

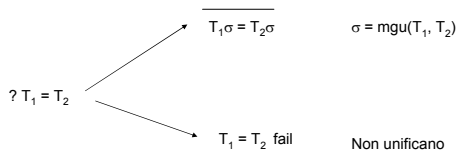
Lezione 12

PROLOG : Aritmetica e cut

Prolog I. Primi elementi

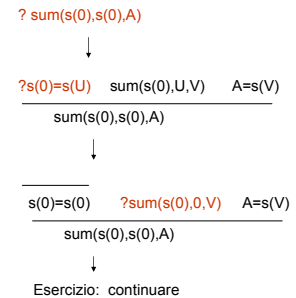
- Uso dell'identità
 - Corrisponde all'unificazione
- Aritmetica
 - Valutazioni di espressioni aritmetiche ground
- Cut
 - Controllo della strategia di ricerca top-down

Identità:



ESEMPIO.

$\text{sum}(X,Y,Z) :- Y=0, X=Y.$
 $\text{sum}(X,Y,Z) :- Y=s(U), \text{sum}(X,U,V), Z=s(V).$



ARITMETICA.

- Costanti intere 53, -5, ..
- Costanti reali: 2.5, ...
- Operazioni e predicati:
 - Operazioni: +, *, -, / (intera), / (reale), mod
 - Predicati: =:=, =\=, <, >, >=, <=
 - Quando un'espressione (termine o predicato) E è valutata con sostituzione corrente σ , E σ dev'essere ground
- Assegnamento/test: X is E; in questo caso:
 - E σ dev'essere ground;
 - X σ dev'essere una costante numerica o una variabile

ESEMPIO. $\text{pow}(A,B,P) :- P = A^B$
 modo: $\text{pow}(\ln1,\ln2,\text{Out})$ con $\text{float}(\ln1)$, $\text{integer}(\ln1)$.

$\text{pow}(A,B,1) :- B =:= 0.$
 $\text{pow}(A,B,P) :- B > 0,$
 H is B - 1,
 $\text{pow}(A,H,P1),$
 P is P1 * A.
 $\text{pow}(A,B,P) :- B < 0,$
 $\text{pow}(A,-B,P1),$
 P is 1/P1.

Esercizio: dimostrare la correttezza positiva.
 Discutere dei problemi di approssimazione.

2.CUT

- Cut: una primitiva di controllo dello spazio di ricerca; DCL verso aspetti implementativi (verso Prolog)
 - Riconsiderando il programma precedente

```

pow(A,B,1) :- B == 0.
pow(A,B,P) :- B > 0,
    H is B - 1,
    pow(A,H,P1),
    P is P1 * A.
pow(A,B,P) :- B < 0,
    pow(A,-B,P1),
    P is 1/P1.
    
```

siccome uno solo fra i casi $B == 0$, $B > 0$, $B < 0$ vale, è inutile fare backtrack sugli altri due

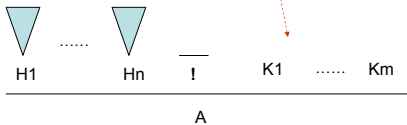
Si può usare il cut. L'algoritmo div con il cut diventa

```

pow(A,B,1) :- B == 0, !.
pow(A,B,P) :- B > 0, !,
    H is B - 1,
    pow(A,H,P1),
    P is P1 * A.
pow(A,B,P) :- pow(A,-B,P1),
    P is 1/P1.
    
```

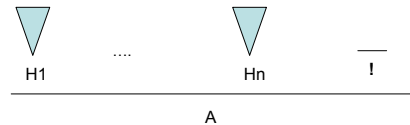
2.2.Semantica del cut con gli alberi di prova

Backtraking solo qui; la parte a sinistra di ! viene congelata e alla fine l'albero è staccato e nessun'altra clausola per A è considerata



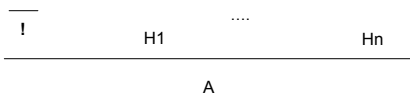
Caso particolare

Blocca l'albero.
L'intero albero viene staccato e nessun'altra clausola considerata



Caso particolare

Blocca l'atomo.
Terminato il backtraking, l'atomo A viene staccato e nessun'altra clausola per A considerata



Esempio

```

max(X,Y,Y) :- X <= Y.
max(X,Y,X) :- Y <= X.
max(X,Y,Z,U) :- X <= Y, max(Y,Z,U).
max(X,Y,Z,U) :- Y <= X, max(X,Z,U).
    
```

CON CUT:

```

max(X,Y,Y) :- X <= Y, !.
max(X,Y,X).
max(X,Y,Z,U) :- X <= Y, !, max(Y,Z,U).
max(X,Y,Z,U) :- max(X,Z,U).
    
```

max(X,Y,Y) :- X <= Y, !.
max(X,Y,X).
max(X,Y,Z,U) :- X <= Y, !, max(Y,Z,U).
max(X,Y,Z,U) :- max(X,Z,U).

?max(1,5,2,M)

max(X,Y,Y) :- X <= Y, !.
max(X,Y,X).
max(X,Y,Z,U) :- X <= Y, !, max(Y,Z,U).
max(X,Y,Z,U) :- max(X,Z,U).

?1<=5 ! max(5,2,M)

 max(1,5,2,M)

max(X,Y,Y) :- X <= Y, !.
max(X,Y,X).
max(X,Y,Z,U) :- X <= Y, !, max(Y,Z,U).
max(X,Y,Z,U) :- max(X,Z,U).

1<=5 ?! max(5,2,M)

 max(1,5,2,M)

max(X,Y,Y) :- X <= Y, !.
max(X,Y,X).
max(X,Y,Z,U) :- X <= Y, !, max(Y,Z,U).
max(X,Y,Z,U) :- max(X,Z,U).

----- -----
1<=5 ! ?max(5,2,M)

 max(1,5,2,M)

max(X,Y,Y) :- X <= Y, !.
max(X,Y,X).
max(X,Y,Z,U) :- X <= Y, !, max(Y,Z,U).
max(X,Y,Z,U) :- max(X,Z,U).

----- -----
1<=5 ! 5<=2 fail !
----- -----
 max(5,2,2)

 max(1,5,2,2)

max(X,Y,Y) :- X <= Y, !.
max(X,Y,X).
max(X,Y,Z,U) :- X <= Y, !, max(Y,Z,U).
max(X,Y,Z,U) :- max(X,Z,U).

----- -----
1<=5 ! ?max(5,2,M)

 max(1,5,2,M)

```
max(X,Y,Y) :- X <= Y, !.  
max(X,Y,X).  
max(X,Y,Z,U) :- X <= Y, !, max(Y,Z,U).  
max(X,Y,Z,U) :- max(X,Z,U).
```

C'è ancora ma non si riprende dal goal `max(1,5,2,U)`.
L'albero è bloccato da !

<u>1 <= 5</u>	<u>!</u>	<u>max(5,2,5)</u>
<u>max(1,5,2,5)</u>		

Esempi ed esercizi alla lavagna