

Esercizio 1 (6 punti)

Si traducano in logica dei predicati le seguenti frasi:

1. Babbo Natale porta i regali ad i bambini buoni
2. Tutte le mamme hanno un bambino
3. I figli delle mamme buone sono buoni
4. Esiste almeno una mamma buona

Si dimostri poi tramite il principio di risoluzione che Babbo Natale porta i regali a qualcuno.

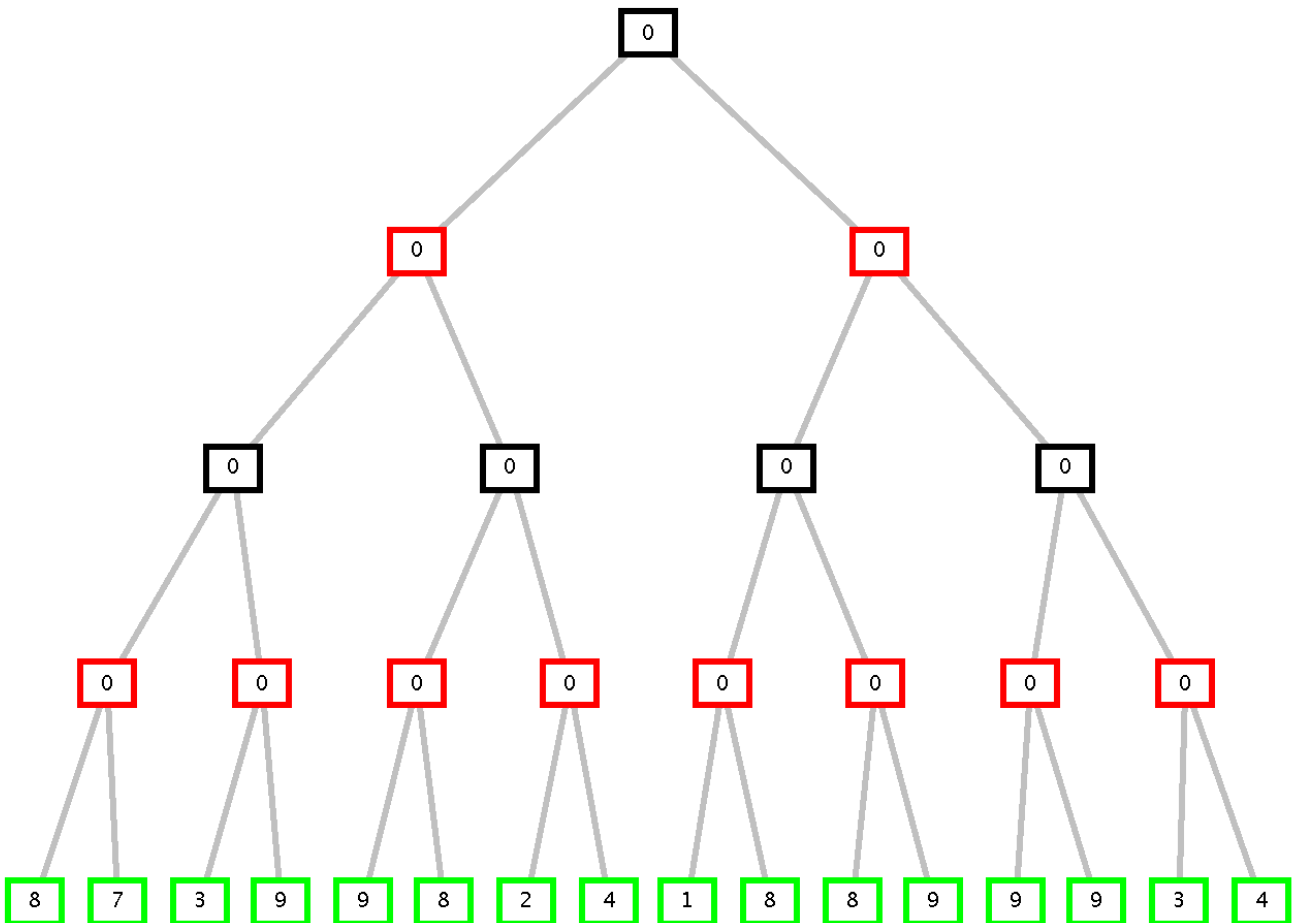
Si usino i seguenti predicati dove ".../n" esprime il numero n di parametri richiesti (cioè l'arietà del predicato):

`porta_regali/2`, `bambino/1`, `mamma/1`, `buono/1`, `figlio/2`.

