FONDAMENTI DI INTELLIGENZA ARTIFICIALE

10 giugno 2021 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

NOTA: Consegnare la soluzione tramite un singolo file denominato con il proprio CognomeNome - Ad esempio: RossiMario

Esercizio 1 (6 punti)

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

1. Se un qualunque alunno di una classe ha il COVID-19, la classe entra in quarantena.
2. Chiunque è positivo al test COVID-19 molecolare o (**OR NON ESCLUSIVO**) è positivo al test COVID-19 antigenico allora ha il COVID-19.
3. Giovanni è un alunno della classe primaA
4. Marco è un alunno della classe primaB
5. Marco è risultato positivo al test COVID-19 molecolare.

Si utilizzi la risoluzione per dimostrare che esiste almeno un alunno di una classe che entra in quarantena.

Si usino i seguenti predicati:

```
alunnoClasse(Persona, Classe)
haCovid(Persona).
quarantena(Classe).
testPosMol(Persona).
testPosAntig(Persona).
```

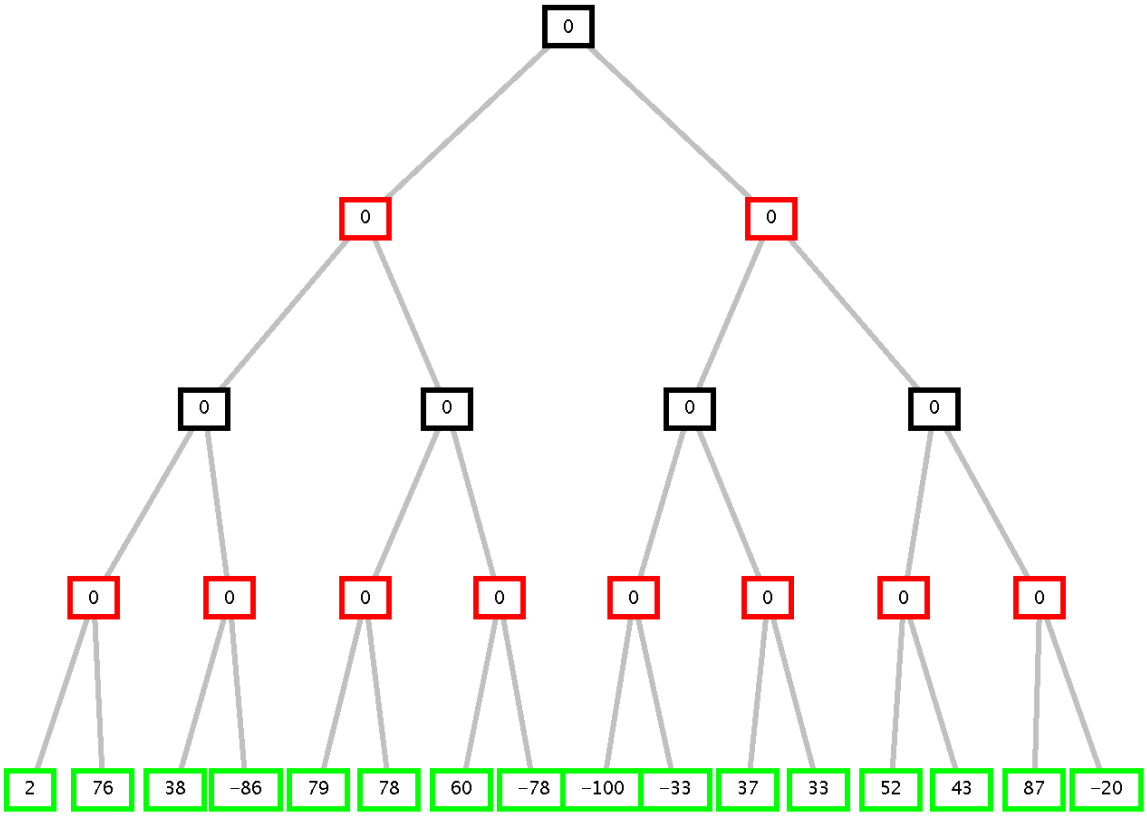
NOTA: Si riportano i simboli degli operatori e quantificatori in logica: ∀ ∃ ∧ ∨ ¬ →

Esercizio 2 (4 punti)

Si consideri il seguente albero di gioco in cui il primo giocatore è MAX.

a) 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 (nome dell'arco dal nodo radice).

b) 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 2) sarà denominato a15, mentre l'ultimo arco che connette il nodo foglia più a destra (valore -20) a30.



