

**NOTA:** Consegnare la soluzione in un singolo file word (e anche file pdf se si vuole), nominati con COGNOME-NOME

**Esercizio 1 (7 punti)**

Si traducano in logica dei predicati le seguenti frasi, e si dimostri poi tramite la regola di risoluzione che Babbo Natale porta i regali a qualcuno:

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 utilizzino i predicati:

- $bambino(X)$   $X$  è un bambino
- $buono(X)$   $X$  è buono
- $figlio(X,Y)$   $X$  è figlio di  $Y$
- $mamma(X)$   $X$  è mamma
- $porta\_regali(X, Y)$   $X$  porta regali a  $Y$

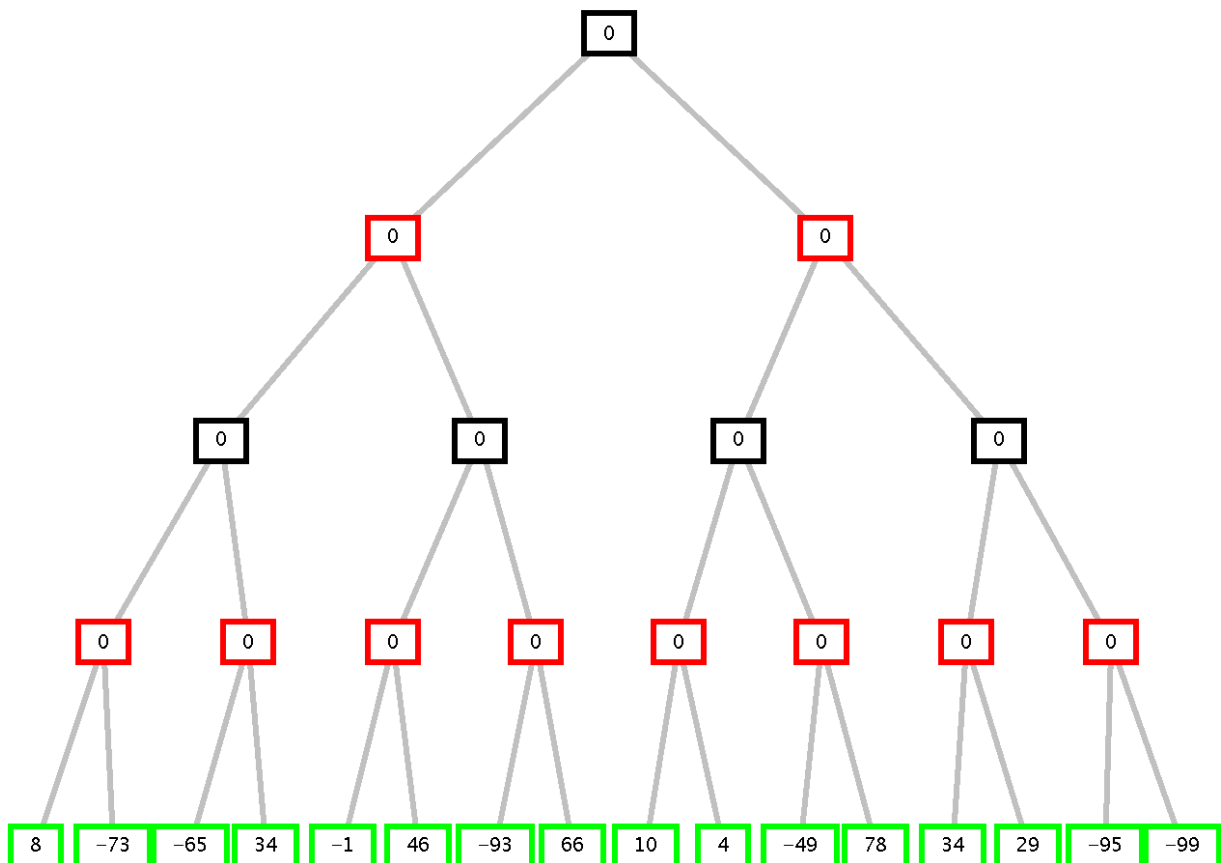
Si riportano i simboli degli operatori e quantificatori in logica:  $\forall \exists \wedge \vee \neg \rightarrow$

**Esercizio 2 (4 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.

Si mostrino poi i tagli che l’algoritmo **alfa-beta** consente indicando gli archi che saranno tagliati. Si indichino i nomi degli archi iniziando con la lettera “a” e facendola seguire con un numero crescente da sinistra a destra e dall’alto al basso. Ad esempio, i due archi che si dipartono dalla radice saranno nominati a1 (quello più a sinistra) e a2. L’arco che connette il nodo foglia più a sinistra (con valore -74) sarà denominato a15, mentre l’ultimo arco che connette il nodo foglia più a destra (valore 100) a30.

