

Esercizio 1 (6 punti)

Si rappresentino in logica dei predicati del I ordine le seguenti frasi, le si trasformi in clausole e si dimostri usando la risoluzione che *esiste un appassionato di gatti che non è appassionato anche di cani*.

- *Giovanni ama gli animali.*
- *Chi ama gli animali è appassionato di cani o gatti (non esclusivo).*
- *Gli appassionati di gatti non amano la fedeltà.*
- *Gli appassionati di cani amano la fedeltà.*
- *Giovanni non ama le cose che ama Maria e viceversa.*
- *Maria ama la fedeltà.*

Si usino i seguenti predicati con ovvio significato.

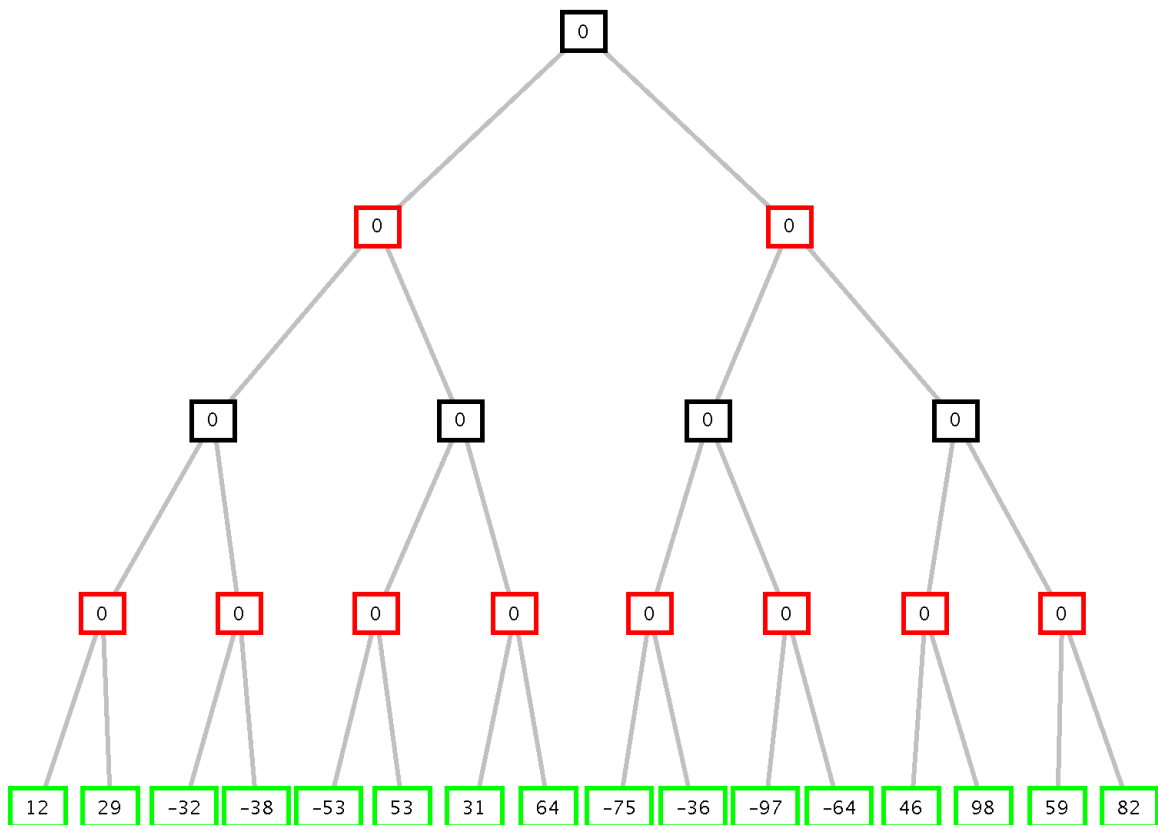
ama(X,Y). X ama Y

appassionato(X,Y). X è appassionato di Y

e le costanti: **cane, gatto, giovanni, maria, fedelta, animali**.

Esercizio 2 (5 punti)

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



- 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).
- 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 **ins_ord(Element,L1,L2)**, che inserisce **Element** nella lista ordinata **L1** ottenendo la lista **L2** ordinata.

Esempi:

?- ins_ord(5,[1,2,6,7],L).

$L = [1,2,5,6,7]$
?- ins_ord(3,[],L).

