FONDAMENTI DI INTELLIGENZA ARTIFICIALE

26 Gennaio 2017 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (6 punti)

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

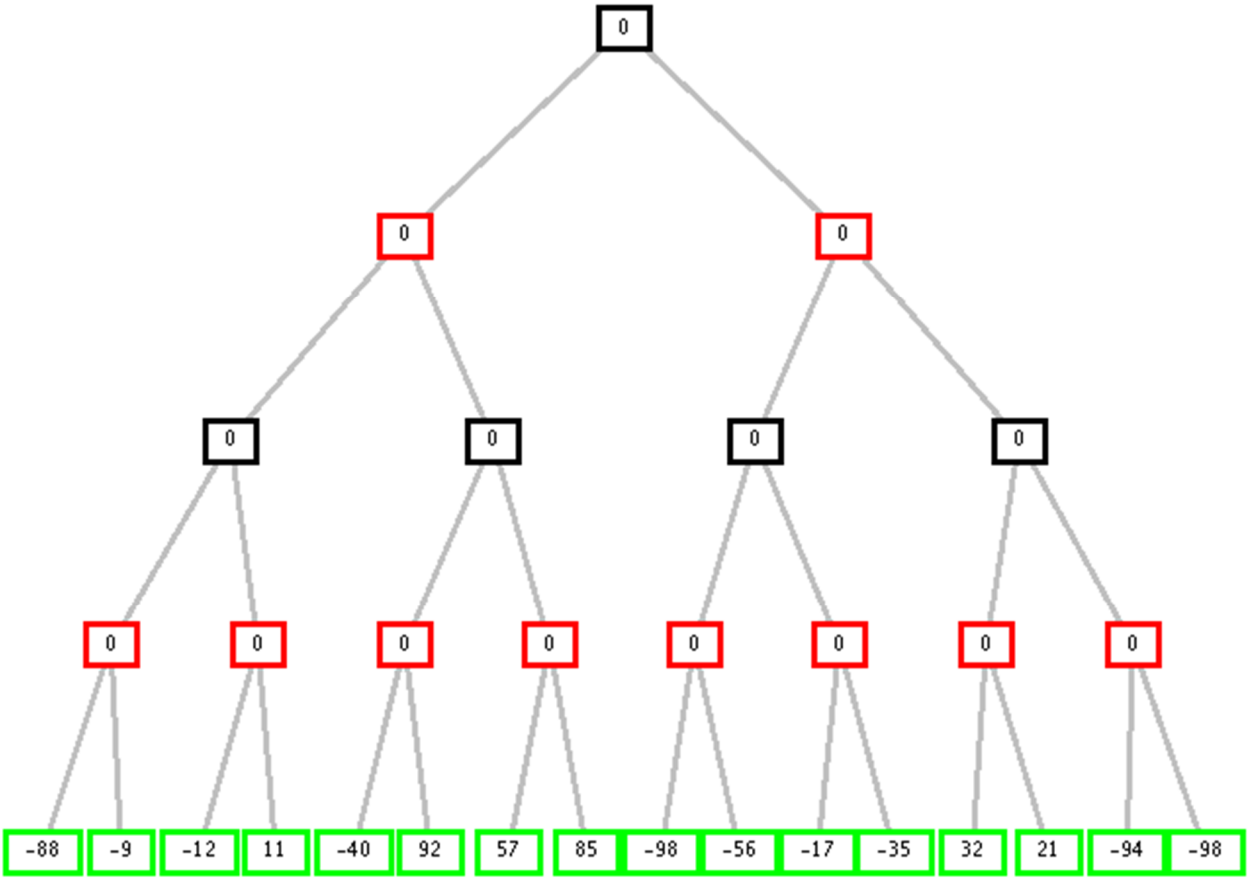
- I cani sono di piccola o (or-ex) grande taglia
- I cani di piccola taglia stanno in appartamento
- I cani di grande taglia stanno in giardino
- I pastori tedeschi sono cani di grande taglia
- Ted è un Pastore tedesco.

Le si trasformi in clausole e si usi la risoluzione per dimostrare che:

- Ted sta in giardino.

