LV „ Programmierung “ SS 11
Prof. Dr. H. Vogler / Dipl.-Math. R. Vater

Aufgabenblatt zur 5. Übung

Zeitraum: 09.05. bis 13.05.2011

Achtung: Übungsraumverlegung: einmalig am Freitag, dem 13.05.2011, 1. DS
von INF/E023 nach HSZ/E01.

1. Aufgabe: (Klausuraufgabe 02.2011)

Folgende Definitionen seien gegeben:

```
add :: Int -> [Int] -> [Int]
add a [] = []
add a (x:xs) = a+x:add a xs
```

Die folgende Aussage soll mittels struktureller Induktion über Listen bewiesen werden. Für jede Liste `l :: [Int]` und jedes Paar von ganzen Zahlen `a, b :: Int` gilt:

`add b (add a l) = add (a+b) l.`

Bearbeiten Sie die folgenden Teilaufgaben; geben Sie jeweils die benutzten Gesetzmäßigkeiten oder Definitionen an.

(a) Zeigen Sie den Induktionsanfang.

(b) Geben Sie die Induktionsvoraussetzung *vollständig* an.

(c) Zeigen Sie den Induktionsschritt.

2. Aufgabe: (AGS 11.35)

Folgende Definitionen seien gegeben:

```
data Tree = Leaf Int | Branch Int Tree Tree

prod:: [Int] -> Int
prod [] = 1
prod (a:xs) = a * (prod xs)

prodTree:: Tree -> Int
prodTree (Leaf x) = x
prodTree (Branch x y z) = x * (prodTree y) * (prodTree z)

collapse:: Tree -> [Int]
collapse (Leaf x) = [x]
collapse (Branch x y z) = (collapse y) ++ [x] ++ (collapse z)
```

Zeigen Sie unter Verwendung der oben aufgeführten Definitionen durch strukturelle Induktion die Gültigkeit der folgenden Gleichung:

```
prodTree t = prod (collapse t)
```

Gehen Sie davon aus, dass die Beziehung `prod (x ++ y) = (prod x) * (prod y)` bereits bewiesen ist.

3. Aufgabe: (AGS 11.55)

(a) Gegeben sei folgender λ–Term:

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

Reduzieren Sie diesen Term solange, 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) Eine Funktion $f : I\!N^+ \times I\!N^+ \to I\!N^+$ sei wie folgt definiert:

$$\begin{array}{lcllll} f(x,y) & = & y & & & \text{für } x = 1 \\ f(x,y) & = & 2 & * & f(x-1,y+1) & \text{wenn } x \text{ teilbar durch } 3 \\ f(x,y) & = & 3 & * & f(x-1,y) & \text{sonst} \end{array}$$

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

(c) Folgender λ–Term sei gegeben:

$$\begin{array}{lcl} < G > & = & (\lambda gxy. < ite > (< iszero > (< pred > x))\ (< succ > y) \quad \hookleftarrow^{1} \\ & & (< mult > (< succ > y)\ (g(< pred > x)(< succ > (< succ > y))))) \end{array}$$

Berechnen Sie schrittweise $< Y >< G >< 2 >< 3 >$. Führen Sie im Rechenprozess zweckmäßige Abkürzungen der λ–Terme ein.

Bei der Lösung dieser Aufgabe dürfen Sie die λ-Terme $< succ >, < pred >, < iszero >$, $< true >, < false >, < ite >, < add >, < mult >, < mod >$ und $< n >$ mit $n \in I\!N$ als bekannt voraussetzen. Des Weiteren dürfen Sie bei der Lösung die folgenden Beziehungen benutzen:

$$< succ >< n > \Longrightarrow^* < n+1 > , \quad < pred >< n > \Longrightarrow^* < n-1 > \text{ für } n > 0 ,$$

$$< add >< n1 >< n2 > \Longrightarrow^* < n1 + n2 > , \quad < mult >< n1 >< n2 > \Longrightarrow^* < n1 * n2 >$$

$$< mod >< n >< m > \Longrightarrow^* < z > \text{ wobei } 0 \le z < m \text{ mit } z = n - i \cdot m,\ i \in I\!N$$

$$< ite >\ \ s\ s_1\ s_2 \Longrightarrow^* \begin{cases} s_1\ wenn\ s \Longrightarrow^* < true > \\ s_2\ sonst \end{cases}$$

$$< iszero >\ \ s \Longrightarrow^* \begin{cases} < true >\ \ wenn\ s \Longrightarrow^* < 0 > \\ < false >\ \ sonst \end{cases}$$

$$< Y > = (\lambda h.((\lambda y.h(yy))(\lambda y.h(yy))))$$

Zusatzaufgabe 1: (AGS 11.48*)

(a) Bestimmen Sie für jeden der folgenden λ-Terme t die Mengen $FV(t)$ und $GV(t)$:

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

[1] Zeilenumbruch, λ–Term erstreckt sich über 2 Zeilen.

(b) Reduzieren Sie die folgenden λ-Terme zu Normalformen. Schreiben Sie – bevor Sie einen Ableitungsschritt ausführen – für die relevanten (Teil-)Ausdrücke die Mengen der freien bzw. der gebundenen Vorkommen von Variablen auf.

- $(\lambda x.(\lambda y.x\ z\ (y\ z)))\ (\lambda x.y\ (\lambda y.y))$
- $(\lambda x.(\lambda y.(\lambda z.z)))\ x\ (+\ y\ 1)$
- $(\lambda x.(\lambda y.x\ (\lambda z.y\ z)))\ (((\lambda x.(\lambda y.y))\ 8)\ (\lambda x.(\lambda y.y)\ x))$
- $(\lambda h.(\lambda x.h\ (x\ x))\ (\lambda x.h\ (x\ x)))\ ((\lambda x.x)\ (+\ 1\ 5))$
- $(\lambda f.(\lambda a.(\lambda b.f\ a\ b)))\ (\lambda x.(\lambda y.x))$

Zusatzaufgabe 2: (AGS 11.47)

Folgende Definitionen seien gegeben:

```
data Tree = Leaf Int | Node Tree Tree

add :: Tree -> Int
add (Leaf a)        = a
add (Node t1 t2)    = (add t1) + (add t2)

sub :: Tree -> Int
sub (Leaf a)        = a
sub (Node t1 t2)    = (sub t1) - (sub t2)

neg :: Int -> Tree -> Tree
neg i (Leaf a)     = Leaf (a*i)
neg i (Node t1 t2) = Node (neg i t1) (neg (-i) t2)
```

Zeigen Sie für die oben aufgeführten Definitionen mit Hilfe der Induktion über Bäumen, dass die folgende Gleichung für jeden Baum `t :: Tree` und jede ganze Zahl `i :: Int` erfüllt ist:

```
add (neg i t) = i * (sub t) .
```

Geben Sie bei Umformungen die jeweils benutzten Gesetzmäßigkeiten/Definitionen an.