

Programmiersprachen

Wintersemester 2006/2007

7. Übungsblatt

21. Dezember 2006

Aufgabe 1:

- a) Schreibe eine Haskell-Funktion `approx :: (Ord a, Num a) => a -> [a] -> a`, die den Grenzwert einer monotonen und konvergenten Folge, die als eine möglicherweise unendliche Liste dargestellt ist, approximiert. `approximate eps s` liefert das erste Element der Folge `s` zurück, das den Grenzwert mit Präzision `eps` approximiert.

(Hinweis: Der Betrag einer reellen Zahl kann mit Hilfe der Funktion `abs :: Num a => a -> a` berechnet werden.)

- b) Schreibe eine Haskell-Funktion `buildSeq :: a -> (a -> a) -> [a]`, die eine unendliche Liste aufbaut. `buildSeq v0 recForm` liefert eine Folge $(x_i)_{i \in \mathbb{N}}$, die wie folgt definiert ist:

$$x_0 = v0 \text{ und } x_{n+1} = \text{recForm } x_n \text{ für } n \geq 0$$

- c) Schreibe eine Haskell-Funktion, um die Wurzel einer positiven Zahl a zu berechnen. \sqrt{a} kann durch ein Element der untenstehenden Folge approximiert werden:

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right) \text{ mit } n \geq 0 \text{ und } x_0 = 1$$

Aufgabe 2:

Die Fibonacci Folge ist wie folgt definiert:

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_k &= F_{k-1} + F_{k-2} \quad \forall k \in \mathbb{N}, k \geq 2 \end{aligned}$$

Definiere einen Haskell-Wert, der die (unendliche) Fibonacci-Folge repräsentiert!

Aufgabe 3:

Schreibe eine Haskell-Funktion, die die Liste aller Paare von Nachbarelementen in einer Liste liefert. (Hinweis: Man kann die vordefinierte Haskell-Funktion `zip :: [a] -> [b] -> [(a,b)]`, die aus zwei Listen, die Liste der Paare der entsprechenden Elemente konstruiert, verwenden.)

Aufgabe 4:

Schreibe eine Haskell-Funktion die das Skalar-Produkt zweier mathematischen Vektoren, die als Listen repräsentiert sind, berechnet. (Hinweis: Man kann die vordefinierten Haskell-Funktionen `zip :: [a] -> [b] -> [(a,b)]` und `sum :: Num a => [a] -> a`, die die Summe aller Elemente einer Liste liefert, verwenden.)

Aufgabe 5:

Eine perfekte Zahl ist eine natürliche Zahl n , die gleich die Summe ihrer Faktoren ausschließlich n ist (z.B, $6 = 1 + 2 + 3$). Schreibe eine Haskell-Funktion, die die Liste aller perfekten Zahlen kleiner als eine gegebene Zahl liefert.

Aufgabe 6:

Schreibe eine Haskell-Funktion, die die Liste aller Permutationen der Elemente einer Menge, die als eine Liste repräsentiert ist, berechnet.