FONDAMENTI DI INTELLIGENZA ARTIFICIALE (6 CFU)

29 Gennaio 2015 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (6 punti)

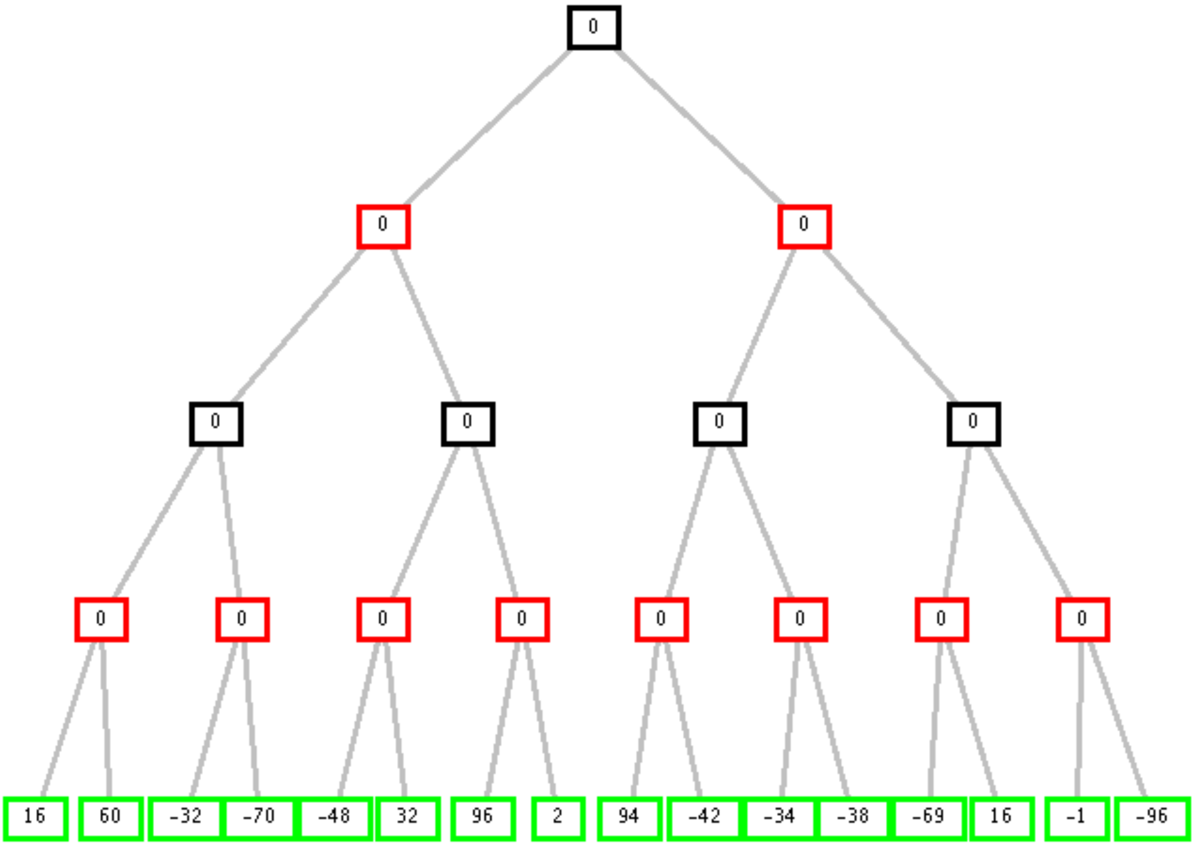
Si esprimano in logica dei predicati del I ordine le seguenti frasi:

- Alcuni botanici sono eccentrici;
- Alcuni botanici non amano gli eccentrici.

SI utlizzi a tal scopo il seguente vocabolario: b(X) per indicare che X è un botanico, e(X) per indicare che X è un eccentrico, e a(X,Y) per indicare che X ama Y.
Si usi poi il principio di risoluzione per dimostrare che "esistono alcuni che non sono amati da tutti i botanici" (in parole equivalenti "esistono alcuni tali che non tutti i botanici li amano") .

Esercizio 2(5 punti)

Si consideri il seguente albero di gioco in cui la valutazione dei nodi terminali è dal punto di vista del primo giocatore (*MAX*). Si mostri come gli algoritmi *min-max* e *alfa-beta* risolvono il problema.



