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

Sommersemester 2015
Lösungsblatt 7
15. Juni 2015

Theoretische Informatik

Hausaufgabe 1 (5 Punkte)

Für alle $k \in \mathbb{N}$ definieren wir die Sprache $L_k = \{(ab^k)^m \,;\, m \in \mathbb{N}\}$.
(Beispiel: $L_2 = \{(abb)^m \,;\, m \in \mathbb{N}\}$)

1. Zeigen Sie für alle $k \in \mathbb{N}$ durch Angabe einer rechtslinearen Grammatik für L_k, dass L_k regulär ist.
2. Zeigen Sie, dass die Sprache $L = \bigcup_{k \in \mathbb{N}} L_k$ nicht kontextfrei ist.
   Hinweis: Sei n eine Pumping-Lemma-Zahl für L. Man betrachte $(ab^n)^3$.

Lösung

1.
$$\begin{array}{rcl} S & \to & aB_0\,, \\ B_0 & \to & bB_1\,, \ldots\,,\; B_{k-1} \to bB_k\,, \\ B_k & \to & aB_0 \mid \epsilon\,. \end{array}$$

(2P)

2. Sei n eine Pumping-Lemma-Zahl für L. Wir betrachten $z = (ab^n)^3$.

   Es gilt $z \in L$ und $|z| \geq n$.

   Seien $z = uvwxy$ mit $|vx| > 0$ und $|vwx| \leq n$, so dass für alle $i \in \mathbb{N}_0$ gilt $uv^iwx^iy \in L$. Wir definieren $z_i := uv^iwx^iy$.

   Zunächst gilt $z = t_1t_2t_3$ mit $t_i = ab^n$. Da $|vwx| \leq n$ gilt, ist vwx entweder Teilwort von t_1t_2 oder Teilwort von t_2t_3. Daraus folgt, dass entweder u das Teilwort ab^n am Anfang oder y das Teilwort ab^n am Ende enthält.

   Daraus folgt, dass $z_i = uv^iwx^iy \in L$ für jedes i die Gestalt $(ab^n)^m$ haben muss, d.h. es gibt für jedes i ein m, so dass $uv^iwx^iy = (ab^n)^m$ gilt. A.a. befinden sich zwischen zwei benachbarten a's genau n b's.

   $|vwx|$ enthält höchstens ein a.

   Falls vx ein a enthält, dann gilt für z_0, dass $\#_a(z_0) = 2$. Dann aber müsste gelten $z_0 = (ab^n)^2$ mit $\#_b(z_0) = 2n$.

   Wegen $\#_b(vx) \leq n-1$ folgt $\#_b(z) \leq 2n+(n-1) < 3n$ im Widerspruch zu $z = (ab^n)^3$.

   Falls vx nur b's enthält, und zwar mindestens ein b, dann führt Aufpumpen zum Widerspruch, weil sich zwischen zwei benachbarten a's genau n b's befinden.

(3P)

Hausaufgabe 2 (5 Punkte)

Sei $\Sigma = \{0, 1\}$. Die zwei Operationen Spiegelung (w^R) und Negation ($\overline{w}$) seien für $w \in \Sigma^*$ wie folgt definiert:

$$w^R = \begin{cases} \epsilon, & \text{falls } w = \epsilon \\ u^R a, & \text{falls } w = au \text{ für } a \in \Sigma \text{ und } u \in \Sigma^* \end{cases}$$

$$\overline{w} = \begin{cases} \epsilon, & \text{falls } w = \epsilon \\ \hat{a}\overline{u}, & \text{falls } w = au \text{ für } a \in \Sigma \text{ und } u \in \Sigma^* \end{cases}$$

Dabei setzen wir $\hat{0} = 1$ und $\hat{1} = 0$. Wie man leicht (etwa per Induktion) zeigen kann, gelten für diese Operationen auch die Gleichungen $(ua)^R = au^R$ und $\overline{ua} = \overline{u}\hat{a}$ für alle $a \in \Sigma$, $u \in \Sigma^*$. Im Folgenden nehmen wir diese Identitäten als bewiesen an.
Wir betrachten nun die Sprache $L = \{w \in \Sigma^* \,;\, w^R = \overline{w}\}$ und die Grammatik

$$G = (\{S\}, \Sigma, \{S \to 0S1 \mid 1S0 \mid \epsilon\}, S).$$

Zeigen Sie: L ist genau die von der Grammatik G beschriebene Sprache.

Lösung

Wie in der ZÜ gezeigt, gehört zu der Grammatik G auch eine induktive Definition von $L(G)$ als kleinste Menge, die die folgenden Eigenschaften aufweist:

$$\epsilon \in L(G) \qquad w \in L(G) \Rightarrow 0w1 \in L(G) \qquad w \in L(G) \Rightarrow 1w0 \in L(G)$$

Daraus ergibt sich auch das folgende Induktionsprinzip über die Ableitung eines Wortes w in G: Sei Q eine Eigenschaft von Wörtern aus Σ^*. Gelten $Q(\epsilon)$ und für alle Wörter $w \in \Sigma^*$ sowohl $Q(w) \Rightarrow Q(0w1)$ als auch $Q(w) \Rightarrow Q(1w0)$, dann gilt $Q(w)$ für alle $w \in L(G)$.

