

FONDAMENTI DI INTELLIGENZA ARTIFICIALE (8 CFU)

10 Settembre 2015 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (6 punti)

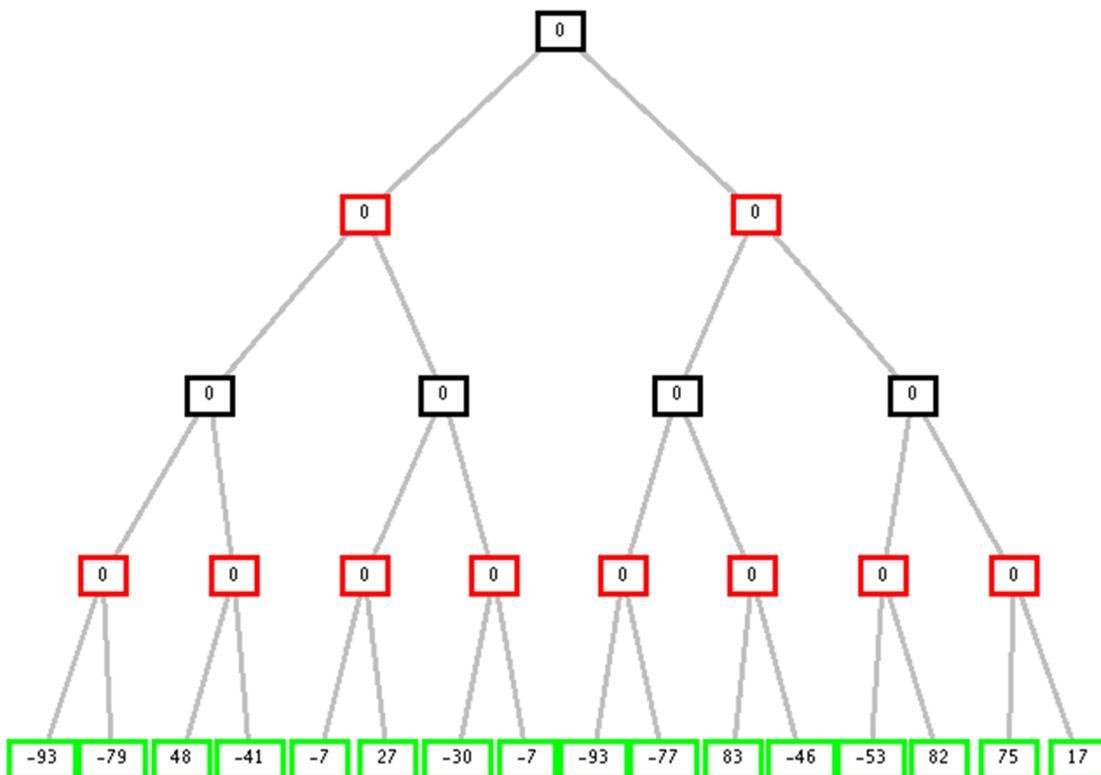
Si rappresentino le seguenti affermazioni in logica dei predicati del I ordine:

- Ogni innamorato è amato da un altro innamorato.
- Dati tre innamorati X, Y, Z, se X ama Y e Y ama Z allora X non ama Z.
- Un innamorato non può amare se stesso.
- Anna, Bebo e Carla sono innamorati, e Anna ama Bebo, ma Bebo ama Carla.

Si esprimano le formule scritte sopra in forma normale congiuntiva, usando i predicati unario innamorato(X) e binario ama(X, Y) (con significato X ama Y). Si dimostri, tramite risoluzione, che Anna non ama Carla.

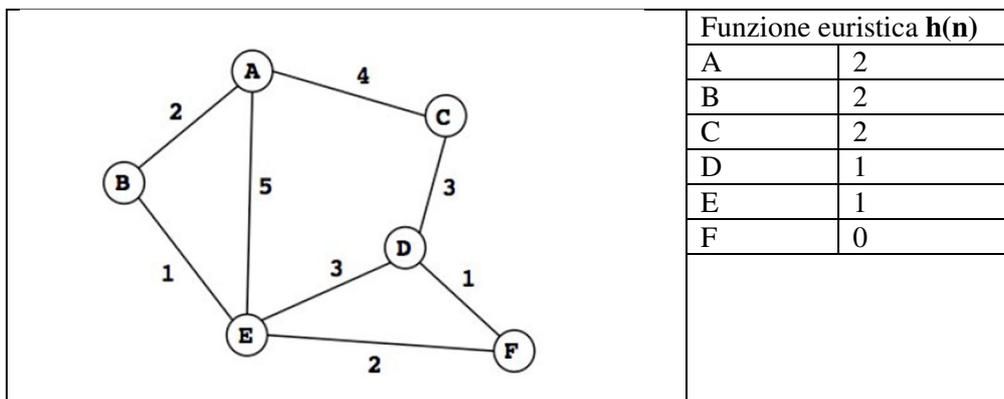
Esercizio 2 (4 punti)