Il termine "babbo_natale" sia considerato una costante.

Esercizio 2 (5 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 l'algoritmo *min-max* e l'algoritmo *alfa-beta* risolvono il problema e la mossa selezionata dal giocatore.



Esercizio 3 (6 punti)

Dato il seguente programma Prolog:

```
member1(X, [X|_T]).  
member1(X, [_H|T]) :-  
    member1(X, T).
```

si disegni l'albero SLD relativo al goal:

```
?- member1(X, [1,2]), \+ member1(X, [1,3]).
```

Esercizio 4 (5 punti)

Data una lista L di numeri si definisca un predicato Prolog ordered(L) in grado di rispondere con successo se la lista L è ordinata in modo strettamente crescente.

Esempio:

```
?- ordered([1,2,5]).
```

Yes

```
?- ordered([9,2,5]).
```

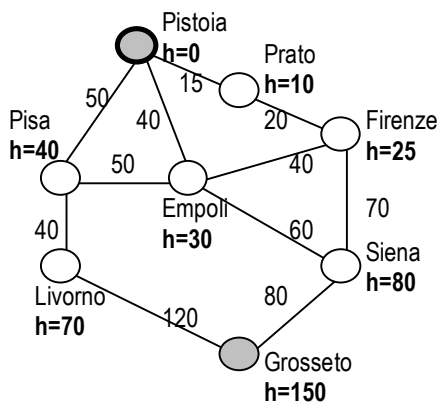
No

```
?- ordered([2,2,5]).
```

No

Esercizio 5 (6 punti)

Dato il seguente grafo\mappa, si utilizzi l'algoritmo A* per individuare il percorso più rapido tra Grosseto e Pistoia. Accanto ad ogni città si trova la stima euristica della sua distanza da Pistoia. Accanto ad ogni strada si trova la lunghezza effettiva del tratto stradale. Si utilizzi A* per trovare la soluzione e si mostri l'albero di ricerca espanso, tenendo conto che i nodi vengono considerati da Ovest a Est (cioè, i nodi a sinistra dell'albero di ricerca saranno quelli più a Ovest e viceversa) e nel dubbio A* espande sempre il nodo più a sinistra.



Basandosi sui valori presenti nella mappa si dica inoltre anche se l'euristica è ammissibile motivando la risposta.

Esercizio 6 (4 punti)

Si descriva, anche aiutandosi con un esempio, cos'è l'arc-consistenza in un sistema CSP e come lavora l'algoritmo di arc-consistenza (AC-3).

