DISKRETE MATHEMATIK, ANWENDUNGSVORLESUNG WS 2012 / 2013

Prof. Dr. Sebastian Iwanowski

Übungsblatt 06 (4 Aufgaben) S.1/3

Aufgabe 1)[1]

Das folgende Programm berechnet eine natürliche Potenz von b deutlich schneller als durch die einfachere Methode, n mal b mit sich selbst zu multiplizieren. Hier soll nicht bewiesen werden, dass das schneller geht, sondern nur dass das Programm überhaupt das gewünschte Ergebnis liefert.

Die Geschwindigkeitsgewinn kann für große n leicht durch ein eigenes Programm nachgeprüft werden. Hinweis: Arbeiten Sie in herkömmlichen Sprachen wie Pascal nicht mit Integerwerten, weil Sie sonst MAXINT überschreiten.

```
{ b ∈ Z, e ∈ N }                                                    φ

   result := 1;
   base := b;
   exp := e;

   while exp > 0
     begin
       if (exp MOD 2 = 1) then
         begin
           result := result * base;
           exp := exp – 1;
         end
       else
         begin
           base := base * base;
           exp := exp DIV 2;
         end;
     end; {while}
{ result = b^e }                                                    ψ
```

a) Beweisen Sie mit vollständiger Induktion über k: $b^e = base_k^{\exp_k} \cdot result_k$ (Hierbei sind $base_k$, exp_k und $result_k$ die Werte der Variablen nach dem k-ten Schleifendurchlauf)
Tipp: Unterscheiden Sie in Ihrem Induktionsbeweis die beiden Fälle, ob exp_k gerade ist oder nicht!

b) Folgern Sie daraus, dass das Programm mit der gewünschten Nachbedingung stoppt.

[1] Diese Aufgabe ist identisch mit Aufgabe 6 des 5. Übungsblatts

Aufgabe 2)

Was berechnet die folgende Prozedur? Was ist die schwächste Vorbedingung dafür? (Beweis!)

Hinweis: Beweis durch vollständige Induktion über einen der Parameter.

```
procedure f(m,n: integer): integer

begin
   if (m <= 0)
      return 1
   else
      return m * f(m-1, n);
end;
```

Ersetzen Sie die Bedingung für den nichtrekursiven Aufruf durch (m < 0).
Was wird dann berechnet?

Aufgabe 3)

Gegeben sei folgende Funktion f:

```
procedure f(x, y, z: ℕ): ℕ;
begin
   if (x ≤ y) then
       return z
   else
      return f(x-1, y, z+1);
end;
```

a) Berechnen Sie f (10,7,3)

b) Was berechnet f(x,y,z) im allgemeinen? Beweisen Sie das durch vollständige Induktion über einen der Parameter!

Aufgabe 4)

a) Geben Sie für die folgenden Funktionen den Rekursionstypen an.
b) Begründen Sie jeweils Ihre Antwort (bei primitiv, end- und linear rekursiv durch Angabe der entspr. Funktionsteile oder (falls Ihnen das nicht gelingt) in Worten, bei allgemein rekursiv durch Argumentation, warum die Funktion noch nicht einmal linear rekursiv ist.
c) Geben Sie für die endrekursiven Funktionen äquivalente nichtrekursive Prozeduren an!

i)
```
procedure ggT(x, y: ℕ): ℕ;
begin
   if (x MOD y = 0)
   then return y
   else return ggT(y, x MOD y)
end;
```

ii) Es sei f: $\mathbb{N} \to \mathbb{N}$ mit

$f(x) = 0$	für $x = 0$
$f(x) = f(x-1) + 1$	für $x = 1$
$f(x) = x^{f(x \bmod 2)}$	sonst.

Zusatzfrage: Was ist f (32)?

iii)
```
procedure f(x: ℕ): ℕ
begin
   if ((x * x) MOD 2 = 1) then
      return x
   else
      return x/2 + f(x DIV 2)
end
```

Zusatzfrage: Was ist f(30)? Haben Sie eine Vermutung, was für allgemeine x herauskommt?

iv) Es sei f: $\mathbb{N} \to \mathbb{N}$ mit

$f(x) = 1$	für $x = 1$
$f(x) = f(x/2)$	für gerade x
$f(x) = f(3x+1)$	für ungerade x.

Berechnen Sie f(10), f(15) und f(20)! Was würden Sie daraus für f(x) im Allgemeinen vermuten?

Statt eines Beweises sollten Sie lieber nach Ulam-Collatz googeln!