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

5 Novembre 2003 – H. 11 – Durata 2 ore

Esercizio 1 (punti 7)

Si formalizzino le seguenti frasi in logica dei predicati:

- Esiste almeno studente di Ingegneria che conosce la logica booleana.
- Chi conosce la logica booleana ha capacità logiche.
- Chi non ha capacità logiche, si contraddice.
- Chi si contraddice, non ha capacità logiche.
- Piero studia ad ingegneria e conosce la logica booleana.

Le si trasformi in clausole e si usi poi il principio di risoluzione per dimostrare che c'è uno studente di Ingegneria che non si contraddice.

Esercizio 2 (punti 6)

Si definisca un predicato `no_ripetuti(Xs, Ys,N)` che è vero se `Ys` è la lista degli elementi in `Xs` senza duplicazioni e `N` il numero di elementi ripetuti.
Esempio:

```
:- no_ripetuti([3,5,3,9,9,8,9], Ys, N).
yes
Ys=[5,3,8,9] N=3
```

Esercizio 3 (punti 8)

Si consideri il seguente programma Prolog:

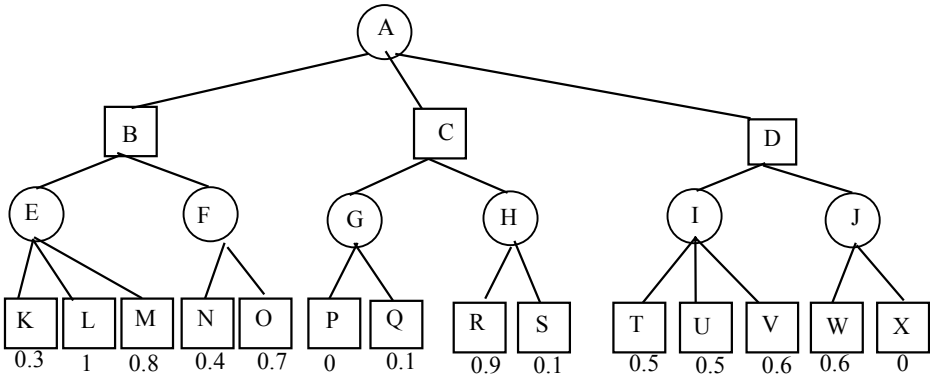
```
listap([]).
listap([A,B|T]):-
      s(A,B),
      listap(T).

s(p(X),X).
s(X,p(X)):-
      not(s(X,X)).
```

Si rappresenti l'albero SLDNF corrispondente al seguente goal:

```
listap([1,F,F,G]).
```

Esercizio 4 (punti 6)

Si consideri il seguente albero di gioco dove i punteggi (compresi tra 0 e 1) sono tutti dal punto di vista del primo giocatore (0 indica la sconfitta di MAX e 1 la sua vittoria) :



Supponendo che il primo giocatore sia MAX, quale mossa dovrebbe scegliere? Si risolva l'esercizio tramite l'algoritmo MIN-MAX. Successivamente si mostrino i tagli alfa-beta.

Esercizio 5 (punti 5)

Discutere la non completezza dell'interprete Prolog, motivando opportunamente.

SOLUZIONE

Esercizio 1

∃Y (studIng(Y) and conosce(Y,boole))
∀ X (conosce(X,boole) => haLogica(X))
∀ X (not haLogica(X) => contraddice(X))
∀ X (contraddice(X) => not haLogica(X))
studIng(piero) and conosce(piero,boole)

Goal: ∃Y studIng(Y) and not contraddice(Y)

Clausole:

C1. studIng(c)
C2. conosce(c,boole)
C3. not conosce(X,boole) or haLogica(X)
C4. haLogica(X) or contraddice(X)
C5. not contraddice(X) or not haLogica(X)
C6. studIng(piero)
C7. conosce(piero,boole)
C8. not studIng(Y) or contraddice(Y) (goal negato)

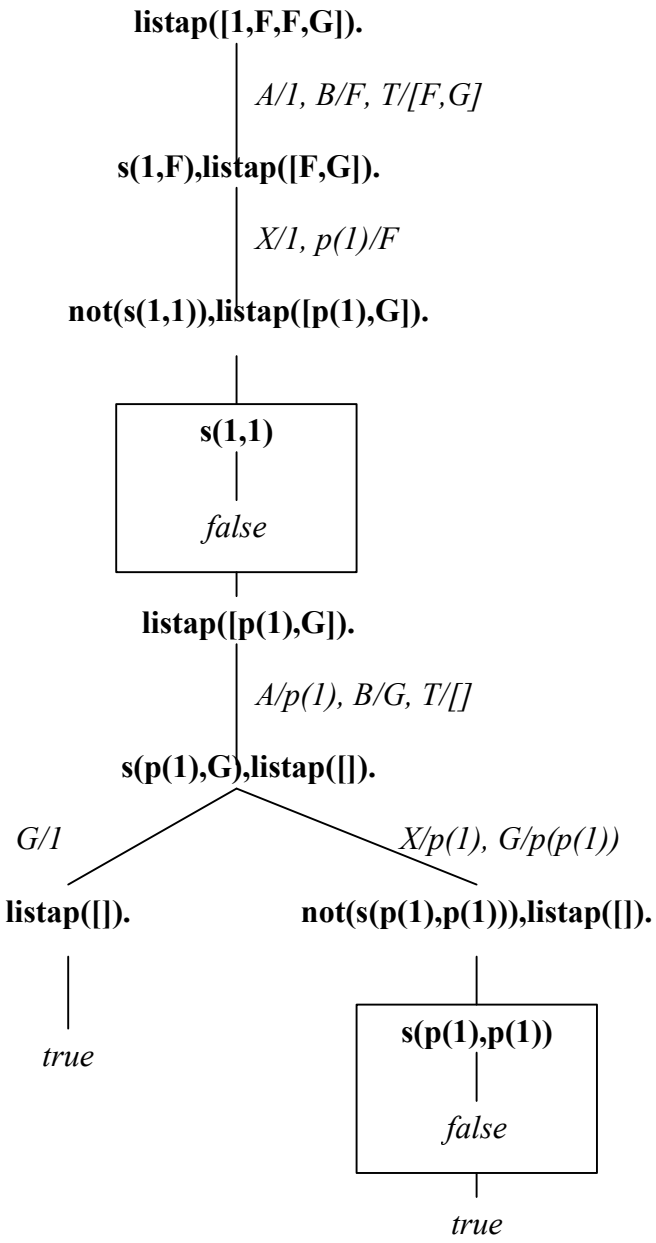
Risoluzione:

