

Übung: Rekursion 1

- Primitive Rekursion
- Endrekursion

1 Aufgabe

Zur Erinnerung, das Schema der primitiven Rekursion:

```
1 prim c g 0 x = c x
2 prim c g n x = g (f (n-1) x) (n-1) x
3   where
4     f = prim c g
```

Verwenden Sie das Schema der primitiven Rekursion zum Implementieren der folgenden Funktionen (Parameter, die nicht benötigt werden, sollen vom Typ () gewählt werden):

- (a) $m2(n) = 2n$ (`m2 :: Integer -> () -> Integer`)
- (b) $e2(n) = 2^n$
- (c) $exp(x, n) = x^n$
- (d) $fact(n) = n!$

2 Aufgabe

Welche der folgenden Funktionen liegen in einer endrekursiven Form vor?

```
1 f g x
2   | x == 0 = g x
3   | otherwise = g $ f g (x -1)
4
5 length xs = case xs of
6   [] -> 0
7   x : xs -> (+1) $ length xs
8
9 length ' ls = aux
10  $ map (const 1)
11  $ ls
12  where
13    aux ys = case ys of
14      [] -> 0
15      [x] -> x
```

```
16 x : xs -> aux $ map (\ y -> (+1) x) xs
```

3 Aufgabe

Gegeben ist die Funktion

```
1 sieve :: (a -> a -> Bool) -> [a] -> [a]
2 sieve pred xs = case xs of
3   [] -> []
4   x : xs -> x:(sieve pred $ filter (pred x) xs)
```

- (a) Implementieren Sie die Funktion in einer endrekursiven Form, einmal mithilfe eines Akkumulators und einmal mithilfe von Continuations.
- (b) Benutzen Sie die endrekursiven Funktionen um die Funktion zu realisieren, die zu gegebenem Int n eine Liste mit allen Primzahlen bis (inklusive) n zurückgibt.

4 Aufgabe

Gegeben Sei:

```
1 data Tree a = Tree a [Tree a]
2
3 depth :: Tree a -> Integer
4 depth (Tree _ subtrees) =
5   1 + (maximum $ 0:(map depth subtrees))
```

Implementieren Sie die Funktion `depth` in endrekursiver Form.

Hinweis: Verwenden Sie als Akkumulator eine Liste von Bäumen.

5 Aufgabe

Die Fibonacci Funktion ist definiert als die Summe der zwei vorherigen Glieder (und Anfangswerte). Die "Tribonacci"-Funktion ist als Summe der letzten drei Glieder definiert:

$$\begin{aligned} \text{trib}(0) &= 1 \\ \text{trib}(1) &= 1 \\ \text{trib}(2) &= 1 \\ \text{trib}(n) &= \text{trib}(n-3) + \text{trib}(n-2) + \text{trib}(n-1) \end{aligned}$$

Implementieren Sie die Tribonacci-Funktion endrekursiv.