

Aufgabenblatt zur 13. Übung

Zeitraum: 13.07. bis 17.07.2009

1. Aufgabe: (AGS 15.16)

(a) Geben Sie ein H_0 -Programm P an, welches die Eingabe einer Zahl $n \in \mathbb{N}$ 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);
}
```

2. Aufgabe: (Klausuraufgabe 02.2009)

(a) Transformieren Sie die folgende Funktion h eines H_0 -Programmes in ein AM_0 -Programm mit baumstrukturierten Adressen, berechnen Sie also $\underline{functrans}(h :: \text{Int} \rightarrow \text{Int} \rightarrow \text{Int} \quad h \ x1 \ x2 = \dots)$. Sie brauchen dabei keine Zwischenschritte anzugeben.

```
h :: Int -> Int -> Int
h x1 x2 = if x1 <= x2 then x1 else h x2 x1
```

(b) Das folgende H_0 -Programm (ohne Funktionstypen) ist durch die Anwendung von trans auf ein C_0 -Programm P entstanden:

```
module Main where

f_1      x1 x2 x3 = if x2 > 0   then f_1_1 x1 x2 x3
                    else f_2 x1 x2 x3
f_1_1    x1 x2 x3 = if x1 >= x2 then f_1_1_1 x1 x2 x3
                    else f_1_1_2 x1 x2 x3
f_1_1_1  x1 x2 x3 = f_1 (x1-x2) x2 x3
f_1_1_2  x1 x2 x3 = f_1_1_2_1 x1 x2 x3
f_1_1_2_1 x1 x2 x3 = f_1_1_2_2 x1 x2 x1
f_1_1_2_2 x1 x2 x3 = f_1_1_2_3 x2 x2 x3
f_1_1_2_3 x1 x2 x3 = f_1 x1 x3 x3
f_2      x1 x2 x3 = x1

main = do x1 <- readLn
          x2 <- readLn
          print (f_1 x1 x2 0)
```

Geben Sie das Programm P an.

Zusatzaufgabe: (AGS 15.18*)

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 x_2 und x_3 treffen, wenn das Argument x_1 den Wert 0 erreicht?