FONDAMENTI DI INTELLIGENZA ARTIFICIALE – PRIMA PARTE

9 Settembre 2010 – Tempo a disposizione 2h – Risultato 32/32 punti

Esercizio 1 (punti 7)

Si formalizzino in logica dei predicati del I ordine le seguenti frasi:

- *Tom e Mike sono membri del Club Alpino.*
- *Tutti i membri del Club Alpino sono sciatori o alpinisti o entrambe le cose.*
- *Gli alpinisti non amano la pioggia e gli sciatori non amano la neve.*
- *Mike non ama niente di ciò che ama Tom, e Tom ama tutto ciò che Mike non ama.*
- *Tom ama la pioggia.*

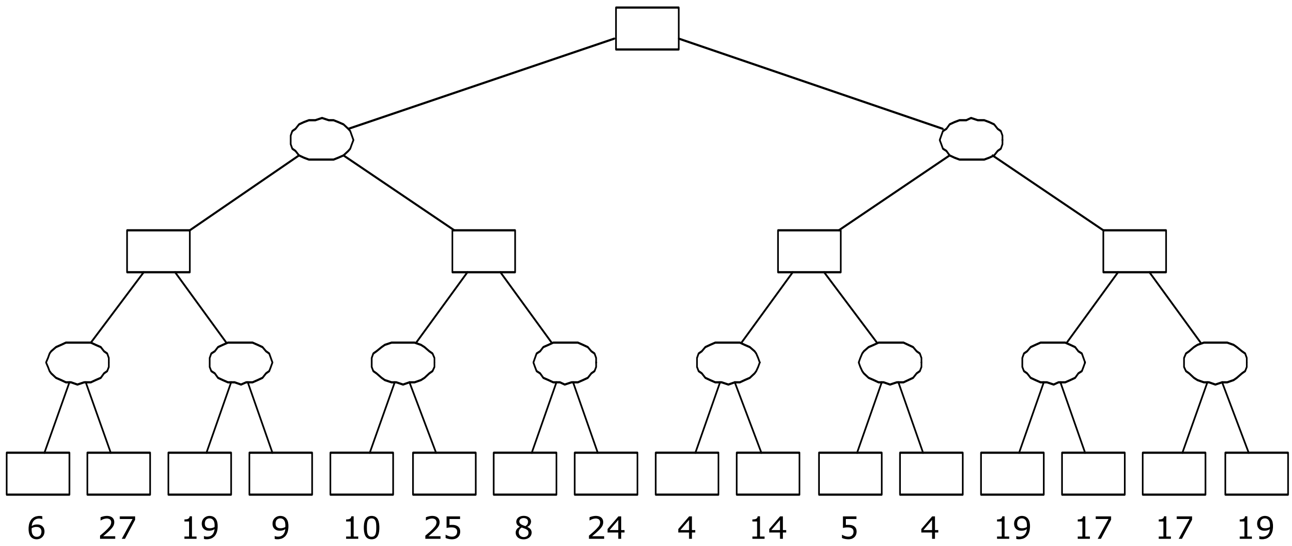
Dimostrare poi applicando il principio di risoluzione che:

- *Mike è un alpinista.*

Dire se è possibile ottenere un programma logico dalla KB così ottenuta.

Esercizio 2 (punti 5)

Si consideri il seguente albero di gioco in cui la valutazione dei nodi terminali è dal punto di vista del primo giocatore. Si assuma che il primo giocatore sia Max.



Si mostri come gli algoritmi min-max e alfa-beta risolvono il problema

Esercizio 3 (punti 6)

Dato il seguente programma Prolog:

```
split([],[],[]).
   split([H],[H],[]):- !.
   split([H|T],[H],T).
   split([H|T],[H|L1],L2) :- split(T,L1,L2).

membchk(X,[X|T]):- !.
   membchk(X,[H|T]):- membchk(X,T).

set10(L):- split(L,M,N), not(membchk(M,N)).
```

Si mostri l'albero di derivazione SLDNF relativo al goal `set10([1,[1],X])`.

Esercizio 4 (punti 6)

Si consideri il problema di scegliere la prossima mossa di esplorazione nel gioco del Minesweeper. Le caselle con il '?" posso contenere una bomba o essere vuote. Le caselle numerate sono state esplorate e i numeri indicano il numero di bombe nelle caselle adiacenti (in orizzontale, verticale e diagonale – le caselle interne hanno 8 caselle adiacenti).