Wir müssen $L = L(G)$ zeigen. Dazu zeigen wir die Aussagen $w \in L(G) \Rightarrow w \in L$ und $w \in L \Rightarrow w \in L(G)$ für alle $w \in \Sigma^*$.

1. $w \in L(G) \Rightarrow w \in L$. Wir beweisen diesen Fall per Induktion über die Ableitung von w in G. Dabei ist $Q(w)$ die Eigenschaft „$w \in L$“.

   - $\underline{w = \epsilon}$: Es gilt $\epsilon^R = \epsilon = \overline{\epsilon}$ und damit $\epsilon \in L$.
   - $\underline{w \in L \Rightarrow 0w1 \in L}$: Es gilt

   $$(0w1)^R = (w1)^R 0 = 1w^R 0 \overset{w \in L}{=} 1\overline{w}0 = \hat{0}\overline{w}\hat{1} = \overline{0w}\hat{1} = \overline{0w1}$$

     und damit $0w1 \in L$.
   - $\underline{w \in L \Rightarrow 1w0 \in L}$: Dieser Fall wird vollkommen analog zum vorherigen Fall bewiesen, wobei 0 und 1 vertauscht werden.

2. $w \in L \Rightarrow w \in L(G)$. Beweis per Induktion über die Länge von w.

   - $\underline{|w| = 0}$: Dann ist $w = \epsilon$ und damit $w \in L(G)$.
   - $\underline{|w| = 1}$: Dann ist $w = w^R$, aber $w \neq \overline{w}$ (denn $\hat{0} \neq 0$ und $\hat{1} \neq 1$). Also ist $w \notin L$, und damit gilt auch $w \in L \Rightarrow w \in L(G)$.

- $|w| \geq 2$: Dann hat w die Form $w = aub$ für gewisse $a, b \in \Sigma$ und $u \in \Sigma^*$, und es gilt
$$bu^R a = (ub)^R a = (aub)^R \overset{w\in L}{=} \overline{aub} = \hat{a}\overline{ub} = \hat{a}\overline{u}\hat{b}$$
und damit $b = \hat{a}$ und $u^R = \overline{u}$. Da $|u| < |w|$ folgt $u \in L(G)$ nach Induktionshypothese. Weiterhin ist $w = aub = au\hat{a}$, also entweder $w = 0u1$ oder $w = 1u0$, und mit $u \in L(G)$ gilt dann $w \in L(G)$.

(5P)

Hausaufgabe 3 (5 Punkte)

Wir betrachten die Grammatik $G = (V, \{a, b, c, d\}, P, S)$ mit den Produktionen

$$\begin{array}{lll} S \to AZ\,, & X \to b \mid XB\,, & B \to b\,, \\ Z \to SD \mid TD\,, & Y \to c \mid YC\,, & C \to c\,, \\ T \to XY\,, & A \to a\,, & D \to d\,. \end{array}$$

1. Zeigen Sie durch Anwendung des CYK-Verfahrens, dass a^2bc^2d nicht in der von G erzeugten Sprache enthalten ist, d. h. $a^2bc^2d \notin L(G)$.

2. Geben Sie eine Ableitung des Wortes a^2bcd^2 mit Produktionen der Grammatik G an.

3. Zeigen Sie, dass die Sprache $L = \{a^k b^m c^k d^m\,;\, k, m \in \mathbb{N}\setminus\{0\}\} \subseteq \{a, b, c, d\}^*$ nicht kontextfrei ist.

4. Zeigen Sie, dass die Sprache $L = \{a^m b^m c^k d^k\,;\, k, m \in \mathbb{N}\setminus\{0\}\} \subseteq \{a, b, c, d\}^*$ kontextfrei ist. Ist L linear, d.h., kann sie von einer Grammatik mit linearen Produktionen erzeugt werden?

Lösung

1.

16 $\emptyset$					
15 $\emptyset$	26 S				
14 $\emptyset$	25 $\emptyset$	36 Z			
13 $\emptyset$	24 $\emptyset$	35 T	46 $\emptyset$		
12 $\emptyset$	23 $\emptyset$	34 T	45 Y	56 $\emptyset$	
11 A	22 A	33 B, X	44 C, Y	55 C, Y	66 D
a	a	b	c	c	d

(2P)

2. $S \to_G AZ \to_G ASD \to_G AAZD \to_G AATDD \to_G AAXYDD \to_G^* aabcdd\,.$ (1P)

3. Widerspruchsbeweis mit Pumping-Lemma:

   Sei n eine Pumping-Lemmazahl und $z = a^n b^n c^n d^n$.

   Es gilt $z \in L$. Damit gilt $z = uvwxy$ mit $|vwx| \leq n$ und $|vx| \geq 1$.

