Institut für Informatik
M. Mutz

Übungsblatt 10

Theoretische Informatik

Bearbeitungszeit: 17.06.2024 bis 23.06.2024, 16:00 Uhr
Besprechung: 24.06.2024, 10:30 Uhr in Hörsaal 5E

Abgabe: als PDF über das ILIAS
Gruppenabgaben möglich und erwünscht

Aufgabe 1 (WHILE-GOTO)10P

(a) Zeigen Sie durch Angabe eines `WHILE`-Programms, dass die Funktion $f : \mathbb{N}^2 \to \mathbb{N}$, $f(m,n) = \lfloor \frac{m}{n} \rfloor$ `WHILE`-berechenbar ist.

(b) Geben Sie die Belegung *aller* benutzten Variablen nach dem Programmdurchlauf an, für die Funktionswerte

(i) $f(2,0)$

(ii) $f(1,2)$

(iii) $f(4,2)$

(c) Zeigen Sie durch Angabe eines `GOTO`-Programms, dass die Funktion $f : \mathbb{N}^2 \to \mathbb{N}$ mit $f(n_1,n_2) = n_1 \cdot n_2$ `GOTO`-berechenbar ist.

Verwenden Sie in der gesamten Aufgabe **nur** die elementaren Befehle, wie sie in der Definition des entsprechenden Programms aufgeführt sind.

Lösungsvorschlag:

(a) $x_1 := x_1 + 1$;
LOOP x_2 **DO** $x_1 := x_1 - 1$ **END**;
WHILE $x_1 \neq 0$ **DO**
  LOOP x_2 **DO** $x_1 := x_1 - 1$ **END**;
  $x_0 := x_0 + 1$
END

(b) In diesem Beispielprogramm wäre das

(i) Es gibt keinen Zustand nach Programmdurchlauf. In der Endlosschleife gilt $x_1 = 3, x_2 = 0$, x_0 wird beliebig hochgezählt

(ii) $x_0 = 0, x_1 = 0, x_2 = 2$

(iii) $x_0 = 2, x_1 = 0, x_2 = 2$

(c) `GOTO`-Programm:

M_1 : $x_0 := 0$;
M_2 : `IF` $x_2 = 0$ `THEN GOTO` M_{10};
M_3 : $x_3 := x_1 + 0$;
M_4 : `IF` $x_3 = 0$ `THEN GOTO` M_8;
M_5 : $x_0 := x_0 + 1$;
M_6 : $x_3 := x_3 - 1$;
M_7 : `GOTO` M_4;
M_8 : $x_2 := x_2 - 1$;
M_9 : `GOTO` M_2;
M_{10} : `HALT`;

Aufgabe 2 (primitive Rekursion)10P

Zeigen Sie, dass folgende Funktionen primitiv rekursiv sind. Verwenden Sie dafür die Normalschemata aus Kapitel 9 der Vorlesung.

Hinweis: Sobald Sie so festgestellt haben, dass eine der Funktionen primitiv rekursiv ist, dürfen Sie diese für die restlichen Funktionen wiederverwenden. Sie dürfen die Addition, Multiplikation, Exponentialfunktion sowie die Fakultät aus Kapitel 9 der Vorlesung als primitiv rekursiv voraussetzen und verwenden.

(a) Die Vorgängerfunktion $V(x)$ (s. Kapitel 9):

$$V(x) = \begin{cases} 0 & \text{falls } x = 0 \\ x - 1 & \text{falls } x \geq 1. \end{cases}$$

(b) Die modifizierte Differenz $\mathrm{md}(x, y)$ (s. Kapitel 9):

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

(c) Die Signumsfunktion $S : \mathbb{N} \to \mathbb{N}$ mit

$$S(x) = \begin{cases} 0, & \text{falls } x = 0 \\ 1, & \text{falls } x \geq 1 \end{cases}$$

