Lösungsblatt 14

Vorbereitungsaufgaben

Vorbereitungsaufgabe 1

Seien $G = (\{S, A, B, C\}, \{a, b\}, P, S)$ eine Grammatik in Kuroda-Normalform und

$$P' = P \cup \{(CaB, aBCbAB)\}.$$

Geben Sie eine zu $G' = (\{S, A, B, C\}, \{a, b\}, P', S)$ äquivalente Grammatik G'' in Kuroda-Normalform an. Gehen Sie wie in den Vorlesungsfolien 35.2 und 35.3 beschrieben vor.

Lösung

$G'' = (\{S, A, B, C, D, E, F, G, V_a, V_b\}, \{a, b\}, P'', S)$ mit

$$P'' = P \cup \{(V_a, a), (V_b, b), (CV_a, V_aD), (DB, BE), (E, CF), (F, V_bG), (G, AB)\}.$$

Herleitung:

1. Verwende Pseudoterminale für a und b, d. h. ersetze die Produktion $CaB \to aBCbAB$ durch $CV_aB \to V_aBCV_bAB$, $V_a \to a$ und $V_b \to b$.
2. Ersetze $CV_aB \to V_aBCV_bAB$ durch $CV_a \to V_aD$ und $DB \to BCV_bAB$.
3. Ersetze $DB \to BCV_bAB$ durch $DB \to BE$ und $E \to CV_bAB$.
4. Ersetze $E \to CV_bAB$ durch $E \to CF$ und $F \to V_bAB$.
5. Ersetze $F \to V_bAB$ durch $F \to V_bG$ und $G \to AB$.

Vorbereitungsaufgabe 2

Sei $M = (Q, \Sigma, \Gamma, \delta, 0, \square, F)$ die DTM aus Ergänzungsblatt 13, Präsenzaufgabe 4. Ziel dieser Aufgabe ist es, M zu formalisieren und wichtige Konzepte anhand von M zu illustrieren.

1. Geben Sie die Mengen Q, Σ, Γ und F konkret an und vervollständigen Sie folgende tabellarische Darstellung der Überführungsfunktion δ:

δ	a	b	$\square$
0	$(0, a, R)$	$(1, a, R)$	$(4, \square, L)$
1	$(2, b, L)$	$(1, b, R)$	$(3, \square, L)$
2			
3			
4			
5			

   Beispielsweise gilt dann: $\delta(1, a) = (2, b, L)$.

   Hinweis: Beachten Sie, dass die Überführungsfunktion von DTMs, im Gegensatz zu der von DFAs, auch partiell sein darf. Somit können einige Felder der Tabelle auch leer bleiben.

2. Was ist die von M akzeptierte Sprache $T(M)$?

3. Geben Sie eine akzeptierende Konfigurationsfolge für das Eingabewort aba an.

Lösung

1. $M = (Q, \Sigma, \Gamma, \delta, 0, \square, F)$ mit $Q = \{0, 1, 2, 3, 4, 5\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, \square\}$, $F = \{5\}$ und $\delta \colon Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$ mit

δ	a	b	$\square$
0	$(0, a, R)$	$(1, a, R)$	$(4, \square, L)$
1	$(2, b, L)$	$(1, b, R)$	$(3, \square, L)$
2	$(0, a, R)$	$(2, b, L)$	
3	$(4, b, L)$	$(3, b, L)$	
4	$(4, a, L)$	$(4, b, L)$	$(5, \square, R)$
5			

   Hinweise:

   - Das obige δ ist für $(2, \square)$, $(3, \square)$, $(5, a)$, $(5, b)$ und $(5, \square)$ nicht definiert und ist daher echt partiell. Dies ist erlaubt, weil wir ab jetzt *partielle Funktionen* vereinfacht nur *Funktionen* nennen. Wir fordern also nicht mehr, dass Funktionen total (also für jede Eingabe definiert) sind.*
   - Die Überführungsfunktion von DFAs ist dagegen immer total. Dies liegt daran, dass wir diese Abschwächung des Begriffs *Funktion* erst ab dem Thema *Turingmaschinen* erlauben. Eine partielle Überführungsfunktion bei DFAs wäre einerseits praktisch, weil man beim Zeichnen von DFAs Fangzustände weglassen könnte, aber andererseits gäbe es keine genaue Entsprechung mehr zwischen den Zuständen des minimalen DFAs und den Äquivalenzklassen der entsprechenden Myhill-Nerode-Äquivalenz.

