FONDAMENTI DI INTELLIGENZA ARTIFICIALE

13 Giugno 2024 - Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (6 punti)

Si formalizzi in logica dei predicati del primo ordine la seguente base di conoscenza:

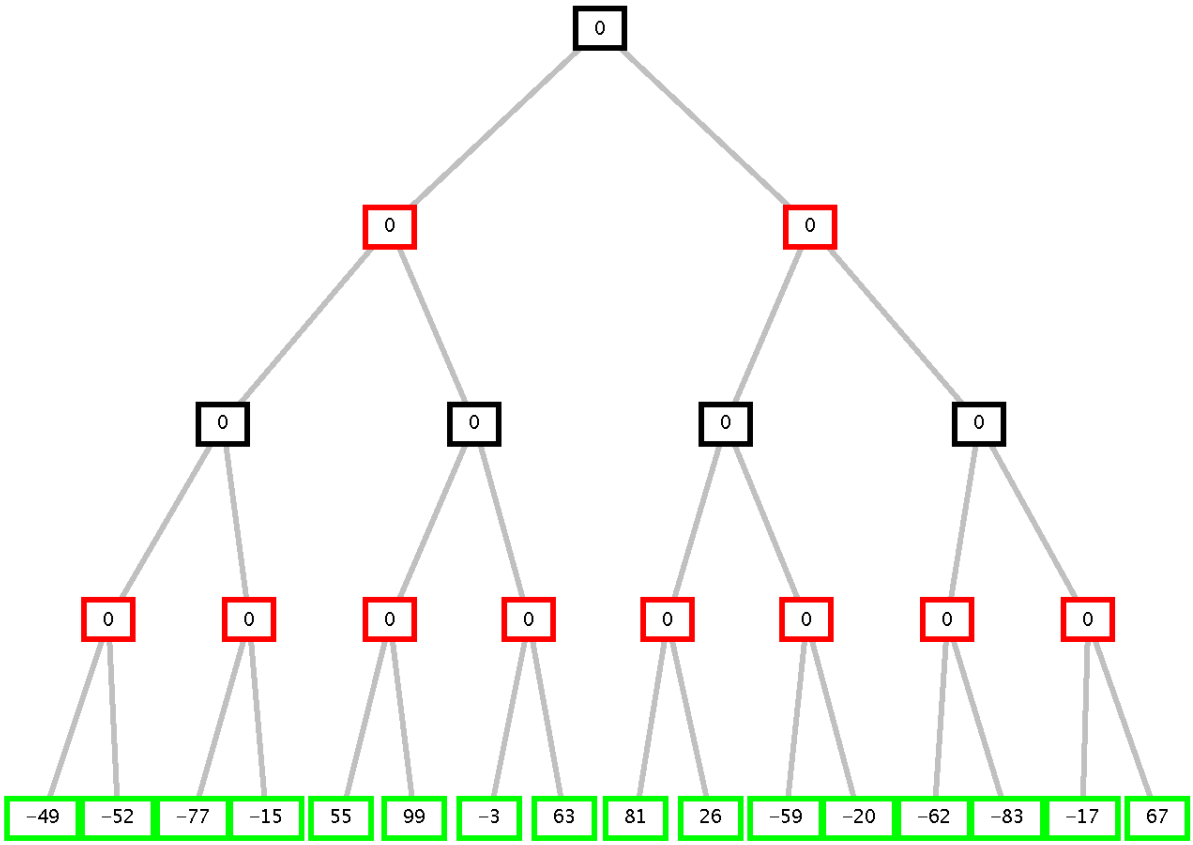
1. *Tutte le persone ammalate sono tristi,*
2. *Tutte le persone non ammalate sono felici.*
3. *Tutte le persone sono tristi o felici, ma non entrambe le cose (or esclusivo)*
4. *Esiste una persona ammalata.*

utilizzando i predicati: ***triste***(*X*), ***ammalato***(*X*), ***felice***(*X*).

Si trasformino poi le formule in clausole, e si dimostri, applicando il principio di risoluzione, che (query) : *Esiste una persona non felice e ammalata.*

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 in Prolog il predicato **sostLista(L1,N,Val,L2)**, che dato il numero intero **N** e una lista **L1** di numeri interi, restituisca in **L2** la lista ottenuta sostituendo l'elemento in posizione **N** in **L1** con l'elemento **Val**. Il primo elemento di **L1** è considerato in posizione **0**. Se la posizione specificata è **>=** della dimensione di **L1** non verrà fatta alcuna sostituzione e la lista in uscita sarà quindi uguale a **L1.** Se **L1** è la lista vuota verrà restituita in uscita la lista vuota.

