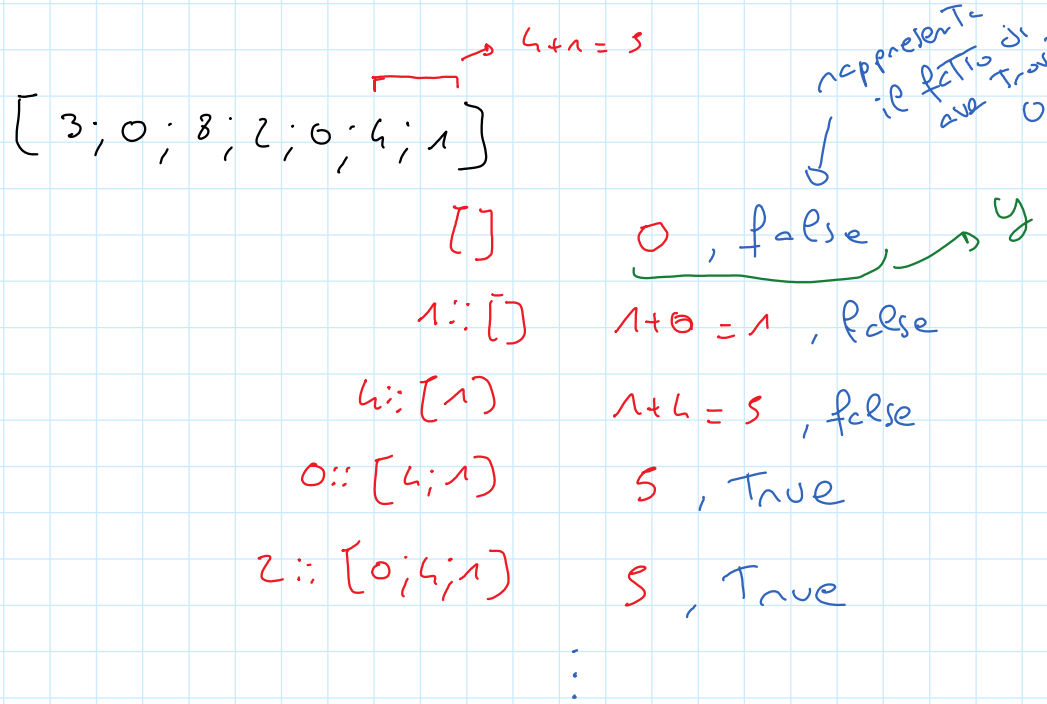
ANCORA ESEMPI CON FOLDR

ESEMPIO Somma degli elementi che seguono l'ultimo zero in una lista (senza rovesciare la lista)




```
let sommadopoultimo l =
    let f x (s, trovato) =
        if not Trovato Then
            if x<>0 Then (s+x, Trovato)
                    else (s, True)
        else (s, Trovato)
    in
```