$L = [3]$

Esercizio 4 (6 punti)

Nel seguente programma Prolog il predicato **member(Element,L)** è vero quando **Element** appartiene alla lista **L**:

member(X,[X|_T]):-!.

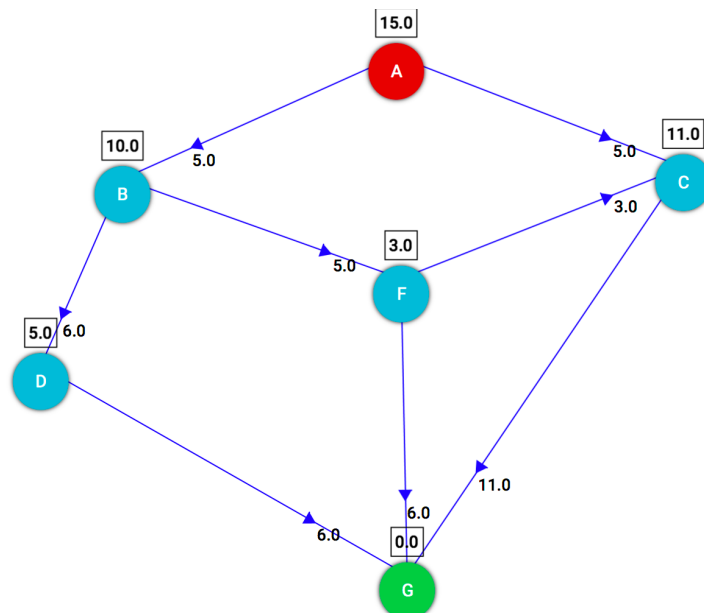
member(X,[_H|T]):-
 member(X,T).

Si mostri l'albero SLD relativo al goal: ?- $X=3$, \+ member(X,[1]), member(X,[3]).

Nota: \+ è il not.

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) **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.

Si mostri poi cosa succederebbe con ricerca **Greedy-Best-First**.

Esercizio 6 (4 punti)

Dopo avere brevemente introdotto l'arc-consistency, se ne mostri l'esecuzione su questo esempio:

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

$B < A$

$C < B$

$A = C + 2$

disegnando il grafo/rete di vincoli e mostrando la riduzione dei domini delle variabili ad ogni iterazione, fino alla quiescenza della rete.

Esercizio 1

Traduzione in predicati in **logica del primo ordine**:

$\text{ama}(\text{giovanni}, \text{animali})$

$\forall X \text{ ama}(X, \text{animali}) \Rightarrow \text{appassionato}(X, \text{cani}) \text{ or } \text{appassionato}(X, \text{gatti})$

$\forall X \text{ appassionato}(X, \text{gatti}) \Rightarrow \text{not } \text{ama}(X, \text{fedelta})$

$\forall X \text{ appassionato}(X, \text{cani}) \Rightarrow \text{ama}(X, \text{fedelta})$

$\forall Y \text{ ama}(\text{giovanni}, Y) \Rightarrow \text{not } \text{ama}(\text{maria}, Y)$

$\forall Y \text{ ama}(\text{maria}, Y) \Rightarrow \text{not } \text{ama}(\text{giovanni}, Y)$

$\text{ama}(\text{maria}, \text{fedelta})$

Query: $\exists X (\text{appassionato}(X, \text{gatto}) \text{ and } \text{not } \text{appassionato}(X, \text{cane}))$

Clausole:

1. $\text{ama}(\text{giovanni}, \text{animali})$
2. $\text{not } \text{ama}(X, \text{animali}) \text{ or } \text{appassionato}(X, \text{cani}) \text{ or } \text{appassionato}(X, \text{gatti})$
3. $\text{not } \text{appassionato}(X, \text{gatti}) \text{ or } \text{not } \text{ama}(X, \text{fedelta})$
4. $\text{not } \text{appassionato}(X, \text{cani}) \text{ or } \text{ama}(X, \text{fedelta})$
5. $\text{not } \text{ama}(\text{giovanni}, Y) \text{ or } \text{not } \text{ama}(\text{maria}, Y)$
6. $\text{not } \text{ama}(\text{maria}, Y) \text{ or } \text{not } \text{ama}(\text{giovanni}, Y)$
7. $\text{ama}(\text{maria}, \text{fedelta})$
8. **GoalNeg:** $\text{not } \text{appassionato}(X, \text{gatti}) \text{ or } \text{appassionato}(X, \text{cani})$

