

Funktionale Programmierung
WiSe 21/22
Tutor: Florian Alex
Studenten: Vasile Popa, Simeon Vasilev, Tomas Proano

Aufgabe 1

-- a)

mersenne :: Integer -> [Integer]

mersenne 1 = [0]

mersenne x = (2^{x-1} - 1) : mersenne (x-1)

✓

-- b)

-- Die Funktion ist rekursiv, da wir einmal einen allgemeinen Fall haben,

-- und dann diesen aufrufen, wenn wir irgendwelche andere Parameter übergeben

Wieso macht das die Funktion rekursiv?

2/3

Aufgabe 2

a) foo (foz 4 3 2) (2⁵ - 1)

Call by value:

foo (foz 4 3 2) (2⁵ - 1)

-> foo (2*4 + 3) (32 - 1)

-> foo (8 + 3) (31)

-> foo (11) (31)

-> baz (11)

-> 34

✓

+2

Lazy evaluation:

foo (foz 4 3 2) (2⁵ - 1)

-> foo (11) (2⁵ - 1)

-> baz (11)

-> 34

Ok...

✓

+2

b) die Funktion **foo** ist im Parameter **a** strikt.

die Funktion **bar** ist ~~immer~~ strikt.

baz ist auch immer strikt

und **foz** ist in den Parametern **a** und **b** strikt.

+3

7/8

Aufgabe 3

Wandeln Sie die folgenden Ausdrücke in Normalform und schwache Normalform um.

(a) (3+7):[4+7, 5+7, 6+7] = [10,11,12,13] ✓

(b) "Guten" ++ "Tag" = "GutenTag" ✓

(c) ("Guten" ++ "Tag", 4 + 8, 4:(3:[])) = ("GutenTag",12,[4,3]) ✓

(d) [if x==y then 'o' else '.' | x <- [1..5], y <- [1..7], (x+y)<9] = "o.....o.....o.....o..." ✓

Ich nehme mal an, dass es die Normalform sein soll.
Wie sieht die schwache Normalform aus?

4/8

Aufgabe 4

SelectionSortList = [8,5,2,3] Find the minimum element in arr[0...4] and place it at the beginning

SelectionSortList = [2,8,5,3] Find the minimum element in arr[1...4] and place it at the beginning

SelectionSortList = [2,3,8,5] Find the minimum element in arr[2...4] and place it at the beginning

SelectionSortList = [2,3,8,5] Find the minimum element in arr[3...4] and place it at the beginning

SelectionSortList = [2,3,5,8]

0k...

+2

InsertionSortList = [8,5,2,3]

3?
↓

Let us loop for $i = 1$ (second element of the array) to 4 (last element of the array)

$i = 1$. Since 5 is smaller than 8, move 8 and insert 5 before 8

InsertionSortList = [5,8,2,3]

$i = 2$. Since 2 is smaller than 8, move 8 and insert 2 before 8

InsertionSortList = [5,2,8,3]

$i = 3$. Since 3 is smaller than 8, move 8 and insert 3 before 8

InsertionSortList = [5,2,3,8] $i = 4$. We repeat the process with 5

InsertionSortList = [2,3,5,8]

Ich erkenne nicht Insertionsort. Welche Elemente werden noch hinzugefügt und welche sind es schon?

Mergesort

MergeSortList = [8,5,2,3]

← das ist die falsche Liste

[8] [5] [2] [3] - 1) We divide the list in separate elements

hier fehlt ein Zwischenschritt

[5, 8] [2, 3] - We merge the elements by two comparing which one is bigger

We choose the first element of the first list (A) and compare it with the first elements from other list (B).

If A is bigger than B, it comes first in the new list, if not, then comes B.

[5,8] [3]

[2, ., ., .]

Then we do the same by comparing 5 with the next element

[2, 3, ., .]

Then there are no other elements in the second list, we simply merge two lists

[2,3,5,8]

+

QuickSort

QuickSortList = {8,5,2,3}  falsche Liste

Pivot is 3. We take all the elements and place them to the left or the right of 3, depending if it's bigger or smaller. We

[2,3] [.,.]

[2,3] [5,8]

We do the same process for the sublists and then merge the lists

[2,3,5,8]

+

Es sind leider nicht alle Zwischenschritte zu sehen

4/8

Aufgabe 5

`sortListe :: (Ord a) => [a] -> Bool`

`sortListe [] = True`

`sortListe [x] = True`

`sortListe (x:y:xs) = x <= y && sortListe (y:xs)` ✓

3/3

20/30