Übungsblatt 3

Aufgabe 1

Zeigen Sie, dass die von folgendem WHILE-Programm berechnete Funktion LOOP-berechenbar ist. Input und Output sind x_1.

$$
\begin{array}{l}
x_2 := 0; \\
x_3 := 0; \\
\textsc{While } x_2 \neq 0 \textsc{ Do} \\
\quad x_2 := x_2 + 1; \\
\quad x_3 := x_2 + x_1; \\
\quad x_2 := x_3 \cdot x_1 \\
\textsc{End}; \\
x_2 := x_1 \cdot x_1; \\
\textsc{While } x_1 \neq 0 \textsc{ Do} \\
\quad x_1 := x_1 - 1; \\
\quad x_2 := x_2 + x_3; \\
\quad x_3 := x_3 + 1 \\
\textsc{End}; \\
x_1 := x_2
\end{array}
$$

Lösung

Wir stellen zunächst fest, dass die erste WHILE-Schleife niemals durchlaufen wird, weil der Wert von x_2 zu Beginn auf 0 gesetzt wurde. Also können wir sämtliche Zeilen der ersten Schleife streichen. Im nächsten Schritt können wir die zweite Schleife in eine LOOP-Schleife umwandeln, da genau x_1 als Counter benutzt wurde. Es wird also einfach $x_2 := x_2 + x_3$ und $x_3 := x_3 + 1$ insgesamt x_1 mal ausgeführt.

$$
\begin{array}{l}
x_2 := x_1 \cdot x_1; \\
x_3 := 0; \\
\textsc{Loop } x_1 \textsc{ Do} \\
\quad x_2 := x_2 + x_3; \\
\quad x_3 := x_3 + 1 \\
\textsc{End}; \\
x_1 := x_2
\end{array}
$$

Alternativ könnten wir den Funktionswert bestimmen und uns daraus ein LOOP-Programm herleiten. Nach dem letzten Durchlauf der WHILE-Schleife wird x_2 nämlich auf den Wert $x_1^2 + 0 + 1 + \cdots + (x_1 - 1) = x_1^2 + \frac{(x_1-1)x_1}{2}$ (der hintere Teil ist eine Gaußsumme) gesetzt.

Aufgabe 2

Geben Sie an, welche Funktionen durch die folgenden Programme berechnet werden. Es handelt sich um Funktionen von $\mathbb{N}^2$ (bzw. $\mathbb{N}^3$) nach $\mathbb{N}$ mit Eingabevariablen x_1, x_2 (bzw. x_1, x_2, x_3) und Ausgabevariable x_1.

(a) $f : \mathbb{N}^3 \to \mathbb{N}$ definiert durch

$$
\begin{array}{l}
x_2 := x_2 \cdot x_2; \\
x_4 := x_2 - x_3; \\
\textsc{If } x_4 = 0 \textsc{ Then } x_2 := x_3 \textsc{ End}; \\
x_3 := x_2 - x_1; \\
x_1 := x_1 \cdot x_2; \\
\textsc{If } x_3 = 0 \textsc{ Then} \\
\quad x_1 := 1; \\
\quad \textsc{Loop } x_2 \textsc{ Do } x_1 := 2 \cdot x_1 \textsc{ End} \\
\textsc{End}
\end{array}
$$

(b) $g : \mathbb{N}^2 \to \mathbb{N}$ definiert durch

$$
\begin{array}{l}
x_3 := x_2 - x_1; \\
\textsc{While } x_3 \neq 0 \textsc{ Do} \\
\quad x_3 := 1 \\
\textsc{End}; \\
x_1 := 2 \cdot x_1; \\
x_1 := x_1 + 3
\end{array}
$$

(c) $h : \mathbb{N}^2 \to \mathbb{N}$ definiert durch

$$
\begin{array}{rl}
M_1: & x_3 := x_2 \cdot x_2; \\
& x_4 := x_3 - 41; \\
& \textsc{If } x_4 = 0 \textsc{ Then Goto } M_1; \\
& \textsc{Goto } M_2; \\
M_2: & x_1 := x_1 + x_3; \\
& \textsc{Halt}
\end{array}
$$

Lösung

(a)

$$f(x,y,z) = \begin{cases} 2^{\max(y^2,z)}, & \text{falls } x \geq \max(y^2,z), \\ x \cdot \max(y^2,z), & \text{sonst.} \end{cases}$$

(b)

$$g(x,y) = \begin{cases} 2x+3, & \text{wenn } x \geq y \\ \text{undefiniert}, & \text{sonst.} \end{cases}$$

(c)

$$h(x,y) = \begin{cases} x+y^2, & \text{wenn } y^2 \geq 42 \\ \text{undefiniert}, & \text{sonst.} \end{cases}$$

Aufgabe 3
Zeigen Sie, dass die folgenden Funktionen primitiv rekursiv sind. Es dürfen primitiv rekursive Funktionen verwendet werden, die in der Vorlesung bereits besprochen wurden.

