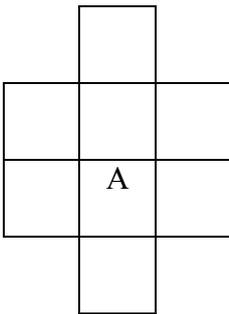


Il predicato `ground` restituisce vero se il suo argomento è `ground` (non ha variabili) e falso altrimenti. Si mostri l'albero SLD relativo al goal `calcola([c(X,Y+1),c(Y,2+3)],[])`. e si dica qual è la risposta calcolata. Che cosa sarebbe successo se invece avessimo invocato il corrispondente goal `X is Y+1, Y is 2+3`?

Esercizio 4 (punti 6)

Nel rompicapo delle lettere distanti l'obiettivo è scrivere le prime otto lettere dell'alfabeto (A, B, C, D, E, F, G e H) ciascuna in una casella della griglia indicata in figura. Ciascuna lettera deve comparire nella griglia una sola volta. Nessuna lettera seguente o precedente nell'alfabeto deve trovarsi in una casella adiacente in orizzontale, verticale o diagonale.



Si modelli il problema come CSP e lo si risolva utilizzando nella fase di ricerca la tecnica del forward checking, a partire dall'assegnamento indicato in figura (si propaghi da tale assegnamento in poi).

Nella ricerca, si consideri la seguente **euristica nella scelta delle variabili**. Nel caso di variabili con dominio avente un solo valore (deterministiche), queste vanno sempre considerate per prime, con ordine indifferente. Negli altri casi, si adotti il *most constrained principle* (considerando per prime le variabili coinvolte nel maggior numero di vincoli con altre variabili libere). A parità di numero di vincoli che le coinvolgono, le si consideri secondo l'ordinamento decrescente su indice di riga, e poi decrescente su indice di colonna a parità di numero di riga.

Come **euristica per il valore da assegnare**, si scelga il valore da assegnare partendo da uno dei due estremi dell'intervallo di valori disponibili, assegnando alle variabili di indice di colonna dispari il primo valore disponibile nel dominio (ordinato alfabeticamente) e a quelle di indice di colonna pari l'ultimo valore disponibile nel dominio.

Esercizio 5 (punti 4)

Definire il predicato Prolog: `no_common_elements(List1,List2)` che date in ingresso due liste `List1` e `List2` sia vero se `List1` e `List2` non hanno alcun elemento in comune.

Esempio:

```
?- no_common_elements([a,b,c],[e,g,f,k]).
```

Yes

```
?- no_common_elements([], [e,g,f,k]).
```

Yes

```
?- no_common_elements([a,b,e],[e,g,f,k]).
```

No

Esercizio 6 (punti 3)

Discutere le caratteristiche della strategia di ricerca iterative deepening (anche in termini di completezza, ottimalità e costo computazionale) e darne una descrizione in pseudo codice.

SOLUZIONE:

Esercizio 1

Ogni bambino ama Babbo Natale.

Chiunque ami Babbo Natale ama ogni renna.

Rudy è una renna ed ha il naso rosso.

Chiunque abbia un naso rosso o è bizzarro o è un clown.

Chiunque sia bizzarro non è un clown.

Pippo non ama nessuno che sia bizzarro (chiunque sia bizzarro non è amato da Pippo).

GOAL: Pippo non è un bambino.

1. $\forall X (\text{bimbo}(X) \rightarrow \text{ama}(X, \text{babbon}))$
2. $\forall X (\text{ama}(X, \text{babbon}) \rightarrow \forall Y (\text{renna}(Y) \rightarrow \text{ama}(X, Y)))$
3. $\text{renna}(\text{rudy})$ and $\text{nasorosso}(\text{rudy})$
4. $\forall X (\text{nasorosso}(X) \rightarrow \text{bizzarro}(X) \text{ or } \text{clown}(X))$
5. $\forall X (\text{bizzarro}(X) \rightarrow \neg \text{clown}(X))$
6. $\forall X (\text{bizzarro}(X) \rightarrow \neg \text{ama}(\text{pippo}, X))$
7. GOAL NEGATO: $\neg \text{bimbo}(\text{pippo})$

Clausole:

