FONDAMENTI DI INTELLIGENZA ARTIFICIALE

10 Settembre 2024 - Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (6 punti)

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

1. Laura è una sportiva
2. Tutti gli sportivi sono atleti
3. Mario è un podista
4. Tutti i podisti sono sportivi
5. Se un atleta batte un altro atleta, allora il secondo non è più forte del primo.
6. Laura batte Mario

Si dimostri, per refutazione, tramite l'applicazione della risoluzione, che *Mario non è più forte di Laura*.

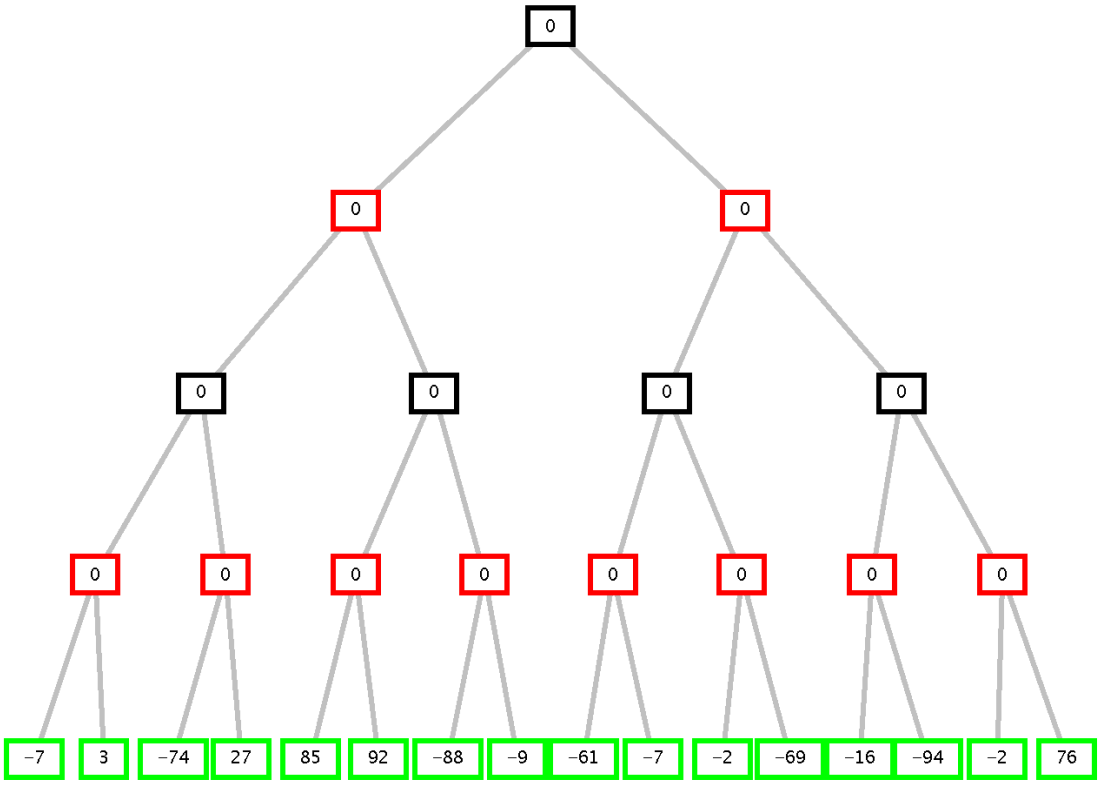
Si usi il seguente vocabolario:

predicati: **sportivo**(X), **atleta**(X), **podista**(X), **piu_forte_di**(X,Y) con significato "X è più forte di Y", **batte**(X,Y) cioè "X batte Y";

costanti: laura, mario.

Esercizio 2 (5 punti)

Si consideri il seguente albero di gioco in cui il primo giocatore è MAX.



a) Si indichi come l'algoritmo min-max risolve il problema indicando il valore con cui viene etichettato il nodo iniziale e la mossa selezionata dal primo giocatore (arco a sinistra o a destra).

b) Si mostrino poi i tagli che l'algoritmo alfa-beta consente, indicando gli archi che verranno tagliati.

