

# FONDAMENTI DI INTELLIGENZA ARTIFICIALE

12 Gennaio 2017 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

## Esercizio 1 (5 punti)

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

1. Tutti quelli che salgono sull'autobus e non hanno un valido titolo di viaggio devono comprare un biglietto dal guidatore, e viceversa;
2. Paolo sale sull'autobus;
3. Paolo ha un valido titolo di viaggio;

Si stabilisca, motivando la risposta, se è corretta la deduzione che:

Query: "Paolo non deve comprare alcun biglietto dal guidatore"

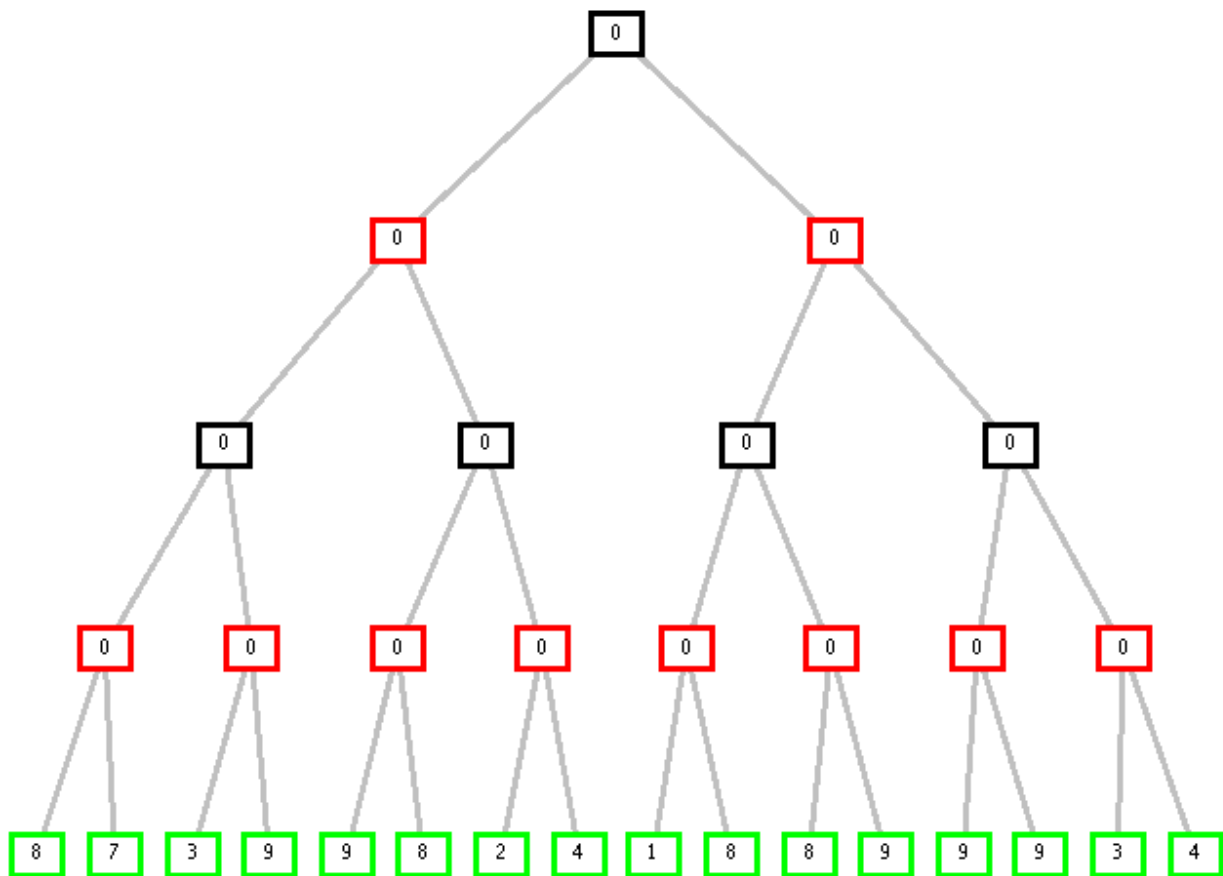
utilizzando il principio di risoluzione. Se la formula non è deducibile, se ne motivi qualitativamente la causa.

