

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

**10 Luglio 2008 (Tempo a disposizione 2h ; su 32 punti)**

**Esercizio 1 (punti 7)**

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

1. Se Elisa ama qualcuno, allora Elisa ama Massimo
2. Massimo ama Elisa o Sabrina
3. Tutti amano Massimo e Massimo non ama nessuno
4. Tutti amano Elisa se e solo se tutti amano Massimo
5. Qualcuno ama Elisa e qualcuno non ama Massimo

Le si trasformi tutte in clausole. Si dimostri poi tramite risoluzione che assumendo come assiomi le clausole derivanti dalla formalizzazione delle frasi 3 e 4, si dimostra la query “tutti amano Elisa”.

**Esercizio 2 (punti 6)**

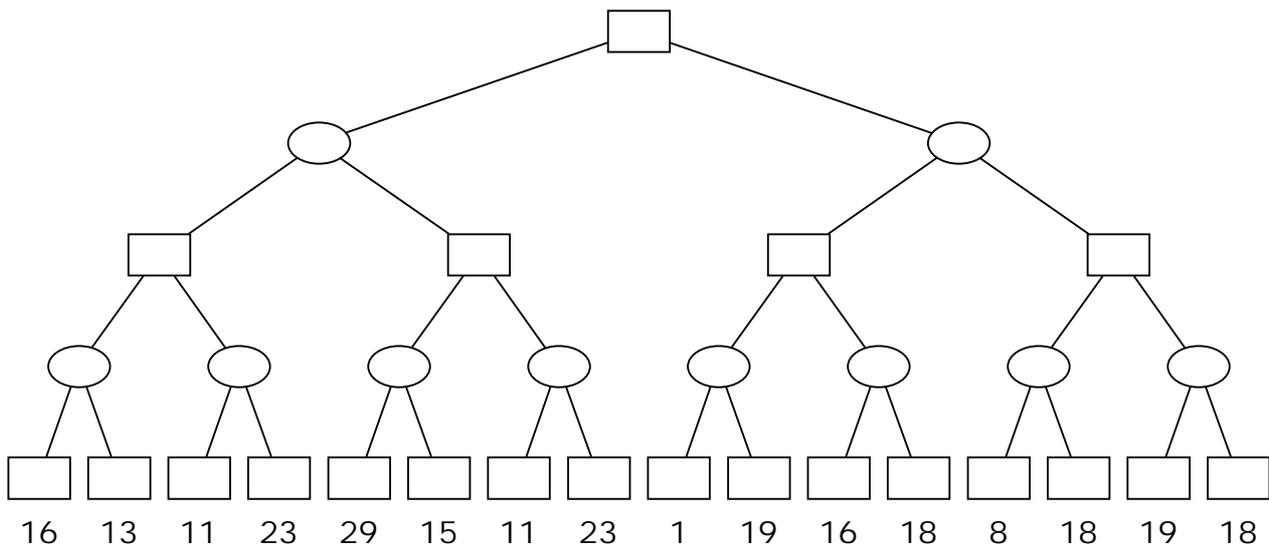
Il seguente predicato, date due liste ordinate, deve fornire la lista ordinata che contiene gli elementi di entrambe:

```
merge([], L, L) :- !.  
merge(L, [], L) :- !.  
merge([A|T], [B|L], [A|M]) :- A < B, !, merge(T, [B|L], M).  
merge([A|T], [B|L], [B|M]) :- merge([A|T], L, M).
```

Si mostri l'albero SLD relativo all'invocazione  $merge([1, 4], [2, 3], [2|L])$  e si dica qual è la risposta calcolata. La risposta è corretta? Nel caso non lo sia, si corregga il programma.

**Esercizio 3 (punti 5)**

Si consideri il seguente albero di gioco, dove i punteggi sono per il primo giocatore (Max). Si mostri come l'algoritmo min-max risolve il problema e qual è la scelta fatta da Max. Si mostrino poi i tagli alfa-beta.

