

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

8 Gennaio 2009 (Tempo a disposizione 2h , su 32 punti)

Esercizio 1 (punti 6)

Siano *finanz*, *univ*, *ebrown*, *bill* simboli di costante, e siano:

prof(X), *stud(X)*, *rabbia(X)*, *taglia(X,Y)*, *boccia(X,Y)*

simboli di predicato unari e binari.

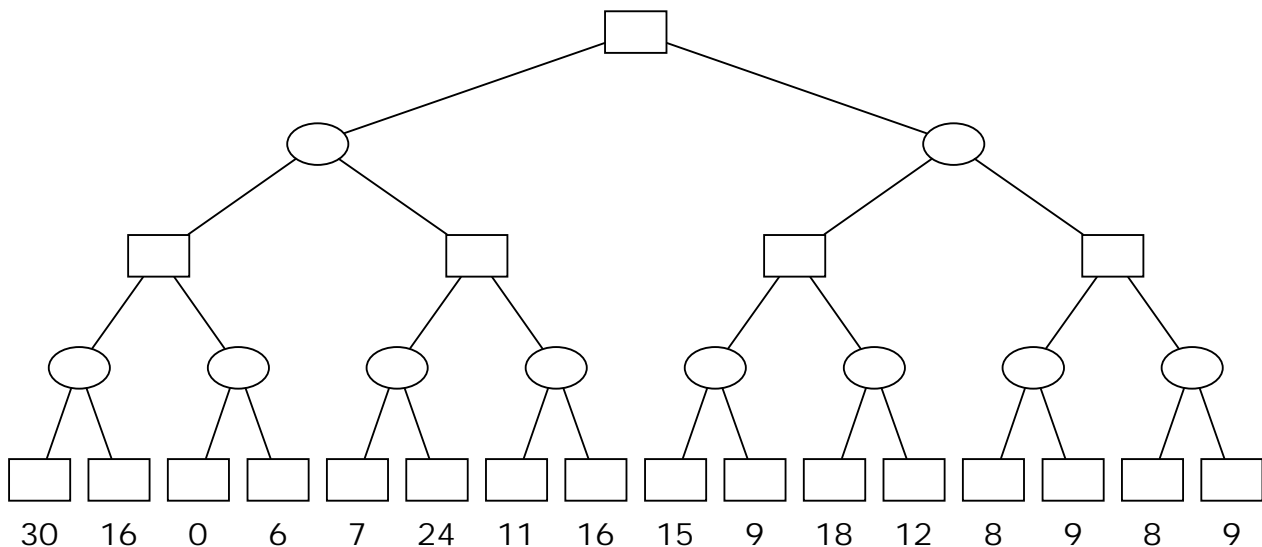
Con questi simboli dati, si esprimano le seguenti frasi in logica dei predicati del I ordine:

1. Se la finanziaria taglia i finanziamenti all'università i professori si arrabbiano.
2. Se i professori si arrabbiano gli studenti sono bocciati.
3. La finanziaria taglia i finanziamenti all'università.
4. E. Brown è una professoressa e Bill è uno studente.

Le si trasformi poi in clausole e si dimostri attraverso l'applicazione del principio di risoluzione che la Professoressa E. Brown boccia lo studente Bill.

Esercizio 2 (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 3 (punti 6)

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,!.`

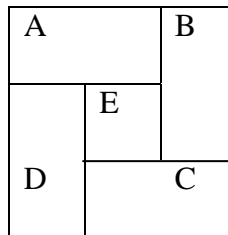
`member(X,[X|_]) .`

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

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

Esercizio 4 (punti 6)

Si abbia il seguente problema di colorazione di una mappa, in cui sono disegnate cinque regioni da colorare avendo a disposizione tre colori (individuati dai primi tre numeri naturali, 1, 2 e 3) in modo che due regioni adiacenti non siano colorate con lo stesso colore:



Si modelli il problema come CSP e lo si risolva utilizzando nella fase di ricerca la tecnica del partial look-ahead. Nella ricerca, si consideri come euristica nella scelta delle variabili il first-fail (secondo l'ordinamento alfabetico, nel caso di cardinalità dei domini identica) e i valori da assegnare secondo l'ordine crescente sugli interi.

Esercizio 5 (punti 5)

Si scriva un predicato Prolog `double(N, Xs, Ys)` che, dati `N, Xs, Ys` in ingresso, ha successo se `N` compare nella lista `Xs` e compare due volte nella lista (ordinata) `Ys`.

Esempi:

```
?-double(3, [1, 2], [1, 2, 3, 3]).
```

```
no
```

```
?-double(3, [1, 2, 3], [1, 2, 3, 3]).
```

```
yes
```

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

```
no
```

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

```
yes
```

Esercizio 6 (punti 3)

Dare la definizione di euristica ammissibile e consistente e quali vantaggi si hanno dal punto di vista algoritmico con un'euristica consistente.

