

Funktionale Programmierung

7. Übungsblatt

Prof. Dr. Margarita Esponda

Thema: Algebraische Datentypen

1. Aufgabe (10 Punkte)

Betrachten Sie folgende algebraische Datentyp-Definition (Baumstruktur ohne Inhalt aus der Vorlesung).

```
data SimpleBT = L | N SimpleBT SimpleBT deriving Show
```

a) Definieren Sie folgenden Funktionen damit:

```
isFull :: SimpleBT -> Bool      -- entscheidet, ob der Baum vollständig ist oder nicht.  
insertLeaves :: Integer -> SimpleBT -> SimpleBT -- eine eingegebene Anzahl von Blättern  
                                                    -- wird eingefügt.  
removeLeaves :: Integer -> SimpleBT -> SimpleBT -- eine eingegebene Anzahl von  
                                                    -- Blättern wird gelöscht.
```

Alle Funktionen sollen selber dafür sorgen, dass der Baum möglichst balanciert bleibt.

b) Schreiben Sie Testfunktionen mit kleinen Bäumen, so dass mit Hilfe der **printSimpleBT** Funktion (siehe Ressource → Material → SimpleBinaryTrees.hs.zip) das Ergebnis visualisiert werden kann.

2. Aufgabe (10 Punkte)

Erweitern Sie die Funktionen für den algebraischen Datentyp **BSearchTree** (Binäre Suchbäume) um folgende Funktionen:

```
postOrder :: (Ord a) => BSearchTree a -> [a]  
    -- siehe die Definition in den Vorlesungsfolien  
  
oneChild :: (Ord a) => BSearchTree a -> Bool  
    -- entscheidet, ob der Baum einen Knoten mit nur einem Kind hat, wobei  
    -- Nil nicht als Kind zählt.  
  
isFull :: (Ord a) => BSearchTree a -> Bool  
    -- überprüft, ob ein Baum vollständig ist.  
  
successor :: (Ord a) => a -> BSearchTree a -> Maybe a  
    -- berechnet den Nachfolger des eingegebenen Objektes  
  
mapTree :: (a -> b) -> BSearchTree a -> BSearchTree b
```

Anwendungsbeispiel:

```
mapTree (^2) (Node 5 (Node 3 Nil Nil) (Node 8 Nil (Node 4 Nil Nil))) =>  
    Node 25 (Node 9 Nil Nil) (Node 64 Nil (Node 16 Nil Nil))
```

foldTree :: b -> (a -> b -> b -> b) -> BSearchTree a -> b

Anwendungsbeispiel:

addMult a b c = a + b*c

foldTree 0 addMult (Node 5 (Node 3 Nil Nil) (Node 8 Nil (Node 4 Nil Nil))) => 29

5. Aufgabe (12 Punkte)

- Definieren Sie einen abstrakten Datentyp **Queue**, indem Sie eine Warteschlange mit den Operationen **enqueue** (Einfügen), **dequeue** (Entfernen), **isEmpty** und **makeQueue** (erzeugt eine leere Warteschlange) modellieren.
- Definieren Sie eine geeignete Funktion **showQueue**, die als Hilfsfunktion verwendet werden soll, um den **Queue**-Datentyp als Instanz der **Show**-Typklasse zu deklarieren.
- Definieren Sie die geeigneten Gleichheit und Vergleich-Infix-Operatoren für Ihren Queue Datentypen und deklarieren Sie damit den **Queue**-Datentyp als Instanz der **Eq**- und **Ord**-Typklassen.
- Schreiben Sie geeignete Testfunktionen für alle Funktionen Ihres **Queue**-Datentyps.

Damit die **dequeue** Operation auf die Verwendung der (++) Funktion verzichten kann, modellieren Sie Ihre Warteschlange mit Hilfe von zwei Listen. Elemente werden immer aus der ersten Liste entfernt und neue Elemente werden am Anfang der zweiten Liste eingefügt. Wenn die erste Liste leer ist und ein weiteres Element entfernt werden soll, wird die zweite Liste umgedreht und als erste Liste gesetzt.

6. Freiwillige Aufgabe (6 Bonuspunkte)

Definieren Sie einen abstrakten Datentypen, **ABaum**, der, anders als die Binärbäume, nicht nur zwei Kinder, sondern beliebig viele Kinder haben darf. Die Kinder können in Listen gespeichert werden.

Schreiben Sie für Ihren **ABaum**-Datentyp folgende Funktionen:

nodes -- zählt die gesamte Anzahl der inneren Knoten.
height -- rechnet die Höhe eines Baumes.
mapTree -- eine map-Funktion für den Baum.

Wichtige Hinweise:

- Verwenden Sie geeignete Namen für Ihre Variablen und Funktionsnamen, die den semantischen Inhalt der Variablen oder die Semantik der Funktionen wiedergeben.
- Verwenden Sie vorgegebene Funktionsnamen, falls diese angegeben werden.
- Kommentieren Sie Ihre Programme.
- Verwenden Sie geeignete lokale Funktionen und Hilfsfunktionen in Ihren Funktionsdefinitionen.
- Geben Sie für alle Funktionen die entsprechende Signatur an.
- Schreiben Sie getrennte Test-Funktionen für alle Aufgaben.
- Die Lösungen sollen elektronisch (nur Whiteboard-Upload) abgegeben werden. **Keine verspätete Abgabe per Email ist erlaubt.**