

**Esercizio 1 (6 punti)**

Si formalizzino le seguenti frasi in logica dei predicati:

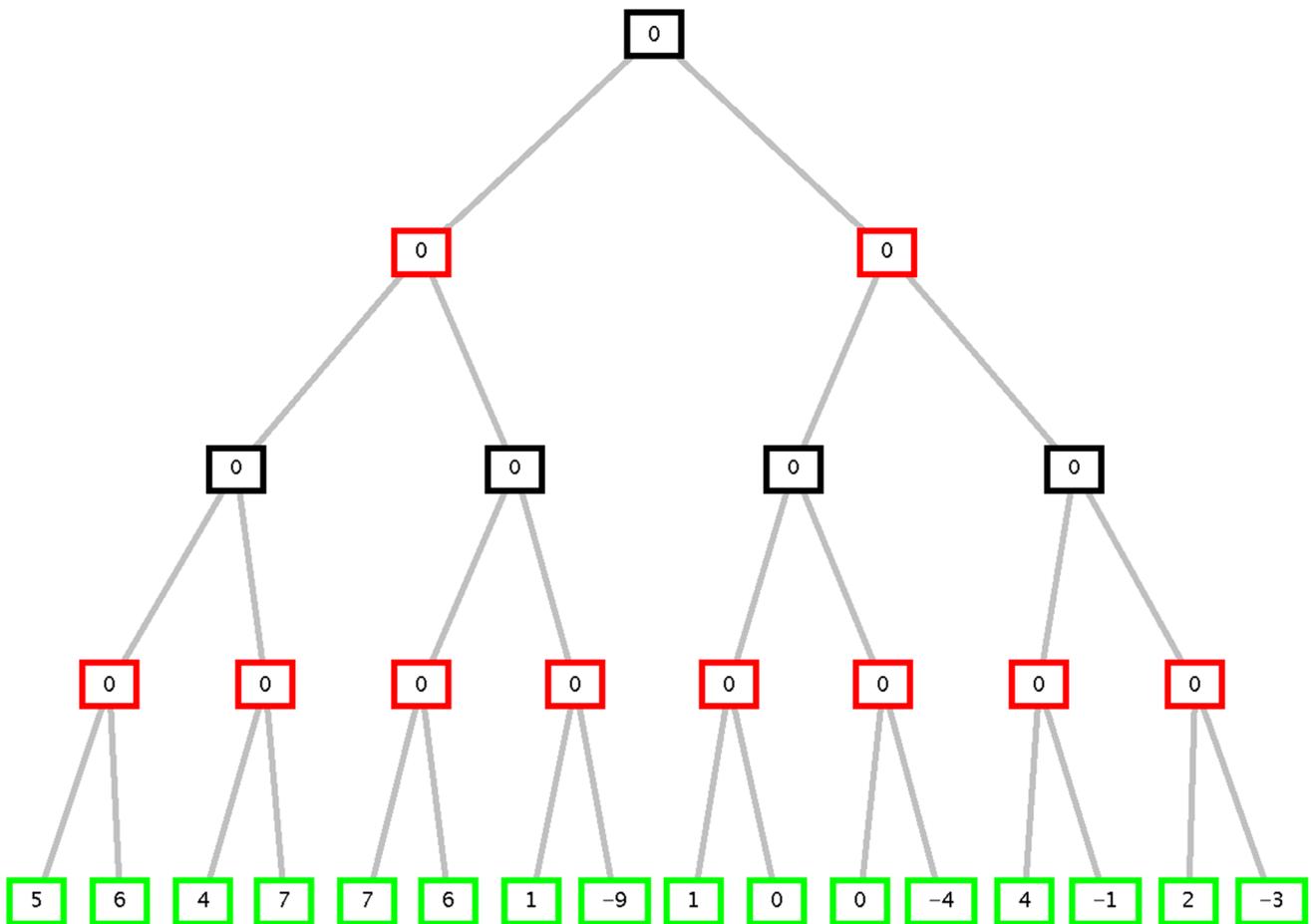
1. Ogni opera di Mozart è un capolavoro o (*or esclusivo*) è incompiuta
2. “Il flauto magico” è un’opera di Mozart e non è incompiuta
3. Esiste almeno un’opera di Mozart che è un capolavoro

Le si trasformi in clausole usando i seguenti predicati: **opera**(X, Y) (X è un’opera di Y), **capolavoro**(X) (X è un capolavoro), **incompiuta**(X) (X è incompiuta).