Esercizio 3 (5 punti)

Definire nel linguaggio *Prolog* il predicato `count(X,L,N)`, che dato un termine `X` ground, e una lista di termini ground `L`, ha successo con `N` numero di volte in cui `X` compare in `L`. Esempi:

```
?-count(1, [1,4,5,4,1,1], N).
yes    N=3
?-count(1, [1,4,5,4,1,1], 3).
yes
?-count(2, [1,4,5,4,1,1], N).
yes N=0
?-count(f(5), [f(1),4,f(5),4,1,1], N).
yes N=1
```

Esercizio 4 (6 punti)

Si modelli come problema a vincoli il problema “delle case e delle famiglie”: 4 famiglie A, B, C e D vivono vicine, in case consecutive, numerate da 1 a 4.

- D vive in una casa con numero inferiore a quello della casa di B;
- B vive di fianco ad A, con numero più alto;
- C’è almeno una casa tra B e C;
- D non vive nella casa n. 2;
- C non vive nella casa n. 4.

Si disegni la rete che rappresenta il problema, resa già node-consistente.

Si cerchi una soluzione applicando la propagazione Forward Checking ad ogni passo, selezionando per il labeling le variabili secondo l’ordine A, B, C, D, etc. e i valori nei domini secondo il loro ordine crescente.

Esercizio 5 (6 punti)

Sia dato il seguente predicato Prolog:

```
factorial(X,1):- X=<1,!.
factorial(X,Y):-
      A is X-1,
      factorial(A,R),
      Y is R*X.
```

Disegnare l’albero SLD esplorato dall’interprete Prolog, per il goal:

```
?- factorial(3,X).
```

Esercizio 6 (4 punti)

Si introduca brevemente il concetto di negazione per fallimento usato in Prolog, e se ne discutano brevemente (con esempi) le possibili problematiche nel suo utilizzo.

FONDAMENTI DI INTELLIGENZA ARTIFICIALE

29 Gennaio 2015 – Soluzioni

Esercizio 1

a) Alcuni botanici sono eccentrici
   $\exists X\ \ b(X) \text{ and } e(X)$
b) Alcuni botanici non amano cose eccentriche
   $\exists X\ \ (\ b(X) \text{ and } \forall Y\ (e(Y) \rightarrow \neg a(X,Y)\ )$

Formula da dimostrare:

c) Alcuni non sono amati da tutti i botanici
   $\exists Y\ \ (\ \neg\forall X\ (b(X) \rightarrow a(X,Y)\ )\ )$

Trasformazione in clausole:

a) $\exists X\ \ b(X) \text{ and } e(X)$
Introducendo Skolem:
   $b(c_0) \text{ and } e(c_0)$
cioè:
   C1: $\mathbf{b(c_0)}$
   C2: $\mathbf{e(c_0)}$

b) $\exists X\ \ (\ b(X) \text{ and } \forall Y\ (e(Y) \rightarrow \neg a(X,Y)\ )$
   $\exists X\ \ \forall Y\ (\ b(X) \text{ and } (\neg e(Y) \text{ or } \neg a(X,Y)\ )$
Introducendo Skolem:
   $b(c_1) \text{ and } (\neg e(Y) \text{ or } \neg a(c_1,Y)\ )$
cioè:
   C3: $\mathbf{b(c_1)}$
   C4: $\mathbf{\neg e(Y) \text{ or } \neg a(c_1,Y)}$

