Formale Grundlagen der Informatik I

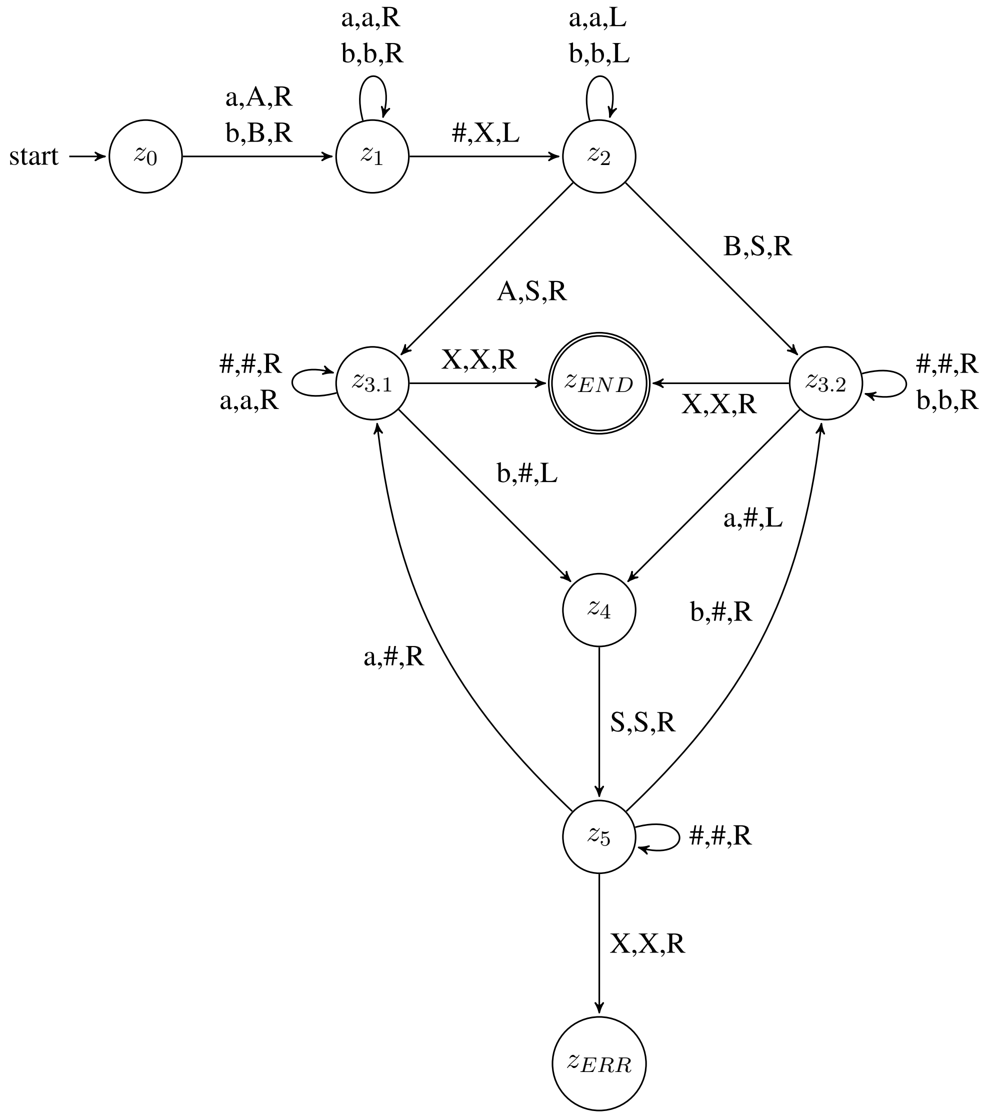
Abgabe der Hausaufgaben in Übungsgruppe 24
Abgabedatum: 22. Mai 2015

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

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

Philipp Quach
6706421
4quach@informatik.uni-hamburg.de

Aufgabe 5.4



$L \subseteq L(A)$:
Unser Wort fängt entweder mit a oder mit b an. Wir zeigen OBDA, dass es für a eine Erfolgsrechnung gibt (der Beweis für b ist analog hierzu): Wir ersetzen den Startbuchstaben durch den entsprechenden Großbuchstaben, um ihn später wiederzufinden. Dann gehen wir das Wort bis zum Ende durch und setzen uns einen Marker X. Danach gehen wir wieder an den Anfang zurück. Dort ersetzen wir das A durch ein S, um es als Wortanfang wiederzufinden.
***F*:** Danach suchen wir uns das erste b und ersetzen es durch eine #. Anschließend gehen zur zu dem S zurück und suchen uns das nächste Zeichen $\neq$ #. Sei es OBDA ein a (analog b). Verfahre nach *F*.
Wenn man innerhalb von *F* an das X gelangt, hat das Wort mindestens ein a (analog b) mehr als bs und akzeptiert, andernfalls nicht.

$L(A) \subseteq L$:
Wir markieren den ersten Buchstaben, schreiben an das Ende des Wortes ein X und gehen zurück an den Anfang. ***G*:** Dort ersetzen wir jetzt das A (analog B), welches sich dort befindet, durch ein S und löscht das nächste b (analog a). Danach geht man wieder zurück zu S und wiederholt *G*.
Sollte man während der Ausführung von *G* auf X treffen, hat man mindestens ein a (analog b) mehr als bs gefunden und akzeptiert das Wort damit, andernfalls nicht.

