FONDAMENTI DI INTELLIGENZA ARTIFICIALE

10 Settembre 2020 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

NOTA: Consegnare la soluzione tramite un singolo file, che lo studente avrà cura di nominare come: CognomeNomeDataAI Ad esempio: RossiMario20200910AI

Esercizio 1 (6 punti)

Si considerino le seguenti frasi:

1. *Tutti i bambini amano i loro giocattoli*
2. *Giorgio è un bambino*
3. *Giorgio ha almeno un giocattolo*
4. *Se un bambino ama qualcosa, allora non lo butta.*

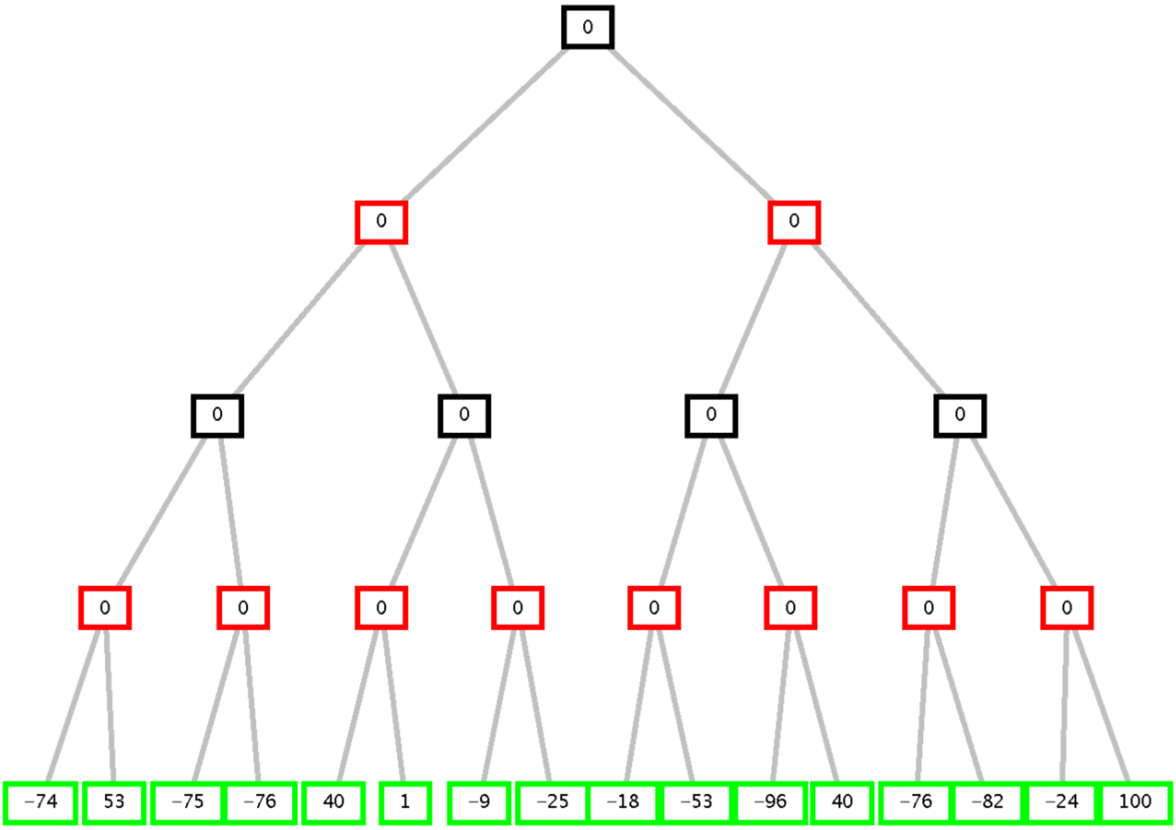
Si formalizzino in *logica dei predicati del I ordine*, utilizzando i seguenti predicati:

- `bambino(X)` *X è un bambino*
- `ama(X, Y)` *X ama Y*
- `ha_gioco(X, Y)` *X ha il giocattolo Y*
- `butta(X, Y)` *X butta Y*

Infine si trasformino in clausole e si dimostri, applicando la risoluzione, che *esiste qualcosa che Giorgio non butta.*

Esercizio 2 (5 punti)

Si consideri il seguente albero di gioco in cui il primo giocatore è MAX. Si indichi come l'algoritmo min-max risolve il problema indicando il valore con cui viene etichettato il nodo iniziale e la mossa selezionata dal primo giocatore. Si mostrino poi i tagli che l'algoritmo alfa-beta consente indicando gli archi che verranno tagliati. Si indichino i nomi degli archi iniziando con la lettera "**a**" e facendola seguire con un numero crescente da sinistra a destra e dall'alto al basso. Ad esempio, i due archi che si dipartono dalla radice saranno nominati a1 (quello più a sinistra) e a2. L'arco che connette il nodo foglia più a sinistra (con valore -74) sarà denominato a15, mentre l'ultimo arco che connette il nodo foglia più a destra (valore 100) a30.



