
Programmierung

Aufgabe 1 (AGS 12.2.14)

- (a) Gegeben seien die Terme $t_1 = \delta(\alpha, \sigma(x_1, \alpha), \sigma(x_2, x_3))$ und $t_2 = \delta(\alpha, \sigma(x_1, x_2), \sigma(x_2, \gamma(x_2)))$ über dem Rangalphabet $\Sigma = \{\delta^{(3)}, \sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$. Wenden Sie den Unifikationsalgorithmus auf die Terme t_1 und t_2 an. Wenden Sie bei jedem Umformungsschritt nur eine Regelsorte an und geben Sie diese jeweils an. Geben Sie anschließend den von Ihnen bestimmten allgemeinsten Unifikator an.
- (b) Gegeben seien die Haskell-Typsterme

$$t_1 = (\mathbf{a}, [\mathbf{a}]), \quad t_2 = (\text{Int}, [\text{Double}]) \quad \text{und} \quad t_3 = (\mathbf{b}, \mathbf{c}).$$

Welche Paare dieser Terme sind unifizierbar? Geben Sie ggf. einen allgemeinsten Unifikator an!

Aufgabe 2 (AGS 12.2.13)

Gegeben seien folgende Terme über dem Rangalphabet $\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$:

$$t_1 = \sigma(\gamma(x_2), \sigma(\gamma(\alpha), x_3)) \quad \text{und}$$

$$t_2 = \sigma(x_1, \sigma(\gamma(\alpha), \sigma(\alpha, x_1))).$$

- (a) Wenden Sie den Unifikationsalgorithmus auf t_1 und t_2 an. Geben Sie den allgemeinsten Unifikator an.
- (b) Geben Sie zwei weitere Unifikatoren an.

Aufgabe 3 (AGS 12.3.20)

Zeigen Sie unter Verwendung der folgenden Definitionen durch strukturelle Induktion die Gültigkeit der Gleichung `sum (foo xs) = 2 * sum xs - length xs` für jedes `xs :: [Int]`.

```

1 foo :: [Int] -> [Int]
2 foo []      = []
3 foo (x:xs) = x : x : (-1) : foo xs
4
5 sum :: [Int] -> Int
6 sum []      = 0
7 sum (x:xs) = x + sum xs
8
9 length :: [Int] -> Int
10 length []      = 0
11 length (x:xs) = 1 + length xs

```

Zusatzaufgabe 1 (AGS 12.2.5)

Gegeben sind die Terme t_1 und t_2 über dem Rangalphabet $\Sigma = \{\delta^{(3)}, \sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$:

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

$$t_2 = \delta(\sigma(\gamma(\alpha), \gamma(\gamma(x_3))), \gamma(x_1), \gamma(x_1))$$

Wenden Sie den Unifikationsalgorithmus auf die Terme t_1 und t_2 an und ermitteln Sie deren allgemeinsten Unifikator. Wenden Sie bei jedem Umformungsschritt nur eine Regelsorte an und geben Sie diese jeweils an.

Zusatzaufgabe 2 (AGS 12.3.21)

Folgende Definitionen seien gegeben:

```
1 dup :: [Int] -> [Int]
2 dup [] = []
3 dup (x:xs) = x : (dup xs ++ dup xs)
4
5 pow :: [Int] -> Int
6 pow [] = 1
7 pow (x:xs) = 2 * pow xs
8
9 length :: [Int] -> Int
10 length [] = 0
11 length (x:xs) = 1 + length xs
```

Die folgende Aussage soll mittels struktureller Induktion über Listen bewiesen werden:

(A): Für jede Liste $xs :: [Int]$ gilt:

$$\text{length (dup xs)} = \text{pow xs} - 1.$$

Bearbeiten Sie die folgenden Teilaufgaben; geben Sie bei jeder Umformung die benutzte *Definition* oder *Beziehung*, bzw. die *Induktionsvoraussetzung* an; quantifizieren Sie alle Variablen.

- (a) Zeigen Sie den Induktionsanfang.
- (b) Geben Sie die Induktionsvoraussetzung *vollständig* an.
- (c) Zeigen Sie den Induktionsschritt. Sie dürfen folgende bereits bewiesene Beziehung nutzen.

(B): Für alle Listen $ys, zs :: [Int]$ gilt:

$$\text{length (ys ++ zs)} = \text{length ys} + \text{length zs}.$$