

let $f \ g \ x =$

$(g \ x) + 1 ; ;$

$f :: ('a \rightarrow int) \rightarrow 'a$

$\rightarrow int = (f \ g)$

$f \ 5 \ true ; ;$

$bool \rightarrow int$

$bool$

results
be type
 int

$- : int = 1$

$(5 \ true) + 1$

let $f \circ g \ x =$

$(g(x) + 1) ; ;$

$f: (a \rightarrow int) \rightarrow a$

$\rightarrow int = (f \circ g)$

$a = int$

$f \ t \ (-3) ; ;$

$f \ t \ (-3)$
calculator

$int \rightarrow int$

$(t(-3) + 1)$
 $1 + 1$

$\therefore int = 2$

dichiarazioni locali.

let associazione in
espressione i)

associare un nome
a un valore
(come nelle expr.
let)

let $x = 3$ in $x + 1$; ;

- l'associazione per x vale solamente durante la valutazione di questa espressione
- il valore dell'espressione dato da questo $expr$

let $x = 3$ in $(x+1)$

$\therefore \text{mt} = 4$

let f $x = \text{let } g$ $y = y + 2$

$\text{in } (g\ x) + 1$

def. f

let $f x = \text{let } g y = \underline{y+2} \text{ in } (g x) + 1$

$f : \underbrace{\text{int}}_{\text{tipo } x} \longrightarrow \underbrace{\text{int}}_{\text{tipo } ns} = \langle \text{fun} \rangle$

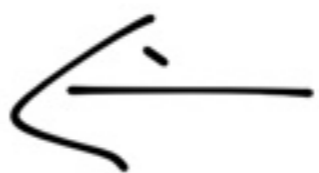
$f 3 = \text{let } g y = y + 2 \text{ in } (g 3) + 1$
 $= (3 + 2) + 1 = 6$

$f 3 ;$
 $\vdash \text{int} = 6$

let $f_x = \text{let } g\ y = y + 2, m\ (g\ x) + 1;$

$(g\ x) + 1$

$g\ (x + 1)$



let $f x =$ let $g y = y + 2$

m $g(x+1)$

$f: \underbrace{int} \rightarrow \underbrace{int} = (fun)$
 x ns

$f 3 =$ let $g y = y + 2$ $m g 4$
 $= 4 + 2 = 6$

3+5;;
- ; mt = 8

motorone
mfissa

(prefix +) 3 5;;

motorone
prefissa

- ; mt = 8

prefix + ;;
 - : mut → mut → mut = (fun)

3 + 4 + 5 ;;
 - : mut = 12

~~prefix + 3 4 5 ;; NO!~~

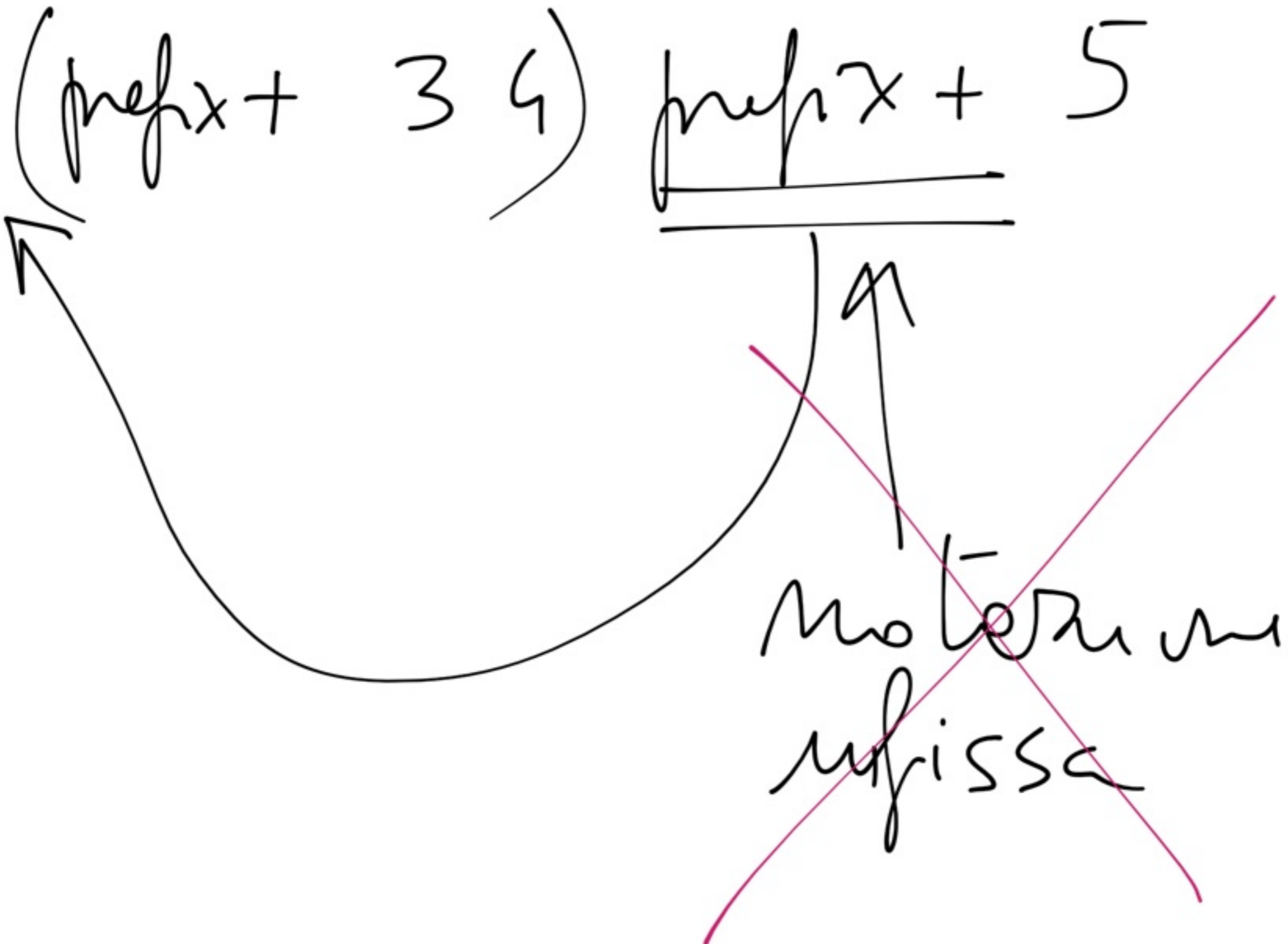
prefix + 3 (prefix + 4 5) ;;
 - : mut = 12