SOLUZIONE

Esercizio 1

1. $\text{taglia}(\text{finanz}, \text{univ}) \rightarrow (\forall X (\text{prof}(X) \rightarrow \text{rabbia}(X)))$
2. $\forall X \forall Y ((\text{prof}(X) \text{ and } \text{rabbia}(X) \text{ and } \text{stud}(Y)) \rightarrow \text{boccia}(X, Y))$
3. $\text{taglia}(\text{finanz}, \text{univ})$
4. $\text{prof}(\text{ebrown}) \text{ and } \text{stud}(\text{bill})$
- Q. $\text{boccia}(\text{ebrown}, \text{bill})$

Forma a clausole:

1. $\neg \text{taglia}(\text{finanz}, \text{univ}) \text{ or } \neg \text{prof}(X) \text{ or } \text{rabbia}(X)$
2. $\neg \text{prof}(X) \text{ or } \neg \text{rabbia}(X) \text{ or } \neg \text{stud}(Y) \text{ or } \text{boccia}(X, Y)$
3. $\text{taglia}(\text{finanz}, \text{univ})$
- 4'. $\text{prof}(\text{ebrown})$
- 4''. $\text{stud}(\text{bill})$
- G. $\neg \text{boccia}(\text{ebrown}, \text{bill})$

Risoluzione:

Da 1 e 3:

5. $\neg \text{prof}(X) \text{ or } \text{rabbia}(X)$

Da 5 e 4':

6. $\text{rabbia}(\text{ebrown})$

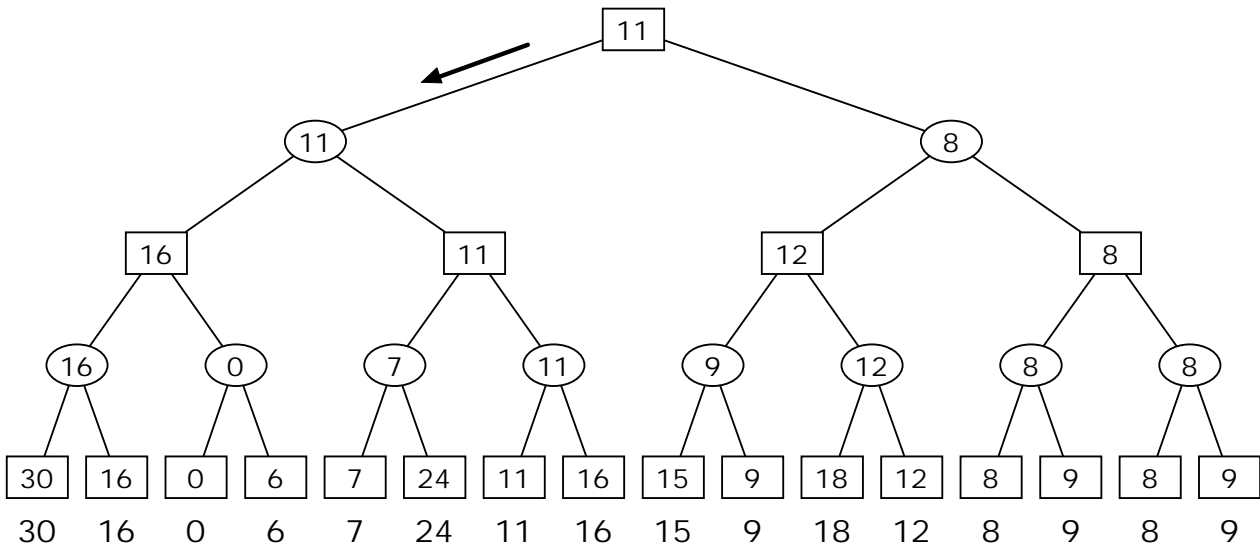
Da 2 e 6, 4', 4'', in più passi:

