FONDAMENTI DI INTELLIGENZA ARTIFICIALE

7 Luglio 2016 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

Esercizio 1 (6 punti)

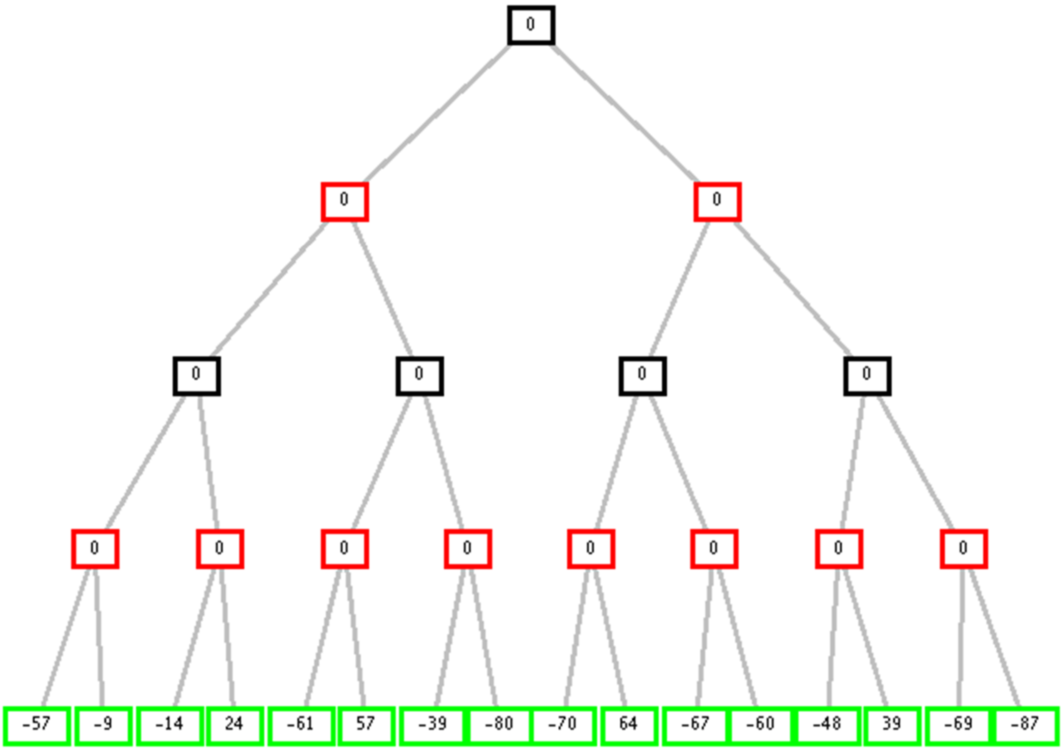
Si traducano le seguenti frasi nella logica dei predicati del primo ordine, poi in forma a clausole:

1. Le persone si dividono in genitori e non genitori
2. Ogni genitore ha almeno un figlio
3. Paolo è una persona
4. Claudio è una persona
5. Claudio non ha alcun figlio

Si usi poi il principio di risoluzione per dimostrare che Claudio non è un genitore. Si usino a tal scopo i predicati **persona(X)**, **genitore(X)**, **figlio(X,Y)** (Y è figlio di X).

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 (5 punti)

Un grafo aciclico G è rappresentato come una lista di coppie [X,Y], dove X e Y sono nodi di G, e [X,Y] appartiene a G se e solo se esiste un arco orientato da X verso Y. Il predicato `grafo/2` indica con un numero il nome del grafo, e con la lista di coppie i suoi archi. Ad esempio, un grafo è rappresentato dal fatto Prolog:

```
grafo(1,[[a,b],[a,c],[c,b],[b,d],[c,d]]).
```

Si scriva un predicato Prolog `camm(G,X,Y)` che è vero se esiste nel grafo G un cammino dal nodo X al nodo Y. Il candidato abbia cura di riportare anche eventuali predicati di supporto. Goal d'esempio:

```
?- camm([[a,b],[a,c],[c,b],[b,d],[c,d]], a, d).
yes
?- grafo(1,G), camm(G,c,X).
X = b;
X = d;
X = d;
no
```

