Universität Karlsruhe
Institut für Logik, Komplexität
und Deduktionssysteme
Prof. Dr. W. Menzel
Martin Giese

16. Mai 2001

Berechenbarkeit SS 01

Übungsblatt 1 – Lösungen

Aufgabe 1

(a) Wir benutzen die folgende Darstellung der Paarfunktion:

$$\langle x, y \rangle = x + \sum_{d=1}^{x+y} d$$

Zunächst konstruieren wir eine Ableitung für die Addition. Das Rekursionsschema der Addition lautet:

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

Eine exakte Ableitung, die nur die Grundfunktionen, Komposition und primitive Rekursion benutzt, benötigt einige Hilfsfunktionen:

$$\begin{aligned} f_1(x) &= x && \text{(Projektion)} \\ f_2(x) &= x + 1 && \text{(Nachfolgerfunktion)} \\ f_3(x_1, x_2, x_3) &= x_2 && \text{(Projektion)} \\ f_4(x_1, x_2, x_3) &= f_2(f_3(x_1, x_2, x_3)) && \text{(Komposition)} \\ add(0, y) &= f_1(y) && \\ add(x + 1, y) &= f_4(x, add(x, y), y) && \text{(primitive Rekursion)} \end{aligned}$$

Nun definieren wir die Summe $Sum(n) = \sum_{i=0}^{n} i$ der ersten n natürlichen Zahlen:

$$\begin{aligned} f_5(x, y) &= f_2(add(x, y)) && \text{(Komposition)} \\ Sum(0) &= 0 && \text{(nullstellige Null)} \\ Sum(n + 1) &= f_5(n, Sum(n)) && \text{(primitive Rekursion)} \end{aligned}$$

Schließlich erhält man die gesuchte Paarfunktion durch:

$$\begin{aligned} f_6(x_1, x_2) &= x_1 && \text{(Projektion)} \\ f_7(x, y) &= Sum(add(x, y)) && \text{(Komposition)} \\ \langle x, y \rangle &= add(f_6(x, y), f_7(x, y)) && \text{(Komposition)} \end{aligned}$$

(b) Da für $x, y > 0$ gilt:

$$\langle x, y \rangle = x + \sum_{d=1}^{x+y} d \geq x + (x + y) + \sum_{d=1}^{x+y-1} d \quad ,$$

und auch $\langle 0,0\rangle = 0 \geq 0{+}0$, gilt $\langle x,y\rangle > x{+}y$ für alle x, y, und insbesondere $z > z.1$ und $z > z.2$ für alle z. Mit Hilfe der *beschränkten Minimalisierung* und der *beschränkten Existenzquantifikation* lassen sich die Komponentenumkehrfunktionen also folgendermaßen darstellen:

$$z.1 = \mu x \leq z.(\exists y \leq z)[\langle x,y\rangle = z]$$
$$z.2 = \mu y \leq z.(\exists x \leq z)[\langle x,y\rangle = z]$$

Bleibt also zu zeigen, daß die Gleichheit ein primitiv rekursives Prädikat ist, und daß die primitiv rekursiven Prädikate gegenüber der beschränkten Minimalisierung

$$\mathit{bmin}_P(z,x) := \mu y \leq z[Pxy] = \begin{cases} \mu y.[Pxy] & \text{falls } \mu y.[Pxy]\downarrow \leq z, \\ z+1 & \text{sonst,} \end{cases}$$

und gegenüber der beschränkten Existenzquantifikation

$$\mathit{bex}_Q(z,x) = \begin{cases} 1 & \text{falls } (\exists y \leq z)[Qxy], \\ 0 & \text{sonst,} \end{cases}$$

abgeschlossen sind.

