Philipps Universität Marburg

Übungen zur „Theoretischen Informatik“, Sommersemester 2011

Prof. Dr. R. Loogen, M. Dieterle · Fachbereich Mathematik und Informatik · Marburg

Nr. 11, Abgabe: Dienstag, 28. Juni 2011 vor der Vorlesung

31. Turingmaschine zum Berechnen von Funktionen — 4 Punkte

Geben Sie eine Turingmaschine mit möglichst kurzer Turingtafel an, die die Funktion

$$translate : \begin{cases} \{0,1\}^* & \rightarrow \quad \{0,1,2,3,4,5,6,7\}^* \\ b & \mapsto \quad okt(b) \end{cases}$$

berechnet, wobei $okt(b)$ die Oktaldarstellung der Binärzahl b bezeichne. Erläutern Sie die Arbeitsweise Ihrer Maschine und geben Sie die Konfigurationsfolge für die Eingabe 11110 an.

32. Zusammenhang Spracherkennung — Funktionsberechnung — 4 Punkte

Zeigen Sie für $L \in \Sigma^*$:

$$\exists\, \mathfrak{A} \in DTM(\Sigma): \ L = L(\mathfrak{A}) = \{w \in \Sigma^* \mid q_0 w \not{b} \overset{*}{\vdash} \alpha q x \beta \text{ mit } q \in F\}$$

$$\curvearrowright \ \exists\, \tilde{\mathfrak{A}} \in DTM(\Sigma): \ f_{\tilde{\mathfrak{A}}} = \chi'_L, \qquad \text{wobei}$$

$$\chi'_L : \begin{cases} \Sigma^* & \dashrightarrow \quad \Sigma^* \\ w & \mapsto \quad \begin{cases} \varepsilon & \text{falls } w \in L \\ \text{nicht definiert,} & \text{falls } w \notin L \end{cases} \end{cases}$$

die semi-charakteristische Funktion von L' ist.

33. LOOP-Programm — 4 Punkte

Schreiben Sie ein LOOP-Programm zur Berechnung der Vorgängerfunktion

$$\text{pred} : \begin{cases} \mathbb{N} & \rightarrow \quad \mathbb{N} \\ n & \mapsto \quad \begin{cases} 0 & \text{falls } n = 0 \\ k & \text{falls } n = k+1 \end{cases} \end{cases}$$

Als Wertzuweisungen sind nur elementare Anweisungen der Form `X_i := 0` bzw. `X_i := X_j + 1` zugelassen.

Beweisen Sie die Korrektheit des Programms anhand der denotationellen Semantik.

Bitte wenden!

34. Zusatzaufgabe: **Fleißige Biber** 4 Punkte

Ein *Biber* ist eine Turingmaschine über $\Gamma = \{|, \flat\}$, die, auf das leere Band angesetzt, stoppt. Dabei wird $q\ a\ b$ stop als Turingzeile zugelassen.

Ein *fleißiger Biber* (busy beaver) ist ein Biber, der unter den Bibern gleicher Zustandszahl die maximale Strichzahl auf dem Band liefert.

Die Funktion $BB : \mathbb{N} \to \mathbb{N}$ ist definiert durch

$$BB(x) := \text{Anzahl der Striche eines fleißigen Bibers mit } x \text{ Zuständen}$$

(a) Geben Sie die Konfigurationsfolge an, die der durch nebenstehende Turingtafel definierte fleißige Biber mit 2 Zuständen durchläuft. / 1

q_0	$\flat$	$\vert$	R	q_1
q_0	$\vert$	$\vert$	L	q_1
q_1	$\flat$	$\vert$	L	q_0
q_1	$\vert$	$\vert$	stop	

(b) Zeigen Sie: $BB(x) < BB(x+1)$. / 2

(c) Folgern Sie aus dem folgenden Satz von Rado, dass BB nicht Turing-berechenbar ist. / 1

Ist $f : \mathbb{N} \to \mathbb{N}$ Turing-berechenbar mit $f(x) < f(x+1)$, so gilt für hinreichend großes x: $f(x) < BB(x)$.