t

FONDAMENTI DI INTELLIGENZA ARTIFICIALE

27 Gennaio 2022 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

NOTA: Consegnare la soluzione tramite un singolo file (word, pdf oppure txt)

Esercizio 1 (6 punti)

Si formalizzino le seguenti frasi in logica dei predicati:

1. I giocatori di scacchi sono persone intelligenti.
2. Qualunque giocatore di scacchi che vince si diverte.
3. Chiunque si diverte ed è intelligente, non è depresso.
4. Esiste un giocatore di scacchi che vince.

usando i seguenti predicati: **giocas(X)** (X è un giocatore di scacchi), **intelligente(X)** (X è una persona intelligente), **vince(X)** (X vince), **diverte(X)** (X si diverte), **depresso(X)** (X è depresso).

Le si trasformi in clausole e si usi poi il principio di risoluzione per dimostrare che c'è una persona intelligente che non è depressa.

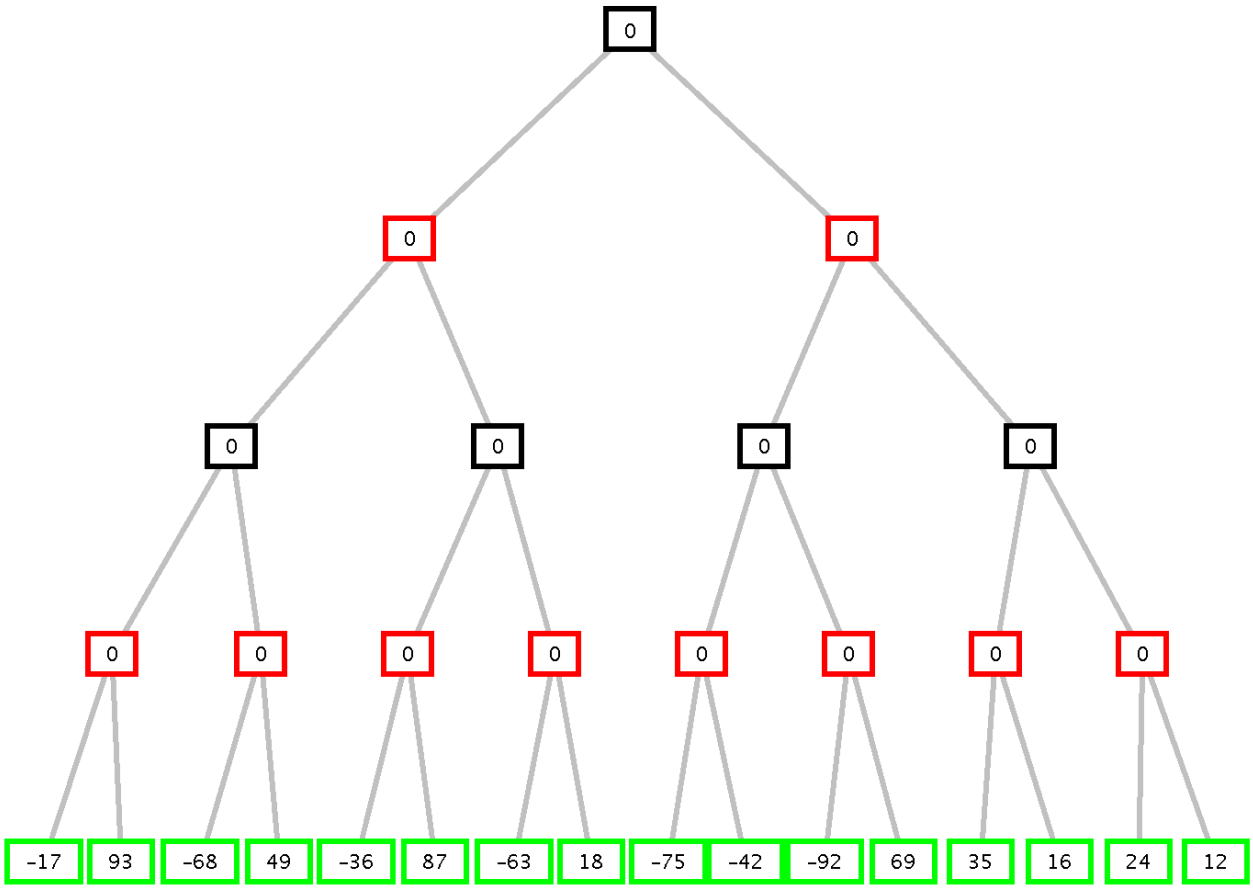
NOTA: Si riportano i simboli degli operatori e quantificatori in logica: $\forall \exists \ \wedge \vee \ \neg \rightarrow$

Esercizio 2 (6 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 (arco a1, o a2).

