Einführung in die Theoretische Informatik

Sommersemester 2020 – Übungsblatt Lösungsskizze 7

AUFGABE 7.1. (*Wichtige Begriffe*) Stufe A

Überprüfen Sie, dass Sie die folgenden Begriffe korrekt definieren können.

- nicht-deterministische/deterministische Turing-Maschine
- Konfiguration einer Turing-Maschine
- akzeptierte Sprache einer Turing-Maschine
- Turing-berechenbar

Turing Maschinen interaktiv

Es gibt verschiedene Websiten auf denen Turing Maschinen interaktiv konstruiert und simuliert werden können, z.B. https://wimmers.github.io/turing-machine-viz/. Beachten Sie dass diese Seite Endzustände nicht visualisiert.

AUFGABE 7.2. (*TM für Sprache*) Stufe C

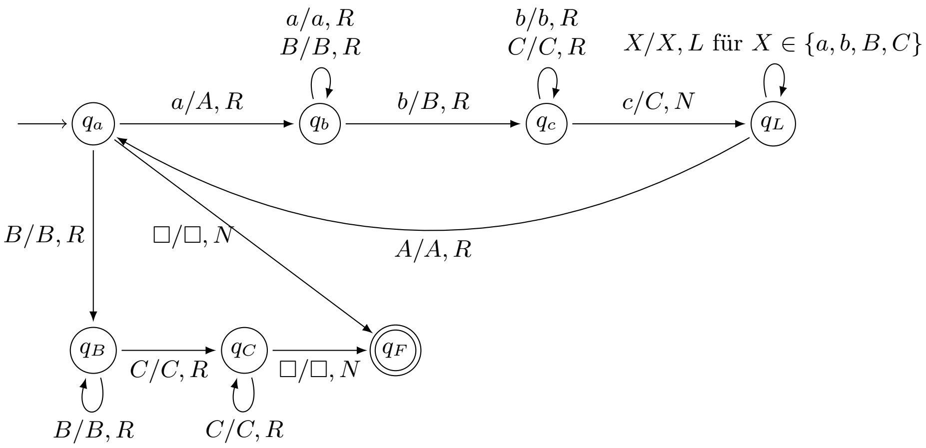
Geben Sie für die angegebene Sprache eine passende TM M an.

$$\text{(a) } \mathsf{L}_F(\mathsf{M}) = \{\mathsf{a}^n\mathsf{b}^n\mathsf{c}^n \mid n \in \mathbb{N}\}$$

Lösungsskizze

Idee: Ersetze für jedes a je ein a, b und c durch x, wobei nach dem Ersetzen die restlichen Buchstaben der gleichen Art unverändert gelassen werden. Wenn am Ende nur noch x auf dem Band stehen, terminiere. Sonst bleibt die Berechnung in einem Nichtendzustand stecken.

Wir schreiben $\Box$ für eine leere Bandzelle. Sei TM $M = (\{q_a, q_b, q_c, q_L, q_B, q_C, q_F\}, \{a, b, c\}, \Sigma \cup \{A, B, C, \Box\}, \delta, q_a, \Box, \{q_F\})$.



AUFGABE 7.3. (*TM Berechnung*) Stufe B

Konstruieren Sie eine Turingmaschine $T = (Q, \Sigma, \Gamma, q_0, \Box, \delta, F)$, mit $\Sigma = \{|\}$, die eine eingegebene Strichzahl verdoppelt.

Lösungsskizze
Die Berechnung erfolgt, indem jeweils ein Strich am linken Anfang der Strichfolge durch eine Markierung x ersetzt wird und anschließend am rechten Ende der Strichfolge eine Markierung y angefügt wird. Falls keine Strichzeichen mehr vorhanden sind, werden alle Zeichen in Striche umgewandelt.
Sei $T = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$ mit $Q = \{q_0, q_1, q_2, q_3, q_4, q_f\}$, $\Sigma = \{|\}$,
$\Gamma = \{|, x, y, \Box\}$ und $F = \{q_f\}$.
Die Zustände und Zeichenmengen kann man auch der folgenden Tabelle der Übergangsfunktion δ entnehmen.

