

# Programmierung

## 07. Übungsblatt

Zeitraum: 22. Mai – 26. Mai 2017

*Beachten Sie die Übungsverlegung am 25. Mai!*

### Übung 1 (AGS 12.4.5)

(a) Gegeben sei der  $\lambda$ -Term

$$(\lambda z x . z x (\lambda y . y x)) (\lambda y . z x) (\lambda z . z).$$

Reduzieren Sie schrittweise diesen Ausdruck, bis seine Normalform erreicht ist. Geben Sie jeweils bevor Sie einen Ableitungsschritt ausführen für die relevanten (Teil-)Ausdrücke die Mengen der freien bzw. der gebundenen Variablen an.

(b) Eine Funktion  $f : \mathbb{N}^+ \times \mathbb{N}^+ \rightarrow \mathbb{N}^+$  sei wie folgt definiert:

$$\begin{aligned} f(x, y) &= 2 \cdot y && \text{für } x = 1 \\ f(x, y) &= y + (x + 1) \cdot f(x - 1, x + y) && \text{für alle } x \geq 2 \end{aligned}$$

Geben Sie zur Funktion  $f$  die entsprechenden Terme  $\langle f \rangle$  und  $\langle F \rangle$  an.

(c) Gegeben sei:

$$\langle G \rangle = \lambda g x y . \langle ite \rangle (\langle iszero \rangle y) (\langle succ \rangle x) (g x (\langle pred \rangle y))$$

Berechnen Sie  $\langle Y \rangle \langle G \rangle \langle 3 \rangle \langle 0 \rangle$ . Dokumentieren Sie die Berechnung wie üblich.

### Übung 2 (AGS 13.13 (b))

Gegeben sei folgender  $AM_0$ -Code:

1: READ 1;	4: GT;	7: LIT 2;	10: WRITE 1;
2: LOAD 1;	5: JMC 12;	8: DIV;	11: JMP 2;
3: LIT 1;	6: LOAD 1;	9: STORE 1;	

Führen Sie ein Ablaufprotokoll der  $AM_0$  weiter, indem Sie sie schrittweise ablaufen lassen, bis der Befehlszähler das Programm verlässt. Die Startkonfiguration ist  $(1, \varepsilon, [], 2, \varepsilon)$ .

Sie müssen nur Zellen ausfüllen, deren Wert sich im Vergleich zur jeweils vorherigen Zeile geändert hat.

### Übung 3 (AGS 13.1 \*)

Gegeben sei folgendes  $C_0$ -Programm *Max*:

```

1  /* Max */
2  #include <stdio.h>
3
4  int main() {
5      int a, b, max;
6      scanf("%i", &a);
7      scanf("%i", &b);
8      if (a > b) max = a;
9      else max = b;
10     printf("%d", max);
11     return 0;
12 }
```

- (a) Berechnen Sie schrittweise das baumstrukturierte Programm  $bMax_0 = \text{trans}(Max)$  mit Hilfe der in der Vorlesung angegebenen Übersetzungsfunktionen. Nutzen Sie dabei zweckmäßige Abkürzungen.
- (b) Wandeln Sie  $bMax_0$  in ein Programm  $Max_0$  mit linearisierten Adressen um und berechnen Sie  $\mathcal{P}[[Max_0]](5 : 7)$ . Dokumentieren Sie den Zustand der  $AM_0$  nach Ausführung jedes Befehls.

### Zusatzaufgabe 1 (AGS 13.14)

- (a) Gegeben sei folgendes  $C_0$ -Programm.

```

1 #include <stdio.h>
2 int main()
3 {
4     int x1, x2;
5     scanf("%i", &x1);
6     scanf("%i", &x2);
7     while (x1 > 0){
8         x1 = x2 - x1;
9         if (x2 > x1)
10            x2 = x2 / 2;
11     }
12     printf("%d", x1);
13     return 0;
14 }
```

Übersetzen Sie das Programm mittels *trans* in  $AM_0$ -Code mit linearen Adressen. Geben Sie nur das Endergebnis der Übersetzung, keine Zwischenschritte an!

- (b) Gegeben sei der folgende Ausschnitt aus einem  $AM_0$ -Programm.

```

3: LOAD 2;      6: JMC 14;      9: LIT 2;      12: STORE 2;
4: LIT 5;       7: LOAD 1;     10: MUL        13: JMP 3;
5: LT;         8: LOAD 2;     11: ADD;       14: WRITE 1;
```

Erstellen Sie ein Ablaufprotokoll für dieses Programmfragment, bis die  $AM_0$  terminiert. Die Startkonfiguration ist  $(7, \varepsilon, [1/3, 2/1], \varepsilon, \varepsilon)$ .

### Zusatzaufgabe 2 (AGS 12.4.27)

- (a) Berechnen Sie schrittweise die Normalform des  $\lambda$ -Terms

$$(\lambda x y z. y z x)(\lambda x. x y)(\lambda x. x).$$

- (b) Gegeben sei die folgende Haskell-Funktion:

```

g :: Int -> Int -> Int -> Int
g 0 _ _ = 0
g n x y = (g (n - 1) x y) + (if n `mod` 2 == 0 then x else y)
```

Geben Sie einen  $\lambda$ -Term  $\langle G \rangle$  an, so dass  $g = \langle Y \rangle \langle G \rangle$  gilt.

- (c) Gegeben sei der  $\lambda$ -Term

$$\langle F \rangle = \lambda f n x y. \langle ite \rangle (\langle iszero \rangle (\langle pred \rangle n)) \\ (\langle add \rangle x y) \\ (f (\langle pred \rangle n) (\langle mult \rangle x n) (\langle add \rangle y n)).$$

Berechnen Sie die Normalform des Terms  $\langle Y \rangle \langle F \rangle \langle 2 \rangle \langle 3 \rangle \langle 5 \rangle$ .