   Damit enthält vx entweder nicht c, d oder nicht a, d oder nicht a, b. vx enthält aber mindestens einen Buchstaben.

   Sei o.B.d.A. a in vx enthalten, und c, d seien nicht in vx enthalten.

   Nach Pumping-Lemma gilt $z' = uv^0wx^0y \in L$. In z' ist aber die Anzahl der c gleich n, wohingegen die Anzahl der a nun kleiner als n ist. Widerspruch, denn damit gilt $z' \notin L$. (1P)

4. Die Sprache $\{a^m b^m c^n d^n\,;\, m, n \in \mathbb{N}\}$ ist kontextfrei, denn sie wird von der kontextfreien Grammatik $(\{S, A, B\}, \{a, b, c, d\}, P, S)$ mit den Produktionen $S \to AA \mid BB$ und $A \to aAb \mid \epsilon,\ B \to cBd \mid \epsilon$ erzeugt. (1P)

   Aber diese Sprache ist nicht linear: Intuitiv liegt das daran, dass bei einer Ableitung einer linearen Grammatik in jedem Schritt genau ein Nichtterminal produziert wird. Wenn ein Fragment $a^k b^l$ mit $k = l$ erzeugt werden soll, liegt das Nichtterminal zwangsläufig zwischen a und b. Daher ist es nicht möglich, die Bedingung $k = l$ für zwei nebeneinanderstehende Paare von as und bs zu erzwingen.

   Formal lässt sich das mit einem speziellen Pumping-Lemma für lineare Sprachen beweisen.

Hausaufgabe 4 (5 Punkte)

Seien $\Sigma \neq \emptyset$ und $V = \{A_1, A_2, \ldots, A_n\}$ Zeichenmengen mit $n \geq 2$ und m eine *Markierungsabbildung* der Form $m(x) = \hat{x}$ bzw. $m(A) = \widehat{A}$ für alle $x \in \Sigma$ bzw. $A \in V$. Wir definieren $\widehat{\Sigma} = \{\hat{x}\,;\, x \in \Sigma\}$ und $\widehat{V} = \{\widehat{A}_1, \widehat{A}_2, \ldots, \widehat{A}_n\}$. Wir setzen Mengendisjunktheit voraus, so dass $|\Sigma \cup \widehat{\Sigma} \cup V \cup \widehat{V}| = 2(n + |\Sigma|)$ gilt, und definieren $\Sigma' = \Sigma \cup \widehat{\Sigma}$ und $V' = V \cup \widehat{V}$.

Wir sagen, dass eine kontextfreie Grammatik $G' = (V', \Sigma', P', S')$ eine *Wortendemarkierung* generiert, falls S' eines der markierten Zeichen $\widehat{A}_i$, $i = 1, \ldots, n$ ist und jede Produktion aus P' eine der folgenden Formen besitzt (mit $x \in \Sigma$):

$$\begin{aligned} A_i &\to A_j A_k\,, & A_i &\to x\,,\\ \widehat{A}_i &\to A_j \widehat{A}_k\,, & \widehat{A}_i &\to \hat{x}\,. \end{aligned}$$

1. Sei G' eine kontextfreie Grammatik, die eine Wortendemarkierung generiert. Man zeige mit <u>struktureller Induktion</u> für alle Wörter w der Sprache $L(G')$ die folgende Eigenschaft

   $$\widehat{P}(w): \quad \text{Es gibt ein } v \in \Sigma^* \text{ und ein } \hat{x} \in \widehat{\Sigma}, \text{ so dass } \ w = v\hat{x} \ \text{ gilt.}$$

   Betrachten Sie dazu geeignete Eigenschaften $P(w)$ bzw. $\widehat{P}(w)$ der aus Variablen $A \in V$ einerseits bzw. $\widehat{A} \in \widehat{V}$ andererseits ableitbaren Wörter $w \in \Sigma'^*$. Verwenden Sie die Bezeichnung $L(X) = \{w \in \Sigma'^*\,;\, X \xrightarrow[G']{}^* w\}$ für $X \in V'$.

2. Seien L eine kontextfreie Sprache, so dass $\epsilon \notin L$, und $E = \{x \in \Sigma^*\,;\, |x| = 1\}$.

   Zeigen Sie, dass der Rechtsquotient L/E kontextfrei ist. Zum Nachweis genügt eine informelle Konstruktionsbeschreibung einer kontextfreien Grammatik für L/E.

Lösung

1. Sei $P(w)$ die Eigenschaft: Es gilt $w \in \Sigma^*$.

   Induktionsanfang:
   Regel $A_i \to x$: Für $w = x$ gilt $P(w)$. (Klar!)
   Regel $\widehat{A}_i \to \hat{x}$: Für $w = \hat{x}$ gilt $\widehat{P}(w)$. (Klar!)

