Technische Universität Dresden
Institut für Theoretische Informatik
Dr. Monika Sturm

Wintersemester 09/10

# Funktionale Programmierung und Typtheorie

## 5. Übungsblatt

### Aufgabe 1

a) Beweisen Sie mithilfe der Listeninduktion folgende Aussagen:

   (a) `xs ++ [] = xs`

   (b) `concat (l1 ++ l2) = concat l1 ++ concat l2`

   (c) `length (l1 ++ l2) = (length l1) + (length l2)`

   (d) `reverse (l1 ++ l2) = (reverse l2) ++ (reverse l1)`

b) Beweisen Sie mithilfe der Listeninduktion die in der Vorlesung gezeigte Eigenschaft der Funktion `foldr`

```
foldr f e (xs++ys) = f (foldr f e xs) (foldr f e ys).
```

   Die Eigenschaft gilt für beliebige endliche Listen `xs :: [a]` und `ys :: [a]` sowie assoziative Funktionen `f :: (a -> a -> a)` mit neutralem Element `e :: a`.

c) Beweisen Sie, dass für beliebige Typen `a, b, c`, Werte `e :: c` sowie Funktionen `f :: a -> b` und `g :: b -> c -> c` die folgende Gleichung gilt

```
(foldr g e) . (map f) = foldr h e
```

   mit `h x y = g (f x) y`.

### Aufgabe 2

Die Funktion `flatten` wandelt einen Baum in eine Liste und ist wie folgt definiert:

```
data TTree a = Leaf a | Node (TTree a) (TTree a)

flatten :: TTree a -> [a]
flatten (Leaf n)   = [n]
flatten (Node l r) = flatten l ++ flatten r
```

Die Funktion `myswap :: TTree a -> TTree a` spiegelt einen Baum.

Geben Sie für `myswap` eine Definition an, die folgende Gleichung erfüllt

```
flatten (myswap t) = reverse (flatten t).
```

Beweisen Sie mithilfe struktureller Induktion.

## Aufgabe 3

Demonstrieren Sie anhand eines selbst gewählten Beispiels (unter Verwendung des Typs Binärbaum), dass sich formale Aussagen über Funktionen nicht auf Gleichheiten beschränken müssen.

Hinweis: Zwischen Anzahl der Knoten in einem Binärbaum und seiner Höhe besteht eine Gesetzmässigkeit, die mathematisch ausgedrückt werden kann. Geben Sie zunächst für die beiden im Zusammenhang stehenden Funktionen Definitionen an. Beweisen Sie dann die Aussage über strukturelle Induktion.

## Aufgabe 4

Zur Berechnung des Mittelwertes einer Liste von ganzen Zahlen ist folgende Definition gegeben:

```
avg l = div (sum l) (length l).
```

Ermitteln Sie mithilfe des UNFOLD/FOLD-Verfahren eine optimierte Definition, die nur einen Listendurchlauf benötigt.

## Aufgabe 5

Sei $(M, \prec)$ eine wohlfundierte Menge und $P$ eine Eigenschaft für Elemente der Menge $M$.

Beweisen Sie die Aussage

$$(\forall b \in M.(\forall a \in M.a \prec b \to P(a)) \to P(b)) \longrightarrow \forall b \in M.P(b).$$

## Aufgabe 6

Zeigen Sie mithilfe wohlfundierter Induktion über `(n,xs)`, dass für alle $n \in \mathcal{N}$ und alle beliebigen Listen `xs` gilt

```
take n xs ++ drop n xs = xs.
```

Dabei gilt für die Funktionen `take` und `drop`:

`take n xs` ergibt längstmögliches Präfix der Länge $\leq n$ von `xs` und
`drop n xs` spaltet Präfix der Länge $n$ von `xs` ab.