FONDAMENTI DI INTELLIGENZA ARTIFICIALE

Giugno 2018 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (6 punti)

Date le seguenti frasi in linguaggio naturale:

1.Mario sa risolvere gli stessi problemi di logica che sa risolvere Maria

2.…e viceversa (Maria sa risolvere gli stessi problemi di logica di Mario)

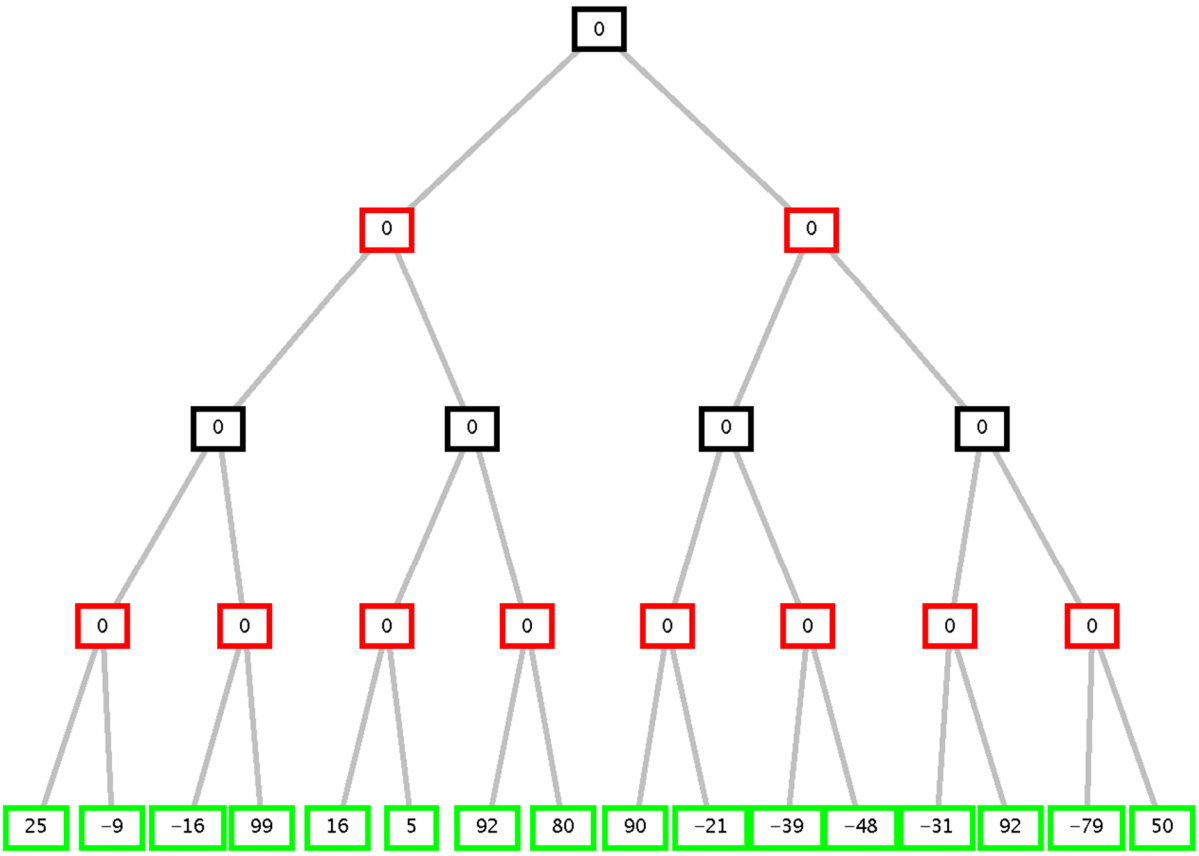
3.Chi sa risolvere un problema di logica difficile, prende 30 all'esame di Intelligenza Artificiale

4.Esiste un problema di logica difficile che Mario sa risolvere.

