Universität Karlsruhe (TH) – Fakultät für Informatik
Institut für Logik, Komplexität und Deduktionssysteme
Lehrstuhl Prof. Dr. P. Deussen

Informatik III – Prof. Dr. P. Deussen im Wintersemester 2001/2002

C. Bierwald, Th. Käufl, F. Schuster http://i10www.ira.uka.de/lehre/info3-winter2001 info3@ira.uka.de

Übungsblatt 12 – Lösungsvorschläge

18.1.2002

Aufgabe 1 s-m-n-Theorem (Bierwald)

Zeigen Sie, daß es eine totale und berechenbare Funktion $g : \mathbb{N}^2 \to \mathbb{N}$ gibt, so daß

$$\varphi_{g(i,j)}(x) = \varphi_i(x) + \varphi_j(x).$$

Dabei sei diese Summe genau dann für $x \in \mathbb{N}$ definiert, wenn $\varphi_i(x)$ und $\varphi_j(x)$ definiert sind.

Wir betrachten die Funktion $\Psi : \mathbb{N}^3 \to \mathbb{N}$, gegeben durch $\Psi(i, j, x) = \varphi_i(x) + \varphi_j(x)$. Diese wird zum Beispiel von folgendem erweiterten While-Programm berechnet:

```
begin
        h1 := Φ (X1, X3);
        h2 := Φ (X2, X3);
        X1 := h1 + h2
end
```

Folglich ist Ψ berechenbar und es gibt einen Index – nennen wir Ihn e – mit $\Psi = \varphi_e^{(3)}$. Das s-m-n-Theorem sagt uns, daß es eine totale und berechenbare Funktion s_1^2 gibt, so daß $\varphi_e^{(3)}(i, j, x) = \varphi^{(1)}_{s_1^2(e,i,j)}(x)$ gilt. Jetzt können wir unsere gesuchte Funktion Funktion $g : \mathbb{N}^2 \to \mathbb{N}$ durch $g(i, j) = s_1^2(e, i, j)$ definieren. Da s_1^2 berechenbar ist, gibt es ein while-Programm $\mathcal{P}$, daß s_1^2 berechnet. Das folgende (erweiterte) while-Programm berechnet dann g:

```
begin X3 := X2; X2 := X1; X1 := e ; 𝒫 end
```

Da s_1^2 auch total ist, terminiert dieses Programm stets, womit neben der Berechenbarkeit auch die Totalität von g nachgewiesen wäre.

Aufgabe 2 Collatz-Funktion, Ulam's Problem (Bierwald)

Die Funktion $c : \mathbb{N} \to \mathbb{N}$ sei gegeben durch $c(x) = \begin{cases} x/2 & \text{, falls } x \text{ gerade} \\ 3x+1 & \text{, sonst} \end{cases}$

Damit definieren wir die Menge

$$M = \{x \in \mathbb{N} : \text{es gibt } i \in \mathbb{N}, \text{ so daß } c^i(x) = 1\}$$

Beispiel. $3 \in M$, wegen $c^0(3) = 3$, $c^1(3) = 10$, $c^2(3) = 5$, $c^3(3) = 16$, …, $c^7(3) = 1$

Zeigen Sie, daß M aufzählbar ist. Ist M auch entscheidbar?

Hinweis. Verzweifeln Sie nicht, wenn Sie bei der Frage nach der Entscheidbarkeit nicht weiterkommen. Sie sind in guter Gesellschaft. Bis jetzt hat noch niemand eine positive, natürliche Zahl > 0 gefunden, die nicht in M liegt. (Und es wurden viele Zahlen untersucht!) Allerdings hat auch noch niemand beweisen können, daß es eine solche Zahl nicht gibt.

Wir betrachten das folgende erweiterte While-Programm:

```
begin
        while X1 ≠ 1 do
                if X1 mod 2 = 0 then
                        X1 := X1 div 2
                else
                        X1 := (3*X1)+1
                fi
        od
        X1 := 1
end
```

Das Programm berechnet eine Funktion $\varphi : \mathbb{N} \to \mathbb{N}$. Dabei ist $\varphi(x) = 1$ genau dann, wenn $x \in M$ und undefiniert sonst. Es ist also $Dom(\varphi) = M$ und somit ist M – als Definitionsbereich einer berechenbaren Funktion – aufzählbar.

