

**COMPITO DI FONDAMENTI DI INTELLIGENZA ARTIFICIALE**  
**INTELLIGENZA ARTIFICIALE (v.o.) – PARTE I**

**3 Aprile 2008 (Tempo a disposizione 2h ; su 32 punti)**

**Esercizio 1 (punti 5)**

Date le seguenti frasi:

- I bambini viziati sono irrequieti.
- Se un bambino irrequieto ha una mamma insofferente, o viene punito dalla mamma o viene lasciato a casa (or non esclusivo).
- Alice, madre di Carletto, è una mamma insofferente
- Carletto è un bambino viziato.
- Carletto non viene punito dalla mamma.

rappresentarle in modo opportuno in logica come formule del primo ordine; trasformarle in forma a clausole e dimostrare applicando il principio di risoluzione che Carletto viene lasciato a casa.

**Esercizio 2 (punti 6)**

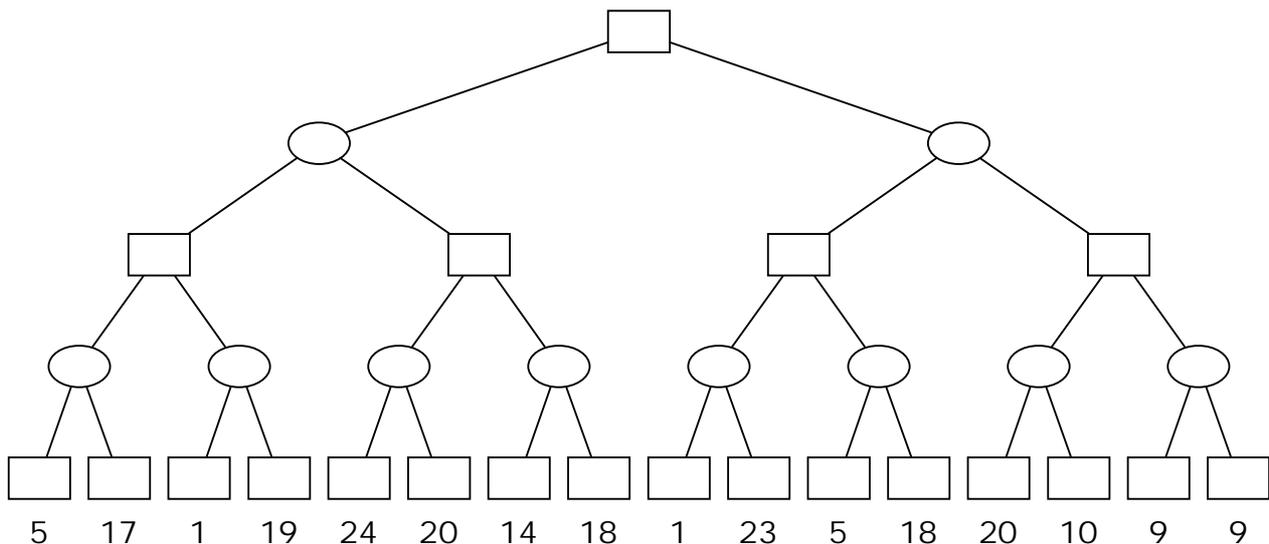
Si consideri la seguente base di conoscenza Prolog:

<pre>apr03 :- sconnesso, !.  sconnesso :- nodo(X), nodo(Y), not(cammino(X,Y)).  cammino(X,X) :- !. cammino(X,Z) :- arco(X,Y), cammino(Y,Z).</pre>	<pre>nodo(a). nodo(b). nodo(c). nodo(d). nodo(e). arco(a,b). arco(b,e). arco(c,d).</pre>
---	--

Si disegni l'albero di derivazione SLDNF relativo al goal ?- apr03.

**Esercizio 3 (punti 6)**

Si consideri il seguente albero di gioco:



dove i punteggi sono dal punto di vista del primo giocatore (Max). Si mostri come l'algoritmo min-max risolve il problema. Si mostrino poi i tagli alfa-beta.