Esercizio 3 (5 punti)

Si scriva un predicato Prolog **selectList**(L1,M,L2,Result) che date due liste di interi L1 e L2 (quindi tutti gli elementi sono *ground*) e un numero intero M, crei una lista Result composta dagli elementi di L1 > M e che non appartengono a L2. Si può utilizzare , nella soluzione, il predicato **member**/2 come definito nel testo dell'esercizio che segue.

Esempi:

?-selectList([1,2,7,3,4,5],3,[1,2,3,5], [7,4]).

Yes

?-selectList([1,2,7,3,4,5],3,[1,2,3,5], X).

Yes X= [7,4].

Esercizio 4 (5 punti)

Sia dato il predicato Prolog **delunicoLista(L1,L2)**, che, data una lista **L1** di numeri interi, restituisce in **L2** la lista ottenuta eliminando da **L1** gli elementi che non sono seguiti da almeno un altro elemento uguale in **L1**. Se **L1** è la lista vuota verrà restituita in uscita la lista vuota. Sia dato anche il predicato **member**:

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

delunicoLista([],[]).
delunicoLista([H|T],[H|Y]):-member(H,T),!, delunicoLista(T,Y).
delunicoLista([_|T],Y):- delunicoLista(T,Y).
```

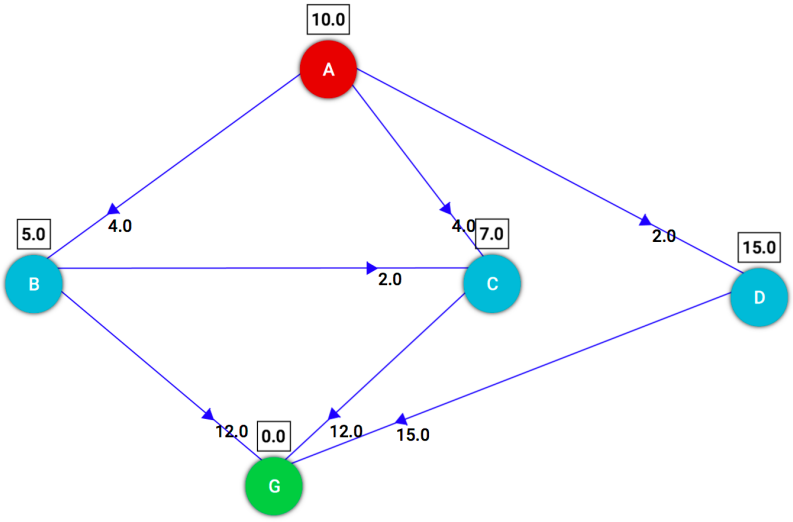
Si mostri l'albero SLD generato dal goal:

?- delunicoLista([6,3,3], X).

indicando eventuali rami di fallimento e quelli tagliati da cut, e la risposta calcolata per la variabile **X.**

Esercizio 5 (6 punti)