Esercizio 2 (4 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:

```
bubble(L,S):- swap(L,L1),!, bubble(L1,S).
bubble(S,S).
swap([X,Y|R],[Y,X|R]):- X>Y.
swap([Z|R],[Z|R1]):- swap(R,R1).
```

si disegni l'albero SLD relativo al goal:

```
?-bubble([1,2,3],M0).
```

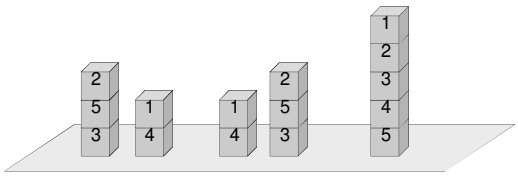
Esercizio 4 (4 punti)

Si definisca un predicato Prolog `listPosition(L,X,N)` che, data una lista `L` e un elemento `X`, restituisce la posizione `N` dell'elemento che si sta cercando al suo interno a partire dall'ultima posizione. Si abbia cura di definire anche eventuali predicati di supporto. Esempio:

```
?- listPosition([a, b, c, d, e], a, P).
yes P is 5
?- listPosition([a, b, c], d, P).
no
```

Esercizio 5 (5 punti)

Si consideri un gioco in cui si hanno a disposizione cinque cubi numerati da 1 a 5, i quali possono essere disposti in vari modi sopra un tavolo (in figura si mostrano tre possibili disposizioni; in una disposizione si possono avere ovviamente fino a 5 pile distinte). Ciascuno dei blocchi può trovarsi sul tavolo oppure sopra un altro blocco. Partendo da una disposizione qualsiasi, l'obbiettivo consiste nel disporre i blocchi in un'unica pila, con il blocco 5 sul tavolo e gli altri sopra di esso nell'ordine indicato nella figura a destra, spostando il minor numero possibile di blocchi. L'unica azione ammessa è lo spostamento di un blocco sul tavolo oppure sopra un altro blocco, purchè sia il blocco da spostare sia l'eventuale blocco su cui questo viene posato non abbiano altri blocchi al di sopra di essi. La posizione relativa delle diverse pile di blocchi non conta (per esempio, la disposizione nella figura a sinistra è equivalente a quella al centro).



(a) Formulare tale problema come un problema di ricerca su grafi, definendo in modo preciso lo spazio degli stati, lo stato obbiettivo, le azioni, e il costo di ogni azione.

(b) Considerando come funzione euristica il numero di blocchi che non si trovino sulla pila più alta (o su una delle più alte, nel caso di più pile aventi la stessa altezza), e come stato di partenza quello nella figura a sinistra, costruire l'albero di ricerca ottenuto con l'algoritmo A* espandendo i primi quattro nodi, ed evitando gli stati ripetuti.

Esercizio 6 (6 punti)

Quattro amiche, Anna, Bea, Carla e Didi, sono andate a fare shopping e a comprare dei gratta e vinci in un centro commerciale. Sappiamo che in ingresso le amiche avevano ciascuna un importo diverso da quelli delle altre, e pari a 100, o 200, o 300, o 400 euro. Uscendo dal centro commerciale, cogliamo le amiche dirsi tra loro:

Anna: Ho guadagnato! Sono uscita con 100 euro in più di ciò che Bea aveva quando è entrata;

Didi: Non ho più un euro! Ho speso la stessa somma che ha speso Bea; ma a lei qualcosa è rimasto, ma era tra noi quella con più soldi!

Carla: Vi offro qualcosa, ho guadagnato una somma pari al doppio di ciò che ha speso Didi e ora ho gli stessi soldi che ha Anna.

Ciascuna delle amiche è uscita con una somma pari a 0, 100, 200, 300, 400 o 500 euro.

Si modelli il problema come un CSP, dove le variabili rappresentano il denaro che le amiche hanno prima di fare shopping (A, B, C, D, rispettivamente), e il denaro che hanno dopo lo shopping (A', B', C' e D').

Si applichi poi alla rete iniziale la tecnica dell'arc-consistenza (si considerino a tale scopo solo i vincoli unari e binari nell'applicazione della arc consistenza).

Esercizio 7 (1 punti)

Si introduca brevemente in cosa consiste il Test di Turing.

