Institut für Informatik
M. Mutz

Übungsblatt 7

Theoretische Informatik

Bearbeitungszeit: 27.05.2024 bis 02.06.2024, 16:00 Uhr
Besprechung: 03.06.2024, 10:30 Uhr in Hörsaal 5E

Abgabe: als PDF über das ILIAS
Gruppenabgaben möglich und erwünscht

Aufgabe 1 (Kellerautomat I)10P

Gegeben sei der DPDA $M = (\{a, b\}, \{A, \#\}, \{z_0, z_1, z_2, z_e\}, \delta, z_0, \#, \{z_0, z_1, z_2\})$ und δ wie folgt:

$z_0 a\# \to z_1 \#$	$z_1 a\# \to z_1 A\#$	$z_1 aA \to z_1 AA$	
$z_1 b\# \to z_e \lambda$	$z_1 bA \to z_2 \lambda$	$z_2 b\# \to z_e \lambda$	$z_2 bA \to z_2 \lambda$

(a) Geben Sie für $w_1 = aaabb$ die Konfigurationenfolge an und entscheiden Sie anhand dessen, ob $w_1 \in L(M)$ gilt.

(b) Geben Sie für $w_2 = aabb$ die Konfigurationenfolge an und entscheiden Sie anhand dessen, ob $w_2 \in L(M)$ gilt.

(c) Geben Sie $L(M)$ formal als Menge von Wörtern an, ohne weiteren Bezug auf M zu nehmen. (Referenzieren Sie auch keine weiteren Konstrukte. Erstellen Sie einen eigenständigen formalen Mengenausdruck)

(d) Ist δ deterministisch?

(e) Ist δ total?

Hinweis: Beachten Sie das Tupel von M und lassen Sie sich nicht von Namen beirren.

Lösungsvorschlag:

(a) $(z_0, aaabb, \#) \vdash_M (z_1, aabb, \#) \vdash_M (z_1, abb, A\#) \vdash_M (z_1, bb, AA\#) \vdash_M (z_2, b, A\#)$
$\vdash_M (z_2, \lambda, \#)$
$z_2 \in F \Rightarrow aaabb \in L(M)$

(b) $(z_0, aabb, \#) \vdash_M (z_1, abb, \#) \vdash_M (z_1, bb, A\#) \vdash_M (z_2, b, \#) \vdash_M (z_e, \lambda, \lambda)$
$z_e \notin F \Rightarrow aabb \notin L(M)$

(c) $L(M) = \{a^n b^m \mid n = m = 0 \text{ oder } n > m \geq 0\}$

(d) δ ist deterministisch, da es sich um einen DPDA handelt.
(Außerdem gilt $|\delta(z, x, X)| \leq 1$, $\forall z \in Z, x \in \Sigma, X \in \Gamma$)

(e) δ ist nicht total, da z.B. $\delta(z_0, a, A)$ nicht definiert ist.

Aufgabe 2 (Kellerautomat II)14P
Konstruieren und geben Sie für die folgenden Sprachen einen deterministischen Kellerautomaten an:

a) $L_1 = \{0^n 1^m \mid m, n \geq 0, n \leq m\}$

b) $L_2 = \{0^n 1^m \mid m, n \geq 0, n \geq m\}$

Erklären Sie die Bedeutung der einzelnen Zustände der Automaten, die Sie konstruiert haben und warum Sie die jeweilige Sprache akzeptieren.

Lösungsvorschlag:

a) Der DPDA $M_a = (\{0,1\}, \{A, \#\}, \{z_0, z_1, z_2, z_3\}, \delta_a, z_0, \#, \{z_0, z_3\})$ mit δ_a:

$$\begin{array}{ccc} z_0 0\# \to z_1\# & z_1 0A \to z_1 AA & z_2 1A \to z_2 \lambda \\ z_0 1\# \to z_3\# & z_1 1A \to z_2 \lambda & z_2 1\# \to z_3\# \\ z_1 0\# \to z_1 A\# & z_1 1\# \to z_3\# & z_3 1\# \to z_3\# \end{array}$$

akzeptiert die Sprache $\{0^n 1^m \mid n \leq m, m, n \geq 0\}$.

Dabei gilt: z_0 akzeptiert das leere Wort. Mit einer 0 geht der DPDA nach z_1 und zählt dort weitere 0en, indem entsprechend viele A auf den Stack gelegt werden. Sobald in z_1 die erste 1 gelesen wird, geht der Automat nach z_2 und zählt dort 1en. Wenn mindestens so viele 1en wie vorher 0en gelesen werden, geht der Automat in den Endzustand z_3, der noch beliebig viele weitere 1en akzeptiert. Mit einer 1 geht z_0 direkt nach z_3, da $n = 0$ gewählt wird. Sollten 0en und 1en im falschen Muster auftreten, ist entsprechend die Übergangsfunktion dort nicht definiert und der Automat lehnt das Wort ab.

b) Ein DPDA, dessen Sprache gleich der Sprache $\{0^n 1^m \mid n \geq m, m, n \geq 0\}$ ist, ist $M_b = (\{0,1\}, \{A, \#\}, \{z_0, z_1\}, \delta_b, z_0, \#, \{z_0, z_1\})$, wobei δ_b definiert ist durch:

$$\begin{array}{cc} z_0 0\# \to z_0 A\# & z_0 1A \to z_1 \lambda \\ z_0 0A \to z_0 AA & z_1 1A \to z_1 \lambda \end{array}$$

Dabei gilt: z_0 zählt 0en, indem entsprechend viele A auf den Stack gelegt werden. Da $n \geq m \geq 0$ gewählt werden darf, ist z_0 ein Endzustand. Mit der ersten 1 geht der Automat nach z_1, wo höchstens so viele 1en wie vorher 0en akzeptiert werden, indem die A wieder vom Stack genommen werden. Ansonsten lehnt der Automat das Wort ab, weil wieder die Übergangsfunktion entsprechend nicht definiert wird.