*An dieser Stelle würden einige Mathematiker Bauchweh kriegen, aber wir orientieren uns nun mal am Buch von Schöning und dort wird es in den Kapiteln 2 und 3 so gehandhabt.

2. Da jedes Wort aus Σ^* in den Endzustand führt, ist $T(M) = \Sigma^*$.
3. $0aba \vdash a0ba \vdash aa1a \vdash a2ab \vdash aa0b \vdash aaa1 \vdash aa3a \vdash a4ab \vdash 4aab \vdash 4\Box aab \vdash 5aab$.

Präsenzaufgaben

Präsenzaufgabe 1

Welche der folgenden Aussagen sind richtig und welche falsch? Begründen Sie die Korrektheit Ihrer Antworten kurz.

1. Zu jeder Typ-1-Grammatik G mit $\varepsilon \notin L(G)$ gibt es eine äquivalente Grammatik bei der alle Regeln von einem der folgenden 5 Typen sind:

$$A \to a \qquad A \to B \qquad A \to BC \qquad AB \to AC \qquad AC \to CB$$

2. Zu jeder Typ-0-Grammatik G mit $\varepsilon \notin L(G)$ gibt es eine äquivalente Grammatik bei der alle Regeln von einem der folgenden 4 Typen sind:

$$A \to a \qquad A \to BC \qquad AB \to CDE \qquad ABC \to DEF$$

3. Zu jeder Typ-0-Grammatik G gibt es eine äquivalente Grammatik bei der alle Regeln von einem der folgenden 5 Typen sind:

$$A \to a \qquad A \to B \qquad A \to BC \qquad AB \to CD \qquad A \to \varepsilon$$

Lösung

1. Die Aussage ist wahr.

   Da zu jeder Typ-1-Grammatik G mit $\varepsilon \notin L(G)$ eine äquivalente Grammatik in Kuroda-Normalform existiert, muss nur gezeigt werden, wie eine Regel vom Typ $AB \to CD$ durch Regeln der oben genannten Typen ersetzt werden kann. Für jede solche Regel fügen wir zwei frische Variablen X und Y hinzu und ersetzen die Regel durch folgende:

$$AB \to XB \qquad XB \to XY \qquad XY \to CY \qquad CY \to CD.$$

   Auf diese Weise erhalten wir eine zu G äquivalente Grammatik G', die die gewünschte Form besitzt.

   Hinweise:

   - Ersetzt man in G jede Regel der Form $AB \to CD$ durch die Regeln $AB \to CB$ und $CB \to CD$, so bekommt man zwar eine Grammatik in der gewünschten Form, die jedes Wort aus $L(G)$ erzeugt, diese kann aber unter Umständen auch Wörter erzeugen, die nicht in $L(G)$ sind.

Als Gegenbeispiel betrachte man $G = (\{S, A, B, C, D\}, \{a\}, P, S)$ mit den Produktionsregeln

$$S \to AB \qquad\qquad AB \to CD \qquad\qquad CB \to aa.$$

Die modifizierte Grammatik könnte mittels

$$S \Rightarrow AB \Rightarrow CB \Rightarrow aa$$

das Wort aa erzeugen, obwohl $aa \notin L(G)$ gilt.

