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.

Übungsblatt 10 : Raphael Kaya

Aufgabe 1:

a) $f(m,n) = \lfloor \frac{m}{n} \rfloor$

```
x1 := m ; x2 := n ; x3 := 0
IF x2 := 0 END;
while x1 ≠ 0 DO
   x1 := x1 - x2 ;
   IF x1 ≠ 0 DO x3 := x3 + 1 ;
END
```

⇒ Da die Fkt. für natürliche Zahlen ausgelegt sind, muss auch das Ergebnis eine nat. Zahl sein. Demnach muss m ein vielfaches von n sein. So kommt es zu keinen neg. Werten. Falls doch keine nat. Zahl und sowieso nicht in Def. von f.

⇒ Die Funktion ist While - Berechenbar

b) i) $f(2,0)$: $x_1 := 2$; $x_2 := 0$; $x_3 := 0$

ii) $f(1,2)$: $x_1 := -1$; $x_2 := 2$; $x_3 := 0$

iii) $f(4,2)$: $x_1 := 0$; $x_2 := 2$; $x_3 := 2$

c) $f(n_1, n_2) = n_1 \cdot n_2$ ⇒ Die Funktion f ist GOTO berechenbar

```
x1 := n1 ; x2 := n2 ; x3 := 0
M1 :  IF x2 := 0 GOTO M5 ;
M2 :  x3 := x3 + x1 ;
M3 :  x2 := x2 - 1 ;
M4 :  GOTO M1 ;
M5 :  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

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

Aufgabe 2:

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

Substitution:

$V(0) = 0 \qquad g(0) = O(n) = 0$

Primitive Rekursion:

$f(n+1) = h(n, f(n)) = id_1^2(n, f(n)) = n$

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

Primitiv rekursiv Schema:

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

$md(x,y) = dm(id_2^2(x,y), id_1^2(x,y)) = dm(y,x)$

$dm(0,x) = id_1^1(x) = x \qquad dm(n,x) = x-n$

$dm(n+1,x) = h(n, dm(n,x), x)$

$= V(id_2^3(n, dm(n,x), x))$

$V(dm(n,x)) = x-n-1$

$= x-(n+1)$

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

$s(0) = 0$

$s(n+1) = h(n, f(n)) = id_2^2(n, exp(0,x)) \Rightarrow x^0 = 0$ für $x=0$ & $x^0 = 1$ für $x \neq 0$

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

$smaller(x,y) = S(md(x,y))$

Denn: $x \geq y \Rightarrow md(x,y) = x-y \geq 1 \Rightarrow S(md(x,y)) = 1$

$x < y \Rightarrow md(x,y) = 0 \Rightarrow S(0) = 0$

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

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

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

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.

Aufgabe 3:

PR[s(0), SUB[PR[0, SUB[~~PR[id I*I, SUB[s, id III*II] (xI, xII, xIII)] (xI, xII)~~; id III*II, id III*III] (xI, xII, xIII)] (xI, xII); id III*II, id III*III] (xI, xII, xIII)] (xI, xII).

PR[id I*I, SUB[s, id III*II] (xI, xII, xIII)] (xI, xII)

= PR[id I*I, xII+1] (xI, xII)

↳ Basis: xI, Rekursion = xII+1

f = [f, g] (x...) in s und g in f

PR[id I*I, xII+1] (xI, xII)

= id I*I (xII+1) = xII+1

PR[0, SUB[$\underbrace{\text{PR[id I*I, xII+1] (xI, xII)}}_{= \text{xII}+1}$; id III*II, id III*III] (xI, xII)] (xI, xII)

= PR[0, SUB[xII+1; id III*II, id III*III] (xI, xII)] (xI, xII)

id III*III (xI, xII, xIII) = xIII

id II*II (xI, xII, xIII) = xII

xII+1 auf : xII+1

⇒ PR[0, xII+1] (xI, xII) = xII+1

SUB[xII+1; id III*II, id III*III] (xI, xII, xIII)

= xII+1

PR[$\underbrace{\text{s(0)}}_{=1}$, xII+1] (xI, xII) = xII+1

⇒ Es wird die Funktion f(xI, xII, xIII) = xII+1 berechnet

Aufgabe 4:

a) Für kleinere Werte, bis 3,3 . Ab 4,4 steigt die Berechnungszeit beträchtlich an, und kann dann auch sehr lange dauern.

b)