

Esercizio 4 (6 punti)

Si definisca un predicato *Prolog* `media_lista/2` che, data una lista `L` di interi in ingresso, ne restituisca il valore medio (la media). Si supponga di avere già definito il predicato `length/2` che, data una lista `L` di interi in ingresso, ne restituisce la lunghezza come secondo argomento.

Esempi: ?- `media_lista([1,2,3,6,3],N)`. yes `N=3`
 ?- `media_lista([2,2,3,3],N)`. yes `N=2.5`

Esercizio 5 (6 punti)

Data una matrice di 3x3 caselle, sia dato il problema di inserire i numeri interi (da 1 a 9) in modo che ogni riga, colonna o diagonale contenga numeri che hanno sempre somma uguale a 15. Si formalizzi questo problema come CSP, e si mostri come risolverlo partendo da questa configurazione iniziale:

2		6
4	3	

Si semplifichino inizialmente i vincoli in base ai valori già assegnati alle variabili nella configurazione iniziale. Nella ricerca della soluzione, nella scelta della prossima variabile da istanziare si scelga quella con indice di riga più alto, e a parità, con indice di colonna più basso. Dopo ogni istanziamento, si applichi il *Partial Look Ahead* (considerando le variabili libere ordinate secondo indice di riga, e poi colonna).

Esercizio 6 (3 punti)

Si illustri la strategia di ricerca *iterative deepening* e le sue proprietà in termini di completezza, ottimalità e complessità spaziale e temporale.

VOTO:

- *Esame da 6 CFU, il voto è determinato da questa I parte*
- *Esame da 9 CFU, è la media pesata della I parte (che vale 2/3) e della II (che vale 1/3) ovvero il voto finale è dato da: $2 \cdot (\text{votoParteI} + \text{votoParteII}) / 3$ e varia quindi da 0 ad un massimo di 32 (equivalente alla lode).*

FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 2° parte (3 CFU)

31 Gennaio 2013 – Tempo: 30 + 15min – Risultato: 16/16 punti

Esercizio 7 (5 punti)

Si descriva cos'è una ontologia presentando anche un piccolo esempio e si dica quali sono i vantaggi di una rappresentazione ontologica.

Esercizio 8 (8 punti)

Si scriva un meta-interprete:

```
solve(Goal, ListTrace)
```

che, dato in ingresso un Goal, restituisce in uscita, in caso di successo, la lista ListTrace di tutti i goal risolti per risolvere quello iniziale Goal.

Ad esempio, dato il programma:

```
p(X,Y):-q(X,Y), r(X), m(Y).  
q(1,2).  
r(1).  
m(2).
```

l'invocazione seguente ha successo con esito come indicato:

```
?- solve(p(X, Y), ListOut).  
yes X=1, Y=2, ListOut=[p(1,2), q(1,2), r(1), m(2)]
```

Esercizio 9 (3 punti) – ulteriori 15 min

Nota: lo deve svolgere solamente chi non ha partecipato alle lezioni/esercitazioni su Prolog e grammatiche.

Si formalizzi la grammatica che definisce cosa è un identificatore (un identificatore inizia con una lettera, seguita da cifre o lettere, senza altri simboli). Ad esempio, i nomi X, A123, Y sono identificatori, 1S non lo è.

A partire dalla grammatica definita, si scrivano poi le corrispondenti clausole in versione DCG che verificano la correttezza di un identificatore.

VOTO:

- *Esame da 6 CFU, il voto è determinato da questa I parte*
- *Esame da 9 CFU, è la media pesata della I parte (che vale 2/3) e della II (che vale 1/3) ovvero il voto finale è dato da: $2 \cdot (\text{votoParteI} + \text{votoParteII}) / 3$ e varia quindi da 0 ad un massimo di 32 (equivalente alla lode).*

FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 1° parte

31 Gennaio 2013 – Soluzioni

Esercizio 1

Fraasi in *logica dei predicati del I ordine* e loro traduzione in *clausole*:

1. *Se una e-mail contiene le parole cheap e buy, allora è spam.*

$$\forall X: \text{email}(X) \wedge \text{contiene}(X, \text{cheap}) \wedge \text{contiene}(X, \text{buy}) \Rightarrow \text{spam}(X)$$

a. $\neg \text{email}(X) \vee \neg \text{contiene}(X, \text{cheap}) \vee \neg \text{contiene}(X, \text{buy}) \vee \text{spam}(X)$

2. *Tutte le e-mail che contengono la parola bargain contengono anche la parola buy.*

$$\forall X: \text{email}(X) \wedge \text{contiene}(X, \text{bargain}) \Rightarrow \text{contiene}(X, \text{buy})$$

b. $\neg \text{email}(X) \vee \neg \text{contiene}(X, \text{bargain}) \vee \text{contiene}(X, \text{buy})$