1. $\neg \text{bimbo}(X) \text{ or } \text{ama}(X, \text{babbon})$
2. $\neg \text{ama}(X, \text{babbon}) \text{ or } (\neg \text{renna}(Y) \text{ or } \text{ama}(X, Y))$
3. $\text{renna}(\text{rudy})$
4. $\text{nasorosso}(\text{rudy})$
5. $\neg \text{nasorosso}(X) \text{ or } \text{bizzarro}(X) \text{ or } \text{clown}(X)$
6. $\neg \text{bizzarro}(X) \text{ or } \neg \text{clown}(X)$
7. $\neg \text{bizzarro}(X) \text{ or } \neg \text{ama}(\text{pippo}, X)$
8. GOAL NEGATO: $\text{bimbo}(\text{pippo})$

Risoluzione:

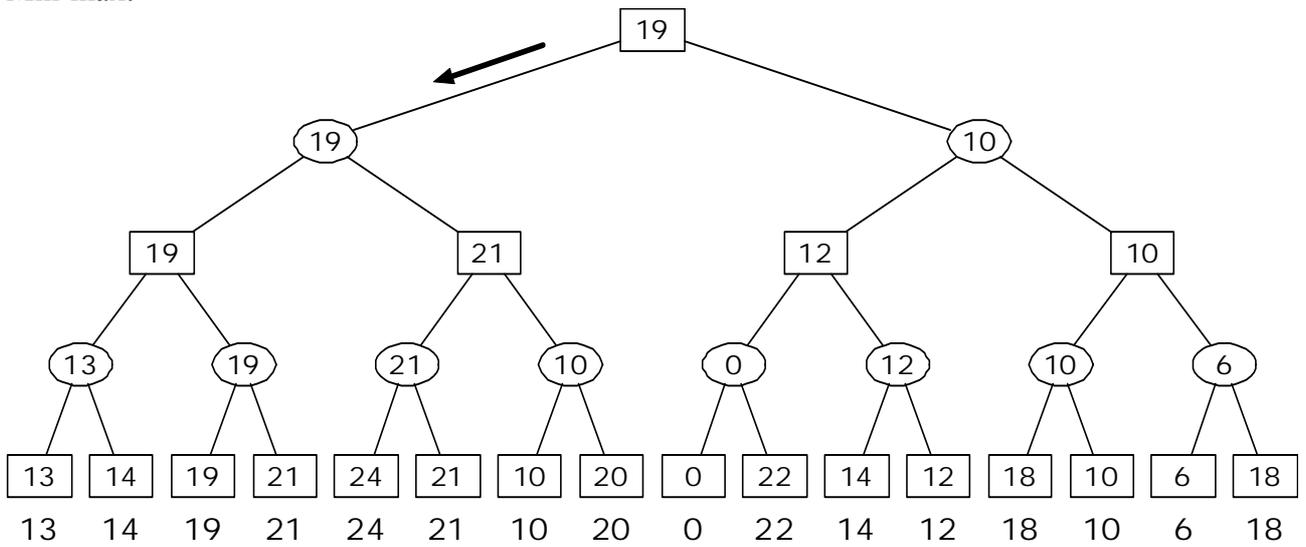
9. $\text{ama}(\text{pippo}, \text{babbon})$ da 8+1
10. $\neg \text{renna}(Y) \text{ or } \text{ama}(\text{pippo}, Y)$ da 9+2
11. $\text{ama}(\text{pippo}, \text{rudy})$ da 10+3
12. $\neg \text{bizzarro}(\text{rudy})$ da 11+7
13. $\text{bizzarro}(\text{rudy}) \text{ or } \text{clown}(\text{rudy})$ da 4+5
14. $\text{clown}(\text{rudy})$ da 13+12
15. $\neg \text{bizzarro}(\text{rudy})$ da 14+6
16. $\neg \text{nasorosso}(\text{rudy}) \text{ or } \text{clown}(\text{rudy})$ da 15+5
17. $\text{clown}(\text{rudy})$ da 16+4

...

