Prof. Dr. J. Rothe Düsseldorf, 28.05.2019

Übung zur Vorlesung
Theoretische Informatik

Blatt 9, Abgabe: 04.06.2019 bis 10:30 Uhr
Besprechung: 11. und 12.06.2019
Verantwortlich: Marc Neveling

Hinweis:
Bitte achten Sie darauf, dass Sie Konfigurationenfolgen für Turingmaschinen entsprechend den Vorgaben in der Vorlesung angeben, insbesondere, dass Blank-Symbole ($\square$) *...*

(a) ...immer in dem Schritt angegeben werden, in dem sie auf das Band geschrieben werden, (auch wenn links oder rechts von ihnen keine anderen Zeichen auftreten) und

(b) ...immer in dem Schritt angegeben werden, in dem sie gelesen werden.

Aufgabe 1: Turingmaschinen I
Die Turingmaschine $M_1 = (\Sigma_1, \Gamma_1, Z_1, \delta_1, z_0, \square, F_1)$ sei definiert durch $\Sigma_1 = \{0, 1\}$, $\Gamma_1 = \{0, 1, \square, \$, £\}$, $Z_1 = \{z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_e\}$, $F_1 = \{z_e\}$, und δ_1 mit:

δ_1	z_0	z_1	z_2	z_3
0	$(z_1, \$, R)$	$(z_1, 0, R)$	$(z_3, \$, L)$	$(z_3, 0, L)$
1	$(z_4, £, R)$	$(z_1, 1, R)$	$(z_2, 1, L)$	$(z_3, 1, L)$
$\square$	$(z_e, \square, N)$	$(z_2, \square, L)$		
\$	$(z_7, \$, R)$	$(z_2, \$, L)$		$(z_0, \$, R)$
£	$(z_7, £, R)$			

δ_1	z_4	z_5	z_6	z_7
0				
1	$(z_4, 1, R)$	$(z_6, £, L)$	$(z_6, 1, L)$	
$\square$	$(z_5, \square, L)$			$(z_e, \square, N)$
\$	$(z_4, \$, R)$		$(z_6, \$, L)$	$(z_7, \$, R)$
£	$(z_5, £, L)$		$(z_0, £, R)$	$(z_7, £, R)$

(a) Geben Sie alle Konfigurationenfolgen von M_1 für die Eingabe 001001 an. Begründen Sie damit, ob $001001 \in L(M_1)$ gilt.

(b) Ordnen Sie die Zustände in Z_1 den folgenden Zustandsbeschreibungen (1) bis (9) zu:

	Zustand	Bedeutung	Absicht
(1)		Start einer Runde	Akzeptiert, wenn das Band leer ist; überschreibt die erste 0 mit \$ oder, falls alle 0en umgewandelt sind, die erste 1 mit £; leitet sonst finale Prüfung ein
(2)		Linke Blöcke fertig abgearbeitet	Prüft, ob alle 0 und 1 in £ und \$ umgewandelt wurden
(3)		Alle 0en und 1en wurden in £ und \$ umgewandelt	Akzeptiert die Eingabe
(4)		Linke 0 im linken 0er-Block mit \$ überschrieben	Läuft zum rechten Rand oder zum nächsten \$
(5)		Linke 1 im linken 1er-Block mit £ überschrieben	Läuft zum rechten Rand oder zum nächsten £
(6)		Rechten Rand des rechten 0er-Blockes erreicht	Überschreibt die rechte 0 im rechten 0er-Block mit \$
(7)		Rechten Rand des rechten 1er-Blockes erreicht	Überschreibt die rechte 1 im rechten 1er-Block mit £
(8)		Rechte 0 im rechten 0er-Block mit \$ überschrieben	Zurück zu rechtem \$ im linken Block, um nächste Runde zu starten
(9)		Rechte 1 im rechten 1er-Block mit £ überschrieben	Zurück zu rechtem £ im linken Block, um nächste Runde zu starten

(c) Betrachten Sie die folgenden Sprachen $L_i \subseteq \Sigma_1^*$ und geben Sie für jede davon an, ob $L(M_1) = L_i$ gilt. Geben Sie für jede Sprache L_i mit $L(M_1) \neq L_i$ ein Wort $w \in \Sigma_1^*$ an mit $w \in L_i$ und $w \notin L(M_1)$ ODER mit $w \notin L_i$ und $w \in L(M_1)$, um diese Ungleichheit zu belegen.

- $L_1 = \{0^x 1^y 0^x 1^y \mid x, y \geq 1\}$
- $L_2 = \{0^x 1^y 0^x 1^y \mid y \geq 0, x = 2y\}$
- $L_3 = \{0^x 1^y 0^x 1^y \mid x, y \geq 0\}$

(d) Ist M_1 deterministisch? Begründen Sie Ihre Antwort.

(e) Begründen Sie, warum M_1 kein LBA ist.

Aufgabe 2: Turingmaschinen II

(a) Geben Sie eine zu M_1 aus Aufgabe 1 äquivalente Turingmaschine M_2 an, deren Überführungsfunktion total ist (bei der also für jede mögliche Konfiguration mindestens eine Folgekonfiguration definiert ist).
Hinweis: Überlegen Sie sich, ob es für die Funktionalität der Turingmaschine relevant ist, ob auch Konfigurationen mit einem Endzustand Folgekonfigurationen haben.

(b) Ist es für jede Turingmaschine möglich, eine äquivalente Turingmaschine mit total definierter Überführungsfunktion zu finden? Falls nein, warum nicht? Falls ja, wie viele zusätzliche Zustände sind dazu maximal notwendig und wieso?

Aufgabe 3: LBAs
Konstruieren Sie einen deterministischen LBA, der die Sprache $\{a^n b^{2n} \mid n > 0\}$ akzeptiert. Geben Sie wie in Aufgabe 1 eine Beschreibung Ihrer Zustände als Tabelle an.
Anmerkung: LBAs besitzen nach Definition nur ein Band.

Aufgabe 4: Abschlusseigenschaften
Wir wissen, dass die Klasse REG der regulären Sprachen abgeschlossen unter Vereinigung ist. Geben Sie für jede der folgenden Behauptungen an, ob die Abgeschlossenheit von REG unter Vereinigung diese Behauptung impliziert.

(1) Wenn $L_1 \in \text{REG}$ und $L_2 \in \text{REG}$, dann gilt auch $(L_1 \cup L_2) \in \text{REG}$.

(2) Wenn $L_1 \in \text{REG}$ und $L_2 \notin \text{REG}$, dann gilt auch $(L_1 \cup L_2) \notin \text{REG}$.

(3) Wenn $L_1 \notin \text{REG}$ und $L_2 \notin \text{REG}$, dann gilt auch $(L_1 \cup L_2) \notin \text{REG}$.

(4) Wenn $L_1 \in \text{REG}$ und $(L_1 \cup L_2) \in \text{REG}$, dann gilt auch $L_2 \in \text{REG}$.

(5) Wenn $L_1 \in \text{REG}$ und $(L_1 \cup L_2) \notin \text{REG}$, dann gilt auch $L_2 \notin \text{REG}$.

(6) Wenn $L_2 \notin \text{REG}$ und $(L_1 \cup L_2) \in \text{REG}$, dann gilt auch $L_1 \notin \text{REG}$.

(7) Wenn $L_2 \in \text{REG}$ und $(L_1 \cup L_2) \notin \text{REG}$, dann gilt auch $L_1 \notin \text{REG}$.

(8) Wenn $L_2 \notin \text{REG}$ und $(L_1 \cup L_2) \notin \text{REG}$, dann gilt auch $L_1 \notin \text{REG}$.