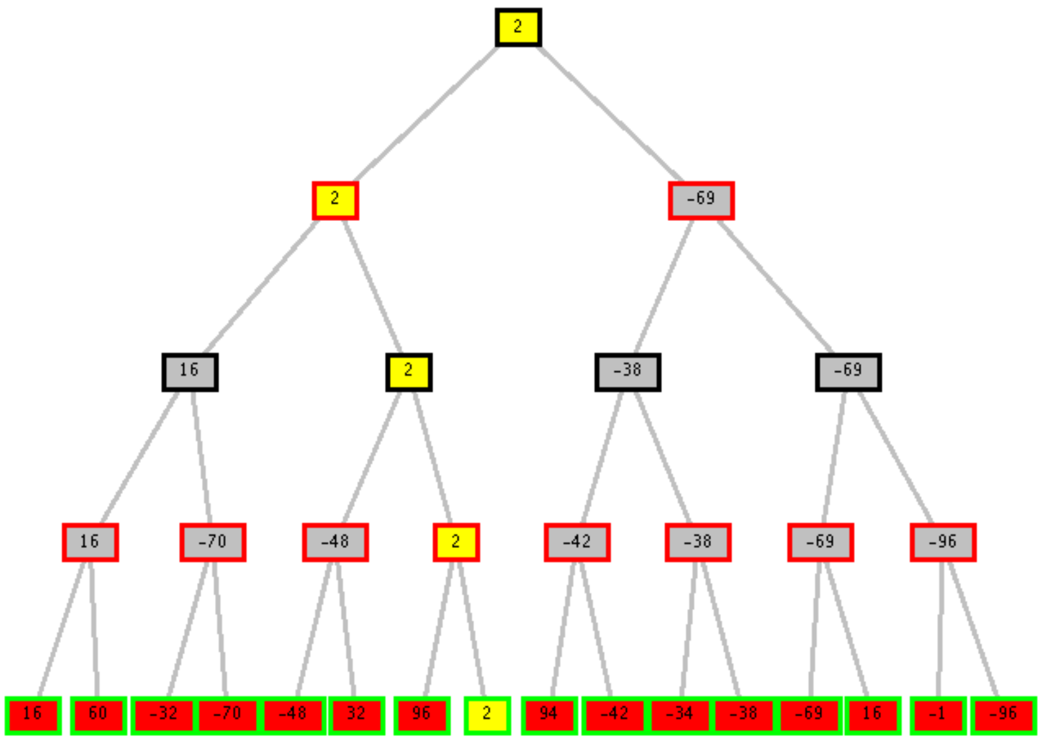
c) $\exists Y\ \ (\ \neg\forall X\ (b(X) \rightarrow a(X,Y)\ )\ )$
Goal negato:
$\neg\ \exists Y\ \ (\ \ \neg\forall X\ (b(X) \rightarrow a(X,Y)\ )\ )$
$\forall Y\ \ \neg\ (\ \ \exists X\ \neg\ (b(X) \rightarrow a(X,Y)\ )\ )$
$\forall Y\ \ \neg\ \exists X\ (\ \ \neg(b(X) \rightarrow a(X,Y)\ )\ )$
$\forall Y\ \ \forall X\ \neg\ \ (\ \ \neg(b(X) \rightarrow a(X,Y)\ )\ )$
$b(X) \rightarrow a(X,Y)$
   C5: $\mathbf{\neg\ b(X) \text{ or } a(X,Y)}$

Applicando il Principio di Risoluzione:

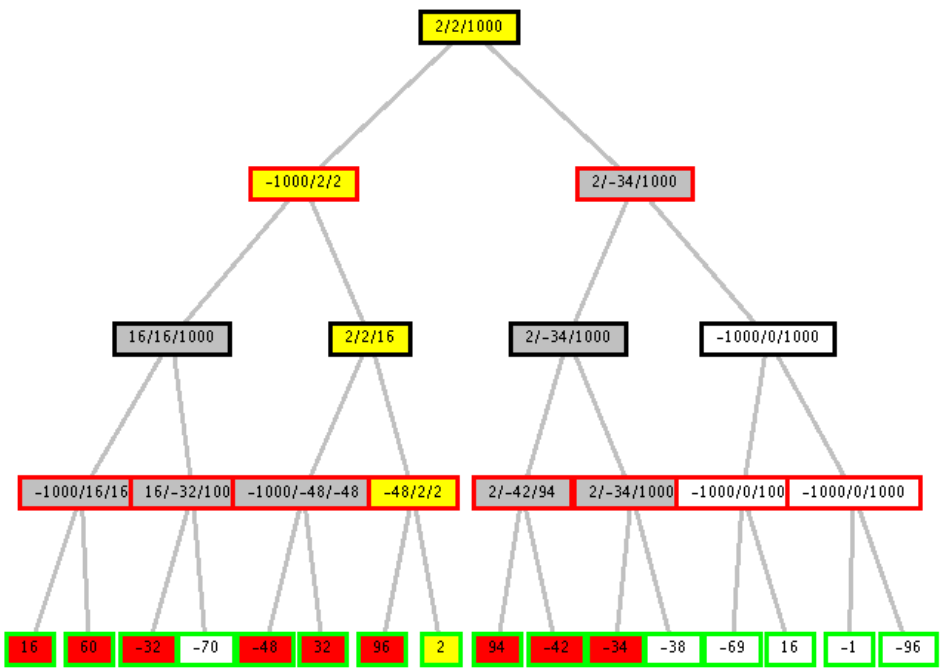
C6: C5+C4: $\neg\ b(c_1)\ \text{ or } \neg e(Y)$
C7: C6+C3: $\neg e(Y)$
C8: C7+C2: clausola vuota

Esercizio 2

Min-max:



Alfa-Beta:



Esercizio 3

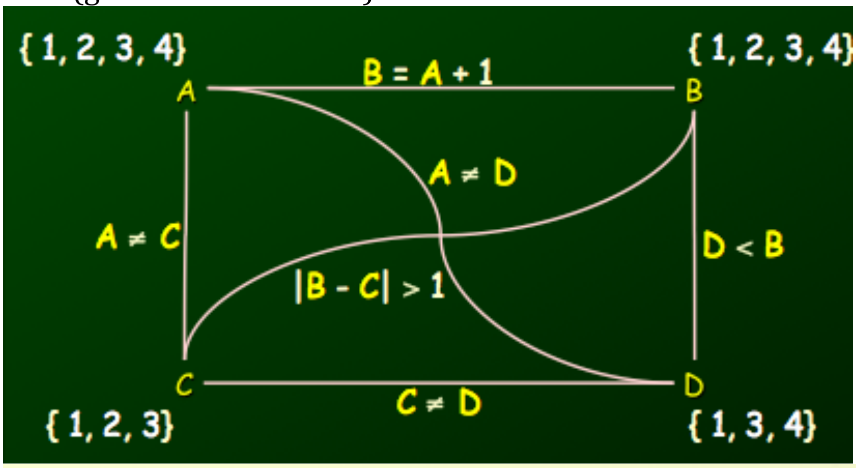
```
count(_,[],0).
count(X,[X|T],N):-!,count(X,T,N1), N is N1+1.
count(X,[_|T],N):-count(X,T,N).
```

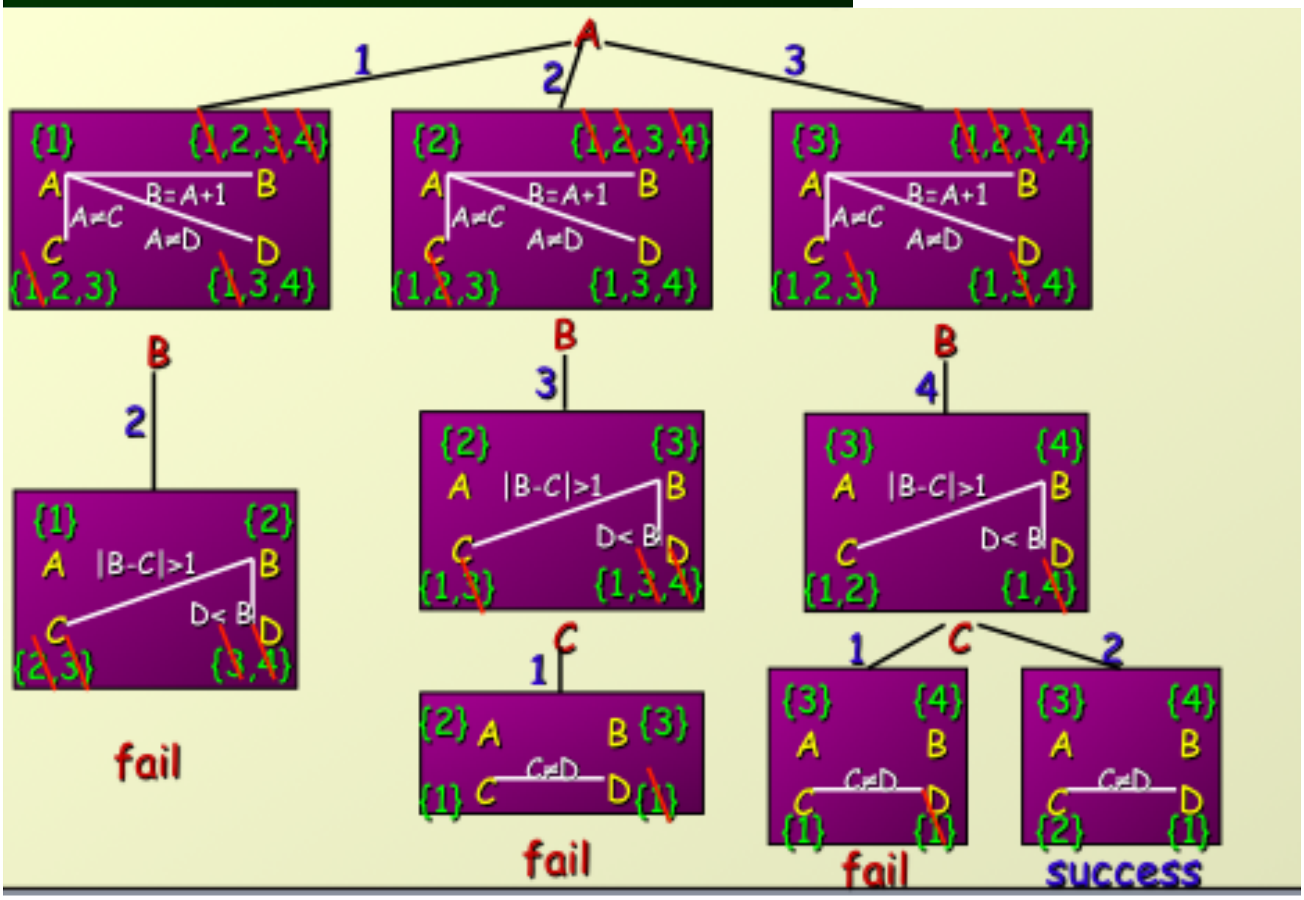
Nota: l’uso del cut evita di generare in backtracking soluzioni spurie