$\text{prefix} + (\text{prefix} + 3 \ 4) \ 5 \ ; \ ;$
 $\rightarrow \text{mut} = 12$

$((\text{prefix} +) \text{prefix} +) \ 3 \ 4 \ 5$

ERRORE DI

TIPO



pagine del corso
materiale didattico

link alle pagine CAML

CAML LIGHT

Interprete

Completare

Interprete esegue
i programmi del
linguaggio senza
tradurli.

(esegua il linguaggio
ad alto livello)

Completare traccia
i proprii su

"linguaggio macchine"

Completare i molto
più efficienti

linguaggi funzionali
 hanno ma
interpreti che
 compilatori

linguaggi interpretati.
 hanno
 compilatori.

ML he dots
origine a F#

liste CAML

liste : strutture det.
valori strutturati

dinamiche

Cambia dimensione
durante l'esecuzione
di programmi

liste CANC

sequenze di valori
(di qualsiasi tipo)
che possono essere
allungate e accorciate

dinamiche

US6 CAML

3 4 -5 6
omogenee

'c' 'c' 'a' 'z' 't'

L'accanto ai valori
è sempre fatto dal
primo valore

③ 4 -5 6

Notazione

una lista di valori CAML
 si scrive elencando i
 valori separati da
 ";" e racchiusi tra
 parentesi quadre

$$[3; -4; 5; 7]$$

Operatore principale

cons :: (CARL)

cons serve per aggiungere
un elemento in testa
a una lista

infixo

3 :: [4; 5; 6]

_____ = [3; 4; 5; 6]

lun :: [? ; 3] ; ;

ERRORE DI TIPO

~~[lun ; 2 ; 3]~~

liste patholere :

[] liste vuote

Che tipo hanno le liste

$[3; 4];;$

- : int list = $[3; 4]$

↑
tipo
degli
elementi

↑
tipo
delle
liste

[true; true; false];;

-: bool list = [true; true; false]

[];;

-: 'a list = []

[3+1; 6+2; 3];;

-: int list = [4; 8; 3]

let $f x = x + 1$ in $[f]_i$

- : $(mt \rightarrow mt)$ list = $[\langle f m \rangle]$

let $f x = x + 1$ in $[f; f]_i$

- : $(mt \rightarrow mt)$ list

= $[\langle f m \rangle; \langle f m \rangle]$

$[[3], [3;4], []];;$

- : mut lst lst = [/]

let f x = x + 1 in [[f]];;

- : (mut → mut) lst lst = [[f in]]

3 :: [];;

- : mut lst = [3]

hd ;
 - : 'a lst \rightarrow 'a = <fun>

hd [3; -3; 10] ; ;
 - : mut = 3

hd ([3; 4]; []; [1]) ; ;
 - : mut lst = [3; 4]

hd (hd ([3;4]; []); [1]);

[3;4]

-: mt = 3

hd [];

mobfinto!

tl;;

-: 'a list \rightarrow 'a list $\equiv \langle fun \rangle$

tl [3;4;-3];;

-: mut list = [4;-3]

tl [[3;()]; [7; [1]]];;

-: mut list list = [[3; [1]]]

tl [];;
 modpunkt!

3::[] \equiv [3]

tl [3];;
 -: mit list = []

tl (3::[]);;
 -: mit list = []

$3 :: (4 :: (-5 :: []))$

\equiv

$[3; 4; -5]$

hd :: 'a list -> 'a

hd $[[], [3], [4]]$

- : int list = []

$\text{tel} :: \text{Int} \rightarrow \text{List} \text{ Int}$
 $\text{tel } 3$
 Int list
 errore di tipo

$\text{Tel} :: \text{Int} \rightarrow \text{List} \text{ Int}$
 $\text{Tel } 3$

let $f\ x = x + 1$ m
 $[f\ 3; f(-2); f\ 5];;$

\rightarrow $mut\ list = [4; -1; 6]$

PATTERN SU LISTE

let $f\ l = \text{match } l \text{ with}$

$[\] \rightarrow \dots$ applica
 elementi
 la lista
 vuota
 $| \ x :: xs \rightarrow \dots$

applicando (istruzioni
 opportunamente x e xs)
 tutte le liste ottenute
 mediante l'operatore

$$a :: b \neq []$$

$$3 :: [] = [3] \neq []$$

$$[3; 4; 5] \quad 3 :: [4; 5]$$

Uguagliata con $x :: xs$
l'unica possibile
estensione di x e xs
è

$$x = 3 \quad xs = [4; 5]$$