   Induktionsschluss:
   Regel $A_i \to A_j A_k$: Aus $w_j \in L(A_j) \wedge P(w_j)$ und $w_k \in L(A_k) \wedge P(w_k)$ folgt $w_i = w_j w_k \in L(A_i) \wedge P(w_i)$.
   Beweis: $w_j, w_k \in \Sigma^*$ impliziert $w_i = w_j w_k \in \Sigma^*$. Es folgt $P(w_i)$.

   Regel $\widehat{A}_i \to A_j \widehat{A}_k$: Aus $w_j \in L(A_j) \wedge P(w_j)$ und $w_k \in L(\widehat{A}_k) \wedge \widehat{P}(w_k)$ folgt $w_i = w_j w_k \in L(\widehat{A}_i) \wedge \widehat{P}(w_i)$.
   Beweis: $w_j \in \Sigma^*$ und $w_k = v\hat{x}$ impliziert $w_i = w_j w_k = w_j v \hat{x}$ mit $w_j v \in \Sigma^*$ und $\hat{x} \in \widehat{\Sigma}$. Es folgt $\widehat{P}(w_i)$.

   Da sich jedes Wort $w \in L(G')$ aus $S' \in \widehat{V}$ ableiten lässt, folgt $\widehat{P}(w)$. (3P)

2. Wenn in allen Wörtern $w \in L$ der letzte Buchstabe gestrichen wird, dann erhält man L/E. Wir konstruieren eine kontextfreie Grammatik, die L/E erzeugt, wie folgt:

   Sei G eine kontextfreie Grammatik mit $L = L(G)$ in Chomsky-Normalform. Wir ergänzen G zu einer Grammatik G', die eine Wortendemarkierung generiert, d.h., dass alle letzten Buchstaben von Wörtern w in L markiert werden.

   Dies geschieht durch

   Hinzufügen von $\widehat{\Sigma}$, $\widehat{V}$,
   Ersetzung von S durch $\widehat{S}$,
   Hinzufügen von Produktionen $\widehat{A}_i \to A_j \widehat{A}_k$ zu jeder Produktion $A_i \to A_j A_k$ und
   Hinzufügen von Produktionen $\widehat{A}_i \to \hat{x}$ zu jeder Produktion $A_i \to x$.

   Um die Grammatik für L/E zu gewinnen, werden alle Produktionen der Form $\widehat{A}_i \to \hat{x}$ von G' ersetzt durch $\widehat{A}_i \to \epsilon$. Dadurch entfällt $\widehat{\Sigma}$.

   Die erhaltene Grammatik G'' erzeugt L/E, wobei sich G'' nach Satz der Vorlesung durch Elimination der ϵ-Produktionen in eine kontextfreie Grammatik umwandeln lässt.

   (2P)

__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

Überführen Sie die folgende Grammatik in Greibach-Normalform:

$$G = (\{S, X\}, \{a, b\}, \{S \to XX, S \to a, X \to SS, X \to b\}, X)\,.$$

Lösung

Wir stellen zunächst fest, dass G eine kontextfreie Grammatik in Chomsky-Normalform ist. X ist das Axiom von G und $\epsilon \notin L(G)$. Offenbar enthält G keine nutzlosen Variablen. Wir wenden das im Beweis von Satz 84 der Vorlesung gegebene Verfahren zur Konstruktion der Greibach-Normalform an.

Vorab nummerieren wir die Variablen, indem wir sie (willkürlich) als A_1, A_2 umbezeichnen. Die Variable S soll jetzt dem A_1 und X dem A_2 entsprechen. Eventuell notwendig werdende zusätzliche Variablen werden dann mit $A_3, \ldots$ bezeichnet. Die Anzahl der Variablen ist also zunächst $m = m_0 = 2$. Der Wert von m wird dann nach Bedarf erhöht.

Im ersten Schritt der Konstruktion wird eine Schleife von $k = 1, 2, \ldots, m$ durchlaufen, die sicherstellt, dass für alle Produktionen der Form $A_i \to A_j\alpha$ mit $1 \le i, j \le m$ stets $i < j$ gilt. Damit gilt dann bereits, dass jede A_m-Produktion mit einem Terminalzeichen beginnt. Im zweiten Schritt der Konstruktion wenden wir Lemma 82 beginnend mit A_{m_0-1} genügend oft an, um zu erreichen, dass jede Produktion mit einem Terminalzeichen beginnt.

1. Schritt:
Wir starten mit den Produktionen

A_1-Produktionen:	A_1	$\to$	$A_2A_2 \mid a$
A_2-Produktionen:	A_2	$\to$	$A_1A_1 \mid b$

Die A_1-Produktionen sind alle in der gewünschten Form, d. h. für $k = 1$ ist nichts zu tun und wir gehen sofort zu $k = 2$ über.
Die A_2-Produktionen sind nicht alle in der gewünschten Form. Die erste der beiden Produktionen muss nach Lemma 82 ersetzt werden durch alle Produktionen, die man durch Anwendung der A_1-Produktionen auf die erste A_1 Variable in $A_2 \to A_1A_1$ erhält. Dies ergibt die neue Produktionenmenge