FONDAMENTI DI INTELLIGENZA ARTIFICIALE

26 Gennaio 2016 – Soluzioni

Esercizio 1

1) ∀ X cane(X) -> (piccolo(X) ex-or grande(X))
2) ∀ X cane(X) and piccolo(X) -> appartamento(X)
3) ∀ X cane(X) and grande(X) -> giardino(X)
4) ∀ X pastoreTedesco(X) -> grande(X) and cane(X)
5) pastoreTedesco(ted)

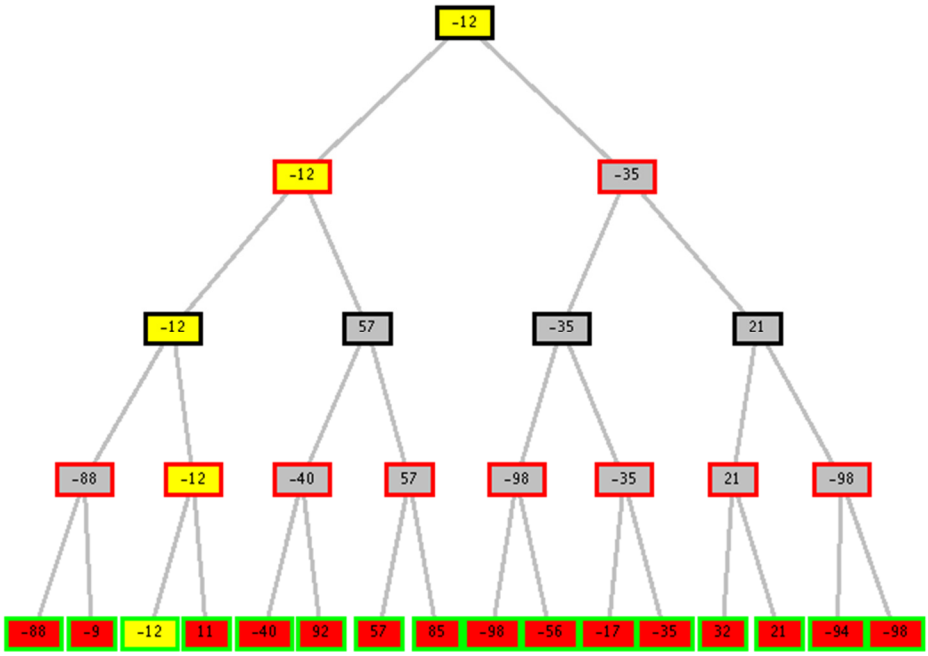
Query: giardino(ted)

Clausole:
C1a: not cane(X) or piccolo(X) or grande(X)
C1b: not cane(X) or not piccolo(X) or not grande(X)
C2: not cane(X) or not piccolo(X) or appartamento(X)
C3: not cane(X) or not grande(X) or giardino(X)
C4a: not pastoreTedesco(X) or grande(X)
C4b: not pastoreTedesco(X) or cane(X)
C5: pastoreTedesco(ted)
GNeg: not giardino(ted)

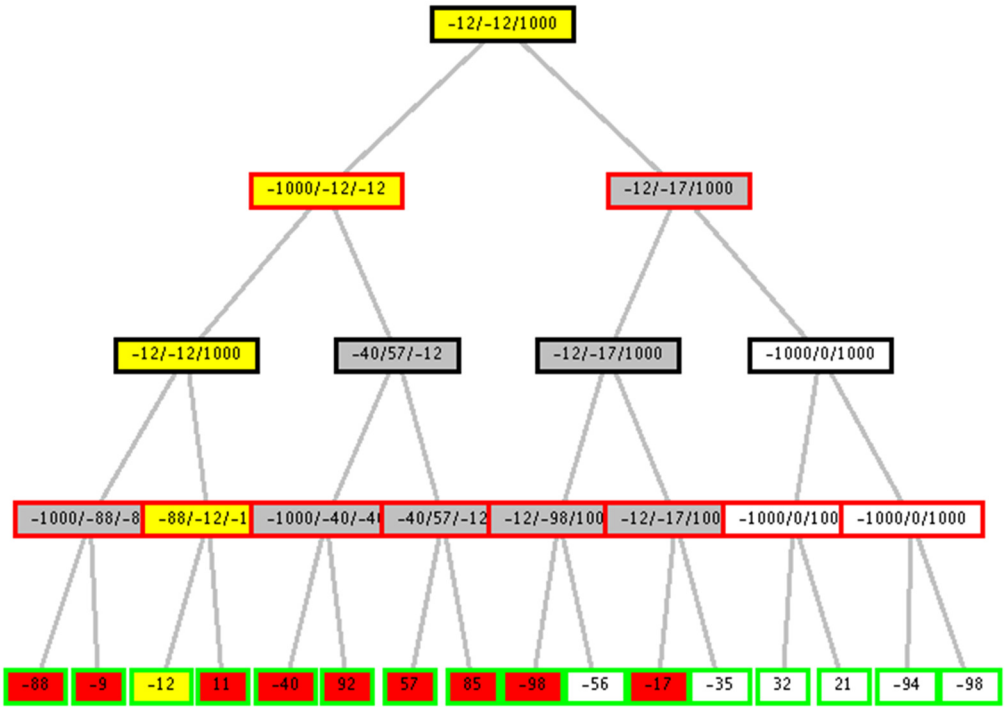
Risoluzione:
C6: C5+C4b: cane(ted)
C7: C5+C4a: grande(ted)
C8: C6+C3: not grande(ted) or giardino(ted)
C9: C8+C6: giardino(ted)
C10: C9 + GNeg : clausola vuota
Oppure:
C6: GNeg+C3: not cane(ted) or not grande(ted)
C7: C6+C4b: not pastoreTedesco(ted) or not grande(ted)
C8: C7+C4a: not pastoreTedesco(ted)
C9: C8+C5: clausola vuota.

