Alp. Übungsblatt 7

1. Aufgabe (7 Punkte)

Geben Sie verbale geschlossene Beschreibungen oder mathematische Formeln für die folgenden in Haskell definierten Funktionen an, und beweisen Sie Ihre Antwort.

a)

```
f :: Int -> Int
f 0 = 1
f n = sum [f x | x <- [0..(n-1)]]
```

als Funktion auf den natürlichen Zahlen N.

Wir benötigen folgende Haskell-Definitionen:

```
      f :: Int -> Int
(1)   f 0 = 1
(2)   f n = sum [f x | x <- [0..(n-1)]]

      sum :: [Int] -> Int
(3)   sum [] = 0
(4)   sum (x:xs) = x + sum xs
```

Wir zeigen zunächst folgenden Hilfssatz:

(5) $\forall$ `xs :: [Int]`: `sum xs` $= \sum_{y \in xs} y$,

wobei die endliche Liste xs eine Zahl mehrfach enthalten kann.

Beweis: per Induktion über `xs`

I.A. `xs = []`

`sum []`	=	0	(3)
	=	$\sum_{y \in \{\}} y$	(RR)

I.V. $\exists$ `xs :: [Int]`: `sum xs` $= \sum_{y \in xs} y$

I.S. `xs` $\mapsto$ `(x:xs)`

`sum (x:xs)`	=	`x + sum xs`	(4)
	=	`x` $+ \sum_{y \in xs} y$	(I.V.)
	=	$\sum_{y \in x:xs} y$	(RR)

qed

Nun der eigentliche Beweis.

Behauptung: $\forall n \in \mathrm{N}: \texttt{f n} = \left| 2^{n-1} \right|$

Beweis: per Induktion über n

I.A. `n = 0`

`f 0`	=	`1`	(1)
	=	$\left\| 2^{-1} \right\|$	(RR)
	=	$\left\| 2^{0-1} \right\|$	(RR)

I.V. $\exists n \in \mathrm{N}: \texttt{f n} = \left| 2^{n-1} \right|$

I.S. $\texttt{n} \mapsto \texttt{n+1}$

`f (n+1)`	=	`sum [f x | x <- [0..n]]`	(2)
	=	$\sum_{y \in \{\mathrm{f\ x}, x \in \{0,..,n\}\}} y$	(5)
	=	$\sum_{y \in \{\mathrm{f\ x}, x \in \{0,..,n-1\}\}} y$ `+ f n`	(RR)
	=	`sum [f x | x <- [0..(n-1)]] + f n`	(5) rückwärts
	=	`f n + f n`	(2) rückwärts
	=	$\left\| 2^{n-1} \right\| + \left\| 2^{n-1} \right\|$	(I.V.)
	=	$2 \cdot \left\| 2^{n-1} \right\|$	(RR)
	=	$\left\| 2^{(n-1)+1} \right\|$	(RR)
	=	$\left\| 2^{(n+1)-1} \right\|$	(RR)

qed

b)

```haskell
f :: String -> String
f [] = []
f (x:xs) = if x=='a' then fs else x:fs
    where fs = f xs
```

Wir benötigen folgende Haskell-Definition:

```haskell
    f :: String -> String
(1) f [] = []
(2) f (x:xs) = if x=='a' then fs else x:fs
        where fs = f xs
```

Behauptung: f entfernt alle ’a’s aus einem String.

Beweis: per Induktion über `xs`

I.A. `xs = []`

	`f []`	=	`[]`	(1)
$\Rightarrow$	`f []`	=	`[]` ohne ’a’s	

I.V. $\exists$ `xs :: String`: `f xs = xs` ohne ’a’s

I.S. `xs` $\mapsto$ `(x:xs)`

Fallunterscheidung bezüglich `x`:

1. `x =` ’a’

	`f (x:xs)`	=	`f xs`	(2)
$\Rightarrow$	`f xs`	=	`xs` ohne ’a’s	(I.V.)
$\Rightarrow$	`f (x:xs)`	=	`xs` ohne ’a’s	(2) rückwärts

2. `x` $\neq$ ’a’

	`f (x:xs)`	=	`x:(f xs)`	(2)
$\Rightarrow$	`f xs`	=	`xs` ohne ’a’s	(I.V.)
$\Rightarrow$	`x:(f xs)`	=	`x:` (`xs` ohne ’a’s)	Def. von `(:)`
$\Rightarrow$	`f (x:xs)`	=	`(x:xs)` ohne ’a’s	(2) rückwärts

qed