Übergang			Kommentar
$\delta(q_0, \Box)$	$\rightarrow$	$(q_3, \Box, L)$	Nichts mehr zu verdoppeln
$\delta(q_0, \|)$	$\rightarrow$	(q_1, x, R)	| wird verdoppelt, x ist ein Hilfszeichen
$\delta(q_0, y)$	$\rightarrow$	(q_0, y, R)	Überspringe Hilfszeichen y
$\delta(q_1, \Box)$	$\rightarrow$	(q_2, y, N)	Schreibe | an rechten Rand, markiert durch Hilfszeichen y
$\delta(q_1, \|)$	$\rightarrow$	$(q_1, \|, R)$	An den rechten Rand gehen
$\delta(q_1, y)$	$\rightarrow$	(q_1, y, R)	An den rechten Rand gehen
$\delta(q_2, \|)$	$\rightarrow$	$(q_2, \|, L)$	Zurück zum nächsten |
$\delta(q_2, y)$	$\rightarrow$	(q_2, y, L)	Zurück zum nächsten |
$\delta(q_2, x)$	$\rightarrow$	(q_0, x, R)	Nächstes | verdoppeln oder halten
$\delta(q_3, y)$	$\rightarrow$	$(q_3, \|, L)$	Hilfszeichen durch | ersetzen
$\delta(q_3, x)$	$\rightarrow$	$(q_3, \|, L)$	Hilfszeichen durch | ersetzen
$\delta(q_3, \Box)$	$\rightarrow$	$(q_f, \Box, R)$	Halten

Definition (Alternative Akzeptanzbedingungen für Turing-Maschinen)

In der Vorlesung wurde die Annahme gemacht, dass die Übergangsfunktion δ einer Turingmaschine folgende Eigenschaft erfüllt:

$$\delta(q, \mathsf{a}) \textit{ ist nicht definiert für alle } q \in \mathsf{F},\ \mathsf{a} \in \Gamma.$$

Sei $\mathcal{M}_A$ die Menge der Turingmaschinen, die diese Annahme erfüllen, und sei $\mathcal{M}$ die Menge aller Turingmaschinen. Es gilt somit $\mathcal{M}_A \subsetneq \mathcal{M}$.
Für $\mathsf{M} \in \mathcal{M}$ mit $\mathsf{M} = (\mathsf{Q}, \Sigma, \Gamma, \delta, \mathsf{q}_0, \Box, \mathsf{F})$ definiere:

- $\mathsf{L}_F(\mathsf{M}) = \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, f \in \mathsf{F}.\ (\varepsilon, \mathsf{q}_0, w) {\rightarrow^*}_{\mathsf{M}} (\alpha, f, \beta)\}$.
  (Menge der Wörter, für die die Maschine einen Endzustand irgendwann besucht.)
- $\mathsf{L}_H(\mathsf{M}) = \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, q \in \mathsf{Q}.\ (\varepsilon, \mathsf{q}_0, w) \rightarrow_M^* (\alpha, q, \beta)$ und $\delta(q, \mathit{first}(\beta))$ ist nicht definiert$\}$.
  (Menge der Wörter, für die die Maschine hält.)
- $\mathsf{L}_{HF}(\mathsf{M}) = \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, f \in \mathsf{F}.\ (\varepsilon, \mathsf{q}_0, w) \rightarrow_M^* (\alpha, f, \beta)$ und $\delta(f, \mathit{first}(\beta))$ ist nicht definiert$\}$.
  (Menge der Wörter, für die die Maschine in einem Endzustand hält.)

AUFGABE 7.4. (*TM Akzeptanzbedingungen*) Stufe B

Begründen Sie folgende Aussagen, indem Sie eine passende Konstruktion angeben.

(a) Für jede Turing-Maschine $\mathsf{M} \in \mathcal{M}_A$ gibt es eine Turing-Maschine $\mathsf{M}' \in \mathcal{M}$ mit $\mathsf{L}_\mathsf{F}(\mathsf{M}) = \mathsf{L}_H(\mathsf{M}')$.
(b) Für jede Turing-Maschine $\mathsf{M} \in \mathcal{M}$ gibt es eine Turing-Maschine $\mathsf{M}' \in \mathcal{M}_A$ mit $\mathsf{L}_H(\mathsf{M}) = \mathsf{L}_F(\mathsf{M}')$.
(c) Für jede Turing-Maschine $\mathsf{M} \in \mathcal{M}_A$ gibt es eine Turing-Maschine $\mathsf{M}' \in \mathcal{M}$ mit $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}_F(\mathsf{M}')$.
(d) Für jede Turing-Maschine $\mathsf{M} \in \mathcal{M}$ gibt es eine Turing-Maschine $\mathsf{M}' \in \mathcal{M}_A$ mit $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}_F(\mathsf{M}')$.
(e) Für jede Turing-Maschine $\mathsf{M} \in \mathcal{M}$ gibt es eine Turing-Maschine $\mathsf{M}' \in \mathcal{M}$ mit $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}_H(\mathsf{M}')$.