Esercizio 1

Trasformazione in clausole

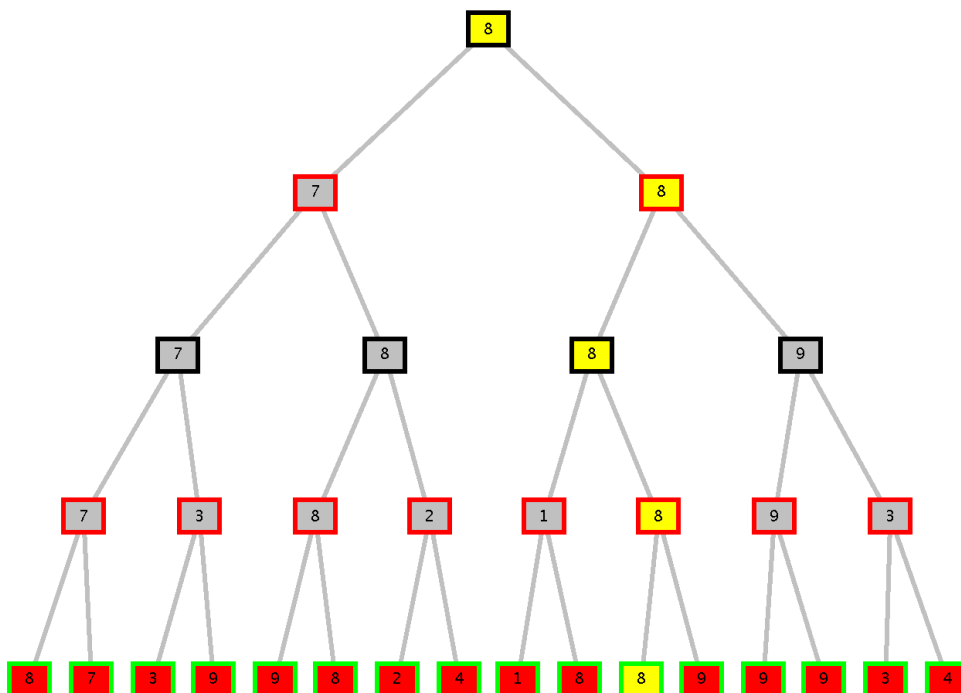
1. $\forall B, bambino(B) \wedge buono(B) \Rightarrow porta_regali(babbo_natale, B)$
 $\sim bambino(B) \vee \sim buono(B) \vee porta_regali(babbo_natale, B)$
 2. $\forall M, mamma(M) \Rightarrow \exists B, figlio(M, B) \wedge bambino(B)$
 $\forall M \exists B \sim mamma(M) \vee (figlio(M, B) \wedge bambino(B))$
 $(\sim mamma(M) \vee figlio(M, f(M))) \wedge (\sim mamma(M) \vee bambino(f(M)))$
 - 2a. $(\sim mamma(M) \vee figlio(M, f(M)))$
 - 2b. $(\sim mamma(M) \vee bambino(f(M)))$
 3. $\forall M \forall B, mamma(M) \wedge buono(M) \wedge figlio(M, B) \Rightarrow buono(B)$
 $\sim mamma(M) \vee \sim buono(M) \vee \sim figlio(M, B) \vee buono(B)$
 4. $\exists X, mamma(X) \wedge buono(X)$
 - 4a. $mamma(maria)$
 - 4b. $buono(maria)$
- Q. $\exists X, porta_regali(babbo_natale, X)$
 $\sim Q. \sim porta_regali(babbo_natale, X)$

