FONDAMENTI DI INTELLIGENZA ARTIFICIALE

16 Febbraio 2023– Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (6 punti)

Si formalizzino le seguenti frasi in logica dei predicati:

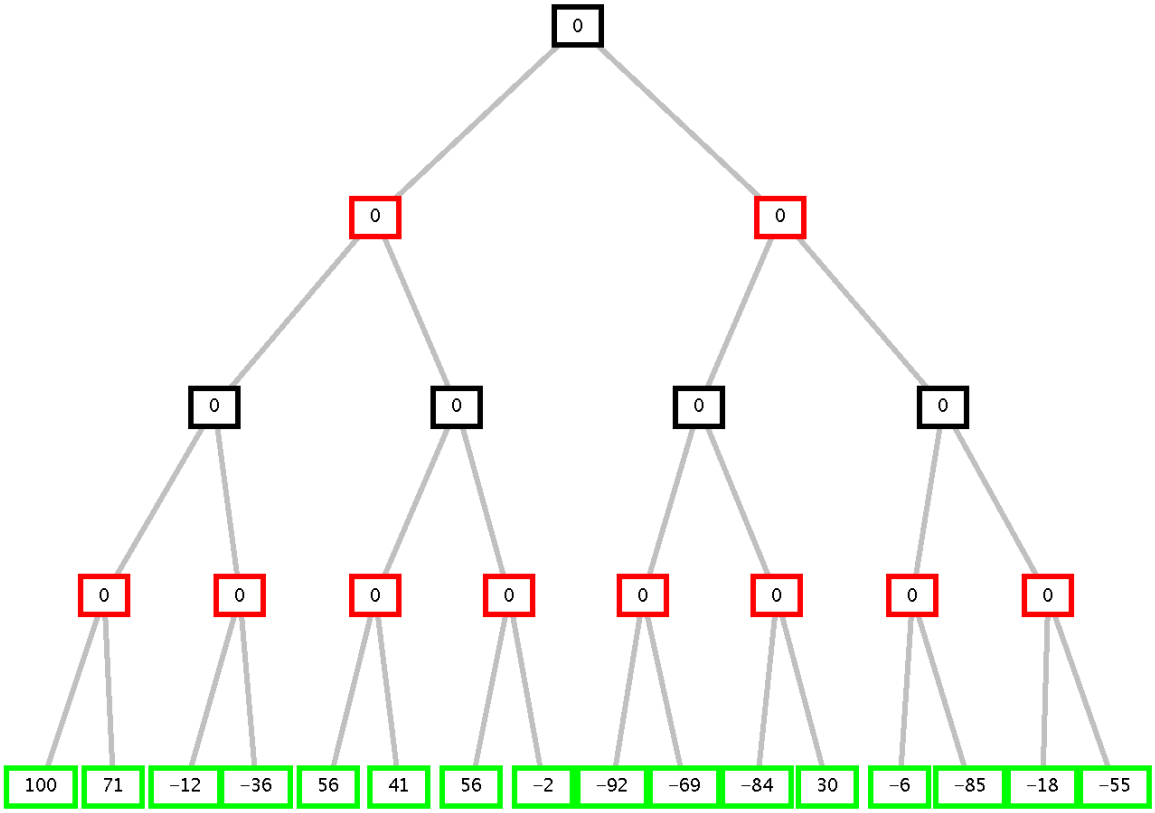
- Tutti i blocchi sono di colore bianco o colore rosso o colore giallo (OR non esclusivo)
- Esiste almeno un blocco
- Non vi è alcun blocco che sia di colore bianco o rosso

Le si trasformi in clausole e poi si usi il principio di risoluzione per derivare il teorema "vi è almeno un blocco che è di colore giallo".

