

let rec filter p l =
 match l with

[] → []

| x :: xs when p x
 → x :: filter p xs

| x :: xs when not (p x)

→ filter p xs ::
 filter : ('a → bool) → 'a list → 'a list
 = (fun)

let $gt0\ x = x > 0$;;
 $gt0: int \rightarrow bool = \langle fun \rangle$

$filter\ gt0\ [-2; 3; -1; 4]$;;
 $= \{ \text{def. filter, def gt0} \}$
 $3^{\text{rd}}\ \text{part.}$

$filter\ gt0\ [3; -1; 4]$
 $= \{ \text{def filter, def gt0} \}$
 $2^{\text{nd}}\ \text{p}$

$3 :: filter\ gt0\ [-1; 4]$

$$= \{ \text{def } \text{filter} . g^{\text{to}} \}_{3^{\circ} p}$$

$$3 :: \text{filter } g^{\text{to}} [4]$$

$$= \{ \dots \}$$

$$3 :: 4 :: \text{filter } g^{\text{to}} []$$

$$= \{ \dots \}^{1^{\circ} p}$$

$$3 :: (4 :: [])$$

$$= \{ \text{noterm } \}$$

$$[3; 4]$$

map

let rec map f l =
 match l with

[] → []

| x :: xs → f x :: map f xs;;

map: $(\underbrace{'a \rightarrow 'b}_f) \rightarrow \underbrace{'a \text{ list}}_l \rightarrow \underbrace{'b \text{ list}}_{ms}$

let inc $x = x + 1$;
inc : int \rightarrow int = (fun)

map inc [1; 2; 3]
= [2; 3; 4]

map gt0 [-2; 3; 4; -1]

=

[false, true, true, false]

folder

folder f a $[x_1; x_2; \dots; x_n]$

= {def intuitiva

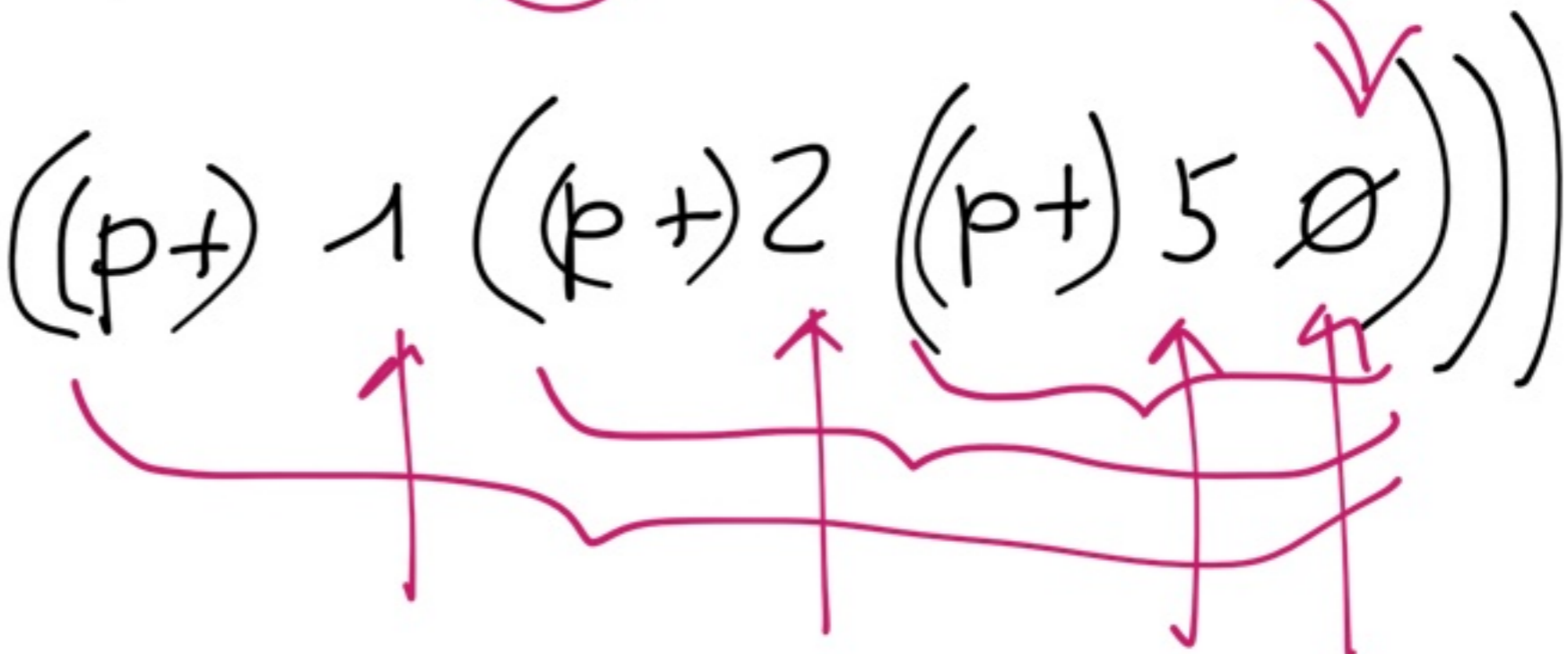
$$f(x_1, f(x_2, \dots, f(x_n, a) \dots))$$

