



Funktionale Programmierung/Höhere Programmiersprachen

Wintersemester 2022/2023

5. Übungsserie

Falls Sie eine Prüfungsvorleistung erbringen müssen, geben Sie Ihre Lösungen bitte **jeweils vor Beginn der ersten Übungseinheit (Donnerstag 15:30 Uhr)** beim Übungsleiter ab oder laden sie bis zu diesem Zeitpunkt unter dem Punkt Abgabe im OPAL hoch.

Aufgabe 1:

Geben Sie Haskell-Funktionen mit folgenden Funktionalitäten unter Verwendung der vordefinierten Funktionen `map`, `filter` und `fold[r,l]` an.

- a) `fac`-Funktion zur Fakultätsberechnung mittels `foldl` und Listengenerator.
- b) Summe der Quadrate der natürlichen Zahlen von 0 bis `n` mittels `map` und `foldr`.
- c) `length`-Funktion zur Bestimmung der Länge einer Liste mittels `foldl` oder `foldr`.
- d) `reverse`-Funktion zum Umkehren einer Liste mittels `foldl` oder `foldr`.
- e) Funktion `numOddEvenSquares` zur Ermittlung der Anzahl gerader und ungerader Quadratzahlen zwischen 0 und einem vorgegebenen Wert `n`.

Aufgabe 2:

Definieren Sie einen rekursiven algebraischen Datentyp `FloatExpr` zur Darstellung von arithmetischen Ausdrücken mit `Float`-Werten und den folgenden Eigenschaften und Funktionen.

Hinweis: Nutzen Sie den rekursiven algebraischen Datentyp `Expr` aus der Vorlesung als Vorlage.

- a) Ein arithmetischer Ausdruck ist entweder ein `Float`-Wert oder wird aus einer der folgenden Operationen mit arithmetischen Ausdrücken als Operanden gebildet: Addition, Multiplikation, Quadratwurzel, absoluter Betrag.
- b) Funktionen `numOps` und `numVals` zur Ermittlung der Anzahl an Operationen und `Float`-Werten in einem arithmetischen Ausdruck.
- c) Funktion `evalExpr` zur Berechnung des Werts eines arithmetischen Ausdrucks.
- d) Funktion `showExpr` zur Darstellung eines arithmetischen Ausdrucks als `String`.
Hinweis: Die Funktion `show :: a -> String` wandelt einen `Float`-Wert in einen `String` um.

Aufgabe 3:

Betrachtet werden Bäume deren innere Knoten beliebig viele Kindknoten enthalten können und deren äußere Knoten ein Element eines beliebigen (einheitlichen) Datentyps aufnehmen können. Definieren Sie einen entsprechenden Datentyp mit den folgenden Eigenschaften und Funktionen:

- a) Der Baum soll als polymorpher algebraischer Datentyp **GenTree** realisiert werden.
Hinweis: Verwenden Sie eine Liste um Kindknoten aufzunehmen.
- b) Funktion zur Ermittlung wie viele Elemente eines Baumes eine gegebene Prädikatsfunktion erfüllen.
- c) Funktion zur Rückgabe aller Elemente eines Baumes als Liste.
- d) Funktionen zur Ermittlung der Tiefe eines Baumes und der maximalen Anzahl an Kindknoten über alle Knoten eines Baumes.
- e) Funktionen zur Ermittlung ob ein Baum ein voller k -ärer Baum ist (d. h., jeder Knoten hat 0 oder k Kindknoten) und ob ein Baum ein perfekter k -ärer Baum ist (d. h., voller k -ärer Baum mit allen äußeren Knoten auf einer Ebene).