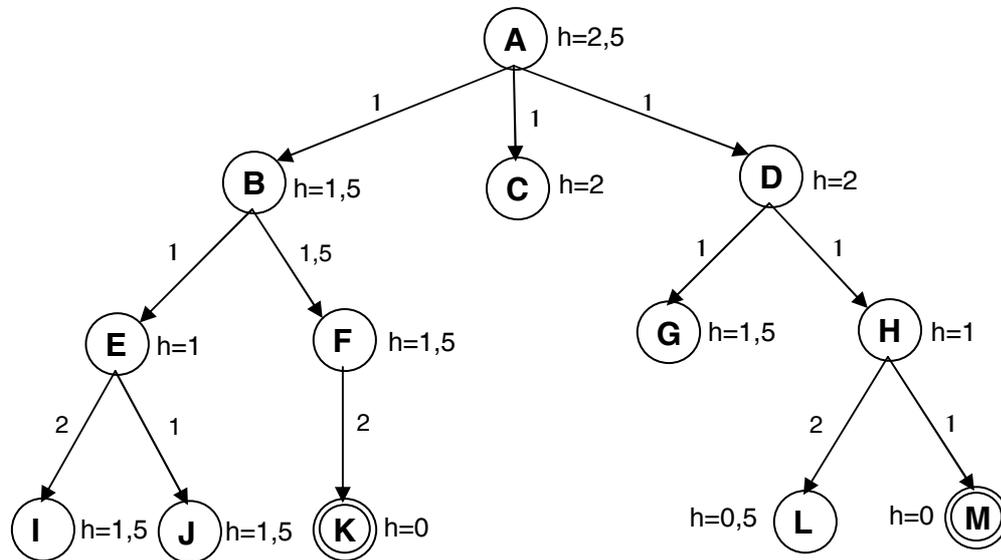


**Esercizio 4 (punti 7)**

Nel seguente spazio di ricerca i numeri sugli archi rappresentano il costo delle azioni, i numeri vicini ai nodi il valore della funzione di valutazione euristica  $h$  e gli stati con doppio cerchio rappresentano stati goal.

- a. Per quello che è dato di vedere, la funzione di valutazione euristica è *ammissibile*? Motivare.

- b. Per quello che è dato di vedere, la funzione di valutazione euristica è *monotona/consistente*?  
 Si riporti la definizione di consistenza di una funzione euristica e si motivi la risposta .
- c. Per ciascuno dei seguenti algoritmi di ricerca si listino i nodi in ordine di espansione e si dica qual è il nodo goal restituito.
- i. Costo uniforme      ii. Approfondimento iterativo      iii. Algoritmo A\*



**Esercizio 5 (punti 5)**

Si scriva un programma Prolog `substitute(X, Y, Xs, Ys)` che produce una lista `Ys` ottenuta sostituendo nella lista `Xs` tutte le occorrenze di `X` con `Y`.

Esempi:

```
?-substitute(a, f(a), [a,b,a,d], L).
yes L=[f(a),b, f(a),d]
```

```
?-substitute(1, g(Z), [1,12,3,1], L).
yes L=[g(Z),12,3,g(Z)]
```

**Esercizio 6 (punti 2)**

Si presentino in modo sintetico le strategie di ricerca backward, forward e combinata (backward e forward assieme).

## SOLUZIONE

### Esercizio 1

1. Se Elisa ama qualcuno, allora Elisa ama Massimo
2. Massimo ama Elisa o Sabrina
3. Tutti amano Massimo e Massimo non ama nessuno
4. Tutti amano Elisa se e solo se tutti amano Massimo
5. Qualcuno ama Elisa e qualcuno non ama Massimo

Logica dei predicati:

1.  $(\exists X \text{ ama}(e,X)) \rightarrow \text{ama}(e,m)$
2.  $\text{ama}(m,e) \vee \text{ama}(m,s)$
3.  $\forall X \text{ ama}(X,m) \wedge \sim \exists Y \text{ ama}(m,Y)$
4.  $(\forall X \text{ ama}(X,e)) \leftrightarrow (\forall Y \text{ ama}(Y,m))$
5.  $\exists X \text{ ama}(X,e) \wedge \exists Y \sim \text{ama}(Y,m)$

