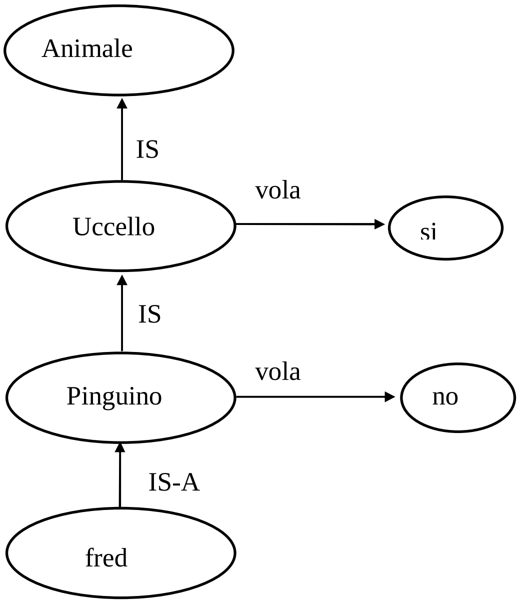
FONDAMENTI DI INTELLIGENZA ARTIFICIALE

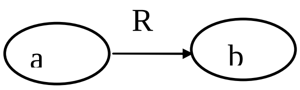
10 Luglio 2009 – Tempo a disposizione 2h – Risultato 32/32 punti

Esercizio 1 (punti 6)

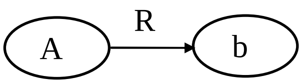
Data la seguente rete semantica:



darne una rappresentazione in logica dei predicati del I ordine, ricordando che:



R(a,b) se a è un'istanza e b un valore



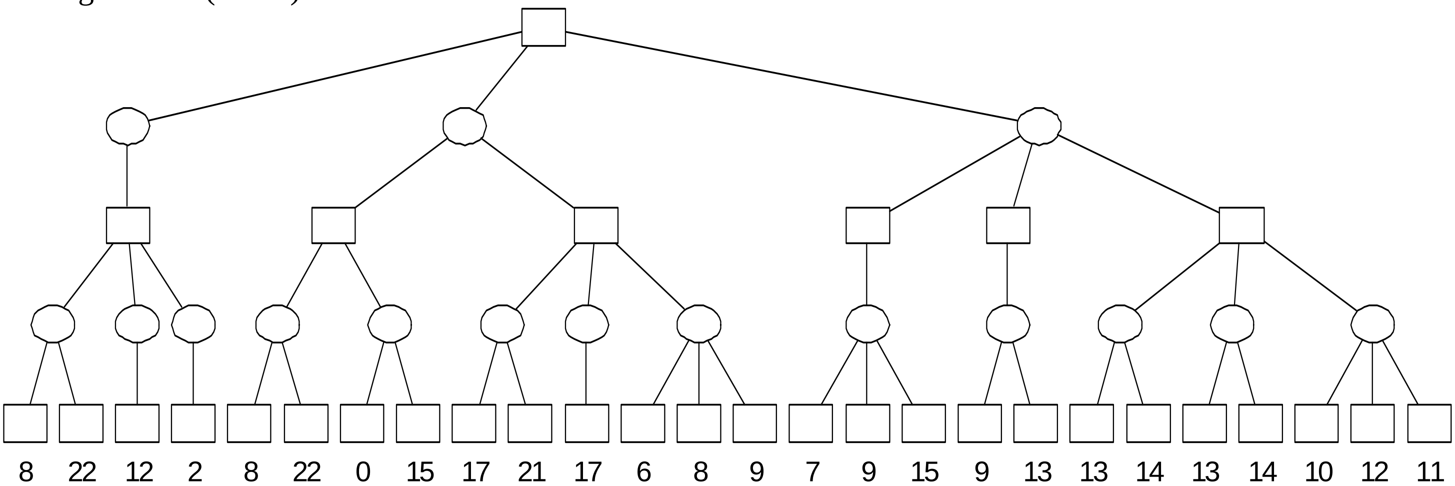
$\forall x\ A(x) \Rightarrow R(x,b)$ se A è una classe e b un valore

e che le relazioni IS e IS-A hanno una lettura particolare (sottoinsieme e appartenenza).

a. Dire quale sarebbe la risposta alla domanda "fred vola?", con il meccanismo deduttivo proprio delle reti semantiche.
b. Dire quale sarebbe la risposta alla stessa query nella teoria logica corrispondente.
c. Come è formalizzabile tale teoria logica in Prolog in modo da ottenere il corretto funzionamento?