Esercizio 2

Min-Max:

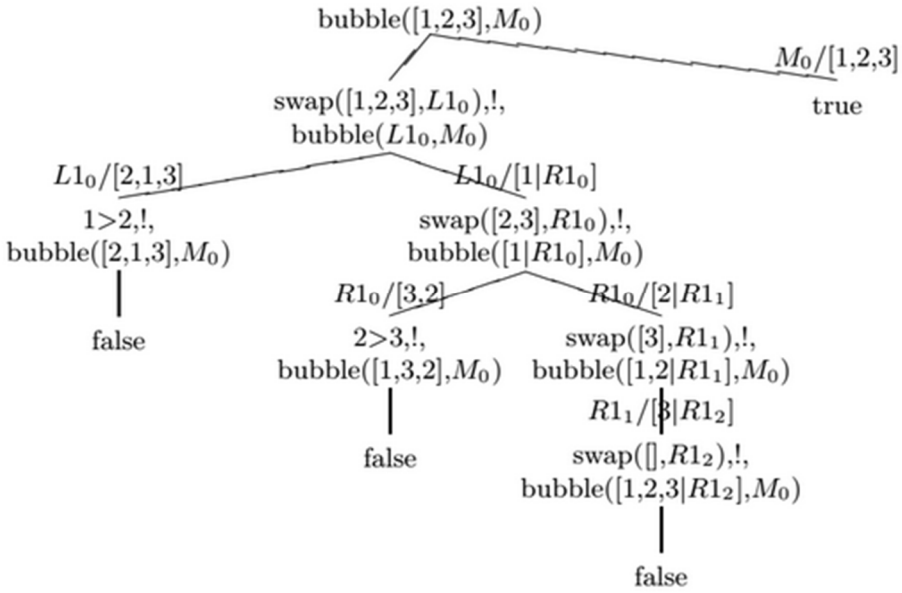


Alfa-beta:



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

Esercizio 3



Esercizio 4

```
listPosition([X|T], X, A) :-
     !,
     length(T,Len),
     A is Len+1.
listPosition([_|T], X, A) :-  listPosition(T, X, A).

length([], 0).
length([_|T], Len) :- length(T, Len1), Len is Len1+1.
```

