Theorie der Informatik (CS 206)

M. Helmert, G. Röger
Frühjahrssemester 2014

Universität Basel
Fachbereich Informatik

Übungsblatt 8

Abgabe: 23. April

Hinweis: Für Abgaben, die ausschliesslich mit LaTeX erstellt wurden, gibt es einen Bonuspunkt. Bitte geben Sie nur die resultierende PS- oder PDF-Datei bzw. einen Ausdruck davon ab.

Aufgabe 8.1 (Syntaktischer Zucker für LOOP-Programme, 1+1+1 Punkte)

Geben Sie an, wie sich die folgenden syntaktischen Konstrukte für LOOP-Programme (mit der offensichtlichen Semantik) durch bekannte Konstrukte simulieren lassen. Sie dürfen dabei neben den Grundkonstrukten von LOOP-Programmen auch die zusätzlichen Konstrukte verwenden, die in Vorlesungsabschnitt 13.2 eingeführt wurden.

(a) **IF** $x_i = c$ **THEN** P **ELSE** P' **END**

(b) **IF** $x_i = x_j$ **THEN** P **END**

(c) **IF** $x_i < x_j$ **THEN** P **END**

Hierbei seien P und P' beliebige LOOP-Programme und $i, j, c \in \mathbb{N}_0$ beliebige natürliche Zahlen.

Aufgabe 8.2 (Simulation von WHILE-Programmen durch DTMs, 3 Punkte)

In dem Beweis, dass jede WHILE-berechenbare Funktion Turing-berechenbar ist (Vorlesungsabschnitt 13.3) wird skizziert, wie verschiedene elementare Operationen von WHILE-Programmen mit deterministischen Turingmaschinen simuliert werden können.
Geben Sie im Detail (unter Angabe eines Zustandsdiagramms) an, wie eine deterministische Turingmaschine die Operation $x_2 := x_2{+}1$ simulieren kann. Gehen Sie davon aus, dass die Eingabe der DTM die kodierte Zahlenfolge $\mathit{bin}(n_0)\texttt{\#}\mathit{bin}(n_1)\texttt{\#}\dots\texttt{\#}\mathit{bin}(n_m)$ über dem Eingabealphabet $\{\texttt{0}, \texttt{1}, \texttt{\#}\}$ ist. Sie dürfen das Bandalphabet beliebig wählen.
Beschreiben Sie bitte zusätzlich kurz in Worten, wie die Turingmaschine funktioniert.
Hinweis: Vergessen Sie nicht den Fall, in dem zusätzlicher Platz auf dem Band benötigt wird, da $\mathit{bin}(n_2)$ die Form $\texttt{11}\dots\texttt{1}$ hat.

Aufgabe 8.3 (Division in LOOP- und WHILE-Programmen, 1+1+1+1 Punkte)

Seien $\texttt{div}_1 : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ und $\texttt{div}_2 : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ die wie folgt definierten Varianten der *Divisionsfunktion*:

$$\texttt{div}_1(a,b) = \begin{cases} \lfloor \frac{a}{b} \rfloor & \text{falls } b \neq 0 \\ \text{undefiniert} & \text{falls } b = 0 \end{cases}$$

$$\texttt{div}_2(a,b) = \begin{cases} \lfloor \frac{a}{b} \rfloor & \text{falls } b \neq 0 \\ 0 & \text{falls } b = 0 \end{cases}$$

Zeigen Sie (durch Angabe eines entsprechenden LOOP- oder WHILE-Programms) oder widerlegen Sie (durch ein allgemeines Argument):

(a) $\texttt{div}_1$ ist WHILE-berechenbar.

(b) $\texttt{div}_2$ ist WHILE-berechenbar.

(c) $\texttt{div}_1$ ist LOOP-berechenbar.

(d) $\texttt{div}_2$ ist LOOP-berechenbar. Die Lösung orientiert sich an der Lösung als WHILE-Programm und nutzt aus, dass immer $\texttt{div}_2(a, b) \leq a$ gilt, so dass die Anzahl Iterationen der LOOP-Schleife auf jeden Fall ausreicht, um x_1 auf 0 zu reduzieren.

Sie dürfen hierbei die zusätzlichen Konstrukte aus Vorlesungsabschnitt 13.2 und aus Aufgabe 8.1 verwenden.