Risoluzione:

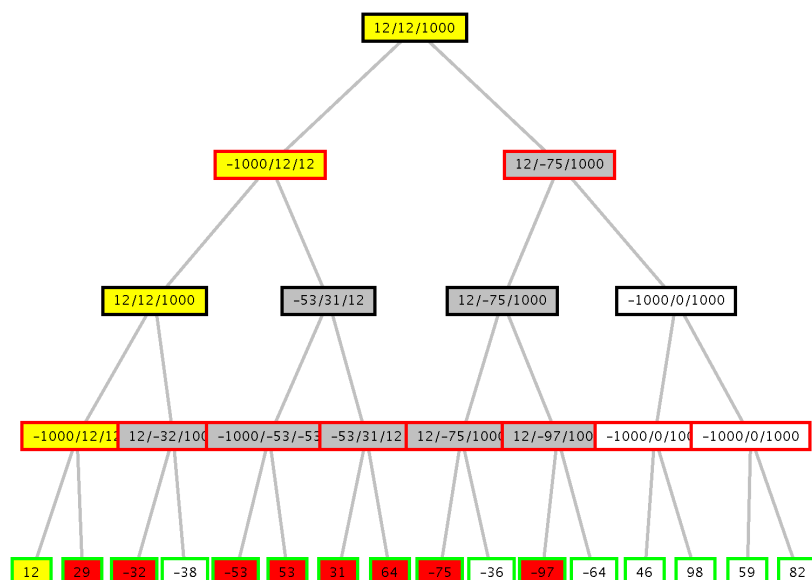
9. Da 6+7: $\text{not } \text{ama}(\text{giovanni}, \text{fedelta})$
10. Da 9+4: $\text{not } \text{appassionato}(\text{giovanni}, \text{cani})$
11. Da 10+8: $\text{not } \text{appassionato}(\text{giovanni}, \text{gatti})$
12. Da 11+2: $\text{not } \text{ama}(\text{giovanni}, \text{animali}) \text{ or } \text{appassionato}(\text{giovanni}, \text{cani})$
13. Da 12+10: $\text{not } \text{ama}(\text{giovanni}, \text{animali})$
14. Da 13+1: clausola vuota CONTRADDIZIONE

Oppure:

9. Da GNeg+2: $\text{not } \text{ama}(X, \text{animali}) \text{ or } \text{appassionato}(X, \text{cani})$
10. Da 9+1: $\text{appassionato}(\text{giovanni}, \text{cani})$
11. Da 10+4: $\text{ama}(\text{giovanni}, \text{fedelta})$
12. Da 11+5: $\text{not } \text{ama}(\text{maria}, \text{fedelta})$
13. Da 12+7: clausola vuota, contraddizione.

Esercizio 2

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.



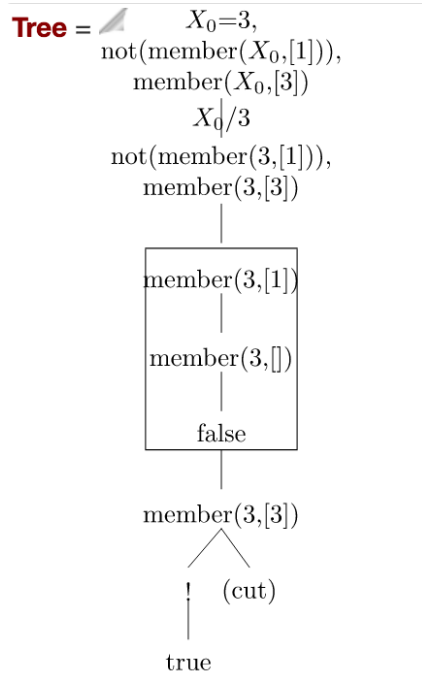
Esercizio 3

```
ins_ord( X, [], [X]).
```

```
ins_ord( X, [Y|T], [X,Y|T]):- X <= Y, !.
```