(d) Die vergleichende Funktion $\mathrm{smaller} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, mit

$$\mathrm{smaller}(x, y) = \begin{cases} 1, & \text{falls } x < y \\ 0, & \text{falls } x \geq y \end{cases}$$

Lösungsvorschlag:

(a) Die Vorgängerfunktion

$$V(x) = \begin{cases} 0 & \text{falls } x = 0 \\ x - 1 & \text{falls } x \geq 1. \end{cases}$$

ist primitiv rekursiv mittels Normalschema der primitiven Rekursion (2b) via:

$V(0) = 0$ (wir verwenden hier die konstante Funktion 0, vgl. (1a)),

$V(n + 1) = h(n, V(n)) = \mathrm{id}_1^2(n, V(n)) = n.$

(b) Die modifizierte Differenz

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

ist primitiv rekursiv mittels Normalschema der Substitution unter Verwendung einer "Hilfsfunktion" $\mathrm{dm}(x,y)$, welche die Parameter vertauscht:

$$\mathrm{md}(x,y) = \mathrm{dm}\left(\mathrm{id}_2^2(x,y), \mathrm{id}_1^2(x,y)\right)$$

(dabei gilt $f = \mathrm{dm}$, $g_1 = \mathrm{id}_2^2$ und $g_2 = \mathrm{id}_1^2$). $\mathrm{dm}(x,y)$ ist wiederum primitiv rekursiv mittels Normalschema der primitiven Rekursion:

$$\begin{aligned} &\mathrm{dm}(0,x) = \mathrm{id}_1^1(x) = x \\ &\mathrm{dm}(n+1,x) = h(n, \mathrm{dm}(n,x), x) = \mathrm{dm}'(n, \mathrm{dm}(n,x), x), \end{aligned}$$

wobei $\mathrm{dm}'(x,y,z) = V(\mathrm{id}_2^3(x,y,z))$ mittels Substitution (mit $f = V$ und $g_1 = \mathrm{id}_2^3$) primitiv rekursiv ist (V ist die Vorgängerfunktion, die nach Aufgabenteil (a) primitiv rekursiv ist).

(c) Die Signumsfunktion $S(X)$ ist primitiv rekursiv unter Verwendung des Normalschemas der primitiven Rekursion:

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

Dabei sind:

$$\begin{aligned} &g = 0 \text{ (konstante nullstellige Funktion) und} \\ &h = 1 \text{ (konstante zweistellige Funktion).} \end{aligned}$$

(d) $\mathrm{smaller}(x,y)$ ist primitiv rekursiv mittels Normalschema der Substitution:

$$\begin{aligned} \mathrm{smaller}(x,y) = f(g_1(x,y)) &= S(\mathrm{dm}(x,y)), \text{ also eine Substitution mit} \\ f &= S \text{ und} \\ g_1 &= \mathrm{dm}. \end{aligned}$$

Aufgabe 3 (De-Gödelisierung von $\mathbb{P}$r)12P

Gegeben sei folgende Gödelisierung einer primitiv-rekursiven Funktion. Welche Funktion wird hier berechnet?

$$\mathrm{PR}[s(0), \mathrm{SUB}[\mathrm{PR}[0, \mathrm{SUB}[\mathrm{PR}[\mathrm{id}| * |, \mathrm{SUB}[s; \mathrm{id}||| * ||](x|, x||, x|||)](x|, x||);$$
$$\mathrm{id}||| * ||, \mathrm{id}||| * |||](x|, x||, x|||)](x|, x||); \mathrm{id}||| * ||, \mathrm{id}||| * |||](x|, x||, x|||)](x|, x||).$$

Lösungsvorschlag:
Wir betrachten die einzelnen Bestandteile von innen nach außen:

- $\mathrm{PR}[s(0), \mathrm{SUB}[\mathrm{PR}[0, \mathrm{SUB}[\boldsymbol{\mathrm{PR}[\mathrm{id}| * |, \mathrm{SUB}[s; \mathrm{id}||| * ||](x|, x||, x|||)](x|, x||)};$ $\mathrm{id}||| * ||, \mathrm{id}||| * |||](x|, x||, x|||)](x|, x||); \mathrm{id}||| * ||, \mathrm{id}||| * |||](x|, x||, x|||)](x|, x||).$

