1. Hausaufgabenblatt

Abgabe bis **23.06.2016 14:00 Uhr**
(in den Tutorien oder im Hörsaal ER 270 direkt vor der Vorlesung)
Alle Antworten sind zu begründen!
Antworten ohne Begründung erhalten 0 Punkte.

Aufgabe 1. Anzahl der Teiler einer natürlichen Zahl 3+3+3+1(P)

Sei $f\colon \mathbb{N} \to \mathbb{N}$ definiert als die Anzahl der ganzzahligen Teiler einer gegebenen natürlichen Zahl. Zur Vereinfachung setzen wir $f(0) = 0$, also

$$f(n) \coloneqq \begin{cases} 0, & \text{falls } n = 0, \\ \left| \{a \in \mathbb{N} \mid \exists b \in \mathbb{N}\colon a \cdot b = n\} \right|, & \text{sonst.} \end{cases}$$

Zum Beispiel gilt $f(4) = 3$, da 4 die drei Teiler $1, 2$ und 4 hat.

Seien außerdem zwei Funktionen $g, h\colon \mathbb{N} \to \mathbb{N}$ wie folgt definiert:

$$g(m) \coloneqq \max\{n \in \mathbb{N} \mid n \leq m^2 \text{ und } f(n) \leq m\}, \text{ und}$$
$$h(m) \coloneqq \min\{n \in \mathbb{N} \mid f(n) \geq m\}.$$

Beispielsweise gilt $g(4) = 15$ und $h(4) = 6$.

1. Geben Sie eine primitiv-rekursive Definition für f an.
2. Geben Sie eine primitiv-rekursive Definition für g an. Verwenden Sie dabei die Funktion f der ersten Teilaufgabe.
3. Geben Sie eine μ-rekursive Definition für h an. Verwenden Sie dabei die Funktion f der ersten Teilaufgabe und den μ-Operator.
4. Finden Sie eine obere Schranke für $h(m)$ in Abhängigkeit von m. Begründen Sie in höchstens drei Sätzen und mit nicht mehr als **90** Wörtern, dass h primitiv-rekursiv ist. (Eine informelle Beschreibung Ihrer Vorgehensweise genügt.)

Für jede der Teilaufgaben 1.–3.: Benutzen Sie nicht mehr als **drei** Hilfsfunktionen abgesehen von den unten aufgelisteten und geben Sie **keine** LOOP-, WHILE- oder GOTO-Programme an.

Zur Vereinfachung dürfen konstante Funktionen und Projektionsfunktionen weggelassen werden und ihre Funktionswerte direkt eingesetzt werden. Zum Beispiel kann die Addition add wie folgt definiert werden: $\mathsf{add}(0, y) = y$ und $\mathsf{add}(x + 1, y) = \mathsf{succ}(\mathsf{add}(x, y))$.

Hinweis: Sie dürfen davon ausgehen, dass die Grundfunktionen aus den Vorlesungsfolien (S. 61) und folgende Funktionen primitiv-rekursiv sind (andere Funktionen sind hiervon ausgenommen):

- Die Addition add mit $\mathsf{add}(x, y) = x + y$, die modifizierte Subtraktion modsub mit $\mathsf{modsub}(x, y) = \max(x - y, 0)$, die Multiplikation mult mit $\mathsf{mult}(x, y) = x \cdot y$,
- die Abfragefunktion isZero, die anzeigt, ob eine gegebene natürliche Zahl gleich 0 ist, d.h. $\mathsf{isZero}(0) = 1$ und $\mathsf{isZero}(x) = 0$ für $x > 0$,
- die Abfragefunktionen $\mathsf{isSmaller}(x, y)$ und $\mathsf{isEqual}(x, y)$, die anzeigen, ob x kleiner oder gleich y ist, d.h. $\mathsf{isSmaller}(x, y) = 1$, falls $x < y$, und $\mathsf{isSmaller}(x, y) = 0$ sonst, und $\mathsf{isEqual}(x, y) = 1$, falls $x = y$, und $\mathsf{isEqual}(x, y) = 0$ sonst.

- die Abfragefunktion $\mathsf{divisible}(x, y)$, die anzeigt, ob x durch y teilbar ist, d.h. $\mathsf{divisible}(x, y) = 1$, falls eine natürliche Zahl k existiert mit $k \cdot y = x$, und $\mathsf{divisible}(x, y) = 0$ sonst.