Nella formalizzazione si utilizzino tre simboli di predicato unario ( $s$ ,  $t$ ,  $c$ ), interpretati come segue:

- $s(X)$ ,  $X$  sale sull'autobus;
- $t(X)$ ,  $X$  ha un valido titolo di viaggio;
- $c(X)$ ,  $X$  deve comprare un biglietto dal guidatore.

## 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 l'algoritmo *min-max* e l'algoritmo *alfa-beta* risolvono il problema e la mossa selezionata dal giocatore.



### Esercizio 3 (4 punti)

Dato seguente programma Prolog:

```
deletel(X, [X|R], R) :- !.
```

```
deletel(X, [_|T], Y) :- deletel(X, T, Y).
```

si mostri l'abero SLD-NF che si origina con il goal:

```
?-deletel(a, [b,c,a], R), \+deletel(a, R, _).
```

### Esercizio 4 (5 punti)

Si scriva un predicato Prolog `memberlist` che cerca un elemento all'interno di una lista composta da sottoliste. Si abbia cura di specificare tutti i predicati utilizzati.

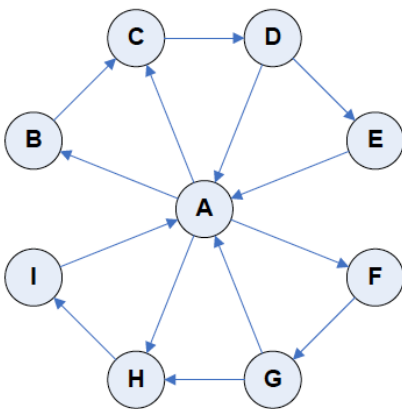
Esempio:

```
?- memberlist(a, [[b,c],[],[n,a,c]])
```

```
yes
```

### Esercizio 5 (6 punti)

Si consideri il problema di ricerca dato in figura, dove lo stato iniziale è A e lo stato obiettivo è E. Ogni arco ha costo unitario.



Risolvere il problema mediante ricerca ad approfondimento iterativo, senza eliminazione degli stati ripetuti. A parità di altro, si preferisca l'espansione dei nodi in ordine alfabetico. Si riporti l'albero di ricerca per ogni livello di approfondimento, segnalando: il livello di approfondimento e l'ordine di espansione dei nodi. Si dica se la soluzione trovata è ottima.

### Esercizio 6 (5 punti)

Si consideri il problema di colorare le celle della griglia 2x3 qui riportata:

1	2	3
4	5	6

Ogni cella può essere colorata di blu (B) o di rosso (R), l'obiettivo è di non avere due celle adiacenti colorate con lo stesso colore. Si considerano adiacenti le celle con un lato in comune; per esempio le celle 1 e 2 e le celle 1 e 4 sono adiacenti, mentre le celle 1 e 5 non lo sono.

Si formuli il problema come CSP, indicando variabili, domini e vincoli.

Si applichi poi l'arc consistenza ai domini iniziali.

Si risolva poi il CSP tramite ricerca nello spazio degli stati, applicando una ricerca in profondità con backtracking. Si riporti l'albero di ricerca generato, considerando le variabili e i valori nel dominio secondo l'ordinamento alfabetico.

### Esercizio 7 (2 punti)

Si introduca brevemente l'architettura di un sistema basato sulla conoscenza e si sottolinei la sua differenza rispetto al concetto di algoritmo.

# FONDAMENTI DI INTELLIGENZA ARTIFICIALE

12 Gennaio 2017 – Soluzioni

## Esercizio 1

1. Tutti quelli che salgono sull'autobus e non hanno un valido titolo di viaggio devono comprare un biglietto dal guidatore;

$$\forall X, s(X) \wedge \neg t(X) \rightarrow c(X)$$

$$\forall X, c(X) \rightarrow s(X) \wedge \neg t(X)$$

$$C1a: c(X) \vee \neg s(X) \vee t(X)$$

forma a clausole

$$C1b: \neg c(X) \vee s(X)$$

$$C1c: \neg c(X) \vee \neg t(X)$$

2. Paolo sale sull'autobus;

$$C2: s(\text{paolo}).$$

3. Paolo ha un valido titolo di viaggio;

$$C3: t(\text{paolo}).$$

Query: Paolo non deve comprare alcun biglietto dal guidatore

$$\neg c(\text{paolo}).$$

GNeg:  $c(\text{paolo})$ .