Dimostrare, tramite il principio di risoluzione, che Maria prenderà 30 all'esame di Intelligenza Artificiale. Si usino, a tal scopo, i seguenti predicati (da significato inteso): **problema(X)**, *X è un problema di logica*; **risolve(X,Y)**, *X sa risolvere il problema Y*; **difficile(X)**, *X è un problema difficile*; **prende30(X)**, *X prenderà 30 nell'esame di Fondamenti di Intelligenza Artificiale*.

Esercizio 2 (5 punti)

Si consideri il seguente albero di gioco in cui la valutazione dei nodi terminali è dal punto di vista del primo giocatore (*MAX*). Si mostri come l'algoritmo *min-max* e l'algoritmo *alfa-beta* risolvono il problema e la mossa selezionata dal giocatore.



Esercizio 3 (6 punti)

Dato il seguente programma Prolog:

```
faiCoppie([],[]).
faiCoppie([X,Y|R],[eq(X,Y)|T]):- (   X == Y),!, faiCoppie([Y|R],T).
faiCoppie([_|R],T):-faiCoppie(R,T).
```

dove il predicato **faiCoppie(Lin,Lout)** produce nella lista **Lout** le coppie **eq(X,Y)** di termini adiacenti **X,Y** di **Lin** che sono uguali, si disegni l'albero SLD relativo al goal:

```
?- faiCoppie([1,3,2,2],X).
X=[eq(2,2)]
```

Esercizio 4 (5 punti)

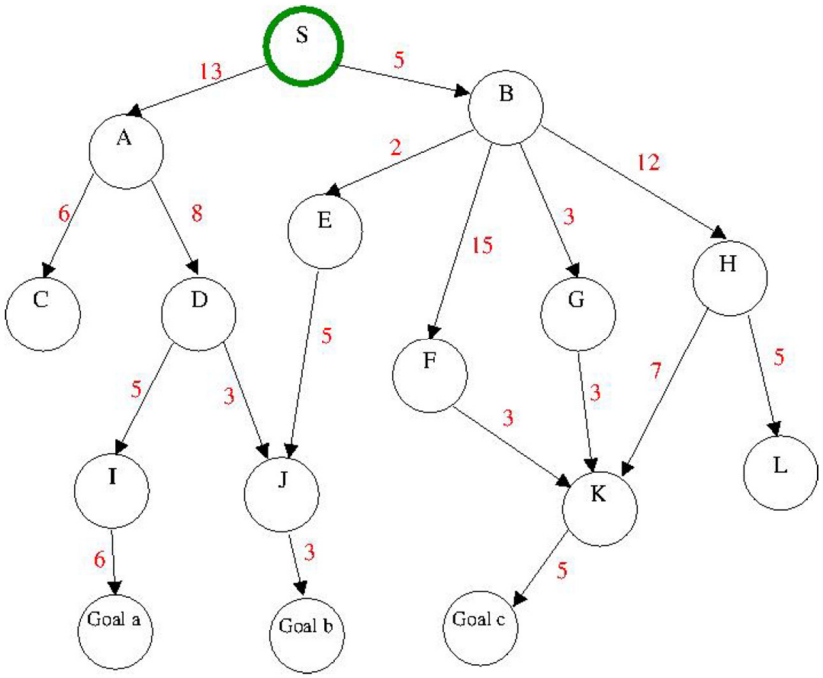
Data una lista `List`, si scriva un predicato `last_element(List,Rest,LastEl)` che restituisce in `LastEl` il suo ultimo elemento e in `Rest` il resto della lista `List`.

Ad esempio:

```
?- last_element([4, a, 7, 2], Rest, LastEl).
yes  Rest = [4, a, 7]    LastEl = 2.
```

Esercizio 5 (4 punti)

Si consideri il seguente grafo in cui gli archi sono annotati col costo e si indichi: (i) l'ordine dei nodi che sarebbero espansi nel caso di una ricerca a costo uniforme, e (ii) la strada che porta alla soluzione individuata.



