

Formale Sprachen und Komplexität, SS 18
Tutoriumsblatt 7

Besprechung am Mo/Di 11./12.06.2018

Aufgabe 7-1 Turingmaschine zum Berechnen einer Funktion auf Zahlen

Sei pre die (partielle) Vorgängerfunktion auf \mathbb{N} , also

$$pre : \mathbb{N} \rightarrow \mathbb{N} \\ n \mapsto \begin{cases} n - 1 & \text{falls } n > 0 \\ \text{undefiniert} & \text{falls } n = 0 \end{cases}$$

Sei $\Sigma = \{0, 1\}$ die Menge der Binärziffern. Die Elemente von Σ^* sollen als natürliche Zahlen in Binärdarstellung interpretiert werden. Um keine Ausnahmen behandeln zu müssen, seien führende Nullen erlaubt und ε eine zulässige Darstellung von $0 \in \mathbb{N}$.

- Geben Sie eine deterministische Turingmaschine $dec = (Z, \Sigma, \Gamma, \delta, r, _, \{s\})$ an, die pre berechnet. Das Blankzeichen ist $_$, der Startzustand ist r , der einzige Zustand, in dem M halten kann (und der einzige Endzustand) ist s .
- Wie verhält sich M mit einer Binärdarstellung von $0 \in \mathbb{N}$ als Eingabewort?

Aufgabe 7-2 Turingmaschine zum Erkennen einer formalen Sprache

Sei $\Sigma = \{a, b, X\}$ und $L = \{wXw \mid w \in \{a, b\}^*\}$. Die Sprache L enthält also Wörter mit genau einem X , das man wie ein Gleichheitszeichen für Wörter über $\{a, b\}$ auffassen kann:

$$X \in L, \quad aXa \in L, \quad abXab \in L, \quad abbabXabbab \in L, \quad \dots \\ aba \notin L, \quad aX \notin L, \quad aXb \notin L, \quad abXabb \notin L, \quad \dots$$

Die Sprache L ist nicht vom Typ 2 (kontextfrei), sie kann also nicht von einem Kellerautomaten akzeptiert werden.

Geben Sie eine deterministische Turingmaschine $M = (Z, \Sigma, \Gamma, \delta, cC, _, E)$ mit $T(M) = L$ an. Wenn die Maschine hält, soll das Band genau das Eingabewort enthalten, der Lese-/Schreibkopf auf dem ersten Zeichen dieses Eingabeworts stehen, und die Maschine im Zustand s_0 sein (wenn sie das Wort nicht akzeptiert) oder im Zustand s_1 (wenn sie das Wort akzeptiert).

Aufgabe 7-3 LOOP-Programme

Implementieren Sie die folgenden Code-Fragmente als LOOP- und GOTO-Programme.

- $x_0 := x_1 + x_2$
- $x_0 := x_1 * x_2$ (Multiplikation)
- $x_0 := x_1^{\{x_2\}}$ (Potenzierung)