Si usi poi il principio di risoluzione per dimostrare che: “Il flauto magico” è un’opera di Mozart ed è un capolavoro.

**Esercizio 2 (6 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:

```

quadrati([], []).
quadrati([X|T], [X|T1]) :- isSquare(X), !, quadrati(T, T1).
quadrati(_|T, T1) :- quadrati(T, T1).
    
```

dove il predicato **isSquare**(X) ha successo se il termine X è legato a un intero che è un numero quadrato (si consideri **isSquare/1** come predefinito e non si espanda tale predicato nell’albero), si disegni l’albero SLDNF relativo al goal:

```

?- quadrati([3,4], L).
    
```

#### Esercizio 4 (4 punti)

Data una lista **L1** di interi si definisca un predicato Prolog **faiCoppie (L1, L2)** che data una lista **L1** di interi contenente un numero pari di elementi, restituisce la lista **L2** delle coppie  $c(X,Y)$ , dove X e Y sono elementi adiacenti nella lista **L1**.

Esempio:

```
?-faiCoppie ([1, 3, 4, 5], L) .
```

```
L=[c(1, 3), c(4, 5)]
```

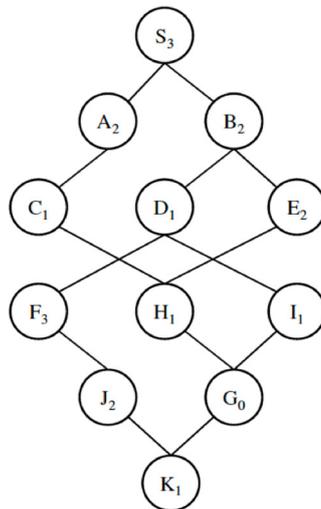
```
?-faiCoppie ([], L) .
```

```
L=[]
```

#### Esercizio 5 (6 punti)

Si consideri il problema di recarsi dalla città S alla città G, città che si trovano nella rete stradale indicata in figura.

La lettera in ogni nodo è il nome della città corrispondente e il pedice la stima euristica della distanza dal nodo goal G. Tutti gli archi hanno peso 1.



Si mostri come l'algoritmo A\* trova la soluzione, nelle ipotesi che non si generino loop e che i nodi siano scelti secondo l'ordine alfabetico a parità di altre condizioni. Si indichi a fianco di ogni nodo n, nello sviluppo dell'albero di ricerca, anche il costo stimato  $f(n)=g(n)+h(n)$ , e infine il costo totale del cammino trovato da A\*

#### Esercizio 6 (4 punti)

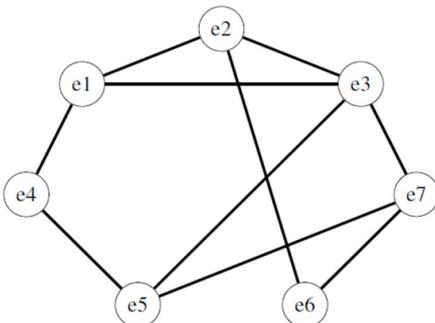
Si consideri il seguente problema di colorazione di sette regioni, e1-e7, con i colori Red, Green e Blu. La regione e2 è colorata di rosso, e la regione e7 di blu. Il problema è modellato come CSP con variabili:

e1, e3, e4, e5, e6 :: [Red,Green,Blu]

e2::[Red]

e7::[Blu]

e vincoli tutti di  $\neq$ , rappresentati dagli archi che legano due variabili nel seguente constraint graph (tutti i vincoli/archi sono di diverso  $\neq$ ):



Si applichi Arc Consistenza e si faccia vedere come variano i domini delle variabili.