Esercizio 6 (4 punti)

Si introduca il sistema STRIPS per la pianificazione, si spieghi come funziona e se ne evidenzino i limiti.

Esercizio 7 (2 punti)

Si descrivano, in maniera concisa, le proprietà di correttezza e di completezza di un sistema automatico di dimostrazione.

FONDAMENTI DI INTELLIGENZA ARTIFICIALE

Febbraio 2018 – Soluzioni

Esercizio 1

Conversione in clausole:

Mario sa risolvere gli stessi problemi di logica che sa risolvere Maria.

$\forall X$ problema(X) $\wedge$ risolve(maria, X) $\Rightarrow$ risolve(mario,X)

$\sim$problema(X) $\vee$ $\sim$risolve(maria,X) $\vee$ risolve(mario,X) C1

Maria sa risolvere gli stessi problemi di logica che sa risolvere Mario.

$\forall X$ problema(X) $\wedge$ risolve(mario,X) $\Rightarrow$ risolve(maria,X)

$\sim$problema(X) $\vee$ $\sim$risolve(mario,X) $\vee$ risolve(maria,X) C2

Chi sa risolvere un problema di logica difficile, prende 30 all'esame di Intelligenza Artificiale.

$\forall X \forall P$ problema(P) $\wedge$ risolve(X,P) $\wedge$ difficile(P) $\Rightarrow$ prende30(X)

$\sim$problema(P) $\vee$ $\sim$risolve(X,P) $\vee$ $\sim$difficile(P) $\vee$ prende30(X) C3

Esiste un problema di logica difficile che Mario sa risolvere.

$\exists P$ problema(P) $\wedge$ risolve(mario,P) $\wedge$ difficile(P).

k1 costante di Skolem.

problema(k1) C4.1

risolve(mario,k1) C4.2

difficile(k1) C4.3

Goal: *Maria prenderà 30 all'esame di Intelligenza Artificiale*

prende30(maria)

Negato: $\sim$prende30(maria) Gneg

Risoluzione:

C5: Gneg + C3: $\sim$problema(P) $\vee$ $\sim$risolve(maria,P) $\vee$ $\sim$difficile(P)

C6: C5 + C4.1: $\sim$risolve(maria,k1)$\vee$ $\sim$difficile(k1)

C7: C6 + C4.3: $\sim$risolve(maria,k1)

