# Programmierung

### Aufgabe 1 (AGS 12.1.16 a, 12.1.48 b,c, 12.1.56)

Gegeben sei der polymorphe algebraische Datentyp

```
data BinTree a = Branch a (BinTree a) (BinTree a) | Leaf a
```

(a) Schreiben Sie einen Baum des Typs `BinTree Int` auf, in dem der Konstruktor `Leaf` mindestens 5-Mal vorkommt.

(b) Geben Sie eine Funktion `depth :: BinTree a -> Int` an, welche für einen Baum `t` die Länge des kürzesten Pfads von der Wurzel zu einem Blattknoten von `t` berechnet. Die Länge eines Pfads ist dabei die Anzahl der auf ihm vorkommenden Knoten, d.h. der Pfad von der Wurzel zur Wurzel selbst hat die Länge `1`.

(c) Geben Sie eine Funktion `paths :: BinTree a -> BinTree [a]` an, die in einem Baum die Beschriftung jedes Knotens $u$ durch die Liste der Knotenbeschriftungen auf dem Pfad vom Wurzelknoten zu $u$ ersetzt. Ist der Wurzelknoten z. B. in `t` mit `5` beschriftet, so ist er in `paths t` mit `[5]` beschriftet, und ist sein erster Kindknoten in `t` mit `3` beschriftet, dann ist dieser in `paths t` mit `[5,3]` beschriftet.

(d) Schreiben Sie eine Funktion `tmap :: (a -> b) -> BinTree a -> BinTree b`, so dass für alle `f :: a -> b` und `t :: BinTree a` gilt, dass `tmap f t` der Baum ist, der entsteht, indem jeder Knoten `v` von `t` durch `f v` ersetzt wird.

### Aufgabe 2 (AGS 12.2.12, 12.2.16 b, 12.2.14 b)

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

$$t_1 = \sigma(\sigma(x_1, \alpha), \sigma(\gamma(x_3), x_3)) \qquad \text{und} \qquad t_2 = \sigma(\sigma(\gamma(x_2), \alpha), \sigma(x_2, x_3)) \ .$$

(a) 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) Geben Sie zwei weitere Unifikatoren an.

(c) Geben Sie zwei Terme $t_1$ und $t_2$ über dem Alphabet $\Sigma$ an, so dass im Laufe der Anwendung des Unifikationsalgorithmus auf $t_1$ und $t_2$ der Occur-Check fehlschlägt.

(d) Gegeben seien die Haskell-Typterme

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

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


*https://www.orchid.inf.tu-dresden.de* *thomas.ruprecht@tu-dresden.de*

## Zusatzaufgabe 1 (AGS 12.1.65 b,c ⋆)

(a) Geben Sie eine polymorphe Haskell-Funktion `apps` samt Typdefintion an, die für eine gegebene Liste *fs* von Funktionen und eine Liste *xs* eine Liste zurückgibt, die für *jede* Funktion $f$ in *fs und jedes* Element $x$ in *xs* das Element $f(x)$ enthält und sonst nichts. Beispielweise soll die Liste `apps [f, g] [x, y]` die Elemente `f x`, `g x`, `f y` und `g y` enthalten.

(b) Gegeben ist die folgende Typdefinition `VarTree`, die zum Aufbau eines Baumes, bei dem jeder Knoten eine beliebige Anzahl an Kindbäumen haben kann, genutzt wird. Wir betrachten dabei jedes Vorkommen des Konstruktors `Node` als Knoten eines Baumes, das Feld des Typs `a` als dessen Beschriftung und die Liste über dem Typ `VarTree` als dessen Nachfolger. Jedes Vorkommen des Konstruktors `Var` soll ein Variablensymbol im Baum darstellen.

```
data VarTree a = Node a [VarTree a] | Var Int
```

Schreiben Sie eine Funktion `insert :: [VarTree a] -> VarTree a -> VarTree a`, sodass der Baum `insert ts u` aus `u` hervorgeht, indem jedes Vorkommen des Konstruktors `Var i` durch den `i`-ten (beginnend bei 1) Baum in `ts` ersetzt wird. Gehen Sie davon aus, dass für alle Vorkommen von `Var i` in `u` ein `i`-tes Element in `ts` existiert.

## Zusatzaufgabe 2 (AGS 12.2.1 ⋆)

Gegeben seien die folgenden Terme über dem Rangalphabet $\Sigma = \{\delta^{(3)}, \gamma^{(1)}, \alpha^{(0)}\}$. Wenden Sie den Unifikationsalgorithmus jeweils auf die Terme $t_1$ und $t_2$ an und ermitteln Sie jeweils den allgemeinsten Unifikator:

(a) $t_1 = \delta(\gamma(x_1), \delta(\gamma(\alpha), \gamma(x_2), \gamma(\gamma(\alpha))), x_3)$
$t_2 = \delta(\gamma(\gamma(x_5)), \delta(x_4, \gamma(x_2), \gamma(x_1)), \alpha)$

(b) $t_1 = \delta(\gamma(\alpha), x_3, \gamma(\gamma(x_3)))$
$t_2 = \delta(x_1, \delta(\alpha, x_2, \alpha), x_2)$