Esempio:

?- sostLista([2,9,5,4], 2, 7,X).

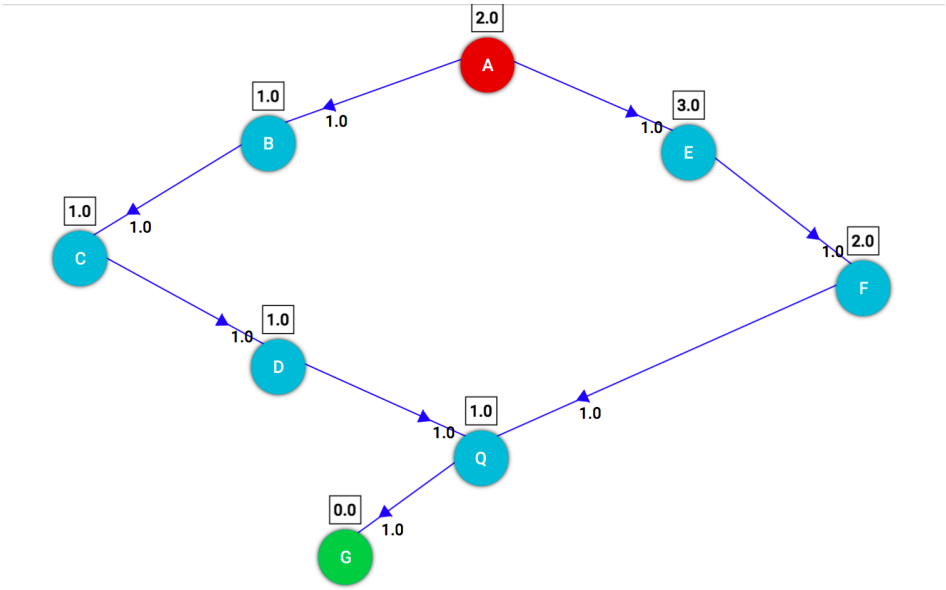
X = [2, 9, 7, 4]

?- sostLista([2,9,5,4], 5, 7,X).

X = [2,9,5,4]

Esercizio 4 (5 punti)
Sia dato il seguente programma Prolog **selezMax(X,L,Lout)** che, dato un numero intero **X** e una lista di numeri interi **L**, ha successo con **Lout** lista che contiene tutte le occorrenze di numeri in **L** maggiori di **X**:
selezMax(_,[],[]):-!.
selezMax(X,[Y|T], [Y|L]) :- Y > X,!,selezMax(X,T,L).
selezMax(X,[_|T], L) :- selezMax(X,T,L).
si mostri l'albero SLD generato dal goal:
?- selezMax(1, [1,4,1], L).
indicando eventuali rami di fallimento e quelli tagliati da cut, e la risposta calcolata per la variabile **L.**

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.

Esercizio 6 (5 punti)
Dopo avere brevemente introdotto l'algoritmo di Forward Checking, si faccia vedere la sua applicazione sul problema in esame, considerando le variabili da istanziare secondo il loro ordine alfabetico. Nella scelta dei valori, si prediliga sempre di assegnare il valore minore.
A::[2, 3, 4, 5, 6]
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-2
A=D+1

13 Giugno 2024 - Soluzioni

Esercizio 1

Rappresentazione in FOL (e in clausole):