Esercizio 4 (5 punti)

Si dato il seguente programma Prolog:

```
p(a).
p(X):-p(f(X)).
p(f(b)).
```

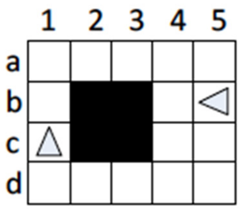
e la query:

```
?-p(Y).
```

si mostri l'albero di derivazione SLD relativo alla query (non sviluppare rami oltre il livello di profondità 3 nel caso di rami infiniti).

Esercizio 5 (6 punti)

Si consideri il seguente ambiente rappresentato come una griglia di celle:



Un agente robotico è nella cella (b,5) ed è orientato verso ovest, come indicato in figura, e deve raggiungere la cella (c,1) con orientamento verso nord. Le azioni che l'agente può compiere sono: (1) spostarsi nella cella adiacente nella direzione del suo orientamento e (2) modificare il proprio orientamento di 90° in senso orario, rimanendo nella stessa cella (gli orientamenti possibili sono: nord, sud, ovest, est). L'agente non può uscire dall'ambiente né occupare le celle ostacolo (in nero).

1. Formulare il problema come problema di ricerca, indicando chiaramente cosa sia uno "stato". Le azioni possibili sono due: *avanza* e *ruota*.
2. Si risolva il problema di ricerca usando l'algoritmo A* con eliminazione degli stati ripetuti. A tal scopo si consideri una funzione euristica che restituisce la minima distanza dalla cella (c,1) espressa in numero di celle, considerando la presenza di ostacoli. Per esempio, la distanza della cella (c,4) dalla cella (c,1) secondo tale euristica è 5. Le due azioni *avanza* e *ruota* hanno costo unitario. In fase di espansione si applichi sempre prima l'azione *avanza* e poi l'azione *ruota*. Riportare i primi 10 nodi dell'albero di ricerca generato (compresa la radice) e, per ogni nodo, il valore della funzione euristica. A parità di altro, si espanda il nodo generato prima.
3. L'euristica considerata è ammissibile? E' consistente? Si trova il cammino ottimo?

Esercizio 6 (4 punti)

Si consideri il seguente problema di soddisfacimento di vincoli:
Variabili:

- X1, dominio D1 = { 0, 1, 2 }
- X2, dominio D2 = { a, b }
- X3, dominio D3 = { f, g }

Vincoli (combinazioni permesse):

- C1(X1,X2) = { <0,a>, <0,b>, <1,b> }
- C2(X1,X3) = { <0,f>, <1,f>, <2,f>, <2,g> }
- C3(X2,X3) = {<a,f>, <a,g>, <b,g>}

Si applichi a tale problema l'arc consistency (si indichino gli elementi eliminati man mano dai domini ed i vincoli che ne hanno permesso l'eliminazione) e si scriva la soluzione eventualmente trovata per il problema di soddisfacimento di vincoli.

Esercizio 7 (2 punti)

Si descrivano sinteticamente come lavorano gli algoritmi genetici ed i loro contesti di applicazione.

