

Theoretische Informatik

Bearbeitungszeit: 17.06.2024 bis 23.06.2024, 16:00 Uhr

Besprechung: 24.06.2024, 10:30 Uhr in Hörsaal 5E

Abgabe: als PDF über das ILIAS
Gruppenabgaben möglich und erwünscht

Aufgabe 1 (WHILE-GOTO)10P

- (a) Zeigen Sie durch Angabe eines **WHILE**-Programms, dass die Funktion $f : \mathbb{N}^2 \rightarrow \mathbb{N}$, $f(m, n) = \lfloor \frac{m}{n} \rfloor$ **WHILE**-berechenbar ist.
- (b) Geben Sie die Belegung *aller* benutzten Variablen nach dem Programmdurchlauf an, für die Funktionswerte
 - (i) $f(2, 0)$
 - (ii) $f(1, 2)$
 - (iii) $f(4, 2)$
- (c) Zeigen Sie durch Angabe eines **GOTO**-Programms, dass die Funktion $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ mit $f(n_1, n_2) = n_1 \cdot n_2$ **GOTO**-berechenbar ist.

Verwenden Sie in der gesamten Aufgabe **nur** die elementaren Befehle, wie sie in der Definition des entsprechenden Programms aufgeführt sind.

Aufgabe 2 (primitive Rekursion)10P

Zeigen Sie, dass folgende Funktionen primitiv rekursiv sind. Verwenden Sie dafür die Normalschemata aus Kapitel 9 der Vorlesung.

Hinweis: Sobald Sie so festgestellt haben, dass eine der Funktionen primitiv rekursiv ist, dürfen Sie diese für die restlichen Funktionen wiederverwenden. Sie dürfen die Addition, Multiplikation, Exponentialfunktion sowie die Fakultät aus Kapitel 9 der Vorlesung als primitiv rekursiv voraussetzen und verwenden.

- (a) Die Vorgängerfunktion $V(x)$ (s. Kapitel 9):

$$V(x) = \begin{cases} 0 & \text{falls } x = 0 \\ x - 1 & \text{falls } x \geq 1. \end{cases}$$

- (b) Die modifizierte Differenz $\text{md}(x, y)$ (s. Kapitel 9):

$$\text{md}(x, y) = \begin{cases} 0 & \text{falls } x < y \\ x - y & \text{falls } x \geq y. \end{cases}$$

- (c) Die Signumsfunktion $S : \mathbb{N} \rightarrow \mathbb{N}$ mit

$$S(x) = \begin{cases} 0, & \text{falls } x = 0 \\ 1, & \text{falls } x \geq 1 \end{cases}$$

- (d) Die vergleichende Funktion $\text{smaller} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, mit

$$\text{smaller}(x, y) = \begin{cases} 1, & \text{falls } x < y \\ 0, & \text{falls } x \geq y \end{cases}$$

Aufgabe 3 (De-Gödelisierung von \mathbb{Pr})12P

Gegeben sei folgende Gödelisierung einer primitiv-rekursiven Funktion. Welche Funktion wird hier berechnet?

$$\begin{aligned} & \text{PR}[s(0), \text{SUB}[\text{PR}[0, \text{SUB}[\text{PR}[\text{id} * |, \text{SUB}[s; \text{id}||| * ||](x|, x|, x|)](x|, x|)]; \\ & \text{id}||| * ||, \text{id}||| * ||](x|, x|, x|)](x|, x|); \text{id}||| * ||, \text{id}||| * ||](x|, x|, x|)](x|, x|). \end{aligned}$$

Aufgabe 4 (Ackermann-Funktion)8P

Betrachten Sie die Ackermann-Funktion $\alpha : \mathbb{N}^2 \rightarrow \mathbb{N}$, die in Kapitel 9 der Vorlesung definiert ist. Schreiben Sie in einer Programmiersprache Ihrer Wahl ein Programm, mit dem Sie $\alpha(m, n)$ berechnen. Nutzen Sie keine Library oder ähnliches, worin die Ackermann-Funktion schon fest implementiert ist.

- (a) Für welche (m, n) , $0 \leq m, n \leq 6$, findet Ihr Programm in angemessener Zeit eine Lösung?
- (b) Geben Sie Ihr Programm zusätzlich in einer separaten Datei ab.