

Linguaggio funzionale

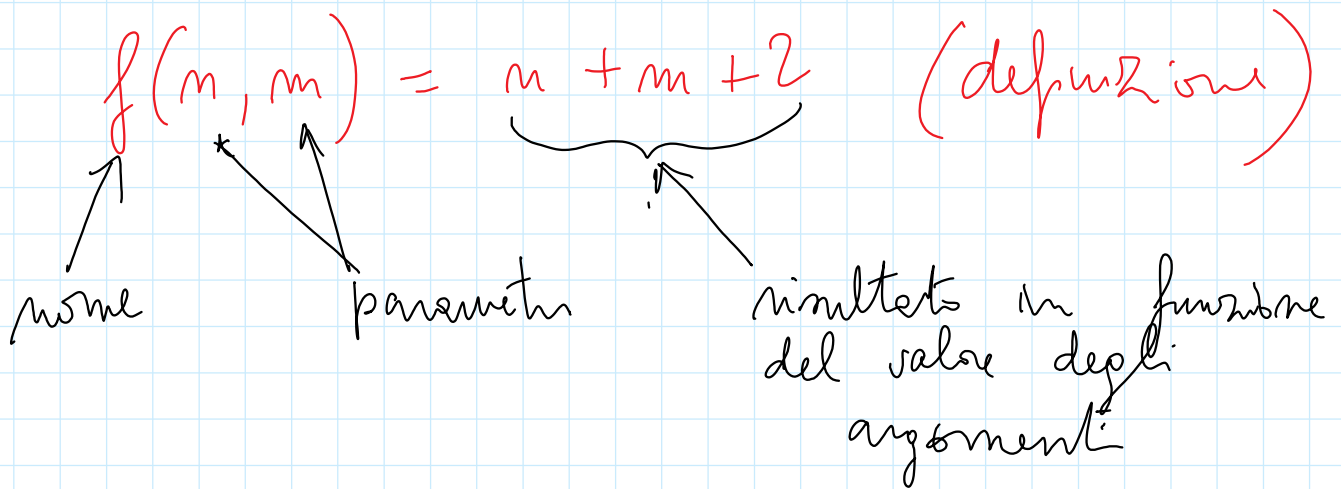
Un linguaggio in cui si

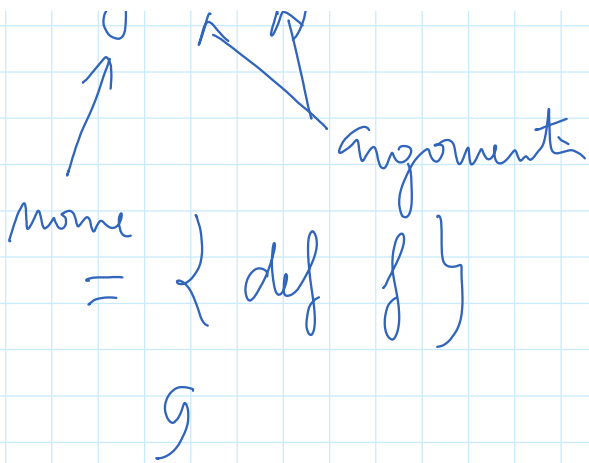
- definiscono e si
applicano

funzioni

La soluzione di un problema è ottenuta attraverso l'uso di funzioni (applicazioni) precedentemente definite.

In un linguaggio funzionale non c'è lo STATO





definizioni ricorsive di funzioni

Uno dei primi linguaggi funzionali definiti si chiamava

ML = Meta Language

→ era utilizzato per definire le semantics dei linguaggi imperativi (quello che abbiamo visto con i disegni funzionali).