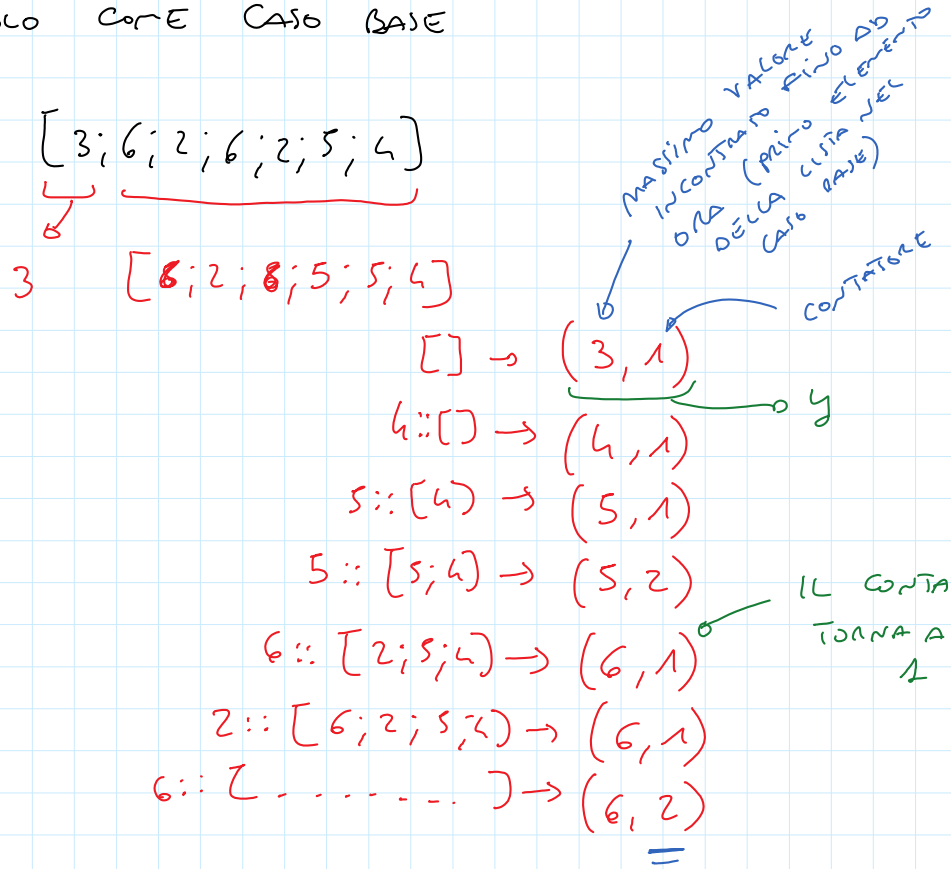
uso come y una coppia

```
let (r, b) = fold r f (0, false) e
in
r ;;
```

buffo via b che non deve fare parte del risultato

ESEMPIO: Calcolare il numero di occorrenze del valore massimo in una lista assumendo la lista non vuota (non definita su [])

COME NELL'ESEMPIO GIÀ VISTO DEL CALCOLO DEL MASSIMO TOGLIAMO IL PRIMO ELEMENTO DALLA LISTA E USIAMOLO COME CASO BASE




```
let contamax l =
    let f x (m, c) =
        if x = m then (m, c+1)
        else if x > m then
            (x, 1)
```

```
else (m, c)

in
  match e with
    z :: zs -> let (max, cont) =
                 foldn f (z,1) zs
in
  cont ;;
```

DEFINITA SOLO SUL LISTA NON VUOTA

IL RISULTATO È SOLO IL CONTATORE

ESEMPIO Data una lista costruire una lista con gli stessi elementi ma senza ripetizioni (lasciando l'ultima occorrenza di ogni valore ripetuto

[3;5;3;7;5;2] => [3;7;5;2]

[3;5;3;7;5;2] y

[]	[]
2::[]	[2] ← IL 2 NON È IN 2 QUINDI LO AGGIUNGO
5::[2]	[5;2]
7::[5;2]	[7;5;2]
3::[7;5;2]	[3;7;5;2]
5::[-----]	[3;7;5;2] ← IL 5 ERA GIÀ PRESENTE
3::[-----]	[3;7;5;2]

```
let nonripetizioni l =
    let f x y =                          (* E' UNA LISTA *)
        let p z = x=z                    (* PREDICATO CHE VERIFICA SE UN ELEMENTO E' UGUALE A X *)
        in
            if (exists p y)              (* SE X E' CONTENUTO IN Y *)
            Then y
            else x::y
    in
    foldn f [] l;;
```

ATTENZIONE AI TIPI

NELL'USO DELLA FOLDR

- IL RISULTATO DEL CASO BASE
- IL PARAMETRO Y IN f
- IL RISULTATO DI f
- IL RISULTATO DI foldr

HANNO TUTTI LO STESSO TIPO!!