Esercizio 3 (6 punti)
Si consideri un problema di scheduling con tasks a, b, c, d con le seguenti caratteristiche:
durata: 2, 3, 5, 4 ore rispettivamente;

1. b precede d.
2. a precede b.
3. a precede c.
4. c non può finire dopo l'ora 9.
5. d non può finire dopo l'ora 9.

Si modelli il problema come CSP, rappresentando rispettivamente le variabili coi tempi di inizio dei tasks: Ta, Tb, Tc, Td e con domini delle varabili gli interi che vanno da 0 .. 10. Si scrivano quindi i vincoli del problema. I primi tre saranno vincoli binari (ad esempio, il primo vincolo per esprimere che *b precede d* sarà Tb $+ 3 \leq$ Td), mentre gli ultimi due unari (ad esempio, il quarto vincolo per esprimere che *c non può finire dopo l'ora 9* sarà Tc $+ 5 \leq 9$).
Si applichi poi l'algoritmo di node-consistency e si indichino i domini (ridotti) risultanti per ogni variabile.
Successivamente al risultato si applichi l'algoritmo di arc-consistency e si indichino i domini (ridotti) risultanti per ogni variabile. Nota: i vincoli binari sono concepiti nell'algoritmo di arc-consistency come rappresentati da due archi orientati. Ad esempio Tb $+ 3 <$ Td esprime fra i nodi Td e Tb due archi , uno da Td verso Tb (Td,Tb) che porterà ad una possibile riduzione del dominio di Td e, analogamente, un altro da Tb verso Td (Tb,Td) che porterà ad una possibile riduzione del dominio di Tb.

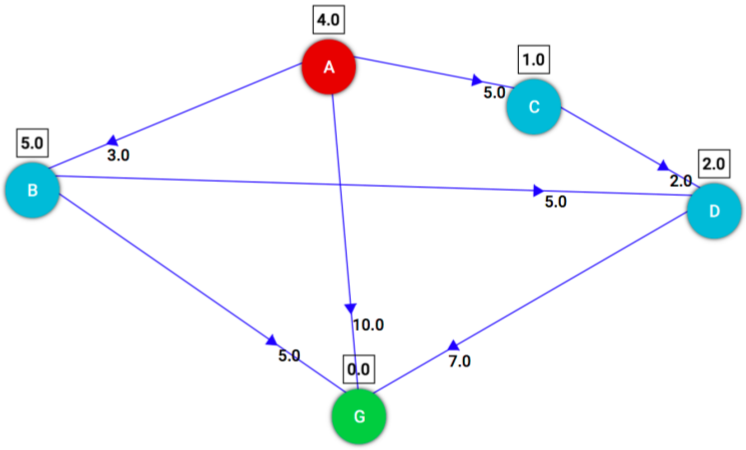
Esercizio 4 (5 punti)
Si scriva un predicato PROLOG:
`negList(P,L)`
che data una lista `P` di liste di interi, non vuote e contenenti ciascuna un (solo) elemento negativo e zero o più positivi, crei una nuova lista `L` contenente come elementi tali elementi negativi. Se la lista `P` è vuota anche `L` risulterà vuota.
Ad esempio:

```
?- negList([[3,-4,3,1], [-2,2], [-1]], L).
L = [-4, -2, 1]
```

```
?- negList([],L).
L = []
```

Esercizio 5 (6 punti)

Si consideri il seguente grafo, dove A è il nodo iniziale e G il nodo goal, e il numero associato agli archi è il costo dell'operatore per andare dal nodo di partenza al nodo di arrivo dell'arco. Vicino ad ogni nodo, in un quadrato, è indicata inoltre la stima euristica della sua distanza dal nodo goal G:



a) Si applichi la ricerca **A*** e si indichino i nodi espansi nell'ordine di espansione. In caso di non-determinismo, si scelgano i nodi da espandere in base all'ordine alfabetico del loro nome. Si consideri come euristica h(n) quella indicata nel quadrato a fianco di ogni nodo in figura.

b) Qual è il costo di cammino trovato per arrivare al goal G a partire dal nodo iniziale A?

c) La soluzione trovata in questo caso è ottimale? (motivare la risposta).

d) In generale, quali sono le condizioni che garantiscono l'ottimalità della ricerca A*?