FONDAMENTI DI INTELLIGENZA ARTIFICIALE
16 Giugno 2016 – Soluzioni

Esercizio 1

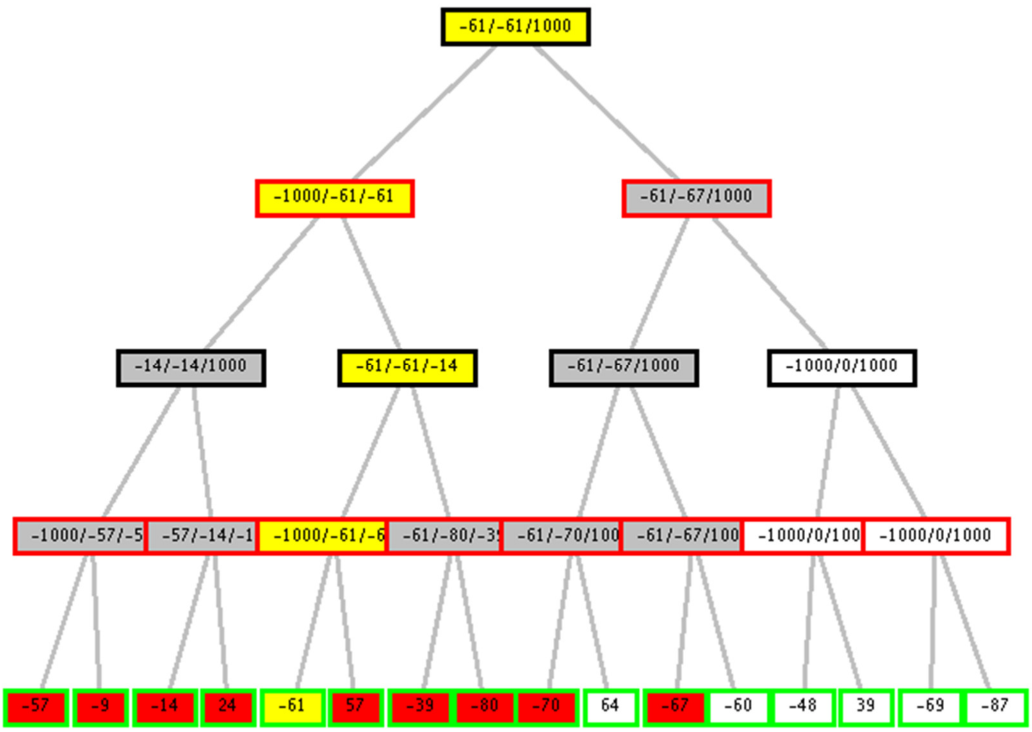
1: ∀X (persona(X) ⟶ genitore(X) xor ¬genitore(X)).
2: ∀X (genitore(X) ⟶ ∃Y (persona(Y) ∧ figlio(X,Y))).
3: persona(paolo).
4: persona(claudio).
5: ¬ ∃Y figlio (claudio,Y).
Goal: ¬ genitore(claudio).

Trasformazione in clausole:
C1a: ¬persona(X) ∨ genitore(X) ∨ ¬genitore(X).
C1b: ¬persona(X) ∨ ¬genitore(X) ∨ genitore(X).
C2a: ¬genitore(X) ∨ persona(f(X)).
C2b: ¬genitore(X) ∨ figlio(X, f(X)).
C3: persona(paolo).
C4: persona(claudio).
C5: ¬figlio(claudio,Y).
GNeg: genitore(claudio).

Applicando il Principio di Risoluzione:
C6: GNeg+C2b: figlio(claudio, f(claudio)).
C7: C6+C5: clausola vuota.

Esercizio 2

Si riporta la soluzione per l'algoritno tagli alfa-beta:

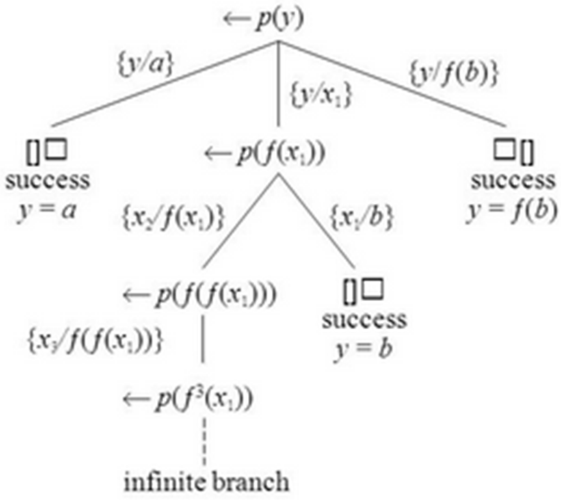


I nodi che portano alla soluzione sono in giallo (gli stessi di min-max), quelli tagliati da alfa-beta in bianco.

