FONDAMENTI DI INTELLIGENZA ARTIFICIALE (6 CFU)

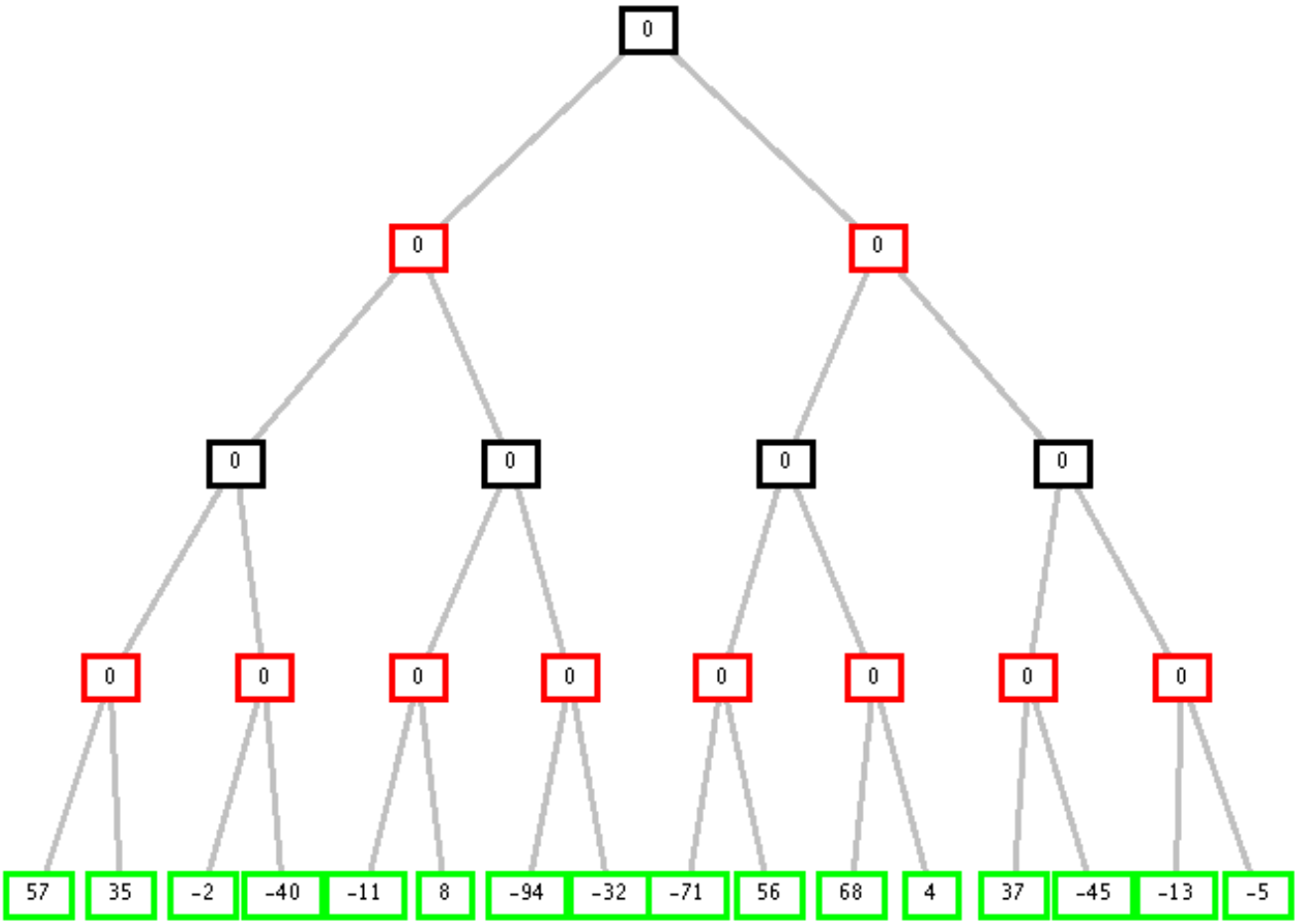
9 Luglio 2015 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (7 punti)