Esercizio 6 (4 punti)

Dopo avere spiegato brevemente il predicato predefinito Prolog: `not(X)` si consideri il seguente programma Prolog:

```
p(3).
c(2,Z):- not (p(Z)).
c(1,3).
```

e si indichino le risposte dell'interprete Prolog alle seguenti query:

```
?- c(Y,3).
?- c(Y,4).
```

10 Settembre 2020 - Soluzioni

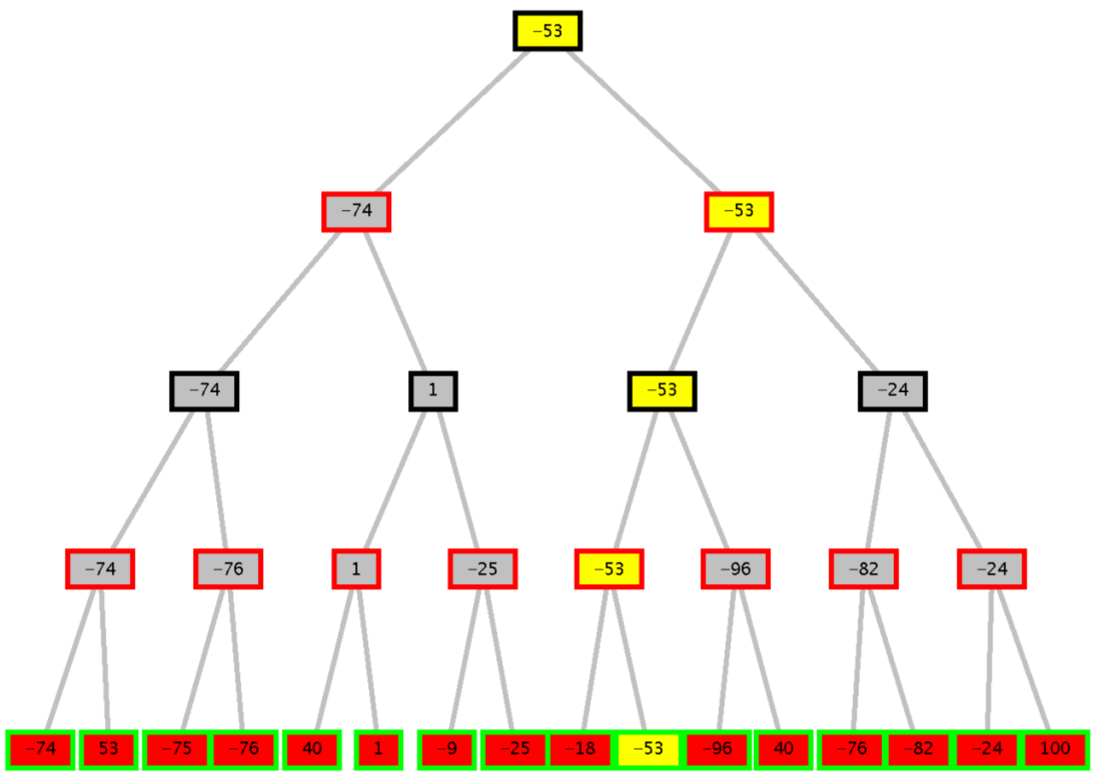
Esercizio 1
1.: ∀X ∀Y bambino(X) ∧ ha_gioco(X,Y)→ama(X,Y).
2.: bambino(giorgio).
3.: ∃Y ha_gioco(giorgio,Y).
4..: ∀X ∀Y bambino(X) ∧ ama(X,Y)→ ¬ butta(X,Y).
Goal: ∃Y ¬ butta(giorgio,Y).

Goal Negato: ∀Y butta(giorgio,Y).

