

FONDAMENTI DI INTELLIGENZA ARTIFICIALE

22 Dicembre 2009 – Tempo a disposizione 2h – Risultato 32/32 punti

Esercizio 1 (punti 6)

a) Si modellino in logica dei predicati del I ordine le seguenti frasi:

Ogni persona che Maria ama è un campione di football

Qualsiasi studente che non passa gli esami non gioca

Giovanni è uno studente

Qualsiasi studente che non studia non passa gli esami

Chiunque non giochi non è un campione del football

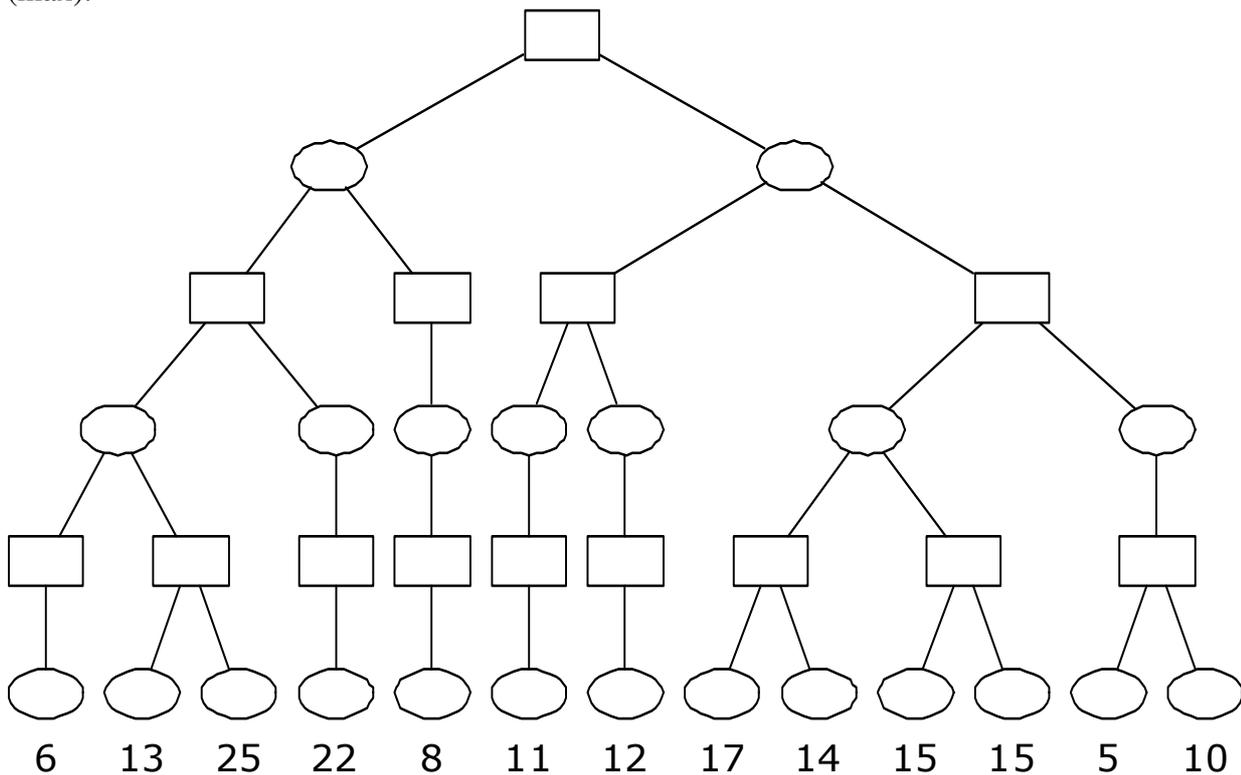
Giovanni non studia

b) Le si trasformi in logica a clausole e si dimostri, applicando il principio di metodo di risoluzione, che:

Maria non ama Giovanni

Esercizio 2 (punti 5)

Sia dato il seguente albero di gioco, dove i punteggi sono tutti dal punto di vista del primo giocatore (max):



Si mostri come l'algoritmo min-max risolve il problema. Si mostrino poi i tagli alfa-beta.