Aufgabe 5.5

$$G = \{V_N, V_T, P, S\}$$

$$L(G) := \{w \in V_T^* | S \overset{*}{\underset{G}{\Rightarrow}} w\}$$

$$V_T := \{a, b, c\}$$

$$V_N := \{S, A, B, C\}$$

$$S := S$$

$$P := \{S \to ABC,\ A \to aA|\lambda,\ B \to aBb|\lambda,\ C \to bCc|\lambda\}$$

$L_1 \underline{\subset} L(G) L(G)$ $(G\ erzeugt\ jedes\ Wort\ aus\ L_1)$

$$L_1 = \{a^j a^i b^i b^k c^k \mid i, j, k \geq 0\}$$

Es werden erst beliebig viele as (I), dann $a^x b^x$ (II) und zum Schluss $b^y c^y$ (III) gelesen $(x, y \in \mathbb{N}_0)$.
I ist gegeben durch $S \to ABC \ \wedge\ A \to aA|\lambda$
$\Rightarrow$ für A kann man schreiben $a^n | n \in \mathbb{N}_0$ $(\hat{=} a^j)$
II ist gegeben durch $S \to ABC \ \wedge\ B \to aBb|\lambda$
$\Rightarrow$ für B kann man schreiben $a^n b^n | n \in \mathbb{N}_0$ $(\hat{=} a^i b^i)$
III ist gegeben durch $S \to ABC \ \wedge\ C \to bCc|\lambda$
$\Rightarrow$ für C kann man schreiben $b^n c^n | n \in \mathbb{N}_0$ $(\hat{=} b^k c^k)$

$$\Rightarrow (a^j)(a^i b^i)(b^k c^k) \ = \ a^{i+j} b^{i+k} c^k \quad | i, j, k \geq 0\ ist\ gegeben$$

$$\Rightarrow L_1 = L(G)\ \ und\ insb.$$

$$L_1 \underline{\subset} L(A)$$

$L(G) \underline{\subset} L_1$ $(Jedes\ von\ G\ erzeugte\ Wort\ ist\ auch\ in\ L_1$
Durch $S \to ABC$ wird jedes Wort in drei Teilworte aufgeteilt: A (I), B (II), und C (III).
I. Aus A wird aA oder λ $\ \Rightarrow a^n | n \in \mathbb{N}_0$, da auch beim ersten Aufruf gleich die λ-Variante genommen werden kann. $(\hat{=} a^j)$
II. Aus B wird aBb oder λ $\ \Rightarrow a^n b^n | n \in \mathbb{N}_0$, da λ von I. $(\hat{=} a^i b^i)$
III. Aus C wird bCc oder λ $\ \Rightarrow b^n c^n | n \in \mathbb{N}_0$, da λ. $(\hat{=} b^k c^k)$

$$\Rightarrow (a^j)(a^i b^i)(b^k c^k) \ = \ a^{i+j} b^{i+k} c^k | i, j, k \geq 0$$

$$L(G) = L_1\ \ undinsb.$$

$$L(G) \underline{\subset} L_1$$

Aufgabe 5.6

Teilaufgabe 1.

Die einfachste Möglichkeit wäre es, eine rechtsunendliche TM mit zwei Bändern zu benutzen. Dann kann das zweite Band wie ein Keller (i.F. 'Stack') behandelt werden:
Wird etwas auf den Stack gepusht, fährt der Automat zu der entsprechenden Position auf dem zweiten Band und schreibt, um danach wieder zurück zu der Stelle im Wort zu gehen, von der aus der Stack aufgerufen wurde. Wird etwas gepopt, so fährt der Automat wieder zu der entsprechenden Stelle und löscht; anschließend fährt er zurück zu der Stelle im Wort, von der aus der Stack aufgerufen wurde.
Da zu jedem PDA, der mit leerem Keller akzeptiert, ein PDA existiert, welcher mit Endzuständen akzeptiert, nimmt man nun jenen mit Endzustand. Da jede Kante des PDA mehrere Kanten der TM ersetzt wurde, nämlich eine für den Schreibvorgang, eine für den Löschvorgang und jeweils ausreichend, um zu den jeweiligen Stellen auf Band 2 zu gelangen, gelangt die TM letztendlich auch in einen Endzustand.

Teilaufgabe 2.

Es handelt sich wiederum um eine TM mit zwei rechtsunendliche Bändern.
Zunächst werden n as gelesen. Für jedes a wird ein A auf Band 2 geschrieben.
Für jedes b wird ein A auf Band 2 in ein B konvertiert.
Analog dazu wird für jedes gelesene c ein B in ein C konvertiert.
Nachdem das erste b gelesen wurde, kann kein A mehr geschrieben werden. Mit a^*b^*a fährt der Automat in einen Fehlerzustand Z_{ERROR}.
Analog für c in Abhängigkeit von b und a.
Daraus folgt ein Wortaufbau in der Form $a^*b^*c^*$.
Sollten nun beispielsweise bei einem b (analog dazu c) mehr bs folgen, als As auf Band 2 stehen, wird Z_{ERROR} aufgerufen.
$\Rightarrow a^n b^m c^o | n \geq m \geq o$
Am Ende wird nur noch geprüft, ob auf Band 2 ausschließlich C (und etwaige #) stehen. Wenn ja, so wurden alle As in Bs und alle Bs in Cs konvertiert, woraus folgt, dass es jeweils gleich viele gewesen sein müssen. $\Rightarrow a^n b^m c^o | n = m = o$.
Ist dies nicht der Fall, bestand das Wort aus mehr as als bs oder aus mehr bs als cs. Folglich wird das Wort nicht akzeptiert. Der Automat geht nur in einen Endzustand, wenn Band in der Form C^* mit beliebig vielen anschließenden # vorliegt und das Wort zu Ende gelesen wurde.