(a) $f(n) = n!$

(b) $g(n) = \frac{n\cdot(n+1)}{2}$

(c) $k(n,m) = m^n$

(d) $h(x_1,x_2,x_3) = \begin{cases} x_2 & \text{für } x_1 = 0 \\ x_3 & \text{sonst} \end{cases}$

Lösung
Wir bezeichnen mit add die Addition, mit mult die Multiplikation und mit $\mathrm{comp}(g, f_1, \ldots, f_k)$ die Komposition der Funktion g mit den Funktionen $f_1, \ldots, f_k$ (siehe Folie 56 im Skript).

(a) Wir definieren die Funktion $\varphi\colon \mathbb{N}^2 \to \mathbb{N}$ mit $\varphi(x,y) = x \cdot (y+1)$. Dann können wir $f(n) = n!$ mittels primitiver Rekursion definieren:

$$\begin{aligned} f(0) &= 1 \\ f(n+1) &= \varphi(f(n), n). \end{aligned}$$

Wir erhalten $f(n+1) = f(n) \cdot (n+1) = n! \cdot (n+1) = (n+1)!$ und für den Basisfall gilt $f(0) = 1 = 0!$. Die Konstante 1 aus dem Basisfall wird formal durch die konstante Funktion $k_1\colon \mathbb{N}^0 \to \mathbb{N}$ mit $k_1() = 1$ repräsentiert. Die Funktion φ ist primitiv rekursiv, da die Nachfolgerfunktion, Kompositionen und Projektionen primitiv rekursiv sind (siehe Folie 56 im Skript). Genauer:

$$\begin{aligned}\varphi(x,y) &= \mathrm{comp}(\mathrm{mult}, \pi_1^2, \mathrm{comp}(s, \pi_2^2))(x,y) \\ &= \mathrm{mult}(\pi_1^2(x,y), \mathrm{comp}(s,\pi_2^2)(x,y)) \\ &= \mathrm{mult}(x, s(\pi_2^2(x,y))) \\ &= \mathrm{mult}(x, s(y)) \\ &= x \cdot (y+1).\end{aligned}$$

(b) Es gilt $g(n) = \sum_{i=1}^{n} i$ (Gauß'sche Summenformel). Damit lässt sich g analog zur Fakultät aus (a) definieren, wobei der Wert $g(0)$ im Basisfall diesmal 0 statt 1 ist, und add statt mult in der Rekursionsdefinition verwendet wird: Sei $\psi\colon \mathbb{N}^2 \to \mathbb{N}$ definiert als $\psi(x,y) = x + (y+1)$, dann kann man g mittels primitiver Rekursion wie folgt definieren:

$$\begin{aligned} g(0) &= 0 \\ g(n+1) &= \psi(g(n), n).\end{aligned}$$

Wir erhalten $g(n+1) = g(n) + (n+1) = \sum_{i=1}^{n} i + (n+1) = \sum_{i=1}^{n+1} i$ und für den Basisfall gilt $g(0) = 0 = \sum_{i=1}^{0} i$.

(c) Definiere $\tau\colon \mathbb{N}^3 \to \mathbb{N}$ mit $\tau(x,y,z) = x \cdot z = \mathrm{mult}(x,z)$. Es gilt

$$\begin{aligned} k(0,m) &= 1 \\ k(n+1,m) &= \tau(k(n,m), n, m).\end{aligned}$$

Wir erhalten $k(n+1,m) = k(n,m) \cdot m = m^n \cdot m = m^{n+1}$.

(d) Wir definieren Funktionen $h_1\colon \mathbb{N}^2 \to \mathbb{N}$ mit $h_1(x,y) = \pi_1^2(x,y) = x$ und $h_2\colon \mathbb{N}^4 \to \mathbb{N}$ mit $h_2(a,b,c,d) = \pi_4^4(a,b,c,d) = d$. Es gilt

$$\begin{aligned} h(0, x_2, x_3) &= h_1(x_2, x_3) = x_2 \\ h(n+1, x_2, x_3) &= h_2(h(n, x_2, x_3), n, x_2, x_3) = x_3.\end{aligned}$$