Si utilizzino i seguenti predicati con ovvio significato: **bianco**(X), **giallo**(X), **rosso**(X), **blocco**(X)

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 **listToSet**(L, S) che trasformi una lista L in un insieme S eliminando gli elementi ripetuti di L. Si può utilizzare anche il predicato **member**, così definito:

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

Esempi:

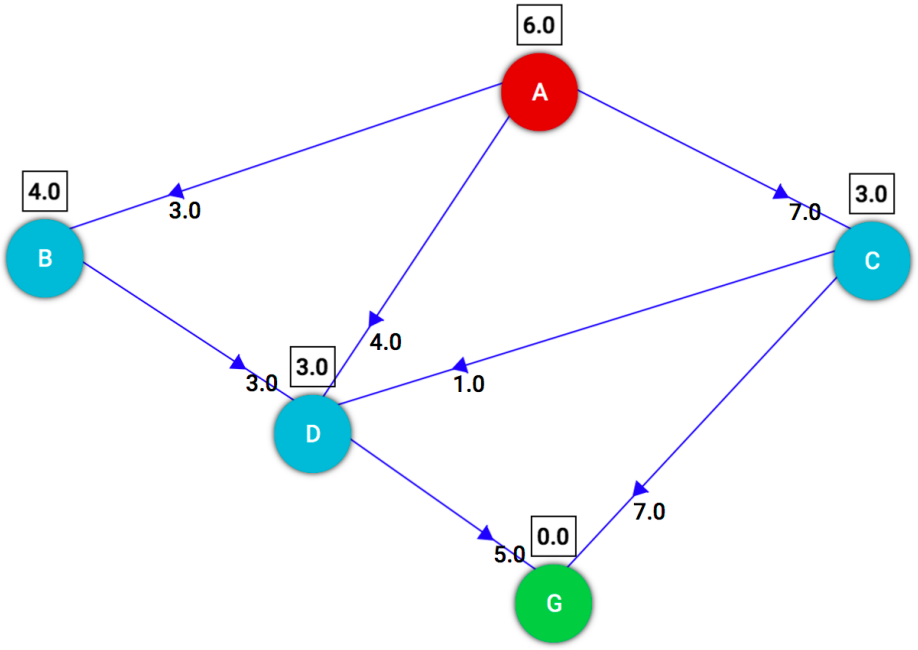
```
?- listToSet([3,2,7,2,3,1,3,2],S).
S = [7, 1, 3, 2]

?-listToSet([1,2,3],S).
S = [1, 2, 3]

?- listToSet([],S).
S = []
```

Esercizio 4 (6 punti)
Nel seguente programma Prolog il predicato **delete(X,L1,L2)** risulta vero quando **L2** rappresenta la lista degli elementi di **L1** in cui sono stati eliminati tutti gli elementi che unificano con **X.**

```
delete( _, [], []).
delete( X,[Y|T], [Y|L]):- X \= Y, !, delete(X,T,L).
delete( X, [_|T], L) :- delete(X,T,L).
```

Si mostri l'*albero SLD* relativo al goal: **?- delete(1, [2,1,4], L).**

Esercizio 5 (5 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) **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;
- se è garantita o meno l'ottimalità.

Esercizio 6 (5 punti)
Dopo avere brevemente introdotto l'algoritmo di Forward Checking, se ne mostri l'esecuzione su questo esempio:
A :: [3, 4, 5]
B :: [1, 2, 3, 4, 5]
C :: [1, 2, 3, 4, 5]
D :: [1, 2, 3, 4, 5]
A<=B+5
C<=B-3
A>=D+3
considerando la prossima variabile da istanziare secondo l'euristica Minimum Remaining Values (MRV); a parità di MRV, si scelga la variabile che viene prima in ordine lessicografico. Nella scelta dei valori, si prediliga sempre di assegnare il valore minore.
Si mostri la riduzione dei domini delle variabili future dopo ogni labeling, fino alla prima soluzione.