Esercizio 4

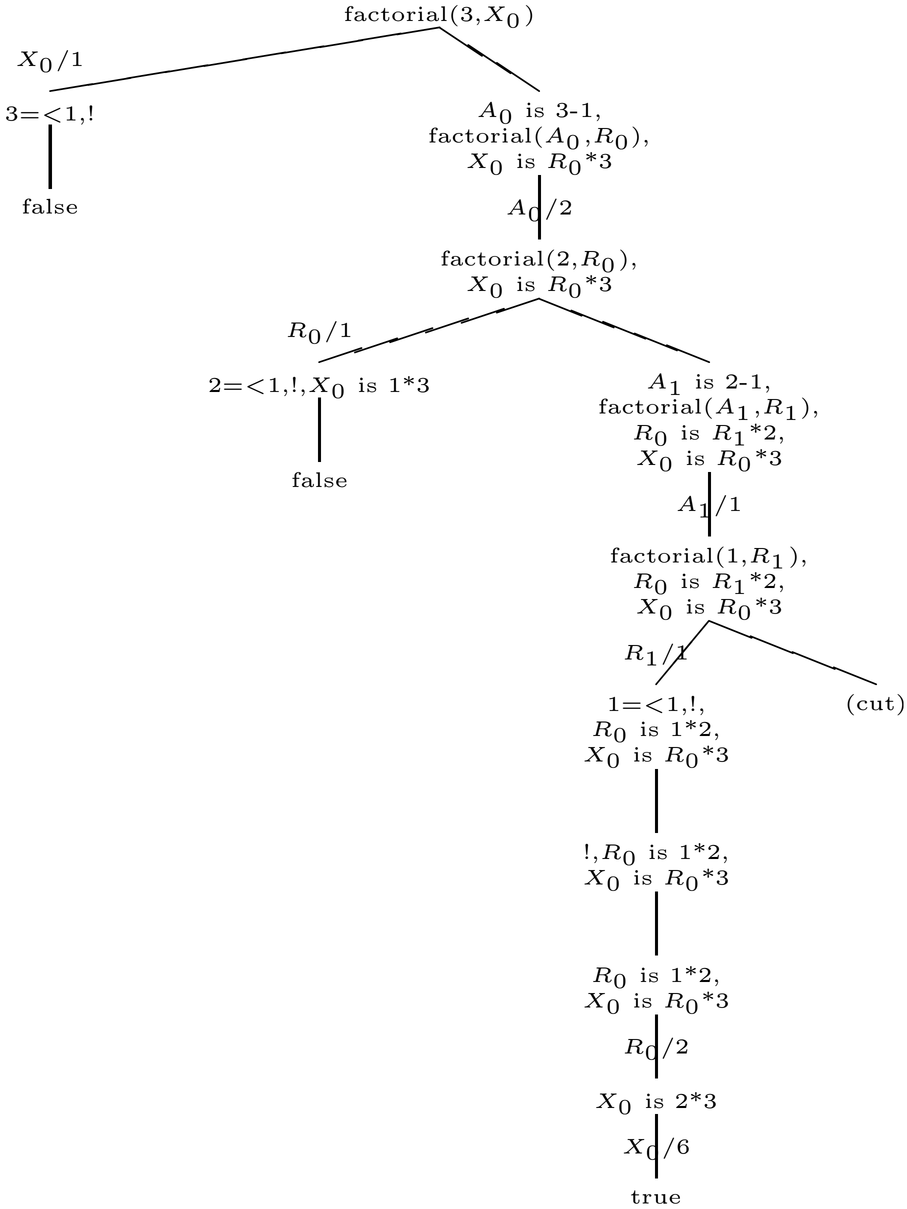
A,B,C,D::[1,2,3,4]

$D<B$
$B=A+1$
$|B-C|>1$
$A \neq C$, $A \neq D$, $C \neq D$
$C \neq 4$
$D \neq 2$
Rete (già node consistente):





Esercizio 5



Esercizio 6
Vedi slide del corso.

Esercizio 7
Vedi slide del corso.