Si rappresentino in logica dei predicati del I ordine, le seguenti affermazioni:
(1) Per chiunque entri nel paese e non sia un VIP, allora esiste un funzionario di dogana che lo perquisisce
(2) Alcuni spacciatori di droga sono entrati nel paese, e tutti coloro che li hanno perquisiti erano anch'essi spacciatori di droga.
(3) Nessuno spacciatore è un VIP.
Si applichi la risoluzione alla teoria formata dalle formule ottenute sopra per verificare se vale l'affermazione seguente:
(Goal) Alcuni funzionari sono spacciatori di droga.
Si utilizzino i seguenti predicati:
e(X) per X è entrato, v(X) X è un VIP, p(X,Y) X ha perquisito Y, f(X) X è un funzionario, s(X) X è uno spacciatore.

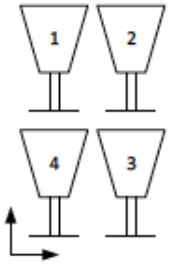
Esercizio 2 (5 punti)

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 gli algoritmi *min-max* e *alfa-beta* risolvono il problema.

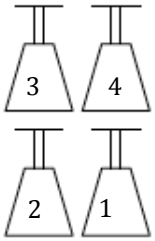


Esercizio 3 (7 punti)

Si consideri il rompicapo di spostare 4 bicchieri dalla configurazione (Init) a quella (Goal):



(Init)



(Goal)

I bicchieri sono numerati, e possono essere girati verso il basso. Si rappresenti lo stato tramite i termini Up(X)/Down(X) per indicare che il bicchiere X è girato in alto/in basso, e TopRow(X,Y) per indicare che i bicchieri X e Y sono nella fila superiore.

Per passare dallo stato Init allo stato Goal, si possono eseguire le seguenti azioni:

- **FlipDown(A,B)**, per girare dall'alto in basso i bicchieri A e B della riga superiore (ad esempio, A e B sono i bicchieri n. 1 e 2 nel caso della configurazione Init); i bicchieri possono essere girati verso il basso se e solo se sono entrambi girati verso l'alto.
- **Rotate**, per ruotare i bicchieri in senso antiorario di una posizione: ad esempio applicando Rotate allo stato Init, il bicchiere 1 si sposta in basso, il bicchiere 2 si sposta a sinistra, il bicchiere 3 si sposta nella riga in alto, e il bicchiere 4 si sposta a destra.

Si risolva il problema applicando la ricerca A* con eliminazione degli stati ripetuti. Si considerino costi unitari per le azioni e come funzione euristica il numero di bicchieri diretti verso l'alto. Si applichino le azioni nell'ordine riportato sopra (prima FlipDown, e poi Rotate) e, a parità di altro, si consideri il nodo generato per ultimo (quello più recente).

Si riporti l'albero di ricerca e il piano trovato. Sono stati eliminati nodi corrispondenti a stati ripetuti?

Esercizio 4 (5 punti)

Il predicato predefinito Prolog `atom(X)` serve per controllare che l'argomento sia atomico (numero o costante). Si scriva un predicato `contaatom(Atom,Lista,Numero)` che restituisce il Numero di volte in cui l'Atomo compare nella Lista. Esempi:

```
?-contaatom(a,[1,f(a),a,2,3,a],X).
yes, X=2
?-contaatom(a,[1,f(a),a,2,3,a],3).
no
```

Esercizio 5 (5 punti)

Si consideri il seguente programma Prolog, che realizza il merge di due liste di interi (ordinate):

```
merge([], L2, L2):-!.
merge(L1, [], L1):-!.
merge([X|REST1],[X|REST2],[X,X|REST]) :- !, merge(REST1,REST2,REST).
merge([X|REST1],[Y|REST2],[X|REST]):-
     X < Y, !, merge(REST1,[Y|REST2],REST).
merge([X|REST1],[Y|REST2],[Y|REST]):- merge([X|REST1],REST2,REST).
```

e si mostri l'albero SLD che si ottiene per il goal Prolog:

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

Esercizio 6 (3 punti)

Si spieghi cosa è un constraint graph per un problema di CSP e si definiscano i diversi livelli di consistenza da quella di I grado (node consistency) al grado k.

