

Lernziele

- Fixpunkte, Endrekursion

Aufgabe 1

Welche der folgenden Funktionen sliegen in einer endrekursiven Form dar?

```
let rec myWhile x condition update effect =
  if condition x then
    effect x
    myWhile (update x) condition update effect

let rec f g x =
  if x = 0 then g x
  else f g (x-1) |> g

let rec h x =
  if x < 2 then (1,1)
  else
    let y = h (x-1)
    let z = h (x-2)
    (1 + fst y, 2 + snd z)

let add1 = (+) 1

let rec length = function
  | [] -> 0
  | x::xs -> add1 <| length xs

let length' ls =
  let rec aux = function
    | [] -> 0
    | [x] -> x
    | x::xs -> aux <| List.map (fun y -> add1 x) xs
  in
  ls
  |> List.map (fun x -> 1)
  |> aux
```

Aufgabe 2

Gegeben ist die Funktion

```
let rec sieve pred = function
| [] -> []
| x::xs -> x::(sieve pred (List.filter (pred x) xs))
```

Implementieren Sie die Funktion in einer endrekursiven Form.

- (a) Mithilfe eines Akkumulators.
- (b) Mithilfe von Continuations.
- (c) Benutzen Sie die endrekursiven Funktionen um die Funktion zu realisieren, die zu gegebenem `int n` eine Liste mit allen Primzahlen bis (inklusive) `n` zurückgibt.

Aufgabe 3

Implementieren Sie ein Funktional `primesF`, so dass `fix primesF` eine Funktion ergibt, die zu gegebenem `int n` eine Liste mit allen Primzahlen bis (inklusive) `n` zurückgibt.