Risoluzione

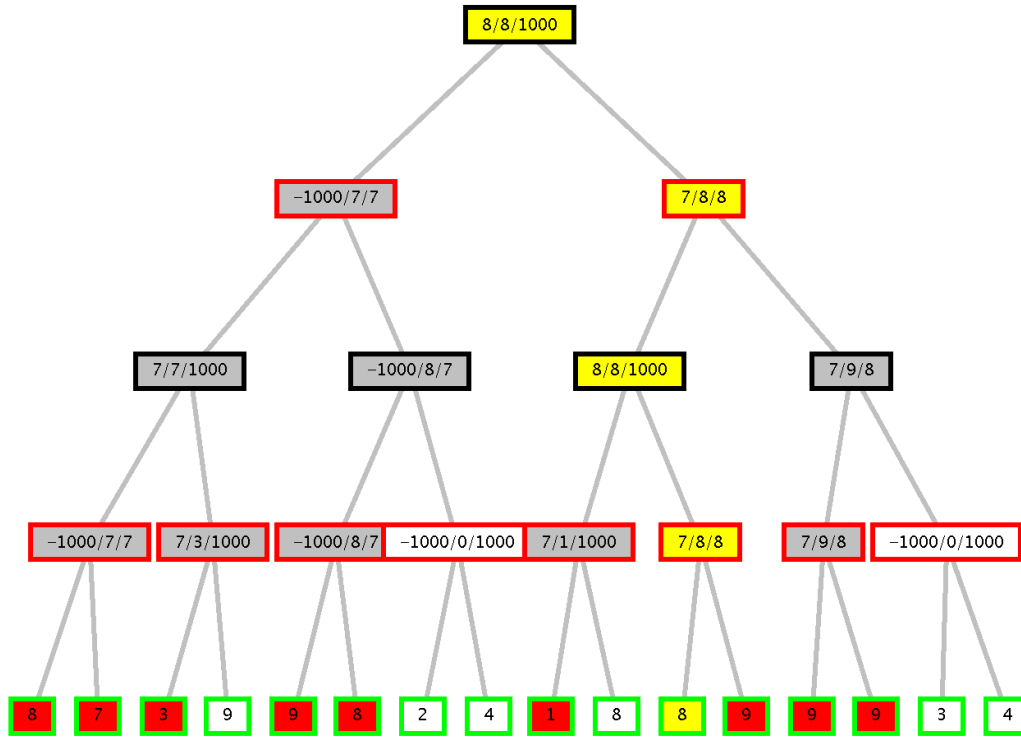
- ($\sim Q+1$) $\sim bambino(X) \vee \sim buono(X)$
 (+2b) $\sim mamma(M) \vee \sim buono(f(M))$
 (+3) $\sim mamma(M) \vee \sim buono(M) \vee \sim figlio(M, f(M))$
 (+2a) $\sim mamma(M) \vee \sim buono(M)$
 (+4a) $\sim buono(maria)$
 (+4b) \square

Esercizio 2

Min-Max:

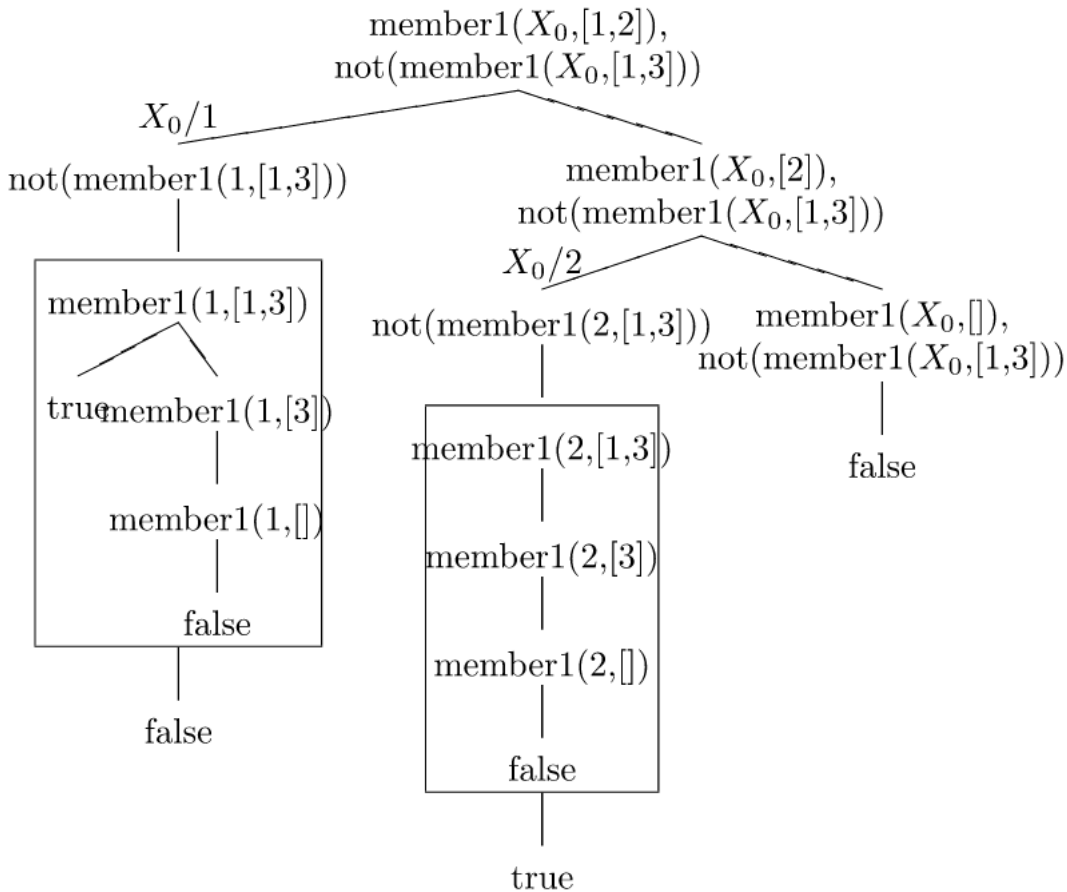


Alfa-beta:



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

Esercizio 3



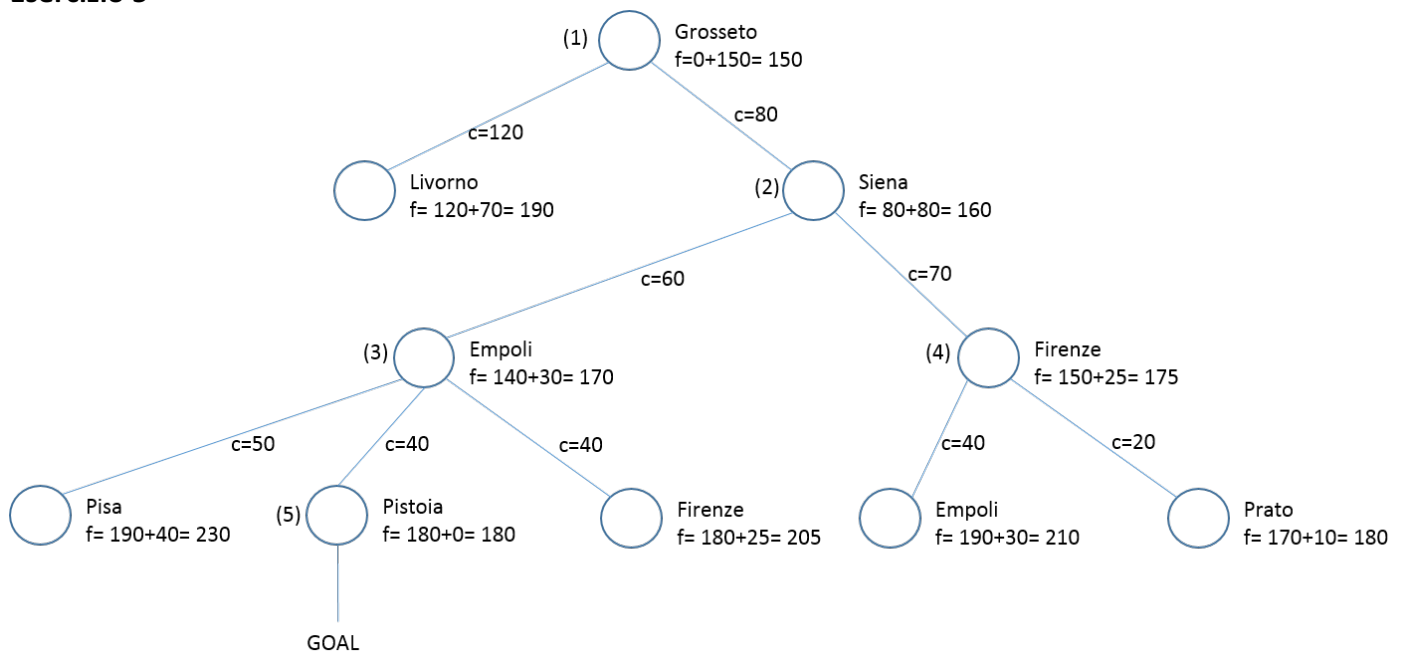
Esercizio 4

`ordered([])`.

`ordered([_])`.

`ordered([X,Y|Ys]) :- X < Y, ordered([Y|Ys])`.

Esercizio 5



Esercizio 6

Vedi slide