Für die Gleichheit definieren wir zuerst durch primitive Rekursion

$$\mathit{eq}_0(0) = 1$$
$$\mathit{eq}_0(x+1) = 0$$

das Prädikat, daß überprüft, ob eine Zahl null ist. Ferner erhalten wir die Vorgängerfunktion *pred* durch

$$\mathit{pred}(0) = 0$$
$$\mathit{pred}(x+1) = x$$

wobei der Vorgänger der 0 willkürlich als 0 festgelegt wird. Die Funktion $\dot{-}$, definiert als $x \dot{-} y = x - y$, falls $x \geq y$, und $x \dot{-} y = 0$ sonst, kann durch primitive Rekursion definiert werden:

$$x \dot{-} 0 = x$$
$$x \dot{-} (y+1) = \mathit{pred}(x \dot{-} y)$$

Damit definiert man dann $(x \geq y) = \mathit{eq}_0(y \dot{-} x)$. Mit

$$\mathit{und}(0,y) = 0$$
$$\mathit{und}(x+1,y) = y$$

wird schließlich $(x = y) = \mathit{und}(x \geq y, y \geq x)$.

Die Abgeschlossenheit gegenüber der beschränkten Existenzquantifikation folgt leicht aus:

$$\mathit{bex}_Q(0,x) = Q(x,0)$$
$$\mathit{bex}_Q(z+1,x) = \mathit{bex}_Q(z,x) \vee Q(x,z+1)$$

Für die beschränkte Minimalisierung benötigt man die Fallunterscheidung nach einem primitiv rekursiven Prädikat:

$$F_{Q,f,g}(x) = \begin{cases} f(x) & \text{falls } Qx, \\ g(x) & \text{sonst.} \end{cases}$$

Diese ist prim. rekursiv, denn man kann eine Hilfsfunktion h definieren durch

$$\begin{aligned} h(0,y) &= g(y) \\ h(x+1,y) &= f(y) \end{aligned}$$

und erhält $F_{Q,f,g} = h(Q(x),x)$.

Mittels einer Fallunterscheidung läßt sich nun auch die beschränkte Minimalisierung primitiv rekursiv definieren:

$$\begin{aligned} bmin_P(0,x) &= 1 \dot{-} P(x,0) \\ bmin_P(z+1,x) &= \begin{cases} bmin_P(z,x) & \text{falls } z \geq bmin_P(z,x) \vee P(x,z+1), \\ z+2 & \text{sonst.} \end{cases} \end{aligned}$$

(c) Wir definieren eine Hilfsfunktion f, die in ihrem Ergebnis jeweils die beiden nächsten Folgenelemente codiert:

$$f(x) := \langle \mathit{fib}(x), \mathit{fib}(x+1) \rangle$$

Die Funktion f ist primitiv rekursiv:

$$\begin{aligned} f(0) &= \langle 1, 1 \rangle \\ f(x+1) &= \langle f(x)._2, f(x)._1 + f(x)._2 \rangle \end{aligned}$$

Die Fibonacci-Folge ergibt sich nun aus der ersten Komponente von f:

$$\mathit{fib}(x) = f(x)._1$$

Aufgabe 2

f^{-1} ist partiell rekursiv. Wenn $f = \varphi_e$, ist

$$f^{-1} = \lambda y.(\mu z[\varphi_{e,z._1}(z._2) \downarrow = y])._2.$$

Eine entsprechende Algorithmenskizze:

> **Eingabe:** y
> Suche per dovetailing t, x, so daß $\varphi_{e,t}(x) \downarrow = y$.
> Sobald t, x gefunden wurden, gib x aus und halte an.

Hierbei bedeutet "per dovetailing", daß systematisch alle möglichen Paare (t, x) ausprobiert werden. Dies geschieht durch den Operator μz mit $t = z._1$ und $x = z._2$

Im allgemeinen gilt $f^{-1} \neq \lambda y.\mu x[f(x) = y]$. Sei etwa

$$f := \lambda x. \begin{cases} \uparrow & \text{falls } x = 0, \\ x & \text{sonst.} \end{cases}$$

Dann ist $f^{-1} = f$ aber $\lambda y.\mu x[f(x) = y] = \lambda y. \uparrow$.

Aufgabe 3

(a) *Annahme:* K^C r.a. . Dann existiert ein e mit $K^C = dom(\varphi_e)$. Es folgt:

$$e \in K \iff \varphi_e(e)\downarrow \iff e \in dom(\varphi_e) \iff e \in K^C.$$

Widerspruch!

