

Übungen zur Vorlesung Funktionale Programmierung

Blatt 1

Aufgabe P-1: Definieren Sie mittels expliziter Rekursion eine Funktion $nub :: [Int] \rightarrow [Int]$, die Duplikate aus einer Liste entfernt, z.B:

$$nub [7, 2, 5, 2, 3, 5] \implies [7, 2, 3, 5]$$

Erweiterung: Entfernen Sie Elemente immer vom Ende der Liste (siehe H-1).

Aufgabe P-2: Definieren Sie eine Funktion $halve :: [a] \rightarrow ([a], [a])$, die eine Liste in zwei annähernd gleich große Hälften auspaltet, z.B:

$$halve [1, 2, 3, 4, 5, 6] \implies ([1, 2, 3], [4, 5, 6])$$

Hinweis: Verwenden Sie $x \text{ 'div' } y$ für ganzzahlige Division. Definieren Sie Funktionen $take'$ und $drop'$, die die ersten n Elemente liefern bzw ignorieren, z.B:

$$\begin{aligned} take' 2 [1, 2, 3, 4, 5, 6] &\implies [1, 2] \\ drop' 2 [1, 2, 3, 4, 5, 6] &\implies [3, 4, 5, 6] \end{aligned}$$

Aufgabe P-3: Implementieren Sie den Mergesort Algorithmus. Verwenden Sie dazu die Funktion $halve$ und implementieren Sie eine Funktion $merge$, die zwei sortierte Listen zu einer sortierten Liste zusammenfügt, z.B:

$$merge [2, 4, 6][1, 3, 5] \implies [1, 2, 3, 4, 5, 6]$$

Aufgabe P-4: Ein Tripel (x, y, z) wird als *pythagoräisch* bezeichnet wenn folgende Gleichung erfüllt ist: $x^2 + y^2 = z^2$. Unter Verwendung einer List Comprehension, definieren Sie eine Funktion die alle pythagoräischen Tripel, mit Komponenten $\leq n$, findet, z.B:

$$pyths 10 \implies [(3, 4, 5), (4, 3, 5), (6, 8, 10), (8, 6, 10)]$$

Erweiterung: Verändern Sie die Funktion so, dass nur geordnete Tupel generiert werden.

Aufgabe P-5: Definieren Sie eine Funktion $prime :: Int \rightarrow Bool$, die testet, ob eine gegebene Zahl eine Primzahl ist.

Aufgabe H-1: Definieren sie die Funktionen *length* (Länge einer Liste) sowie *reverse* (Umkehren einer Liste), mithilfe der Funktionen *foldl* und *foldr*.

Aufgabe H-2: Definieren Sie als Datentyp Bäume mit beliebiger Anzahl von Nachfolgern, wobei Knoten Werte eines beliebigen, fixen Datentyps beinhalten. Definieren Sie auf diesem Datentyp eine Funktion *showTree*, die eine eindeutige Stringrepräsentation des Baums in Präfix Ordnung erstellt.