**Esercizio 4 (punti 8)**

Sia data una scacchiera di dimensione 4 x 4 ed un insieme di 4 colori {R,B,G,Y}. Si deve

collocare un colore in ciascuna cella della scacchiera in modo che ogni riga, ogni colonna e le due diagonali principali della scacchiera contengano colori diversi. Formalizzare il problema come CSP. In particolare:

- individuare una variabile per ogni casella della scacchiera;
- per ciascuna variabile introdotta, specificare il dominio dei valori ammissibili;
- definire un insieme di vincoli opportuno tra le variabili introdotte;
- risolvere il problema fino alla prima soluzione, usando la tecnica del forward checking con euristica first-fail per la selezione delle variabili da istanziare. Nel caso di variabili con uguale dimensione del dominio si proceda per riga.

### **Esercizio 5 (punti 4)**

Si scriva un programma Prolog con due predicati dedicati a calcolare la somma e la media di tutti gli elementi di una lista di interi (il secondo dei due utilizza il risultato del primo)

Esempi:

```
?- somma([4,5,3,6], S) yes S=18
```

```
?- media([4,5,3,7], M) yes M=4.25
```

### **Esercizio 6 (punti 3)**

Si spieghi cosa si intende per algoritmo di ricerca locale ed il suo funzionamento, quando e' utile applicarlo ed i suoi limiti.

## SOLUZIONE

### Esercizio 1

#### Vocabolario:

Simboli di costante

Alice

Carletto

Simboli di predicato unari

bimbo\_viziato(x)

bimbo\_irrequieto(x)

mamma\_insofferente(x)

lasciatoacasa(x)

Simboli di predicato binari

mamma(x; y)

punito(x; y)

#### Rappresentare le frasi in modo opportuno come formule del primo ordine.

- I bambini viziati sono irrequieti.
- Se un bambino irrequieto ha una mamma insofferente, o viene punito dalla mamma o viene lasciato a casa.
- Alice, madre di Carletto, è una mamma insofferente
- Carletto è un bambino viziato, ma non viene punito dalla mamma.

(1) bimbo\_viziato(Carletto)

(2) mamma\_insofferente(Alice)

(3) mamma(Alice;Carletto)

(4)  $\forall x$  bimbo\_viziato(x)  $\Rightarrow$  bimbo\_irrequieto(x)

(5)  $\forall x \forall y$  [(bimbo\_irrequieto(x) and mamma(y,x) and mamma\_insofferente(x))  $\Rightarrow$  (punito(x, y) or lasciatoacasa(x))]

(6) not punito(Carletto, Alice)

#### Dedurre con la risoluzione che Carletto viene lasciato a casa.

Nego la tesi e trasformo in clausole:

(1) bimbo\_viziato(Carletto)

(2) mamma\_insofferente(Alice)

(3) mamma(Alice,Carletto)

(4) not bimbo\_viziato(x) or bimbo\_irrequieto(x)

(5) not bimbo\_irrequieto(x) or not mamma(y, x) or not mamma\_insofferente(x) or punito(x, y)

Or lasciatoacasa(x)

(6) not punito(Carletto; Alice)

(7) not lasciatoacasa(Carletto)

#### poi risolvo:

(8) bimbo\_irrequieto(Carletto) da (1) e (4) con x/Carletto

(9) not mamma(Alice,Carletto) or not mamma\_insofferente(Alice) or punito(Carletto, Alice) or lasciatoacasa(Carletto) da (8) e (5) con con y/Alice