A_1-Produktionen:	A_1	$\to$	$A_2A_2 \mid a$
A_2-Produktionen:	A_2	$\to$	$A_2A_2A_1 \mid aA_1 \mid b$

Nun ist die linksrekursive A_2-Produktion mit Lemma 82 zu eliminieren. Um zu verdeutlichen, wie dies geschieht, führen wir die entsprechenden Bezeichnungen ein, also $\alpha_1 = A_2A_1$ und $\beta_1 = aA_1, \beta_2 = b$. Die laut Lemma 82 neu einzuführende Variable B bezeichnen wir mit A_3, wobei wir m inkrementieren, also $m = 3$ setzen. Die Linksrekursion ist nun zu ersetzen durch

$$\begin{array}{lcl} A_2 & \to & \beta_1 A_3 \mid \beta_2 A_3 \\ A_3 & \to & \alpha_1 \mid \alpha_1 A_3 \end{array}$$

Als Ergebnis für $k = 2$ erhalten wir

A_1-Produktionen:	A_1	$\rightarrow$	$A_2A_2 \mid a$
A_2-Produktionen:	A_2	$\rightarrow$	$aA_1A_3 \mid bA_3 \mid aA_1 \mid b$
	A_3	$\rightarrow$	$A_2A_1 \mid A_2A_1A_3$

Nun setzen wir $k = 3$. Die A_3-Produktionen sind nicht in der gewünschten Form. Die zwei Produktionen müssen nach Lemma 82 ersetzt werden durch die Menge der Produktionen, die man durch Anwendung der A_2-Produktionen auf die erste Variable in den A_3-Produktionen erhält. Dies ergibt für $k = 3$ die neue Produktionenmenge

A_1-Produktionen:	A_1	$\rightarrow$	$A_2A_2 \mid a$
A_2-Produktionen:	A_2	$\rightarrow$	$aA_1A_3 \mid bA_3 \mid aA_1 \mid b$
A_3-Produktionen:	A_3	$\rightarrow$	$aA_1A_3A_1 \mid bA_3A_1 \mid aA_1A_1 \mid bA_1 \mid$
			$aA_1A_3A_1A_3 \mid bA_3A_1A_3 \mid aA_1A_1A_3 \mid bA_1A_3$

Man beachte, dass für die neuen Variablen, wie z. B. A_3, keine linksrekursiven Produktionen vorhanden sein können. Deshalb sind wir bereits mit dem ersten Schritt fertig.

2. Schritt:
Man beachte, dass alle A_k-Produktionen mit $k \geq m_0$ notwendigerweise bereits mit einem Terminalzeichen beginnen. Es bleibt in unserem Fall also lediglich die Anwendung von Lemma 82 auf die A_1 Produktionen, hier also auf $A_1 \rightarrow A_2A_2$, wobei alle A_2-Produktionen auf das erste A_2-Vorkommen angewendet werden müssen. Wir erhalten als Menge der A_1-Produktionen

A_1-Produktionen:	A_1	$\rightarrow$	$aA_1A_3A_2 \mid bA_3A_2 \mid aA_1A_2 \mid bA_2 \mid a$

Diese bilden zusammen mit den A_2- und A_3-Produktionen aus dem 1. Schritt das Endergebnis.

Vorbereitung 2

Sei $K = (Q, \Sigma, \Delta, q_0, Z_0, F, \delta)$ ein Kellerautomat mit Startzustand $q_0 \in Q$, Startkellerzeichen $Z_0 \in \Delta$, Menge $F \subseteq Q$ von akzeptierenden Zuständen und der Übergangsfunktion δ. Eine Folge $(p_0, w_0, \gamma_0), (p_1, w_1, \gamma_1), \ldots, (p_k, w_k, \gamma_k)$ mit nicht leerem γ_0 heiße Berechnung der Konfiguration (p_0, w_0, γ_0) mit $k \in \mathbb{N}_0$ Schritten, falls gilt

$$(p_0, w_0, \gamma_0) \rightarrow (p_1, w_1, \gamma_1) \rightarrow \ \ldots \ \rightarrow (p_k, w_k, \gamma_k)\,.$$

Falls für $c = (p, w, \gamma)$ keine Berechnung mit $k > 0$ Schritten existiert, dann nennen wir c eine Endkonfiguration

Wir nehmen nun an, dass K ein deterministischer Kellerautomat in Normalform ist. Man zeige:

1. Für alle $w \in \Sigma^*$ gibt es genau eine Berechnung $(q_0, w, Z_0), (p_1, w_1, \gamma_1), \ldots, (p_k, \epsilon, \gamma_k)$, so dass $(p_k, \epsilon, \gamma_k)$ eine Endkonfiguration ist. Für diese Berechnung gilt $\gamma_i \neq \lambda$ mit leerem Wort $\lambda \in \Delta^*$.

2. Es gibt eindeutige Funktionen $\sigma : \Sigma^* \rightarrow \mathbb{N}_0$, $\eta : \Sigma^* \rightarrow Q$ und $\kappa : \Sigma^* \rightarrow \Delta^+$, so dass für alle $w \in \Sigma^*$ gilt

   $(\eta(w), \epsilon, \kappa(w))$ ist Endkonfiguration einer Berechnung von (q_0, w, Z_0) mit $\sigma(w)$ Schritten.

