Formale Grundlagen der Informatik I

Abgabe der Hausaufgaben in Übungsgruppe 19

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

Aufgabe 5.5

$$G = \{V_N, V_T, P, S\}$$
$$L(G) := \{w \in V_T^* | S \underset{G}{\overset{*}{\Rightarrow}} w\}$$
$$V_T := \{a, b, c\}$$
$$V_N := \{S, A, B, C\}$$
$$S := S$$
$$P := \{S \rightarrow ABC,\ A \rightarrow aA|\lambda,\ B \rightarrow aBb|\lambda,\ C \rightarrow bCc|\lambda\}$$

$L_1 \subseteq L(G) L(G)\ (G\ erzeugt\ jedes\ Wort\ aus\ L_1)$

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

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 $S \rightarrow ABC\ \wedge\ A \rightarrow aA|\lambda$
 $\Rightarrow$ für A kann man schreiben $a^n | n \in \mathbb{N}_0\ \ (\hat{=} a^j)$
II ist gegeben durch $S \rightarrow ABC\ \wedge\ B \rightarrow aBb|\lambda$
 $\Rightarrow$ für B kann man schreiben $a^n b^n | n \in \mathbb{N}_0\ \ (\hat{=} a^i b^i)$
III ist gegeben durch $S \rightarrow ABC\ \wedge\ C \rightarrow bCc|\lambda$
 $\Rightarrow$ für C kann man schreiben $b^n c^n | n \in \mathbb{N}_0\ \ (\hat{=} b^k c^k)$

$$\Rightarrow (a^j)(a^i b^i)(b^k c^k)\ =\ a^{i+j} b^{i+k} c^k \quad | i, j, k \geq 0\ ist\ gegeben$$
$$\Rightarrow L_1 = L(G)\ \ und\ insb.$$
$$L_1 \subseteq L(A)$$

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

$$\Rightarrow (a^j)(a^i b^i)(b^k c^k)\ =\ a^{i+j} b^{i+k} c^k | i, j, k \geq 0$$
$$L(G) = L_1\ \ undinsb.$$
$$L(G) \subseteq L_1$$

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.