

**Esercizio 1 (6 punti)**

Si traducano in logica dei predicati le seguenti frasi:

1. La maestra premia i bambini educati.
2. Tutte le mamme hanno almeno un figlio e è un bambino.
3. I figli delle mamme brave sono educati.
4. Esiste almeno una mamma brava.

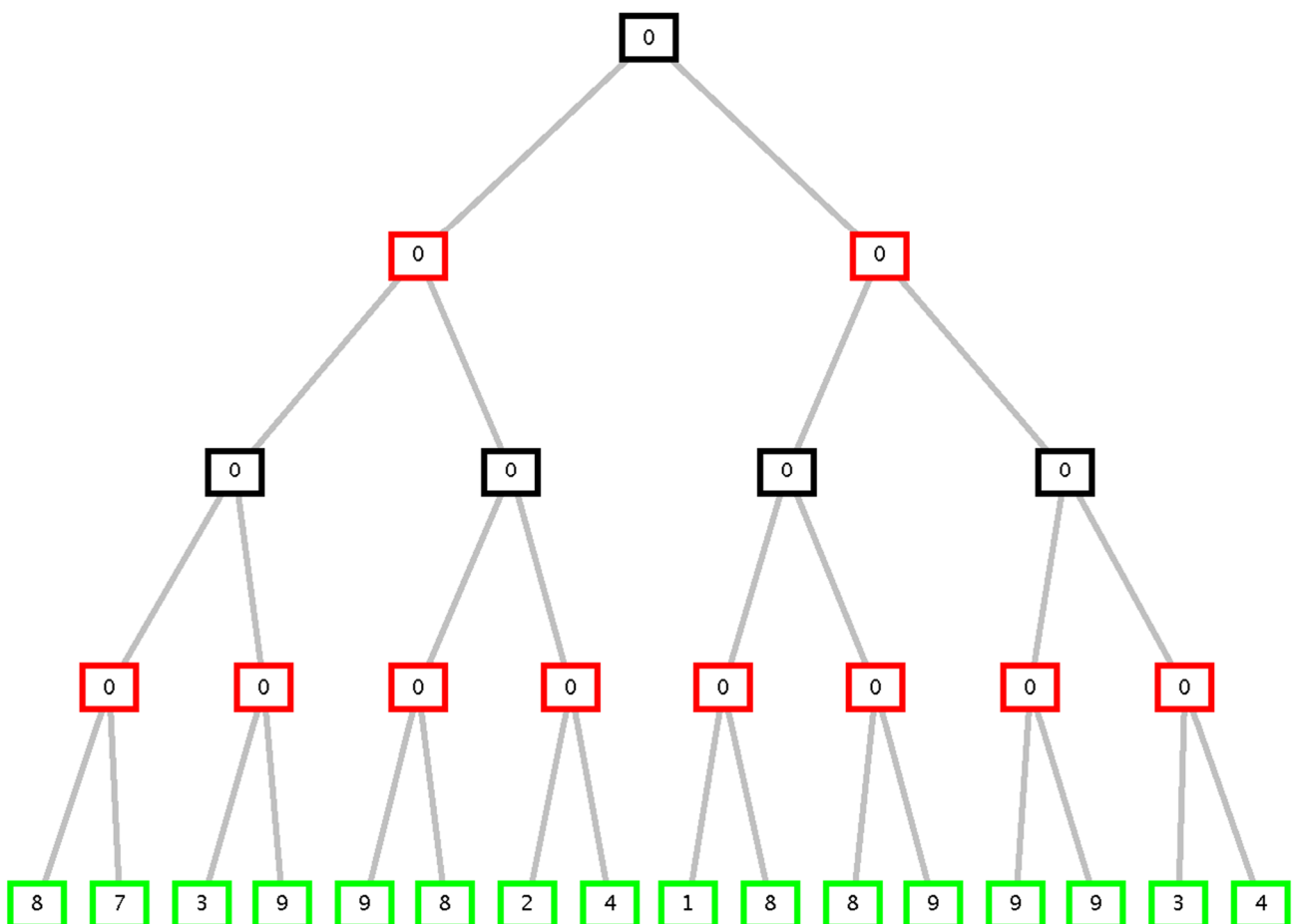
Si dimostri poi, tramite il principio di risoluzione, che la maestra premia qualcuno. Si usino i seguenti predicati dove ".../x" esprime il numero x di parametri richiesti (cioè l'arietà del predicato):

**premia/2, bambino/1, mamma/1, brava/1, educato/1, figlio/2.**

Il termine "maestra" 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|_]) .  
member1(X, [_|T]) :-  
    member1(X, T) .
```

si disegni l'albero SLDNF relativo al goal:

```
?- member1(X, [1,2]), \+ member1(X, [1,3]) . (nota: \+ equivale a not)
```

### 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]) .
```

