

Aufgabenblatt zur 4. Übung

Zeitraum: 02.05. bis 06.05.2011

1. Aufgabe: (AGS 11.31)

Es seien δ ein dreistelliges, σ ein zweistelliges, γ ein einstelliges und α ein nullstelliges Funktionssymbol. $V = \{x_1, x_2, x_3\}$ sei eine Menge von Variablen.

Wenden Sie den Unifikationsalgorithmus auf die Terme t_1 und t_2 an und ermitteln Sie deren allgemeinsten Unifikator:

$$t_1 = \delta(\gamma(x_3), \gamma(\gamma(\alpha)), \sigma(\gamma(x_2), x_1))$$

$$t_2 = \delta(\gamma(x_3), \gamma(x_3), \sigma(\gamma(\alpha), \gamma(\gamma(x_2))))$$

Wenden Sie bei jedem Umformungsschritt nur eine Regelsorte an und geben Sie diese jeweils an.

2. Aufgabe: (AGS 11.34*)

Gegeben seien folgende Definitionen von Funktionen:

```
1  sumlist1 :: [Int] -> Int
2  sumlist1 [] = 0
3  sumlist1 (b:x) = b + (sumlist1 x)
4
5  sumlist2 :: [(Int,Int)] -> Int
6  sumlist2 [] = 0
7  sumlist2 ((a,b):y) = (a + b) + (sumlist2 y)
8
9  zip :: [t] -> [u] -> [(t,u)]
10 zip [] [] = []
11 zip (a:x) [] = []
12 zip [] (b:y) = []
13 zip (a:x) (b:y) = ((a,b):(zip x y))
```

Zeigen Sie mit Hilfe des Prinzips der Induktion über Listen, dass folgende Gleichung unter der Bedingung $length(x1) = length(x2)$ gilt:

$$sumlist2 (zip x1 x2) = (sumlist1 x1) + (sumlist1 x2)$$

3. Aufgabe: (AGS 11.44)

Folgende Definitionen seien gegeben:

```
1  data Tree = Leaf Float | Branch Float Tree Tree
2
3  add :: Tree -> Float -> Tree
4  add (Leaf x)      a = Leaf (x+a)
5  add (Branch x l r) a = Branch (x+a/3) (add l (a/3)) (add r (a/3))
6
7  rev :: Tree -> Tree
8  rev (Leaf x)      = Leaf x
9  rev (Branch x l r) = Branch x (rev r) (rev l)
10
11 sum :: Tree -> Float
12 sum (Leaf x)      = x
13 sum (Branch x l r) = x + (sum l) + (sum r)
```

Zeigen Sie für die oben aufgeführten Definitionen mit Hilfe der strukturellen Induktion, dass die folgende Gleichung für einen beliebigen Baum $t :: \text{Tree}$ und eine beliebige Zahl $a :: \text{Float}$ erfüllt ist:

$$\text{sum (add t a)} = (\text{sum (rev t)}) + a$$

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

Zusatzaufgabe: (AGS 11.41*)

```
1  data Tree = Leaf Int | Branch Int Tree Tree
2
3  sumTree :: Tree -> Int
4  sumTree (Leaf i) = i
5  sumTree (Branch i t1 t2) = i + (sumTree t1) + (sumTree t2)
6
7  revTree :: Tree -> Tree
8  revTree (Leaf i) = Leaf i
9  revTree (Branch i t1 t2) = Branch i (revTree t2) (revTree t1)
10
11 overlay :: Tree -> Tree -> Tree
12 overlay (Leaf i1) (Leaf i2) = Leaf (i1+i2)
13 overlay (Branch i1 t11 t12) (Branch i2 t21 t22) =
14   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

$\text{sumTree (overlay t t)} = 2 * (\text{sumTree (revTree t)})$ für jedes $t :: \text{Tree}$.

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