09:29
CAML si pronuncia Camel

F# linguaggio funzionale della Microsoft

CAML light versione più semplice di CAML

scaricabili gratuitamente

CAML è un linguaggio INTERPRETATO.
I programmi (che sono ESPRESSIONI)
vengono eseguiti in modo interattivo

5 ; ;

espressione da eseguire

int = 5 ← valore

non è stato definito nessun nome
tipo del risultato delle espressioni

let x = 9 ; ;

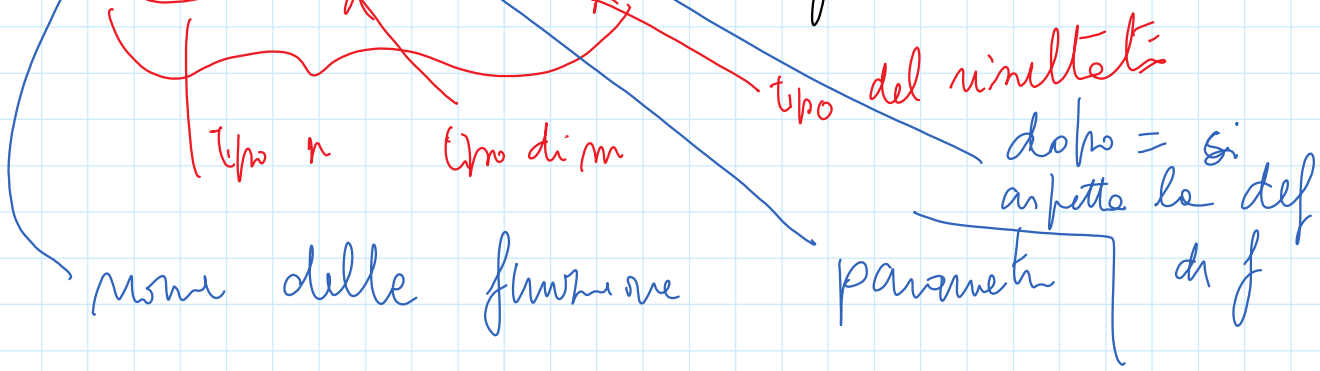
l'espressione let lega un nome al valore di una espressione

Calcolo del tipo delle espressioni si chiama:

INFERENZA DEI TIPI

let $f(m, m) = m + m + 2;$

$f: \text{int} * \text{int} \rightarrow \text{int} \ll \langle \text{fun} \rangle$



$f(3, 4);$
- : int = 9

let somma m m = m + m ;

lunedì 20 novembre 2017

Somma : $\text{int} \rightarrow (\text{int} \rightarrow \text{int}) = \langle \text{fun} \rangle$

Somma prende un
argomento intero
e restituisce una
funzione da int a int

perché non mi
dice che funzione
è ?
Perché, in generale,
non è possibile
dirlo !!

let somma m m = ...

si dice Curry ed

Curry matematico e logico americano
del 1900

let somme m m = m + m;;
Somme: int → (int → int) = <fun>

Somme è una funzione che prende un
argomento int e restituisce una
funzione di int in int.

Somme 5;;
—: int → int = <fun>

let id m = m ;;

id : 'a → 'a = <fun>

↑
variabili di tipo

funzioni
polimorfe
(possono essere applicate a tipi diversi)

id 3.5 ;;
- : float = 3.5

id 3 ;;
- : int = 3

id "str" ;;

- : string = "str"

let id1 (m, n) = (m, m+1) ;;

id1 : 'a * int → 'a * int = <fun>

Definizione ricorsiva

lunedì 20 novembre 2017 10:57

let rec $f\ n =$ if $n = 0$ then 0
else $n * f(n-1)$;;

$f : \underbrace{\text{int}}_{\text{tipo di } n} \rightarrow \underbrace{\text{int}}_{\text{tipo risultato}} = \langle f\ n \rangle$

let rec $f(m, m) =$

if $m > 0$ then $2 + f(m-1, m)$

else if $m > 0$

then $3 + f(m, m-1)$

else 3 ;

$$\forall m, m \in \mathbb{N} \cdot f(m, m) = 2 \cdot m + 3 \cdot m + 3$$

Liste CAML

lunedì 20 novembre 2017 10:58

$[]$ liste vuote

$[3; 4; 5]$ notazione per liste

-: $\text{int list} = [3; 4; 5]$

Sulle liste è definito l'operatore

`cons` (in CAML `::`)

che ha come argomenti un elemento di liste e una lista. Mette in testa alla lista risultante l'elemento

$3 :: [4; 5] = [3; 4; 5]$

$3 :: [3; 4] = [3; 3; 4]$

notazioni equivalenti:

(CAML usa la più compatte)