Lösungsskizze

(a) Füge einen neuen Fangzustand q_t ein und mache die Transitionsfunktion total für Nichtendzustände. Formal:

$$\mathsf{Q}' = \mathsf{Q} \cup \{q_t\} \qquad \delta' = \delta \cup \{(q_t, a, q_t, a, N) \mid a \in \Gamma\} \cup \{(q, a, q_t, a, N) \mid a \in \Gamma \wedge q \in \mathsf{Q} \setminus \mathsf{F} \wedge \delta(q, a) \text{ ist undefiniert}\}$$

(b) Füge einen neuen Endzustand q_f ein und mache die Transitionsfunktion total (außer für q_f). Formal:

$$\mathsf{Q}' = \mathsf{Q} \cup \{q_f\} \qquad \mathsf{F}' = \{q_f\} \qquad \delta' = \delta \cup \{(q, a, q_f, a, N) \mid a \in \Gamma \wedge q \in \mathsf{Q} \wedge \delta(q, a) \text{ ist undefiniert}\}$$

(c) Folgt sofort, da $\mathcal{M}_A \subsetneq \mathcal{M}$.

(d) Lösche alle ausgehenden Kanten von Endzuständen. Formal:

$$\delta' = \delta \setminus \mathsf{F} \times \Gamma \times \mathsf{Q} \times \Gamma \times \{L, R, N\}$$

(e) Lösche alle ausgehenden Kanten von Endzuständen und mache die Transitionsfunktion total für Nichtendzustände. Formal:

$$\mathsf{Q}' = \mathsf{Q} \cup \{q_t\}$$
$$\delta' = (\delta \cup \{(q_t, a, q_t, a, N) \mid a \in \Gamma\} \cup \{(q, a, q_t, a, N) \mid a \in \Gamma \wedge q \in \mathsf{Q} \setminus \mathsf{F} \wedge \delta(q, a) \text{ ist undefiniert}\}) \setminus \mathsf{F} \times \Gamma \times \mathsf{Q} \times \Gamma \times \{L, R, N\}$$

AUFGABE 7.5. (*2-PDA und TMs*) Stufe D

Ein 2-PDA ist ein PDA, der 2 Stacks zur Verfügung hat. In jedem Schritt kann der PDA in Abhängigkeit vom aktuellen Zustand, dem gelesenen Eingabezeichen und den Symbolen, die oben auf den beiden Stacks liegen, in einen neuen Zustand wechseln und jeden der Stacks, wie im Fall eines gewöhnlichen, PDAs modifizieren.

(a) Geben Sie eine formale Definition für 2-PDAs an.

(b) Geben Sie eine Sprache an, die von einem 2-PDA, aber von keinem 1-PDA akzeptiert wird. Begründen Sie (informell), wie Sie einen 2-PDA für diese Sprache bauen könnten.

(c) Beschreiben Sie informell, wie Sie einen PDA P in eine Turingmaschine M übersetzen können, sodass $\mathsf{L}_\varepsilon(P) = \mathsf{L}_F(\mathsf{M})$ gilt.
Tipp: Verwenden Sie eine TM mit zwei Bändern.

(d) Überlegen Sie wie, Sie das Verfahren aus (c) erweitern können, sodass ein 2-PDA A in eine Turingmaschine M übersetzt werden kann.

(e) Geben Sie eine formale Übersetzung von einer Turingmaschine M in einen 2-PDA A an, sodass $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}_F(A)$. Beschreiben Sie dabei ihre Idee auch informell.

(f) Zeigen Sie unter Verwendung der vorherigen Ergebnisse, dass jeder k-PDA A ($k \geq 3$) von einem 2-PDA A' simuliert werden kann, d.h. $\mathsf{L}_\varepsilon(A) = \mathsf{L}_\varepsilon(A')$. Somit können beliebig viele Stacks immer durch genau zwei Stacks simuliert werden.

Lösungsskizze

(a) $M = (\mathsf{Q}, \Sigma, \Gamma, \mathsf{q}_0, (Z_0, Z_0), \delta, \mathsf{F})$ mit $\delta : \mathsf{Q} \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^2 \to \mathcal{P}_e(\mathsf{Q} \times (\Gamma^*)^2)$.