**Esercizio 1**

C1:  $\forall X (\text{opera}(X,\text{mozart}) \rightarrow \text{capolavoro}(X) \text{ xor } \text{incompiuta}(X))$

C2:  $\text{opera}(\text{flautoMagico},\text{mozart}) \wedge \neg \text{incompiuta}(\text{flautoMagico})$

C3:  $\exists X (\text{opera}(X,\text{mozart}) \wedge \text{capolavoro}(X))$

Goal:  $\text{opera}(\text{flautoMagico},\text{mozart}) \wedge \text{capolavoro}(\text{flautoMagico})$

Clausole:

C1a  $\neg \text{opera}(X,\text{mozart}) \vee \text{capolavoro}(X) \vee \text{incompiuta}(X)$

C1b  $\neg \text{opera}(X,\text{mozart}) \vee \neg \text{capolavoro}(X) \vee \neg \text{incompiuta}(X)$

C2a  $\text{opera}(\text{flautoMagico},\text{mozart})$

C2b  $\neg \text{incompiuta}(\text{flautoMagico})$

C3a  $\text{opera}(c,\text{mozart})$

C3b  $\text{capolavoro}(c)$

GNeg:  $\neg \text{opera}(\text{flautoMagico},\text{mozart}) \vee \neg \text{capolavoro}(\text{flautoMagico})$

Risoluzione:

C4: GNeg + C2a:  $\neg \text{capolavoro}(\text{flautoMagico})$

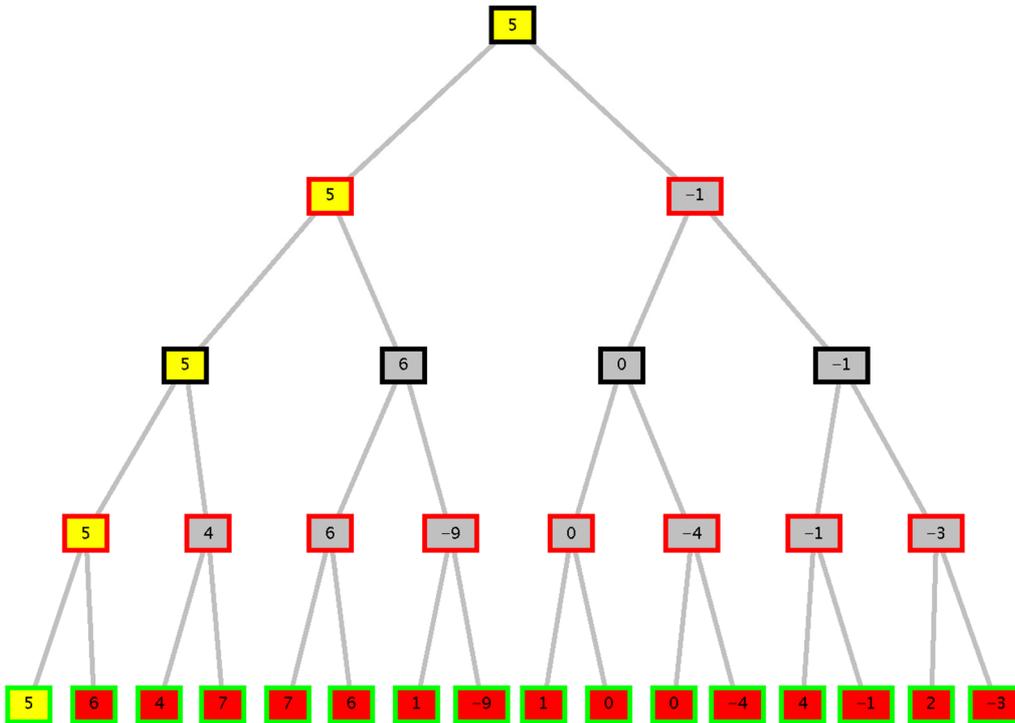
C5: C4 + C1a:  $\neg \text{opera}(\text{flautoMagico},\text{mozart}) \vee \text{incompiuta}(\text{flautoMagico})$