16 Febbraio 2023 - Soluzioni

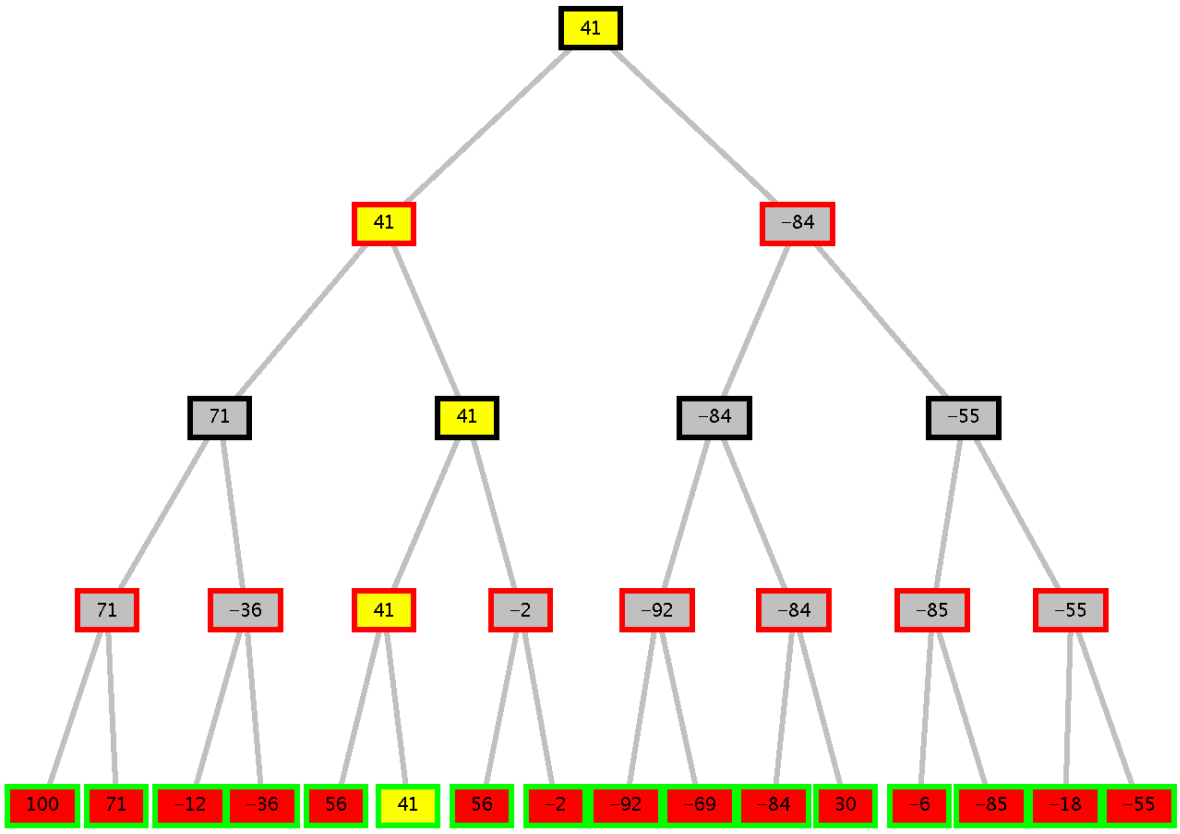
Esercizio 1

Traduzione in logica
$\forall X$ (*blocco*(X)$\Rightarrow$ *bianco*(X) $\vee$ *rosso*(X) $\vee$ *giallo*(X))
$\exists X$ *blocco*(X)
not ($\exists X$(*blocco*(X) $\wedge$ (*bianco*(X) $\vee$ *rosso*(X))))
Goal negato: not $\exists X$(*blocco(X)* $\wedge$ *giallo(X)*)

Trasformazione in clausole
1. not *blocco*(X) $\vee$ *bianco*(X) $\vee$ *rosso*(X) $\vee$ *giallo*(X)
2. *blocco*(*b1*) // b1 costante di Skolem
3. not *blocco*(X) $\vee$ not *bianco*(X)
4. not *blocco*(X) $\vee$ not *rosso*(X)
5. not *blocco*(X) $\vee$ not *giallo*(X)

