

## FONDAMENTI DI INTELLIGENZA ARTIFICIALE

2 Aprile 2009 – Tempo a disposizione 2h – Risultato 32/32 punti

### Esercizio 1 (punti 6)

Sam, Clyde e Oscar sono elefanti (ma questa informazione può essere trascurata). Su di essi conosciamo i seguenti fatti:

*Sam è rosa.*

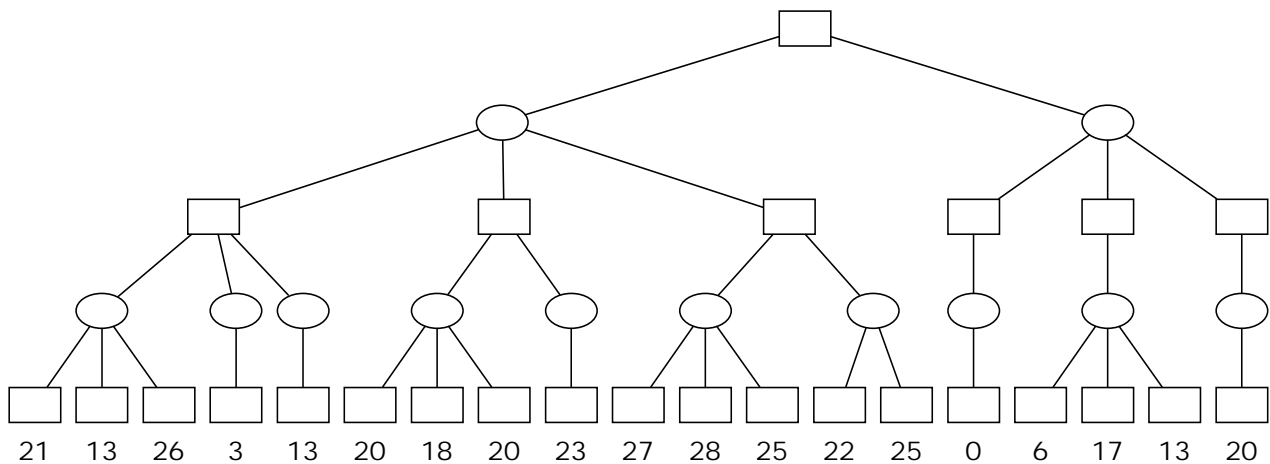
*Clyde è grigio ed è amico di Oscar.*

*Oscar è grigio o rosa (ma non di entrambi i colori) ed è amico di Sam.*

- Rappresentare le frasi precedenti in logica dei predicati del I ordine (per il predicato amico si scriva anche la clausola che definisce la relazione come simmetrica). Dimostrare con il metodo di risoluzione che *un elefante grigio è amico di un elefante rosa*.
- È possibile risolvere questo problema in programmazione logica? Mostrare la soluzione o spiegare perché non si può.

### Esercizio 2 (punti 5)

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 3 (punti 7)

Si consideri il seguente programma Prolog, che, dati due operandi ed un risultato, cerca un'operazione che dia il risultato richiesto:

```
operazione(A+B,A,B).
```

```
operazione(A-B,A,B).
```

```
operazione(A*B,A,B).
```

```
produci(R,Op,La,Lb):-operazione(Op,A,B),  
                        member(A,La), member(B,Lb),  
                        R is Op, !.
```

```
member(X,[X|_]).
```

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

Si mostri l'albero SLDNF relativo all'invocazione del goal `?-produci(1,Op,[2],[1,2]).`

#### Esercizio 4 (punti 7)

Si consideri una semplificazione del gioco del Sudoku dove i numeri ammissibili appartengono all'intervallo  $\{1,2,3,4\}$ . Si ricorda che un Sudoku è costituito da una griglia, divisa in più sottogriglie, in cui obiettivo è determinare per ogni casella della griglia un numero tale che:

- le caselle appartenenti alla stessa riga,
- le caselle appartenenti alla stessa colonna
- e le caselle appartenenti alla stessa sotto-griglia

presentino numeri diversi.

Si vuole formalizzare tale problema come un problema di soddisfacimento di vincoli dove le variabili da assegnare (tutte con dominio  $\{1,2,3,4\}$ ) sono riportate in figura:

$X_{11}$	$X_{12}$	$X_{13}$	$X_{14}$
$X_{21}$	$X_{22}$	$X_{23}$	$X_{24}$
$X_{31}$	$X_{32}$	$X_{33}$	$X_{34}$
$X_{41}$	$X_{42}$	$X_{43}$	$X_{44}$

Si esprimano (se possibile, impiegando una notazione compatta con indici  $X_{ij}$ ) i vincoli tra le variabili. Successivamente, si risolva il problema con **euristica MRV** e **forward checking** partendo dal seguente assegnamento parziale:

	<b>2</b>		<b>4</b>
<b>1</b>	<b>4</b>		
<b>4</b>	<b>1</b>		<b>3</b>
<b>2</b>	<b>3</b>	<b>4</b>	<b>1</b>

Si riportino i domini prodotti ad ogni passo dal forward checking e gli assegnamenti effettuati. A parità di altro, per l'assegnamento dei valori si scelgano le variabili secondo l'ordine lessicografico (prima secondo la riga e poi secondo la colonna).