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 -17) sarà denominato a15, mentre l'ultimo arco che connette il nodo foglia più a destra (valore 12) a30.



Esercizio 3 (5 punti)

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

A::[4, 5, 6, 7, 8, 9, 10]	A>=B*2
B::[0, 1, 2, 3, 4, 5, 6]	C<=B-1
C::[0, 1, 2, 3, 4, 5, 6]	A=D-1
D::[0, 1, 2, 3, 4, 5, 6]	

Si applichi, durante la ricerca fino alla prima soluzione, il Forward Checking dopo ogni passo di labeling, considerando, nella scelta della prossima variabile da istanziare l'euristica **Minimum Remaing Value** (poi, a parità di cardinalità di dominio, scegliere in base all'ordine alfabetico dei nomi delle variabili). Nel labeling , per il valore da assegnare alla variabile, si considerino i valori di dominio in ordine crescente, partendo dal più piccolo.

Esercizio 4 (5 punti)

Si realizzi un predicato Prolog **lista_coppie(L1, L2, L3)** che date due liste **L1** e **L2** di interi di uguale lunghezza deve restituire una nuova lista **L3** contenente delle liste di due elementi **[el1,el2]** che appartengono a **L1** o **L2**, compaiono nelle stessa posizione nelle due liste e in cui il primo elemento **el1** è minore o uguale del secondo **el2**. Si tenga conto anche del possibile caso in cui **L1** e **L2** siano liste vuote: in tal caso **L3** sarà anche essa vuota.

Esempi:

```
?- lista_coppie([3,7,0,5],[4,6,0,3],L3).
```

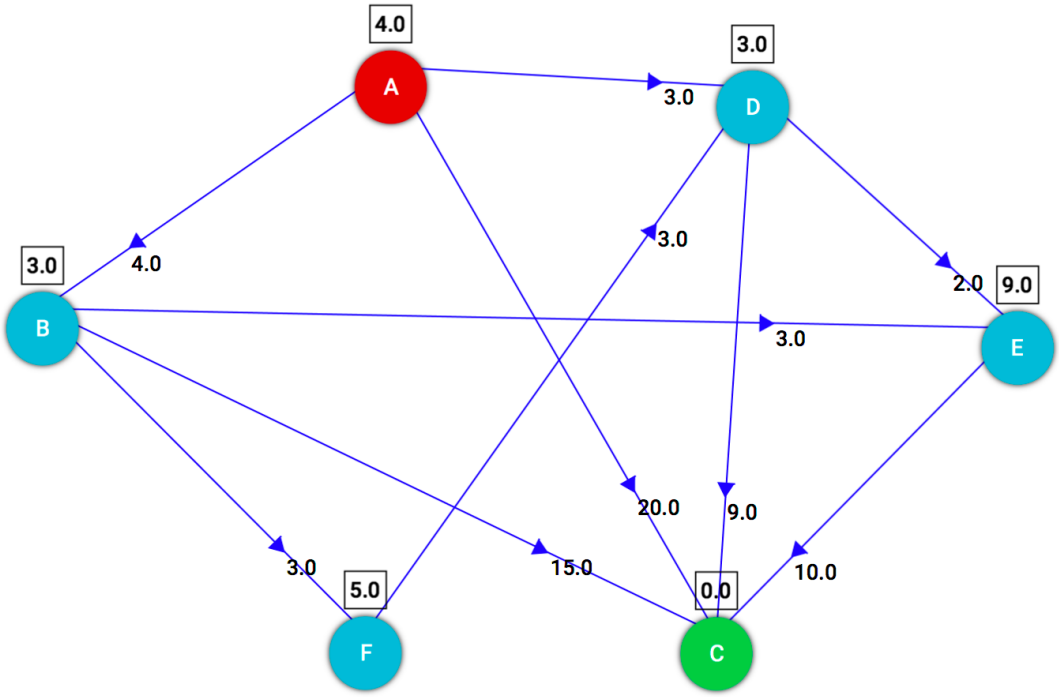
L3 = [[3, 4], [6, 7], [0, 0], [3, 5]]

```
?- lista_coppie([],[],L3).
```

L3 = []

Esercizio 5 (6 punti)

Si consideri il seguente grafo, dove A è il nodo iniziale e C 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 C:



Si applichi la ricerca **A*** su alberi (non tenendo quindi traccia dei nodi già visitati) indicando:

- i nodi espansi nell'ordine di espansione;
- i nodi sulla strada della soluzione e il costo della soluzione;
- se è garantita o meno l'ottimalità.