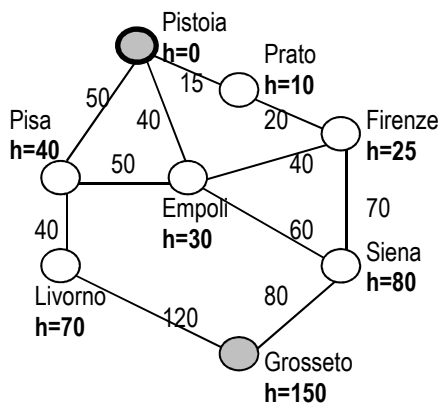
No

```
?- 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 (nota: per figlio si è considerata l'interpretazione per cui il primo argomento è la madre e il secondo il figlio).

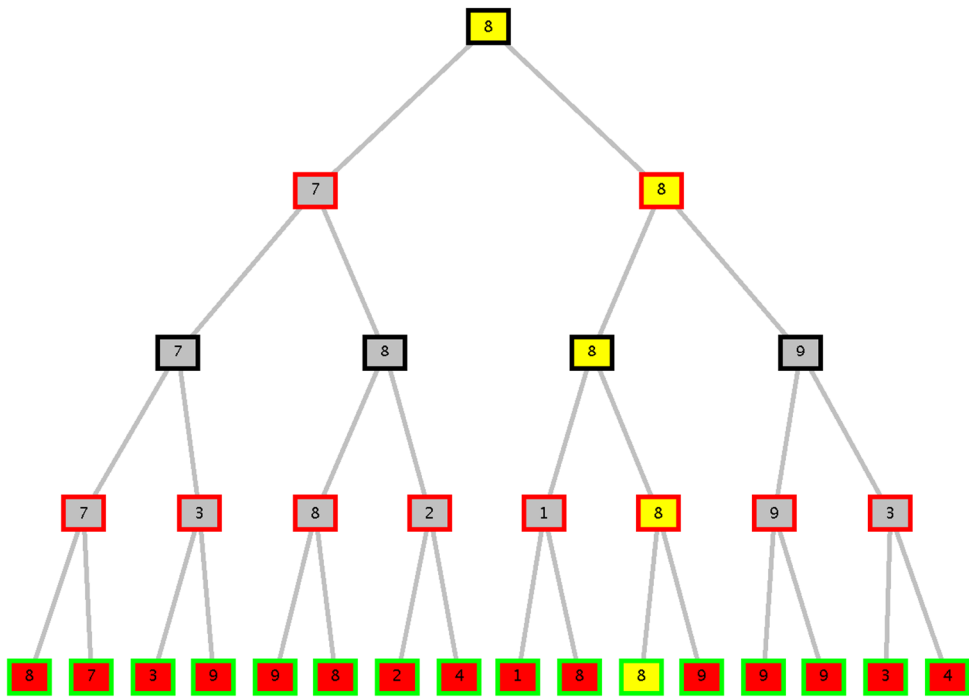
1.  $\forall X \text{ bambino}(X) \wedge \text{educato}(X) \rightarrow \text{ premia}(\text{maestra}, X).$   
 $\neg \text{ bambino}(X) \vee \neg \text{ educato}(X) \vee \text{ premia}(\text{maestra}, X)$
2.  $\forall M \text{ mamma}(M) \rightarrow \exists B \text{ figlio}(M, B) \wedge \text{ bambino}(B)$   
 $\forall M \exists B \neg \text{mamma}(M) \vee (\text{figlio}(M, B) \wedge \text{ bambino}(B) )$   
 $(\neg \text{mamma}(M) \vee \text{figlio}(M, f(M))) \wedge (\neg \text{mamma}(B) \vee \text{ bambino}(f(M)))$ 
