Christoph Lüth
Marcus Ermler
Christian Maeder
Martin Ring
Diedrich Wolter

9. Übungsblatt

Ausgabe: 19.12.12
Abgabe: 11.01.13

9.1 *Right or Wrong?* — *8 Punkte*

Zu Hause nach der Vorlesung sichten Sie Ihre Unterlagen, und entdecken folgende Behauptungen in Ihren Notizen:

$$\texttt{map}\ (f \circ g) = \texttt{map}\ f \circ \texttt{map}\ g \tag{1}$$
$$\texttt{elem}\ x\ xs\ \&\&\ p\ x \Longrightarrow \texttt{elem}\ x\ (\texttt{filter}\ p\ xs) \tag{2}$$
$$\texttt{length}\ (\texttt{take}\ n\ l) = n \tag{3}$$
$$\texttt{map}\ f\ (\texttt{filter}\ (p \circ f)\ xs) = \texttt{filter}\ p\ (\texttt{map}\ f\ xs) \tag{4}$$
$$\texttt{all}\ p\ l = \texttt{null}\ (\texttt{filter}\ p\ l) \tag{5}$$

Leider haben Sie vergessen mitzuschreiben, ob diese wahr oder falsch sind, weil Ihr Nachbar Sie mit lustigen Katzenvideos auf Youtube abgelenkt hat. Finden Sie deshalb für jede der Behauptungen (1)–(5) heraus, ob sie wahr oder falsch ist. Für wahre Behauptungen geben Sie einen Beweis, für falsche Behauptungen ein Gegenbeispiel. [1]

9.2 *Korrekturen* — *2 Punkte*

Natürlich werden in der Vorlesung keine falschen Behauptungen aufgestellt; es muss sich um Flüchtigkeitsfehler beim Abschreiben handeln. Wie können Sie die falschen Behauptungen aus Aufgabe 9.1 mit möglichst geringen Korrekturen in richtige Behauptungen umwandeln?

9.3 *Flippin' Trees* — *10 Punkte*

In der Vorlesung wurde die Funktion flipT auf binären Bäumen eingeführt. Die Funktion inorder wurde schon früher definiert:

```
data Tree α = Null | Node (Tree α) α (Tree α)

flipT :: Tree α→ Tree α
flipT Null         = Null
flipT (Node l a r) = Node (flipT r) a (flipT l)

inorder :: Tree α→ [α]
inorder Null         = []
inorder (Node l a r) = inorder l ++ [a]++ inorder r
```

Zeigen Sie, dass flipT wirklich alle Knoten des Baumes umdreht, indem Sie folgendes Theorem zeigen:

```
inorder (flipT t) = reverse (inorder t)
```

? *Verständnisfragen*

1. Der Datentyp Stream α ist definiert als **data** Stream α=Cons α(Stream α).
   Gibt es für diesen Datentyp ein Induktionsprinzip? Ist es sinnvoll?
2. Welche nichtausführbaren Prädikate haben wir in der Vorlesung kennengelernt?
3. Wie kann man in einem Induktionsbeweis die Induktionsvoraussetzung stärken?
4. Gibt es einen Weihnachtsmann?

[1] Alle in diesen Behauptungen verwendeten Funktionen finden sich im Haskell98 Standard-Prelude.

Revision 1947 vom 2012-12-19