Esercizio 3

```
camm(G,X,Y) :-
          member([X,Y],G).
camm(G,X,Y) :-
          member([X,Z],G),
          camm(G,Z,Y).
member(El, [El|_]).
member(El, [_|Tail]) :- member(El, Tail).
```

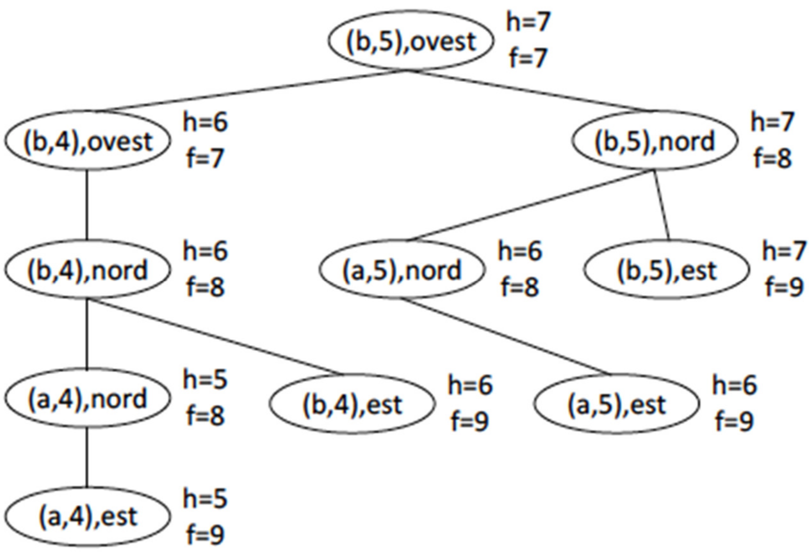
Esercizio 4



Esercizio 5

1. Lo stato rappresenta la posizione dell'agente e il suo orientamento.
Stato iniziale: s0=[(b,5),ovest].
AZIONI(s) = {avanza, ruota} (se in s non è possibile avanzare a causa di un ostacolo o del bordo dell'ambiente, AZIONI(s) = {ruota}).
RISULTATO(s,a) = s', dove s' rappresenta una nuova cella e lo stesso orientamento di s se a = avanza o la stessa cella e un nuovo orientamento se a = ruota.
Test obiettivo: lo stato corrente è [(c,1),nord]?
Costo di passo: è unitario, dal testo.

2. Porzione dell'albero di ricerca:



3. L'euristica considerata è chiaramente ammissibile in quanto non considerando le eventuali rotazioni non sovrastima mai il costo per raggiungere la cella (c,1).
E' facile verificare che l'euristica è anche consistente (i costi di passo sono unitari e l'euristica di due stati di cui uno differisce al più di 1). Per questo motivo, la soluzione trovata da A* con eliminazione degli stati ripetuti è ottima.

Esercizio 6
Gli elementi successivamente eliminati per l'arc consistency sono i seguenti:

- Si elimina "2" da D1 a causa del vincolo C1 (poiché in virtù di C1 non c'è alcun elemento del dominio di X2 che sia compatibile con l'elemento "2" di D1);
- Si elimina "g" da D3 a causa del vincolo C2;
- Si elimina "b" da D2 a causa del vincolo C3;
- Si elimina "1" da D1 a causa del vincolo C1.

A questo punto i domini sono:
D1 = { 0 }
D2 = { a }
D3 = { f }
Ogni arco è consistente, per cui per l'arc consistency non si possono eliminare più elementi. Inoltre, i domini restanti contengono esattamente un elemento ciascuno: tali elementi sono consistenti fra loro e costituiscono, dunque, una soluzione.
In conclusione, viene trovata la seguente soluzione al problema di soddisfacimento di vincoli dato:
< X1, X2, X3 > = < 0, a, f >

Esercizio 7
Vedi slide.