Lösung

Erinnerung: Für die Relation der direkten Berechnung $\rightarrow$ in einem Schritt über der Menge der Konfigurationen gilt $(p, u, \gamma) \rightarrow (q, u', \gamma')$ genau dann, wenn $u = au'$ mit $a \in \Sigma \cup \{\epsilon\}$, $\gamma = Z\alpha$ mit $Z \in \Delta$, $\gamma' = \beta\alpha$ und $(q, \beta) \in \delta(p, a, Z)$ gilt. Dabei ist $u \in \Sigma^*$ und $\alpha, \beta, \gamma' \in \Delta^*$. Die transitive Hülle von $\rightarrow$ wird mit $\rightarrow^*$ notiert.

1. Sei (q_0, w, Z_0) eine Endkonfiguration. Dann muss $w = \epsilon$ und $\delta(q_0, \epsilon, Z_0) = \emptyset$ gelten. Wenn $w = aw'$ mit $a \in \Sigma$ gelten würde, dann würde K das Zeichen a oder ϵ lesen, weil K in Normalform ist. Wenn $\delta(q_0, \epsilon, Z_0) \neq \emptyset$ wäre, dann gäbe es einen weiteren Berechnungsschritt.

   Entsprechend sei $(q_0, w, Z_0), (p_1, w_1, \gamma_1), \ldots, (p_i, w_i, \gamma_i)$ eine eindeutige Berechnung von (q_0, w, Z_0) mit i Schritten. Falls (p_i, w_i, γ_i) keine Endkonfiguration ist, dann gibt es in K eine eindeutige Fortsetzung der Berechnung mit $i + 1$ Schritten. Falls (p_i, w_i, γ_i) eine Endkonfiguration ist, dann gilt $w_i = \epsilon$ und $\delta(p_i, \epsilon, \gamma_i) = \emptyset$ (Halt durch Ende der Eingabe) mit analogem Beweis wie im Fall (q_0, w, Z_0). In diesem Fall folgt nun $\gamma \neq \lambda$, da sonst die Konfiguration (q_0, wa, Z_0) mit $a \in \Sigma$ nicht berechenbar wäre im Widerspruch zur Normalform von K.

2. Da bei Normalform keine unendliche Berechnung existiert, gibt es für alle Startkonfigurationen (q_0, w, Z_0) genau eine Zahl $k \in \mathbb{N}_0$, so dass die Berechnung im k-ten Schritt mit Endkonfigiration (p_k, w_k, γ_k). Wie oben bewiesen folgt $w_k = \epsilon$ und $\gamma_k \neq \lambda$ endet.

   Wir setzen $\sigma(w) = k$, $\eta(w) = p_k$ und $\kappa(w) = \gamma_k$.

Vorbereitung 3

Man beweise die folgende Aussage:

Für alle deterministischen kontextfreien Sprachen L gilt, dass es genau dann einen deterministischen Kellerautomaten gibt, der L mit leerem Keller akzeptiert, wenn L die Präfixbedingung erfüllt.

Lösung

$\Rightarrow$:
Sei K ein deterministischer Kellerautomat, der L mit leerem Keller akzeptiert. Wir zeigen die Präfixbedingung durch Widerspruch.

Seien $u, uv \in L$ mit $v \neq \epsilon$. Nach Eingabe von uv durchläuft der Automat zunächst alle Konfigurationen, die durch Eingabe von u veranlasst werden. Da K deterministisch ist und u durch leeren Keller akzeptiert wird, ist nun der Keller leer. Eine weitere Eingabe von $v \neq \epsilon$ wir nicht mehr verarbeitet, weil die Übergangsfunktion nicht definiert ist, wenn der Keller leer ist.

$\Leftarrow$:
Wir modifizieren den Beweis des Satzes 88 der Vorlesung.
Sei $K = (Q, \Sigma, \Delta, q_0, Z_0, \delta, F)$ ein deterministischer Kellerautomat, der L mit Endzustand akzeptiert. L erfülle die Präfixbedingung. Wir konstruieren einen DPDA $K' = (Q', \Sigma, \Delta', q_0', Z_0', \delta')$, der L mit leerem Keller akzeptiert, so dass K' den Keller K simuliert und bei Erreichen eines Endzustands den Keller leert, wie folgt.

Seien $Q' = Q \cup \{\bar{q}, q'_0\}$, $\Delta' = \Delta \cup \{Z'_0\}$. Wir definieren

$\delta'(q'_0, \epsilon, Z'_0) = (q_0, Z_0 Z'_0)$,
$\delta'(q, \epsilon, Z) = (\bar{q}, \epsilon)$ für alle $q \in F$, $Z \in \Delta'$,
$\delta'(\bar{q}, \epsilon, Z) = (\bar{q}, \epsilon)$ für alle $Z \in \Delta'$ und
$\delta'(q, a, Z) = \delta(q, a, Z)$ für alle $q \in Q \setminus F$, $a \in \Sigma$, $Z \in \Delta$.