  Der fett markierte Ausdruck beschreibt folgende primitiv rekursive Funktion:

  $$\left.\begin{aligned} f(0,x) &= \mathrm{id}_1^1(x) = x \\ f(n+1,x) &= s(\mathrm{id}_2^3(n, f(n,x), x)) = f(n,x) + 1 \end{aligned}\right\} \implies f = \mathrm{add}$$

- $\mathrm{PR}[s(0), \mathrm{SUB}[\mathrm{PR}[0, \boldsymbol{\mathrm{SUB}[G(\mathrm{add}); \mathrm{id}||| * ||, \mathrm{id}||| * |||](x|, x||, x|||)}](x|, x||); \mathrm{id}||| * ||, \mathrm{id}||| * |||](x|, x||, x|||)](x|, x||).$

  Der fett markierte Ausdruck beschreibt folgende primitiv rekursive Funktion:

  $$h(n_1, n_2, n_3) = \mathrm{add}(\mathrm{id}_2^3(n_1, n_2, n_3), \mathrm{id}_3^3(n_1, n_2, n_3)) = n_2 + n_3.$$

- $\mathrm{PR}[s(0), \mathrm{SUB}[\boldsymbol{\mathrm{PR}[0, G(h)](x|, x||)}; \mathrm{id}||| * ||, \mathrm{id}||| * |||](x|, x||, x|||)](x|, x||).$

  Der fett markierte Ausdruck beschreibt folgende primitiv rekursive Funktion:

  $$\left.\begin{aligned} f(0,x) &= 0 \\ f(n+1,x) &= h(n, f(n,x), x)) = f(n,x) + x \end{aligned}\right\} \implies f = \mathrm{mult}$$

- $\mathrm{PR}[s(0), \boldsymbol{\mathrm{SUB}[G(\mathrm{mult}); \mathrm{id}||| * ||, \mathrm{id}||| * |||](x|, x||, x|||)}](x|, x||).$

  Der fett markierte Ausdruck beschreibt folgende primitiv rekursive Funktion:

  $$h(n_1, n_2, n_3) = \mathrm{mult}(\mathrm{id}_2^3(n_1, n_2, n_3), \mathrm{id}_3^3(n_1, n_2, n_3)) = n_2 \cdot n_3.$$

- $\boldsymbol{\mathrm{PR}[s(0), G(h)](x|, x||)}.$

  Der fett markierte Ausdruck beschreibt folgende primitiv rekursive Funktion:

  $$\left.\begin{aligned} f(0,x) &= s(0) = 1 \\ f(n+1,x) &= h(n, f(n,x), x)) = f(n,x) \cdot x \end{aligned}\right\} \implies f(x_1, x_2) = x_2^{x_1}.$$

Aufgabe 4 (Ackermann-Funktion)8P

Betrachten Sie die Ackermann-Funktion $\alpha : \mathbb{N}^2 \to \mathbb{N}$, die in Kapitel 9 der Vorlesung definiert ist. Schreiben Sie in einer Programmiersprache Ihrer Wahl ein Programm, mit dem Sie $\alpha(m,n)$ berechnen. Nutzen Sie keine Library oder ähnliches, worin die Ackermann-Funktion schon fest implementiert ist.

(a) Für welche (m,n), $0 \leq m,n \leq 6$, findet Ihr Programm in angemessener Zeit eine Lösung?

(b) Geben Sie Ihr Programm zusätzlich in einer separaten Datei ab.

Lösungsvorschlag: Bei den meisten Programmen sollte soetwas wie

- (m,n) mit $m \leq 3$,
- (m,n) mit $n = 0$, $m \leq 5$
- $(4,1)$

als Eingaben herauskommen, für die das Programm noch eine Lösung liefert.