Aufgabenblatt zur 12. Übung

Zeitraum: 12.07. bis 16.07.2010

1. Aufgabe: (AGS 15.4 (a))

(a) Gegeben ist folgendes H_0-Programm:

```
module Main where

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

main = do x1 <- readLn
          x2 <- readLn
          print (test x2 3 x1)
```

Transformieren Sie dieses Programm nach den Transformationsvorschriften, wie sie in der Vorlesung angegeben wurden, in ein AM_0-Programm mit baumstrukturierten Adressen. Sie brauchen dabei keine Zwischenschritte anzugeben.

2. Aufgabe: (AGS 15.14(a)*)

(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 :: Int -> Int    h x1 = ...`). Sie brauchen dabei keine Zwischenschritte anzugeben.

```
h :: Int -> Int
h x1 = if x1 == 0 then h 1 else x1
```

3. Aufgabe: (AGS 15.5 (b,c)*)

(b) Gegeben sei der folgende Ausschnitt aus einem C_0-Programm:

```
...
if (x1 > x2) x1 = x1 + x2;
else while (x1 < x2) x1 = x1 + 1;
printf("%d", x1);
...
```

Durch Nutzung der Transformationsfunktionen aus der Vorlesung/Übung entsteht daraus das folgende Fragment eines H_0-Programms:

```
...
f_3     x1 x2 = if x1 > x2 then A
                           else B
f_3_1   x1 x2 = C
f_3_2   x1 x2 = if x1 < x2 then D
                           else E
f_3_2_1 x1 x2 = F
f_4     x1 x2 = G
...
```

Ergänzen Sie das Fragment durch die Angabe der Ausdrücke `A`, `B`, `C`, `D`, `E`, `F` und `G`.

(c) Übersetzen Sie den C_0-Programmausschnitt aus Teilaufgabe (b) entsprechend der Ihnen bekannten Übersetzungsfunktionen in ein baumstrukturiertes AM_0-Programm. Das `if`-Statement habe die baumstrukturierte Adresse `a`. Zwischenschritte brauchen Sie nicht anzugeben.

4. Aufgabe: (AGS 15.10*)

Gegeben ist das folgende C_0-Programm:

```
#include <stdio.h>
int main() {
  int x1, x2;
  scanf("%i", &x1);
  while (x1 > 0) {
    x2 = x1;
    while (x2 > 0) x2 = x2 - 1;
    x1 = x1 - 1;
  }
  printf("%d", x1);
  return 0;
}
```

Durch Nutzung der Transformationsfunktionen aus der Vorlesung entsteht daraus ein H_0-Programm (ohne Angabe der Funktionstypen), von dem nur das folgende Fragment gegeben ist:

```
module Main where

f_1       x1 x2 = if x1 > 0 then A
                            else B
f_1_1     x1 x2 = C
f_1_1_1   x1 x2 = D
f_1_1_2   x1 x2 = if x2 > 0 then E
                            else F
f_1_1_2_1 x1 x2 = G
f_1_1_3   x1 x2 = H
f_2       x1 x2 = I

main = do x1 <- readLn
          print (J)
```

Ergänzen Sie das Fragment durch die Angabe der Ausdrücke für `A`, `B`, `C`, `D`, `E`, `F`, `G`, `H`, `I` und `J`!

Zusatzaufgabe 1: (AGS 15.3 (c)*)

(c) Übersetzen Sie das Programm `P` aus Teilaufgabe `(b)` in ein entsprechendes baumstrukturiertes AM_0–Programm. Zwischenschritte brauchen Sie nicht anzugeben.

Zusatzaufgabe 2: (AGS 15.8*)

```
/* Fac */
#include <stdio.h>

int main() {
  int x1, x2;
  scanf("%i", &x1);
```

```
  x2 = 1;
  while (x1 > 0) {
    x2 = x2 * x1;
    x1 = x1 - 1;
  }
  printf("%d", x2);
  return 0;
}
```

(a) Geben Sie eine intuitive Übersetzung in ein H_0–Programm mit nur einer Funktion `f` an.

(b) Übersetzen Sie schrittweise dieses C_0–Programm in ein H_0–Programm.