?	?			
2	?			
1	?	?		
1	2	?	?	?
0	1	1	1	0

a. Si formuli il problema come un problema di soddisfacimento di vincoli dove le caselle con il "?" sono rappresentate come variabili CSP, numerate per riga come X1,……X8 a partire dall'alto e da sinistra a destra. Sappiamo inoltre che X4 e X6 contengono una bomba.

b. A partire dalla situazione iniziale, in cui i domini sono opportunamente ristretti tenendo conto dei soli vincoli unari, si dica quale variabile verrebbe scelta con l'euristica MRV (Minimum Remaining Values) e, a parità di valutazione, in base all'euristica del grado (scelta della variabile coinvolta in un numero maggiore di vincoli).

c. Scelta la variabile, si mostri come verrebbero ristretti i domini con un passo di verifica in avanti (Forward Checking).

Esercizio 5 (punti 5)

Si scriva un predicato programma Prolog `no_doubles(Xs, Ys)` che è vero se `Ys` è la lista degli elementi di `Xs` senza ripetizioni. Gli elementi di `Ys` sono in ordine inverso rispetto a quello in cui comparivano la prima volta in `Xs`. Esempio:

```
?-no_doubles([1,3,2,2,4,1,5,6,6], Ys).
Yes  Ys=[6,5,4,2,3,1]
```

Esercizio 6 (punti 3)

Un metodo di inferenza è *completo* se:

a. può derivare un qualunque formula che è conseguenza logica
b. deriva solo formule che sono conseguenza logica
c. è efficiente sia nel tempo che nello spazio
d. è di complessità polinomiale

Indicare la o le risposte corrette. Indicare inoltre se il principio di risoluzione per la logica a clausole è corretto e completo, motivando adeguatamente la risposta. Si discutano inoltre le caratteristiche e i limiti di Prolog relativamente a completezza e correttezza.

SOLUZIONE

Esercizio 1

Soluzione, già in forma a clausole (ciascuna clausola è un insieme di disgiunti):

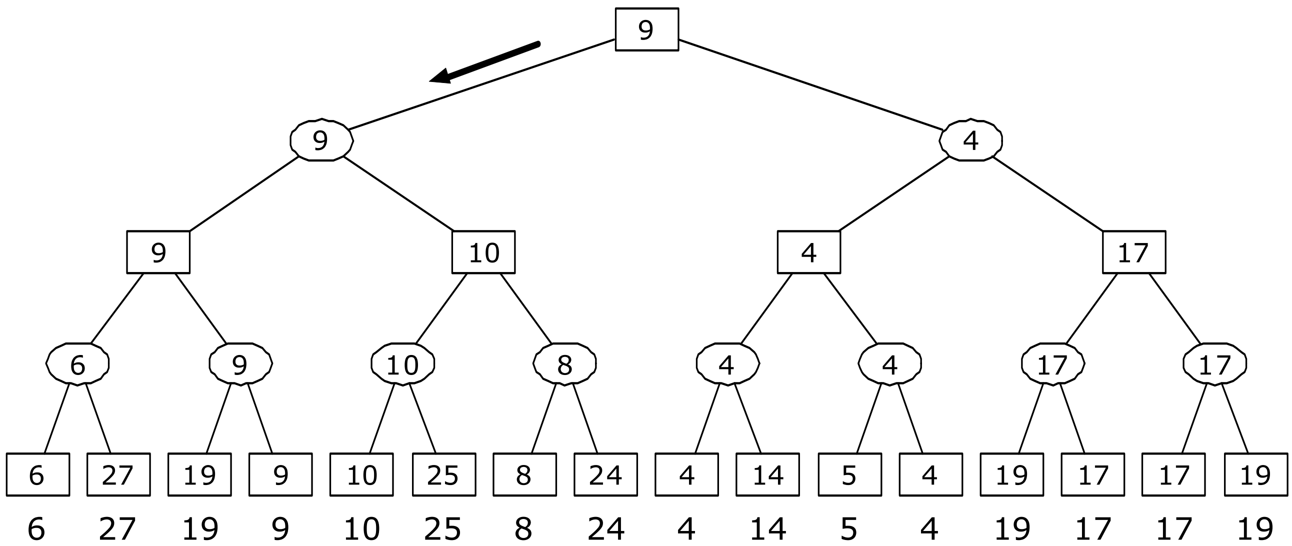
1. {MC(Tom)}
2. {MC(Mike)}
3. {¬MC(x), Sc(x), Al(x)}
4. {¬Al(x), ¬Ama(x, Pioggia)}
5. {¬Sc(x), ¬Ama(x, Neve)}
6. {¬Ama(Tom, x), ¬Ama(Mike, x)}
7. {Ama(Mike, x), Ama(Tom, x)}
8. {Ama(Tom, Pioggia)}
9. {¬Al(Mike)}