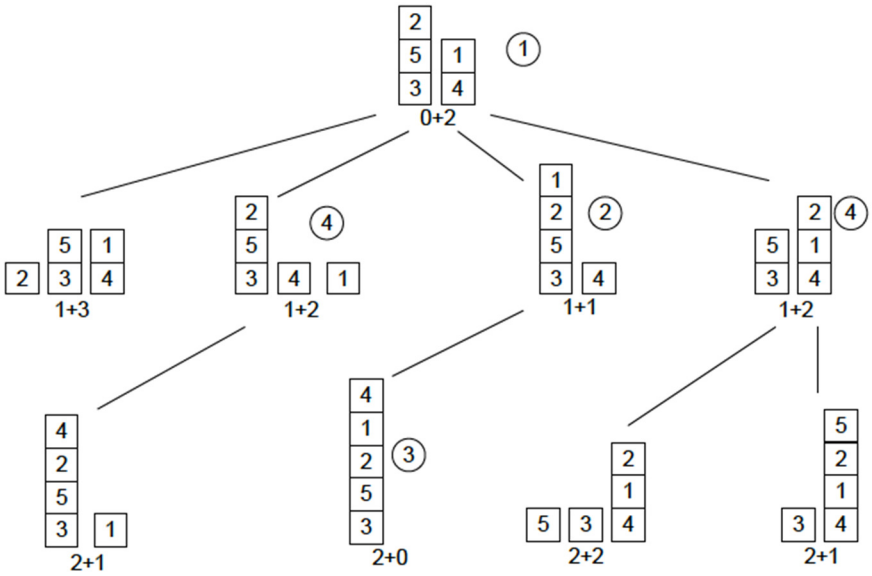
Esercizio 5

(a) Lo spazio degli stati è costituito dall'insieme delle possibili disposizioni dei blocchi in una o più pile (fino a cinque), indipendentemente dalla posizione relativa di queste ultime. Uno stato può essere rappresentato da un insieme di sequenze ordinate, ciascuna delle quali corrisponde a una pila; per es., la disposizione a sinistra e quella al centro in figura (equivalenti) sono rappresentate da $\{(2,5,3),(1,4)\}$,
seguendo la convenzione che la sequenza dei blocchi di ogni pila da sinistra a destra corrisponde alla loro disposizione dall'alto in basso. Uno solo di tali stati sarà lo stato obbiettivo.

Le azioni possono essere definite attraverso una funzione successore SF (s) che, dato uno stato s (rappresentato come indicato sopra), restituisce tutte le coppie (s^r, a), dove s^r è uno stato raggiungibile da s con l'azione a, e quest'ultima è una delle azioni ammesse.

Tutte le azioni hanno lo stesso costo, dato che si vuole arrivare allo stato obbiettivo spostando il minor numero di blocchi. Il costo può essere assunto pari a 1.

(b) L'albero di ricerca generato espandendo i primi quattro nodi con la strategia A* e l'euristica indicata, ed evitando gli stati ripetuti, è il seguente.



Sotto ogni stato è indicato il valore corrispondente della funzione di valutazione (*path cost* + funzione euristica). I numeri indicati nei cerchietti indicano l'ordine di **espansione** dei nodi. Si noti che il terzo nodo selezionato per l'espansione non ha successori, dato che l'unico stato raggiungibile corrisponde a quello del nodo padre. Per il quarto nodo esistono due possibili scelte, cioè due nodi con lo stesso valore della funzione di valutazione (indicati entrambi con il numero 4): in figura è mostrato il risultato che si otterrebbe espandendo sia un nodo che l'altro.

Esercizio 6

Variabili: le somme possedute dalle 4 amiche A, B, C, D in ingresso e quelle possedute in uscita A', B', C', D'
Dominii: (1..4) per A, B, C, D e valore intero per quelle in uscita
A:: [1,2,3,4]
B:: [1,2,3,4]
C:: [1,2,3,4]
D:: [1,2,3,4]
A':: [0,1,2,3,4,5]
B':: [0,1,2,3,4,5]

C':: [0,1,2,3,4,5]
D':: [0,1,2,3,4,5]
Vincoli:
- Variabili A, B, C, D diverse tra loro:

$A \neq B \neq C \neq D$

Anna: Ho guadagnato! Sono uscita con 100 euro in più di ciò che Bea aveva quando è entrata;

A' > A

A' = 1 + B

Didi: Non ho più una lira! Ho speso la stessa somma che ha speso Bea; ma a lei qualcosa è rimasto, ma era tra noi quella con più soldi!

D' = 0

D – D' = B – B'

D' < D

B' < B

B = 4 (>A,C,D)

B' > 0

Carla: Vi offro qualcosa, ho guadagnato una somma pari al doppio di ciò che ha speso Didi e ora ho gli stessi soldi che ha Anna.

C' > C

C' – C = 2* (*D – D') = 2*D essendo D'=0*

C' = A'

Arc consistenza
A:: [1,2,3]
B:: [4]
C:: [1,2,3]
D:: [1,2,3]
A':: [5]
B':: [1, 2, 3]
C':: [5]
D':: [0]
Labeling:

Esercizio 7

Vedi slide.