

Übungen zu Funktionaler Programmierung

Übungsblatt 10

Ausgabe: 22.12.2017, **Abgabe:** 12.1.2017 – 16:00 Uhr, **Block:** 5

Aufgabe 10.1 (2 Punkte) *Functor*

Schreiben Sie eine Instanz der Klasse Functor für den folgenden Datentyp für nichtleere binäre Bäume:

```
data STree a = BinS (STree a) a (STree a) | LeftS (STree a) a
              | RightS a (STree a) | LeafS a deriving Show
```

Beispiel: `fmap succ $ BinS (LeftS (LeafS 9) 2) 4 (RightS 7 (LeafS 3))`
 \leadsto `BinS (LeftS (LeafS 10) 3) 5 (RightS 8 (LeafS 4))`

Aufgabe 10.2 (3 Punkte) *do-Notation*

Gegeben sei folgende Listenkomprehension:

```
solutions :: [(Int , Int , Int )]
solutions = [ (x,y,z) | z <- [0..] , y <- [0..z^2] , x <- [0..z^2]
               , 2*x^3 + 5*y + 2 == z^2]
```

- Überführen Sie die Listenkomprehension in die do-Notation.
- Überführen Sie die do-Notation in monadische Operatoren und Funktionen (`>=>`, `>>`, `return`).

Aufgabe 10.3 (3 Punkte) *Plusmonade*

Implementieren Sie die Funktion `preorderM` als Verallgemeinerung der Funktion `preorderB`:

```
preorderM :: MonadPlus m => Bintree a -> m a.
```

Die Funktion `preorderM` soll sich bei einer Festlegung des Rückgabetyps auf eine Liste von Werten wie `preorderB` verhalten:

```
preorderM $ Fork 3 (leaf 4) (leaf 5) :: [Int] ~> [3,4,5]
```

Darüber hinaus soll `preorderM` aber beispielsweise auch für einen in `Maybe` eingebetteten Wert funktionieren und dann den Wert des Wurzelknotens zurückgeben:

```
preorderM $ Fork 3 (leaf 4) (leaf 5) :: Maybe Int ~> Just 3
```

Ist der Baum leer, gibt die Funktion für den Rückgabety `Maybe Int` den Fehlerwert `Nothing` zurück.

Für die folgenden Aufgaben lesen Sie bitte selbstständig den Abschnitt *Maybe- und Listenmonade* auf den Folien 189 und 190.

Aufgabe 10.4 (2 Punkte) *Maybe-Monade*

Gegeben sei die sichere Division `sdiv`.

```
sdiv :: Int -> Int -> Maybe Int
_ `sdiv` 0 = Nothing
x `sdiv` y = Just $ x `div` y
```

Definieren Sie die Funktion `f` vom Typ `Int -> Int -> Int -> Maybe Int`, welche die Gleichung

$$f(x, y, z) = \frac{\frac{18}{x}}{y} + \frac{6}{z}$$

erfüllt. Für die Divisionen muss die Funktion `sdiv` benutzt werden. Nutzen Sie nur die Monadeneigenschaft von `Maybe` und lösen Sie die Aufgabe in der `do`-Notation.

Beispiel: `f 2 3 2 ~> Just 5`

Aufgabe 10.5 (2 Punkte) *Listenmonade*

Gegeben sei die nichtdeterministische Transitionsfunktion `delta`, welche auf `graph1` basiert.

```
G _ delta = graph1
```

Definieren Sie die Funktion `g` vom Typ `Int -> Int -> [Int]`, welche die Gleichung

$$g(x, y) = \delta(x) * \delta(\delta(y))$$

erfüllt. Nutzen Sie nur die Monadeneigenschaft der Liste und lösen Sie die Aufgabe in der `do`-Notation.

Beispiel: `g 1 3 ~> [2,2,8,12,12,12,3,3,12,18,18,18]`

Frohes Fest und einen guten Rutsch ins neue Jahr