Risoluzione:

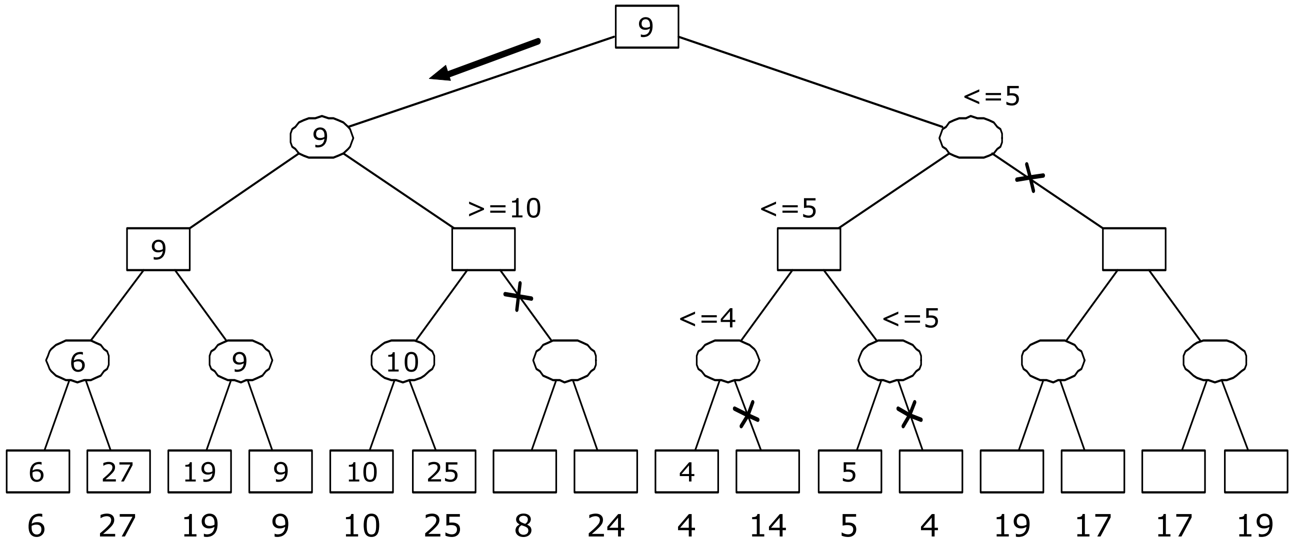
{¬MC(Mike), Sc(Mike)}
{Sc(Mike)}
{¬Ama(Mike, Neve)}
{Ama(Tom, Neve)}
{¬Sc(Tom)}
{¬MC(Tom), Al(Tom)}
{Al(Tom)}
{¬Ama(Tom, Pioggia)}
{}

Esercizio 2

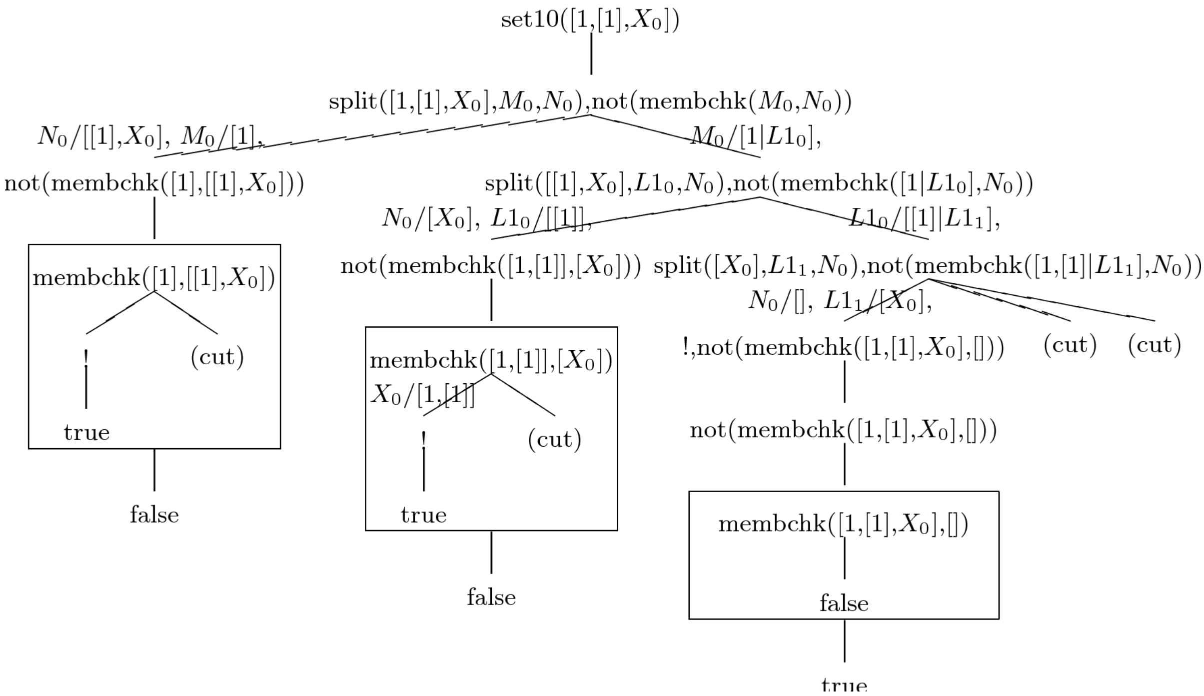
min-max:



Alfa-beta:



Esercizio 3



Esercizio 4

a. Una formulazione naturale è la seguente:

Variabili: X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8
Considerando variabili in corrispondenza delle caselle con ?
Domini: $Dom(X_i) = \{0, 1\}$, dove il valore 1 indica presenza della bomba.

X_1	X_2			
2	X_3			
1	X_4	X_5		
1	**2**	X_6	X_7	X_8
0	**1**	**1**	**1**	**0**

Vincoli:
C'è un vincolo per ogni numero scoperto (a parte lo 0 nell'angolo in basso a sinistra che non coinvolge variabili).

1. $X_1 + X_2 + X_3 + X_4 = 2$
2. $X_3 + X_4 = 1$
3. $X_4 = 1$
4. $X_4 + X_5 + X_6 = 2$

5. $X_6 = 1$
6. $X_6 + X_7 = 1$
7. $X_6 + X_7 + X_8 = 1$
8. $X_7 + X_8 = 0$

Passo iniziale che tiene conto dei soli vincoli unari (3 e 4) per ridurre i domini:

$X_1 = \{0, 1\}$
$X_2 = \{0, 1\}$
$X_3 = \{0, 1\}$
$X_4 = \{1\}$
$X_5 = \{0, 1\}$
$X_6 = \{1\}$
$X_7 = \{0, 1\}$
$X_8 = \{0, 1\}$

L'euristica MRV ci dice di partire ad assegnare una tra X_4 e X_6 perché hanno i domini più piccoli. Dovendo scegliere tra queste in base all'euristica del grado, osserviamo che X_4 è coinvolta in 3 vincoli (1, 2, 4) con variabili non ancora assegnate, ed anche X_6 è coinvolta in 3 vincoli (4, 6 e 7). Non disponendo di ulteriori criteri, scelgo X_4 perchè ha indice più basso (ma la scelta di X_6 sarebbe ugualmente giusta). Le assegno valore 1 e procedo con un passo di verifica in avanti. Il risultato è il seguente:

$X_1 = \{0, 1\}$
$X_2 = \{0, 1\}$
$X_3 = \{0\}$
$\mathbf{X_4 = 1}$
$X_5 = \{0, 1\}$
$X_6 = \{1\}$
$X_7 = \{0, 1\}$
$X_8 = \{0, 1\}$

Si restringe solo il dominio di X_3.

Esercizio 5

```
/* no_doubles(Xs, Ys) is true if Ys is the list of the elements appearing  */
/*   in Xs without duplication.  The elements in Ys are in the reverse     */
/*   order of Xs with the first duplicate values being kept.               */
no_doubles(Xs, Ys):-no_doubles_1(Xs, [], Ys).

no_doubles_1([], Ys, Ys).
no_doubles_1([X|Xs], As, Ys):-
  member(X, As),
  no_doubles_1(Xs, As, Ys).
no_doubles_1([X|Xs], As, Ys):-
  nonmember(X, As),
  no_doubles_1(Xs, [X|As], Ys).

/* member(X,Xs) is true if X is a member of the list Xs.                   */
member(X,[X|Xs]).
member(X,[Y|Ys]):-member(X,Ys).

/* nonmember(X,Xs) is true if X is not a member of the list Xs.            */
nonmember(X,[Y|Ys]):-X=\=Y, nonmember(X,Ys).
nonmember(X,[]).
```

OPPURE

```
no_doubles([],[]).
no_doubles([H|T],L):-
member(H,T),!,
no_doubles(T,L).
no_doubles([H|T],L):-
no_doubles(T,T2),
append(T2,[H],L).
```