7. $\text{boccia}(\text{ebrown}, \text{bill})$

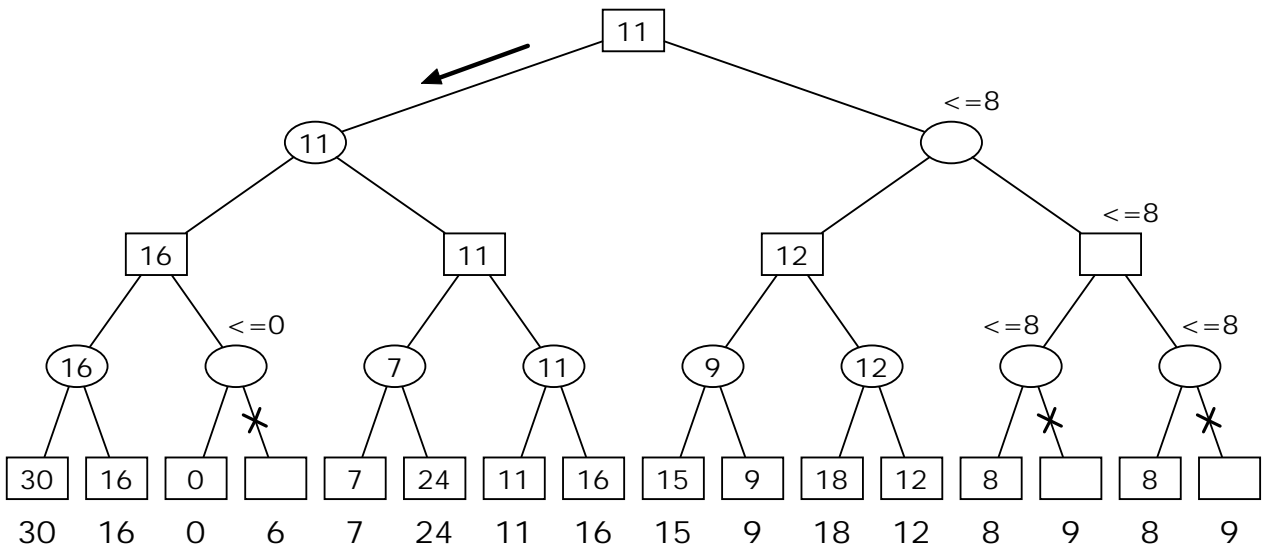
Da 7 e G, si deriva la clausola vuota.

Esercizio 2

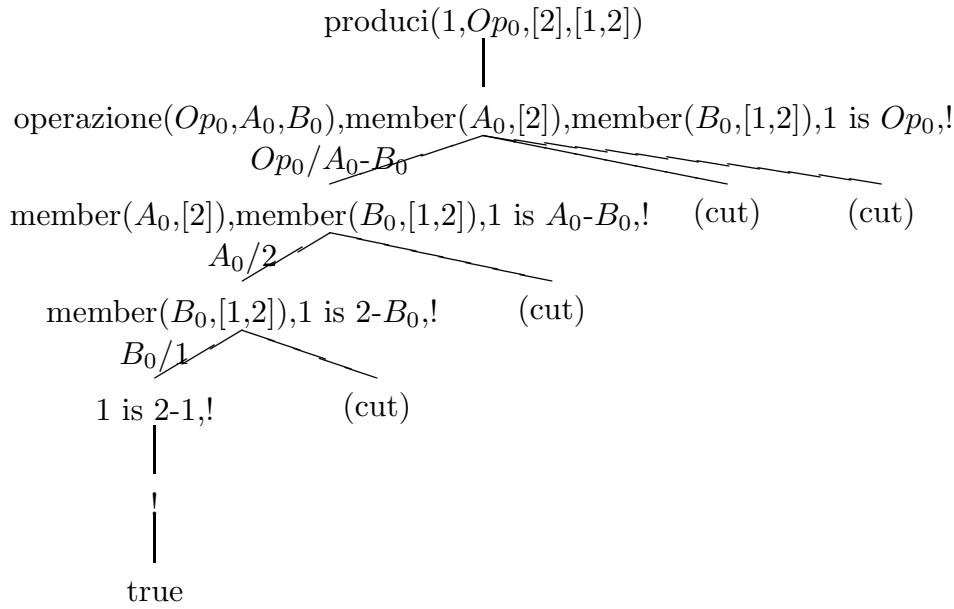
Min-Max:



Alfa-Beta:



Esercizio 3



Esercizio 4

Variabili:

A,B,C,D,E :: [1,2,3]

Vincoli:

A/=B

A/=D

A/=E

B/=C

B/=E

C/=D

D/=E

	A	B	C	D	E
Labeling + FC	1	[2,3]	[1,2,3]	[2,3]	[2,3]
Propagaz.		[2,3]	[1,2,3]	[2,3]	[2,3]
Labeling + FC		2	[1,3]	[2,3]	[3]
Propagaz.			[1]	[2]	[3]
Labeling + FC			1	[2]	[3]
Propagaz.				[2]	[3]
Labeling + FC				2	[3]
Propagaz.					[3]
Labeling + FC					3

Esercizio 5

```
/* double(N, Xs,Ys) is true if the element N is in the list Xs */  
/* and appears twice in the list Ys (ordered) */
```

```
double(X, [X|Xs], [X,X|Ys]) :- !.
```

```
double(X, [X|Xs], [_|Ys]) :- !, double(X, [X|Xs], Ys).
```

```
double(X, [_|Xs], Ys) :- !, double(X, Xs, Ys).
```