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 (6 punti)

Si consideri il seguente grafo, dove A è il nodo iniziale e F il nodo goal, e il numero associato agli archi è il costo dell'operatore per andare dal primo al secondo estremo dell'arco. Gli archi sono da intendersi orientati con gli estremi in ordine alfabetico (quindi si può andare da A a B, ma non da B ad A, e neppure da E a D, ecc.).



Si usino come euristica $h(n)$ i valori specificati nella tabellina posta a fianco della figura.

- L'euristica proposta è ammissibile?
- Si mostri come è trovata la soluzione applicando la ricerca A^* (e qual è il costo della soluzione).

Esercizio 4 (5 punti)

Si scriva un predicato Prolog `greatest(L, X)` che data in ingresso la lista (non vuota) di valori interi positivi, ne calcola il valore più grande in lista. Esempio:

```
?- greatest([1, 10, 3, 5, 9, 7], X).
Yes, X=10.
```

Esercizio 5 (6 punti)

Si consideri il seguente programma Prolog, che data una lista in ingresso come primo argomento, produce una seconda lista che contiene gli elementi della prima, ma senza duplicazioni:

```
nodupl([], []).
nodupl([X|Xs], Ys):-
    member(X, Xs),
    nodupl(Xs, Ys).
nodupl([X|Xs], [X|Ys]):-
    \+(member(X, Xs)),
    nodupl(Xs, Ys).
```

Si mostri l'albero SLD-NF che si ottiene per il goal Prolog seguente, **espandendolo fino alla prima soluzione**:

```
:-nodupl([1, 2, 1], Y).
```

Esercizio 6 (3 punti)

Si descrivano gli algoritmi per CSP che utilizzano attivamente i vincoli nel corso della ricerca. Si descrivano in particolare gli algoritmi di propagazione visti (dal Forward Checking al Partial e Full Look Ahead), discutendone le differenze sull'esempio seguente:

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

$A < B < C$

dopo il passo di labeling $A=1$. Mostrare come cambiano i domini di B e C nei tre casi (FC, PLA e FLA).

Esercizio 7 (2 punti)

Spiegare brevemente in cosa consistono gli algoritmi genetici e quando è bene applicarli.

FONDAMENTI DI INTELLIGENZA ARTIFICIALE

11 Settembre 2015 – Soluzioni

Esercizio 1

1. Ogni innamorato è amato da un altro innamorato.

$$\forall X \exists Y \text{ innamorato}(X) \Rightarrow \text{innamorato}(Y) \wedge \text{ama}(Y,X).$$

2. Dati tre innamorati X, Y, Z, se X ama Y e Y ama Z allora X non ama Z.

$$\forall X \forall Y \forall Z \text{ innamorato}(X) \wedge \text{innamorato}(Y) \wedge \text{innamorato}(Z) \wedge \text{ama}(X,Y) \wedge \text{ama}(Y,Z) \Rightarrow \neg \text{ama}(X,Z).$$

3. Un innamorato non può amare sè stesso.

$$\forall X \text{ innamorato}(X) \Rightarrow \neg \text{ama}(X,X)$$

4. a,b,c sono innamorati, e a ama b, e b ama c.

$$\text{innamorato}(a) \wedge \text{innamorato}(b) \wedge \text{innamorato}(c) \wedge \text{ama}(a,b) \wedge \text{ama}(b,c)$$

5. Query: $\neg \text{ama}(a,c)$

Forma normale congiuntiva

1a. $\neg \text{innamorato}(X) \vee \text{innamorato}(f(X)).$

1b. $\neg \text{innamorato}(X) \vee \text{ama}(f(X), X).$

2. $\neg \text{innamorato}(X) \vee \neg \text{innamorato}(Y) \vee \neg \text{innamorato}(Z) \vee \neg \text{ama}(X,Y) \vee \neg \text{ama}(Y,Z) \vee \neg \text{ama}(X,Z).$