Die Kommentare müssen nicht so ausführlich wie hier sein, sondern dienen dazu, verrückte Automaten korrigierbar zu machen. Die Zustandsbeschreibungen sollten euch helfen, die Automaten zu verstehen
Falls Beschreibungen nicht verständlich sind, merkt das auf jeden Fall an.

Aufgabe 3 (Kellerautomat III: Kontextfreie Grammatik)16P

a) Gegeben sei die kontextfreie Grammatik $G = (\Sigma, N, S, P)$ mit $\Sigma = \{a, b\}, N = \{S, A, X\}$ und

$$\begin{aligned} P = \{S &\to AX \mid b, \\ X &\to SA, \\ A &\to a\}. \end{aligned}$$

Geben Sie einen Kellerautomaten M mit $L(M) = L(G)$ an. Verwenden Sie dabei die Konstruktion aus der Vorlesung.

b) Gegeben sei der Kellerautomat $M = (\{a, b\}, \Gamma, Z, \delta, z_0, \#)$ mit $\Gamma = \{A, \#\}, Z = \{z_0, z_1\}$ und der folgenden Überführungsfunktion δ:

$$\begin{array}{rl} z_0 a\# \to z_0 A\# & z_1 \lambda\# \to z_1 \lambda \\ z_0 aA \to z_0 AA & z_1 bA \to z_1 \lambda \\ z_0 bA \to z_1 \lambda & \end{array}$$

Konstruieren Sie nach dem Verfahren aus der Vorlesung eine kontextfreie Grammatik G mit $L(G) = L(M)$. Geben Sie alle relevanten Zwischenschritte an und geben Sie alle Nichtterminale explizit an.

Lösungsvorschlag:

a) $M = (\Sigma, \{S, A, X, a, b\}, \{z\}, \delta, z, S)$ mit $\Sigma = \{a, b\}$ und der folgenden Überführungsfunktion

$$\begin{aligned} z\lambda S &\to zAX & zaa &\to z\lambda \\ z\lambda S &\to zb & zbb &\to z\lambda \\ z\lambda X &\to zSA \\ z\lambda A &\to za \end{aligned}$$

b) $G = (\Sigma, N, S, P)$ mit
$\Sigma = \{a, b\}$,
$N = \{S\} \cup Z \times \Gamma \times Z =$
$\{S, (z_0, A, z_0), (z_0, A, z_1), (z_1, A, z_0), (z_1, A, z_1),$
$(z_0, \#, z_0), (z_0, \#, z_1), (z_1, \#, z_0), (z_1, \#, z_1)\}$
und P in folgenden Schritten:

(1) $S \to (z_0, \#, z)$ für jedes $z \in Z$, also
- $S \to (z_0, \#, z_0)$,
- $S \to (z_0, \#, z_1)$

(2) $(z, A, z') \to a$ falls $(z', \lambda) \in \delta(z, a, A), z, z' \in Z, a \in \Sigma \cup \{\lambda\}, A \in \Gamma$
- $(z_0, A, z_1) \to b$, da $(z_1, \lambda) \in \delta(z_0, b, A)$
- $(z_1, \#, z_1) \to \lambda$, da $(z_1, \lambda) \in \delta(z_1, \lambda, \#)$
- $(z_1, A, z_1) \to b$, da $(z_1, \lambda) \in \delta(z_1, \lambda, \#)$

(3) $(z, A, z') \to a(z_1, B, z')$, falls $(z_1, B) \in \delta(z, a, A)$ mit $z_1, z, z' \in Z, a \in \Sigma \cup \{\lambda\}, A, B \in \Gamma$
Regeln dieser Form kommen hier nicht vor.

(4) $(z, A, z') \to a(z_1, B, z_2)(z_2, C, z')$, falls $(z_1, BC) \in \delta(z, a, A)$ mit $z_1, z_2, z, z' \in Z, a \in \Sigma \cup \{\lambda\}, A, B \in \Gamma$
- $(z_0, \#, z_0) \to a(z_0, A, z_0)(z_0, \#, z_0)$
- $(z_0, \#, z_0) \to a(z_0, A, z_1)(z_1, \#, z_0)$
- $(z_0, \#, z_1) \to a(z_0, A, z_0)(z_0, \#, z_1)$
- $(z_0, \#, z_1) \to a(z_0, A, z_1)(z_1, \#, z_1)$, da $(z_0, A\#) \in \delta(z_0, a, \#)$

- $(z_0, A, z_0) \to a(z_0, A, z_0)(z_0, A, z_0)$
- $(z_0, A, z_0) \to a(z_0, A, z_1)(z_1, A, z_0)$
- $(z_0, A, z_1) \to a(z_0, A, z_0)(z_0, A, z_1)$
- $(z_0, A, z_1) \to a(z_0, A, z_1)(z_1, A, z_1)$