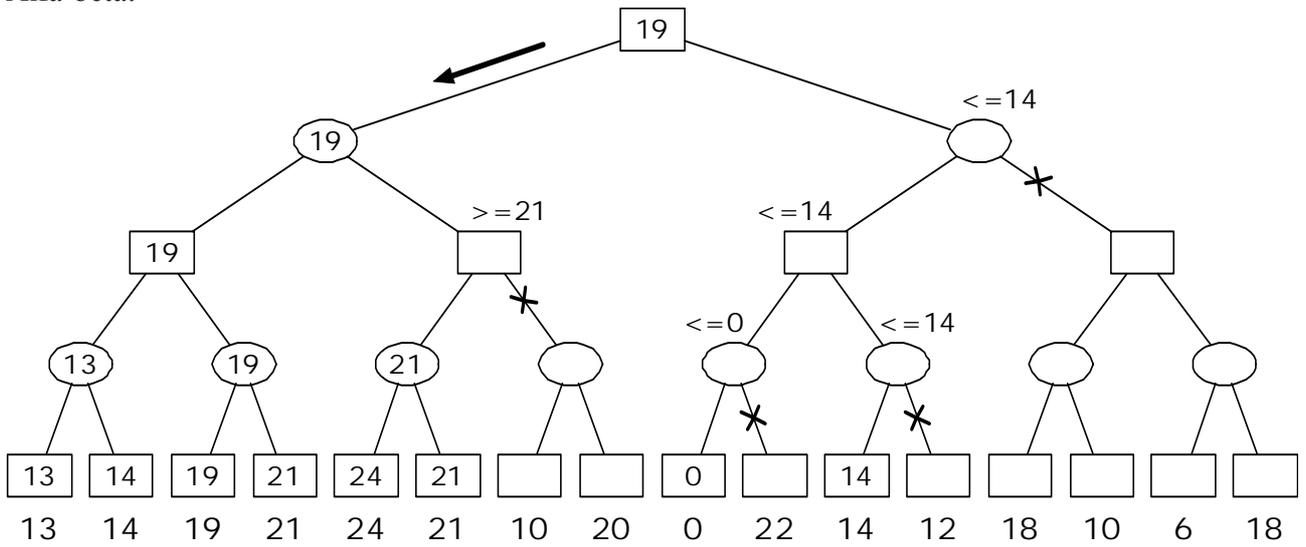
Non è possibile derivare la clausola vuota, quindi il goal iniziale non è dimostrabile.

Esercizio 2

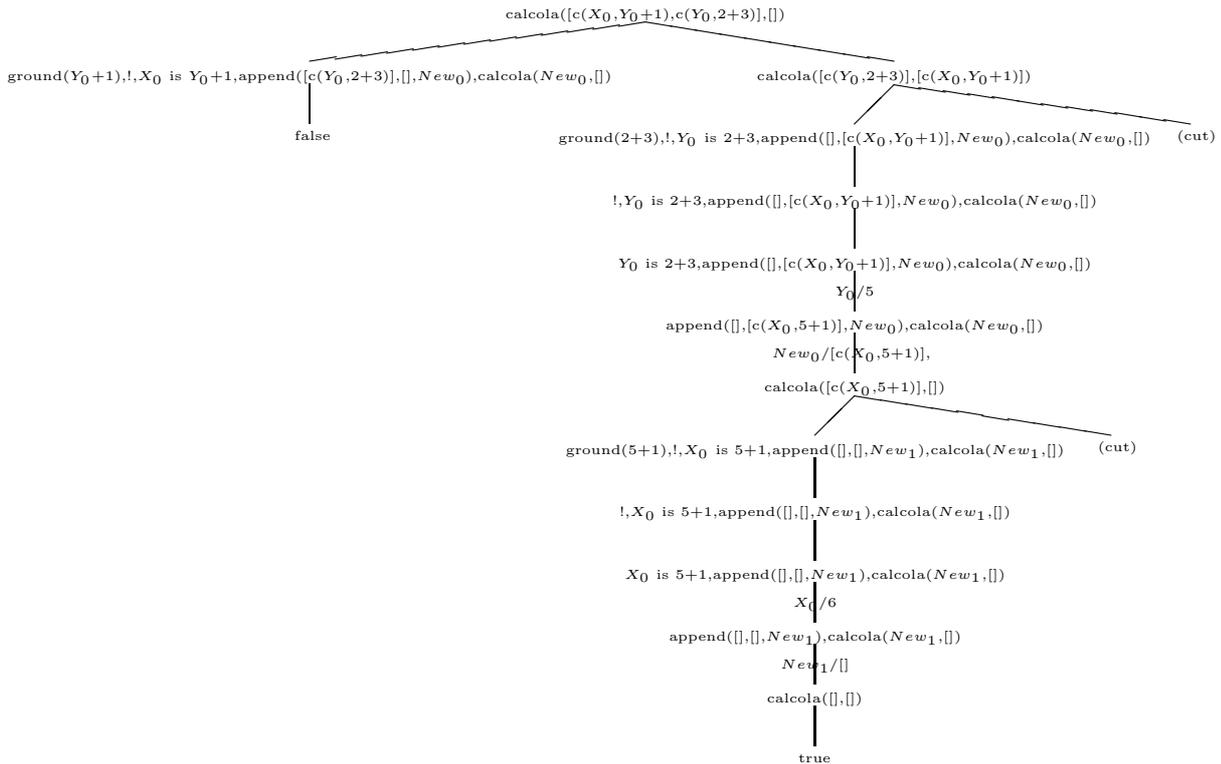
Min-max:



