Einführung in die theoretische Informatik – Aufgabenblatt 11

Beachten Sie: Soweit nicht explizit angegeben, sind Ergebnisse stets zu begründen!

Hausaufgaben: Abgabe bis zum 06.07.2016 (Mittwoch) um 12:00

Aufgabe 11.1 — 2P+1P

Wir übersetzen primitiv rekursive und μ-rekursive Funktionen in Programme der Form definiert in TA10.3. Geben Sie hierzu für jedes Konstrukt der PR- und μ-rekursiven Funktionen den entsprechend erzeugten Code an.

(a) Geben Sie die Regeln für primitiv rekursive Funktionen an.

(b) Erweitern Sie (a) und geben Sie die Regeln für μ-rekursive Funktionen an.

Aufgabe 11.2 — 3P+2P

Sei P das folgende folgendes GOTO-Programm mit Stack mit Eingabevariablen x_1, x_2 und Ausgabevariable x_0 und $f_P(x_1, x_2)$ die von P berechnete Funktion:

```
0        PUSH x1 ;
1        PUSH x2 ;
2        x4 := 1;
3   M0 : x2 := POP;
4        x3 := EMPTY;
5        IF x3 = 1 GOTO M5 ;
6        x1 := POP;
7        IF x1 = 0 GOTO M1 ;
8        GOTO M2 ;
9   M1 : x2 := x2 + 1;
10       PUSH x2 ;
11       GOTO M0 ;
12  M2 : x5 := x1 − 1;
13       IF x2 = 0 GOTO M3 ;
14       GOTO M4 ;
15  M3 : PUSH x5 ;
16       PUSH x4 ;
17       GOTO M0 ;
18  M4 : PUSH x5 ;
19       PUSH x1 ;
20       x5 := x2 − 1;
21       PUSH x5 ;
22       GOTO M0 ;
23  M5 : x0 := x2 ;
24       HALT;
```

(a) Beschreiben Sie den Ablauf des Programms auf Eingabe $x_1 = 1, x_2 = 0$ explizit.

Stellen Sie eine Konfiguration in der Form $(l, x_0, x_1, x_2, x_3, x_4, x_5, s)$ dar, wobei s der Stackinhalt ist und l die Zeilennummer der Anweisungm, die als nächstes ausgeführt werden wird. Beschreiben Sie diesen in der Form $\langle s_1, s_2, \dots, s_k\rangle$, wobei s_1 die unterste Zahl auf dem Stack ist, s_k die oberste Zahle.

Die ersten Schritte des Programms bei Aufruf mit $x_1 = 1, x_2 = 0$ sind hiermit:

$$(0, 0, 1, 0, 0, 0, 0, \langle\rangle) \to (1, 0, 1, 0, 0, 0, 0, \langle 1\rangle) \to (2, 0, 1, 0, 0, 0, 0, \langle 1, 0\rangle) \to (3, 0, 1, 0, 0, 1, 0, \langle 1, 0\rangle) \to (4, 0, 1, 0, 0, 1, 0, \langle 1\rangle)$$

Beschreiben Sie entsprechend den Rest des Programmablaufs.

(b) Übersetzen Sie P zunächst in ein WHILE-Programm P' mit Stack, d.h. neben den Anweisung für ein WHILE-Programm dürfen auch die Anweisungen **PUSH**,**POP**,**EMPTY** verwendet werden, die sich wie im Fall eines GOTO-Programms mit Stack verhalten.

Übersetzen Sie P' dann in ein Programm P'' der Form3, welche in TA10.3 definiert wurde (C-ähnliche Prozeduren mit $\mathbb{N}$ als einzigem Datentyp, ohne Schleifen).

Argumentieren Sie anhand von P'', welche Funktion $f_P(x_1, x_2)$ durch P berechnet wird.

Aufgabe 11.3 **2P+2P+3P**

Sei $a(k, n)$ die Ackermann-Funktion. Für festes $k \in \mathbb{N}$ sei $a_k(n) := a(k, n)$.

(a) Geben Sie ein LOOP-Programm an, das $a_2(n)$ berechnet.

(b) Geben Sie ein LOOP-Programm an, das $a_3(n)$ berechnet.

(c) Zeigen Sie, dass zu jedem $k \in \mathbb{N}$ ein LOOP-Programm P_k existiert, das die Funktion $a_k(n)$ berechnet.

Aufgabe 11.4 **3P+2P**

Zeigen Sie jeweils explizit, dass die folgenden Funktionen primitiv rekursiv sind.

(a) $div(x, y) = x \text{ div } y$

(b) $mod(x, y) = x \text{ mod } y$

Verwenden Sie für die Definition der Funktionen nur die Basisfunktionen und das erweiterte Schema der primitiven Rekursion. Weiterhin dürfen Sie verwenden, dass die Addition, Multiplikation, die modifizierte Differenz und der beschränkten max-Operator primitiv rekursiv sind. Für alle andere Funktionen müssen Sie diese Eigenschaft zeigen.

Der Beweis muss direkt sein! Die Verwendung von LOOP-Programmen ist nicht gestattet.

Tutoraufgaben: Besprechung in KW27

Hinweis: Verwenden Sie für die Definition der folgenden Funktionen nur die Basisfunktionen und das erweiterte Schema der primitiven Rekursion. Weiterhin dürfen Sie verwenden, dass die Addition, Multiplikation, die modifizierte Differenz und der beschränkten max-Operator primitiv rekursiv sind. Für alle andere Funktionen müssen Sie diese Eigenschaft zeigen.

Der Beweis muss direkt sein! Die Verwendung von LOOP-Programmen ist nicht gestattet.

Aufgabe 11.1

Zeigen Sie, dass folgende Funktionen primitiv rekursiv sind:

(a)

$$\mathsf{ite}(x_0, x_1, x_2) = \begin{cases} x_1 & \text{falls } x_0 \neq 0 \\ x_2 & \text{sonst} \end{cases}$$

(b) Sei F_n die n-te Fibonacci-Zahl (mit $F_0 = 0$, $F_1 = 1$, $F_{n+2} = F_n + F_{n+1}$). Sei dann $f(n)$:

$$f(n) := F_n$$

Hinweis: Bei dieser Aufgabe dürfen Sie verwenden, dass die Cantorsche Paarungsfunktion c und die Umkehrungsfunktionen p_1, p_2 primitiv rekursiv sind.

Definieren Sie die Funktionen jeweils einmal nur unter Verwendung des normalen Schemas der primitiven Rekursion und zum Vergleich einmal unter Verwendung des erweiterten Schemas.

Aufgabe 11.2

Sei $P(x)$ mit $x \in \mathbb{N}$ ein primitiv rekursives Prädikat. Zeigen Sie, dass die Funktionen

(a) $f(n) := \forall x \leq n.\ P(x)$ und

(b) $g(n) := \min \{x \leq n \mid P(x)\}$ (wobei $min\ \emptyset = n + 1$)

primitiv rekursiv sind.

Aufgabe 11.3

Sei $a : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ die Ackermann-Funktion. Zeigen Sie, dass $f(m, n) := \log_2(a(m, n))$ nicht primitiv rekursiv ist, wobei $\log_2$ der abgerundete ganzzahlige Logarithmus zur Basis 2 ist (mit $\log_2 0 := 0$).