Theoretische Informatik und Logik
Übungsblatt 1 (2017W)
Lösungen

Aufgabe 1.1 Sei die Sprache L gegeben durch folgende induktive Definition: L ist die kleinste Menge, sodass

- $\underline{\$} \in L$
- $w \in L \Rightarrow xwx \in L$ für $x \in \{\underline{\mathtt{a}}, \underline{\mathtt{b}}\}$

(*Anmerkung*: L beschreibt Palindrome über $\{\underline{\mathtt{a}}, \underline{\mathtt{b}}\}$, bei welchen das Symbol $\underline{\$}$ als Trennzeichen zwischen einem Wort und seinem Spiegelbild fungiert.)

a) Geben Sie eine deterministische Turingmaschine M an, welche die Sprache L akzeptiert, und erläutern Sie (jeweils) auch kurz verbal die Arbeitsweise Ihrer Maschine. Es steht Ihnen dabei frei, ob Sie das auf Folie 26 definierte Modell (mit einem Band) oder das auf Folie 72 definierte Modell (mit zwei Bändern, einem Eingabe- und einem Arbeitsband, wobei M in diesem Fall die Kellerautomatenbedingung erfüllen soll) verwenden.

b) Erweitern Sie Ihre unter a) gefundene Maschine so, dass sie $L' = \{w\underline{\$}w^r \mid w \in \Sigma^*\}$, wobei $\Sigma = \{\underline{\mathtt{a}}_i \mid 1 \leq i \leq n\}$, akzeptiert.

Lösung
<u>Variante 1:</u>

a) Wir definieren eine (deterministische) Turingmaschine

$$M = (\{q_i \mid 0 \leq i \leq 7\}, \{\underline{\mathtt{a}}, \underline{\mathtt{b}}, \underline{\$}\}, \{\underline{\mathtt{a}}, \underline{\mathtt{b}}, \underline{\$}, B\}, \delta, q_0, B, \{q_7\})$$

wobei

δ	$\underline{\mathtt{a}}$	$\underline{\mathtt{b}}$	B	$\underline{\$}$
q_0	(q_1, B, R)	(q_4, B, R)		$(q_6, \underline{\$}, R)$
q_1	$(q_1, \underline{\mathtt{a}}, R)$	$(q_1, \underline{\mathtt{b}}, R)$	(q_2, B, L)	$(q_1, \underline{\$}, R)$
q_2	(q_3, B, L)			
q_3	$(q_3, \underline{\mathtt{a}}, L)$	$(q_3, \underline{\mathtt{b}}, L)$	(q_0, B, R)	$(q_3, \underline{\$}, L)$
q_4	$(q_4, \underline{\mathtt{a}}, R)$	$(q_4, \underline{\mathtt{b}}, R)$	(q_5, B, L)	$(q_4, \underline{\$}, R)$
q_5		(q_3, B, L)		
q_6			(q_7, B, S)	
q_7				

Idee:

q_0: Abhängig vom gelesenen Eingabesymbol wird in folgende Zustände gewechselt:
q_1: Ein Symbol $\underline{\mathtt{a}}$ wurde links gelesen und gelöscht; nun wandert die Maschine nach rechts bis zum ersten Blank.
q_2: Findet sich auch ganz rechts ein Symbol $\underline{\mathtt{a}}$, so wird dieses ebenfalls gelöscht.
q_3: Die Maschine wandert zurück bis zum ersten Blank links.
q_4: Ein Symbol $\underline{\mathtt{b}}$ wurde links gelesen und gelöscht; nun wandert die Maschine nach rechts bis zum ersten Blank.
q_5: Findet sich auch ganz rechts ein Symbol $\underline{\mathtt{b}}$, so wird dieses ebenfalls gelöscht.
q_6: Die Maschine erreicht diesen Zustand nur, wenn vor und nach dem Symbol $\underline{\$}$ ein Blank ist, und begibt sich daraufhin in den Endzustand q_7.