Alfa-beta:



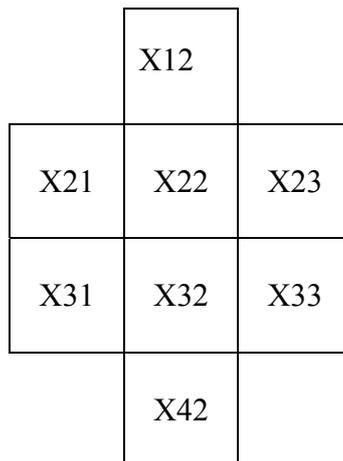
Esercizio 3



La risposta calcolata è $X/6, Y/5$, il programma ha quindi eseguito prima i sottogol in cui il secondo argomento era `ground` e successivamente gli altri. Se avessimo invocato i goal nell'ordine dato (ovvero con il goal `X is Y+1, Y is 2+3`), avremmo avuto un errore, in quanto la prima `is` ha l'argomento non istanziato.

Esercizio 4

Variabili:



Dominii: [A, B, C, D, E, F, G, H]

- Vincoli:
- $X_{ij} \neq X_{kj}$ per $i=k+1, k-1$
 - $X_{ij} \neq X_{ik}$ per $j=k+1, k-1$
 - $X_{ij} \neq X_{i+1,j+1}$
 - $X_{ij} \neq X_{i+1,j-1}$

Passo 1:

X32 con valore A. Si applica il FC:

	[B, C, D, E, F, G, H]	
[C, D, E, F, G, H]	[C, D, E, F, G, H]	[C, D, E, F, G, H]
[C, D, E, F, G, H]	A	[C, D, E, F, G, H]
	[C, D, E, F, G, H]	

Passo 2:

X22 è coinvolta in 5 vincoli; l'euristica la sceglie e come valore sceglie l'ultimo del dominio (H). Si applica il FC:

	[B, C, D, E, F]	
[C, D, E, F]	H	[C, D, E, F]
[C, D, E, F]	A	[C, D, E, F]
	[C, D, E, F, G]	

Passo 3:

Tutte le variabili libere sono a questo punto coinvolte in due vincoli; l'euristica sceglie X42 e come valore sceglie l'ultimo del dominio (G). Si applica il FC:

	[B, C, D, E, F]	
[C, D, E, F]	H	[C, D, E, F]
[C, D, E]	A	[C, D, E]
	G	

Passo 4:

X12, X21 e X23 sono ciascuna coinvolta in due vincoli con variabili libere. L'euristica sceglie X23 e sono è coinvolta in 2 vincoli; l'euristica la sceglie e come valore sceglie l'ultimo del dominio (F). Si applica il FC:

	[B, C, D]	
[C, D, E]	H	F
[C, D, E]	A	[C, D]
	G	

Passo 5:

X21 è coinvolta in 2 vincoli; l'euristica la sceglie e come valore sceglie l'ultimo del dominio (E). Si applica il FC:

	[B, C]	
E	H	F
[C]	A	[C, D]
	G	