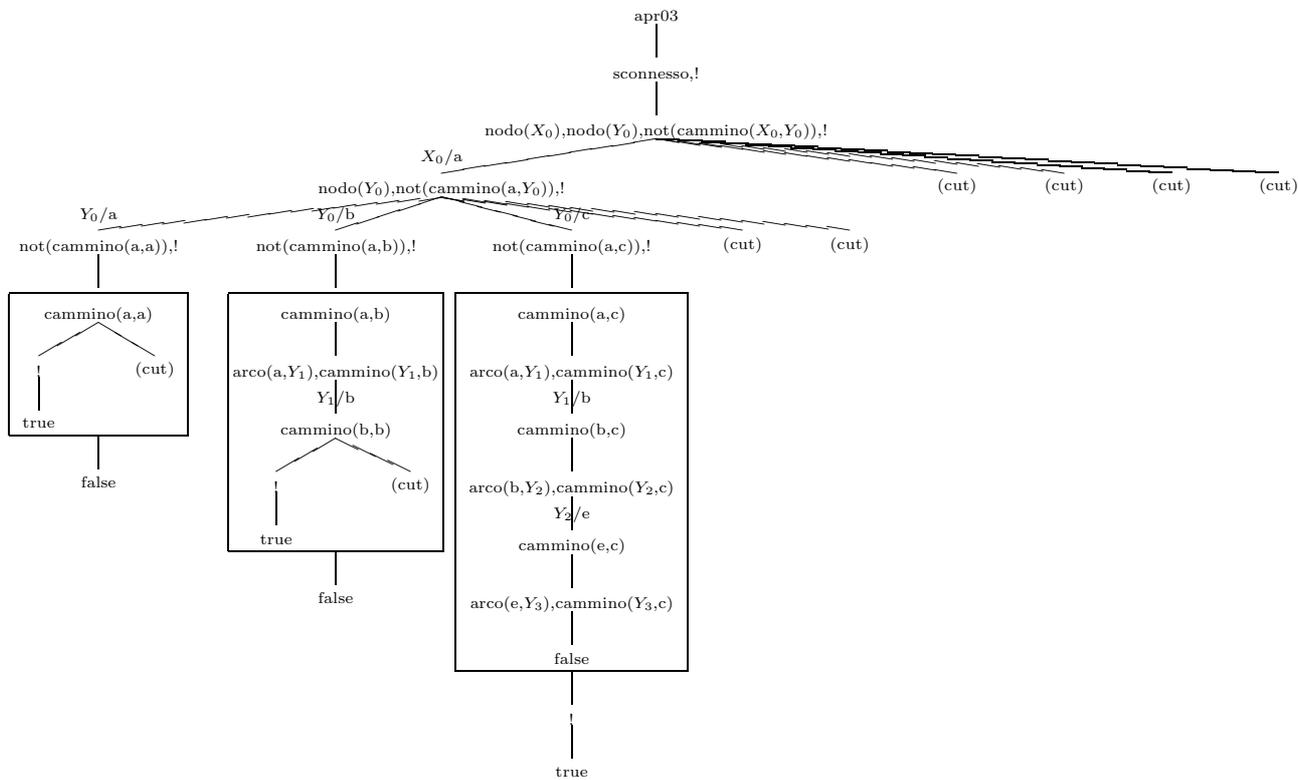
(10) not mamma\_insofferente(Alice) or punito(Carletto, Alice) or lasciatoacasa(Carletto) da (9) e (3)

(11) punito(Carletto, Alice) or lasciatoacasa(Carletto) da (10) e (2)

(12) lasciatoacasa(Carletto) da (11) e (6)

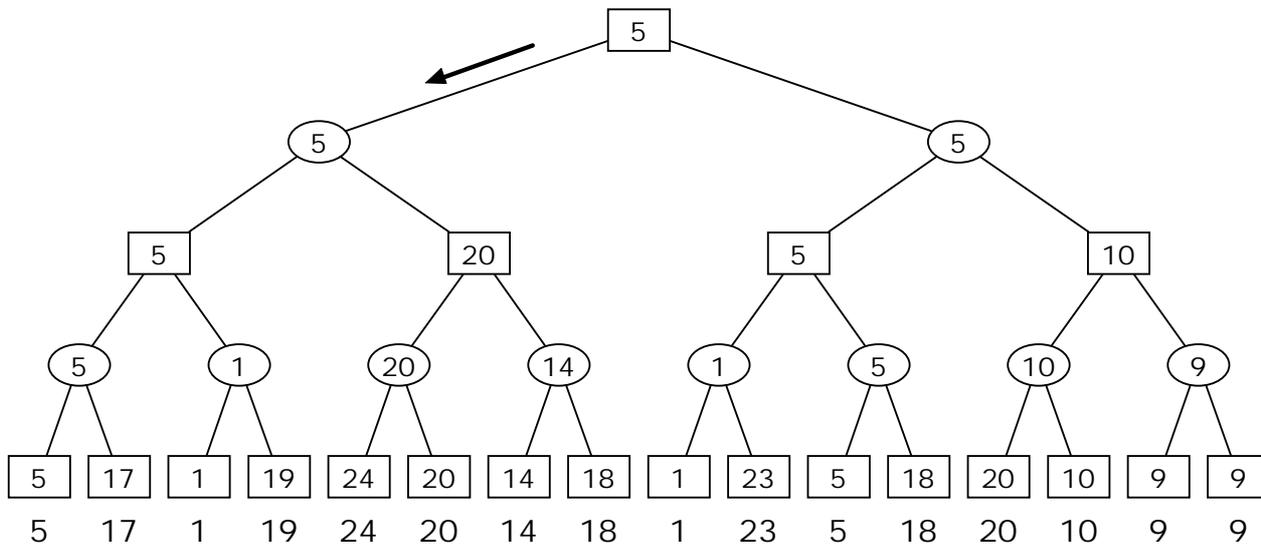
(13) formula vuota da (7) e (12)

