Prof. Dr. J. Rothe Düsseldorf, 11.06.2019

Übung zur Vorlesung
Theoretische Informatik

Blatt 10, Abgabe: 18.06.2019 bis 10:30 Uhr
Besprechung: 25. und 26.06.2019
Verantwortlich: Marc Neveling

Aufgabe 1: Turingberechenbarkeit I
Gegeben sei die Turingmaschine $M = (\Sigma, \Gamma, Z, \delta, z_0, \square, F)$ mit $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \square\}$, $Z = \{z_0, z_1, z_2, z_3, z_A\}$, $F = \{z_A\}$ und der folgenden Überführungsfunktion δ:

δ	z_0	z_1	z_2	z_3
0	$(z_A, 0, N)$	$(z_1, 0, R)$		$(z_3, 0, L)$
1	$(z_1, 1, R)$	$(z_1, 1, R)$		$(z_3, 1, L)$
$\square$		$(z_2, 0, R)$	$(z_3, 0, L)$	$(z_A, \square, R)$

Die Turingmaschine M berechnet eine Funktion $f : \mathbb{N} \to \mathbb{N}$. Die Menge $\mathbb{N}$ bezeichnet dabei wie im Skript die natürlichen Zahlen inklusive der 0. Eingabewörter von M sind binär kodierte Zahlen *ohne* führende Nullen.

	Bedeutung	Absicht
z_0		
z_1		
z_2		
z_3		
z_A	Zum Wortanfang zurückgekehrt	Akzeptiert die Eingabe

(a) Geben Sie jeweils vollständige Konfigurationenfolgen von M für die Berechnung von $f(2)$, $f(5)$ und $f(0)$ an.

(b) Füllen Sie die Zustandsbeschreibung für die Zustände z_0, z_1, z_2 und z_3 aus und begründen Sie, warum der Zustand z_3 notwendig ist, damit M korrekt arbeitet.

(c) Geben Sie die Funktion f ausdrücklich an und begründen Sie Ihre Antwort.

Aufgabe 2: Turingberechenbarkeit II
Betrachten Sie die Funktion $g : \mathbb{N} \to \mathbb{N}$ mit

$$g(n) = \begin{cases} n-2 & \text{falls } n \geq 2, \\ \text{undefiniert} & \text{falls } n < 2. \end{cases}$$

(a) Geben Sie an, wie sich eine Turingmaschine verhalten muss, die eine Funktion berechnet, wenn ein Wort eingegeben wird, für das die Funktion undefiniert ist.

(b) Zeigen Sie, dass g Turing-berechenbar ist, indem Sie eine Turingmaschine M_g angeben, die g berechnet. Eingabewörter sind wieder binär kodiert und Sie dürfen annehmen, dass keine führenden Nullen vorkommen. Erläutern Sie knapp, wie Ihre Maschine M_g arbeitet und geben Sie für jeden Zustand dessen Bedeutung und Absicht an.

Hinweis: Achten Sie aber darauf, dass Ihre Maschine auch Ausgaben ohne führende Nullen erzeugen muss.

(c) Geben Sie jeweils Konfigurationenfolgen Ihrer Turingmaschine M_g für die Berechnung von $g(2)$, $g(9)$ und $g(1)$ an.

(d) Betrachten Sie M_g als Akzeptor. Geben Sie $L(M_g) \subseteq \{0,1\}^*$ formal als Menge von Wörtern an. Führende Nullen sind hier nicht mehr ausgeschlossen.

Aufgabe 3: LOOP-, WHILE-, GOTO-Programme
Verwenden Sie in der gesamten Aufgabe nur die elementaren Befehle, wie sie in der Definition des entsprechenden Programms aufgeführt sind.

(a) Zeigen Sie durch Angabe eines entsprechenden `WHILE`-Programms:

- `IF` $x_1 < c$ `THEN` P `END` ist `WHILE`-berechenbar.

(b) Zeigen Sie die folgende Aussage aus dem Skript durch Angabe eines `LOOP`-Programms:

- `IF` $x_1 = c$ `THEN` P `ELSE` P' `END` ist `LOOP`-berechenbar.

(c) Zeigen Sie durch Angabe eines `GOTO`-Programms, dass die Funktion $f : \mathbb{N}^2 \to \mathbb{N}$ mit $f(n_1, n_2) = n_1 \cdot n_2$ `GOTO`-berechenbar ist.

(d) Geben Sie ein `WHILE`-Programm an, das folgende Funktion f berechnet (Tipp: Nutzen Sie, dass $m^{k+1} = m^k + m^k(m-1)$ gilt):

$$f : \mathbb{N}^2 \to \mathbb{N},\ f(m,n) = m^n$$