
Programmierung

Aufgabe 1 (AGS 12.3.20)

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

```

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

```

Zeigen Sie dazu den Induktionsanfang und den Induktionsschritt; geben Sie beim Induktionsschritt die Induktionsvoraussetzung an. Geben Sie bei jeder Umformung die benutzte *Definition*, *Eigenschaft* bzw. *Induktionsvoraussetzung* an. Quantifizieren Sie alle Variablen.

Aufgabe 2 (AGS 12.3.29 ★)

Folgende Definitionen seien gegeben:

```

1 data BinTree a = Node a (BinTree a) (BinTree a) | Leaf a
2
3 preOrder :: BinTree a -> [a]
4 preOrder (Leaf x) = [x]
5 preOrder (Node x l r) = [x] ++ preOrder l ++ preOrder r
6
7 mPostOrder :: BinTree a -> [a]
8 mPostOrder (Leaf x) = [x]
9 mPostOrder (Node x l r) = mPostOrder r ++ mPostOrder l ++ [x]

```

Sei außerdem $\text{rev} :: [a] \rightarrow [a]$ eine Funktion, sodass für jeden Typ a folgende zwei Eigenschaften gelten:

$$\forall x :: a: \quad \text{rev } [x] = [x] \quad (\text{H1})$$

$$\forall xs, ys :: [a]: \quad \text{rev } (xs ++ ys) = \text{rev } ys ++ \text{rev } xs \quad (\text{H2})$$

Gehen Sie davon aus, dass die Funktion $(++) :: [a] \rightarrow [a] \rightarrow [a]$ assoziativ ist.

(a) Sei a ein Typ, $x :: a$ und $xs, ys :: [a]$. Zeigen Sie, dass folgende Gleichung gilt:

$$[x] ++ \text{rev } ys ++ \text{rev } xs = \text{rev } (xs ++ ys ++ [x]) \quad (\text{H3})$$

Hinweis: Sie dürfen (H1) und (H2) verwenden. Für den Beweis der Gültigkeit dieser Gleichung ist *keine* Induktion nötig.

(b) Zeigen Sie durch strukturelle Induktion, dass die Aussage

$$\text{preOrder } t = \text{rev } (\text{mPostOrder } t)$$

Für jeden Typ a und jeden Baum $t :: \text{BinTree } a$ gilt. Zeigen Sie dazu den Induktionsanfang und den Induktionsschritt; geben Sie beim Induktionsschritt die Induktionsvoraussetzung an. Geben Sie bei jeder Umformung die benutzte *Definition*, *Eigenschaft* bzw. *Induktionsvoraussetzung* an. Quantifizieren Sie alle Variablen.

Hinweis: Sie dürfen dafür die Eigenschaften (H1), (H2) und (H3) verwenden.

Zusatzaufgabe 1 (AGS 12.3.28 ★)

Folgende Definitionen seien gegeben:

```
1 data IntTree = Node Int [IntTree]
2
3 yield :: IntTree -> [Int]
4 yield (Node i []) = [i]
5 yield (Node i ts) = concat (map yield ts)
6
7 yieldProd :: IntTree -> Int
8 yieldProd (Node i []) = i
9 yieldProd (Node i ts) = product (map yieldProd ts)
```

Zeigen Sie unter Verwendung der obigen Definitionen durch strukturelle Induktion die Gültigkeit der Gleichung

$$\text{product } (\text{yield } t) = \text{yieldProd } t$$

für jeden Baum $t :: \text{IntTree}$. Sie dürfen dabei nutzen, dass für alle Typen a, b , positive Zahlen $k > 0$, ganze Zahlen $i :: \text{Int}$, Funktionen $f :: a \rightarrow b$, Werte $a_1, \dots, a_k :: a$ und Listen $l_1, \dots, l_k :: [\text{Int}]$ folgende Eigenschaften gelten:

$$\text{product } [i] = i \tag{H1}$$

$$\text{map } f [a_1, \dots, a_k] = [f a_1, \dots, f a_k] \tag{H2}$$

$$\text{product } (\text{concat } [l_1, \dots, l_k]) = \text{product } [\text{product } l_1, \dots, \text{product } l_k] \tag{H3}$$

Zeigen Sie dazu den Induktionsanfang und den Induktionsschritt; geben Sie beim Induktionsschritt die Induktionsvoraussetzung an. Geben Sie bei jeder Umformung die benutzte *Definition*, *Eigenschaft* bzw. *Induktionsvoraussetzung* an. Quantifizieren Sie alle Variablen.

Zusatzaufgabe 2 (AGS 12.3.32 ★)

Folgende Definitionen seien gegeben:

```
1 data BinTree = Node BinTree BinTree | Leaf Int
2
3 leafNum :: BinTree -> Int
```

```

4 leafNum (Leaf x) = 1
5 leafNum (Node l r) = leafNum l + leafNum r
6
7 treeSum :: BinTree -> Int
8 treeSum (Leaf x) = x
9 treeSum (Node l r) = treeSum l + treeSum r
10
11 toNode :: Int -> BinTree -> BinTree
12 toNode a (Leaf x) = Leaf (a * x + 1)
13 toNode a (Node l r) = Node (toNode a l) (toNode a r)

```

Zeigen Sie durch strukturelle Induktion, dass für jeden Baum $t :: \text{BinTree}$ und jede ganze Zahl $a :: \text{Int}$ die folgende Gleichung erfüllt ist:

$$a * \text{treeSum } t + \text{leafNum } t = \text{treeSum } (\text{toNode } a \ t)$$

Zeigen Sie dazu den Induktionsanfang und den Induktionsschritt; geben Sie beim Induktionsschritt die Induktionsvoraussetzung an. Geben Sie bei jeder Umformung die benutzte *Definition*, *Eigenschaft* bzw. *Induktionsvoraussetzung* an. Quantifizieren Sie alle Variablen.

Hinweis: Benutzen Sie die üblichen arithmetischen Rechenregeln.