(b) Ein 2-PDA kann die Sprache $\mathsf{L} = \{\mathsf{a}^n\mathsf{b}^n\mathsf{c}^n \mid n \in \mathbb{N}_0\}$ akzeptieren, indem er zuerst alle as auf den ersten, dann alle bs auf den zweiten Stack schiebt, dann für jedes c jeweils ein Zeichen von beiden Stacks popt und schließlich nur dann akzeptiert, falls nach Lesen der Eingabe beide Stacks leer sind. Wie aber in der Vorlesung gezeigt worden ist, ist L nicht kontextfrei und somit existiert kein PDA, der L akzeptiert.

(c) Sei $P = (\mathsf{Q}_P, \Sigma, \Gamma, \mathsf{q}_0, \perp, \delta_P)$ ein PDA, der mit leerem Keller akzeptiert. Die TM simuliert den Stack des PDA auf einem zweiten Band. Das zweite Band wird dazu zu Beginn mit $\perp$ initialisiert. Das rechteste Symbol des zweiten Bandes entspricht dann dem obersten Symbol auf dem Stack des simulierten PDA. Pusht der PDA Zeichen auf den Stack, schreibt die TM dieselben Zeichen rechts auf das zweite Band. Entfernt der PDA ein Stacksymbol, so überschreibt die TM das rechteste Symbol auf dem zweiten Band mit $\square$ und bewegt den Kopf nach links.

Wird ein Zeichen im PDA eingelesen, so bewegt die TM zusätzlich den Kopf im ersten Band um eins nach rechts. Bei ε-Übergängen bleibt der Kopf im ersten Band stehen. Sobald die TM auf beiden Bändern $\square$ liest, geht sie in den akzeptierenden Zustand und hält.

(d) Wir verwenden anstatt einer 2-Band eine 3-Band TM, um beide Stacks auf separaten Bändern zu simulieren.

(e) Idee: Jeder 2-PDA kann eine TM mit einem Band simulieren, indem der erste Stack alle Symbole links vom Lesekopf und der zweite Stack alle Symbole rechts und unter dem Lesekopf speichert. Sei $\mathsf{M} = (\mathsf{Q}, \Sigma, \Gamma, \delta, \mathsf{q}_0, \square, \mathsf{F})$ die zu simulierende TM. Wir können annehmen, dass M deterministisch ist. Formal konstruieren wir A wie folgt:

$$A = (\mathsf{Q} \cup \{q_{\mathsf{read}}, q_{\mathsf{rev}}\}, \Sigma, \Gamma \cup \{\perp\}, q_{\mathsf{read}}, (\perp, \perp), \delta, \mathsf{F})$$

Zuerst schreiben wir die Eingabe auf den linken Stack:

$$q_{\mathsf{read}}(X,\bot) \xrightarrow{\varepsilon} q_{\mathsf{rev}}(X,\bot) \qquad q_{\mathsf{read}}(X,\bot) \xrightarrow{a} q_{\mathsf{read}}(aX,\bot) \qquad \text{für alle } a \in \Sigma, X \in \Sigma \cup \{\bot\}$$

Dann drehen wir die Eingabe durch Verschieben auf den rechten Stack um:

$$q_{\mathsf{rev}}(\bot,Y) \xrightarrow{\varepsilon} \mathsf{q}_0(\bot,Y) \qquad q_{\mathsf{rev}}(X,Y) \xrightarrow{\varepsilon} q_{\mathsf{rev}}(\varepsilon,XY) \qquad \text{für alle } X \in \Sigma$$

Wir simulieren dann wie folgt die TM:

$$\begin{aligned} q(X,Y) &\xrightarrow{\varepsilon} p(\varepsilon,XZ) && \text{für alle } X \in \Gamma, \delta(q,Y) = (p,Z,L) \\ q(X,Y) &\xrightarrow{\varepsilon} p(X,Z) && \text{für alle } X \in \Gamma, \delta(q,Y) = (p,Z,N) \\ q(X,Y) &\xrightarrow{\varepsilon} p(ZX,\varepsilon) && \text{für alle } X \in \Gamma, \delta(q,Y) = (p,Z,R) \end{aligned}$$

Schlussendlich müssen wir bei Bedarf auf $\bot$ immer ein $\Box$ erzeugen können:

$$q(\bot,X) \xrightarrow{\varepsilon} q(\Box\bot,X) \qquad q(X,\bot) \xrightarrow{\varepsilon} q(X,\Box\bot) \qquad \text{für alle } q \in \mathsf{Q}, X \in \Gamma \cup \{\bot\}$$

(f) Eine $(k+1)$-Band-TM kann jeden k-PDA simulieren. Jede $k+1$-Band-TM kann von einer 1-Band-TM simuliert werden. Jede 1-Band-TM kan von einem 2-PDA simuliert werden. Das Ergebnis folgt nun transitiv.