

Übungen zu Funktionaler Programmierung

Übungsblatt 7

Ausgabe: 25.11.2016, Abgabe: 2.12.2016

Aufgabe 7.1 (3 Punkte)

1. Definieren Sie den Typ `Person` mit den Attributen (Destruktoren) `name`, `familyName`, `age`. Benutzen Sie sinnvolle Typen für die Attribute.
2. Schreiben Sie die Funktionen `getX` und `setX` aus der Aufgabe 6.3 so um, dass ausschließlich Destruktoren verwendet werden.

Der Datentyp `Point` sei jetzt wie folgt definiert:

```
data Point = Point{ x :: Double, y :: Double }
```

Aufgabe 7.2 (3 Punkte)

1. Stellen Sie den Ausdruck $3x^2 + 2y + 1$ als Element vom Typ `Exp String` da.
2. Schreiben Sie die Listenkompensation `solutions :: [(Int,Int,Int)]` um. Machen Sie gebrauch von `exp2store`.

Aufgabe 7.3 (3 Punkte) Schreiben Sie eine Funktion `simplify :: Exp x -> Exp x`, welche arithmetische Ausdrücke vereinfacht. Dabei sollen folgende Gleichungen zur Vereinfachung genutzt werden:

$$\begin{array}{ll} 0 + x = x & x^0 = 1 \\ 0 * x = 0 & x^1 = x \\ 1 * x = x & \end{array}$$

Aufgabe 7.4 (3 Punkte) 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 -> BStore x -> Store x -> Bool
```