**NOTA:** oppure indicare il percorso min-max e il valore del nodo radice trovato con min-max e i tagli fatti da alfa-beta sulla figura copiata nel word che caricherete come soluzione.



### Esercizio 3 (7 punti)

Si consideri il seguente problema a vincoli

A::[1..10], B::[3..25], C::[2..22], D::[4..32], E::[1..4],

A > B

A >= C

B ≠ C

C = D

E < D.

Si applichi al problema l'algoritmo di arc-consistenza e si indichino i domini ridotti ottenuti per ogni variabile.

Una volta raggiunta la arc-consistenza, sul problema risultante si applichi la ricerca con propagazione forward checking ed euristica first fail fino a trovare la prima soluzione. Nel caso di uguale dimensione dei domini, le variabili si considerino nell'ordine seguente: A, B, C, D, E.

### Esercizio 4 (5 punti)

Si scriva un predicato PROLOG: listaMeno(L,B,L1) che, data la lista di numeri interi L, e il numero intero B, produca in uscita la lista L1 in cui per ogni elemento Y di L è inserito in L1 un elemento Y1 che vale Y-B se B < Y oppure B-Y se B >= Y.

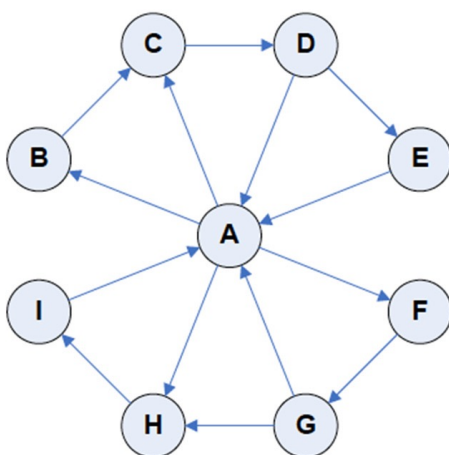
Ad esempio:

```
?- listaMeno([4,2,3,7],4,Z).
```

```
Z = [0, 2, 1, 3]
```

### Esercizio 5 (6 punti)

Si consideri il problema di ricerca dato in figura, dove lo **stato iniziale** è A e lo **stato obiettivo** è D. Ogni arco ha costo unitario.



Si risolva il problema mediante ricerca in profondità, senza eliminazione degli stati ripetuti. A parità di profondità, si preferisca l'espansione dei nodi in ordine alfabetico. Si riporti per ogni passo dell'algoritmo la lista dei nodi da espandere (frontiera) e il nodo selezionato per l'espansione. Si dica se la soluzione trovata è ottima, considerando il costo di ogni arco uguale a 1.

### Esercizio 6 (3 punti)

Si introduca brevemente l'architettura di un sistema basato sulla conoscenza e si sottolinei la sua differenza rispetto al concetto classico di algoritmo.

## Soluzioni:

### 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.  $\exists X, porta\_regali(babbo\_natale, X)$   
 $\sim Q. \sim porta\_regali(babbo\_natale, X)$