Esercizio 3 (6 punti)

Si consideri il seguente CSP che lega le variabili A, B, C, D:

A::[4, 5, 6, 7, 8, 9, 10] A>=B-7

B::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] C>=B-5
C::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] A=D+5
D::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Durante la ricerca, fino alla prima soluzione, si applichi il Forward Checking dopo ogni passo di labeling. Nella scelta della prossima variabile da istanziare applicare l'euristica **Minimum Remaing Value** (poi, a parità di cardinalità di dominio, scegliere in base all'ordine alfabetico dei nomi delle variabili) e per il labeling si considerino i valori di dominio in ordine crescente, partendo dal più piccolo. Si mostri come si raggiunge la soluzione indicando ad ogni passo i valori della variabili via via istanziate (labeling), i domini delle variabili non ancora istanziate (eventualmente ridotti a causa del forward checking) e l' eventuale presenza di backtracking.

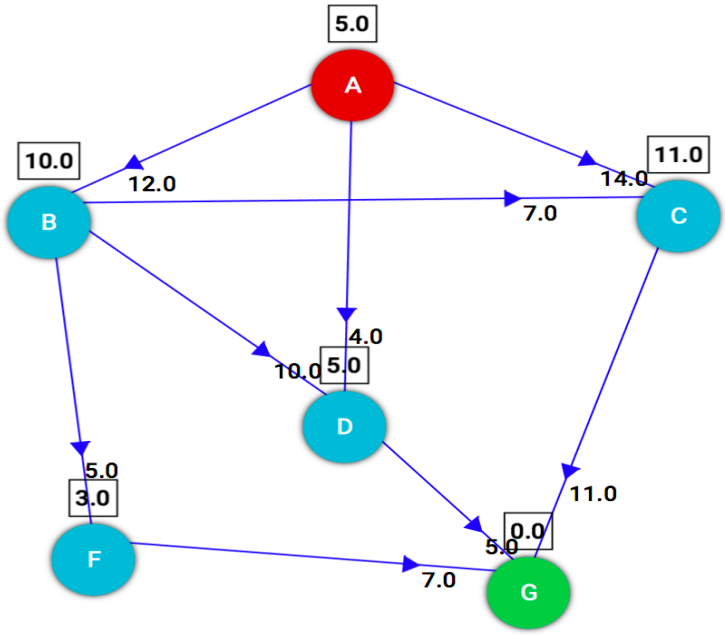
Esercizio 4 (4 punti)

Si realizzi un predicato Prolog **`checkMax(List,N)`** che data una lista **`List`** e un numero intero **`N`**, abbia successo se **`N`** è strettamente maggiore di tutti gli elementi della lista **`List`**. Se **`List`** è vuota, il predicato ha sempre successo. Di seguito si riporta qualche esempio di esecuzione Prolog:

```
?- checkMax(7,[4,2,5,6]).
yes
?- checkMax(4,[]).
yes
?- checkMax(4,[3,10]).
no
```

Esercizio 5 (7 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 **Depth-first** su alberi (che non tiene traccia dei nodi già visitati) e si indichino i nodi espansi nell'ordine di espansione. In caso di non-determinismo (più figli di un nodo), si scelga il nodo da espandere in base all'ordine alfabetico del nome. Non si consideri l'euristica h(n) indicata nel quadrato a fianco di ogni nodo in figura e neppure il costo degli archi, ma solo la profondità.
b) Si applichi poi la ricerca **A*** su alberi (che non tiene traccia dei nodi già visitati) e si indichino i nodi espansi nell'ordine di espansione. In caso di non-determinismo si scelga il nodo da espandere in base all'ordine alfabetico.
c) h(n) è ammissibile? Si motivi la risposta.
d) Si cosiderino i costi degli archi e si commentino i costi delle soluzioni delle due strategie (depth-first e A*) in termini di ottimalità .

Esercizio 6 (5 punti)

Si descriva sinteticamente l'unificazione, dove è utilizzata in Prolog, cos'è l'occur-check e cosa implichi il suo non utilizzo in Prolog.

10 giugno 2021 - Soluzioni

Esercizio 1

1. ∀Classe, ∀Persona alunnoClasse(Persona, Classe) ∧ haCovid(Persona) ⟶ quarantena(Classe).
2. ∀Persona (testPosMol(Persona) ∨ testPosAntig(Persona)) ⟶ haCovid(Persona).
3. alunnoClasse(giovanni, primaA).
4. alunnoClasse(marco, primaB).
5. testPosMol(marco).