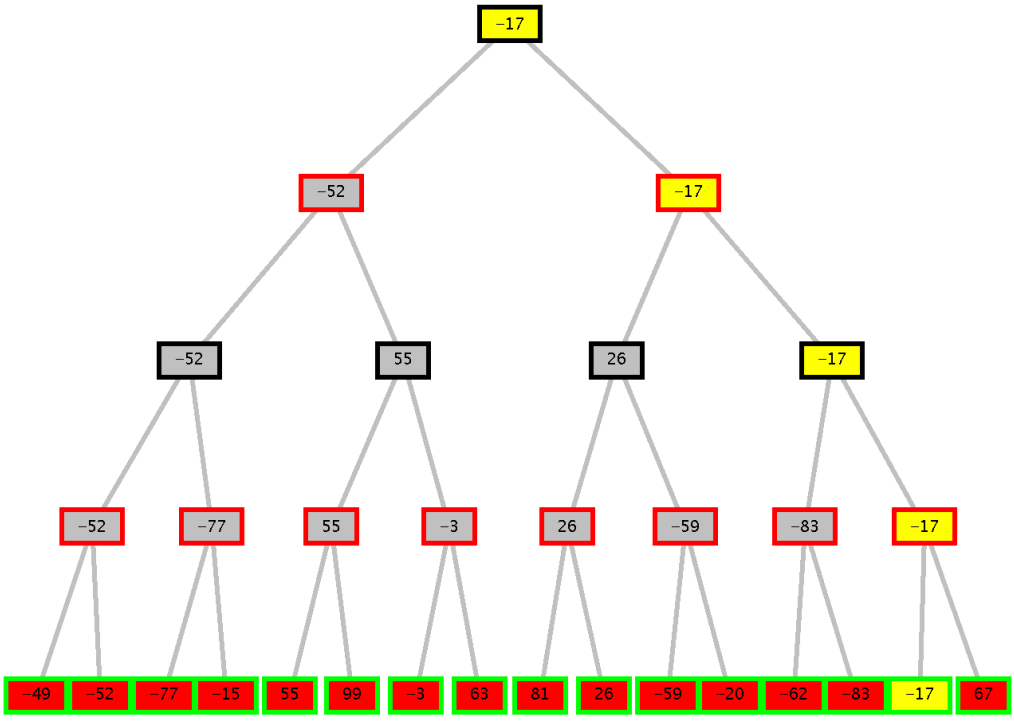
1. ∀X ammalato(X) ⟹ triste(X)
   ¬ ammalato(X) ∨ triste(X)
2. ∀X ¬ ammalato(X) ⟹ felice(X)
   ammalato(X) ∨ felice(X)
3. ∀X triste (X) **EX** ∨ felice(X)
   3.1 triste (X) ∨ felice(X)
   3.2 ¬ triste (X) ∨ ¬ felice(X)
4. ∃X ammalato(X)
   Skolem: ammalato(c1).
5. Query: ∃X ¬ $felice$(X) ∧ ammalata(X)
   Negandola si ottiene: felice(X) ∨ ¬ ammalato(X).

Applicando il Principio di Risoluzione:

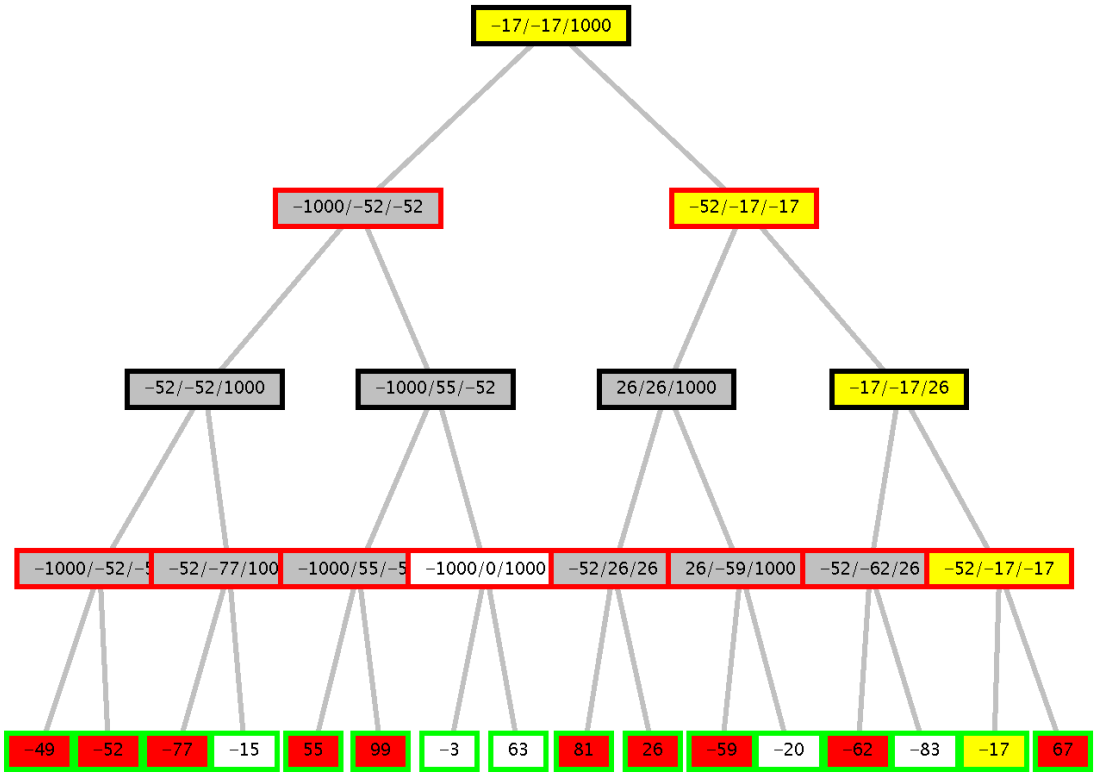
5 + 3.2 = 6: ¬ triste (X) ∨ ¬ ammalato(X).
6 + 1 = 7: ¬ ammalato(X)
7 + 4: contraddizione logica!