Esercizio 6 (4 punti)

Si spieghi brevemente il predicato predefinito Prolog "cut", le sue caratteristiche e si riporti un esempio che ne mostri l'utilizzo.

27 Gennaio 2022 - Soluzioni

Esercizio 1

1. ∀X giocas(X) ⟶ intelligente(X)
2. ∀X giocas(X) ∧ vince(X) ⟶ diverte(X)
3. ∀X diverte(X) ∧ intelligente(X) ⟶ ¬depresso(X)
4. ∃X giocas(X) ∧ vince(X)

Goal: ∃X intelligente(X) ∧ ¬depresso(X)

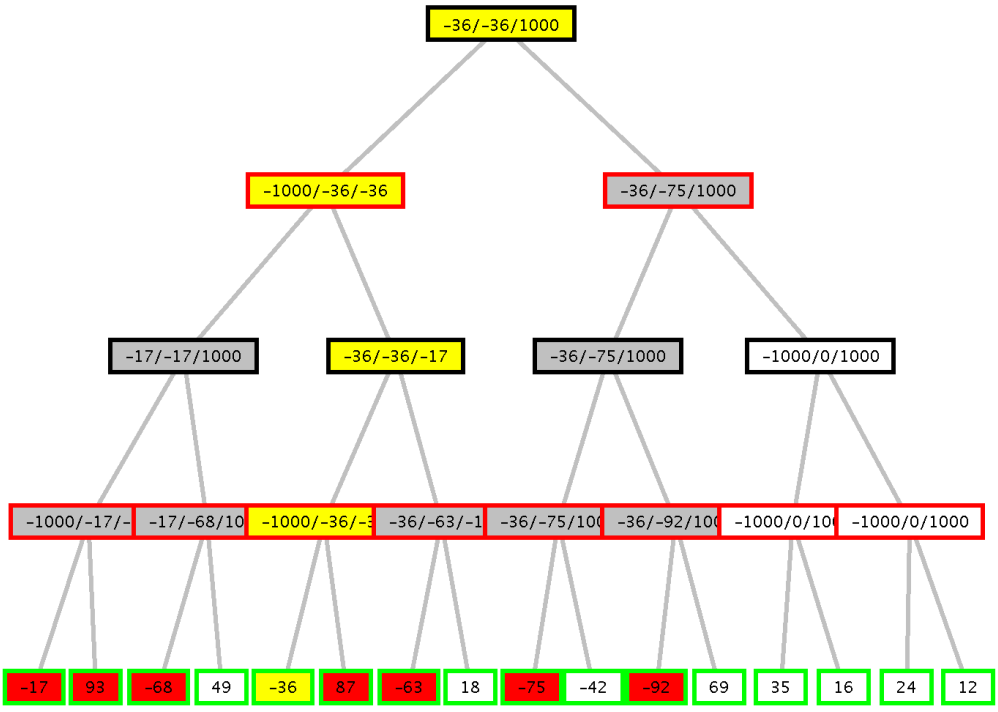
Clausole:

1. ¬ giocas(X) ∨ intelligente(X)
2. ¬ giocas(X) ∨ ¬ vince(X) ∨ diverte(X)
3. ¬ diverte (X) ∨ ¬intelligente(X) ∨ ¬depresso(X).
4. giocas(c1). (c1 costante di Skolem)
5. vince(c1).
6. G(Neg): ¬ intelligente(X) ∨ depresso(X).