3. $\neg \text{innamorato}(X) \vee \neg \text{ama}(X,X).$

4a. $\text{innamorato}(a).$

4b. $\text{innamorato}(b).$

4c. $\text{innamorato}(c).$

4d. $\text{ama}(a,b).$

4e. $\text{ama}(b,c).$

GNeg: $\text{ama}(a,c).$

Risoluzione

5: GNeg+2: $\neg \text{innamorato}(a) \vee \neg \text{innamorato}(Y) \vee \neg \text{innamorato}(c) \vee \neg \text{ama}(a,Y) \vee \neg \text{ama}(Y,c).$

6: 5+4a: $\neg \text{innamorato}(Y) \vee \neg \text{innamorato}(c) \vee \neg \text{ama}(a,Y) \vee \neg \text{ama}(Y,c).$

7: 6+4c: $\neg \text{innamorato}(Y) \vee \neg \text{ama}(a,Y) \vee \neg \text{ama}(Y,c).$

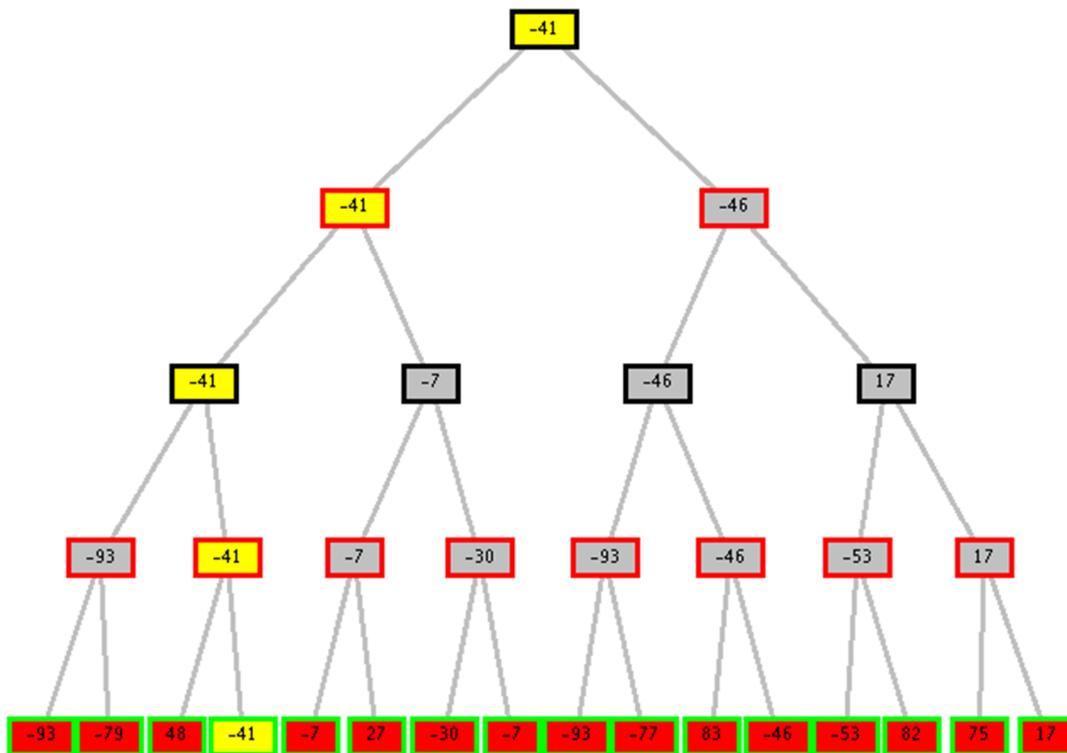
8: 7+4d: $\neg \text{innamorato}(b) \vee \neg \text{ama}(b,c).$

