Aufgabe 1 (AGS 12.3.20)

Dienstag

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
```

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.

(IA) Sei xs = []. Dann gilt

- links: $\text{sum}\ (\text{foo}\ xs) = \text{sum}\ (\text{foo}\ [\,]) \overset{\#2}{=} \text{sum}\ (\ [\,]\ ) \overset{\#6}{=} 0$

- rechts: $2 * \text{sum}\ xs - \text{length}\ xs = 2 * \text{sum}\ [\,] - \text{length}\ [\,] \overset{\#6}{=} 2 * 0 - \text{length}\ [\,] \overset{\#10}{=} 2 * 0 - 0 = 0$

$\Rightarrow$ "IA gilt"

(IV) Sei xs :: [Int], sodass gilt: $\text{sum}\ (\text{foo}\ xs) = 2 * \text{sum}\ xs - \text{length}\ xs$

(IS) Sei x :: Int (beliebig). Dann gilt

$$
\begin{aligned}
\text{sum}\ (\text{foo}\ (x:xs)) &\overset{\#3}{=} \text{sum}(\ x : x : (-1) : \text{foo}\ xs\ ) \\
&\overset{3\cdot\#7}{=} x + x + (-1) + \text{sum}(\text{foo}\ xs) \\
&\overset{IV}{=} x + x + (-1) + 2 * \text{sum}\ xs - \text{length}\ xs \\
&\overset{\text{Komm, Distr.}}{=} 2 * x + 2 * \text{sum}\ xs - (1 + \text{length}\ xs) \\
&\overset{\text{Distr.}}{=} 2 * (x + \text{sum}\ xs) - (1 + \text{length}\ xs) \\
&\overset{\#7}{=} 2 * \text{sum}\ (x:xs) - (1 + \text{length}\ xs) \\
&\overset{\#11}{=} 2 * \text{sum}\ (x:xs) - \text{length}\ (x:xs)
\end{aligned}
$$

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 `rev :: [a] -> [a]` eine Funktion, sodass für jeden Typ `a` folgende zwei Eigenschaften gelten:

∀`x :: a`:	`rev [x]` = `[x]`	(H1)
∀`xs, ys :: [a]`:	`rev (xs ++ ys)` = `rev ys ++ rev xs`	(H2)

Gehen Sie davon aus, dass die Funktion `(++) :: [a] -> [a] -> [a]` assoziativ ist.

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

`[x] ++ rev ys ++ rev xs` = `rev (xs ++ ys ++ [x])` (H3)

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

$$
\begin{aligned}
[x] \mathbin{++} \text{rev } ys \mathbin{++} \text{rev } xs &\overset{\text{Ass.}}{=} [x] \mathbin{++} \underline{(\text{rev } ys \mathbin{++} \text{rev } xs)} \\
&\overset{H2}{=} [x] \mathbin{++} \text{rev}(xs \mathbin{++} ys) \\
&\overset{H1}{=} \text{rev } [x] \mathbin{++} \text{rev}(xs \mathbin{++} ys) \\
&\overset{H2}{=} \text{rev}((xs \mathbin{++} ys) \mathbin{++} [x]) \\
&\overset{\text{Ass.}}{=} \text{rev}(xs \mathbin{++} ys \mathbin{++} [x])
\end{aligned}
$$

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

`preOrder t` = `rev (mPostOrder t)`

Für jeden Typ `a` und jeden Baum `t :: 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.

(IA) Sei $x :: a$ (beliebig) und $t = \text{Leaf } x$.

- links: $\text{preOrder } t = \text{preOrder } \underline{(\text{Leaf } x)} \overset{\#4}{=} [x]$

- rechts: $\text{rev}(\text{mPostOrder } t) = \text{rev}(\text{mPostOrder }(\text{Leaf } x)) \overset{\#8}{=} \text{rev}([x]) \overset{H1}{=} [x]$

⟹ "links = rechts" ✓

(IV) Seien $l, r ::$ BinTree a, sodass

(IV1) $\text{preOrder } l = \text{rev }(\text{mPostOrder } l)$

(IV2) $\text{preOrder } r = \text{rev }(\text{mPostOrder } r)$

(IS) Sei $x :: a$ beliebig und $t = \text{Node}\ x\ l\ r$.

$$\text{preOrder}\ (\text{Node}\ x\ l\ r)$$
$$\overset{\#5}{=} [x] \mathbin{++} \underline{\text{preOrder}\ l} \mathbin{++} \text{preOrder}\ r$$
$$\overset{IV1}{=} [x] \mathbin{++} \text{rev}\ (\text{mPostOrder}\ l) \mathbin{++} \underline{\text{preOrder}\ r}$$
$$\overset{IV2}{=} \underline{[x]} \mathbin{++} \text{rev}\ (\underbrace{\text{mPostOrder}\ l}_{ys}) \mathbin{++} \text{rev}\ (\underbrace{\text{mPostOrder}\ r}_{xs})$$
$$\overset{\#3}{=} \text{rev}\ (\overset{xs}{\text{mPostOrder}\ r} \mathbin{++} \overset{ys}{\text{mPostOrder}\ l} \mathbin{++} \overline{[x]})$$
$$\overset{\#9}{=} \text{rev}\ (\text{mPostOrder}\ (\text{Node}\ x\ l\ r))$$

---

data RoseTree a = Node [RoseTree a]

(IV) Sei $k \in \mathbb{N}$ und $t_1, \dots, t_k :: \text{RoseTree}\ a$, sodass

$$P(t_i) \qquad \forall i \in \{1, \dots, k\}$$