Risoluzione:
7: 6 + 1: ¬ giocas(X) ∨ depresso(X)
8: 7 + 4: depresso(c1).
9: 8 + 3: ¬ diverte (c1) ∨ ¬intelligente(c1)
10: 9 + 2: ¬ giocas(c1) ∨ ¬ vince(c1) ∨ ¬intelligente(c1)
11: 10 + 1: ¬ giocas(c1) ∨ ¬ vince(c1)
12: 11 + 4: ¬ vince(c1)
13: 12 + 5: Clausola vuota Contraddizione!!

Oppure (qualche passaggio in meno):
7: 6+3: ¬ diverte(X) ∨ ¬intelligente(X)
8: 7+2: ¬ giocas(X) ∨ ¬ vince(X) ∨ ¬ intelligente(X)
9: 8+1: ¬ giocas(X) ∨ ¬ vince(X)
10: 9+4: ¬ vince(c1)
11: 10+5: clausola vuota

Esercizio 2
Min-max: strada in giallo, ma senza i tagli – valore nodo radice -36, ramo a1.
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 a6, a18, a22, a24, a26. In totale cinque tagli.
Scelta per il ramo a1, valore propagato -36.

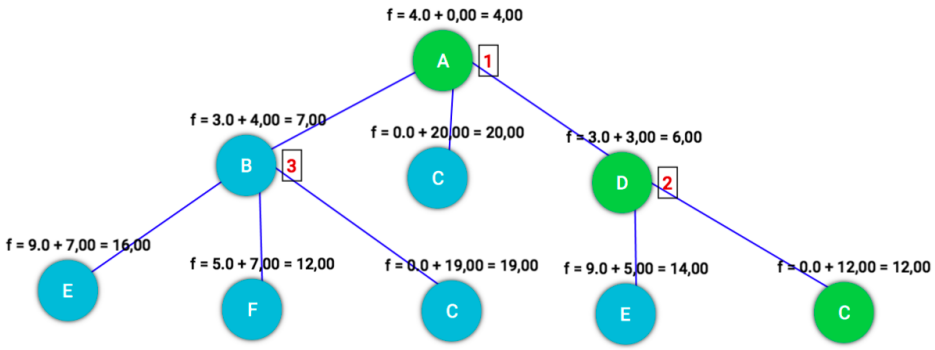
Esercizio 3
Con euristica MRV:

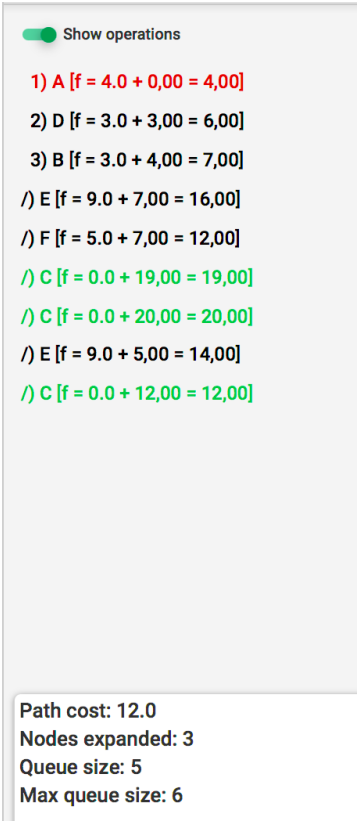
	A	B	C	D
Labeling	**A=4**	[0...6]	[0...6]	[0...6]
FC		[0...2]	[0...6]	**[5]**
Labeling	A=4	[0...2]	[0...6]	**D=5**
FC		[0...2]	[0...6]	
Labeling	A=4	**B=0**	[0...6]	D=5
FC			**[]**	
Backtracking				
Labeling	A=4	**B=1**	[0...6]	D=5
FC			**[0]**	
Labeling	A=4	B=1	**C=0**	D=5
Soluzione:	**A=4**	**B=1**	**C=0**	**D=5**

Esercizio 4

```
lista_coppie([],[],[]):-!.
lista_coppie([El1|Y1],[El2|Y2],[[El1,El2]|Y3]):-
      El1 =< El2, !, lista_coppie(Y1,Y2,Y3).
lista_coppie([El1|Y1],[El2|Y2],[[El2,El1]|Y3]):- lista_coppie(Y1,Y2,Y3).
```

Esercizio 5
Con A*, i nodi espansi sono ADBC, la soluzione trovata ADC è ottimale perché l'euristica è ammissibile (si verifichi per ogni nodo) e il costo della soluzione trovata è 12:





Esercizio 6
Si vedano le slide del corso.