Esercizio 2 (punti 7)

Si consideri il seguente albero di gioco, dove i punteggi sono tutti dal punto di vista del primo giocatore (MAX):



Si mostri come l'algoritmo min-max sceglie la mossa migliore. Si mostrino poi i tagli alfa-beta.

Esercizio 3 (punti 6)

Si mostri l'albero SLDNF relativo al goal:

```
?-member(X,[p(Y),q]),not(member(X,[p(q)]).
```

utilizzando la definizione standard (senza cut) per il predicato `member`:

```
member(H,[H|_]).
member(H,[_|T]):-member(H,T).
```

Esercizio 4 (punti 6)

Il seguente problema rientra nella categoria dei 'quadrati magici'. Si tratta di assegnare ad ognuna delle 8 casella vuote una cifra da 0 a 3 in modo che la somma delle cifre in orizzontale verticale e diagonale sia come indicato in figura dai numeri fuori dal quadrato.

			0
X1	X2	X3	3
X4	**0**	X5	3
X6	X7	X8	3
3	3	3	3

a) Si imposti il problema come un problema di soddisfacimento di vincoli, indicando chiaramente quali sono le variabili, i relativi domini e i vincoli.
b) Si risolva il problema con **euristica MRV** e **forward checking** partendo dalla configurazione indicata in figura. Nel caso di più variabili selezionabili con MRV, si scelga in base all'ordine crescente del pedice.

Esercizio 5 (punti 5)

Data una matrice `M` (come lista di liste) si scriva un programma Prolog per un predicato `constant(M,C)` che è vero se `C` è un numero che compare in ciascuna riga della matrice `M`.

<u>Esempio:</u>

```
?- constant (
  [[4, 9, 23, 55, 63, 107, 239],
   [5, 9, 31, 55, 60, 73, 82, 99, 107],
   [9, 23, 55, 107, 128, 512],
   [6, 9, 13, 17, 22, 55, 63, 107 ]], 9).
Yes
?- constant (
  [[4, 9, 23, 55, 63, 107, 239],
   [5, 9, 31, 55, 60, 73, 82, 99, 107],
   [9, 23, 55, 107, 128, 512],
   [6, 9, 13, 17, 22, 55, 63, 107 ]], 60).
No
```

Inoltre dire che cosa produce la chiamata:

```
?-constant1(
  [[4, 9, 23, 55, 63, 107, 239],
   [5, 9, 31, 55, 60, 73, 82, 99, 107],
   [9, 23, 55, 107, 128, 512],
   [6, 9, 13, 17, 22, 55, 63, 107 ]], CC).
```

per il predicato così definito::

```
constant1(M,Cols):-findall(Col, constant(M,Col), Cols).
```

e si spieghi il significato del predicato predefinito `findall`.

Esercizio 6 (punti 2)

Si consideri il problema del *route-finding* (trovare un percorso su una mappa, tipo quella della Romania) e si supponga per semplicità che le città si trovino su una griglia le cui celle sono quadrate e di lato unitario. Si usi come euristica la distanza dalla destinazione calcolata usando la "Distanza di Manhattan" (tale distanza coincide con la somma delle distanze in orizzontale ed in verticale per raggiungere l'obiettivo) Si discuta se questa euristica è ammissibile, motivando opportunamente la risposta.

SOLUZIONE:

Esercizio 1

Teoria FOL corrispondente alla rete semantica:

1. 1. ∀x Uccello(x) ⇒ Animale(x)
2. ∀x Pinguino(x) ⇒ Uccello(x)
3. Pinguino (fred)
4. ∀x Uccello(x) ⇒ Vola(x,si)
5. ∀x Pinguino(x) ⇒ Vola(x,no)