- Ersetzt man in G jede Regel der Form $AB \to CD$ durch die Regeln $AB \to AX$, $AX \to CX$ und $CX \to CD$, so bekommt man wieder eine Grammatik in der gewünschten Form, die jedes Wort aus $L(G)$ erzeugt, aber unter Umständen auch Wörter erzeugt, die nicht in $L(G)$ sind.

  Als Gegenbeispiel betrachte man $G = (\{S, A, B, C, D\}, \{a\}, P, S)$ mit den Produktionen

  $$S \to AAB \qquad AB \to CD \qquad AA \to CC \qquad C \to a \qquad D \to a.$$

  Die modifizierte Grammatik könnte mittels

  $$S \Rightarrow AAB \Rightarrow AAX \Rightarrow CCX \Rightarrow CCD \Rightarrow^3 aaa$$

  das Wort aaa erzeugen, obwohl $aaa \notin L(G)$ gilt.

2. Die Aussage ist falsch.

   Wäre die Aussage richtig, dann könnte man zu jeder Typ-0-Grammatik eine äquivalente Typ-1-Grammatik finden. Somit wäre jede Typ-0-Sprache vom Typ 1. Aus der Vorlesung wissen wir allerdings, dass das nicht so ist.

3. Die Aussage ist wahr.

   Sei $G = (V, \Sigma, P, S)$ eine beliebige Typ-0-Grammatik. Ersetzt man in G jede verkürzende Regel $\alpha \to \beta$ durch $\alpha \to \beta X^{|\alpha|-|\beta|}$ (mit $X \notin V$ beliebig), so erhält man eine Typ-1-Grammatik. Diese kann laut Vorlesung in Kuroda-Normalform gebracht werden. Beim Hinzufügen der verkürzenden Regel $X \to \varepsilon$ erhält man eine zu G äquivalente Grammatik in der gewünschten Form.

Präsenzaufgabe 2

Wir betrachten folgende Sprache über dem Alphabet $\Sigma = \{a, b, c\}$:

$$L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$$

1. Geben Sie eine Turingmaschine mit höchstens 6 Zuständen für L an.
2. Geben Sie eine akzeptierende Konfigurationsfolge für das Eingabewort abc an.

Lösung

Die Lösung ist nicht eindeutig. Elegantere Lösungen mit weniger Zuständen oder Bandsymbolen sind immer herzlich willkommen!

1. Wir geben eine DTM M mit $T(M) = L$ an.

   Intuitive Beschreibung

   Die DTM M durchläuft das Wort mehrmals von links nach rechts und markiert bei jedem Durchlauf (mit #) jeweils das erste a, das erste b und das erste c, das sie findet (in dieser Reihenfolge). Sind alle Zeichen markiert, dann akzeptiert sie. Wird ein Zeichen, das markiert werden soll, nicht gefunden oder werden Zeichen an falschen Positionen im Wort gefunden, kommt M zum Stillstand.

   Für die Implementierung dieser Strategie verwenden wir folgende Zustände:

   0: Suche nach rechts das erste a, markiere es und gehe in den Zustand 1 über. Falls ein b oder ein c gelesen wird, lehne ab. Falls nur # vorhanden sind, akzeptiere.

   1: Suche nach rechts das erste b, markiere es und gehe in den Zustand 2 über. Falls ein c gelesen wird oder kein b gefunden wird, lehne ab.

   2: Suche nach rechts das erste c, markiere es und gehe in den Zustand 3 über. Falls ein a gelesen wird oder kein c gefunden wird, lehne ab.

   3: Bewege den Leseschreibkopf bis zum Ende des Wortes und gehe in den Zustand 4 über. Falls ein a, ein b oder ein # gelesen wird, lehne ab.

   4: Bewege den Leseschreibkopf bis zum Anfang des Wortes und gehe dann in den Zustand 0 über.

   5: Akzeptiere die Eingabe.