↓

folder f a $[\] = a$

(mefx +)

foldr (p ref x +) \emptyset [1; 2; 5]



8

let rec foldbr f a l =
 match l with

[] → a

| x :: xs → f x (foldbr f a xs)

foldbr ('a → 'b → 'b) → 'b → 'a list
 f
 'b
 xs

Conta gli element d: una
liste attraverso le foldr

f
elementi
corrente
della
liste

risultato
della foldr
migli elementi
nummeranti
a b e il

loro numero

$$f \times m = m + 1$$

↓ numero
degli element
di X

$X :: X$

let conte $l =$

let $f \times m = m + 1$

m

folobr $f \neq l ; ;$

conte: 'a list \rightarrow mt

conta $[-2; -2; 7]$

$$= \{ \text{let } f \times m = m+1 \}$$

folabr $f \emptyset [-2; -2; 7]$

$$= \{ \text{def folabr, } 2^{\circ} p \}$$

$f \ 2 \ (\text{folabr } f \emptyset [-2; 7])$

$$\Rightarrow \{ \text{def folabr, } 2^{\circ} p \}$$

$f \ 2 \ (\ f \ (-2) \ (\text{folabr } f \emptyset [7]))$

$$\begin{aligned} &= \text{def folder, } 2^{\circ} p \\ & f 2 (f (-2) (f 7 (\text{folder } f \text{ or}))) \end{aligned}$$

$$= \text{def folder } 1^{\circ} p$$

$$f 2 (f (-2) (f 7 \emptyset))$$

$$f 2 (f (-2) 1)$$

$$f 2 \quad 2 = \textcircled{3}$$

Definizione di filter
mediante foldr

let filter p l =

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

in foldr f [] l ;;

filter gto $[-1, 2; -3]$
 $= \{ \text{def. filter, } f \times y = f(p \times) \text{ then } x::y \}$

foldr $f \ [] \ [-1; 2; -3]$

$= \{ \text{def foldr, } 2^{\circ} p \}$

$f(-1)$ (foldr $f \ [] \ [2; -3]$)

$= \{ \text{def foldr, } 2^{\circ} p \}$

$f(-1)$ ($f \ 2$ (foldr $f \ [] \ [-3]$)))

$= \{ \text{def foldr, } 2^{\circ} p \}$

$$f(-1)(f_2(f(-3)(\text{folch } f \uparrow \uparrow)))$$

$$= \{ \text{def folch. } \frac{1^0}{f} \}$$

$$f(-1)(f_2(f(-3) \square))$$

$$= f(-1)(f_2 \square)$$

$$= f(-1) \square_2$$

$$= \square_2$$

let map f l =

let g x y = f x :: y

in foldr g [] l j;

risultato delle

map f su tutti

gli elementi che riguardano
x

Date una lista dare
 come risultato la
 coppia (n, s) dove n
 \bar{i} è il numero degli
 element della lista i
 s è la loro somma

- Ricorsivamente
- Usando foldr

let rec msum e =
 match e with

[] → (0, 0)

| x :: xs →

let (m, s) = msum xs

m (m + 1, s + x) ;

msum : nat list → int * nat

let ft $(x, y) = x \text{ ;}$
 ft: 'a * 'b \rightarrow 'a = <fun>

let sd $(x, y) = y \text{ ;}$

sd: 'a * 'b \rightarrow 'b = <fun>

let rec msum l = match l with
 [] \rightarrow (\emptyset , \emptyset)

| x :: xs \rightarrow let r = msum xs

in (1 + ft r, x + sd r) ; ;

msum: int list \rightarrow int * int
 = <fun>

let $msum\ l =$

let $f\ x\ (y, z) =$
 $(y + 1, x + z)$

in $foldr\ f\ (0, 0)\ l;$

$msum : list \rightarrow int * int$
 $= \langle sum \rangle$

$$f(x, (y, z)) = x + y + z; i;$$

$$f: \underbrace{m \times n}_{x} \rightarrow \underbrace{m \times m}_{\text{tipo } (y, z)} \rightarrow m$$

let forall p l =

let f x y = p x & y

→ if y then p x
else y

in foldr f true l;

forall: ('a → bool) → 'a list → bool

= (fmap)

let exists $p \ell =$

let $f \times y = p \times$ or y

\rightarrow if y then y else $p \times$

in folder of false ℓ ; ;

exists: $(a \rightarrow \text{bool}) \rightarrow a \text{ list} \rightarrow \text{bool}$
 $= \langle \text{fun} \rangle$

Portare l'ultimo elemento (l e l')
di una lista intera

let lastfirst $l =$

let f x $y =$ match y with
 $[\] \rightarrow [x]$

| $w :: xs \rightarrow w :: x :: xs$

in foldr f $[\]$ l ;;