NOTA

É LA PRIMA COSA CHE VADO A CONTROLLARE QUANDO CORREGGO UN ESERCIZIO

martedì 5 dicembre 2017 12:16

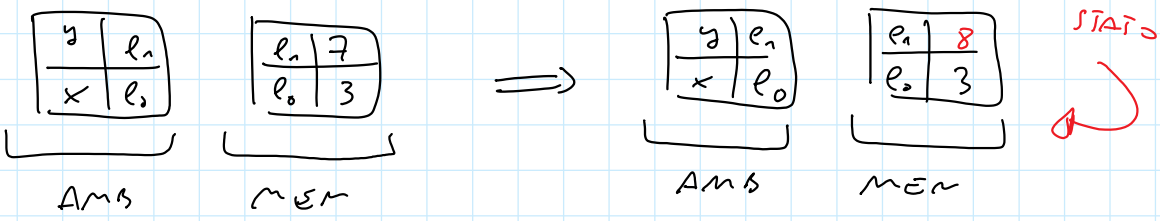
SEMANTICA DEL LINGUAGGIO C

- SEMANTICA ~ SIGNIFICATO
- IL SIGNIFICATO (SEMANTICA) DI UN PROGRAMMA È DATO DAL SUO COMPORTAMENTO

  "COME CALCOLA IL RISULTATO"
- IL COMPORTAMENTO DI UN PROGRAMMA C È DESCRITTO COME SEQUENZA DI CAMBIAMENTI DELLO STATO

```
⋮
if (x>0) y=8;
⋮
```

IL COMPORTAMENTO (SEMANTICA) DI QUESTO COMANDO È L'AGGIORNAMENTO DELLO STATO



- PER ORA ABBIAMO DESCRITTO IL COMPORTAMENTO A PAROLE USANDO I DISEGNI. ORA LO DESCRIVEREMO IN TERMINI MATEMATICI (TRAMITE FUNZIONI)

GUARDATE LA DISPENSA SUL SITO

CONTIENE ANCHE L'IMPLEMENTAZIONE IN

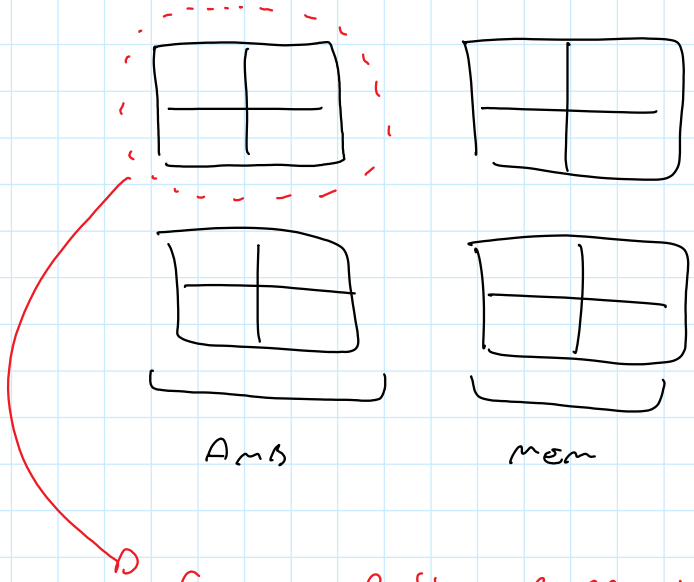
CAML DELLE funtioni CHE DEFINIREMO

(INTERPRETE DEL C SCRITTO IN CAML)

RAPPRESENTAZIONE DI UN FRAME

martedì 5 dicembre 2017 12:25

Ambiente e memoria sono pile di frame



COME POSSO RAPPRESENTARE UN FRAME?

Un frame dell ambiente

es.

y	e_1
x	e_0

è una rappresentazione Tabellare di un insieme di associazioni (coppie)

$$\{\langle x, e_0\rangle, \langle y, e_1\rangle\}$$

corrisponde a una funzione

Assumiamo

$$Ide = \{x, y, z, \dots\}$$

INSIEME DI TUTTI I POSSIBILI IDENTIFICATORI (NOMI DI VARIABILI)

$Loc = \{\ell_0, \ell_1, \ell_2, \ldots\}$ INSIEME DI TUTTI LE POSSIBILI LOCAZIONI DI MEMORIA

UN FRAME È UNA FUNZIONE PARZIALE

$$Ide \to Loc$$

es.

y	ℓ_1
x	ℓ_0

$\rightsquigarrow$ φ (fi)

$$\varphi(m) = \begin{cases} \ell_0 & m = x \\ \ell_1 & m = y \\ \text{NON DEFINITA} & \text{altrimenti} \end{cases}$$

POSSIAMO DEFINIRE UNA RAPPRESENTAZIONE DEL FRAME COME FUNZIONE TOTALE USANDO UN SIMBOLO SPECIALE $\bot$ ← bottom

$$Loc_\bot = Loc \cup \{\bot\}$$

DEFINIAMO LA FUNZIONE COME

$$Ide \to Loc_\bot$$

es.

y	ℓ_1
x	ℓ_0

$\rightsquigarrow$

$$\varphi(m) = \begin{cases} \ell_0 & \text{se } m = x \\ \ell_1 & \text{se } m = y \\ \bot & \text{altrimenti} \end{cases}$$

La funzione restituisce $\bot$ se la variabile non è presente nel frame.

