

Theoretische Informatik 2

Blatt 6 (Gewertete Aufgaben)

Abgabe: Dienstag, 26.05.2015 bis 12 Uhr

Besprechung: KW 21

1. (30%=10%+10%+10%) Geben Sie für folgende Funktionen LOOP-Programme an (für Aufgabe b) dürfen Sie Aufgabe a) und für Aufgabe c) dürfen Sie Aufgabe b) als Unterprogramm verwenden):

a) teilt : $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ mit

$$\text{teilt}(x, y) = \begin{cases} 0 & \text{falls } x = 0 \text{ oder } (x > 0 \text{ und } x \text{ teilt } y \text{ nicht}), \\ 1 & \text{falls } x > 0 \text{ und } x \text{ teilt } y. \end{cases}$$

b) tzahl : $\mathbb{N} \rightarrow \mathbb{N}$ mit

$$\text{tzahl}(x) = \sum_{i=0}^x \text{teilt}(i, x).$$

c) prim : $\mathbb{N} \rightarrow \mathbb{N}$ mit

$$\text{prim}(x) = \begin{cases} 1 & \text{falls } x \text{ eine Primzahl ist,} \\ 0 & \text{sonst.} \end{cases}$$

2. (20%=10%+10%) Stellen Sie folgende Funktionen als primitiv-rekursive Funktionen dar. Wenn Sie eine Funktion durch primitive Rekursion definieren, dann geben Sie bitte explizit die Funktionen g und h wie in Beispiel 14.1 im Skript an.

a) sub : $\mathbb{N}^2 \rightarrow \mathbb{N}$ mit

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

b) sdiff : $\mathbb{N}^2 \rightarrow \mathbb{N}$ mit

$$\text{sdiff}(x, y) = \begin{cases} x - y & \text{falls } x \geq y \\ y - x & \text{sonst.} \end{cases}$$

3. (20%=10%+10%)

a) Geben Sie ein WHILE-Programm an, das die folgende Funktion f berechnet:

$$f(x, y) = \begin{cases} 1 & \text{falls } x \geq 1 \text{ und } x \text{ teilt } y; \\ 0 & \text{falls } x \geq 1 \text{ und } x \text{ teilt } y \text{ nicht}; \\ \text{undefiniert} & \text{falls } x = 0. \end{cases}$$

b) Notieren Sie die folgende Funktion $\text{cube}(x)$ als μ -rekursive Funktion:

$$\text{cube}(x) = \begin{cases} 1 & \text{falls } x = n^3 \text{ für ein } n \in \mathbb{N}; \\ \text{undefiniert} & \text{sonst.} \end{cases}$$

4. (30%=15%+15%) GOTO-Programme bestehen aus Sequenzen von Anweisungen A_i , die jeweils durch eine Marke M_i eingeleitet werden, also Sequenzen der Art

$$M_1 : A_1; M_2 : A_2; \dots; M_k : A_k,$$

wobei mögliche Anweisungen A_i einer der folgenden Gestalt sind:

- Wertzuweisung, Addition: $x_i := x_j + c$ für $c \in \mathbb{N}$
- Wertzuweisung, Subtraktion: $x_i := x_j \div c$ für $c \in \mathbb{N}$
- Unbedingter Sprung: GOTO M_i
- Bedingter Sprung: IF $x_i = x_j$ THEN GOTO M_i
- Stoppanweisung: HALT

Die Semantik der GOTO-Programme ist auf die natürliche Art definiert; der berechnete Wert steht (wie bei WHILE-Programmen) in der Variablen x_0 nachdem die HALT-Anweisung durchgeführt wurde. Wir nennen eine Funktion GOTO-berechenbar, wenn es ein GOTO-Programm gibt, das diese Funktion berechnet.

Zeigen Sie, dass

- a) jede WHILE-berechenbare Funktion GOTO-berechenbar ist, und
- b) jede GOTO-berechenbare Funktion WHILE-berechenbar ist.

In beiden Richtungen genügt es, die Konstruktion anzugeben und kurz für Korrektheit zu argumentieren. Ein formaler Korrektheitsbeweis ist nicht erforderlich.

Hinweis. Für die Aufgaben 1-3 dürfen Sie alle Konstrukte verwenden, die bereits in Vorlesung oder Übung als primitiv rekursiv, μ -rekursiv, LOOP- oder WHILE-berechenbar nachgewiesen wurden.