

foldr f a [x₁; x₂; ... x_n] =

f x₁ (f x₂ ... (f x_n a) ...)



elementi di
liste



risultato sulle liste
che segue

let rec foldr f a l =
 match l with

 [] -> a

 |x::xs -> f x (foldr f a xs);;

foldr : ('a -> 'b -> 'b) -> 'b -> 'a list -> 'b

let sum l =

let f x y = x + y

in foldr f 0 l;;

sum : int list -> int = (sum)

filter

let filter p l =

let f x y = if p x then x :: y
 else y

in foldr f [] l;;

filter : ('a -> bool) -> 'a list -> 'a list

map

let map f l =

let g x y = f x :: y

in foldr g [] l;;

map : ('a → 'b) → 'a list → 'b list = <fun>

let forall p l =

let f x y = if p x then y
 else false

in foldr f true l ;;

forall : ('a → bool) → 'a list → bool

→ if y then p x
 else false

let exists p l =

let f x y = if p x then true
 else y

in foldr f false l ;;

exists : ('a → bool) → 'a list → bool

date una lista di interi dividerle in due liste $(l1, l2)$ tale che $l1$ contiene tutti gli element < 0 e $l2$ quelli ≥ 0

split $[-2; 3; 4; -1; 5] = ([-2; -1], [3; 4; 5])$

let rec split l =
 match l with

$[] \rightarrow ([], [])$

$| x :: xs \rightarrow \text{let } (l1, l2) = \text{split } xs$

in if $x < 0$ then $(x :: l1, l2)$

else $(l1, x :: l2);;$

split : int list \rightarrow int list * int list

$\#[(3, 5)];;$

$-: (\text{int} * \text{int}) \text{ list} = [\dots]$

let r = split xs
 ↑
 copia