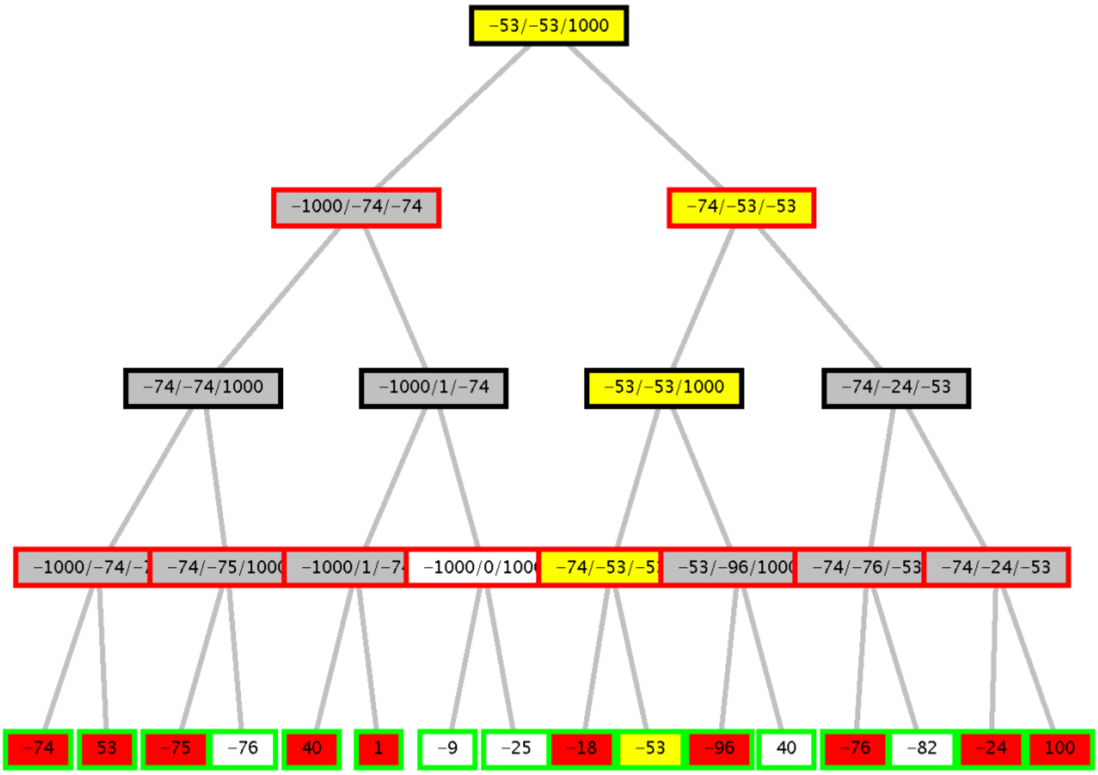
Tradotti in clausole:
1.: ¬bambino(X) ∨ ¬ ha_gioco(X,Y) ∨ ama(X,Y).
2.: bambino(giorgio)
3.: ha_gioco(giorgio,c1). skolem
4.: ¬ama(X,Y) ∨ ¬bambino(X) ∨ ¬ butta(X,Y).
GNeg.: butta(giorgio,X).

Refutazione:
5.: GNeg+4 : .: ¬ama(giorgio,Y) ∨ ¬bambino(giorgio)
6.: 5+2.: ¬ama(giorgio,Y)
7.: 6 + 1: ¬bambino(giorgio) ∨ ¬ ha_gioco(giorgio,Y)
8.: 7 + 2: ¬ ha_gioco(giorgio,Y)
9.: 8 + 3: contraddizione logica!

Esercizio 2
Min-max:



Alfa-beta:



In rosso i nodi espansi, in giallo la strada trovata, i nodi in bianco non sono esplorati per effetto dei tagli alfa-beta. Archi tagliati: a18, a10, a26, a28. La mossa selezionata sarà quella a destra dalla radice.

Esercizio 3

Variabili: Ta, Tb, Tc, Td

Valori del dominio: interi da 0 .. 10.

Vincoli:

binari:

$Ta + 2 \leq Tb$

$Ta + 2 \leq Tc$

$Tb + 3 \leq Td$

Unari:

$Tc + 5 \leq 9$

$Td + 4 \leq 9$

Node consistency: $Tc + 5 \leq 9$, $Td + 4 \leq 9$

Ta: 0..10, Tb: 0..10; Tc: 0..4; Td: 0..5;

Arc-consistency:

Step	Arc	Ta	Tb	Tc	Td
Start		0..10	0..10	0..4	0..5
1	(Tb,Ta)		2..10		
2	(Td,Tb)				5
5	(Tb,Td)		2		
6	(Ta,Tb)	0			
7	(Tc,Ta)			2..4	
8	(Ta,Tc)				

Risultato Ta=0; Tb=2; Tc=2..4; Td=5;

