Berechenbarkeit und Komplexität Sommersemester 01

Aufgaben zu (primitiv) rekursiven Funktionen

Beachten Sie die Hinweise zu rekursiven Funktionen im autotool `http://www.informatik.uni-leipzig.de/~autotool/doc/fun.html`, insbesondere die Listen der vordefinierten (`Builtin`) Funktionen.

FUN1) Kurt Gödel definiert in *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme* (1931)

x ist teilbar durch y	$x/y \equiv \exists z \leq x : x = y \cdot z$
x ist Primzahl	$\mathrm{Prim}(x) \equiv x > 1 \wedge \neg\exists z \leq x : (z \neq 1 \wedge z \neq x \wedge x/z)$
Fakultät	$0! \equiv 1; (n+1)! \equiv (n+1) \cdot n!$
die n–te Primzahl	$\mathrm{Pr}(0) \equiv 0; \mathrm{Pr}(n+1) \equiv \min\{y : y \leq \mathrm{Pr}(n)! + 1 \wedge \mathrm{Prim}(y) \wedge y > \mathrm{Pr}(n)\}$

Das sind die originalen Definitionen in Gödels Bezeichnungen.
Die letzte Zeile besagt $\mathrm{Pr}(0) = 0, \mathrm{Pr}(1) = 2, \mathrm{Pr}(2) = 3, \mathrm{Pr}(3) = 5, \mathrm{Pr}(4) = 7, \mathrm{Pr}(5) = 11, \ldots$

1. Wieso sind diese Funktionen und Prädikate primitiv rekursiv? Definieren Sie (/), Prim, (!), Pr (autotool–Subjects: FUN111, FUN112, FUN113, FUN114). Sie dürfen die Definitionen aus den jeweils vorherigen Teilaufgaben als `Builtins` benutzen (`Builtin 2 "teilbar"`, `Builtin 1 "prim"` , `Builtin 1 "fak"` , `Builtin 1 "pr"`)
2. Diskutieren Sie die Benutzung der Fakultät in der Definition von Pr.
3. (autotool: FUN13) Finden Sie eine primitiv rekursive Definition für die Funktion $g(n, x) =$ „der n–kleinste Primfaktor von x “. Beispiel: $g(2, 360) = g(2, 2^3 \cdot \boxed{3}^2 \cdot 5) = 3$.
4. (autotool: FUN14) Ebenso für die Funktion $e(n, x) =$ „der Exponent des n–kleinsten Primfaktors von x “. Beispiel: $e(2, 360) = e(2, 2^3 \cdot 3^{\boxed{2}} \cdot 5) = 2$.

   (Sie dürfen die Funktion g aus der vorigen Teilaufgabe als `Builtin 2 "g"` benutzen.)

Hinweis: Gödels Aufsatz ist nachgedruckt in: Karel Berka und Lothar Kreiser: *Logik–Texte*, Akademie–Verlag, Berlin, 1986 (siehe Mathematik–Bibliothek). Dort finden Sie die Lösungen zu FUN13 und FUN14.

FUN2) Unter welchen Bedingungen an g, h ist die Funktion

```
f = Ind 2 [ g , Sub 3 [ h, Proj 3 1, Proj 3 3 ] ]
```

assoziativ? kommutativ? (Ist es ausreichend, daß h selbst assoziativ bzw. kommutativ ist?)

FUN3) (Bastelaufgabe) Die Takeuchi–Funktion ist definiert durch

$$T(x,y,z) = \textbf{if } x \le y \textbf{ then } z \textbf{ else } T(T(x-1,y,z), T(y-1,z,x), T(z-1,x,y))$$

1. Berechnen Sie $T(18,12,6)$.
2. Beweisen Sie, daß $T(x,y,z)$ für alle *natürlichen* Zahlen x, y, z wohldefiniert ist.
3. Finden Sie eine *explizite* Darstellung von T, d. h. eine äquivalente Aussage der Form $T(x,y,z) = \ldots$, auf deren rechter Seite T *nicht* benutzt wird (sondern nur arithmetische Funktionen und Fallunterscheidungen).
4. Ist $T(x,y,z)$ sogar für alle *ganzen* Zahlen wohldefiniert?

FUN4) Beweisen Sie für die Ackermann–Funktion

$$A(x,0) = x+1;\ A(0,y+1) = A(1,y);\ A(x+1,y+1) = A(A(x,y+1),y)$$

die Aussage

$$\forall x,y \ge 0:\ x < A(x,y) \le A(x+1,y) \le A(x,y+1)$$

Wann gilt Gleichheit?

FUN5) Zeigen Sie, daß es genau eine Funktion $f : \mathbb{Z} \to \mathbb{Z}$ gibt, die diese Bedingungen erfüllt:

$$f(0) = 1;\ \forall z \in \mathbb{Z} : f(f(z)) = z;\ \forall z \in \mathbb{Z} : f(f(z+2)+2) = z$$

(53. William Lowell Putnam Mathematical Competition, 1993)

FUN6) (Goodstein–Folge) Die Eingabe des folgenden Programms ist eine ganze Zahl $x \ge 0$:

$b := 2$;
while $x > 0$ **do**
    $x := x - 1$;
    $c_n c_{n-1} \ldots c_0 :=$ Darstellung von x zur Basis b;
    $b := b + 1$;
    $x :=$ Wert von $c_n c_{n-1} \ldots c_0$ zur Basis b;

Beispiel: Die Eingabe ist $x = 8$. Es gilt $7 = (111)_2$, und $(111)_3 = 9 + 3 + 1 = 13$. Beim zweiten Schleifendurchlauf rechnen wir $12 = (110)_3$ und $(110)_4 = 16 + 4 + 9 = 20$. Dann $19 = (103)_4$ und $(103)_5 = 25 + 3 = 28$, usw.

1. Untersuchen Sie einige Beispiele.
2. Zeigen Sie, daß das Programm für jede Eingabe hält.
3. Zeigen Sie, daß die Anzahl $g(x)$ der Schleifendurchläufe bis zur Termination keine primitiv rekursive Funktion von x ist.