Clausole:

1.  $\sim \text{ama}(e,X) \vee \text{ama}(e,m)$
2.  $\text{ama}(m,e) \vee \text{ama}(m,s)$
3.  $\forall X \text{ ama}(X,m) \wedge \sim \exists Y \text{ ama}(m,Y)$   
 $\forall X \text{ ama}(X,m) \wedge \forall Y \sim \text{ama}(m,Y)$   
 $\forall X \forall Y \text{ ama}(X,m) \wedge \sim \text{ama}(m,Y)$   
3.1  $\text{ama}(X,m)$   
3.2  $\sim \text{ama}(m,Y)$
4.  $(\forall X \text{ ama}(X,e)) \leftrightarrow (\forall Y \text{ ama}(Y,m))$   
 $((\forall X \text{ ama}(X,e)) \rightarrow (\forall Y \text{ ama}(Y,m))) \wedge ((\forall Y \text{ ama}(Y,m)) \rightarrow (\forall X \text{ ama}(X,e)))$   
 $((\exists X \sim \text{ama}(X,e)) \vee (\forall Y \text{ ama}(Y,m))) \wedge ((\exists Y \text{ ama}(Y,m)) \vee (\forall X \text{ ama}(X,e)))$   
*Introducendo due costanti di Skolem per le variabili esistenziali e separando la congiunzione si ottengono le clausole:*  
4.1  $\sim \text{ama}(p,e) \vee \text{ama}(Y,m)$   
4.2  $\sim \text{ama}(q,m) \vee \text{ama}(X,e)$
5.  $\exists X \text{ ama}(X,e) \wedge \exists Y \sim \text{ama}(Y,m)$  introducendo due costanti di Skolem  
5.1  $\text{ama}(e,c1)$   
5.2  $\sim \text{ama}(c2,m)$

Assunte 3.1,3.2,4.1 e 4.2 come assiomi, con clausola goal ottenuta negando la query:

$\sim \forall X \text{ ama}(X,e)$

$\exists X \sim \text{ama}(X,e)$  anche qui introducendo una costante di Skolem si ottiene la clausola:

6.  $\sim \text{ama}(d,e)$

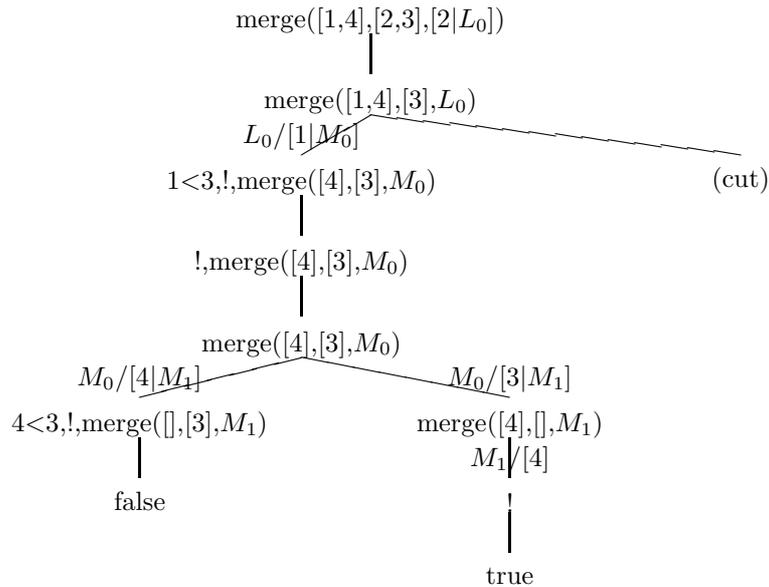
Applicando la risoluzione:

da 3.1 e 4.2 ottengo

7.  $\text{ama}(X,e)$

Da 7 e 6 ottengo la contraddizione.