Also ist K^C nicht r.a. und damit K nicht entscheidbar.

Aber K ist r.a.: $K = dom(\lambda i.\varphi_i(i)) = \{i\colon (\exists t)[Titi]\}$.

(b) B ist endlich, also entscheidbar.

(c) C ist entscheidbar: $\chi_C = \lambda i.Tii^2i$.

(d) D^C ist r.a.:

$$\begin{aligned} D^C &= \{\langle e,i\rangle\colon \varphi_e(e)\downarrow \wedge \neg Ti\varphi_e(e)i\} \\ &= \{\langle e,i\rangle\colon (\exists t,y)[\varphi_{e,t}(e)\downarrow = y \wedge \neg Tiyi]\} \end{aligned}$$

D ist nicht r.a.: Wähle i_0 mit $\varphi_{i_0} = \lambda x.\uparrow$. Dann gilt:

$$e \in K^C \iff \varphi_e(e)\uparrow \iff \langle e,i_0\rangle \in D.$$

Annahme: D ist r.a., d.h. es gibt ein j mit $D = W_j$ (wobei W_j definitionsgemäß für $dom(\varphi_j)$ steht). Dann ist $K^C = dom(\lambda e.\varphi_j(\langle e,i_0\rangle))$ rekursiv aufzählbar. *Widerspruch!*

Also ist D auch nicht entscheidbar.

Aufgabe 4

(a) Wir definieren

$$\psi(x,y,e) := \mu s[Txse \vee Tyse].$$

Dann folgt $\psi \in P_3$ und für alle x,y:

$$dom(\lambda e.\psi(x,y,e)) = W_x \cup W_y.$$

Also existiert ein i mit $\psi = \varphi_i^{(3)}$. Nach dem Substitutionstheorem existiert ein $s \in R_3$ ($s = sub^{(2)}$) mit

$$(\forall x,y)[\varphi_{s(i,x,y)} = \lambda e.\varphi_i^{(3)}(x,y,e)].$$

Die Funktion $h := \lambda x,y.s(i,x,y)$ leistet nun das Geforderte:

$$W_{h(x,y)} = dom(\varphi_{s(i,x,y)}) = dom(\lambda e.\varphi_i^{(3)}(x,y,e)) = dom(\lambda e.\psi(x,y,e)) = W_x \cup W_y.$$

(b) Wir setzen

$$\psi(e,x) = \varphi_e(e)$$

Nach dem Substitutionstheorem existiert ein $g \in R_1$ mit $\varphi_{g(e)} = \lambda x.\psi(e,x) = \lambda x.\varphi_e(e)$. Es folgt:

$$\varphi_e(e)\downarrow \iff \varphi_{g(e)}(0)\downarrow \iff g(e) \in K_0.$$

D.h. $\chi_K = \chi_{K_0} \circ g$. Wäre K_0 entscheidbar, dann auch K. *Widerspruch!*

Aufgabe 5

(a) Die Notation wird übersichtlicher mit der Definition

$$A_m := \lambda n.A(m,n) \quad \text{für } m \in \mathbb{N}.$$

Man hat dann:

$$\begin{aligned} A_0(n) &= n+1 \\ A_{m+1}(0) &= A_m(1) \\ A_{m+1}(n+1) &= A_m(A_{m+1}(n)) \end{aligned}$$

Wir zeigen durch vollständige Induktion über m, daß alle A_m primitiv rekursiv sind.

Für $m = 0$ ist A_0 die Nachfolgerfunktion, die definitionsgemäß primitiv rekursiv ist.

Für $m > 0$ nehmen wir an, $g := A_m$ sei prim. rek. Wir müssen zeigen, daß auch $h := A_{m+1}$ prim. rek. ist. Es gilt

$$\begin{aligned} h(0) &= A_{m+1}(0) = A_m(1) = g(1) \\ h(n+1) &= A_{m+1}(n+1) = A_m(A_{m+1}(n)) = g(h(n)) \end{aligned}$$

Da $g(1)$ eine Konstante ist, entspricht dies dem Schema der primitiven Rekursion. Demnach ist h primitiv rekursiv.