b) Für die Sprache L' definieren wir eine (deterministische) Turingmaschine

$$M = (\{q_i, q_i' \mid 0 \le i \le n{+}1\}, \{\underline{\mathtt{a}}_i \mid 1 \le i \le n\} \cup \{\underline{\$}\}, \{\underline{\mathtt{a}}_i \mid 1 \le i \le n\} \cup \{\underline{\$}\} \cup \{B\}, \delta, q_0, B, \{q_{n+1}'\})$$

wobei

$$\begin{array}{rl}
\delta = & \{(q_0, \underline{\mathtt{a}}_k; q_k, B, R) \mid 1 \le k \le n\} \\
\cup & \{(q_0, \underline{\$}; q_{n+1}, \underline{\$}, R)\} \\
\cup & \{(q_0', x; q_0', x, L) \mid x \in \{\underline{\mathtt{a}}_i \mid 1 \le i \le n\} \cup \{\underline{\$}\}\} \\
\cup & \{(q_0', B; q_0, B, R)\} \\
\cup & \{(q_k, x; q_k, x, R) \mid 1 \le k \le n, x \in \{\underline{\mathtt{a}}_i \mid 1 \le i \le n\} \cup \{\underline{\$}\}\} \\
\cup & \{(q_k, B; q_k', B, L)\} \\
\cup & \{(q_k', \underline{\mathtt{a}}_k; q_0', B, L) \mid 1 \le k \le n\} \\
\cup & \{(q_{n+1}, B; q_{n+1}', B, S)\}
\end{array}$$

Variante 2:

a) Wir definieren eine (deterministische) Turingmaschine

$$M = (\{q_0, q_f\}, \{\underline{\mathtt{a}}, \underline{\mathtt{b}}, \underline{\$}\}, \{A, B, C, Z_0\}, \delta, q_0, \{Z_0, Z_1, Z_2\}, B, \{q_f\})$$

in Normalform, welche L akzeptiert; die Übergangsfunktion δ kann z.B. folgendermaßen definiert werden:

$$\begin{array}{rl}
1: & \delta(q_0, \underline{\mathtt{a}}, B) = (q_0, A, R, R) \\
2: & \delta(q_0, \underline{\mathtt{b}}, B) = (q_0, C, R, R) \\
3: & \delta(q_0, \underline{\$}, B) = (q_0, B, R, L) \\
4: & \delta(q_0, \underline{\mathtt{a}}, A) = (q_0, B, R, L) \\
5: & \delta(q_0, \underline{\mathtt{b}}, C) = (q_0, B, R, L) \\
6: & \delta(q_0, Z_2, Z_0) = (q_f, Z_0, S, R)
\end{array}$$

Erläuterung:

1 : Für jedes eingelesene Symbol $\underline{\mathtt{a}}$ wird ein Symbol A in den Keller (bzw. auf das Arbeitsband) geschrieben.

2 : Für jedes eingelesene Symbol $\underline{\mathtt{b}}$ wird ein Symbol C in den Keller (bzw. auf das Arbeitsband) geschrieben.

3, 4, 5 : Nachdem das Symbol $\underline{\$}$ eingelesen wird, wird nun für jedes eingelesene Symbol $\underline{\mathtt{a}}$ ein Symbol A bzw. für jedes eingelesene Symbol $\underline{\mathtt{b}}$ ein Symbol C im Keller (bzw. auf dem Arbeitsband) gelöscht.

6 : wird Z_2 auf dem Eingabeband (d.h., das Ende der Eingabe) erreicht, so sollte das Arbeitsband leer sein. M geht dann in den (einzigen) Endzustand q_f über und akzeptiert somit die Eingabe.

b) Für die Sprache L' definieren wir eine (deterministische) Turingmaschine

$$M = (\{q_0, q_f\}, \{\underline{\mathtt{a}}_i \mid 1 \le i \le n\} \cup \{\underline{\$}\}, \{X_i \mid 1 \le i \le n\} \cup \{B, Z_0\}, \delta, q_0, \{Z_0, Z_1, Z_2\}, B, \{q_f\})$$

in Normalform, welche die Kellerautomatenbedingung erfüllt; die Übergangsfunktion δ kann z.B. folgendermaßen definiert werden:

$$\begin{array}{rll}
12': & \delta(q_0, \underline{\mathtt{a}}_i, B) = (q_0, X_i, R, R) & \text{für } 1 \le i \le n \\
3: & \delta(q_0, \underline{\$}, B) = (q_0, B, R, L) & \\
45': & \delta(q_0, \underline{\mathtt{a}}_i, X_i) = (q_0, B, R, L) & \text{für } 1 \le i \le n \\
6: & \delta(q_0, Z_2, Z_0) = (q_f, Z_0, S, R) &
\end{array}$$

Aufgabe 1.2 Seien A, B, C und D Sprachen, die rekursiv aufzählbar sein können oder auch nicht. Wir wissen allerdings Folgendes:

- $A \leq B$
- $B \leq C$
- $D \leq C$

Geben Sie für jede der folgenden Aussagen an, ob sie

- *jedenfalls* zutrifft (unabhängig davon, um welche Probleme es sich bei A bis D handelt)
- *vielleicht* zutrifft (je nach dem worum es sich bei A bis D handelt)
- *keinesfalls* zutrifft (unabhängig davon, um welche Probleme es sich bei A bis D handelt)

