LV „ Programmierung “
Prof. Dr. H. Vogler / Dipl.-Math. R. Vater

SS 10

Aufgabenblatt zur 13. Übung

Zeitraum: 19.07. bis 23.07.2010

1. Aufgabe: (Klausuraufgabe 02.2010)

(a) Eine Folge e_i ($i \geq 1$) von Zahlen sei wie folgt definiert:

- Das erste Glied der Folge soll anwenderspezifisch gewählt werden können.
- Das zweite Glied der Folge sei 3.
- Ab dem dritten Glied soll gelten: Jedes Folgeglied ist gleich dem dreifachen Produkt der beiden Vorgängerglieder.

Schreiben Sie ein H_0-Programm, welches die Werte n und e_1 vom Benutzer abfragt und dann den Wert e_n dieser Folge berechnet und ausgibt.

(b) Geben Sie ein C_0-Programm an, das durch Anwendung der aus der Vorlesung bekannten Transformationsfunktion in das folgende H_0-Programm (auf die Typdefinition wurde hier verzichtet) überführt werden kann:

```
module Main where

f1      x1 x2 = f2 x1 (x1*2)
f2      x1 x2 = if x1 > x2 then f21 x1 x2
                           else f22 x1 x2
f21     x1 x2 = f211 x1 x2
f211    x1 x2 = f3 (x1*x2) x2
f22     x1 x2 = f3 x1 (x2-x1)
f3      x1 x2 = x1

main = do x1 <- readLn
          print (f1 x1 0)
```

2. Aufgabe: (AGS 15.20*)

Folgendes H_0-Programm sei gegeben:

```
module Main where

f :: Int -> Int -> Int -> Int
f x1 x2 x3 = if x1 == 0 then x2
                        else g (x1 - 1) (x2 + x1 * x3) (x3 + x1 * x2)

g :: Int -> Int -> Int -> Int
g x1 x2 x3 = if x1 == 0 then x3
                        else f (x1 - 1) (x2 + x1 * x3) (x3 + x1 * x2)

main = do x1 <- readLn
          print (f x1 0 1)
```

(a) Wandeln Sie das Programm in ein äquivalentes C_0-Programm um. Verwenden Sie Variablen für die aktuellen Argumente, die zukünftigen Argumente, das Resultat, die aktuell arbeitende Funktion und das Flag für die Schleifenausführung.

(b) Welche Aussage lässt sich über die Argumente `x2` und `x3` treffen, wenn das Argument `x1` den Wert `0` erreicht?

3. Aufgabe: (AGS 15.16)

(a) Geben Sie ein H_0-Programm P an, welches die Eingabe einer Zahl $n \in I\!N$ fordert und dann als Ergebnis $s(n) = \sum_{j=1}^{n} j(j+n)$ ermittelt und ausgibt.

(b) Folgendes H_0-Programm sei gegeben:

```
module Main where

g :: Int -> Int -> Int
g x1 x2 = if x1 < x2 then g (x2-x1) (x2 'div' 2)
                     else h x1 x1

h :: Int -> Int -> Int
h x1 x2 = x1+x2

main = do x1 <- readLn
          print (g x1 (2*x1))
```

Vervollständigen Sie die Angaben `/*A*/` bis `/*E*/` in der folgenden Übersetzung des H_0-Programmes in ein äquivalentes C_0-Programm:

```
#include <stdio.h>

int main()
{
  int x1, x2, function, flag, result;
  /*A*/
  while (flag == 1)
  {
    if (function == 1)
      if (/*B*/) {
        /*C*/
      } else {
        /*D*/
        function = 2;
      }
    else if (function == 2) {
      /*E*/
    }
  }
  printf("%d", result);
}
```

Zusatzaufgabe: (AGS 15.15)

(a) Schreiben Sie ein C_0-Programm auf, das durch Anwendung der Ihnen bekannten Transformationsfunktion in das folgende H_0-Programm (auf die Typdefinition wurde hier verzichtet) überführt werden kann:

```
module Main where

f_1     x1 x2 = if x1 > 0 then f_1_1 x1 x2
                       else f_2 x1 x2
f_1_1   x1 x2 = f_1_1_1 x1 x2
f_1_1_1 x1 x2 = f_1_1_2 x1-1 x2
f_1_1_2 x1 x2 = f_1 x1 x2*2
f_2     x1 x2 = x2

main = do x1 <- readLn
          x2 <- readLn
          print (f_1 x1 x2)
```

(b) Folgendes H_0-Programm sei gegeben:

```
module Main where

g :: Int -> Int -> Int
g x1 x2 = if x1 > 0 then g (x1-1) (x2*2)
                    else h x2 x2

h :: Int -> Int -> Int
h x1 x2 = x2

main = do x1 <- readLn
          x2 <- readLn
          print (g x1 x2)
```

Vervollständigen Sie die Angaben `<A>` bis `<E>` in der folgenden Übersetzung des H_0-Programmes in ein äquivalentes C_0-Programm:

```
#include <stdio.h>
int main()
{
  int x1, x2, function, flag, result;
  <A>
  while (flag == 1)
  {
    if (function == 1)
      if (<B>) {
        <C>
      } else {
        <D>

        function = 2;
      }
    else if (function == 2) {
      <E>
    }
  }
  printf("%d", result);
}
```