## Esercizio 2

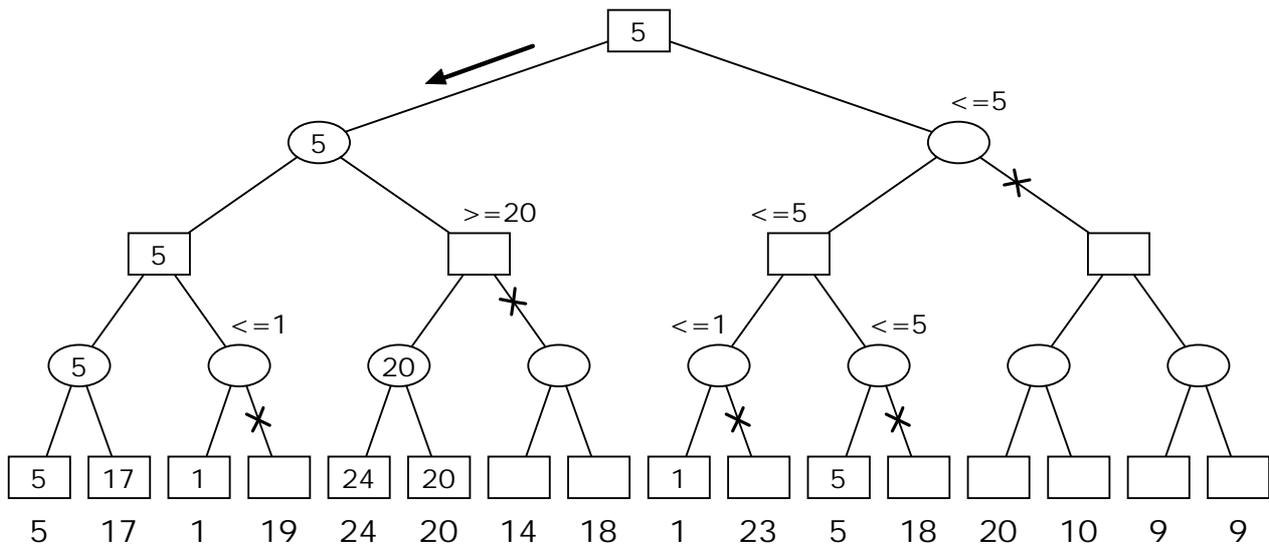


## Esercizio 3

Min-Max:



Alfa-beta:



#### Esercizio 4

Variabili

Ogni cella della scacchiera rappresenta una variabile da  $X_{11}$  a  $X_{44}$  (16 variabili in tutto).

Domini

$X_{i,j} :: \{R; B; G; Y\}$ , per  $i = 1..4$  e  $j = 1..4$ .

Vincoli

Riga: per ogni  $i j k :: [1; 2; 3; 4]$   $X_{i,j} \neq X_{i,k}$  per  $j \neq k$

Colonna: per ogni  $i j k :: [1; 2; 3; 4]$   $X_{j,i} \neq X_{k,i}$  per  $j \neq k$

Diagonale principale: per ogni  $i, j :: [1; 2; 3; 4]$   $X_{i,i} \neq X_{j,j}$  con  $i \neq j$

Diagonale secondaria: per ogni  $i, j :: [1; 2; 3; 4]$   $X_{i,4-i+1} \neq X_{j,4-j+1}$  con  $i \neq j$

Risoluzione

Si rappresentano i domini all'interno delle celle della scacchiera. Con il forward checking si arriva a una soluzione senza mai fallire. La scelta della variabile da istanziare è fatta usando l'euristica first-fail.