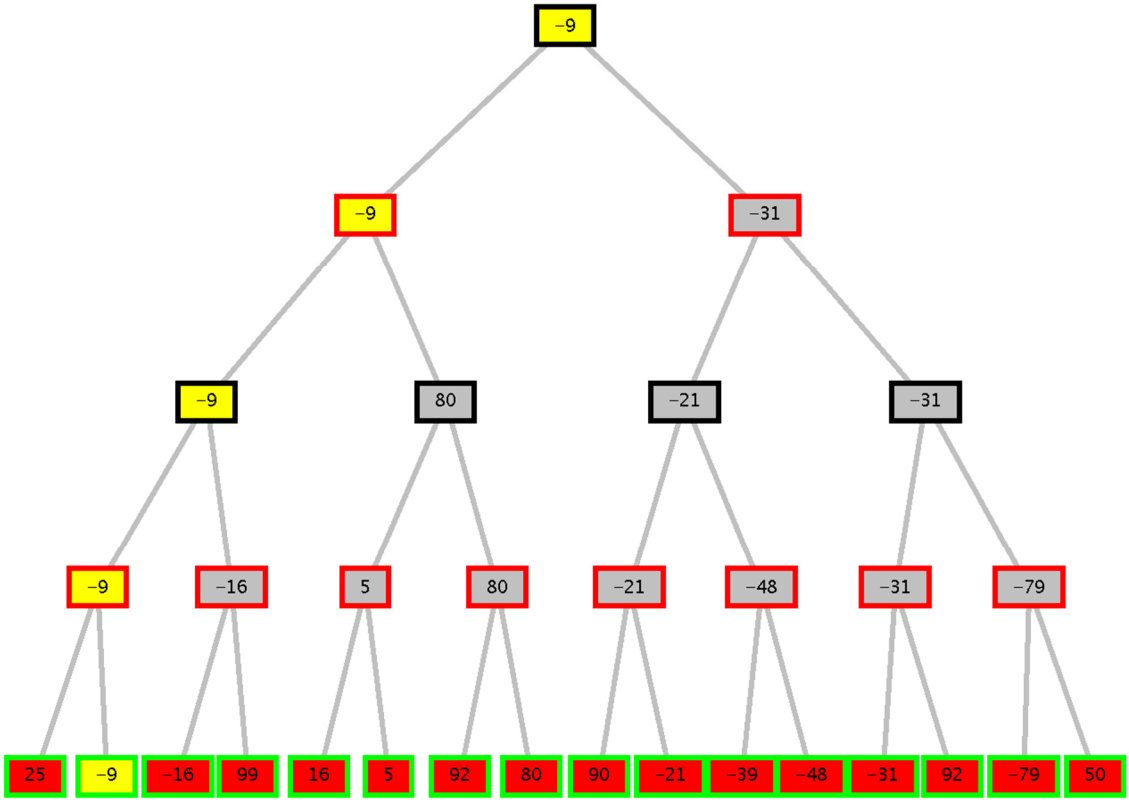
C8: C7 + C2: $\sim$risolve(mario,k1) $\vee$ $\sim$problema(k1)

C9: C8 + C4.2: $\sim$problema(k1)

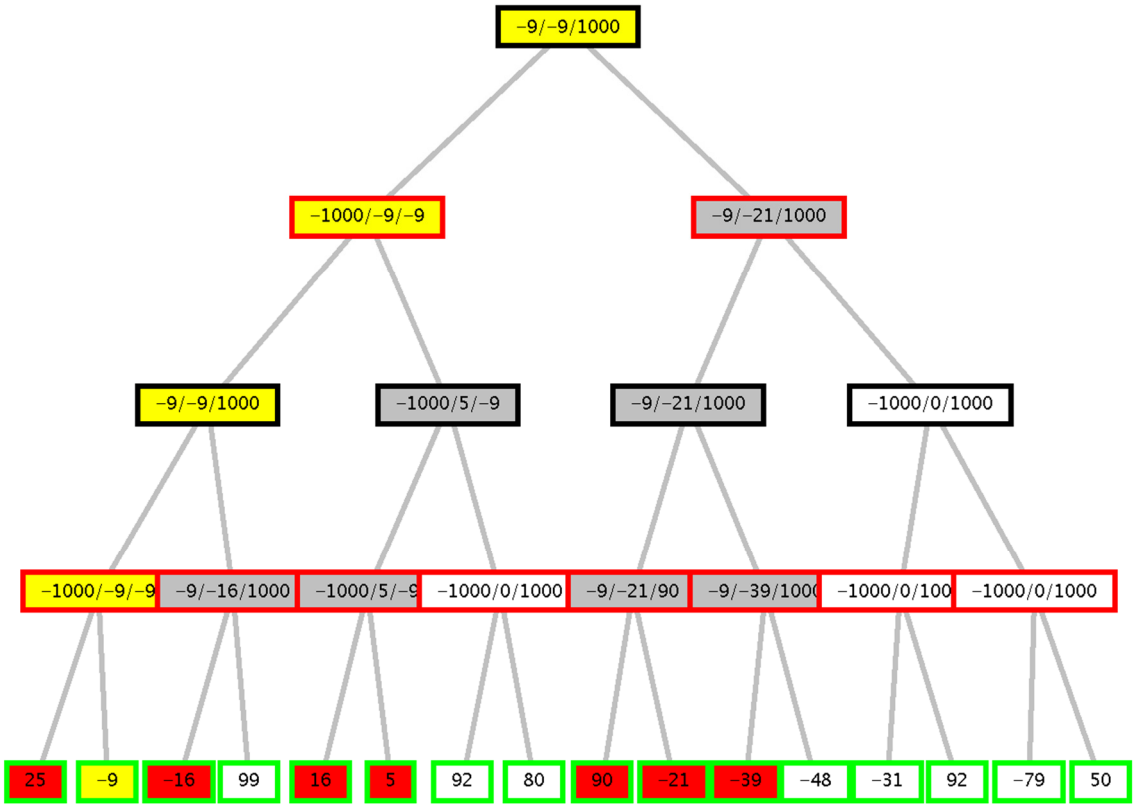
C10: C9 + C4.1: clausola vuota, contraddizione.