Esercizio 2

Min max:



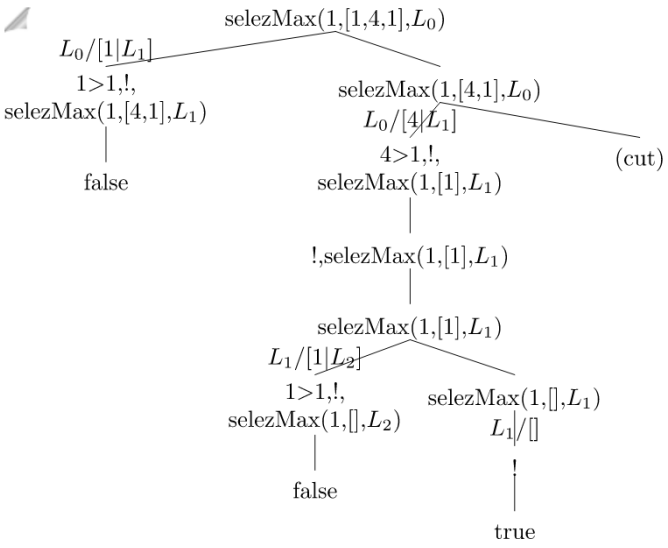
Alfa Beta:



Esercizio 3

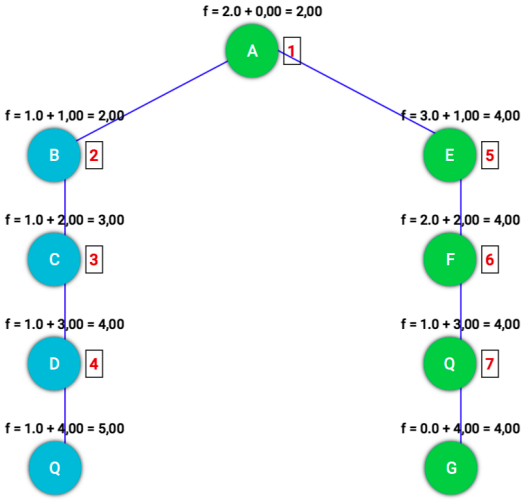
```
sostLista([],_,_,[]).
sostLista([H|Rest],0,Val,[Val|Rest]).
sostLista([H|Rest],N,Val,[H|Part]):-N>0,N1 is N-1,sostLista(Rest,N1,Val,Part).
```

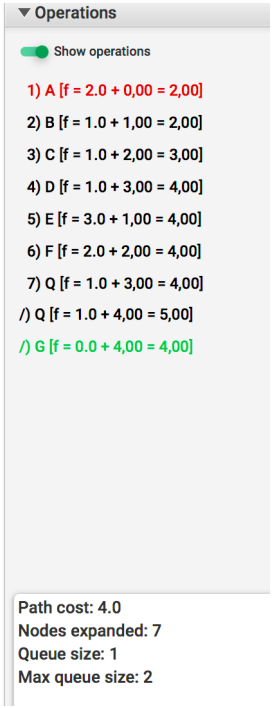
Esercizio 4



Risposta calcolata L= [4]

Esercizio 5
Con A*:





- I nodi espansi nell'ordine di espansione sono: ABCDEFQ(G);
- I nodi sulla strada della soluzione sono AEFQ e il costo della soluzione è 4;

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.

Esercizio 6
Per la spiegazione del forward checking si consulti il materiale del corso.

	A [2…6]	B [1…10]	C [1…10]	D [1…10]
Labeling	**A=2**			
Backtracking		Fail		
Labeling	**A=3**	[1…10]	[1…10]	[1…10]
FC	A=3	**[1]**	[1…10]	**[2]**
Labeling	A=3	**B=1**	[1…10]	[2]
FC	A=3	B=1	[1…10]	[2]
Labeling	A=3	B=1	**C=1**	[2]
FC	A=3	B=1	C=1	[2]
Labeling	A=3	B=1	C=1	**D=2**
soluzione	**A=3**	**B=1**	**C=1**	**D=2**