

Formale Systeme

9. Übungsblatt

Hinweis

Folgende Aufgaben dienen der Selbstkontrolle und werden in der Übung nicht besprochen.

*) Wiederholen Sie die Begriffe Kellerautomat, endliche Übergangsrelation, Konfiguration, deterministischer Kellerautomat, nichtdeterministischer Kellerautomat und Akzeptanzbegriff für Kellerautomaten.

***) Geben Sie einen deterministischen Kellerautomaten \mathcal{A} an mit $L(\mathcal{A}) = \{wcw^R \mid w \in \{a, b\}^*\}$, wobei für $w = a_1 \dots a_n$ gilt $w^R = a_n \dots a_1$.

Aufgabe 1

Geben Sie Turingmaschinen an, die folgende Funktionen berechnen. Dabei gilt, analog zum Beispiel in der Vorlesung, für $n \in \mathbb{N} : n$ wird dargestellt als a^n , $a \in \Sigma$.

- Die Turingmaschine \mathcal{A}_0 berechnet die Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$, $n \mapsto 0$, d. h. das Eingabewort auf dem Band wird gelöscht.
- Die Turingmaschine \mathcal{A}_{succ} berechnet die Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$, $n \mapsto n + 1$.
- Für $i, n \in \mathbb{N}$ berechnet die Turingmaschine \mathcal{A}_n^i die Funktion $f_n^i : \mathbb{N}^n \rightarrow \mathbb{N}$, $(x_1, \dots, x_n) \mapsto x_i$. Es wird empfohlen, zunächst die Turingmaschine \mathcal{A}_4^2 anzugeben und diese dann zu \mathcal{A}_n^i zu verallgemeinern.
(Hinweis: $(3, 2, 4, 0)$ wird dargestellt als $\#aaa\#aa\#aaaa\#b\#$.)

Aufgabe 2

In dieser Aufgabe werden Zahlen *binär* kodiert. Beachten Sie, dass sie in der Vorlesung *unär* kodiert werden. Ganze Zahlen in Binärdarstellung seien wie folgt gegeben

$$code(n) = bin(n)\#i \text{ mit } i \in \{0, 1\}.$$

Dabei bedeutet $bin(n)$ die Binärdarstellung des Betrages der natürlichen Zahl n (ohne Vornullen), $i = 0$ bedeutet, dass die Zahl nicht negativ ist und $i = 1$, dass die Zahl negativ ist.

- Geben Sie eine Turingmaschine \mathcal{A}_0 an, die für eine beliebige Eingabe w (Wort über dem Alphabet $\Sigma = \{0, 1, \#\}$) entscheidet, ob w der vorgegebenen Binärdarstellung einer ganzen Zahl entspricht oder nicht.

- b) Geben Sie eine Turingmaschine \mathcal{A}_{2x} an, die die Funktion
 $f : \{0, 1, \#\}^* \rightarrow \{0, 1, \#\}^*$, $code(x) \mapsto code(2x)$ berechnet.
- c) Geben Sie eine Turingmaschine \mathcal{A}_{x-1} an, die die Funktion
 $f : \{0, 1, \#\}^* \rightarrow \{0, 1, \#\}^*$, $code(x) \mapsto code(x - 1)$ berechnet.
- d) Verknüpfen Sie die Turingmaschinen derart, dass die Ergebnismaschine die Funktion
 $f : \{0, 1, \#\}^* \rightarrow \{0, 1, \#\}^*$, $code(x) \mapsto code(2(x - 1))$ berechnet.

Hinweis: f ist eine partielle Funktion, da sie nur für solche Wörter $w \in \{0, 1, \#\}^*$ definiert ist, die der vorgegebenen Binärdarstellung entsprechen. Im Fall eines undefinierten Funktionswertes von f ist in der Turingmaschine eine Endlosschleife vorzusehen.