FONDAMENTI DI INTELLIGENZA ARTIFICIALE

9 Luglio 2015 – Soluzioni

Esercizio 1

1. Per chiunque entri nel paese e non sia un VIP, allora esiste un funzionario di dogana che lo perquisisce
   $\forall X (e(X) \wedge \neg v(X) \longrightarrow \exists Y (f(Y) \wedge p(Y, X)))$

2. Alcuni spacciatori di droga sono entrati nel paese, e tutti coloro che li hanno perquisiti erano anch'essi spacciatori di droga
   $\exists X (s(X) \wedge e(X) \wedge \forall Y(p(Y, X) \longrightarrow s(Y)))$

3. Nessuno spacciatore è un VIP
   $\forall X (s(X) \longrightarrow \neg v(X))$

Goal: $\exists X\ f(X) \wedge s(X)$

Trasformazione in clausole:

1a: $\neg e(X) \vee v(X) \vee f(sko(X))$.
1b: $\neg e(X) \vee v(X) \vee p(sko(X), X)$.
2a: $s(a)$.
2b: $e(a)$.
2c: $\neg p(Y, a) \vee s(Y)$.
3: $\neg s(X) \vee \neg v(X)$.

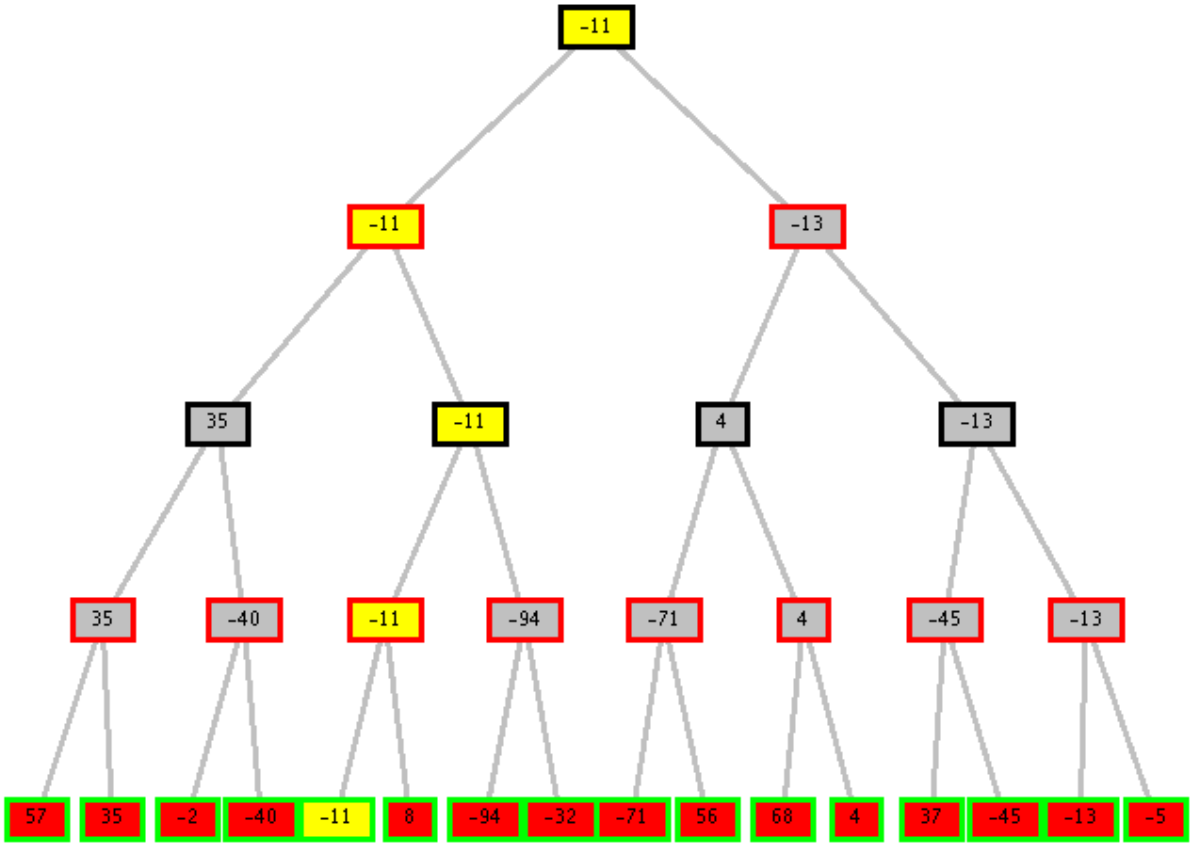
GNeg: $\neg f(X) \vee \neg\ s(X)$.