   Formale Definition

   Wir wählen die DTM

$$M = (\{0, 1, 2, 3, 4, 5\}, \{a, b, c\}, \{a, b, c, \#, \square\}, \delta, 0, \square, \{5\})$$

   mit

δ	a	b	c	$\#$	$\square$
0	$(1, \#, R)$			$(0, \#, R)$	$(5, \square, N)$
1	$(1, a, R)$	$(2, \#, R)$		$(1, \#, R)$	
2		$(2, b, R)$	$(3, \#, R)$	$(2, \#, R)$	
3			$(3, c, R)$		$(4, \square, L)$
4	$(4, a, L)$	$(4, b, L))$	$(4, c, L)$	$(4, \#, L)$	$(0, \square, R)$
5					

   Hinweis: Unter

   `http://morphett.info/turing/?ee7f00ead9bdc914747bf0d75b6d27da`

   kann diese Turingmaschine simuliert werden.

2. $0abc \vdash \#1bc \vdash \#\#2c \vdash \#\#\#3 \vdash \#\#4\# \vdash \#4\#\# \vdash 4\#\#\# \vdash 4\Box\#\#\# \vdash 0\#\#\# \vdash$ $\#0\#\# \vdash \#\#0\# \vdash \#\#\#0 \vdash \#\#\#5$.

Präsenzaufgabe 3

Wir betrachten folgende Sprache über dem Alphabet $\Sigma = \{a, b\}$:

$$L = \{ww \,|\, w \in \{a, b\}^*\}$$

1. Geben Sie eine Turingmaschine mit höchstens 10 Zuständen für L an.
2. Geben Sie eine akzeptierende Konfigurationsfolge für das Eingabewort $bbbb$ an.

Lösung

Die Lösung ist nicht eindeutig. Elegantere Lösungen mit weniger Zuständen oder Bandsymbolen sind immer herzlich willkommen!

1. Wir geben diesmal eine NTM M mit $T(M) = L$ an.

   Intuitive Beschreibung

   M durchläuft das Wort mehrmals von links nach rechts und vergleicht bei jedem Durchlauf das erste unmarkierte Zeichen der ersten Worthälfte mit dem der zweiten. Stimmen sie überein, so werden sie markiert (mit #). Stimmen sie nicht überein, so wird das Eingabewort abgelehnt. Wir nutzen den Nichtdeterminismus aus, um das erste Zeichen der zweiten Hälfte des Wortes zu „raten“.

   Für die Implementierung dieser Strategie verwenden wir folgende Zustände:

   0: Startzustand. Markiere das erste Zeichen im Wort und gehe entsprechend in den Zustand 1 oder 2 über.

   1: Das erste Zeichen war ein a. Suche nichtdeterministisch das erste Zeichen der zweiten Hälte des Wortes und falls dieses ein a ist, markiere es und gehe in den Zustand 3 über.

   2: Das erste Zeichen war ein b. Suche nichtdeterministisch das erste Zeichen der zweiten Hälte des Wortes und falls dieses ein b ist, markiere es und gehe in den Zustand 3 über.

   3: Bewege den Leseschreibkopf bis zum Anfang des Wortes und gehe dann in den Zustand 4 über.

   4: markiere das linkeste a oder b und gehe entsprechend in den Zustand 5 oder 6 über. Falls nur # vorhanden sind, akzeptiere.

   5: Ein a wurde in der linken Hälfte des Wortes markiert. Bewege den Leseschreibkopf zur zweiten Hälfte des Wortes und gehe in den Zustand 7 über.

   6: Ein b wurde in der linken Hälfte des Wortes markiert. Bewege den Leseschreibkopf zur zweiten Hälfte des Wortes und gehe in den Zustand 8 über.

   7: Falls das erste unmarkierte Zeichen in der zweiten Worthälfte ein a ist, markiere es und gehe in den Zustand 3 über. Lehne sonst ab.

8: Falls das erste unmarkierte Zeichen in der zweiten Worthälfte ein b ist, markiere es und gehe in den Zustand 3 über. Lehne sonst ab.

9: Endzustand. Akzeptiere das Eingabewort.