Si consideri il seguente grafo, dove A è il nodo iniziale e G il nodo goal, e il numero associato agli archi è il costo dell'operatore per andare dal nodo di partenza al nodo di arrivo dell'arco. Vicino ad ogni nodo, in un quadrato, è indicata inoltre la stima euristica della sua distanza dal nodo goal G:



Si applichi la ricerca **A*** su alberi (non tenendo quindi traccia dei nodi già visitati che non vengono automaticamente eliminati) **disegnando l'albero generato dinamicamente.** In caso di non determinismo si selezionino i nodi da espandere secondo l'ordine alfabetico. Si indichino:

- i nodi espansi nell'ordine di espansione;
- i nodi sulla strada della soluzione e il costo della soluzione;

Si indichi per la ricerca **A*** la condizione sulla stima euristica h(n) che garantisce l'ottimalità di A* su alberi e se è soddisfatta in questo caso.

Si mostri poi cosa si ottiene se applica, invece, la ricerca **Depth-first,** discutendo l'ottimalità o meno della soluzione individuata con questa seconda ricerca.

Esercizio 6 (5 punti)

Dopo avere brevemente introdotto l'algoritmo di Forward Checking, si faccia vedere la sua applicazione sul problema in esame. Per la scelta della variabile da istanziare, si considerino le variabili secondo il loro ordine alfabetico. Per la scelta del valore, si prediliga sempre il valore minore disponibile nel dominio.