ALLO STESSO MODO POSSO RAPPRESENTARE i FRAME DI MEMORIA

$$Loc \to Val_\perp$$

dove $Val = \{0, 1, 2, -1, -2, \dots\}$ insieme di TUTTI POSSIBILI VALORI MEMORIZZABILI

$$Val_\perp = Val \cup \{\perp\}$$

es.

l_1	2
l_0	7

$\rightsquigarrow$ μ (mu)

$$\mu(m) = \begin{cases} 7 & \text{se } m = l_0 \\ 2 & \text{se } m = l_1 \\ \perp & \text{altrimenti} \end{cases}$$

Quindi φ (fi) e υ (mu) sono funzioni che rappresentano singoli frame (di ambiente e di memoria)

OPERAZIONI SUI FRAME

- **LETTURA** in un frame di ambiente φ voglio sapere qual è la locazione associata a un dato identificabile

es. | y | ℓ_1 |
| --- | --- |
| x | ℓ_0 |

$\rightsquigarrow$ $\varphi(n) = \begin{cases} \ell_0 & n = x \\ \ell_1 & n = y \\ \perp & \text{altri} \end{cases}$

basta richiamare la funzione

$$\varphi(x) = \ell_0$$

analogo se si tratta di un frame di memoria

es. | ℓ_1 | 4 |
| --- | --- |
| ℓ_0 | 7 |

$\rightsquigarrow$ $\upsilon(n) = \begin{cases} 7 & \text{se } n = \ell_0 \\ 4 & \text{se } n = \ell_1 \\ \perp & \text{altri} \end{cases}$

$$\upsilon(\ell_0) = 7$$

- **AGGIUNTA DI UN'ASSOCIAZIONE** (es. diciamo che è una nuova variabile)

es.

y	e_1
x	e_0

int z;

z	e_2
y	e_1
x	e_0

$$\varphi(m) = \begin{cases} e_0 & \text{se } m = x \\ e_1 & \text{se } m = y \\ \perp & \text{altrimenti} \end{cases}$$

$$\varphi'(m) = \begin{cases} e_0 & \text{se } m = x \\ e_1 & \text{se } m = y \\ e_2 & \text{se } m = z \\ \perp & \text{altrimenti} \end{cases}$$

NUOVA FUNZIONE φ' È UNA ESTENSIONE DI φ (MA UN CASO IN PIÙ)

DEFINIAMO L'AGGIUNTA DI UNA ASSOCIAZIONE COME UNA FUNZIONE add

$$\text{add}(\varphi, z, e_2) = \begin{cases} \varphi \cup \{\langle z, e_2 \rangle\} & \text{se } \varphi(z) = \perp \\ \perp & \text{altrimenti} \end{cases}$$

AGGIUNGE UNA COPPIA A φ — SOLO SE z NON VI È GIÀ PRESENTE

φ: frame iniziale; z: nuova variabile; e_2: nuova locazione

DEFINITA SOLO SU φ CHE NON CONTIENE z

INVECE DI $\text{add}(\varphi, x, e)$ IN SEGUITO SCRIVEREMO

$$\varphi\left[{}^{e}/_{x}\right]^{\text{add}}$$

MODO DI SCRIVERE LA CHIAMATA DELLA FUNZIONE CHE

DA CUI DEDUCIAMO CHE STIAMO AGGIORNANDO ψ

ANALOGAMENTE POSSIAMO DEFINIRE L'AGGIUNTA DI UNA ASSOCIAZIONE A UN FRAME DI MEMORIA

$$V[v/e]^{add}$$

CORRISPONDE AD AGGIUNGERE LA NUOVA ASSOCIAZIONE (e,v) NEL FRAME V

DEFINITA SOLO SE e NON ANCORA ASSOCIATA A UN VALORE NEL FRAME V