<u>Formale Definition</u>

Wir wählen eine NTM

$$M = (\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, \{a, b\}, \{a, b, \#, \square\}, \delta, 0, \square, \{9\})$$

mit

δ	a	b	$\#$	$\square$
0	$\{(1, \#, R)\}$	$\{(2, \#, R)\}$	$\emptyset$	$\{(9, \square, N)\}$
1	$\{(1, a, R), (3, \#, L)\}$	$\{(1, b, R)\}$	$\emptyset$	$\emptyset$
2	$\{(2, a, R)\}$	$\{(2, b, R), (3, \#, L)\}$	$\emptyset$	$\emptyset$
3	$\{(3, a, L)\}$	$\{(3, b, L)\}$	$\{(3, \#, L)\}$	$\{(4, \square, R)\}$
4	$\{(5, \#, R)\}$	$\{(6, \#, R)\}$	$\{(4, \#, R)\}$	$\{(9, \square, N)\}$
5	$\{(5, a, R)\}$	$\{(5, b, R)\}$	$\{(7, \#, R)\}$	$\emptyset$
6	$\{(6, a, R)\}$	$\{(6, b, R)\}$	$\{(8, \#, R)\}$	$\emptyset$
7	$\{(3, \#, L)\}$	$\emptyset$	$\{(7, \#, R)\}$	$\emptyset$
8	$\emptyset$	$\{(3, \#, L)\}$	$\{(8, \#, R)\}$	$\emptyset$
9	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Hinweise:

- Man beachte, dass die Überführungsfunktion einer nichtdeterministischen Turingmaschine keine einzelnen Tripel, sondern Mengen von Tripeln zurückgibt.
- Weil M nichtdeterministisch ist, ist eine Simulation dieser Turingmaschine leider nicht möglich.

2. $0bbbb \vdash \#2bbb \vdash \#b2bb \vdash \#3b\#b \vdash 3\#b\#b \vdash 3\square\#b\#b \vdash 4\#b\#b \vdash \#4b\#b \vdash \#\#6\#b \vdash \#\#\#8b \vdash \#\#3\#\# \vdash \#3\#\#\# \vdash 3\#\#\#\# \vdash 3\square\#\#\#\# \vdash 4\#\#\#\# \vdash \#4\#\#\# \vdash \#\#4\#\# \vdash \#\#\#4\# \vdash \#\#\#\#4\square \vdash \#\#\#\#9\square$.

Präsenzaufgabe 4

Wir betrachten folgende Sprache über dem Alphabet $\Sigma = \{a, b\}$:

$$L = \left\{w \in \Sigma^* \,\middle|\, w^R = w\right\}$$

Erinnerung: w^R ist das zu w gespiegelte Wort, z. B. $(baaba)^R = abaab$.

1. Geben Sie eine DTM M_1 mit höchstens 7 Zuständen für L an.
2. Geben Sie eine NTM M_2 mit höchstens 6 Zuständen für L an.
3. Geben Sie für M_1 und M_2 jeweils eine akzeptierende Konfigurationsfolge für das Eingabewort $abba$ an.

Lösung

Die Lösung ist nicht eindeutig. Elegantere Lösungen mit weniger Zuständen oder Bandsymbolen sind immer herzlich willkommen!

1. Intuitive Beschreibung

   Die DTM M_1 löscht immer abweselnd die äußersten Zeichen im Wort, indem sie diese durch das Blanksymbol ersetzt. Dabei vergleicht sie, ob diese gleich sind.

   Für die Implementierung dieser Strategie verwenden wir folgende Zustände:

   0: Startzustand. Lösche das gelesene Zeichen. War es vor dem Löschen ein a, gehe in den Zustand 1 über. War es vor dem Löschen ein b, gehe in den Zustand 2 über. War es vor dem Löschen bereits ein $\square$, gehe in den Endzustand 6 über.

   1: Das zuletzt gelöschte Zeichen war ein a. Bewege den Leseschreibkopf zum letzten Zeichen im Wort.

