Theoretische Informatik II

Prof. Dr. Christoph Kreitz / Holger Arnold
Universität Potsdam, Theoretische Informatik, Sommersemester 2006

Übung 3 (Version 2)

Was Sie als Vorbereitung auf diese Übung wissen sollten

- Aus welchen Grundfunktionen und welchen Operationen auf Funktionen können rekursive Funktionen gebildet werden? Wodurch unterscheiden sich rekursive und primitiv-rekursive Funktionen?
- Welche Programmiertechniken für primitiv-rekursive Funktionen gibt es? Welche Beispiele für primitiv-rekursive Funktionen gibt es?
- Was versteht man unter der „Tupelfunktion“ $\langle \cdot, \cdot \rangle$ und ihren Umkehrfunktionen?
- Gibt es berechenbare totale Funktionen, die nicht primitiv-rekursiv sind?

Aufgabe 3.1

Sei $h : \mathbb{N}^2 \to \mathbb{N}$ definiert durch $h = \mathrm{Pr}[c_0^1, g]$ mit $g = \mathrm{Pr}[pr_1^2, s \circ pr_4^4]$. Berechnen Sie schrittweise $h(2,0)$, $h(2,1)$, und $h(2,2)$. Welche Funktion stellt h dar?

Aufgabe 3.2

Beweisen Sie folgenden Satz: Wenn f, g und t primitiv-rekursive Funktionen sind, dann ist auch die Funktion $h = \mathrm{Pr}_t[f, g]$ mit

$$\begin{aligned} h(x,0) &= f(x) \quad \text{und} \\ h(x,y+1) &= g(x,y,h(t(x),y)) \end{aligned}$$

primitiv-rekursiv (dieser Satz lässt sich leicht auf mehr als einen Parameter verallgemeinern).

Aufgabe 3.3

Die Fibonacci-Funktion F ist definert als

$$F(n) = \begin{cases} 0 & \text{wenn } n = 0, \\ 1 & \text{wenn } n = 1, \\ F(n-1) + F(n-2) & \text{wenn } n > 1. \end{cases}$$

Zeigen Sie, dass F eine primitiv-rekursive Funktion ist. Beachten Sie, dass Sie F nicht direkt mit dem Operator Pr ausdrücken können. Verwenden Sie stattdessen den Operator Pr_t für eine geeignete Transformation t.

Aufgabe 3.4

Für jedes $n \in \mathbb{N}$ sei

$$\begin{aligned} A_0(x) &= \begin{cases} 1 & \text{wenn } x = 0, \\ 2 & \text{wenn } x = 1, \\ x+2 & \text{sonst }, \end{cases} \\ A_{n+1}(0) &= 1, \\ A_{n+1}(x+1) &= A_n(A_{n+1}(x)). \end{aligned}$$

Zeigen Sie, dass jede der Funktionen A_n primitiv-rekursiv ist.