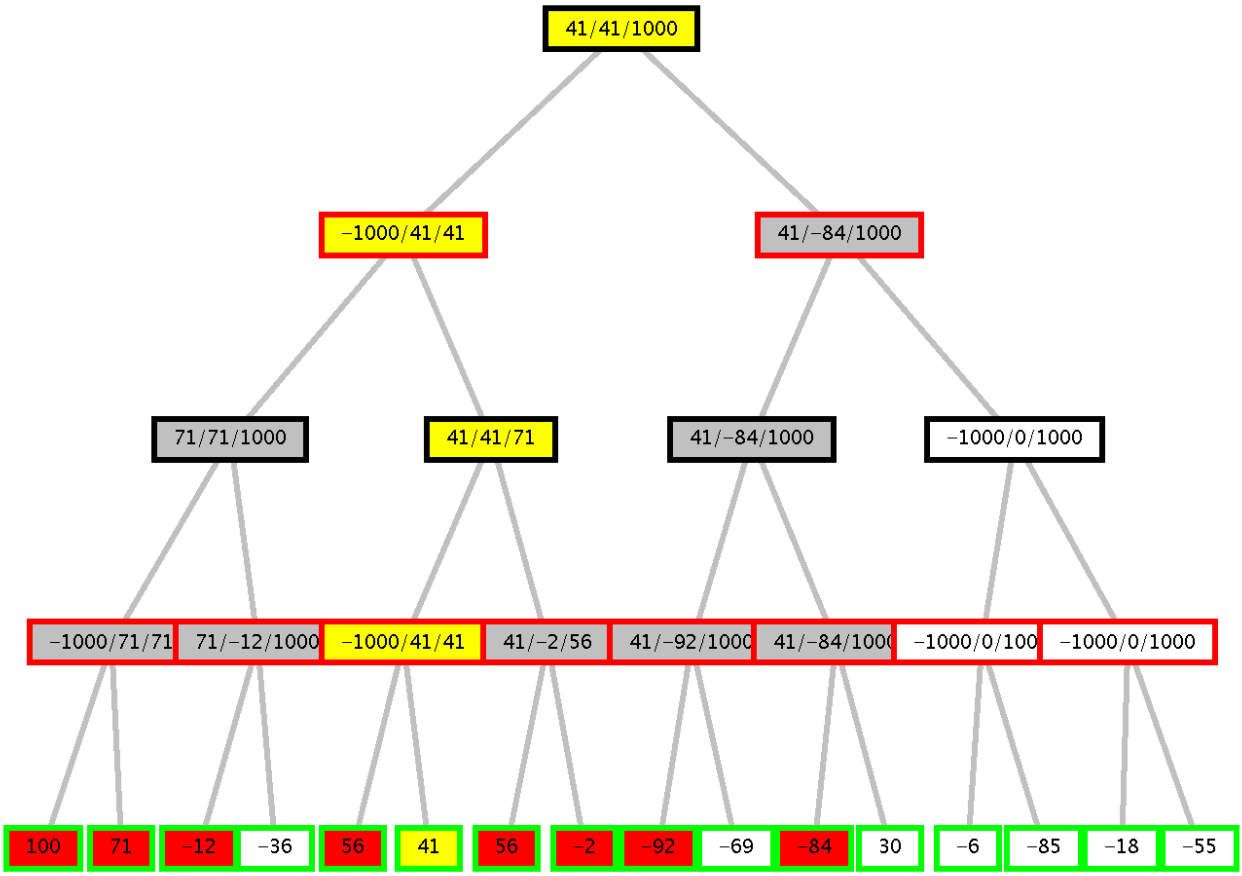
Risoluzione
6. (da 1 + 5) not *blocco*(X) $\vee$ *bianco*(X) $\vee$ *rosso*(X)
7. (da 3 + 6) not *blocco*(X) $\vee$ *rosso*(X)
8. (da 7 + 4) not *blocco*(X)
9. (da 8 + 2) contraddizione!!

Esercizio 2
Min-Max: in giallo la strada selezionata.



Alfa-Beta: In rosso i nodi espansi, in giallo la strada trovata, i nodi in bianco non sono esplorati per effetto dei tagli

alfa-beta.