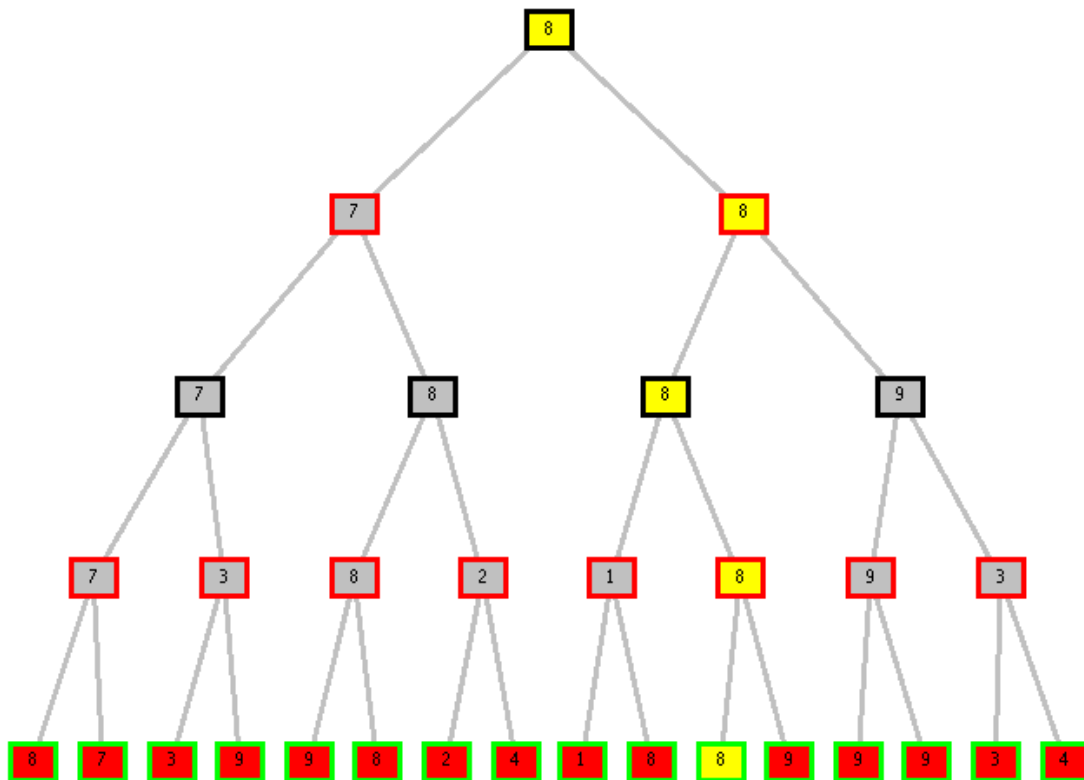
Risoluzione:

$$C4: \text{GNeg} \cup C1c: \neg t(\text{paolo})$$

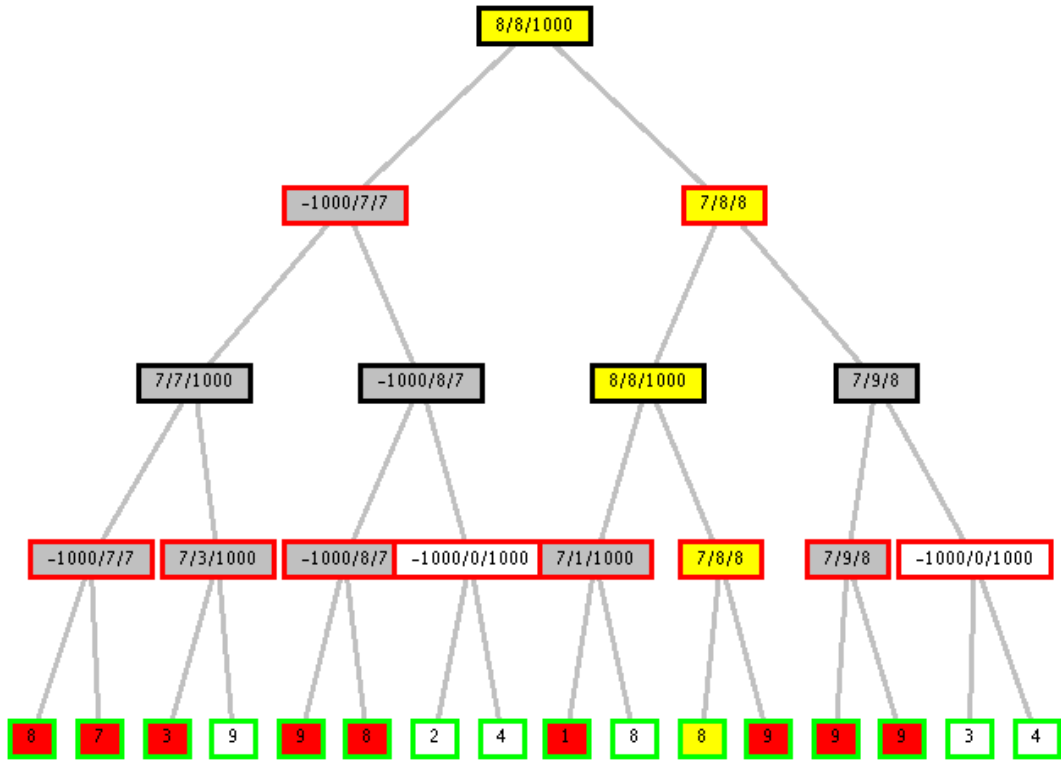
C5:  $C4 \cup C3$ : clausola vuota, contraddizione.

## Esercizio 2

Min-Max:

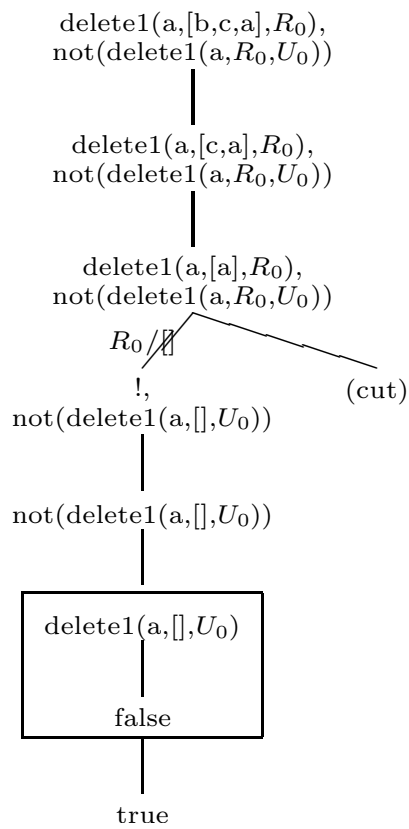


Alfa-beta:



I nodi che portano alla soluzione sono in giallo, quelli tagliati in bianco.

### Esercizio 3



## Esercizio 4

```
member(X, [X|_]) :- !.  
member(X, [_|T]) :- member(X, T).
```

```
memberlist(X, [H|_]) :- member(X, H), !.  
memberlist(X, [_|T]) :- memberlist(X, T).
```

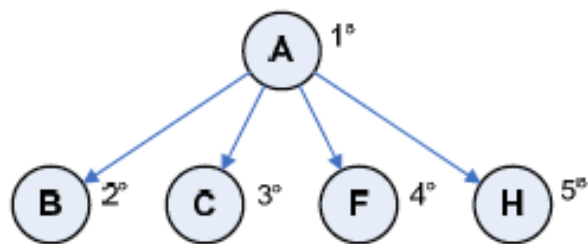
## Esercizio 5

Riporto gli alberi frutto della ricerca ad approfondimento iterativo:

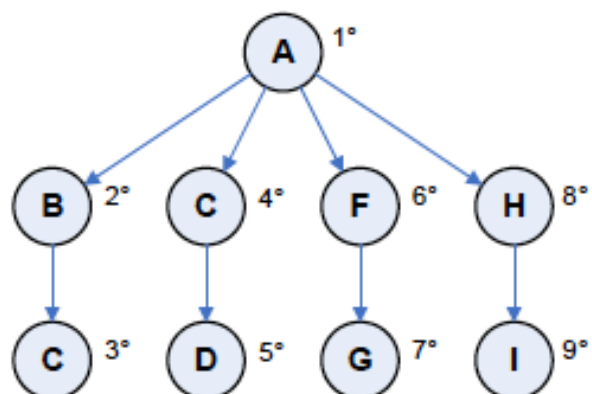
$l=0$



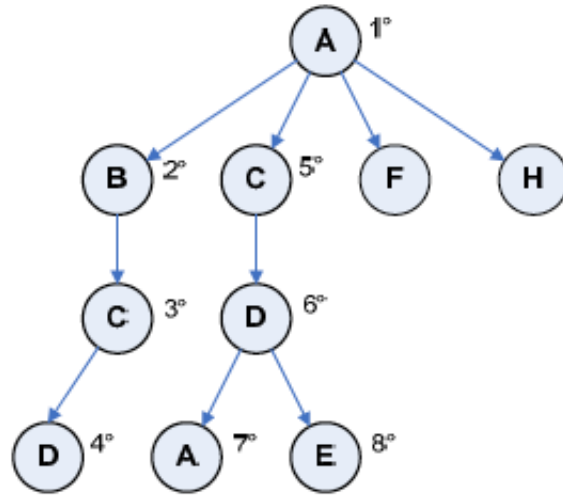
$l=1$



$l=2$



$l=3$



La soluzione trovata è A-C-D-E ed è ottima perché la ricerca ad approfondimento iterativo è ottima con costi di arco uguali a 1.

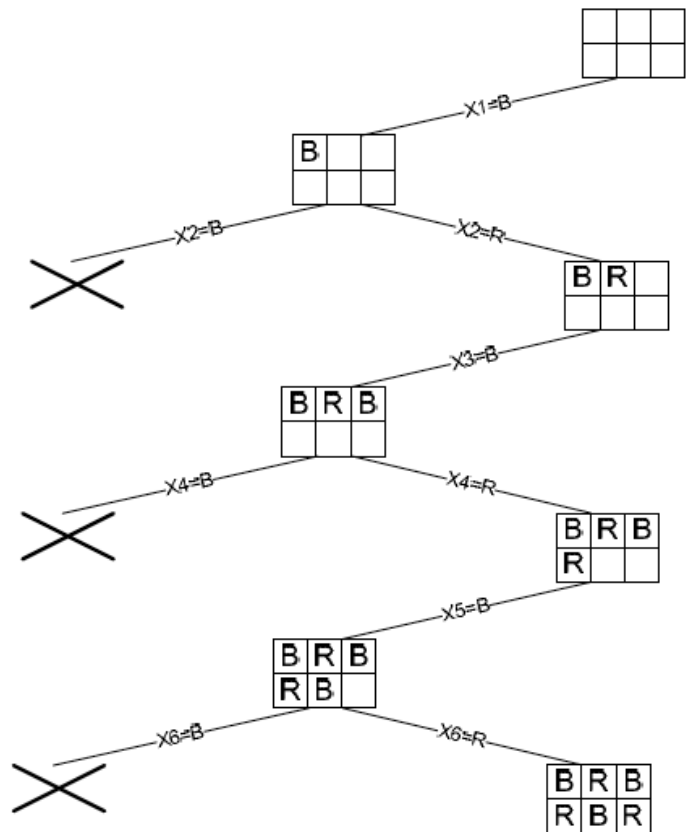
### Esercizio 6

Variabili e domini:  
 $X_1, X_2, X_3, X_4, X_5, X_6 :: [B, R]$

Vincoli:

- $X_1 \neq X_2$
- $X_1 \neq X_4$
- $X_2 \neq X_3$
- $X_2 \neq X_5$
- $X_3 \neq X_6$
- $X_4 \neq X_5$
- $X_5 \neq X_6$

Applicando arc consistency i domini restano invariati. Albero di ricerca:



### Esercizio 7

Vedi slide.