3. *Una data e-mail contiene le parole cheap, offer e bargain.*

$$\text{email}(m) \wedge \text{contiene}(m, \text{cheap}) \wedge \text{contiene}(m, \text{offer}) \wedge \text{contiene}(m, \text{bargain})$$

c. $\text{email}(m)$

d. $\text{contiene}(m, \text{cheap})$

e. $\text{contiene}(m, \text{offer})$

f. $\text{contiene}(m, \text{bargain})$

4. *Una data e-mail contiene le parole cheap, offer e bargain.*

$$\text{spam}(m)$$

g. $\neg \text{spam}(m)$

Risoluzione:

h. b. \cup c. $\neg \text{contiene}(m, \text{bargain}) \vee \text{contiene}(m, \text{buy})$ {X/m}

i. h. \cup f. $\text{contiene}(m, \text{buy})$

j. a. \cup c. $\neg \text{contiene}(m, \text{cheap}) \vee \neg \text{contiene}(m, \text{buy}) \vee \text{spam}(m)$ {X/m}

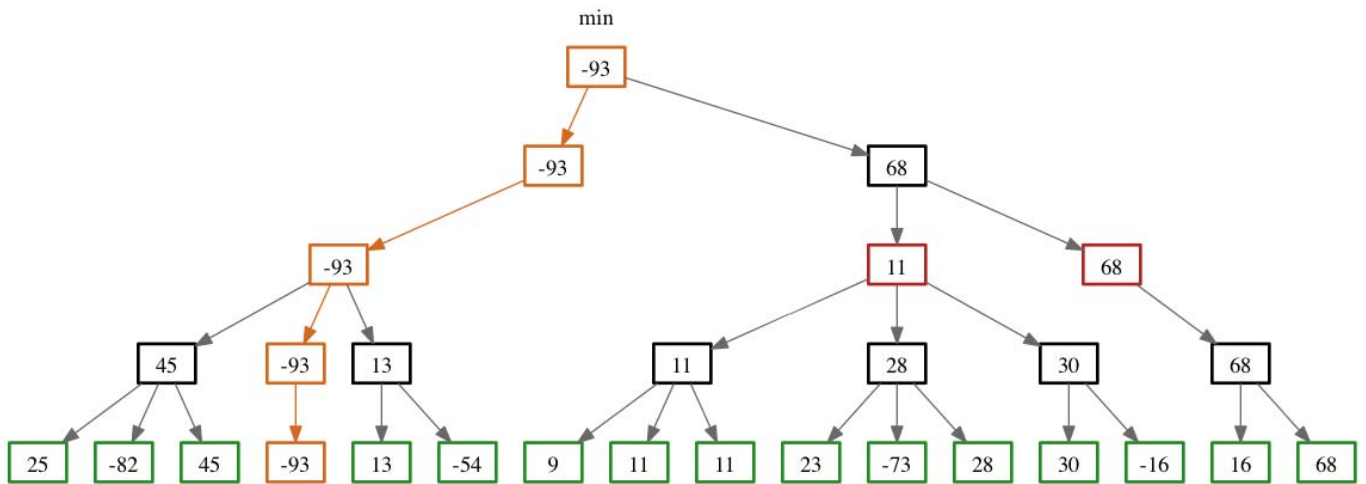
k. j. \cup d. $\neg \text{contiene}(m, \text{buy}) \vee \text{spam}(m)$