————————Lösungsskizze————————

1. Wir definieren zuerst eine zweistellige Hilfsfunktion $f_1 \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit $f_1(x, y) =$ Anzahl der Teiler von y die kleiner gleich x sind. Da die Menge der Teiler von y, die kleiner gleich $x+1$ sind genau die Menge der Teiler von y, die kleiner gleich x sind und der Zahl $x+1$ falls y durch $x+1$ teilbar ist, erfüllt f_1 folgende Gleichung:

$$f_1(x, y) = \begin{cases} 0, & \text{falls } x = 0 \\ f_1(x-1, y) & \text{falls } x > 0 \text{ und } y \text{ nicht durch } x \text{ teilbar ist,} \\ f_1(x-1, y) + 1 & \text{falls } x > 0 \text{ und } y \text{ durch } x \text{ teilbar ist.} \end{cases}$$

   Also kann man f_1 wie folgt primitiv-rekursiv definieren:

$$\begin{aligned} f_1(0, y) &= 0 \\ f_1(x+1, y) &= \mathsf{add}(f_1(x, y), \mathsf{divisible}(y, \mathsf{succ}(x))) \end{aligned}$$

   Laut der Definition von f ist $f(n)$ die Anzahl der ganzzahligen Teiler von n. Also kann man f wie folgt definieren: $f(n) = f_1(n, n)$.

   Da $\mathsf{add}, \mathsf{divisible}, \mathsf{succ}$, die konstanten Funktionen und Projektionen primitiv-rekursiv sind, und f_1 durch primitive Rekursionsvorschrift definiert ist, ist f_1 primitiv-rekursiv. Da außerdem f durch die Kompositionsvorschrift definiert ist, ist f auch pritimiv-rekursiv.

2. Wir definieren wiederum zuerst eine zweistellige Hilfsfunktion $g_1 \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, wobei $g_1(x, y)$ die größte natürliche Zahl $n \leq x$ bezeichnet mit $f(n) \leq y$. Beachte, dass, falls $f(x) > y$, dann gilt auch, dass $g_1(x, y)$ die größte natürliche Zahl ist, die kleiner gleich $x - 1$ ist und höchstens y Teiler hat. Ansonsten gilt $g_1(x, y) = x$. Somit erfüllt die Funktion g_1 folgende Gleichung:

$$g_1(x, y) = \begin{cases} 0 & \text{falls } x = 0 \\ g_1(x-1, y) & \text{falls } f(x) > y \\ x & \text{falls } f(x) \leq y \end{cases}$$

   Also, kann g_1 wie folgt primitiv-rekursiv definieren:

$$\begin{aligned} g_1(0, y) &= 0 \\ g_1(x+1, y) &= \mathsf{add}(\mathsf{mult}(\mathsf{isSmaller}(y, f(\mathsf{succ}(x))), g_1(x, y)), \\ &\qquad \mathsf{mult}(\mathsf{isSmaller}(f(\mathsf{succ}(x)), \mathsf{succ}(y)), \mathsf{succ}(x))) \end{aligned}$$

   Nun, gilt laut der Definition von g, dass $\forall x \in \mathbb{N} \colon g(x) = g_1(\mathsf{mult}(x, x), x)$. Nun folgt, dass g primitiv-rekursiv ist: Da $\mathsf{add}, \mathsf{mult}, \mathsf{isSmaller}, \mathsf{succ}$ und f primitiv-rekursiv sind, sind sowohl $g_1(0, y)$ als auch $g_1(x+1, y)$ primitiv-rekursiv, da sie demnach aus Kompositionen von primitiv-rekursiven Funktionen entstehen. Insgesamt entsteht g_1 also aus der primitiven Rekursionsvorschrift basierend auf primitiv-rekursiven Funktionen und ist damit primitiv-rekursiv. Funktion g entsteht also aus der Komposition von primitiv-rekursiven Funktionen und ist damit auch primitiv-rekursiv.

3. Wir definieren eine Hilfsfunktion $h_1 \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, die bei Eingabe (x, y) anzeigt, ob die Anzahl der Teiler von x mindestens y ist, d.h. $h_1(x, y) = 0$ falls $f(x) \geq y$ und $h_1(x, y) = 1$ sonst. Diese Hilfsfunktion kann per Kompositionsvorschrift wie folgt primitiv-rekursiv definiert werden:

$$h_1(x, y) = \mathsf{isSmaller}(f(x), y).$$

Laut Definition ist $h(m)$ die kleinste Zahl n mit $h_1(n, m) = 0$. Da $h_1(n', m) = 1$ für alle $n' < n$ gilt, folgt also, dass $h = \mu(h_1)$. Die Funktion h ist also μ-rekursiv, denn sie ergibt sich aus der Anwendung des μ-Operators auf die primitiv-rekursive Funktion h_1.

4. Da z.B. $m!$ mindestens m Teiler enthält, gilt $h(m) \leq m!$. Man kann also ein LOOP-Programm schreiben, das jede natürliche Zahl zwischen 0 und $m!$ mit Hilfe von f untersucht, bis man eine kleinste Zahl findet, die mindestens m Teiler hat. Dies ist möglich, da die Fakultätsfunktion LOOP-berechenbar ist und auch f als primitiv-rekursive Funktion ebenso LOOP-berechenbar ist. Da LOOP-Programme primitiv-rekursive Funktionen berechnen, ist h auch primitiv-rekursiv.
