Aufgabenblatt zur 5. Übung

Zeitraum: 17.05. bis 21.05.2010

1. Aufgabe: (Klausuraufgabe 02.2010)

Folgende Definitionen seien gegeben:

```
sum :: [Int] -> Int
sum []          = 0
sum (a:l)       = a + (sum l)

mix :: [Int] -> [Int] -> [Int]
mix [] []       = []
mix (a:l) (b:r) = (b-a) : (mix l r)

scale :: [Int] -> Int -> [Int]
scale [] i      = []
scale (a:l) i   = (a*i) : (scale l i)
```

Zeigen Sie für die oben aufgeführten Definitionen mit Hilfe der Induktion über Listen, dass die folgende Gleichung für alle Listen `l, r :: [Int]` **gleicher Länge** und jede ganze Zahl `i :: Int` erfüllt ist:

`(sum l) - (i * sum r) = sum (mix (scale r i) l)`.

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

2. Aufgabe: (AGS 11.33)

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 11.45*)

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

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

(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: (AGS 11.39*)

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

sumTree :: Tree -> Int
sumTree (Leaf i) = i
sumTree (Branch i t1 t2) = i + (sumTree t1) + (sumTree t2)

revTree :: Tree -> Tree
revTree (Leaf i) = Leaf i
revTree (Branch i t1 t2) = Branch i (revTree t2) (revTree t1)

overlay :: Tree -> Tree -> Tree
overlay (Leaf i1) (Leaf i2) = Leaf (i1+i2)
overlay (Branch i1 t11 t12) (Branch i2 t21 t22) =
    Branch (i1+i2) (overlay t11 t21) (overlay t12 t22)
```

Zeigen Sie unter Verwendung der oben aufgeführten Definitionen mit Hilfe der strukturellen Induktion die Gültigkeit der Gleichung

`sumTree (overlay t t) = 2 * (sumTree (revTree t))` für jedes `t :: Tree`.

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