

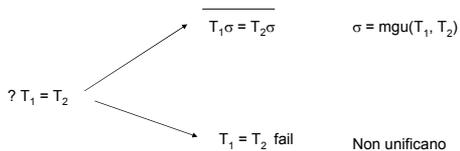
Lezione 11

- A) PROLOG I: Aritmetica e cut
- B) Preparazione prima prova in itinere

Prolog I. Primi elementi

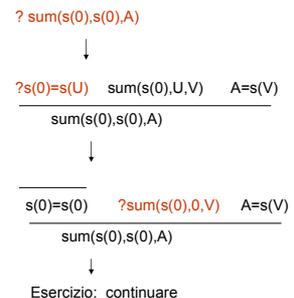
- Uso dell'identità
 - Corrisponde all'unificazione
- Aritmetica
 - Valutazioni di espressioni aritmetiche ground
- Cut
 - Controllo della strategia di ricerca top-down

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, Eσ dev'essere ground*
- **Assegnamento/test:** X is E; in questo caso:
 - *Eσ dev'essere ground;*
 - *Xσ 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

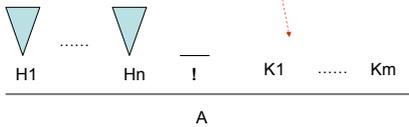
Si può usare il cut. L'algoritmo div con il cut diventa

```

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.
    
```

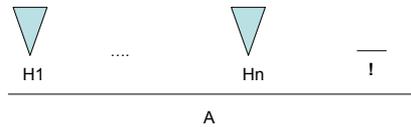
2.2.Semantica del cut con gli alberi di prova

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



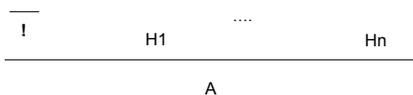
Caso particolare

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).
    
```

$\max(X,Y,Y) :- X \leq Y, !.$
 $\max(X,Y,X).$
 $\max(X,Y,Z,U) :- X \leq Y, !, \max(Y,Z,U).$
 $\max(X,Y,Z,U) :- \max(X,Z,U).$

?max(1,5,2,M)

$\max(X,Y,Y) :- X \leq Y, !.$
 $\max(X,Y,X).$
 $\max(X,Y,Z,U) :- X \leq Y, !, \max(Y,Z,U).$
 $\max(X,Y,Z,U) :- \max(X,Z,U).$

$\frac{?1 \leq 5 \quad ! \quad \max(5,2,M)}{\max(1,5,2,M)}$

$\max(X,Y,Y) :- X \leq Y, !.$
 $\max(X,Y,X).$
 $\max(X,Y,Z,U) :- X \leq Y, !, \max(Y,Z,U).$
 $\max(X,Y,Z,U) :- \max(X,Z,U).$

$\frac{1 \leq 5 \quad ?! \quad \max(5,2,M)}{\max(1,5,2,M)}$

$\max(X,Y,Y) :- X \leq Y, !.$
 $\max(X,Y,X).$
 $\max(X,Y,Z,U) :- X \leq Y, !, \max(Y,Z,U).$
 $\max(X,Y,Z,U) :- \max(X,Z,U).$

$\frac{1 \leq 5 \quad ! \quad ?\max(5,2,M)}{\max(1,5,2,M)}$

$\max(X,Y,Y) :- X \leq Y, !.$
 $\max(X,Y,X).$
 $\max(X,Y,Z,U) :- X \leq Y, !, \max(Y,Z,U).$
 $\max(X,Y,Z,U) :- \max(X,Z,U).$

$\frac{1 \leq 5 \quad ! \quad 5 \leq 2 \text{ fail} \quad !}{\max(1,5,2,2)}$

$\max(X,Y,Y) :- X \leq Y, !.$
 $\max(X,Y,X).$
 $\max(X,Y,Z,U) :- X \leq Y, !, \max(Y,Z,U).$
 $\max(X,Y,Z,U) :- \max(X,Z,U).$

$\frac{1 \leq 5 \quad ! \quad ?\max(5,2,M)}{\max(1,5,2,M)}$

Preparazione prima prova in itinere

$\max(X,Y,Z) :- X \leq Y, I.$
 $\max(X,Y,X).$
 $\max(X,Y,Z,U) :- X \leq Y, I, \max(Y,Z,U).$
 $\max(X,Y,Z,U) :- \max(X,Z,U).$

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

$1 \leq 5$! $\max(5,2,5)$
 $\max(1,5,2,5)$

Struttura prova

- 6/8 domande
- 2 esercizi

Argomenti Domande

- Agenti intelligenti: generalità (prime 3 lezioni)
- La nozione di RRS:
 - architettura, sintassi e semantica, implementazione, RS
- Uso di un RRS
 - semantica interna/esterna, astrazione

- Logica Proposizionale come RRS:
 - Astrazione di LP;
 - Sintassi e semantica (interpretazioni $I, I \models A$)
 - Teorie (KB) e loro modelli ($KB \models A$)
 - Nozione (generica) di dimostrazione ($KB \vdash A$)
 - validità e completezza ($KB \vdash A$ sse $KB \models A$)

- Datalog Proposizionale (DLP)
 - Sintassi (atomi, testa, corpo, clausole, fatti, regole, KB)
 - Il sistema di ragionamento e le prove ad albero
 - Procedura top-down
 - Non determinismo don't know e don't care
 - Esercizi di costruzione di semplici alberi di prova
 - Procedura bottom-up
 - Operatore T_{KB}
 - Modello minimo = minimo punto fisso
 - Esercizi di generazione del modello minimo

- Datalog
 - Predicati, variabili, costanti; termini, atomi
 - Interpretazioni $I = (D, \Phi, \pi)$
 - Interpretazioni di Herbrandt = insiemi di atomi ground
 - Modelli di una KB DataLog
 - Le prove in DataLog
 - Procedura top-down
 - Unificazione e mgu, applicazione di una regola
 - Esercizi: ricavare alberi di prova
 - Procedura bottom-up
 - Operatore T_{KB}
 - KB^* , validità, completezza, minimo punto fisso, modello minimo
 - Esercizi: calcolo KB^*

- DCL
 - Sintassi estesa con le funzioni
 - Interpretazioni $I = (D, \Phi, \pi)$
 - Universo e base di Herbrandt
 - Dominio chiuso/aperto, nome unico
 - Interpretazioni di Herbrandt $H \subseteq B_{KB}$
 - Modelli di un programma in DCL
 - Procedura top-down: come per datalog, ma algoritmo di unificazione
 - Procedura bottom-up: come per datalog, ma infinite istanze e iterazione sino ad ω

- Come costruire una KB:
 - Correttezza e copertura
- CWA e definizioni induttive:
 - Mondo chiuso ed ipotesi di minimalità in una definizione induttiva

Esercizi tipo

- Verranno presentati a lezione e poi messi in linea