(b) Betrachte die obigen Rekursionsgleichlungen für h. Es gilt offenbar $h(n) = g^{n+1}(1)$, wobei g^{n+1} die $n+1$-fache Iteration der Funktion g ist. Also:

$$A_{m+1}(n) = (A_m)^{n+1}(1)$$

Sei k ein beliebiger Index der Nachfolgerfunktion A_0, also $\varphi_k = A_0$. Wir wollen das s_n^m-Theorem benutzen, um aus einem Index von A_m einen Index von A_{m+1} zu berechnen. Die Funktion

$$\lambda i, n.\, (\varphi_i)^{n+1}(1)$$

ist sicher berechenbar. Laut s_n^m-Theorem existiert also eine primitiv rekursive Funktion s mit

$$\varphi_{s(i)}(n) = (\varphi_i)^{n+1}(1) \quad \text{für alle } n \in \mathbb{N}.$$

Damit haben wir

$$\begin{aligned} A_0(n) &= \varphi_k(n) \\ A_1(n) &= \varphi_{s(k)}(n) \\ &\vdots \\ A_m(n) &= \varphi_{s^m(k)}(n) \end{aligned}$$

Definiere $a := \lambda m.s^m(k)$. Als Iteration einer primitiv rekursiven Funktion ist auch a primitiv rekursiv.

(c) A ist berechenbar, da $A(m,n) = \varphi_{a(m)}(n) = \Phi(a(m), n)$. A ist total, da nach (a) die Funktion $A_m = \lambda n.A(m,n)$ für jedes m primitiv rekursiv, und damit total ist.

Aufgabe 6

(a) *Prim* ist effektiv numerierbar. Der Einfachheit halber gehen wir davon aus, daß unsere Standardnumerierung durch Arithmetisierung der μ-rekursiven Ausdrücke gewonnen wurde (andernfalls müßten wir entsprechende Übersetzungen zwischen μ-rekursiven Ausdrücken und z.B. Turing Maschinen verwenden).

Sei also α : μ-rekursive Ausdrücke $\rightarrow \mathbb{N}$ die entsprechende Arithmetisierung der μ-rekursiven Ausdrücke. $h(e)$ wird nun als Index des folgenden Programms definiert:

> **Eingabe:** e
> Falls $e \in rg(\alpha)$ und in $\alpha^{-1}(e)$ kommt kein μ-Operator vor, dann gib e aus.
> Sonst gib $\alpha(\lambda x.0)$ aus.

Offensichtlich ist h total berechenbar und alle $\varphi_{h(e)}$ sind primitiv rekursiv. Da jede primitiv rekursive Funktion eine Darstellung ohne μ-Operator besitzt, wird auch jede primitiv rekursive Funktion durch $\varphi_{h(e)}$ für mindestens ein e angenommen.

(b) Sei h die Funktion aus Teil (a), also $Prim = \{\varphi_{h(e)} : e \in \mathbb{N}\}$. Wir definieren $g := \lambda x.1 + \varphi_{h(x)}$. Dann ist g in R_1.

Annahme: $g \in Prim$. Dann existiert ein e mit $g = \varphi_{h(e)}$. Es folgt:

$$g(e) = 1 + \varphi_{h(e)}(e) = 1 + g(e). \quad \textit{Widerspruch!}$$

Hier noch ein alternativer, kürzerer und eleganterer Weg:

Wäre $Prim = R_1$, dann wäre nach Teil (a) auch R_1 effektiv numerierbar. *Widerspruch!*

Noch eine Bitte: Falls Ihr auf den Übungsblättern oder in den Musterlösungen Fehler findet, oder falls Ihr andere, evetentuell schönere Lösungen findet, teilt sie mir bitte mit.

Die Unterlagen zur Vorlesung und zu den Übungen sind erhältlich via:

`http://i11www.ira.uka.de/~giese/brb01/`

Martin Giese: Informatik Neubau, Zi. 307, Tel. 608-4245, Email: `giese@ira.uka.de`