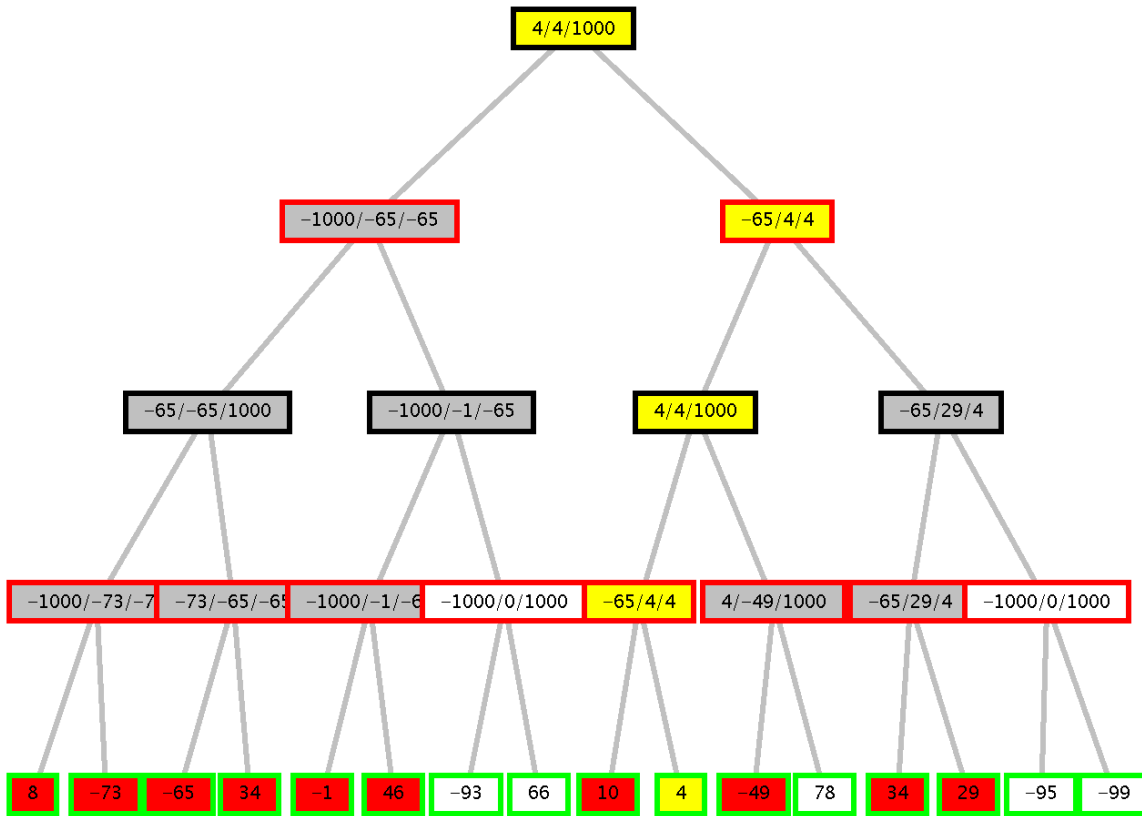
#### Risoluzione

- ( $\sim Q+1$ )  $\sim bambino(X) \vee \sim buono(X)$   
(+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 (primo nodo 4, cammino percorso nodi in giallo)

Alfa Beta



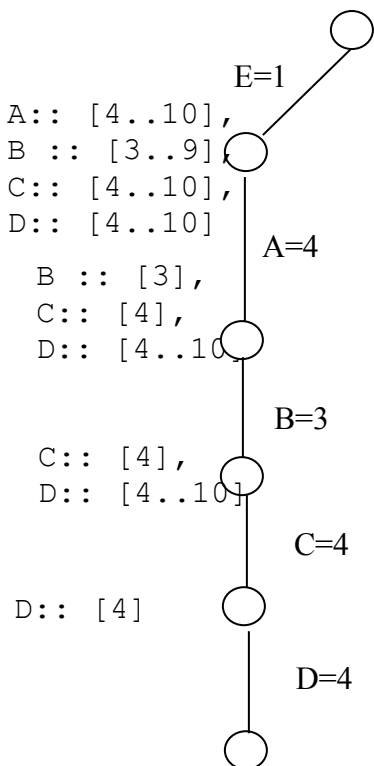
Archi tagliati: a10, a14, a26

### Esercizio 3

Rete Arc consistente

A:: [4..10], B :: [3..9], C:: [4..10], D:: [4..10], E[1..4]

Forward checking



#### Esercizio 4

```
listaMeno([],_,[]).
listaMeno([H|T],B,[H1|T1]):- B<H, !, H1 is H-B,
                               listaMeno(T,B,T1).
listaMeno([H|T],B,[H1|T1]):- H1 is B-H,
                               listaMeno(T,B,T1).
```

#### Esercizio 5

Frontiera:

A           ESPANDO A

BCFH       ESPANDO B.

CCFH       ESPANDO C

DCFH       SELEZIONO D test GOAL!

No, la soluzione ottima sarebbe ACD