  - 2a.  $\neg \text{mamma}(M) \vee \text{figlio}(M, f(M)).$
  - 2b.  $\neg \text{mamma}(M) \vee \text{ bambino}(f(M)).$
3.  $\forall M \forall B, \text{mamma}(M) \wedge \text{brava}(M) \wedge \text{figlio}(M, B) \rightarrow \text{educato}(B).$   
 $\neg \text{mamma}(M) \vee \neg \text{brava}(M) \vee \neg \text{figlio}(M, B) \vee \text{educato}(B).$
4.  $\exists M \text{mamma}(M) \wedge \text{brava}(M)$ 
  - 4a.  $\text{mamma}(m)$
  - 4b.  $\text{brava}(m)$
- G.  $\exists X \text{ premia}(\text{maestra}, X)$   
Gneg.  $\neg \text{ premia}(\text{maestra}, X)$

Risoluzione

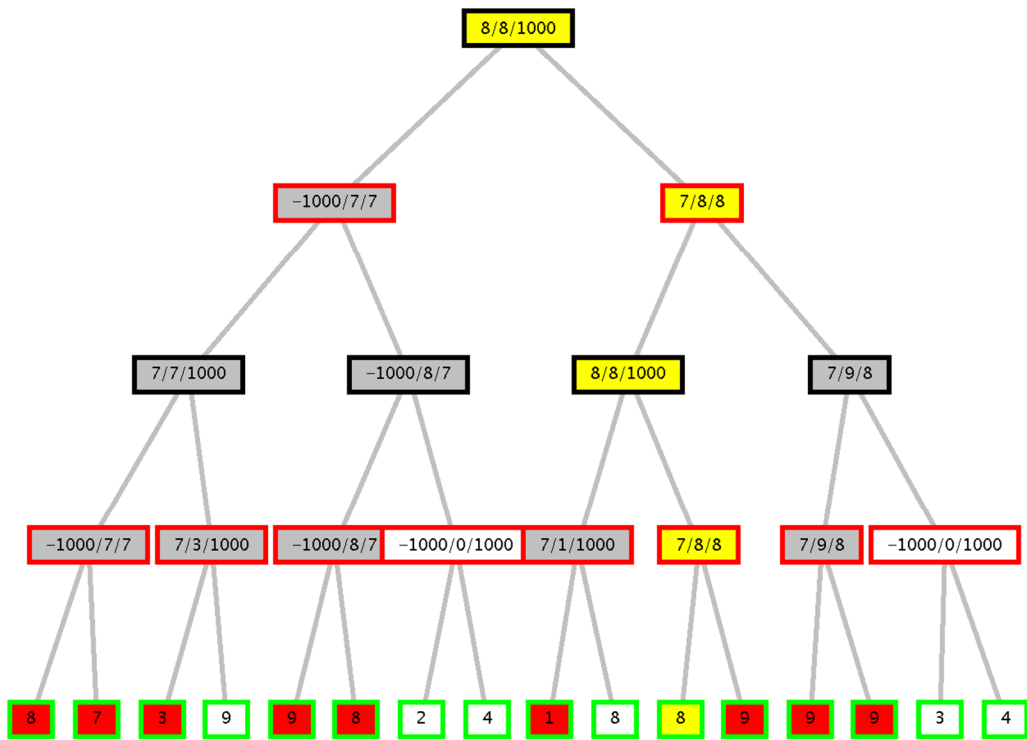
- 5: Gneg + 1:  $\neg \text{bambino}(X) \vee \neg \text{educato}(X).$
- 6: 5 + 2b:  $\neg \text{mamma}(M) \vee \neg \text{educato}(f(M)).$