A::[4, 5, 6, 7, 8, 9, 10]
B::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
C::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
D::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
A<B-1
C>B-6
A=D+4

10 Settembre 2024 - Soluzioni

Esercizio 1

Formule in FOL:

- sportivo(laura)
- ∀ X (sportivo (X) → atleta(X))
- podista(mario)
- ∀ X (podista(X) → sportivo(X))
- ∀ X (∀Y (atleta(X) ∧ atleta(Y) ∧ batte(X,Y) → ¬ più_forte_di(Y,X)))
- batte(laura,mario)

Goal: ¬ più_forte_di(mario,laura)

Traduzione in clausole:

1. C1: sportivo(laura)
2. C2: ¬ sportivo(X) ∨ atleta(X)
3. C3: podista(mario)
4. C4: ¬ podista(X) ∨ sportivo(X))
5. C5: ¬ atleta(X) ∨ ¬ atleta(Y) ∨ ¬ batte (X,Y) ∨ ¬ più_forte_di(Y,X)
6. C6: batte(laura,mario)

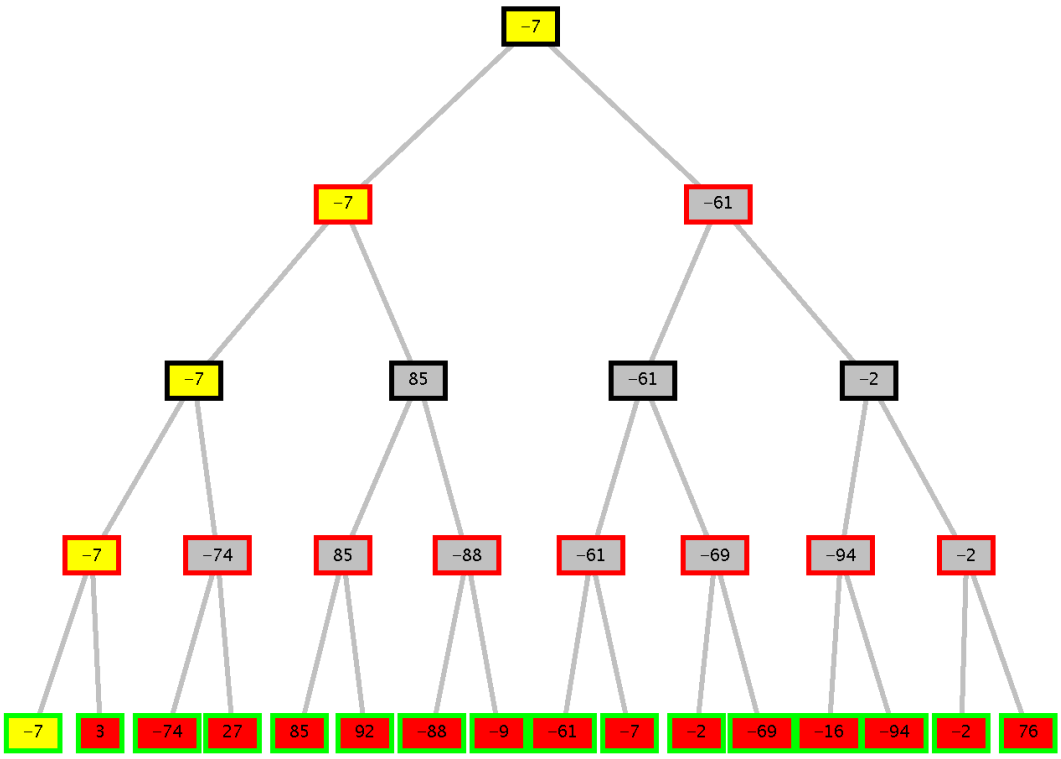
Gneg: più_forte_di(mario,laura)

Risoluzione:

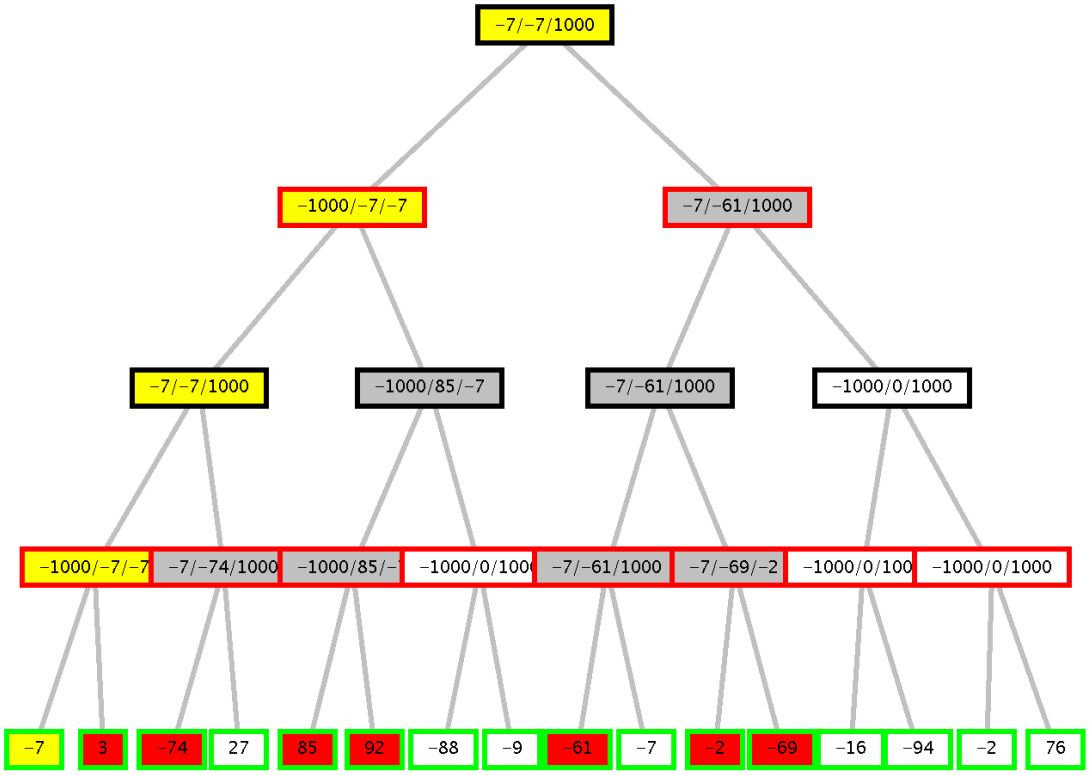
C7: Gneg+C5: ¬ atleta(laura) ∨ ¬ atleta(mario) ∨ ¬ batte (laura,mario)
C8: C7+C6: ¬ atleta(laura) ∨ ¬ atleta(mario)
C9: C8+C2: ¬ sportivo(laura) ∨ ¬ atleta(mario)
C10: C9 + C1: ¬ atleta(mario)
C11: C10+C2: ¬ sportivo(mario)
C12: C11+C4: ¬ podista(mario)
C13: C12+C3: contraddizione logica!!!

Esercizio 2

Min-max:



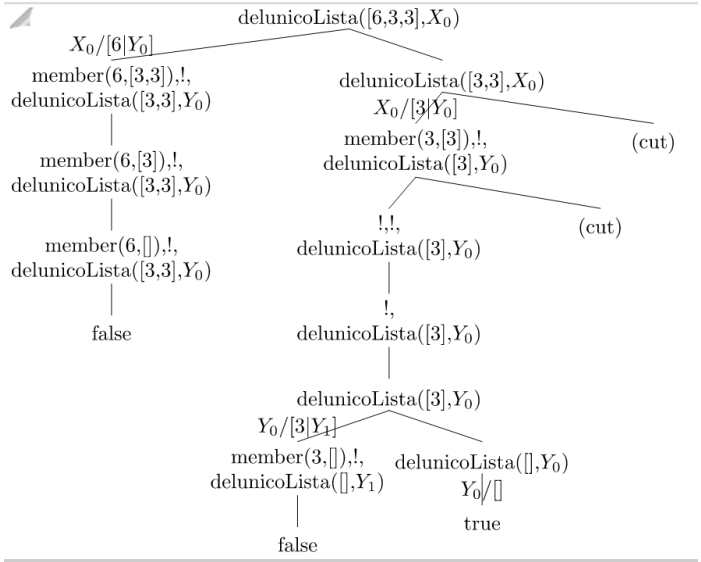
Alfa-beta:



Esercizio 3

