

Übungsblatt 8
Abgabe: 19. Dezember 2011

Aufgabe 8.1 (Abschlusseigenschaften kontextfreier Sprachen; 3+1+2+2 Punkte)

Wir betrachten die folgenden Sprachen über dem Alphabet $\Sigma = \{\$, a, b, c\}$:

$$L_1 = L(\$^*)$$

$$L_2 = \{a^i b^j c^k : i, j, k > 0, i \neq j\} \cup \{\$a^i b^j c^k : i, j, k > 0, j \neq k\}$$

$$L_3 = \{\$\}\{a^n b^n c^n : n > 0\}$$

Zeigen Sie:

- (a) L_2 und $L_1 \cup L_2$ sind deterministisch kontextfrei. Geben Sie hierzu deterministische Kellerautomaten an, die L_2 bzw. $L_1 \cup L_2$ (per Endzustand) erkennen.
- (b) L_3 ist nicht kontextfrei, also auch nicht deterministisch kontextfrei.
- (c) Deterministisch kontextfreie Sprachen sind nicht unter Konkatenation abgeschlossen. Zeigen Sie hierzu, dass $L_1 \cdot L_2$ nicht deterministisch kontextfrei ist.
- (d) Deterministisch kontextfreie Sprachen sind nicht unter Iteration abgeschlossen. Zeigen Sie hierzu, dass $(L_1 \cup L_2)^*$ nicht deterministisch kontextfrei ist.

Hinweis: Sie dürfen voraussetzen, dass die Sprache $L = \{\$a^i b^j c^k : i, j, k > 0\}$ regulär ist, und ferner, dass der Schnitt einer deterministisch kontextfreien Sprache mit einer regulären Sprache wiederum deterministisch kontextfrei ist.

Aufgabe 8.2 (Kellerautomaten vs Turingmaschinen; 3 Punkte)

Die Funktionsweise eines deterministischen Kellerautomaten (mit Endzuständen) lässt sich durch eine deterministische Turingmaschine simulieren. Skizzieren Sie die Funktionsweise einer solchen Turingmaschine und erläutern Sie, warum diese Turingmaschine dieselbe Sprache wie der Kellerautomat akzeptiert.

Aufgabe 8.3 (Turingmaschine, Berechnung einfacher Funktionen; 2+2+2+2 Punkte)

In dieser Aufgabe sollen Turingmaschinen konstruiert werden, die die folgenden einfachen numerischen Funktionen berechnen. Wir verwenden dazu das Alphabet $\Sigma = \{[, \# \}$, um die jeweilige Eingabe zu codieren. Eine natürliche Zahl $n \in \mathbb{N}$ wird durch n Striche dargestellt. Das Symbol $\#$ wird benutzt, um zwei Zahlen voneinander zu trennen. Konstruieren Sie für die folgenden Funktionen jeweils eine deterministische Turingmaschine $M = \langle Z, \Sigma, \Gamma, \delta, z_0, \square, E \rangle$, die die jeweilige Funktion berechnet: angesetzt auf die Eingabe, stoppt sie in einem Endzustand mit dem korrekten Ergebnis auf dem Band, wobei der Schreib-Lesekopf von M sich wieder über dem ersten nicht-leeren Zeichen befindet (d.h. M stoppt in einer Konfiguration der Gestalt $\square \dots \square z \mid \dots \mid \square \dots \square$ mit $z \in E$; H_M bezeichne dann das Wort, das auf dem Band steht, wenn M stoppt).

- (a) Addition zweier Zahlen (z.B. $H_M(\# \# \#) = \# \# \# \#$);

- (b) Subtraktion einer Zahl von einer größeren Zahl (z.B. $H_M(||\#||) = |)$;
- (c) Zwei Kopien einer eingegebenen Zahl, die durch $\#$ voneinander getrennt sind (z.B. $H_M(||) = ||\#||)$;
- (d) Vergleich zweier Zahlen: M gibt 0 zurück, falls beide Zahlen gleich groß sind; 1, falls die erste größer ist; 2, falls die zweite größer ist (z.B. $H_M(|\#|) = 2)$.

Geben Sie die jeweiligen Turingmaschinen in Form eines Flussdiagramms an.

Flussdiagramm-Darstellung für Turingmaschinen: Um die Übergangsfunktion von Turingmaschinen (TM) kompakt darzustellen, führen wir sogenannte Flussdiagramme ein. Ausgehend von elementaren TM lassen sich dann komplexere TM zusammensetzen. Man beschränkt sich hierbei auf die Darstellung der Ausführungsschritte der Teil-TM (insbesondere werden die einzelnen Zustände nicht konkret benannt). Wir nehmen im Folgenden an, dass alle konstruierten TM genau einen *Endzustand* besitzen. Ferner sei $\Gamma = \{a_0, \dots, a_n\}$ das Bandalphabet.

- Kleine Rechtsmaschine r : geht einen Schritt nach rechts und hält dann.

	r	z_0	a_0	z_e	a_0	R
Turingtafel:		$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
		z_0	a_n	z_e	a_n	R

- Kleine Linksmaschine l : geht einen Schritt nach links und hält dann.

	l	z_0	a_0	z_e	a_0	L
Turingtafel:		$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
		z_0	a_n	z_e	a_n	L

- Druckmaschine a für $a \in \Gamma$: schreibt das Symbol a auf das Band und hält dann.

	a	z_0	a_0	z_e	a	S
Turingtafel:		$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
		z_0	a_n	z_e	a	S

- Zusammensetzungen:

$M_1 \xrightarrow{a} M_2$ bedeutet, dass zuerst M_1 arbeitet. Hält M_1 auf einem Feld mit dem Symbol a an, wird M_2 gestartet.

$M_1 \rightarrow M_2$ bedeutet, dass zuerst M_1 arbeitet. Sobald M_1 anhält, wird M_2 gestartet.

$M_1 M_2$ ist eine Abkürzung für $M_1 \rightarrow M_2$.

$M_1 \xrightarrow{\neq a} M_2$ bedeutet, dass zuerst M_1 arbeitet. Hält M_1 auf einem Feld mit dem Symbol $\neq a$ an, wird M_2 gestartet.

Aus gegebenen TM können Flussdiagramme aufgebaut werden. Die Knoten dieser Flussdiagramme sind mit den Namen der TM bezeichnet. Die Kanten werden durch Pfeile der Form $\xrightarrow{a}$, $\rightarrow$ oder $\xrightarrow{\neq a}$ bezeichnet. Schleifen sind erlaubt. Eine der TM im Flussdiagramm ist durch einen Pfeil $\rightarrow M$ als Start-TM gekennzeichnet.

Beispiele: Die große Rechtsmaschine R (links dargestellt) geht einen Schritt nach rechts und anschließend so lange weiter nach rechts, bis sie ein $\square$ liest. Die rechts dargestellte TM geht zunächst einen Schritt nach links; steht dort kein a , so geht sie einen Schritt nach rechts, druckt a und geht danach nach rechts, bis sie ein $\square$ liest:

