

ANCORA SU FOLDER

let foo l = let f x y =

and a =

in foldr f a l

1) a deve essere il risultato di foo quando l = []

2) Come determinare f x y

$l = l1 @ (x :: xs)$

$y = foo \dots xs$

f deve AGGIUNGERE
il contributo di x
al risultato

FUNZIONE MEMBER : member x l

deve restituire true se x occorre in l
false altrimenti

let rec member x l = match l with

$[]$ \rightarrow false |

$y :: ys$ when $x = y$ \rightarrow true |

$y :: ys$ when $x \neq y$ \rightarrow member x ys

member : 'a \rightarrow 'a list \rightarrow bool = <fun>

member 3 [2; 3; 5]

= { 3^o pattern $x=3$ $y=2$ $ys=[3; 5]$ }

member 3 [3; 5]

= { 2^o pattern $x=3$ $y=3$ $ys=[5]$ }

true

let member1 x l =

let f z y = y or (x = z)

and a = false

in

foldr f a l

f z y = if y then y else x = z

y or (x = z)

l = l1 @ (z :: zs)

y = member x zs

member 1 3 [2; 3; 5]

= { def di member 1 }

foldr f false [2; 3; 5]

= { 2° patt. di foldr }

(foldr f false [3; 5]) or (3=2)

= { 2° pattern di foldr }

(foldr f false [5]) or (3=3) or (3=2)

= { 1° pattern di foldr }

(foldr f false []) or (3=5) or (3=3) or (3=2)

= { 1° pattern di foldr }

false or (3=5) or (3=3) or (3=2) = true

f z y = y or (3=z)

let rec foldr f a l =

match l with

[] → a |

x::xs → f x (foldr f a xs)

Esempio di uso di filter

Togliere da una lista di interi tutti i multipli di un dato intero z

```
let rec toglim z l =  
  match l with  
  [] → [] |  
  x::XS when x mod z = 0  
    → toglim z XS |  
  x::XS when x mod z <> 0  
    → x::(toglim z XS);;
```

```
let toglim z l =  
  let pr x = (x mod z <> 0)  
  in filter pr l;;
```

Calcolare il massimo di una lista e il numero di occorrenze del massimo

$$\text{maxocc } [10; 3; 2; 10; -1; 10; 7] = (10, 3)$$

$$\text{maxocc } [10; 3; 2] = (10, 1)$$

let maxocc l = let f x (m, n) = if x > m then (x, 1)
else if x = m then (x, n+1)
else (m, n)

[in match l with
y :: ys → (foldr f (y, 1) ys) ; ;

Calcolare il "primo" e il "secondo" massimo in una lista, supponendo che la lista contenga elementi tutti distinti fra loro.

$$\text{duemax } [10; 3; 8; 45] = (45, 10)$$

let $\text{duemax } l = \text{let } f \ x \ (m_1, m_2) =$
if $x > m_1$ then (x, m_1)
else if $x > m_2$ then (m_1, x)
else (m_1, m_2)

in match l with

$y :: z :: zs \rightarrow \text{foldr } f \ (\underbrace{\text{if } y > z \text{ then } (y, z) \text{ else } (z, y)}_{\text{foldr } f @ l}) \ zs$