C9. not haLogica(Y) or not studIng(Y) (da C5 e C8)
C10. not conosce(Y,boole) or not studIng(Y) (da C9 e C3)
C11. not conosce(piero,boole) (da C10 e C6)
C12. Clausola vuota (da C11 e C7)

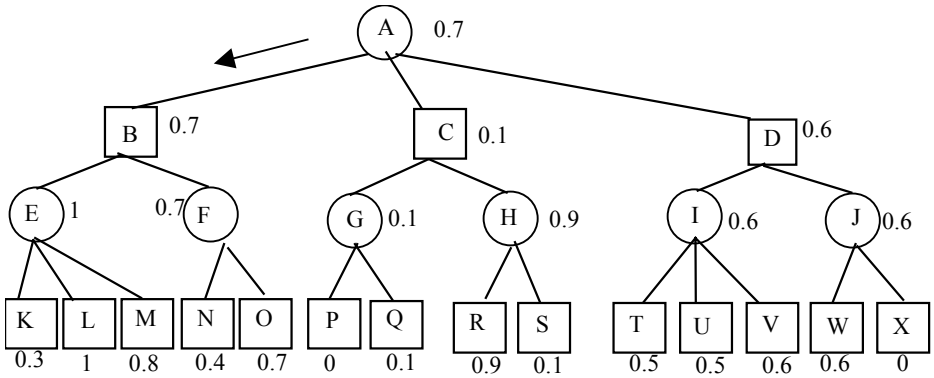
Esercizio 2

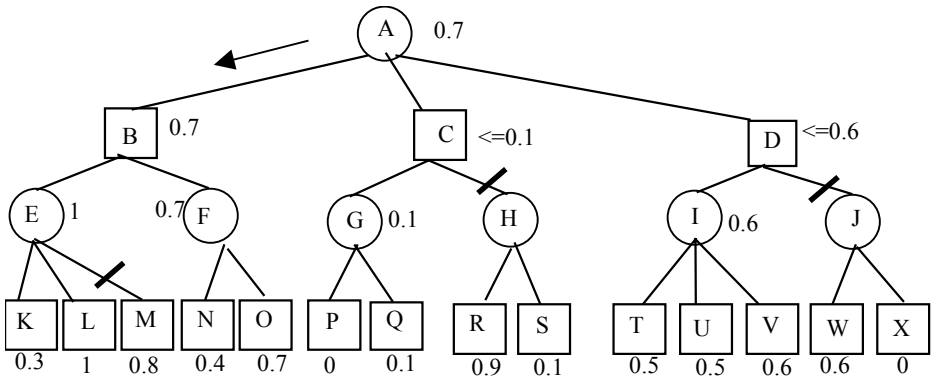
```
no_ripetuti([], [], 0).
no_ripetuti([X|Xs], Ys, N):-
  member(X, Xs),
  no_ripetuti(Xs, Ys, N1), N is N1 + 1.
no_ripetuti([X|Xs], [X|Ys], N):-
  not member(X, Xs),
  no_ripetuti(Xs, Ys, N).

member(X,[X|Xs]).
member(X,[Y|Ys]):-member(X,Ys).
```

Esercizio 3


listap([1,F,F,G]).
A/1, B/F, T/[F,G]
s(1,F),listap([F,G]).
X/1, p(1)/F
not(s(1,1)),listap([p(1),G]).
s(1,1)
false
listap([p(1),G]).
A/p(1), B/G, T/[]
s(p(1),G),listap([]).
G/1
X/p(1), G/p(p(1))
listap([]).
not(s(p(1),p(1))),listap([]).
true
s(p(1),p(1))
false
true

Esercizio 4


A 0.7
B 0.7
C 0.1
D 0.6
E 1
0.7 F
G 0.1
H 0.9
I 0.6
J 0.6
K L M N O P Q R S T U V W X
0.3 1 0.8 0.4 0.7 0 0.1 0.9 0.1 0.5 0.5 0.6 0.6 0


A 0.7
B 0.7
C <=0.1
D <=0.6
E 1
0.7 F
G 0.1
H
I 0.6
J
K L M N O P Q R S T U V W X
0.3 1 0.8 0.4 0.7 0 0.1 0.9 0.1 0.5 0.5 0.6 0.6 0