Risoluzione:

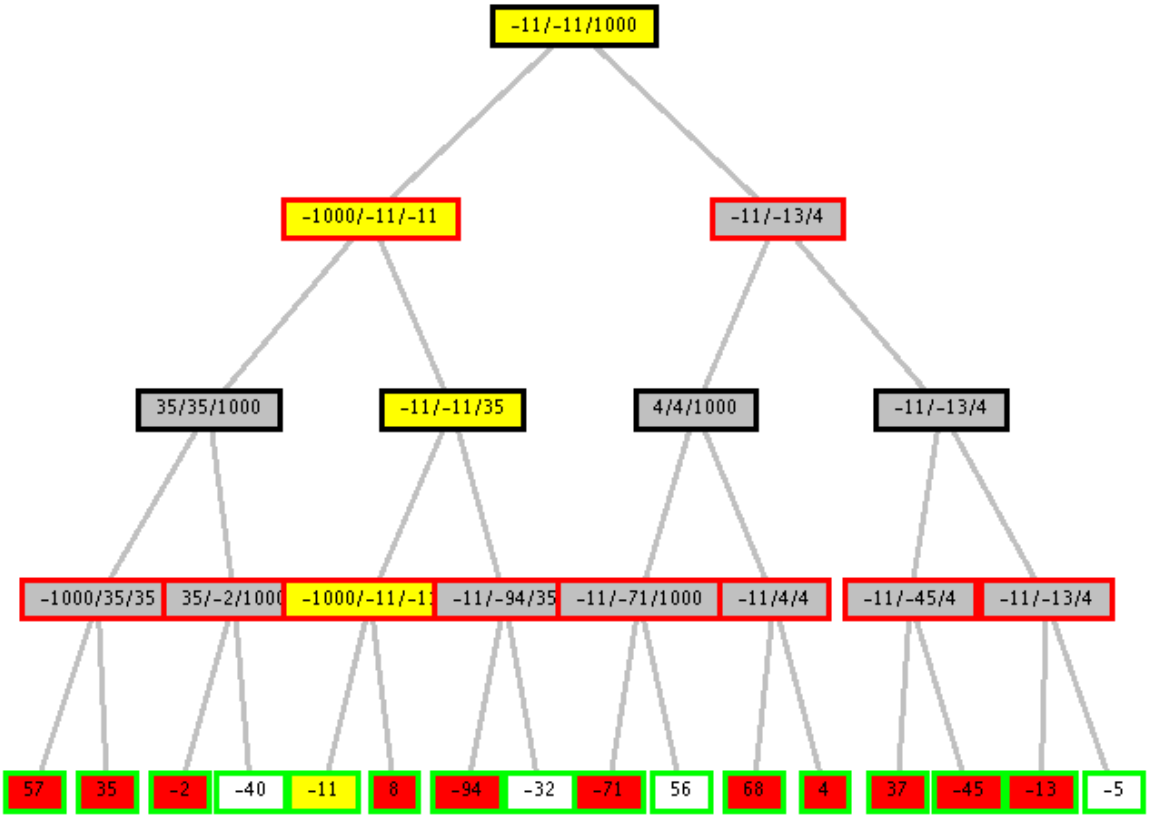
4 (GNeg+1a): $\neg e(X) \vee v(X) \vee \neg s(sko(X))$.
5 (4+2b): $v(a) \vee \neg s(sko(a))$.
6 (5+3): $\neg s(a) \vee \neg s(sko(a))$.
7 (6+2a): $\neg s(sko(a))$.
8 (7+2c): $\neg p(sko(a), a)$.
9 (8+1b): $\neg e(a) \vee v(a)$.
10 (9+2b): $v(a)$.
11 (10+3): $\neg s(a)$.
12 (11+2a): $\square$

Esercizio 2

Min-Max:

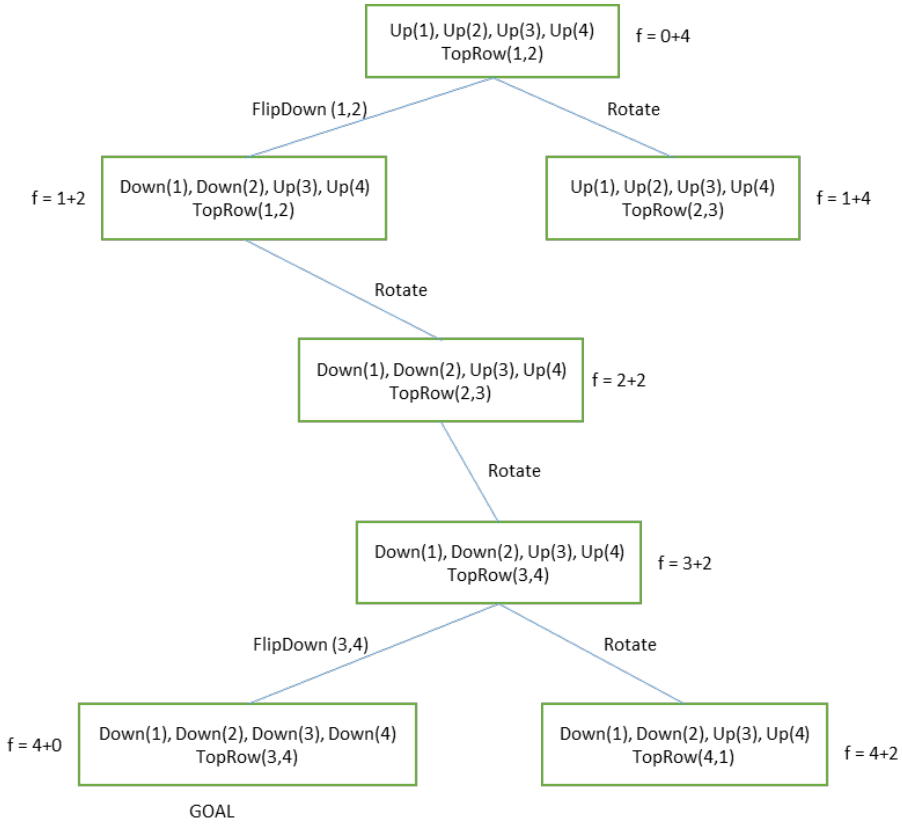


Alfa-Beta:



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

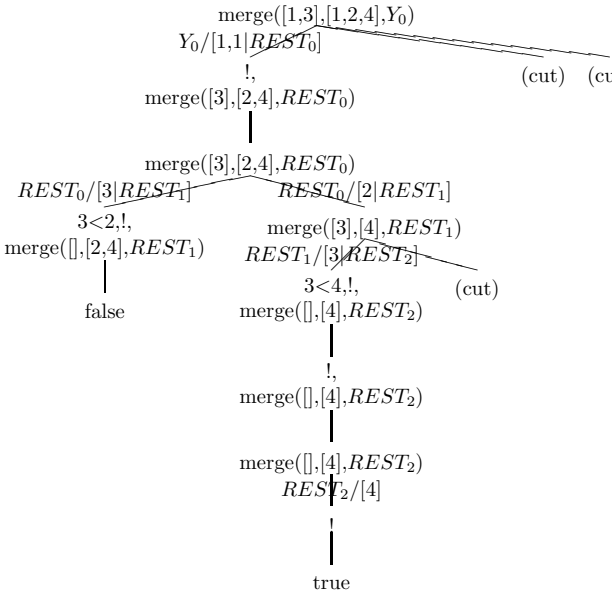
Esercizio 3



Esercizio 4

```
contaatom(_,[],0).
contaatom(Atomo,[Primo|Resto],N):-
      atom(Primo), Primo=Atomo, !,
      contaatom(Atomo,Resto,N1), N is N1 + 1.
contaatom(Atomo,[X|Resto],N):- contaatom(Atomo,Resto,N).
```

Esercizio 5



Esercizio 6 Vedi slide del corso.