   2: Das zuletzt gelöschte Zeichen war ein b. Bewege den Leseschreibkopf zum letzten Zeichen im Wort.

   3: Falls das gelesene Zeichen ein a ist, lösche es und gehe in den Zustand 5 über. Falls es ein b ist, lehne ab. Falls kein Zeichen mehr übrig bleibt, gehe in den Endzustand über.

   4: Falls das gelesene Zeichen ein b ist, lösche es und gehe in den Zustand 5 über. Falls es ein a ist, lehne ab. Falls kein Zeichen mehr übrig bleibt, gehe in den Endzustand über.

   5: Bewege den Leseschreibkopf zum ersten Zeichen im Wort und gehe in den Zustand 0 über.

   6: Endzustand. Akzeptiere das Eingabewort.

   Formale Definition

   Wir wählen die DTM

$$M_1 = (\{0,1,2,3,4,5,6\}, \{a,b\}, \{a,b,\square\}, \delta, 0, \square, \{6\})$$

   mit

δ	a	b	$\square$
0	$(1,\square,R)$	$(2,\square,R)$	$(6,\square,N)$
1	$(1,a,R)$	$(1,b,R)$	$(3,\square,L)$
2	$(2,a,R)$	$(2,b,R)$	$(4,\square,L)$
3	$(5,\square,L)$		$(6,\square,N)$
4		$(5,\square,L)$	$(6,\square,N)$
5	$(5,a,L)$	$(5,b,L)$	$(0,\square,R)$
6			

   Hinweis: Unter

   `http://morphett.info/turing/?81d1e848b78a65363fd0fb37df2bfad2` kann diese Turingmaschine simuliert werden.

2. Intuitive Beschreibung

   Hat das Eingabewort gerade Länge, so wird die Position des letzten Zeichen in der linken Hälfte nichtdeterministisch geraten. Hat das Eingabewort ungerade Länge, so wird nichtdeterministisch die Mitte des Wortes geraten und entfernt. Danach wird „von innen nach außen“ verglichen, ob die eine Worthälfte Spiegelung der anderen ist.

   Für die Implementierung dieser Strategie verwenden wir folgende Zustände:

   0: Startzustand. Suche nichtdeterministisch die Mitte des Wortes. Falls die Wortlänge ungerade ist, markiere das Zeichen in der Mitte. Bewege den Leseschreibkopf eine Position nach links und gene in den Zustand 1 über.

   1: Suche nach links das erste unmarkierte Zeichen und markiere es. War es ein a, dann gehe in den Zustand 2 über. War es ein b, dann gehe in den Zustand 3 über. Der Leseschreibkopf kann sich danach nach rechts bewegen oder stehen bleiben. Wird kein unmarkiertes Zeichen gefunden, gehe in den Zustand 4 über.

   2: Suche nach rechts das erste unmarkierte Zeichen. Falls es ein a ist, markiere es und gehe in den Zustand 1 über. Der Leseschreibkopf kann sich danach entweder nach links bewegen oder er kann stehen bleiben. Falls es kein a ist, lehne das Eingabewort ab.

   3: Suche nach rechts das erste unmarkierte Zeichen. Falls es ein b ist, markiere es und gehe in den Zustand 1 über. Der Leseschreibkopf kann sich danach entweder nach links bewegen oder er kann stehen bleiben. Falls es kein a ist, lehne das Eingabewort ab.

   4: Suche nach rechts das erste unmarkiert Zeichen. Falls keins gefunden wird, gehe in den Endzustand über. Lehne sonst ab.

   5: Endzustand. Akzeptiere das Eingabewort.