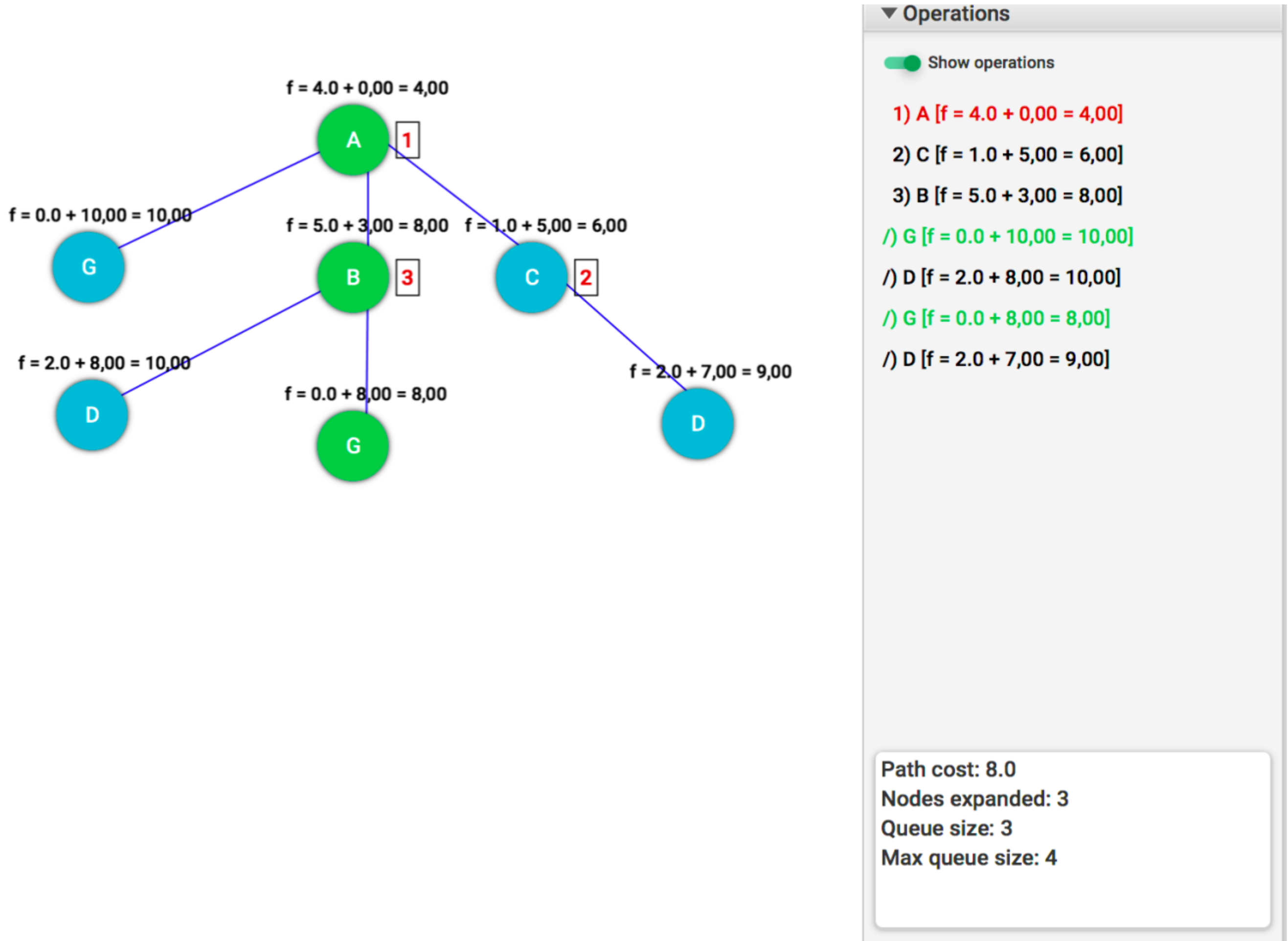
Esercizio 4

```
negList([],[]):-!.
negList([L|Tail], [X1|Tail1]) :- neg(L,X1),negList(Tail,Tail1).

neg([T|_],T) :- T < 0, !.
neg([_|C],X) :- neg(C,X).
```

Esercizio 5

L'euristica è ammissibile, e A* trova quindi la soluzione ottimale. I nodi espansi sono 3 , nell'ordine ACB, e il costo della strada pe arrivare alla soluzione 8.



Esercizio 6

Per il predicato Prolog `not` si veda il materiale del corso.
Per le due query nell'esercizio le risposte sono le seguenti:

```
?- c(Y,3)        Y = 1
?- c(Y,4)        Y = 2
```