- 7: 6+3:  $\neg \text{mamma}(M) \vee \neg \text{brava}(M) \vee \neg \text{figlio}(M, f(M)).$
- 8: 7+2a:  $\neg \text{mamma}(M) \vee \neg \text{brava}(M).$
- 9: 8+4a:  $\neg \text{brava}(m).$
- 10: 9+4b:  $\square$

**Esercizio 2**

Min-Max:

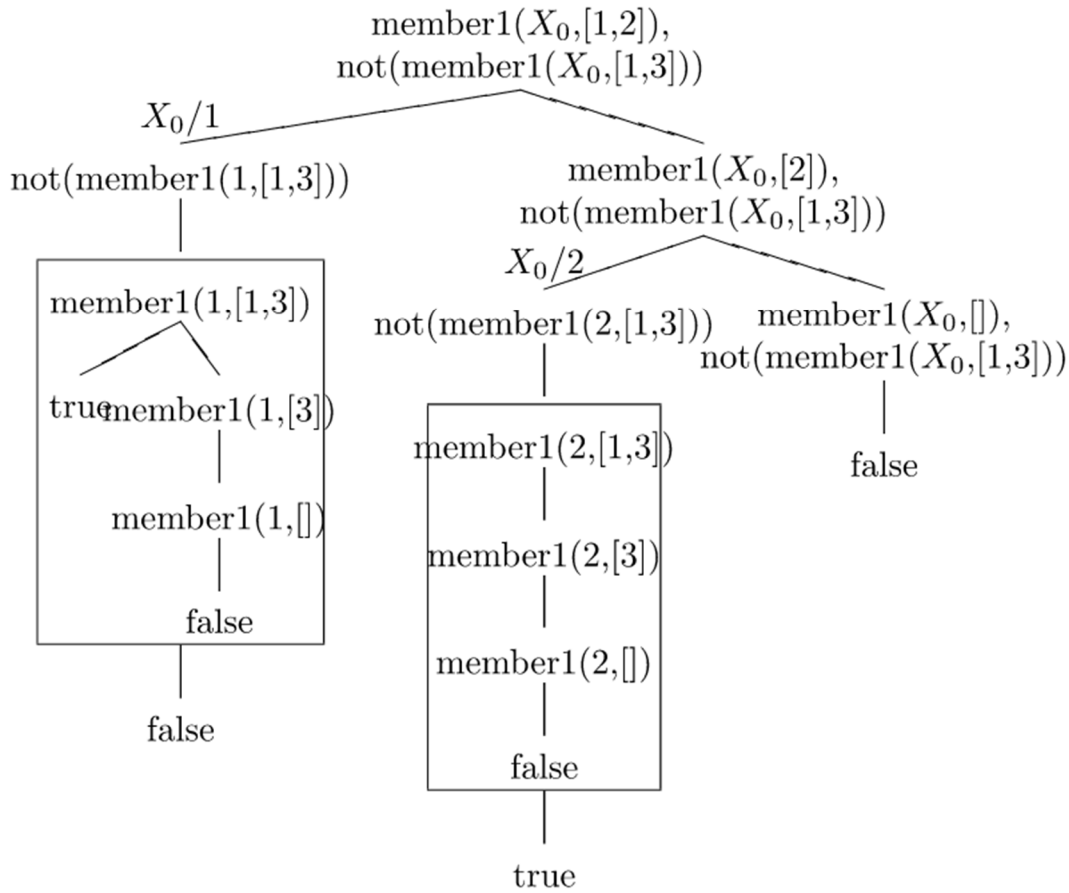


Alfa-beta:



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

### Esercizio 3



### Esercizio 4

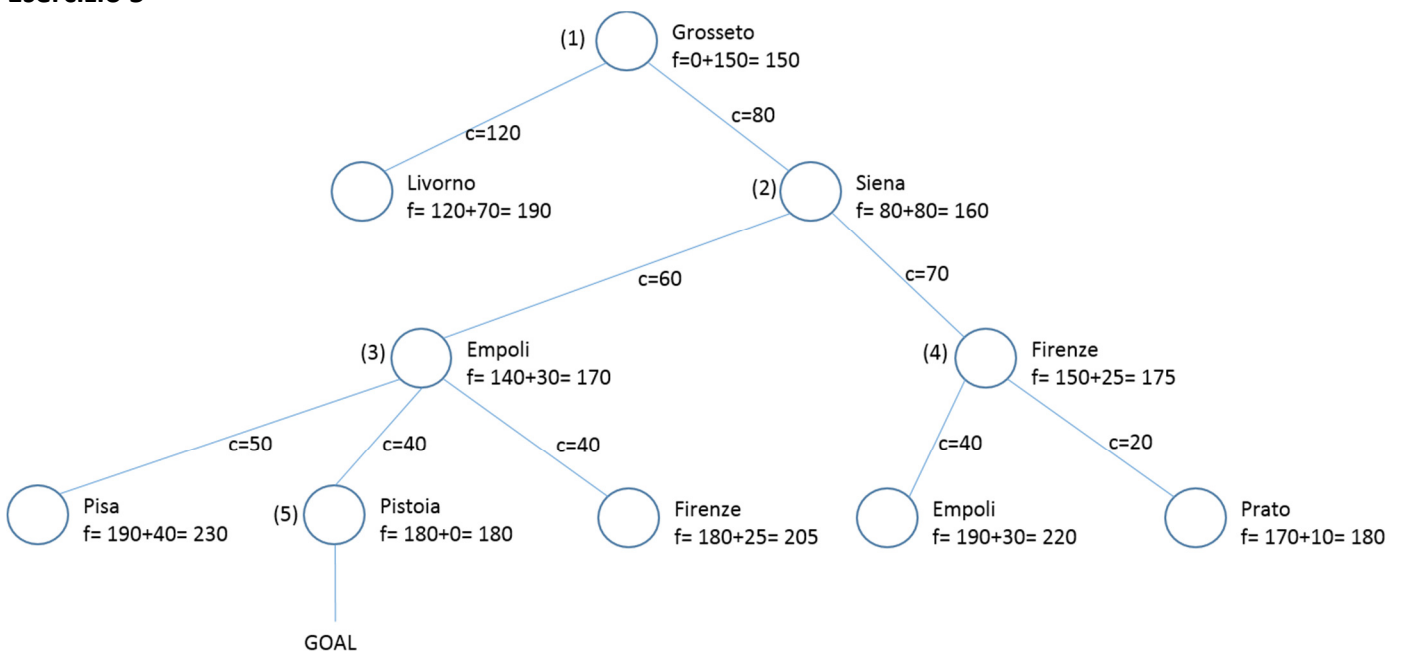
`ordered([ ])`.

`ordered([_])`.

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

Nota: si è interpretato che una lista vuota sia ordinata.

### Esercizio 5



L'euristica è ammissibile in quanto la stima euristica, per ogni nodo, è sempre minore o uguale della vera distanza dal goal (e quindi ottimista).

### Esercizio 6 Vedi slides