Begründen Sie jeweils Ihre Antwort.

a) Ist A entscheidbar, so ist auch das Komplement von C entscheidbar.

b) Ist das Komplement von B nicht entscheidbar, so kann das Komplement von C entscheidbar sein.

c) Ist C rekursiv aufzählbar, so ist auch $B \cup D$ rekursiv aufzählbar.

d) Ist A rekursiv aufzählbar, so ist B entscheidbar.

e) Ist C nicht entscheidbar, so ist auch D nicht entscheidbar.

Lösung

a) **Vielleicht**. Entscheidbare Sprachen sind unter Komplement abgeschlossen. Ist also $\overline{C}$ entscheidbar, so muss auch C entscheidbar sein. Weiters sind Reduktionen transitiv, und nachdem $A \leq B$ und $B \leq C$ gilt auch $A \leq C$. Gibt es also eine Reduktion von A auf C, so muss C mindestens so schwierig wie A sein. C könnte aber durchaus auch ein schwierigeres Problem sein, also eines, welches nicht rekursiv aber rekursiv aufzählbar ist, oder auch eines, welches nicht einmal rekursiv aufzählbar ist.

b) **Keinesfalls**. Entscheidbare Sprachen sind unter Komplement abgeschlossen. Ist also $\overline{C}$ entscheidbar, so muss auch C entscheidbar sein. Die Reduktion und ein Algorithmus, der C entscheidet, können dazu verwendet werden, B zu entscheiden. Dies ist aber im Widerspruch zur Angabe (B kann nicht entscheidbar sein, nachdem $\overline{B}$ nicht entscheidbar ist).

c) **Jedenfalls**. Nachdem C rekursiv aufzählbar ist, müssen dies, aufgrund der möglichen Reduktion, jedenfalls auch B und D sein. Dann ist aber auch $B \cup D$ rekursiv aufzählbar: B (bzw. D) werde durch die Turingmaschine M_1 (bzw. M_2) erkannt. Um $B \cup D$ zu erkennen, entwerfen wir eine Turingmaschine M, die zuerst M_1 simuliert und dann M_2 simuliert. M wird genau dann akzeptieren, wenn eine der beiden Turingmaschinen akzeptiert.

d) **Vielleicht**. Eine solche Reduktion ist allerdings nur dann möglich, wenn A auch entscheidbar ist. Denn die Reduktion und ein Algorithmus, der B entscheidet, können dazu verwendet werden, A zu entscheiden.

e) **Vielleicht**. Ist C nicht rekursiv aufzählbar, so muss dies auf D nicht notwendigerweise zutreffen. Ein einfaches Gegenbeispiel: Sei $D = \{\}$, also die Leersprache, welche jedenfalls entscheidbar ist: Die Frage ob $w \in A$ ist, kann für jedes Wort w mit "nein" beantwortet werden. Sei nun M eine Turingmaschine, die D akzeptiert und C die unentscheidbare Sprache $L_{ne} = \{M \mid L(M) \neq \{\}\}$ (s. Folie 58). Dann können wir eine Reduktion von D auf C so konstruieren: Gegeben eine Instanz w von D, fragen wir ob M in L_{ne} ist. Nachdem $L(M) = \{\}$, ist die Antwort immer "nein".

Aufgabe 1.3 Geben Sie an, ob folgende Probleme (un)entscheidbar sind, und begründen Sie jeweils Ihre Antwort. Sofern möglich, verwenden Sie dafür den *Satz von Rice*. (Das Alphabet ist dabei jeweils $\Sigma = \{\underline{0}, \underline{1}\}$.)

a) Gibt es für die von einer Turingmaschine akzeptierte Sprache genau eine Turingmaschine, die sie akzeptiert?

b) Ist die von einer Turingmaschine akzeptierte Sprache das Komplement von Σ^*?

c) Ist die Codierung der Turingmaschine, welche die Sprache L akzeptiert, weniger als 1000 Symbole lang?

d) Ist das Komplement der von einer Turingmaschine akzeptierten Sprache endlich?

e) Ist die von einer Turingmaschine akzeptierte Sprache eine Teilmenge von Σ^*?

