

Programmierung

Hinweis In der 14. Übung werden nur Aufgaben gestellt, die bisherige Themen wiederholen. Anstelle der Besprechung wird am 17. Juli um 9:20 Uhr ein Repetitorium angeboten, in dem Fragen zu vergangenen Themen ausführlich beantwortet bzw. Themen wiederholt besprochen werden können.

Aufgabe 1 (AGS 17.27 a, b)

- (a) Sei $f: \mathbb{N} \rightarrow \mathbb{N}$ wie folgt definiert: $f(n) = \sum_{i=1}^n \prod_{j=1}^i j$. Schreiben Sie ein H_0 -Programm, welches eine natürliche Zahl n einliest und $f(n)$ ausgibt.
- (b) Transformieren Sie die folgende Funktion f eines H_0 -Programms in ein AM_0 -Programm. Sie können dabei die Adressen frei vergeben, sie müssen nicht baumstrukturiert sein. Berechnen Sie also $\text{funtrans}(f :: \text{Int} \rightarrow \dots)$. Geben Sie dabei keine Zwischenschritte an.

```
f :: Int -> Int
f x1 = if x1 < 42
      then x1
      else if x1 > 42
           then f (x1 `div` 2)
           else 42
```

Aufgabe 2 (AGS 17.25 a)

Transformieren Sie die folgende Funktion h eines H_0 -Programms in ein AM_0 -Programm mit baumstrukturierten Adressen, berechnen Sie also das Ergebnis $\text{funtrans}(h :: \text{Int} \rightarrow \dots)$. Geben Sie dabei keine Zwischenschritte an.

```
h :: Int -> Int -> Int -> Int
h x1 x2 x3 = if x3 > x1
             then (x2 - 1)
             else h x2 (x1 - x3) x2
```

Hinweis Die Transformation von C_0 nach H_0 und H_0 nach C_0 wurde in der Vorlesung nicht besprochen, damit sind die nachfolgenden Übungsaufgaben fakultativ. Die nötigen Übersetzungsvorschriften finden Sie im Kapitel 19.3 des Skripts.

Zusatzaufgabe 1 (AGS 17.26)

- (a) Schreiben Sie ein H_0 -Programm, welches die Eingabe einer Zahl $n \in \mathbb{N}$ fordert und $f(n)$ ausgibt, wobei $f: \mathbb{N} \rightarrow \mathbb{N}$ für alle $n \in \mathbb{N}$ wie folgt definiert ist:

$$f(n) = \left(\sum_{i=1}^n i^2 \right) + \left(\prod_{i=1}^n i \right).$$

- (b) Transformieren Sie die folgende Funktion g (eines H_0 -Programmes) in ein AM_0 -Programm mit baumstrukturierten Adressen, berechnen Sie also $funtrans(g :: Int \rightarrow \dots)$. Geben Sie dabei keine Zwischenschritte an.

```

g :: Int -> Int -> Int
g x1 x2 = if x1 < x2
          then g (x1 + 1) (x2 `div` 2)
          else x1

```

- (c) Sei das folgendes C_0 -Programm P gegeben.

```

1  #include <stdio.h>
2
3  int main(){
4      int x1;
5      scanf("%i", &x1);
6      while (x1 % 2 == 0)
7          x1 = x1 / 2;
8      printf("%d", x1);
9      return 0;
10 }

```

Übersetzen Sie dieses Programm in ein H_0 -Programm, berechnen Sie also $trans(P)$. Geben Sie nur das Endergebnis an. Sie brauchen keine Typen anzugeben.

Zusatzaufgabe 2 (AGS 17.27 c, AGS 17.25 b)

- (a) Das folgende H_0 -Programmstück ist durch Nutzung der Transformationsfunktion aus der Vorlesung/Übung aus Statements eines entsprechenden C_0 -Programms entstanden. Geben Sie diese Statements des C_0 -Programms an.

```

f1  x1 = if ((x1 `mod` 2) == 0) then f11 x1
          else f12 x1
f11 x1 = f2 (x1 `div` 2)
f12 x1 = f2 (x1 - 1)
f2  x1 = f3 (2 * x1)

```

- (b) Folgendes H_0 -Programm sei gegeben:

```

1  module Main where
2
3  h :: Int -> Int -> Int
4  h x1 x2 = if x2 == x1 then 30
5             else x2
6
7  g :: Int -> Int -> Int
8  g x1 x2 = if 10 <= x2 then g (x1-x2) (x2-1)
9             else h (x1+x2) 10
10
11 main = do x1 <- readLn
12           print (g (3+x1) 5)

```

Vervollständigen Sie die Angaben `/*A*/` bis `/*F*/` in der folgenden Übersetzung des H₀-Programms in ein äquivalentes C₀-Programm:

```
#include <stdio.h>

int main() {
    int x1, x2, function = 2, flag, result;
    /*A*/
    while (flag == 1) {
        if (function == 1)
            if (/*B*/) {
                /*C*/
            } else {
                /*D*/
            }
        else if (/*E*/) {
            /*F*/
        }
    }
    printf("%d", result);
    return 0;
}
```