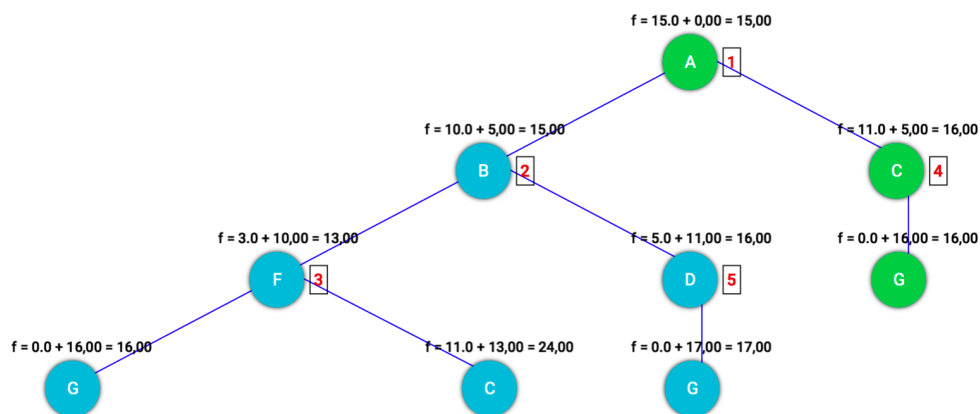
```
ins_ord( X, [Y|T], [Y|L]) :- ins_ord (X,T,L).
```

Esercizio 4



Esercizio 5

Con A* i nodi espansi sono ABFCDG (nodi sul cammino soluzione) e la soluzione ha costo 16 (ottimale):



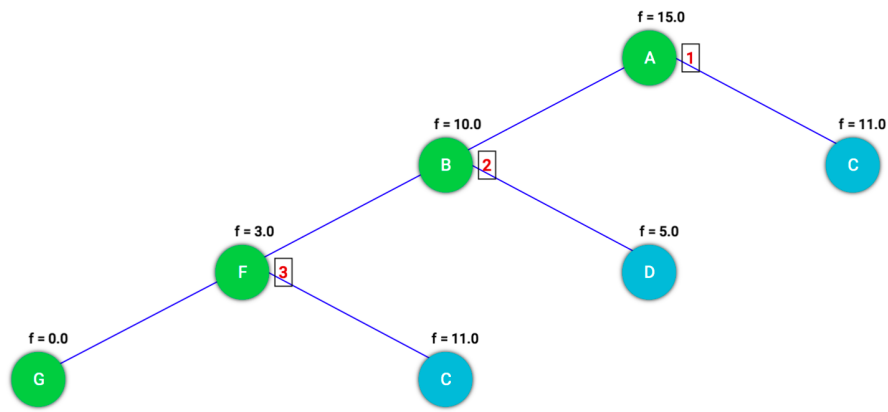
Operations

Show operations

- 1) A [f = 15.0 + 0.00 = 15.00]
 - 2) B [f = 10.0 + 5.00 = 15.00]
 - 3) F [f = 3.0 + 10.00 = 13.00]
 - 4) C [f = 11.0 + 5.00 = 16.00]
 - 5) D [f = 5.0 + 11.00 = 16.00]
- /) G [f = 0.0 + 16.00 = 16.00]
- /) C [f = 11.0 + 13.00 = 24.00]
- /) G [f = 0.0 + 17.00 = 17.00]
- /) G [f = 0.0 + 16.00 = 16.00]

Path cost: 16.0
 Nodes expanded: 5
 Queue size: 3
 Max queue size: 4

Con Greedy Best First i nodi espansi sono ABFG (nodi sul cammino soluzione) e la soluzione ha costo sempre 16.



Operations

☒ Show operations

1) A [f = 15.0]
2) B [f = 10.0]
3) F [f = 3.0]
/) G [f = 0.0]
/) C [f = 11.0]
/) D [f = 5.0]
/) C [f = 11.0]

Path cost: 16.0
Nodes expanded: 3
Queue size: 3
Max queue size: 4

Esercizio 6

Vedi slide del corso per spiegare arc consistency.

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

B<A

C<B

A=C+2

I iterazione	A	B	C
B<A	[1,2,3,4]	[1,2,3]	
A>B	[2,3,4]	[1,2,3]	
C<B			[1,2]
B>C		[2,3]	
A=C+2	[3,4]		
C=A-2			[1,2]
II iterazione	A	B	C
B<A		[2,3]	
A>B	[3,4]		
C<B			[1,2]
B>C		[2,3]	
A=C+2	[3,4]		
C=A-2			[1,2]

Nessuna riduzione alla seconda iterazione, quiescenza.