Ob M entscheidbar ist, wissen wir nicht? Es wird vermutet, daß $M = \mathbb{N} \setminus \{0\}$ ist, aber wie schon im Hinweis erwähnt, ist dies ein offenes Problem. Falls die Vermutung zutrifft, wäre M natürlich entscheidbar. Es gibt viele WWW-Seiten, die sich mit diesem Problem befassen. Einfach mal im Web nach „Collatz" suchen.

Aufgabe 3 Einige primitiv rekursive Funktionen (Käufl)

Wir verwenden das folgende Einsetzungsschema zur Definition von primitiv rekursiven Funktionen:

Ist g eine primitiv rekursive Funktion $Nat^n \to Nat$, und sind $h_1, \ldots, h_n$ primitiv rekursive Funktionen $Nat^m \to Nat$, dann auch $f(\boldsymbol{x}) = g(h_1(\boldsymbol{x}), \ldots, h_n(\boldsymbol{x}))$, wobei $\boldsymbol{x} \in Nat^m$.

a) Zeigen Sie: Sind $g(x_1, \ldots, x_j, \ldots, x_n)$ und $h(y_1, \ldots, y_m)$ primitiv rekursiv, dann auch $f(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n, y_1, \ldots, y_m) = g(x_1, \ldots, x_{j-1}, h(y_1, \ldots, y_m), x_{j+1}, \ldots, x_n)$.

Damit ist

$$f(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n, y_1, \ldots, y_m) = g(X_1, \ldots, X_{j-1}, h(Y_1, \ldots, Y_m), X_{j+1}, \ldots, X_n),$$

wobei

$$X_1 = \mathrm{U}_1^{n+m-1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n, y_1, \ldots, y_m)$$

$$X_{j-1} = \mathrm{U}_{j-1}^{n+m-1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n, y_1, \ldots, y_m)$$

$$X_{j+1} = \mathrm{U}_{j+1}^{n+m-1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n, y_1, \ldots, y_m)$$

$$X_n = \mathrm{U}_{n-1}^{n+m-1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n, y_1, \ldots, y_m)$$

$$Y_1 = \mathrm{U}_n^{n+m-1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n, y_1, \ldots, y_m)$$

$$Y_m = \mathrm{U}_{n+m-1}^{n+m-1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n, y_1, \ldots, y_m)$$

Da $g(x_1, \ldots, x_{j-1}, h(y_1, \ldots, y_m), x_{j+1}, \ldots, x_n)$ ein mit Hilfe primitiv rekursiver Funktionen gebildeter Term ist, folgt die primitive Rekursivität von f auch unter Verwendung der in der Vorlesung gegebenen Definition.

b) Sei g: $Nat^n \to Nat$ eine primitiv rekursive Funktion. Zeigen Sie, daß auch f: $Nat^n \to Nat$, wobei $f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n) = g(x_1, \ldots, x_j, \ldots, x_i, \ldots, x_n)$ primitiv rekursiv ist. (f entsteht durch Vertauschen der Argumente i und j.)

Wir verwenden im Einsetzungsschema

$$h_1 = \mathrm{U}_1^n, h_2 = \mathrm{U}_2^n, \ldots, h_i = \mathrm{U}_j^n, \ldots, h_j = \mathrm{U}_i^n, \ldots, h_n = \mathrm{U}_n^n$$

und erhalten $f(x_1, \ldots x_n) = g(h_1(x_1, \ldots x_n), \ldots, h_n(x_1, \ldots x_n))$

Aufgabe 4 Einige primitiv rekursive Funktionen (Käufl)

a) Zeigen Sie, daß die folgenden Funktionen primitiv rekursiv sind: *vorg* (Vorgänger), $\dot{-}$, die Signumfunktion *sg* (vgl. Übungsblatt 10, Aufgabe 4) und $kl(x, y)$ mit $kl(x, y) = 1$ falls $x < y$ und $kl(x, y) = 0$ sonst.