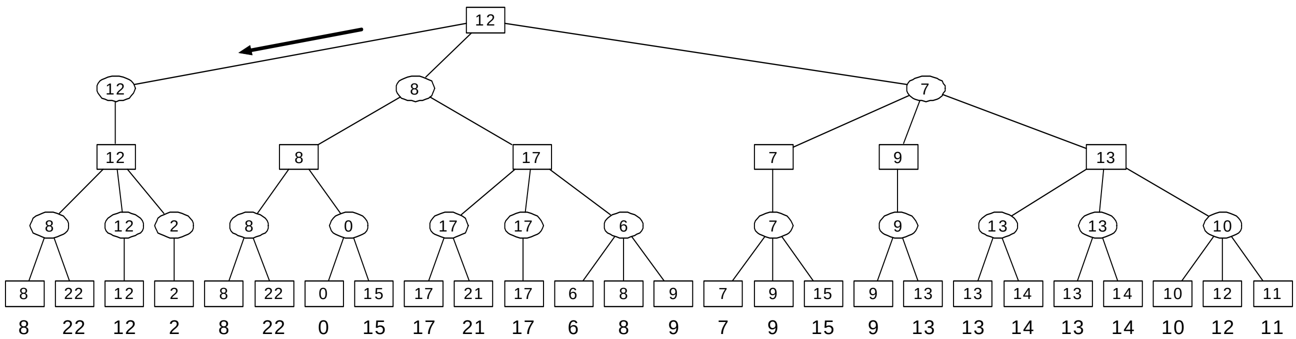
a. La risposta alla domanda “Fred vola?”, con il meccanismo deduttivo proprio delle reti semantiche, è NO.
b. La risposta nella teoria logica corrispondente è sia SI’ sia NO.
c. Tale logica e’ traducibile in Prolog?

Si potrebbe utilizzare una traduzione in Prolog usando predicati extra-logici o la negazione per default per realizzare il corretto funzionamento, ad esempio per la clausola 6.:

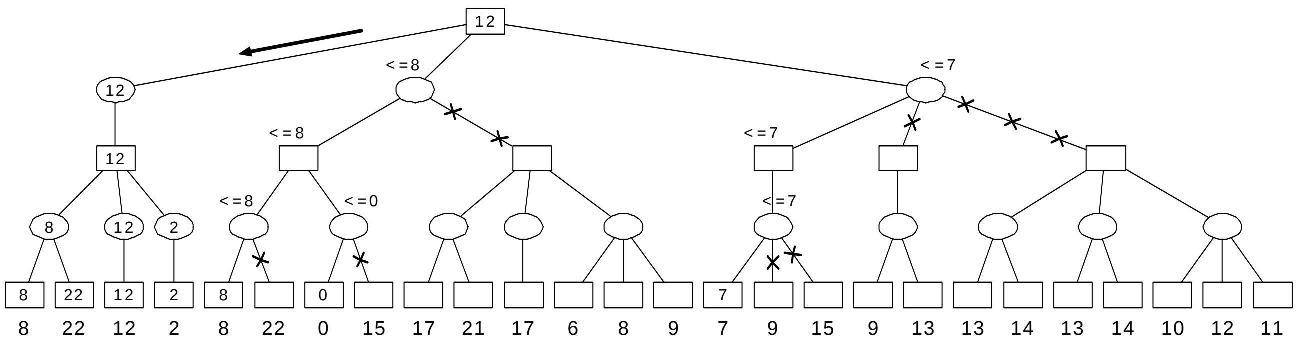
vola(X,si) :- uccello(X), not pinguino(X).

Esercizio 2

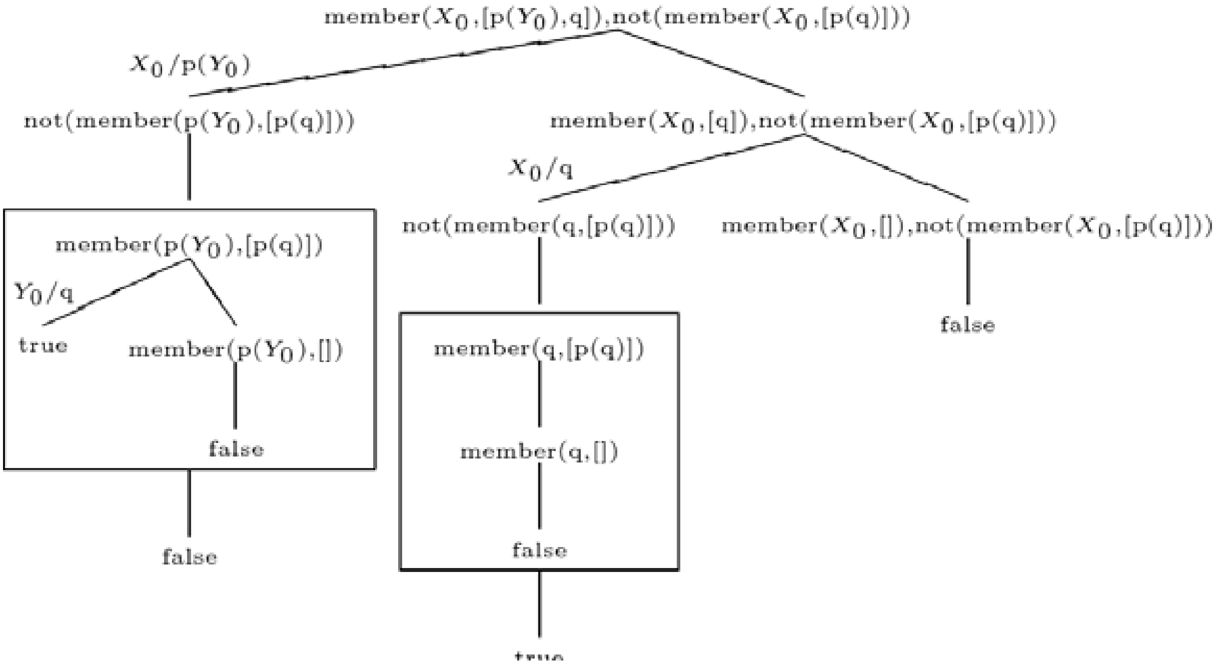
Min-max:



