HTWK Leipzig, Fakultät IM

Prof. Dr. Sibylle Schwarz `sibylle.schwarz@htwk-leipzig.de`

4.Übung zur Vorlesung „Fortgeschrittene Programmierung“
Sommersemester 2020 zu lösen bis 6. Mai 2020

Aufgabe 4.1 (Rekursion)

Definieren und implementieren Sie die folgenden rekursiven Funktionen in Haskell sowie sinnvolle Testfälle dafür:

a. Fibonacci-Zahlen: $\text{fib}(n) = \begin{cases} 0 & \text{falls } n = 0 \\ 1 & \text{falls } n = 1 \\ \text{fib}(n-2) + \text{fib}(n-1) & \text{sonst} \end{cases}$

b. die Funktion $\text{sb}(n) = \begin{cases} 0 & \text{falls } n = 0 \\ 1 & \text{falls } n = 1 \\ \text{sb}(m) & \text{falls } \exists n \in \mathbb{N} : n = 2m \\ \text{sb}(m) + \text{sb}(m+1) & \text{falls } \exists n \in \mathbb{N} : n = 2m+1 \end{cases}$

c. Funktion `toPeano :: Integer -> N` zur Umrechnung von der üblichen Dezimaldarstellung natürlicher Zahlen in Peano-Zahlen

Aufgabe 4.2 (Operationen mit Peano-Zahlen)

Definieren Sie die folgenden Funktionen auf Peano-Zahlen in Haskell

a. Multiplikation `mult`

b. Potenzieren `pot`

Aufgabe 4.3 (Strukturelle Induktion)

Hier werden die Funktionen `add` von den Folien sowie `mult` und `pot` aus der vorigen Aufgabe verwendet.

a. Bestimmen Sie durch schrittweise Reduktion:

   (a) `add (S (S Zero)) (S ( S Zero))`

   (b) `mult (S Zero) (S ( S Zero))`

b. Zeigen Sie durch strukturelle Induktion, dass

   (a) für jede Peanozahl x gilt: `add Zero x = x`

   (b) die Addition der Peanozahlen assoziativ ist,

   (c) die Addition der Peanozahlen kommutativ ist,

   (d) für jede Peanozahl x gilt: `mult x (S Zero) = x`

   (e) für jede Peanozahl x gilt: `mult (S Zero) x = x`

   (f) für jede Peanozahl x gilt: `pot (S Zero) x = S Zero`