#### Esercizio 5 (punti 4)

Si scriva un predicato Prolog `prodlista(L,P)` che è vero se  $P$  è il prodotto degli elementi interi della lista semplice  $L$ .

Esempio:

```
?-prodlista([1,2,3,4],P).
```

```
Yes P=24
```

```
?-prodlista([3,4],10).
```

```
no
```

#### Esercizio 6 (punti 3)

Definire la proprietà di monotonia della logica classica e citare almeno un problema di rappresentazione della conoscenza che richiede un tipo di ragionamento non monotono.

## SOLUZIONE

### Esercizio 1

a)

*Sam è rosa.*

*Clyde è grigio ed è amico di Oscar.*

*Oscar è grigio o rosa (ma non di entrambi i colori) ed è amico di Sam.*

- $rosa(sam)$
- $grigio(clyde)$
- $amico(clyde,oscar)$
- $rosa(oscar) \text{ XOR } grigio(oscar)$
- $amico(oscar,sam)$
- $\forall XY(amico(X,Y) \Rightarrow amico(Y,X))$

Goal: *un elefante grigio è amico di un elefante rosa.*

Goal negato:

- $\neg(\exists XY(amico(X,Y) \wedge grigio(X) \wedge rosa(Y)))$

Trasformazione in clausole (si noti che la quarta formula non è rappresentabile in clausole di Horn, quindi la risposta al punto b) dell'esercizio è NO):

- $rosa(sam)$
- $grigio(clyde)$
- $amico(clyde,oscar)$
- $rosa(oscar) \vee grigio(oscar)$
- $\neg rosa(oscar) \vee \neg grigio(oscar)$
- $amico(oscar,sam)$
- $\neg amico(X,Y) \vee amico(Y,X)$
- $\neg amico(X,Y) \vee \neg grigio(X) \vee \neg rosa(Y)$

Risoluzione:

1.  $\neg amico(A,B) \vee \neg grigio(A) \vee \neg rosa(B)$
2.  $rosa(sam)$
3.  $grigio(clyde)$
4.  $amico(clyde,oscar)$
5.  $rosa(oscar) \vee grigio(oscar)$
6.  $amico(oscar,sam)$
7. (da 5, 1, 4, 3)  $grigio(oscar)$
8. (da 7, 1, 6, 2)  $\square$



## Esercizio 4

E' possibile esprimere i vincoli tra le variabili nel seguente modo:

- (vincoli sulle righe)  $\forall i, k \neq j$  deve valere:  $X_{ij} \neq X_{ik}$
- (vincoli sulle colonne)  $\forall i, k \neq j$  deve valere:  $X_{ji} \neq X_{ki}$
- (vincoli sulle sotto-griglie) deve valere:  $X_{11} \neq X_{22}, X_{21} \neq X_{12}, X_{31} \neq X_{42}, X_{41} \neq X_{32},$   
 $X_{13} \neq X_{24}, X_{23} \neq X_{14}, X_{33} \neq X_{44}, X_{43} \neq X_{34}$

La risoluzione del CSP mediante MRV e forward checking produce, passo dopo passo, i seguenti assegnamenti e domini:

{3}	2	{1,3}	4
1	4	{2,3}	{2}
4	1	{2}	3
2	3	4	1

3	2	{1}	4
1	4	{2,3}	{2}
4	1	{2}	3
2	3	4	1

3	2	1	4
1	4	{2,3}	{2}
4	1	{2}	3
2	3	4	1

3	2	1	4
1	4	{3}	2
4	1	{2}	3
2	3	4	1

3	2	1	4
1	4	3	2
4	1	{2}	3
2	3	4	1

3	2	1	4
1	4	3	2
4	1	2	3
2	3	4	1

## Esercizio 5

```
prodlista([I|Is],Prod):-timeslist(Is,I,Prod).
```

```
timeslist([],Prod,Prod).
```

```
timeslist([I|Is],Temp,Prod):-Temp1 is Temp*I,
```

```
timeslist(Is,Temp1,Prod).
```

Oppure (ricorsione non-tail):

```
prodlista([X],X):-!. 
```

```
prodlista([H|T],P):-
```

```
    prodlista(T,P2),
```

```
    P is H * P2.
```