Philipps Universität Marburg

Übungen zu „Konzepte von Programmiersprachen“, WS 2010/11

Prof. Dr. R. Loogen · Fachbereich Mathematik und Informatik · Hans-Meerwein-Straße, D-35032 Marburg

Nr. 7, Abgabe: Dienstag, 7. Dezember 2010 vor der Vorlesung

17. Funktionskomposition — 2 Punkte

(a) Schreiben Sie als Komposition von vordefinierten Funktionen eine Funktion `interleave :: [a] -> [a] -> [a]`, die zwei Listen abwechselnd elementweise zur Ergebnisliste mischt. Beispiel: `interleave "123" "abcde"  =>* "1a2b3c"`. / 1

(b) Definieren Sie Funktionen `f1` und `f2` als Komposition vordefinierter Funktionen, so dass die Reduktion des folgenden Ausdrucks das angegebene Ergebnis liefert: / 1

```
f2 2 ( f1 (^) [1,2,3,4] )  =>*  [2,4,8,16]
```

18. Rosenbäume — 6 Punkte

Implementieren Sie die folgenden Funktionen über Rosenbäumen und verwenden Sie dabei, falls möglich, die Faltungsfunktion `foldRose`.

```
data Rose a = Node a [Rose a]

foldRose               :: (a -> [b] -> b) -> Rose a -> b
foldRose f (Node x ts) =  f x (map (foldRose f) ts)
```

(a) `dfo :: Rose a -> [a]` traversiert einen Rosenbaum in Tiefensuche. / 1

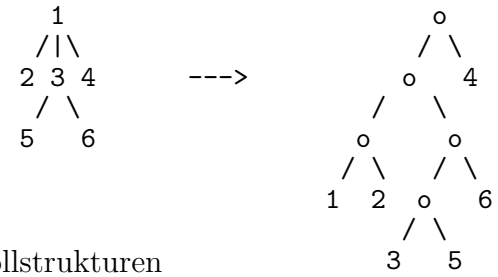
(b) `bfo :: Rose a -> [a]` traversiert einen Rosenbaum in Breitensuche. / 2

(c) `degree :: Rose a -> Int` ermittelt den maximalen Verzweigungsgrad eines Rosenbaums. Dabei handelt es sich um die größte Anzahl von Teilbäumen, die ein Knoten des Rosenbaums besitzt. Im Beispiel zu Teil (d) ist der maximale Verzweigungsgrad 3. / 1

(d) `toB :: Rose a -> BTree a` transformiert einen Rosenbaum in einen binären Baum vom Typ `BTree a`: / 2

```
data BTree a = Leaf a  | Fork (BTree a) (BTree a)
```

Dabei soll gemäß dem folgenden Beispiel vorgegangen werden:



19. Monadische Kontrollstrukturen — 4 Punkte

(a) Schreiben Sie eine Funktion / 2

```
while :: (a -> Bool) -> (a -> IO a) -> a -> IO a
```

.

`while p f` iteriert die durch `f` bestimmte Aktion, solange das Prädikat `p` für den Eingabewert bzw. den Rückgabewert der Aktion wahr ist.

(b) Schreiben Sie mit `while` aus (a) ein interaktives Programm `power2 :: IO ()`, das wiederholt ganze Zahlen `i` von der Standardeingabe liest und den Wert `2^i` auf die Standardausgabe zurückschreibt. Das Programm soll Eingabeaufforderungen und Ausgabeerläuterungen ausgeben und nach Eingabe einer Leerzeile stoppen. / 2