

Programmierung

01. Übungsblatt

Zeitraum: 11. – 15. April 2016

Übung 1

- Schreiben Sie in Haskell eine Funktion `fac :: Int -> Int`, so dass `fac n` die Fakultät $n!$ berechnet, also $\prod_{i=1}^n i$.
- Schreiben Sie nun eine Funktion `sumFacs :: Int -> Int -> Int`, so dass `sumFacs n m` den Wert $\sum_{i=n}^m i!$ berechnet.

Übung 2

Die Folge der Fibonacci-Zahlen f_0, f_1, \dots , ist definiert durch $f_0 = 1, f_1 = 1$, und $f_{i+2} = f_i + f_{i+1}$ für jedes $i \in \mathbb{N}$. Implementieren Sie eine Haskell-Funktion, die für die Eingabe i die Zahl f_i berechnet.

Übung 3

Sei $n \in \mathbb{N}$. Die Collatz-Folge mit Startzahl n , $C(n)$, ist die unendliche Folge von natürlichen Zahlen m_1, m_2, m_3, \dots , so dass $m_1 = n$, und für jedes $i \geq 1$,

- falls m_i gerade ist, dann ist $m_{i+1} = m_i/2$,
- ansonsten ist $m_{i+1} = 3m_i + 1$.

Es wird vermutet, dass für jede Startzahl n die dazugehörige Collatz-Folge $C(n)$ den Wert 1 beinhaltet, also ein $i \geq 1$ existiert, so dass $m_i = 1$. Das kleinste solche i nennen wir dann $k(n)$. Implementieren Sie jeweils eine Haskell-Funktion, welche gegeben eine Startzahl n

- den Wert $k(n)$ berechnet;
- den größten Wert $k(i)$, für $1 \leq i \leq n$, berechnet.

Existiert eine Folge $C(n)$, die den Wert 1 nicht beinhaltet, dann ist das Verhalten der obigen Funktionen mit Eingabe n undefiniert.

Zusatzaufgabe: Implementieren Sie eine Funktion, die für die Eingabe n die Liste der Glieder $m_1, \dots, m_{k(n)}$ der Folge $C(n)$ berechnet.

Übung 4

Implementieren Sie eine Haskell-Funktion, welche für $n \in \mathbb{N}$ die Anzahl der Binärbäume mit Knotenzahl n berechnet. Dabei soll jeder Knoten entweder zwei oder keinen Nachfolger haben.

Zusatzaufgabe 1

Machen Sie sich mit `ghc(i)` (dem Glasgow Haskell Compiler, <https://www.haskell.org/ghc>) vertraut, insbesondere mit den Befehlen `:type`, `:info`, `:browse` und `?:`, und mit der Suchmaschine <https://haskell.org/hoogle>.