technische universität dortmund

FLDIT

Lehrstuhl 1
Logik in der Informatik

Prof. Dr. Peter Padawitz

Jos Kusiek (*jos.kusiek@tu-dortmund.de*) Wintersemester 2017/2018

Übungen zu Funktionaler Programmierung
Übungsblatt 6

Ausgabe: 24.11.2017, **Abgabe:** 1.12.2017 – 16:00 Uhr, **Block:** 3

Aufgabe 6.1 (3 Punkte) *Arithmetische Ausdrücke*

a) Stellen Sie den Ausdruck $2x^3 + 5y + 2$ und z^2 als Elemente vom Typ `Exp String` da.

b) Schreiben Sie die Listenkomprehension `solutions :: [(Int,Int,Int)]` um. Machen Sie sinvollen gebrauch von den beiden Ausdrücken und `exp2store`.

```
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
  ]
```

Hinweis: Importieren Sie das Modul `Expr`.

Aufgabe 6.2 (3 Punkte) *Boolesche Ausdrücke*
Schreiben Sie eine Funktion `bexp2store`, welche sich ähnlich wie `exp2store` verhält. Anstelle arithmetischer Ausdrücke sollen boolesche Ausdrücke vom Typ `BExp x` ausgewertet werden. Diese Funktion benötigt zwei Variablenbelegungen. Eine für boolesche Ausdrücke und die andere für arithmetische Ausdrücke.

Benutzen Sie folgende Typen:

```
type BStore x = x -> Bool
bexp2store :: BExp x -> Store x -> BStore x -> Bool
```

Hinweis: Importieren Sie das Modul `Expr`.

Aufgabe 6.3 (3 Punkte) *Typklassen*
Schreiben Sie eine Klasse für eine überladene Funktion `drop'`. Diese soll sich wie `drop` verhalten, aber nicht auf den Listentyp `[a]` beschränkt sein. Instanziieren Sie die Klasse für `[a]`, `Colist a` und `Stream a`.

Aufgabe 6.4 (3 Punkte) *Binäre Bäume*
Gegeben seien folgende Datentypen:

```
data Bintree a = Empty | Fork a (Bintree a) (Bintree a) deriving Show
data Edge = Links | Rechts deriving Show
type Node = [Edge]
```

Definieren Sie folgende Funktionen.

a) `value :: Node -> Bintree a -> Maybe a` – Gibt den Wert des Knoten zurück. Falls der Knoten nicht existiert, wird `Nothing` ausgegeben.

b) `search :: Eq a => a -> Bintree a -> Maybe Node` – Durchsucht den Baum nach dem angegebenen Wert. Falls Knoten mit dem Wert existieren, wird der Erste ausgegeben. Ansonsten wird `Nothing` zurückgegeben. Die Suchreihenfolge ist beliebig.