Esercizio 2

Min-Max:

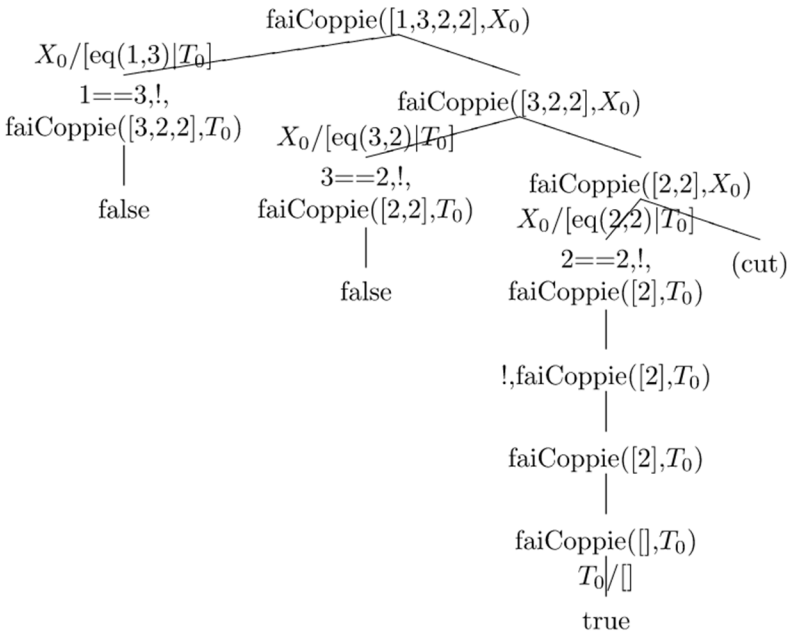


Alfa-beta:



I nodi che portano alla soluzione sono in giallo, quelli tagliati in bianco.

Esercizio 3



Esercizio 4

```
Soluzione col cut:
last_element([El], [], El) :- !.
last_element([El|Tail], [El|Rest], Last) :-
      last_element(Tail, Rest, Last).

Soluzione senza cut:
last_element([El], [], El).
last_element([El1, El2 | Tail], [El1|Rest], Last) :-
      last_element( [El2 | Tail], Rest, Last).

Soluzione senza cut, solo un poco meno efficiente:
last_element([El], [], El).
last_element([El|Tail], [El|Rest], Last) :-
     last_element(Tail, Rest, Last).

Soluzione alternative con uso di append:
last_element(L, R, LE) :-
    append(R, [LE], L).
```

Esercizio 5

Nodi espansi (in grassetto il nodo scelto):
S(0)
A(13) **B (5)**
A(13) **E(7)** F(20) G(8) H(17)
A(13) J(12) F(20) **G(8)** H(17)
A(13) J(12) F(20) **K(11)** H(17)
A(13) **J(12)** F(20) Goalc(16) H(17)
A(13) Goalb(15) F(20) GoalC(16) H(17)
C(19) D(21) **Goalb(15)** F(20) GoalC(16) H(17)

Percorso: S B E J Goalb.

Esercizio 6 & 7:

Si vedano le slide del Corso