Offenbar ist K' deterministisch. Die Präfixbedingung für L bewirkt, dass K und K' die gleiche Sprache akzeptieren.

Tutoraufgabe 1

Kellerautomaten in ihrer einfachsten Form haben nur einen einzigen Zustand und akzeptieren mit leerem Keller. Man kann in diesem Fall auf Zustände sogar ganz verzichten. Sie haben auch keine sogenannten ϵ-Übergänge. Wir bezeichnen solche Kellerautomaten als einfache Kellerautomaten $E = (\Sigma, \Delta, Z_0, \delta)$, abgekürzt EKA. Entsprechend ist die Funktionalität von δ nun $\delta : \Sigma \times \Delta \to \mathcal{P}_f(\Delta^*)$, wobei $\mathcal{P}_f(\Delta^*)$ die Menge aller endlichen Teilmengen von Δ^* bedeutet.

1. Zeigen Sie durch Anwendung bzw. Modifikation von Sätzen der Vorlesung, dass es für jede kontextfreie Sprache L mit $\epsilon \notin L$ einen einfachen Kellerautomaten $E = (\Sigma, \Delta, Z_0, \delta)$ gibt, der die Sprache L akzeptiert, d.h. $L = L(E)$.

2. Ein EKA E ist deterministisch, falls $|\delta(a, Z)| \leq 1$ für alle $a \in \Sigma, Z \in \Delta$ gilt.

   Zeigen Sie, dass $L = \{a^n b^n \,;\, n \in \mathbb{N}\}$ von einem deterministischen EKA erzeugt werden kann.

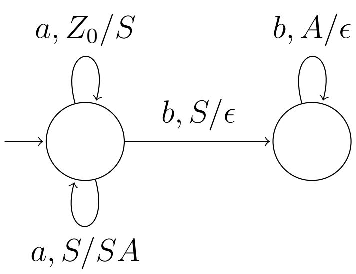
Lösung

1. Sei L von einer Greibach–Normalform–Grammatik (V, Σ, P, S) erzeugt (Satz 83 der Vorlesung). Wir konstruieren $E = (\Sigma, \Delta, Z_0, \delta)$ analog wie in Satz 90.

   Seien $\Delta = V$, $Z_0 = S$ und δ wie folgt für alle $a \in \Sigma$ und $A \in \Delta$ definiert:

   $$\delta(a, A) \ni \alpha \quad \text{für} \quad (A \to a\alpha) \in P\,.$$

2. Die graphische Darstellung der Übergangsfunktion δ von $E = (\Sigma, \Delta, Z_0, \delta)$ benötigt keine Bezeichnung von Zuständen und kann übersichtlich wie folgt dargestellt werden.



   Entsprechend gelten $\delta(a, Z_0) = S$, $\delta(a, S) = SA$, $\delta(b, S) = \epsilon$ und $\delta(b, A) = \epsilon$.

Tutoraufgabe 2

Sei $A = (Q, \Sigma, \Delta, q_0, Z_0, \delta, F)$ ein deterministischer Kellerautomat. Dann nennen wir einen Zustand $q \in Q$ spontan bzw. stabil, wenn $|\delta(q, a, X)| = 0$ für alle $a \in \Sigma, X \in \Delta$ gilt bzw. $|\delta(q, \epsilon, X)| = 0$ für alle $X \in \Delta$ gilt. Die Mengen der spontanen bzw. stabilen Zustände bezeichnen wir mit Q_{sp} bzw. Q_{st}. A nennen wir ϵ-separiert, falls $Q = Q_{sp} \cup Q_{st}$ gilt.

Nach Konstruktion in der Vorlesung gibt es zu A einen äquivalenten ϵ-separierten Kellerautomaten $K = (Q, \Sigma, \Delta, q_0, Z_0, \delta, F)$ in Normalform. Man beweise nun unter Berücksichtigung der Definitionen in VA 2 die folgenden Aussagen über Kellerautomaten K in Normalform.

1. Sei $\beta = (q_0, w, Z_0), (q_1, w_1, \gamma_1) \ \ldots \ (q_{n-1}, w_{n-1}, \gamma_{n-1}), (\eta(w), \epsilon, \kappa(w))$ eine Berechnung von (q_0, w, Z_0) und j die kleinste Zahl, so dass die Teilsequenz $\beta_j = (q_j, w_j, \gamma_j) \ \ldots \ (q_{n-1}, w_{n-1}, \gamma_{n-1}), (\eta(w), \epsilon, \kappa(w))$ von β nur spontane Zustände enthält mit Ausnahme von $\eta(w)$. Dann gilt:

   w wird von K akzeptiert genau dann, wenn mindestens einer der in β enthaltenen Zustände q_i ein akzeptierender Zustand aus F ist.

