
11. Übung zur Vorlesung „Fortgeschrittene Programmierung“

Sommersemester 2020

zu lösen bis 8. Juli 2020

Aufgabe 11.1 (Monoton nicht-fallende Folgen)

Gegeben ist die Funktion

```
weakly_increasing :: Ord a => [a] -> Bool
weakly_increasing xs = and $ zipWith (<=) xs $ drop 1 xs
```

- Geben Sie die schrittweise Auswertung folgender Ausdrücke an:

```
weakly_increasing [], weakly_increasing [2] ,
weakly_increasing [3,2], weakly_increasing [3,4]
```
- Zeigen Sie, dass genau für alle monoton nicht-fallenden Folgen `xs` gilt

```
weakly_increasing xs == True
```

Für alle folgenden Aufgaben:Gegeben ist der Datentyp für binäre Bäume mit dem passenden `fold`

```
data Tree k = Leaf | Branch (Tree k) k (Tree k)
```

```
fold :: a -> (a -> k -> a -> a) -> Tree k -> a
fold leaf branch t = case t of
  Leaf -> leaf
  Branch l k r -> branch (fold leaf branch l) k (fold leaf branch r)
```

Aufgabe 11.2 (Rekursion über Bäume)Definieren Sie eine Funktion, die zu jedem Baum vom Typ `Tree k` die Menge aller in diesem Baum vorkommenden Positionen zurückgibt

- als Liste von `Bool`
- als Liste von Peano-Zahlen
- als Liste von `Int`

Aufgabe 11.3 (Suchbäume)

- Definieren Sie die Funktion

```
ist_suchbaum :: Ord k => Tree k -> Bool
```

die genau alle Bäume mit Suchbaum-Eigenschaft haben.

- Zeigen Sie, dass sich die Eigenschaft `ist_suchbaum` nicht direkt durch `fold` ausdrücken lässt, es also keine Funktionen `leaf` und `branch` gibt, so dass gilt:

```
ist_suchbaum = fold leaf branch
```

- Definieren Sie die Eigenschaft `ist_suchbaum` durch eine Verkettung aus `fold` und einer Projektion. (Autotool)

Aufgabe 11.4 (Operationen auf Suchbäumen)

a. Definieren Sie Funktionen

- **member**, welche feststellt, ob ein Schlüssel in einem binären Suchbaum vorkommt,
- **findmin**, welche den kleinsten Schlüssel in einem binären Suchbaum zurückgibt,
- **deletemin**, welche den Knoten mit dem kleinsten Schlüssel aus einem binären Suchbaum entfernt.

Überlegen Sie sich zuerst den Typ jeder Funktion.

b. Zeigen Sie durch strukturelle Induktion, dass Ihre Funktion **deletemin** die Suchbaum-Eigenschaft erhält.

Aufgabe 11.5 (Balancierte Bäume)

a. Definieren Sie die Eigenschaft

```
balanced :: Tree a -> Bool
```

die genau alle Bäume mit AVL-Balance-Eigenschaft haben.

b. Zeigen Sie, dass sich die Eigenschaft **balanced** nicht direkt durch **fold** ausdrücken lässt, es also keine Funktionen **leaf** und **branch** gibt, so dass gilt:

```
balanced = fold leaf branch
```

c. Definieren Sie die Eigenschaft **balanced** durch eine Verkettung aus **fold** und einer Projektion. (Autotool)