

Aufgabenblatt zur 7. Übung

Zeitraum: 07.06. bis 11.06.2010

1. Aufgabe: (AGS 11.50*)

(a) Gegeben sei folgender λ -Term:

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

Reduzieren Sie diesen Term bis seine Normalform erreicht ist. Schreiben Sie – bevor Sie einen Ableitungsschritt ausführen – für die relevanten (Teil-)Ausdrücke die Mengen der freien bzw. der gebundenen Vorkommen von Variablen auf.

(b) Gegeben sei:

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

Berechnen Sie schrittweise $\langle Y \rangle \langle G \rangle \langle 4 \rangle \langle 1 \rangle \langle 0 \rangle$ mit $Y = (\lambda h.((\lambda y.h(yy))(\lambda y.h(yy))))$. Führen Sie im Rechenprozess zweckmäßige Abkürzungen der λ -Terme ein.

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

$$\begin{aligned} f(x, y) &= y && \text{für } x = 0 \\ f(x, y) &= y + 1 && \text{für } x = 1 \\ f(x, y) &= f(x - 1, y + 1) * f(x - 2, y + 2) && \text{für } x \geq 2 \end{aligned}$$

Geben Sie zur Funktion f den zugehörigen λ -Term $\langle F \rangle$ an, so dass $\langle f \rangle = \langle Y \rangle \langle F \rangle$ gilt.

Bei der Lösung dieser Aufgabe dürfen Sie die λ -Terme $\langle succ \rangle$, $\langle pred \rangle$, $\langle iszero \rangle$, $\langle ite \rangle$, $\langle add \rangle$, $\langle mult \rangle$ und $\langle n \rangle$ mit $n \in \mathbb{N}$ als bekannt voraussetzen.

2. Aufgabe: (AGS 2.27*)

Erstellen Sie einen Ableitungsbaum für das unten angegebene C_0 -Programm. Verwenden Sie hierfür das Wissen über BNF-Definitionen aus der Vorlesung AuD. Auf Knoten mit der Beschriftung $\langle Ident \rangle$, $\langle BoolExpression \rangle$ oder $\langle SimpleExpression \rangle$ sollen jeweils direkt Blätter mit dem jeweiligen Wort folgen.

```
#include <stdio.h>

int main() {
    int a, b;

    scanf("%i", &a);
    scanf("%i", &b);
    a = a * a + b * b;
    printf("%d", a);
    return 0;
}
```

3. Aufgabe: (AGS 12.6* b, c)

(b) Folgendes AM_0 -Programm sei gegeben:

1: LIT 0;	7: JMC 17;	13: LIT 1;
2: STORE 2;	8: LOAD 1;	14: ADD;
3: READ 1;	9: LIT 2;	15: STORE 2;
4: LOAD 1;	10: SUB;	16: JMP 4;
5: LIT 0;	11: STORE 1;	17: WRITE 2;
6: GT;	12: LOAD 2;	

Lassen Sie dieses Programm auf der AM_0 mit der Startkonfiguration $(10, 2 : 2, [1/2, 2/0], \varepsilon, \varepsilon)$ schrittweise ablaufen.

(c) Geben Sie für das unter (b) gegebene AM_0 -Programm für den Befehlsbereich 4 bis 17 die zugehörigen C_0 -Statements an, deren Übersetzungen zu dieser AM_0 -Befehlsfolge geführt haben. Nehmen Sie dabei an, dass der Speicherplatz i für die Variable xi vergeben wurde.

Zusatzaufgabe: (AGS 12.7* b, c)

(b) Folgender Ausschnitt aus einem AM_0 -Programm sei gegeben:

9: ...	14: LOAD 2;	19: LOAD 2;
10: LOAD 1;	15: LOAD 1;	20: STORE 1;
11: LOAD 2;	16: SUB;	21: ...
12: LT;	17: STORE 1;	
13: JMC 19;	18: JMP 21;	

Lassen Sie dieses Programm auf der AM_0 mit der Ausgangskonfiguration $(10, \varepsilon, [1/2, 2/3, 3/0], \varepsilon, \varepsilon)$ schrittweise ablaufen bis der Befehlszähler größer bzw. gleich 21 ist.

(c) Geben Sie für den unter (b) gegebenen Ausschnitt aus einem AM_0 -Programm die zugehörigen C_0 -Statements an, deren Übersetzung (bis auf eine eventuelle Verschiebung der Befehlsadressen) zu dieser AM_0 -Befehlsfolge führt. Vergeben Sie dabei für den Speicherplatz i die Variable xi .