
6. Übung zur Vorlesung „Fortgeschrittene Programmierung“

Sommersemester 2020

zu lösen bis 27. Mai 2020

Aufgabe 6.1:

Überlegen Sie (ohne Computer), welche Funktionen über Listen hier mit Hilfe der in der Vorlesung vorgestellten Funktionen höherer Ordnung definiert werden:

- a. `map (+2)`
- b. `foldr (||) False`
- c. `filter (\ x -> (mod x 2 == 0))`
- d. `foldr (\ x y -> y || (mod x 2 == 0)) False`
- e. `foldr (+) 0 . map (\ x -> x * x)`
- f. `foldr (+) 0 . map (\ x -> x * x) . filter (\ x -> (mod x 2 == 0))`

Überlegen Sie sich zunächst, welchen Typ die Funktionen haben und bestimmen Sie die Funktionswerte für einige geeignet gewählte Argumente.

Aufgabe 6.2:

Berechnen Sie den Wert der folgenden Ausdrücke durch schrittweise Reduktion. Geben Sie dazu die Definitionen der Funktionen und in jedem Reduktionsschritt die verwendete Regel an.

- a. `foldl (+) 5 $ take 3 [1 ..]`
- b. `sum $ map (\ n -> n * n) [1, 2, 3]`

Aufgabe 6.3:

- a. Definieren Sie das Skalarprodukt zweier Vektoren mit Hilfe geeigneter Funktionen höherer Ordnung.
- b. Verwenden Sie die `zipWith` zur Definition einer Funktion, die feststellt, ob von zwei eingegebenen Listen eine Präfix der anderen ist.

Geben Sie dazu jeweils geeignete LeanCheck-Tests an.

Aufgabe 6.4:

Definieren Sie die folgenden Funktionen auf Listen mit Hilfe der Funktion `foldr` und geben Sie jeweils geeignete LeanCheck-Tests an:

- a. `append`
- b. `reverse`
- c. `map`
- d. `filter`