2. Zu jedem ϵ-separierter Kellerautomaten in Normalform K gibt es einen äquivalenten Kellerautomaten $K' = (Q', \Sigma, \Delta', q_0', Z_0', \delta', F')$, der keine spontanen akzeptierenden Zustände besitzt, d.h., dass alle akzeptierenden Zustände stabil sind.

3. K' ist in dem folgenden Sinn komplementierbar: Der komplementierte Kellerautomat $\overline{K} = (Q', \Sigma, \Delta', q_0', Z_0', \delta', {Q'}_{st} \setminus F')$ akzeptiert das Komplement von $L(K')$:

$$L(\overline{K}) = \overline{L(K')} = \Sigma^* \setminus L(K') \,.$$

Lösung

1. Da von spontanen Zuständen aus keine Eingabezeichen gelesen werden, gilt $w_i = \ldots = w_{n-1} = \epsilon$. Falls also ein Zustand q_i mit $j \le i \le n-1$ ein akzeptierender Zustand ist, dann wird w nach Definition akzeptiert. Insbesondere wird w akzeptiert, falls $\eta(w)$ akzeptierend ist.

   Umgekehrt muss eine Berechnung von q_0, w, Z_0 mit der Endkonfiguration $(\eta(w), \epsilon, \kappa(w))$ enden. Falls w akzeptiert wird, muss dann einer der Zustände q_i mit $j \le i \le n-1$ oder $\eta(w)$ selbst akzeptierend sein.

2. Die Konstruktion ist analog dem Beweis Teil 2. Gedächtniserweiterung von Satz 94 der Vorlesung.

   Wir definieren $Q' = Q \times \{0, 1\}$ und schreiben $q^{(0)}$ bzw. $q^{(1)}$ für $(q, 0)$ bzw. $(q, 1)$. Seien $q_0' = q_0^{(0)}$, $Z_0' = Z_0$, $\Delta' = \Delta$. Die Übergangsfunktion δ' ist wie folgt definiert:

   Seien $\delta(q, a, Z) = (p, \gamma)$ mit $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$. Es gilt

$$\delta'(q^{(0)}, a, Z) = \begin{cases} (p^{(1)}, \gamma) : q \text{ spontan}, \ q \in F \\ (p^{(0)}, \gamma) : \text{sonst} \end{cases} ,$$

$$\delta'(q^{(1)}, a, Z) = \begin{cases} (p^{(1)}, \gamma) : q \text{ spontan} \\ (p^{(0)}, \gamma) : q \text{ stabil} \end{cases} .$$

Falls $\delta(q, a, Z) = \emptyset$, dann wird $\delta'(q^{(0)}, a, Z) = \delta'(q^{(1)}, a, Z) = \emptyset$ gesetzt.

Nun wird die Menge der akzeptierenden Zustände wie folgt definiert:

$$F' = \{q^{(1)}\,;\, q \in Q_{st}\} \cup \{q^{(0)}\,;\, q \in Q_{st} \cap F\}$$

Die Äquivalenz von K und K' kann nun gezeigt werden.

3. Ähnlich wie in Satz 95 der Vorlesung

Tutoraufgabe 3

Wir betrachten die Grammatik $G = (V, \Sigma, P, S)$ mit den Produktionen

$$\begin{aligned} S &\rightarrow E\,, \\ E &\rightarrow E + T \mid T\,. \\ T &\rightarrow a \mid (E)\,. \end{aligned}$$

Ist G eine $LR(1)$ Grammatik? Begründen Sie Ihre Antwort.

Lösung

Die Grammatik erzeugt alle korrekt geklammerter arithmetischen (assoziativen) Mehrfachsummen von Konstanten a.

G ist eine $LR(1)$ Grammatik.
Man kann die folgende Lookahead-Tabelle für die Anwendung der Produktionen erstellen.

Produktionen	Lookaheads der Länge 0 oder 1
$S \rightarrow E$	ϵ d.h., E steht am Ende
sonstige	beliebig

Man beachte, dass beim Parsen von links nach rechts stets die am weitesten links ansetzbare Reduktion angewandt werden muss.

Dies bedeutet z.B., dass $T + a$ nicht zu $T + T$ reduziert werden darf, sondern es muss zu $E + a$ reduziert werden gefolgt von $E + T$. $E + T$ wiederum darf nicht zu $E + E$ reduziert werden, weil die Reduktion zu E weiter links ansetzbar ist.

Man kann zeigen, dass die Wörter von $L(G)$ mit obiger Lookahead-Tabelle eindeutig von links nach rechts analysiert werden können. Daraus folgt, dass G eine $LR(1)$ Grammatik ist.

Es gibt auch andere Lookahead-Tabellen, wie z.B.

Produktionen	Lookaheads der Länge 0 oder 1
$S \rightarrow E$	ϵ d.h., E steht am Ende
$E \rightarrow E + T$	$+,), \epsilon$
$E \rightarrow T$	$+,), \epsilon$
$T \rightarrow (E)$	$+,), \epsilon$
$T \rightarrow (E)$	$+,), \epsilon$