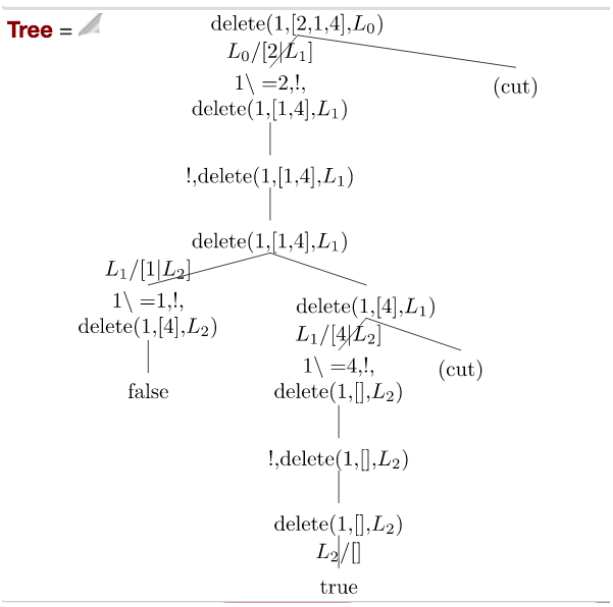


Esercizio 3

```
listToSet([],[]).
listToSet([H|T],Set) :-
        member(H,T),
        !,
        listToSet(T,Set).
listToSet([H|T], [H|Partial]) :-
        listToSet(T,Partial).
```

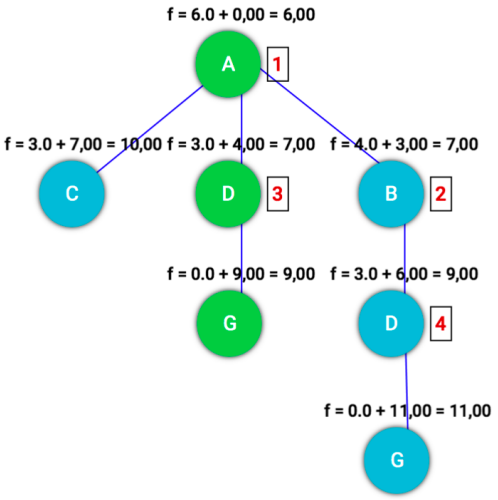
Esercizio 4

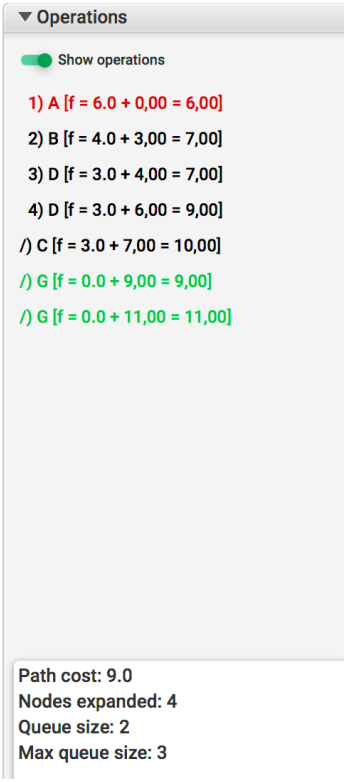
Goal: delete(1,[2,1,4],L).



Esercizio 5

Con A* i nodi espansi sono ABDDG e la soluzione ha costo 9 (ottimale):





Esercizio 6

Vedi slide del corso per spiegare FC.

	A	**B**	**C**	**D**
Labeling	**A=3**	[1...5]	[1...5]	[1...5]
FC	A=3	[1...5]	[1...5]	**Fail**
Backtracking e Labeling	**A=4**	[1...5]	[1...5]	[1...5]
FC	A=4	[1...5]	[1...5]	**[1]**
Labeling	A=4	[1...5]	[1...5]	**D=1**
FC	A=4	[1..5]	[1..5]	D=1
Labeling	A=4	**B=1**	[1...5]	D=1
FC	A=4	B=1	**Fail**	D=1
Backtracking e Labeling	A=4	**B=2**	[1...5]	D=1
FC	A=4	B=2	**Fail**	D=1
Backtracking e Labeling	A=4	**B=3**	[1...5]	D=1
FC	A=4	B=3	**Fail**	D=1
Backtracking e Labeling	A=4	**B=4**	[1...5]	D=1
FC	A=4	B=4	**[1]**	D=1
Labeling	A=4	B=4	**C=1**	D=1
Soluzione	**A=4**	**B=4**	**C=1**	**D=1**