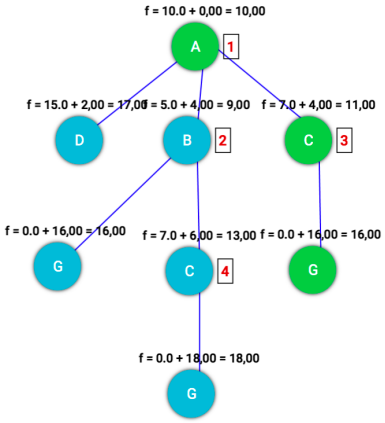
selectList([], _ , _ , []).
selectList([A|R], M, L2, [A|Ra]) :- A>M, \+(member(A,L2)), !, selectList(R, M, L2, Ra).
selectList([_|R], M, L2, Ra) :- selectList(R, M, L2, Ra).

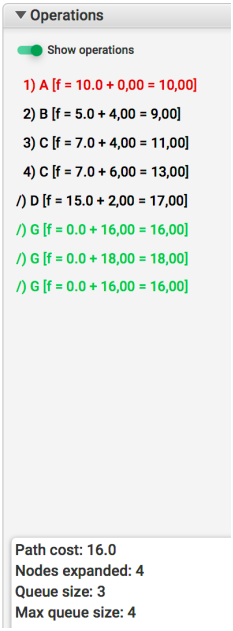
Esercizio 4



Risposta calcolata: **X** = [3]

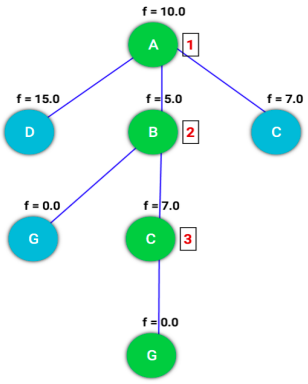
Esercizio 5 Con A*:

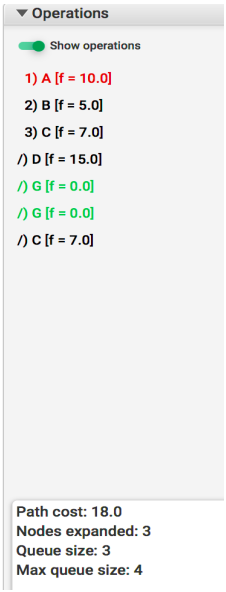




Nodi espansi: ABCC
Nodi sulla strada della soluzione: ACG (o ABG). Costo 16. Ottimale.
La condizione sulla funzione euristica stimata h* (n) che garantisce l'ottimalità della ricerca è la condizione di ammissibilità che deve valere per ogni nodo dell'albero e che è verificata se la h*(n) è ottimista cioè h*(n) <= h(n). Tale condizione è soddisfatta in questo caso.

Depth-first:





Nodi espansi: ABC. Nodi sulla strada della soluzione: ABCG. Costo 18 non Ottimale (la strategia non lo garantisce).

Esercizio 6 (per la descrizione del FC si consulti il materiale del Corso)

	A	**B**	**C**	**D**
Labeling e FC	**A=4**	[1...10]	[1...10]	[1...10]
FC		**[6...10]**	[1...10]	**[]**
Backtracking				Fail
Labeling	**A=5**	[1...10]	[1...10]	[1...10]
FC		**[7...10]**	[1...10]	**[1]**
Labeling		**B=7**	[1...10]	[1]
FC			**[2...10]**	[1]
Labeling			**C=2**	[1]
FC				[1]
Labeling (e soluzione)	A=5	B=7	C=2	**D=1**