Alfa-beta:



Esercizio 3



Esercizio 4

a) Varibili X1, X2, X3, X4, X5, X6, X7, X8
Dom(Xi) = {0, 1, 2, 3}
Vincoli:

X1 + X2+ X3 = 3   X4 + X5 = 3
X6 + X7+ X8 = 3   X2 + X7 = 3
X1 + X4+ X6 = 3   X1 + X8 = 3
X3 + X5+ X8 = 3   X3 + X6 = 0

Risoluzione con MRV + Euristica del grado per la scelta delle variabili (e a parità di scelte, per numero di pedice)+ FC
Passo1. Scelgo una delle variabili più vincolate: per esempio X1=0. Faccio FC.

			0
0	{0,1,2,3}	{0,1,2,3}	3
{0,1,2,3}	**0**	{0,1,2,3}	3
{0,1,2,3}	{0,1,2,3}	{3}	3
3	3	3	3

Passo2. Asssegno X8=3. Faccio FC.

			0
0	{0,1,2,3}	{0}	3
{0,1,2,3}	**0**	{0}	3
{0}	{0}	3	3
3	3	3	3

Passo3. Assegno X3=0. Faccio FC.

			0
0	{3}	0	3
{0,1,2,3}	**0**	{0}	3
{0}	{0}	3	3
3	3	3	3

Passo 4. Assegno X2=3. Faccio FC.

			0
0	3	0	3
{0,1,2,3}	**0**	{0}	3
{0}	{0}	3	3
3	3	3	3

A questo punto sono scelte obbligate fino alla fine. Si istanziano prima X5=0 (l'applicazione del FC porta il dominio di X4 a {3}), poi X4=3, poi X6=0, etc. La soluzione finale è:

			0
0	3	0	3
3	**0**	0	3
0	0	3	3
3	3	3	3

Esercizio 5

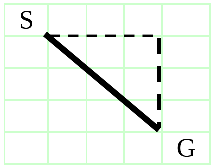
```
constant([], _).
constant([R|Rs], C) :-
    member(C, R),
    constant(Rs, C).

member(H,[H|_]).
member(H,[_|T]) :-
    member(H,T).

constant1(M,Cols) :-
     findall(Col, constant(M,Col), Cols).

?- constant1 (
   [[4, 9, 23, 55, 63, 107, 239],
    [5, 9, 31, 55, 60, 73, 82, 99, 107],
    [9, 23, 55, 107, 128, 512],
    [6, 9, 13, 17, 22, 55, 63, 107 ]], CC).
Yes CC = [ 9, 55, 107 ]
```

Il predicato **findall(X,P,S)** che è vero se **S** è la lista delle istanze **X** (senza ripetizioni) per cui la proprietà **P** è vera. In questo caso **P** potrebbe essere il predicato `constant(M,C)` che è vero se `C` è un numero che compare in ciascuna riga della matrice `M`

Esercizio 6

a. No. L'euristica non è ammissibile. Nella seguente situazione

h(S) = 6
h*(S) = sqrt (9+9) = 4,24


S
G