

let $\text{minmax } l =$

let $f \ x \ (\text{min}, \text{max}) =$

if $x < \text{min}$ then (x, max)

else if $x > \text{max}$ then (min, x)

else (min, max)

in match l with

[3]

$x :: xs \rightarrow \text{foldr } f \ (x, x) \ xs$

$x = 3$

$xs = []$

foldr f

$(3, 3)$

$[]$

$=$

$(3, 3)$

dete una lista di interi: eliminare i valori < 0
e invertire l'ordine di tutti gli altri

es:

$$f[-2; 3; 4; -5; -7; 0] = [0; 4; 3]$$

- ricorsivamente
- ricorsivamente con accumulatore
- foldr

let f l =

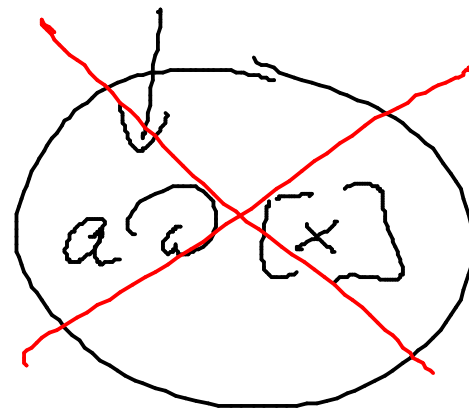
let fa l a = match l with

[] → a

| x::xs → if x < 0 then fa xs a
else fa xs (x::a)

in

fa l [] ;;



let f l =

let geo x = x >= 0

in reverse (filter geo l) []

let rec f l = match l with

[] → []

| x :: xs → if x < 0 then f xs

else (f xs) @ [x] ;;

(f xs) :: x NO!

let f l =

let g x y = if x < 0 then y
else y @ [x]

in foldr g [] l ji

$f : \text{'a list} \rightarrow \text{'a list}$

f cancella da una lista le occorrenze ripetute e consecutive di ciascun elemento

$$f [\underline{1; 1}; 3; 4; 4; 1; 2; 2] =$$

$$[1; 3; 4; 1; 2]$$

$$f [\underline{1; 1; 1}; 3] = [1; 3]$$

$$f [1; 3; 1] = [1; 3; 1]$$

let rec f l = match l with

[] → []

| [x] → [x]

| x::y::ys → if x < y then x::(f (y::ys))
else f (y::ys)::
y::(f ys)

let f $l =$ let g x $y =$

in

$l = \dots$ $x :: \underline{\underline{x\ s}}$
 $y = f\ x\ s$

$l = \dots$ $1 :: \underbrace{(1 :: 1 :: [])}_{[1]}$
 \uparrow
 x

let f l = let g x y = match y with

z::zs → if x <> z then x::y
else y

in foldr g [] l

f [3]

= foldr g [] [3]

= g 3 (foldr g [] []) = g 3 [] = ?

let f l = let g x y = match y with
[] -> [x]

| z::zs when x = z -> y

| z::zs when x <> z -> x::y

in

foldr g [] l []