Query: ∃X ∃Y alunnoClasse(X,Y) ∧ quarantena(Y).

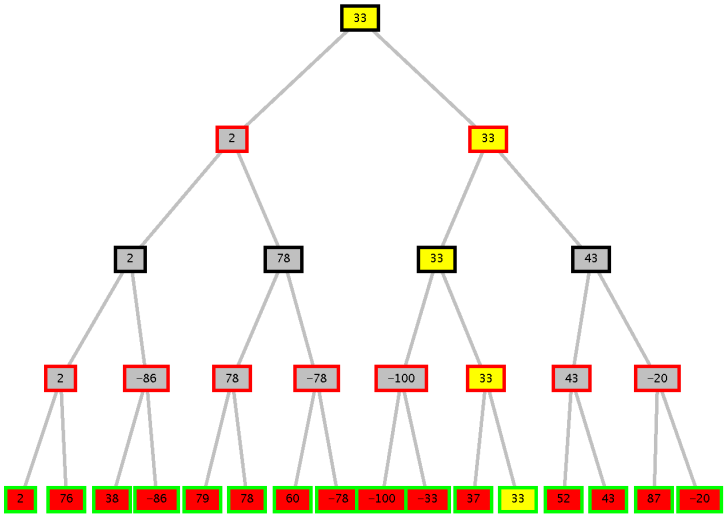
Trasformazione in clausole:

C1: ¬ alunnoClasse(Persona, Classe) ∨ ¬haCovid(Persona) ∨ quarantena(Classe).
C2a: ¬testPosMol(Persona) ∨ haCovid(Persona).
C2b: ¬testPosiAntig(Persona) ∨ haCovid(Persona).
C3: alunnoClasse(giovanni, primaA)
C4: alunnoClasse(marco, primaB)
C5: testPosMol(marco)
QNeg: ¬quarantena(Y) ∨ ¬alunnoClasse(X,Y).

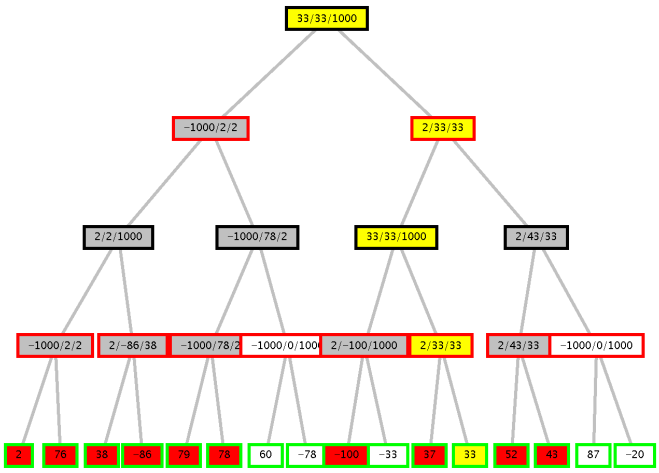
Risoluzione:

C6: QNeg+C1: ¬alunnoClasse(Persona, Classe) ∨ ¬haCovid(Alunno)
C7: C6+C4: ¬haCovid(marco)
C8: C7+C2a: ¬testPosMol(marco)
C9: C8 + C5 contraddizione!!

Esercizio 2. Min-max: strada in giallo (arco a2) – valore nodo radice 33.



Alfa-beta: In rosso i nodi espansi, in giallo la strada trovata, i nodi in bianco non sono esplorati per effetto dei tagli.



Archi tagliati a10, a14, a24.
Scelta per il ramo a sinistra, valore propagato 33.

Esercizio 3

Con euristica MRV:

A::[4, 5, 6, 7, 8, 9, 10]
B::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
C::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
D::[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
A>=B-7
C>=B-5
A=D+5