l. k. \cup i. $\text{spam}(m)$

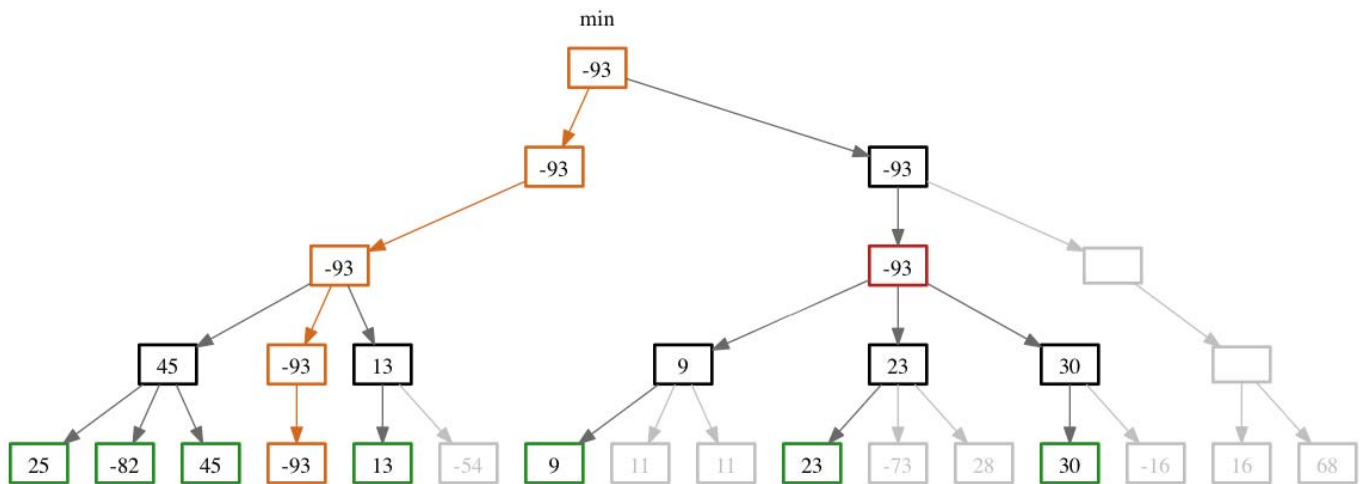
m. l. \cup g. \square

Esercizio 2

min-max:



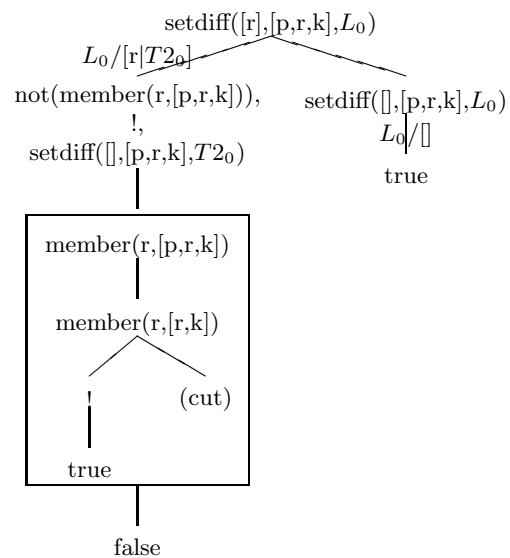
alfa-beta:



I nodi in grigio chiaro sono quelli che vengono tagliati nell'algoritmo alfa-beta.

Esercizio 3

La risposta calcolata è L_0/\square



Esercizio 4

```
somma_lista([], 0).
somma_lista([Head|Tail], Media) :- somma_lista(Tail, Head, 1, Media).
somma_lista([], Somma, N, Media) :- Media is Somma / N.
somma_lista([Head|Tail], Somma, N, Media) :-
    SommaTR is Somma + Head,
    NTR is N + 1,
    somma_lista(Tail, SommaTR, NTR, Media).
```

Oppure:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% 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(L,Media) :-
    lung_h_lista(L,N),
    somma_lista(L,S),
    Media is S / N.
```

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

```
lung_h_lista([],0).
lung_h_lista([_|R],LunghTot) :-
    lung_h_lista(R,LunghR),
```

Esercizio 5

Una possibile formulazione del problema prevede l'uso di una variabile $X_{i,j}$ per ognuna delle 9 caselle, dove i e $j \in \{1, 2, 3\}$ indicano rispettivamente l'indice di riga e di colonna della casella a partire dalla casella in alto a destra dello schema.

Il dominio di ognuna di queste variabili è $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

I vincoli del problema sono

$$\begin{aligned}
X_{1,k} + X_{2,k} + X_{3,k} &= 15, & k \in \{1, 2, 3\} \\
X_{k,1} + X_{k,2} + X_{k,3} &= 15, & k \in \{1, 2, 3\} \\
X_{1,1} + X_{2,2} + X_{3,3} &= 15 \\
X_{3,1} + X_{2,2} + X_{1,3} &= 15 \\
X_{1,1} \neq X_{1,2} \neq X_{1,3} \neq X_{2,1} \neq X_{2,2} \neq X_{3,2} \neq X_{3,1} \neq X_{3,2} \neq X_{3,3}
\end{aligned}$$

Scegliendo la variabile da instanziare con il criterio indicato e propagando i vincoli mediante *Partial Look Ahead* nell'ordine suggerito per le variabili, la soluzione si ottiene dall'assegnamento parziale proposto dal testo nel modo che segue (in arancione è indicato il dominio residuo delle variabili ancora da assegnare e in verde la prossima variabile da assegnare):

2	{7}	6
{9}	{5}	{1,8}
4	3	{8}

2	{7}	6
{9}	{5}	{1}
4	3	8

2	{7}	6
9	{5}	{1}
4	3	8

2	{7}	6
9	5	{1}
4	3	8

2	{7}	6
9	5	1
4	3	8

2	7	6
9	5	1
4	3	8

Esercizio 6

Vedi slides del corso.

FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 2° parte

31 Gennaio 2013 – Soluzioni

Esercizio 7

Vedi slides del corso.

Esercizio 8

```
solve(true, []).  
solve((A,B), L) :- solve(A,LA), solve(B,LB), append(LA, LB,L).  
solve(A,[A|LB]) :- clause(A,Body), solve(Body,LB).
```

Esercizio 9

Vedi slides del corso.