Lösung

a) **Entscheidbar**. Diese Eigenschaft trifft auf keine rekursiv aufzählbare Sprache zu, d.h., $P = \{\}$, und ist somit, nach dem Satz von Rice trivial, also entscheidbar. (Jede Sprache hat unendlich viele Turingmaschinen, die sie akzeptieren.)

b) **Unentscheidbar**, Satz von Rice: Es handelt sich um die Eigenschaft $P = \{\{\}\}$. Diese Eigenschaft kommt einer Sprache L zu, nämlich $L = \{\}$. Keine andere rekursiv aufzählbare Sprache ist in P, dementsprechend ist P nicht trivial, und damit nach dem Satz von Rice unentscheidbar.

(*Anmerkung:* Beachten Sie den Unterschied zwischen $P = \{\{\}\}$, der Eigenschaft die Leersprache zu sein und der leeren Eigenschaft $P = \{\}$, welche keiner rekursiv aufzählbaren Sprache zukommt, siehe auch a))

c) **Entscheidbar**. Die Menge aller Codes von Turingmaschinen, welche weniger als 1000 Zeichen umfassen ist endlich, und daher entscheidbar. Der Satz von Rice ist hier aber nicht anwendbar.

d) **Unentscheidbar**: $P = \{L \mid \overline{L} \text{ ist endlich}\}$ ist keine triviale Eigenschaft, denn es gilt z.B. $L = \Sigma^* \in P$ aber $\{\varepsilon\} \notin P$. Daher ist dieses Problem nach dem Satz von Rice unentscheidbar.

e) **Entscheidbar**. Die Eigenschaft eine Menge von Wörtern zu sein trifft auf alle rekursiv aufzählbaren Sprachen zu. Es handelt sich also um eine triviale Eigenschaft, welche nach dem Satz von Rice entscheidbar ist.

Aufgabe 1.4 Sind folgende Aussagen korrekt? Begründen Sie jeweils Ihre Antwort.

a) Ist L_1 regulär, so ist auch $L_1 \cup L_2$ regulär.

b) Ist $L_1 \cap L_2$ entscheidbar, so sind L_1 und auch L_2 entscheidbar.

c) Für jede rekursiv aufzählbare Sprache L gilt: $L \cup \overline{L}$ ist entscheidbar.

d) Jedes unentscheidbare Problem enthält eine entscheidbare Teilmenge.

e) Jede Teilmenge einer regulären Sprache ist regulär.

Lösung

a) Nein, diese Aussage trifft nur dann zu, wenn L_2 auch regulär ist. Wir wissen aber nichts über L_2. Ist z.B. $L_1 = \{\}$, also jedenfalls regulär, so ist $L_1 \cup L_2 = L_2$, welche selbst aber keinesfalls regulär sein muss.

b) Nein. Denn es könnte z.B. gelten, dass $L_1 \cap L_2 = \{\}$. Damit können wir aber keine Aussage über L_1 oder L_2 treffen.

c) Das ist jedenfalls richtig. Denn für jede Sprache L über einem Alphabet Σ gilt: $L \cup \overline{L} = \Sigma^*$, welches eine reguläre, also sicher entscheidbare Sprache ist.

d) Ja, denn für jede Sprache L gilt: $\{\} \subseteq L$. Bekanntlich ist die Leersprache $\{\}$ regulär, also jedenfalls entscheidbar.

e) Nein, dies gilt sicher nicht im Allgemeinen. Ein Gegenbeispiel: die Sprache $L = \{\underline{0}, \underline{1}\}^*$ ist regulär, deren Teilmenge $\{\underline{0}^n \underline{1}^n \mid n \geq 0\}$ aber sicher nicht.

Aufgabe 1.5 Sind folgende Sprachen regulär? Falls ja, so geben Sie einen entsprechenden deterministischen endlichen Automaten an; falls nein, so beweisen Sie dies mit Hilfe des Pumping Lemmas für reguläre Sprachen. (Wählen Sie mindestens zwei Unterpunkte.)
(*Hinweis*: Nur eine der folgenden drei Sprachen ist regulär.)

a) $\{u\underline{\#}v^r \mid u$ ist Binärdarstellung (ohne führende Nullen) von n, und v jene von $n+1, n > 0\}$.
(Wörter in dieser Sprache sind also z.B. $\underline{101\#011}$ und $\underline{1111\#00001}$)

b) Die Menge aller Binärstrings (also Wörter über $\Sigma = \{\underline{0}, \underline{1}\}$), welche, interpretiert als Binärzahl, nicht durch 5 teilbar sind (ohne Berücksichtigung führender Nullen).

c) $\{uu^r u \mid u \in \{\underline{a}, \underline{b}, \underline{c}\}^*\}$

(*Hinweis*: w^r bezeichnet das Spiegelbild von w.)