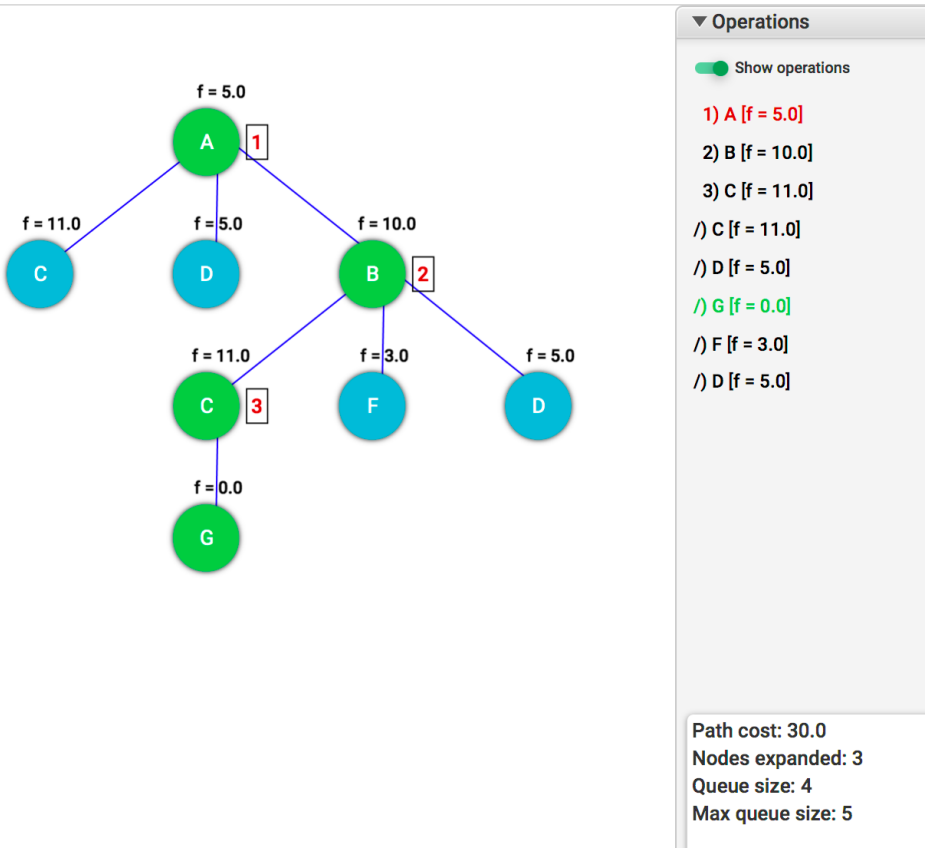
	A	B	C	D
Labeling	**A=4**	[1...10]	[1...10]	[1...10]
FC - Backtracking	A=4	[1...10]	[1...10]	**Fail**
Labeling	**A=5**	[1...10]	[1...10]	[1...10]
FC - Backtracking	A=5	[1...10]	[1...10]	**Fail**
Labeling	**A=6**	[1...10]	[1...10]	[1...10]
FC	A=6	[1...10]	[1...10]	**[1]**
Labeling	A=6	[1...10]	[1...10]	**D=1**
FC	A=6	[1...10]	[1...10]	D=1
Labeling	A=6	**B=1**	[1...10]	D=1
FC	A=6	B=1	[1...10]	D=1
Labeling	A=6	B=1	**C=1**	D=1

Esercizio 4

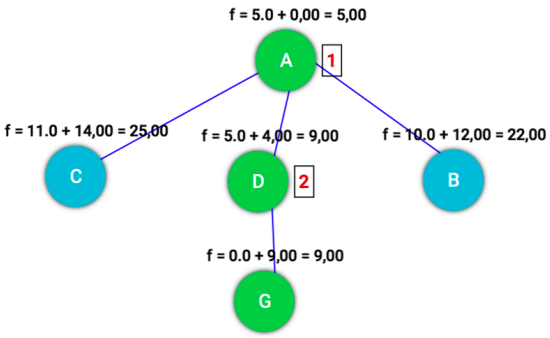
```
checkMax([], _).
checkMax([Hs|Ss],N):- N > Hs, checkMax(Ss,N).
```

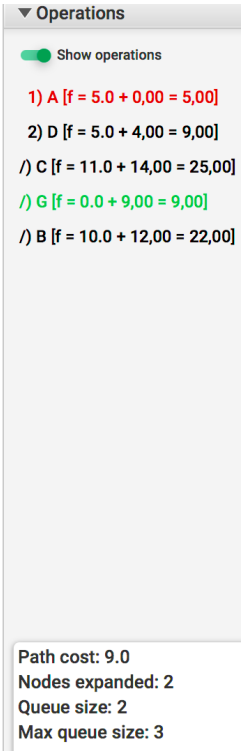
Esercizio 5

Con ricerca Depth-first e nodi espansi ABC si trova la soluzione (non ottimale): ABCG con costo 30.



Con ricerca A* e nodi espansi AD si trova la soluzione trovata ADG è a costo inferiore 9 (ottimale perché l'euristica è ammissibile).





Esercizio 6
Si veda il materiale del corso.