$vorg(0) = 0$, $vorg(nachf(y)) = \mathrm{U}_2^2(vorg(y), y)$
Die Funktion g des primitiven Rekursionsschemas ist 0, U_2^2 wird als Funktion h verwendet.

$\dot{-}$ ist durch $x \dot{-} 0 = x$ und $x \dot{-} nachf(y) = vorg(x \dot{-} y)$ festgelegt. Das Schema der primitiven Rekursion ist nicht unmittelbar anwendbar. Wir betrachten daher die Funktion g mit

$$g(0, x) = x \text{ und } g(nachf(y), x) = vorg(g(y, x))$$

g ist primitiv rekursiv, da die zweite Gleichung für g in der Form

$$g(nachf(y), x) = vorg(\mathrm{U}_1^3(g(y, x), y, x))$$

geschrieben werden kann. Da $x \dot{-} y = g(y, x)$ ist auch $\dot{-}$ primitiv rekursiv.

Die Funktion $c(0) = nachf(0)$, $c(nachf(x)) = \mathrm{U}_1^2(c(x), x)$ ist primitiv rekursiv. Sie bildet jede natürliche Zahl auf 1 ab.

Die Funktion $c_2(x, y) = c(\mathrm{U}_1^2(x, y)$ ist primitiv rekursiv und bildet jedes Paar natürlicher Zahlen auf 1 ab.

Damit ist auch $sg(0) = 0$, $sg(nachf(x)) = c_2(sg(x), x)$ primitiv rekursiv.

$kl(x, y) = sg(y \dot{-} x) = sg(y \dot{-} x)$. Die primitive Rekursivität von $sg(y \dot{-} x)$ folgt aus dem in Teil a bewiesenen Einsetzungsschema, (g ist sg und h die Differenz $\dot{-}$) und der oben nachgewiesenen primitiven Rekursivität des Permutierens von Argumenten.

b) Der beschränkte μ-Operator $\mu y\ (0 \le y \le u)\ [p(y, z) = 1]$, ist wie der μ-Operator definiert, wobei das kleinste y mit $p(y, z) = 1$ nur im Anfangsstück $0 \ldots u$ der natürlichen Zahlen gesucht wird:

$\mu y\ (0 \le y \le u)\ [p(y, z) = 1] = 0$, falls es kein y $(0 \le y \le u)$ mit $p(y, z) = 1$ gibt und sonst das kleinste y aus diesem Anfangsstück, für das $p(y, z) = 1$.

Zeigen Sie, daß der beschränkte μ-Operator primitiv rekursiv ist. (Hinweis: Geben Sie ein Loop-Programm für die Ermittlung von y an.)

Das folgende Loop-Programm entspricht dem beschränkten μ-Operator.

```
h := 0; y := 0; r := 0;
while Nachf(u) ≠ y do
        if h = 0 then
            h := p(y, z);
            if h = 1 then r := y fi
        fi;
        y := nachf(y)
od;
y := r
```

c) Zeigen sie, daß die Funktionen *div* und *mod* (aus der While-Sprache mit Erweiterungen) primitiv rekursiv sind. (Hinweis: Benutzen Sie den beschränkten μ-Operator. Weiters sei $x \; div \; 0 = 0$ und $x \; mod \; 0 = 0$.)

$x \; div \; y = \mu z(0 \leq z \leq x) \; [kl(x, y \times nachf(z)) = 1]$
$x \; mod \; y = x \dot{-} y \times (x \; div \; y)$

Aufgabe 5 μ-Rekursion (Bierwald)

Beschreibt die Funktion $f(z) = \mu y[z \dot{-} y = 1]$ die Vorgängerfunktion, wie wir sie aus der while-Sprache kennen? Begründen Sie Ihre Antwort.

Für $z > 0$ ist $z - 1$ das kleinste y mit $z \dot{-} y = 1$, also $f(z) = vorg(z)$. Aber $f(0)$ ist undefiniert, da es kein y mit $0 \dot{-} y = 1$ gibt. Daher ist f nicht die Vorgängerfunktion.