Esercizio 3 (punti 7)

Si consideri il seguente programma Prolog:

```
between(L,U,L).
```

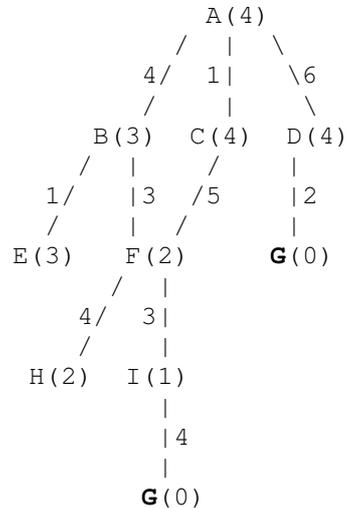
```
between(L,U,X):- L1 is L+1, L1 =<U, between(L1,U,X).
```

```
csp:- between(1,3,X), between(1,3,Y), X>Y,!.
```

Si disegni l'albero SLD relativo al goal `csp`.

Esercizio 4 (punti 6)

Si consideri il seguente spazio di ricerca con stato iniziale A e stato goal G. Un arco da stato i a stato j indica che un'azione può trasformare i in j. Le etichette intiere degli archi indicano il costo di tale azione. A fianco di ciascun stato, tra parentesi, è indicata la stima euristica della distanza dallo stato allo stato goal G.



Si indichi l'espansione dei nodi secondo la strategia best-first e la strategia A*. A parità del valore della funzione di valutazione ($f(n)$), si espandano i nodi in ordine lessicografico. L'euristica proposta è ammissibile? Cosa comporta l'ammissibilità sull'algoritmo A* in termini di ottimalità della soluzione?

Esercizio 5 (punti 5)

Si scriva un predicato Prolog `split(L1,N,L2,L3)` che data una lista di interi $L1$ ed un numero intero N , crei due liste, inserendo nella lista $L2$ i numeri minori del numero assegnato e nella lista $L3$ i numeri maggiori del numero assegnato. I numeri uguali al numero assegnato non devono essere inseriti in alcuna delle due liste. Il predicato può essere invocato con tutti e 4 gli argomenti in ingresso, oppure solo con i primi due in ingresso.

Esempio:

```
?-split([1,2,7,3,4,5,6,9,6,9],4,[1,2,3],[7,6,9,6,9]).
Yes
?-split([1,2,7,3,4,5,6,9,6,9],4,A,B).
Yes      A=[1,2,3]      B=[7,6,9,6,9]
```

Esercizio 6 (punti 3)

Si presentino le tecniche di consistenza e si descriva l'algoritmo di arc-consistenza in pseudocodice, dandone poi una esemplificazione di funzionamento su un grafo (rete) a vincoli.

SOLUZIONE:

Esercizio 1

$\forall X (\text{loves}(\text{mary}, X) \rightarrow \text{star}(X))$

$\forall X (\text{student}(X) \text{ and } \neg \text{pass}(X) \rightarrow \neg \text{play}(X))$

$\text{student}(\text{john})$

$\forall X (\text{student}(X) \text{ and } \neg \text{study}(X) \rightarrow \neg \text{pass}(X))$

$\forall X (\neg \text{play}(X) \rightarrow \neg \text{star}(X))$

$\neg \text{study}(\text{john})$

Query:

$\neg \text{loves}(\text{mary}, \text{john})$

Trasformazione in clausole:

1. $\neg \text{loves}(\text{mary}, X) \text{ or } \text{star}(X)$
2. $\neg \text{student}(X) \text{ or } \text{pass}(X) \text{ or } \neg \text{play}(X)$
3. $\text{student}(\text{john})$
4. $\neg \text{student}(X) \text{ or } \text{study}(X) \text{ or } \neg \text{pass}(X)$
5. $\text{play}(X) \text{ or } \neg \text{star}(X)$
6. $\neg \text{study}(\text{john})$