   Formale Definition

   Wir wählen die NTM

$$M_2 = (\{0,1,2,3,4,5\}, \{a,b\}, \{a,b,\#,\square\}, \delta, 0, \square, \{5\})$$

   mit

δ	a	b	$\#$	$\square$
0	$\{(0,a,N),(1,a,N),(1,\#,N)\}$	$\{(0,b,N),(1,b,N),(1,\#,n)\}$	$\emptyset$	$\emptyset$
1	$\{(2,\#,R)\}$	$\{(3,\#,R)\}$	$\{(1,\#,L)\}$	$\{(4,\square,R)\}$
2	$\{(1,\#,L)\}$	$\emptyset$	$\{(2,\#,R)\}$	$\emptyset$
3	$\emptyset$	$\{(1,\#,L)\}$	$\{(3,\#,R)\}$	$\emptyset$
4	$\emptyset$	$\emptyset$	$\{(4,\#,R)\}$	$\{(5,\square,N)\}$
5	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

   Hinweis: Weil M nichtdeterministisch ist, ist eine Simulation dieser Turingmaschine leider nicht möglich.

Präsenzaufgabe 5

Seien $m \in \mathbb{N} \setminus \{0\}$ eine natürliche Zahl, $\Sigma = \{a_1, \dots, a_m\}$ ein m-elementiges Alphabet und L folgende Sprache über Σ:

$$L = \{w \in \Sigma^* \mid \text{in } w \text{ kommt jeder Buchstabe aus } \Sigma \text{ vor}\}.$$

Geben Sie jeweils eine DTM M für L mit den angefordertern Eigenschaften an.

1. M bewegt den Leseschreibkopf nie nach links.
2. M besitzt höchstens $2m$ Zustände und höchstens $m+1$ Bandsymbole.
3. M besitzt höchstens $m+2$ Zustände und höchstens $m+2$ Bandsymbole.

Lösung

Erinnerung: Für alle $m \in \mathbb{N}$ gilt $[m] = \{k \in \mathbb{N} \mid 1 \leq k \leq m\}$.

1. $M = (\mathcal{P}(\Sigma), \Sigma, \Sigma \cup \{\square\}, \delta, \emptyset, \square, \{\Sigma\})$ mit

$$\delta(q, x) = \begin{cases} (q, \square, N) & \text{für } x = \square \\ (q \cup \{x\}, x, R) & \text{für } x \neq \square. \end{cases}$$

2. $M = (\{s_1, \dots, s_m, f_1, \dots, f_m\}, \Sigma, \Sigma \cup \{\square\}, \delta, s_1, \square, \{f_m\})$ mit

$$\delta(s_i, a_j) = \begin{cases} (f_i, a_j, N) & \text{für } i = j \\ (s_i, a_j, R) & \text{für } i \neq j \end{cases}$$

für alle $i, j \in [m]$ und

$$\delta(f_i, a_j) = (f_i, a_j, L)$$
$$\delta(f_i, \square) = (s_{i+1}, \square, R)$$

für alle $i \in [m-1]$.

3. $M = (\{s_1, \dots, s_m, f, a\}, \Sigma, \Sigma \cup \{\#, \square\}, \delta, s_1, \square, \{a\})$ für

$$\delta \colon Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$$

mit

$$\delta(s_i, a_j) = \begin{cases} (f, a_j, N) & \text{für } i = j \\ (s_i, a_j, R) & \text{für } i \neq j \end{cases}$$
$$\delta(s_i, \#) = \begin{cases} (a, \#, N) & \text{für } i = m \\ (s_{i+1}, \#, R) & \text{für } i \neq m \end{cases}$$
$$\delta(f, a_j) = (s_i, \square, N)$$
$$\delta(f, \square) = (s_1, \#, N)$$

für alle $i, j \in [m]$.

Knobelaufgaben

Knobelaufgabe 1

Geben jeweils eine Turingmaschine für folgende Sprachen über $\Sigma = \{a\}$ an.

1. $L = \left\{a^k \,\middle|\, k \text{ ist eine Zweierpotenz}\right\}$
2. $L = \left\{a^k \,\middle|\, k \text{ ist eine Quadratzahl}\right\}$