9: 8+4b: $\neg \text{ama}(b,c).$

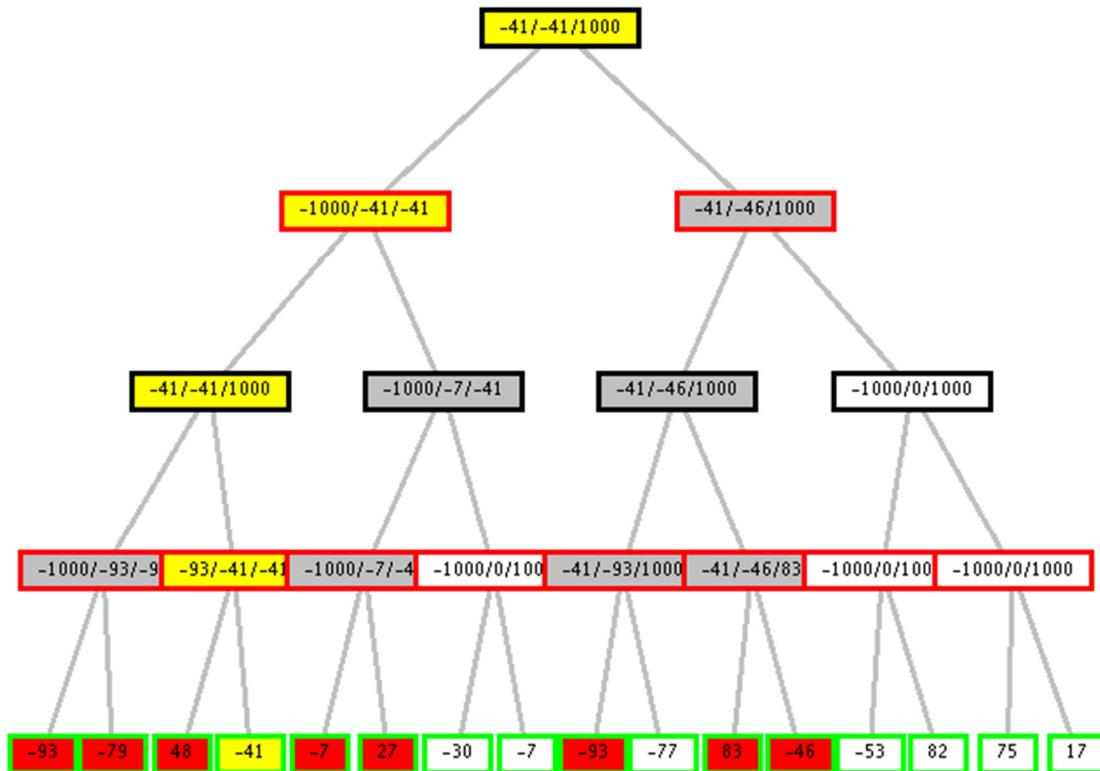
10: 9+4e: clausola vuota.

Esercizio 2

Min-Max:



Alfa-Beta:



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

Esercizio 3 (punti)

a) L'euristica è ammissibile: si può verificare direttamente che è sempre ottimista rispetto ai costi per raggiungere l'obiettivo.

b) Nella ricerca A*, i nodi sono espansi come segue (si riporta il valore dell'euristica):

- Seleziono il nodo A. Non essendo il goal, lo espando e genero:
B: $[f(B) = g(B) + h(B) = 2 + 2 = 4]$
E: $[f(E) = g(E) + h(E) = 5 + 1 = 6]$
C: $[f(C) = g(C) + h(C) = 4 + 2 = 6]$
- Seleziono il nodo B. non essendo il goal, lo espando.
Ottengo un nuovo percorso per il nodo E, la cui valutazione è:
E: $[f(E) = g(E) + h(E) = 3 + 1 = 4]$
- Seleziono il nodo E. non essendo il goal, lo espando e ottengo il nodo:
F: $[f(F) = g(F) + h(F) = 5 + 0 = 5]$
- Seleziono il nodo F. Tale nodo è il GOAL.

Il costo della soluzione trovata da A* è $[f(F) = 5]$

Esercizio 4 (punti)

Versione ricorsiva:

```
greatest([X], X).  
greatest([H|T], H) :- greatest(T, H1), H1 < H.  
greatest([H|T], H1) :- greatest(T, H1), H1 >= H.
```

Versione ricorsiva tail:

```
greatest(L, X) :- greatest_aux(L, 0, X).  
greatest_aux([], X, X).  
greatest_aux([H|T], Temp, X) :- H > Temp, !, greatest_aux(T, H, X).  
greatest_aux([_|T], Temp, X) :- greatest_aux(T, Temp, X).
```