## Esercizio 2



La risposta calcolata è  $L0/[1,3,4]$ , quindi la lista complessiva fornita è  $[2,1,3,4]$ , che non è ordinata. Il problema è nella terza clausola:

```
merge([A|T],[B|L],[A|M]):- A<B,!, merge(T,[B|L],M).
```

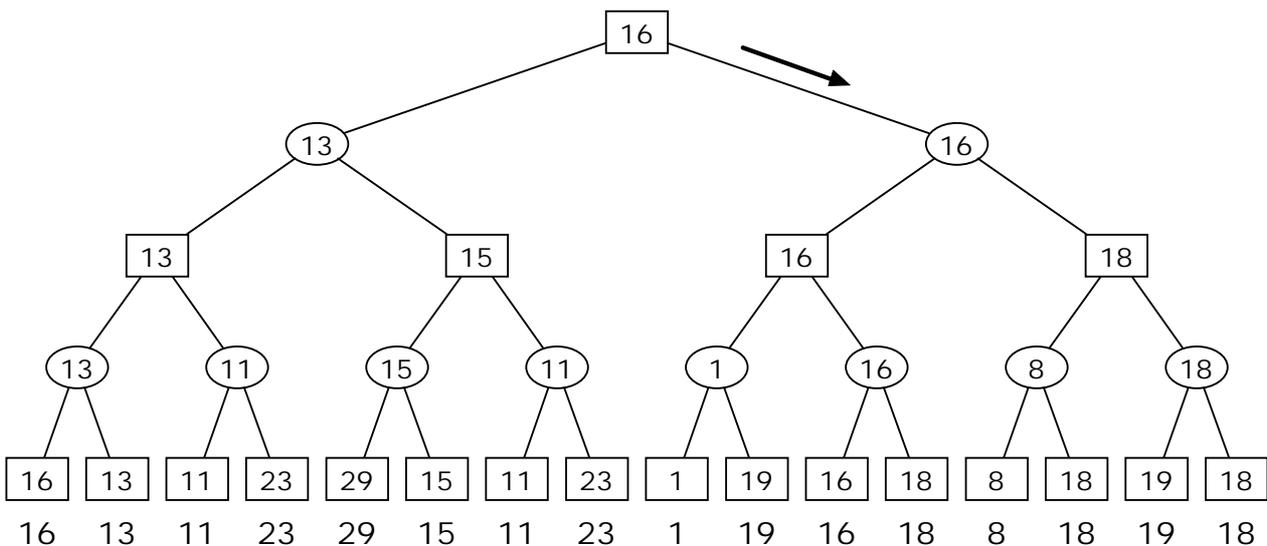
che fornisce il risultato nel terzo parametro prima che venga eseguito il cut, per cui nella prima invocazione la clausola non viene selezionata.

Un modo per correggere l'errore è quello di spostare l'unificazione dopo il cut:

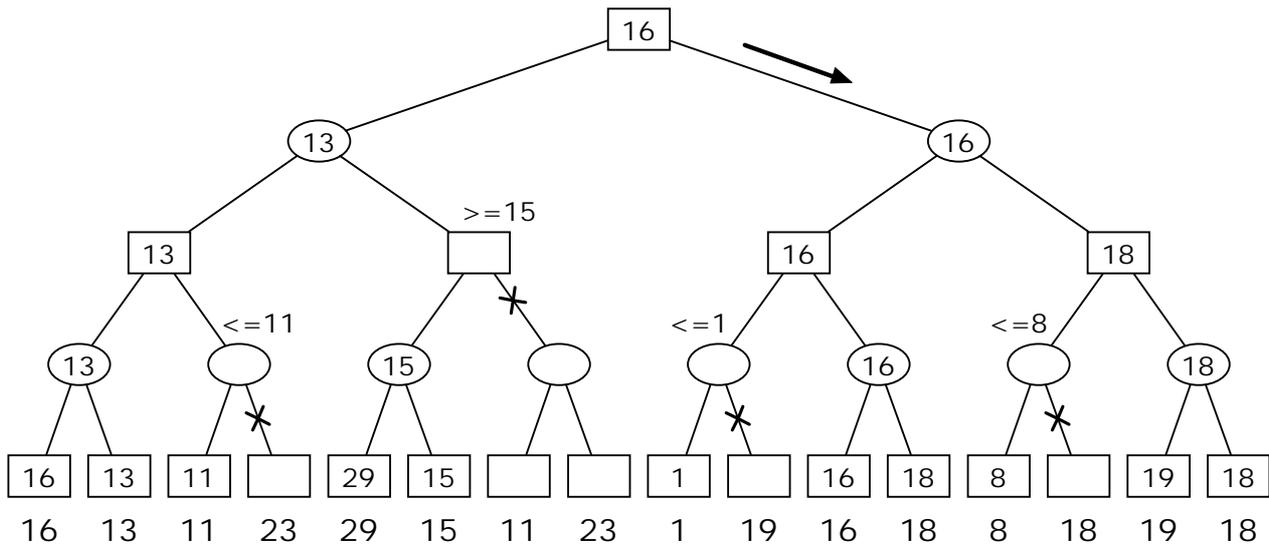
```
merge([A|T],[B|L],Lout):- A<B,!,Lout=[A|M], merge(T,[B|L],M).
```

## Esercizio 3

Min-max:



alfa-beta:



### Esercizio 4

- Sì. L'euristica è ammissibile per quello che è dato di vedere sugli stati illustrati in figura. Non succede mai che lo stato obiettivo sia raggiungibile con un costo minore di quanto stimato. In particolare:  $h(A) \leq 3$ ;  $h(B) \leq 3.5$ ;  $h(F) \leq 2$ ;  $h(D) \leq 2$ ;  $h(H) \leq 1$ . Dagli altri stati il goal non è raggiungibile.
- Sì. L'euristica è monotona per quello che è dato di vedere sugli stati illustrati in figura. In particolare, la definizione di monotonia richiede che se  $n'$  è  $\text{succ}(n)$  allora  $h(n) - h(n') \leq \text{costo}(n, n')$ . In concreto questo si verifica sempre nel nostro esempio:  
 $h(A) - h(B) \leq 1$ ;  $h(B) - h(E) \leq 1$ ;  $h(E) - h(I) \leq 2$ ;  $h(E) - h(J) \leq 1$ ;  $h(B) - h(F) \leq 1.5$ ;  
 $h(F) - h(K) \leq 2$ ;  $h(A) - h(C) \leq 1$ ;  $h(A) - h(D) \leq 1$ ;  $h(D) - h(G) \leq 1$ ;  $h(D) - h(H) \leq 1$ ;  
 $h(H) - h(L) \leq 2$ ;  $h(H) - h(M) \leq 2$ .
- Costo uniforme.  
Ordine di espansione dei nodi: A, B, C, D, E, G, H, F, J, M  
Goal trovato: M
  - Approfondimento iterativo.  
Ordine di espansione dei nodi: A, | A, B, C, D, | A, B, E, F, C, D, G, H, | A, B, E, I, J, F, K  
Goal trovato: K
  - Algoritmo A\*  
Ordine di espansione dei nodi: A, B, C, D, E, H, M  
Goal trovato: M

### Esercizio 5

```
% Programma per la sostituzione nella lista Xs di tutte le occorrenze
% di X con Y e che produce Ys
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
/* substitute(X,Y,Xs,Ys) is true if the list Ys is the result of */
/* substituting Y for all occurrences of X in the list Xs.      */
```

```
substitute(X,Y,[],[]).
substitute(X,Y,[X|Xs],[Y|Ys):-!, substitute(X,Y,Xs,Ys).
substitute(X,Y,[Z|Xs],[Z|Ys):- substitute(X,Y,Xs,Ys).
```