Goal:

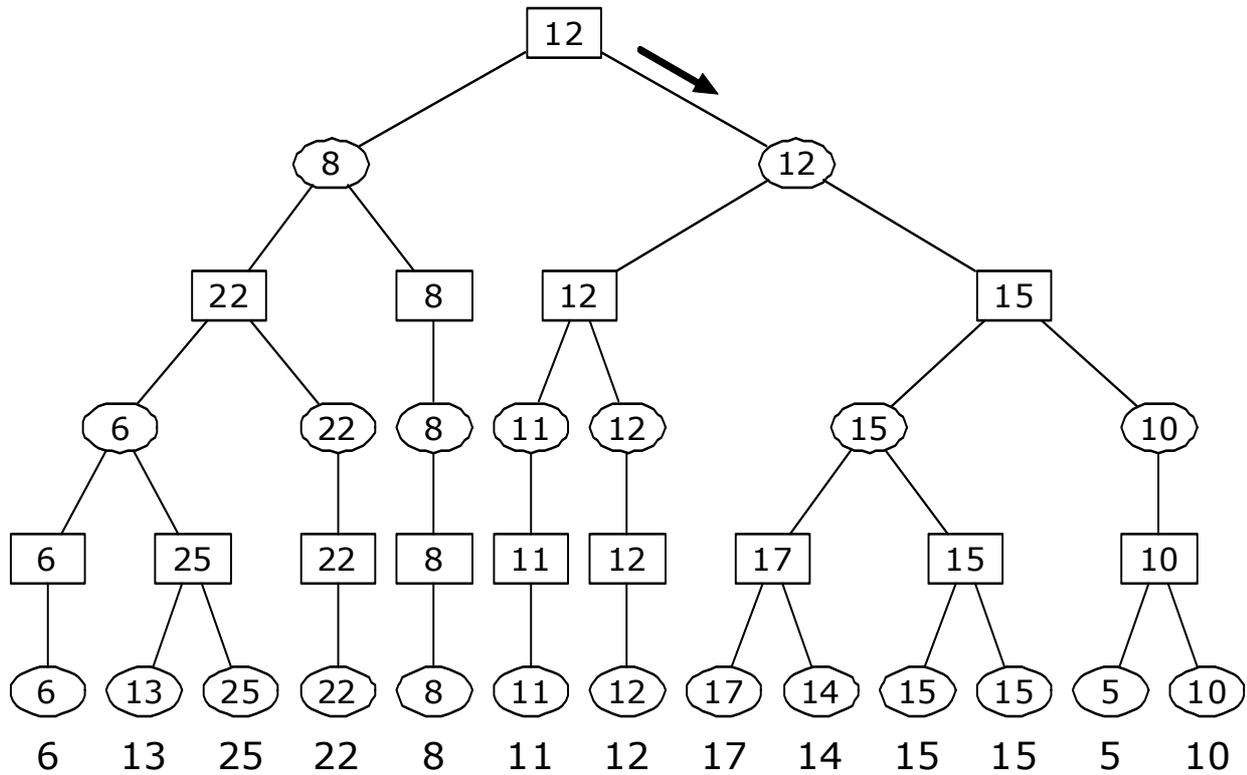
7. $\text{loves}(\text{mary}, \text{john})$

Risoluzione:

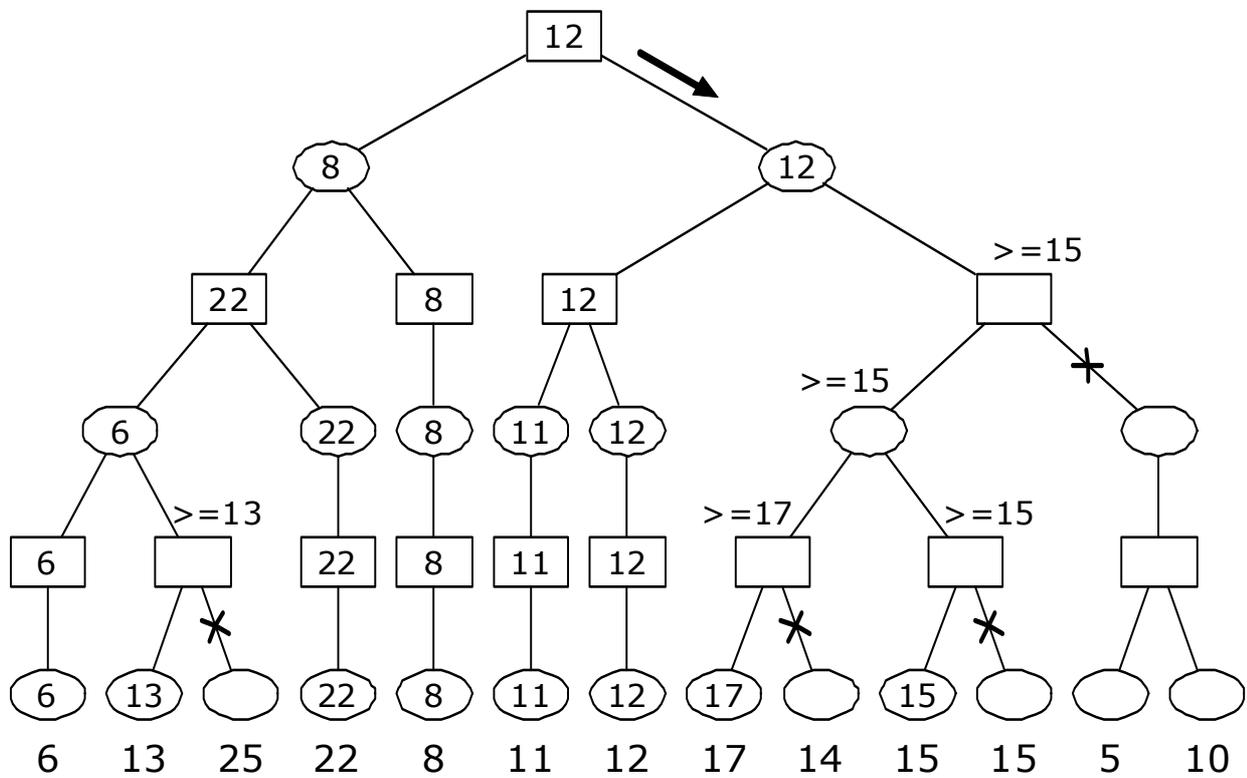
8. $\text{pass}(\text{john}) \text{ or } \neg \text{play}(\text{john})$ da 2+3
9. $\text{study}(\text{john}) \text{ or } \neg \text{pass}(\text{john})$ da 3+4
10. $\neg \text{pass}(\text{john})$ da 9+6
11. $\neg \text{play}(\text{john})$ da 10+8
12. $\neg \text{star}(\text{john})$ da 11+5
13. $\neg \text{loves}(\text{mary}, \text{john})$ da 12+1
14. [] da 13+7

Esercizio 2

min-max:



alfa-beta:



Strategia A*:

A lista nodi aperti: [C(5), B(7), D(10)]

C lista nodi aperti: [B(7), F(8), D(10)]

F lista nodi aperti: [E(8), F(8), D(10)] - F già visitato (il nuovo valore sarebbe 9, quindi non aggiorna)

B lista nodi aperti: [F(8), D(10)]

E lista nodi aperti: [D(10), I(10), H(12)]

D lista nodi aperti: [G(8), I(10), H(12)]

G obiettivo – costo strada percorsa A, D, G: 8

Esercizio 5

```
split([],N, [], []).
```

```
split([A|R],N, [A|Ra],Rb) :- A<N, !, split(R,N, Ra,Rb).
```

```
split([A|R],N, Ra, [A|Rb] ) :- A>N, !, split(R,N, Ra,Rb).
```

```
split([A|R],N, Ra, Rb ) :- split(R,N, Ra,Rb).
```