Lösung a) L ist sicher nicht regulär. Beweis indirekt.

Angenommen, L ist regulär. Sei dann m die Konstante aus dem Pumping Lemma und das gewählte Wort z.B.

$$w = \underline{1}^m \underline{\#} \underline{0}^m \underline{1}.$$

Dann gilt $w \in L$ und $|w| = 2m + 2 > m$.

Wir teilen nun w in xyz so auf, dass $|xy| \leq m$ und $|y| > 0$. Nachdem $|xy| \leq m$ und $w = \underline{1}^m \underline{\#} \underline{0}^m \underline{1}$, kann xy nur aus Symbolen $\underline{1}$ bestehen.

Nach dem Pumping Lemma muss aber $xy^i z \in L$ für alle $i \geq 0$ gelten.

Wenn wir nun z.B. $i = 2$ wählen, müsste auch $xy^2 z = \underline{1}^{m+|y|} \underline{\#} \underline{0}^m \underline{1}$ aus L sein, was aber nicht der Fall ist! Wir haben also einen Widerspruch gefunden, L kann somit keine reguläre Sprache sein.

b) L ist regulär. Um einen DEA zu konstruieren, der L akzeptiert, überlegen wir Folgendes:

Sei $z(x) =$ die durch die Darstellung x repräsentierte Binärzahl.

Dann gilt beispielsweise:

$z(\underline{1000}) = 8 = 3 \bmod 5$, $\quad z(\underline{10000}) = 8 * 2 = 1 \bmod 5$, $\quad z(\underline{10001}) = 8 * 2 + 1 = 2 \bmod 5$,

Wir erkennen folgendes **Prinzip:** Ist $z(x) = i \bmod 5$ dann ist

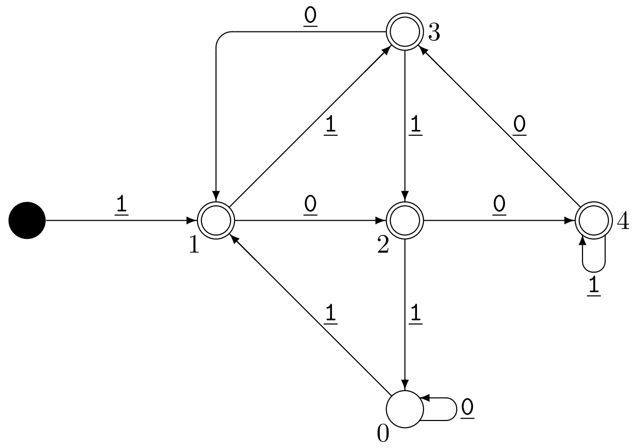
$z(x\underline{0}) = 2 * i \bmod 5$, sowie

$z(x\underline{1}) = 2 * i + 1 \bmod 5$.

Im DEA entspricht also der Zustand i der Information „die bisher eingelesene Zahl ist kongruent $i \bmod 5$“.

$\delta(i, \underline{0}) = (i \cdot 2 + 0) \bmod 5$
$\delta(i, \underline{1}) = (i \cdot 2 + 1) \bmod 5$

Demnach erhalten wir z.B. folgenden DEA für L:



c) Beweis indirekt. Angenommen, L ist regulär. Sei dann m die Konstante aus dem Pumping Lemma und das gewählte Wort z.B.

$$w = \underline{\mathtt{a}}^m \underline{\mathtt{b}}^{2m} \underline{\mathtt{a}}^{2m} \underline{\mathtt{b}}^m.$$

Dann gilt $w \in L$ und $|w| = 6m > m$.

Wir teilen nun w in xyz so auf, dass $|xy| \leq m$ und $|y| > 0$. Nachdem $|xy| \leq m$ und $w = \underline{\mathtt{a}}^m \underline{\mathtt{b}}^{2m} \underline{\mathtt{a}}^{2m} \underline{\mathtt{b}}^m$, kann xy nur aus Symbolen $\underline{\mathtt{a}}$ bestehen.

Nach dem Pumping Lemma muss aber $xy^iz \in L$ für alle $i \geq 0$ gelten.

Wenn wir nun z.B. $i = 0$ wählen, müsste auch $xy^0z = \underline{\mathtt{a}}^{m-|y|} \underline{\mathtt{b}}^{2m} \underline{\mathtt{a}}^{2m} \underline{\mathtt{b}}^m$ aus L sein, was aber nicht der Fall ist! Wir haben also einen Widerspruch gefunden, L kann somit keine reguläre Sprache sein.