

COMPITO DI INTELLIGENZA ARTIFICIALE (v.o.) – PARTE I
FONDAMENTI DI INTELLIGENZA ARTIFICIALE

13 Settembre 2004 (Tempo a disposizione 2h; su 32 punti)

Esercizio 1: (punti 7)

Si rappresentino in logica del I ordine le seguenti frasi:

- Gli aerei viaggiano più lontano dei treni
- Gli eurostar sono treni
- Gli eurostar viaggiano più lontano delle automobili
- AZ154 è un aereo
- ES123 è un eurostar
- FIAT500 è un'auto

Le si trasformi in clausole e, applicando il principio di risoluzione, si verifichi se esiste un mezzo che viaggia più lontano dell'ES123.

Esercizio 2 (punti 6)

Si scriva un programma Prolog:

substitutelist(X, Y, Xs, Ys)

che produce la lista Ys ottenuta come risultato sostituendo Y ad ogni occorrenza di X nella lista Xs.

Esempio:

```
?- substitutelist(a,c,[a,f,c,a],Ys).  
yes Ys = [c,f,c,c]
```

Esercizio 3 (punti 6)

Si consideri il seguente problema di soddisfacimento di vincoli:

A::[1..29], B::[6..30], C::[4..25], D::[1..100]

$D \leq A$, $C + 5 > B$, $D - 6 > C$, $A \neq B$

Si applichi alla rete la arc consistenza. Si proceda alla ricerca di una soluzione applicando una euristica first-fail e propagando i vincoli tramite il forward checking.

Esercizio 4 (punti 6)

Si consideri il seguente programma Prolog:

```
sum(0, X, X).  
sum(s(A), B, s(C)) :-  
    sum(A, B, C).
```

Si rappresenti l'albero SLDNF corrispondente al goal

```
?- sum(s(0), Y, X), not(sum(0, X, Y)).
```

sia nel caso in cui venga verificato l'occur-check sia nel caso non venga verificato.

Esercizio 5 (punti 5)

Si discuta sinteticamente il trattamento della negazione nella programmazione logica e Prolog.

SOLUZIONE

Esercizio 1:

Frasi in Logica

$\forall X, \forall Y \text{ aereo}(X) \text{ and treno}(Y) \Rightarrow \text{piu_lontano}(X, Y)$

$\forall X \text{ eurostar}(X) \Rightarrow \text{treno}(X)$

$\forall X, \forall Y \text{ eurostar}(X) \text{ and auto}(Y) \Rightarrow \text{piu_lontano}(X, Y)$

$\text{aereo}(\text{az154})$

$\text{eurostar}(\text{es123})$

$\text{auto}(\text{fiat500})$

Query: esiste un mezzo che va più lontano di es123?

$\exists X \text{ piu_lontano}(X, \text{es123})$

Clausole

1. not aereo (X) or not treno(Y) or piu_lontano(X,Y)
2. not eurostar(X) or treno(X)
3. not eurostar(X) or not auto(Y) or piu_lontano(X,Y)
4. aereo(az154)
5. eurostar(es123)
6. auto(fiat500)

Goal:

7. not piu_lontano(X,es123)

Da 7 e 1:

8. not aereo (X) or not treno(es123)

da 8 e 2:

9. not aereo (X) or not eurostar(es123)

da 9 e 5:

10. not aereo (X)

da 10 e 4 si ottiene la clausola vuota.

Esercizio 2

$\text{substitutelist}(X, Y, [], []) .$

$\text{substitutelist}(X, Y, [X|Xs], [Y|Ys]) :- \text{substitutelist}(X, Y, Xs, Ys) .$

$\text{substitutelist}(X, Y, [Z|Xs], [Z|Ys]) :- X \neq Z,$

$\text{substitutelist}(X, Y, Xs, Ys) .$

Esercizio 3

$A::[1..29], B::[6..30], C::[4..25], D::[1..100],$

$D \leq A, C + 5 > B, D - 6 > C, A \neq B$

$D \leq A$

$A::[1..29], D::[1..29]$

$C + 5 > B$
 $B::[6..29], C::[4..25],$

$D - 6 > C$
 $D::[11..29], C::[4..22]$

risveglio del vincolo $D \leq A$
 $A::[11..29], D::[11..29]$

Risveglio del vincolo $C + 5 > B$
 $B::[6..26], C::[4..22],$

$A \neq B$ non provoca propagazioni

$A::[11..29], B::[6..26], C::[4..22], D::[11..29]$

Applico il forward checking con first fail

$A=11$
 $C::[4..22]$
 $B::[6..10,12..26]$
 $D::[11]$

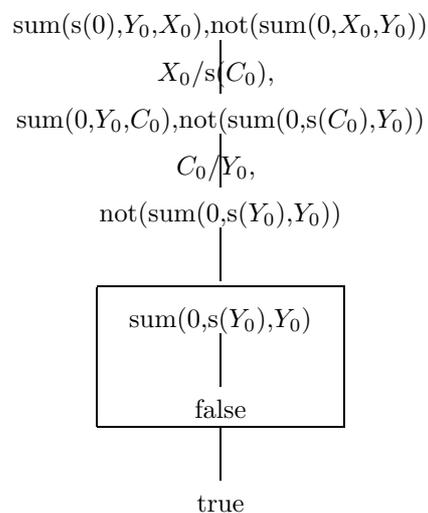
$D=11$
 $B::[6..10,12..26]$
 $C::[4]$

$C=4$
 $B::[6..8]$

$B=6$

Esercizio 4

Nel caso in cui l'occur-check venga verificato:



si noti come il goal $\text{sum}(0, s(Y_0), Y_0)$ non unifichi con la prima clausola.

Nel caso in cui l'occur-check non venga verificato:

$\text{sum}(s(0), Y_0, X_0), \text{not}(\text{sum}(0, X_0, Y_0))$

$X_0/s(C_0),$

$\text{sum}(0, Y_0, C_0), \text{not}(\text{sum}(0, s(C_0), Y_0))$

$C_0/Y_0,$

$\text{not}(\text{sum}(0, s(Y_0), Y_0))$

$\text{sum}(0, s(Y_0), Y_0)$
 $Y_0/s(Y_0)$
 true

false