folgenden Fälle:

   - $\#_a(vx) > 0$: Wegen (2) gilt $\#_c(vx) = 0$. Allerdings gilt auch:

     $$\#_a\left(uv^3wx^3y\right) = n + 2\#_a(vx) \geq n + 2 = \#_c\left(uv^3wx^3y\right)$$

     Dies ist ein Widerspruch zu (3).
   - $\#_a(vx) = 0$ und $\#_b(vx) > 0$: Dann gilt:

     $$\#_b\left(uv^0wx^0y\right) = n + 1 - \#_b(vx) \leq n = \#_a\left(uv^0wx^0y\right)$$

     Daher ist $uv^0wx^0y \notin L$, ein Widerspruch zu (3).
   - $\#_a(vx) = 0$ und $\#_b(vx) = 0$: Dann muss $\#_c(vx) > 0$ gelten, und es folgt:

     $$\#_c\left(uv^0wx^0y\right) = n + 2 - \#_c(vx) \leq n + 1 = \#_b\left(uv^0wx^0y\right)$$

     Daher ist $uv^0wx^0y \notin L$, ein Widerspruch zu (3).

   Da jeder Fall zu einem Widerspruch führt und die obigen Fälle alle möglichen Zerlegungen abdecken, kann die ursprüngliche Annahme nicht gelten. Also ist L nicht kontextfrei.

2. Sei L kontextfrei. Wir führen die Annahme zum Widerspruch, dass f nicht beschränkt ist. Sei also f nicht beschränkt.

   Sei n eine Pumping-Lemma-Zahl für kontextfreie Sprachen zu L. Da f nicht beschränkt ist, können wir ein m annehmen mit $f(m) > n$.

   Sei $z = a^m b^{f(m)} c^m$. Es gilt $z \in L$ und $|z| \geq n$. Nach Pumping Lemma können wir eine Zerlegung $uvwxy$ von z annehmen, d. h. $z = uvwxy$, so dass $|vx| \geq 1$, $|vwx| \leq n$ und $uv^iwx^iy \in L$ für alle $i \geq 0$ gilt.

   Da $|b^{f(m)}| > n$ gilt, enthalten v und x beide kein a oder beide kein c.

   Fall 1: v und x enthalten beide kein c:
   Falls v und x nur aus b's bestehen, dann ist wegen $|vx| > 0$ die Anzahl der b in uv^2wx^2y größer als die Anzahl der b in $uvwxy$, während die Anzahl der c unverändert geblieben ist. Es folgt $\#_b(uv^2wx^2y) \neq f(m)$, mithin $uv^2wx^2y \notin L$. Widerspruch!

   Falls v ein a enthält, dann kann in uv^2wx^2y die Anzahl der a nicht gleich der Anzahl der c sein. Widerspruch!

   Fall 2: v und x enthalten beide kein a: Analog zu Fall 1.