RB	RB	RB	RB
YG	YG	YG	YG
RB	RB	RB	RB
YG	YG	YG	YG
RB	RB	RB	RB
YG	YG	YG	YG
RB	RB	RB	RB
YG	YG	YG	YG

X11=R

R	B	B	B
YG	YG	YG	YG
B	B	RB	RB
YG	YG	YG	YG
B	RB	B	RB
YG	YG	YG	YG
B	RB	RB	B
YG	YG	YG	YG

X12=B

R	B	YG	YG
B	YG	RB	RB
YG	YG	YG	YG
B	YG	B	RB
YG	YG	YG	YG
B	YG	RB	B
YG	YG	YG	YG
B	YG	RB	B
YG	YG	YG	YG

X13=Y

R	B	Y	G
B	YG	RB	RB
YG	YG	G	YG
B	YG	B	RB
YG	YG	YG	YG
B	YG	RB	B
YG	YG	YG	YG
B	YG	RB	B
YG	YG	YG	YG

X14=G

R	B	Y	G
B	YG	RB	RB
YG	YG	B	RB
B	YG	B	RB
YG	YG	YG	YG
B	YG	RB	B
YG	YG	YG	YG
B	YG	RB	B
YG	YG	YG	YG

X22=Y

R	B	Y	G
B	YG	RB	RB
YG	YG	B	RB
B	YG	B	B
YG	YG	YG	YG
B	YG	RB	B
YG	YG	YG	YG
B	YG	RB	B
YG	YG	YG	YG

X32=R

R	B	Y	G
B	YG	B	RB
YG	YG	R	B
B	YG	B	B
YG	YG	YG	YG
B	YG	RB	B
YG	YG	YG	YG
B	YG	RB	B
YG	YG	YG	YG

X23=B

R	B	Y	G
B	YG	B	R
YG	YG	R	B
B	YG	G	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG

X24=R

R	B	Y	G
B	YG	B	R
YG	YG	R	B
B	YG	G	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG

X21=G

R	B	Y	G
B	YG	B	R
YG	YG	R	B
B	YG	G	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG

X33=G

R	B	Y	G
B	YG	B	R
YG	YG	R	B
B	YG	G	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG

X41=Y

R	B	Y	G
B	YG	B	R
YG	YG	R	B
B	YG	G	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG

X31=B

R	B	Y	G
B	YG	B	R
YG	YG	R	B
B	YG	G	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG
B	YG	R	B
YG	YG	YG	YG

Gli stati relativi alle istanziazioni X34=Y, X42=G, X43=R, X44=B sono identici a quest'ultimo.

La soluzione trovata è consistente.

Gli stati relativi alle istanziazioni X<sub>3,4</sub>=Y, X<sub>4,2</sub>=G, X<sub>4,3</sub>=R, X<sub>4,4</sub>=B sono identici a quest'ultimo. La soluzione trovata è consistente.

## Esercizio 5

```
% Programma per il calcolo della somma e della media di tutti
% gli elementi di una lista di numeri data
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% somma_lista(L,Somma): vero se Somma e' la somma di tutti gli
%%% elementi della lista L

somma_lista([],0).
somma_lista([X|R],SommaTot) :-
    somma_lista(R,SommaR),
    SommaTot is SommaR + X.

%%% media_lista(L,Media): vero se Media e' la media aritmetica
%%% degli elementi della lista L

media_lista([],_) :- !, fail.

media_lista(L,Media) :-
    lungh_lista(L,N),
    somma_lista(L,S),
    Media is S / N.

%%% lungh_lista(L,N): vero se N e' il numero di elementi
%%% della lista L

lungh_lista([],0).
lungh_lista([_X|R],LunghTot) :-
    lungh_lista(R,LunghR),
    LunghTot is LunghR + 1.
```