C5: C4 + C2a:  $\text{incompiuta}(\text{flautoMagico})$

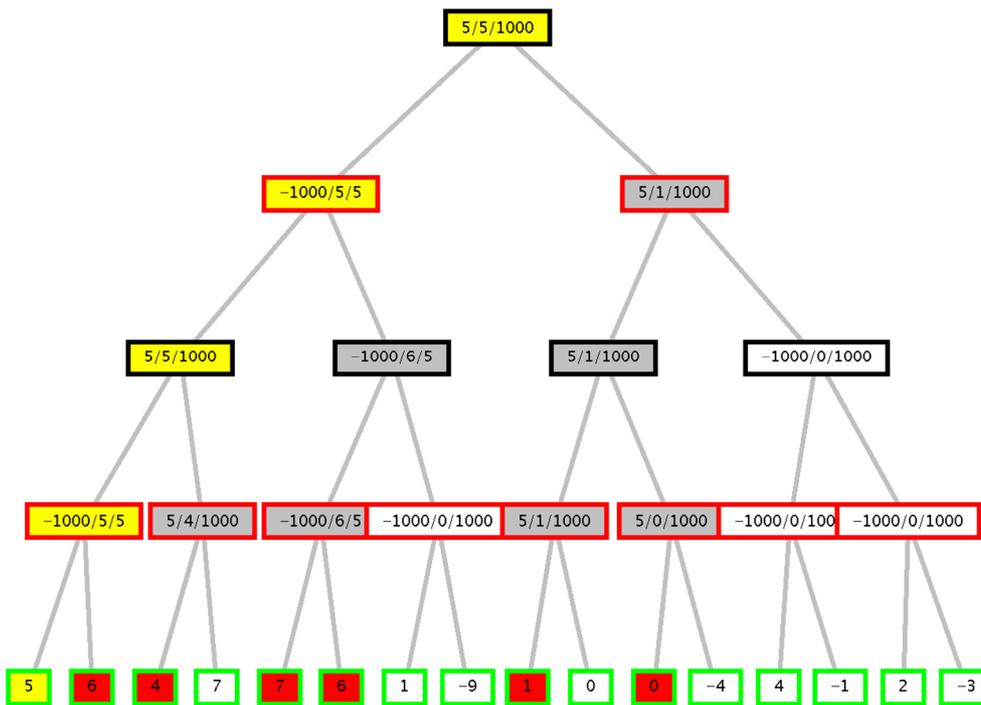
C6: C5 + C2b: clausola vuota.

## Esercizio 2

Min-Max:



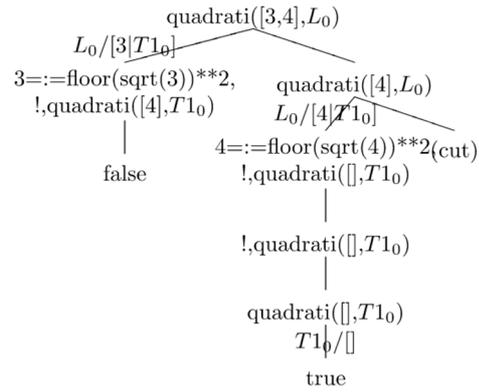
Alfa-beta:



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

### Esercizio 3

Si noti che nella soluzione il predicato `isSquare(X)` è stato sostituito con il test `X:=floor(sqrt(3))**2`.



### Esercizio 4

`faiCoppie([], []).`

`faiCoppie([X,Y|R],[c(X,Y)|T]) :- faiCoppie(R,T).`

### Esercizio 5

A\* search:  $f(n)=g(n)+h(n)$

S (0+3=3), A (1+2=3), B (1+2=3), C (2+1=3), D (2+1=3), E (2+2=4), H (3+1=4), G (4+0=4)

Cammino trovato: SACHG (si noti che potrebbe anche essere trovato il cammino SBEHG).

Costo: 4 (4 mosse)

### Esercizio 6

Applicazione di AC:

E1	E2	E3	E4	E5	E6	E7
B	R	G	G	R	G	B