Hausaufgabe der dritten Woche

Formale Grundlagen der Informatik I: Übungsgruppe 19

Louis Kobras
6658699
4kobras@informatik.uni-hamburg.de

Utz Pöhlmann
6663579
4poehlma@informatik.uni-hamburg.de

Aufgabe 3.3

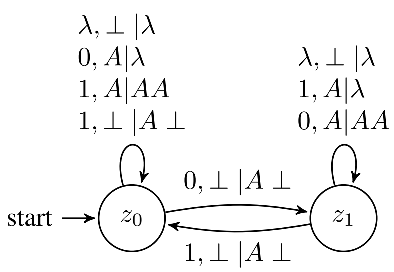
$w \in \{0,1\}*; \quad |w|_0 = |w|_1$
Angenommen, $\mathtt{L} \in \mathtt{REG}$, $\Rightarrow$Pumping-Lemma gilt.
Sei k die Zahl des Pumping-Lemmas. Wir betrachten die Zahl $a^k b^k$.
$uv = 1^s \quad | \; s \leq k \qquad\qquad (|uv| = s)$
$w = 1^t \quad |t \leq s; t \geq 1 \qquad (|v| = t)$

(3)

$$\begin{aligned} &\Rightarrow\; 1^{s+t^i} \\ &\Rightarrow\; 1^{s+t=k} \qquad |i = 1 \\ &\Rightarrow\; 1^{k+t} \qquad |i = 2 \\ &\Rightarrow\; 1^{k+t\cdot(i-1)} \qquad |i \in \mathbb{N} \end{aligned}$$

Für das Wort $1^{k+t\cdot(i-1)} a^k$ gilt $|1| = |w| \Leftrightarrow i = 1$.
Für zum Beispiel $i = 0$ gilt es *nicht*, da $k + t \cdot (0-1) = k + t \cdot (-1) = k - t$ ist und $\mathtt{L} \geq 1$ (laut (2)) sein muss.
$\Rightarrow$mind. 1 Zerlegung gefunden$\Rightarrow \mathtt{L} \notin \mathtt{REG}$.

Aufgabe 3.4



$L_\lambda(A) \subset M_1$:
Sei $\omega \in L_\lambda(A)$, so wurde ω akzeptiert. Nach der Konstruktion von A und da nur mit $0, \perp$ nach z_1 gewechselt werden kann, bleibt der Automat in z_0, solange mehr 1en als 0en gelesen wurden. Dies wird genau dann erreicht, wenn mit einer 0 gestartet wurde oder durch lesen von mehr 0en als 1en alle *A*s vom Stack genommen wurden.
Werden dann nach dem Leeren des Stacks weiter 1en gelesen, wird der Stack in z_1 mit *A*s gefüllt, welche durch das Lesen von 0en wieder gelöscht werden. Tritt dann wieder der Fall ein, dass mehr 1en als 0en gelesen werden, so geht der Automat zurück nach z_0, wo durch 0en wieder *A*s hochgezählt werden.
Dies wird solange wiederholt, bis nach dem Lesen von genau gleich vielen 0en und 1en der Keller leer und gleichzeitig das Wort zuende sind.
$\Rightarrow L(a) \subset M_1$.

$M_1 \subset L_\lambda(A)$:
Sei $\omega \in M_1$, so ist $|\omega|_0 = |\omega|_1$. ω ist eine Konkatenation von Worten $\omega_1, ..., \omega_n; n \in \mathbb{N}^+$, für die gilt: $|\omega_i|_0 = |\omega_i|_1; i > 1; \omega_i$ kann nicht geteilt werden, ohne dass diese Bedingungen verletzt werden.
Nach Konstruktion gilt: Für jedes ω_i wird bei einer gelesenen 1 ein *A* auf den Stack gepusht; für eine gelesene 0 wird ein *A* gepopt.
Wird $0, \perp$ gelesen, so findet ein Zustandswechsel von z_0 nach z_1 statt, wo dann für jede gelesene 0 ein *A* gepusht und für jede gelesene 1 ein *A* gepopt wird.
Bei $1, \perp$ findet dann wiederum ein Zustandswechsel von z_1 nach z_0 statt, wo wiederum nach o.g. Verfahren mit *A* auf dem Stack gearbeitet wird.
Bei $\lambda, \perp$, wird $\perp$ gepopt und der Automat ist fertig, also wird das Wort akzeptiert.
$\Rightarrow L_\lambda(A) \subset M_1 \quad \Rightarrow \quad M_1 = L_\lambda[A]$.

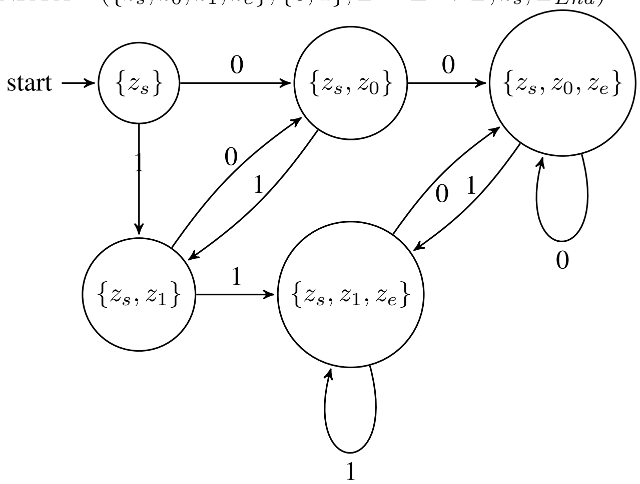
Aufgabe 3.5

3.5.1

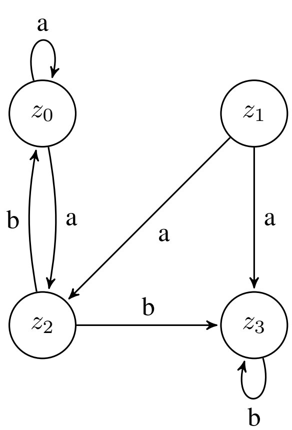
$$\hat{\delta}(\{z_s, z_0\}, 01) = \hat{\delta}(\{z_s, z_0, z_e\}, 1) = \hat{\delta}(\{z_s, z_1, z_e\}, \lambda) = \{z_s, z_1, z_e\}$$
$$\hat{\delta}(\{z_0, z_1\}, 01) = \hat{\delta}(\{z_e\}, 1) = \hat{\delta}(\{z_e\}, \lambda) = \{z_e\}$$

3.5.2

NFA A= $(\{z_s, z_0, z_1, z_e\}, \{0,1\}, Z \times \Sigma \to Z, z_s, Z_{End})$



3.5.3



Wir können weder Start- noch Endzustand rekonstruieren, da wir nicht wissen, ob das Wort vom Automaten akzeptiert wird, oder ob es sich überhaupt um ein vollständiges Wort handelt bzw. ob es ein Teilwort ist.
Wir können zudem die a-Kanten von z_2 und z_3 nicht bestimmen, ebenso wie die b-Kante von z_1, da keine entsprechenden Rechnungen vorliegen.
Ebenfalls können wir nicht wissen, ob es noch weitere Zustände gibt.
Wir wissen lediglich, dass z_0 keine b-Kante besitzt.