

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

CognomeNomeDataAI

Ad esempio:

RossiMario20210107AI

Esercizio 1 (6 punti)

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

1. Mario è uno sciatore.
2. Tutti gli sciatori sono atleti.
3. Giorgio è uno slalomista.
4. Gli slalomisti sono sciatori.
5. Se un atleta teme un altro atleta, allora il secondo è più forte del primo.
6. Mario teme Giorgio.

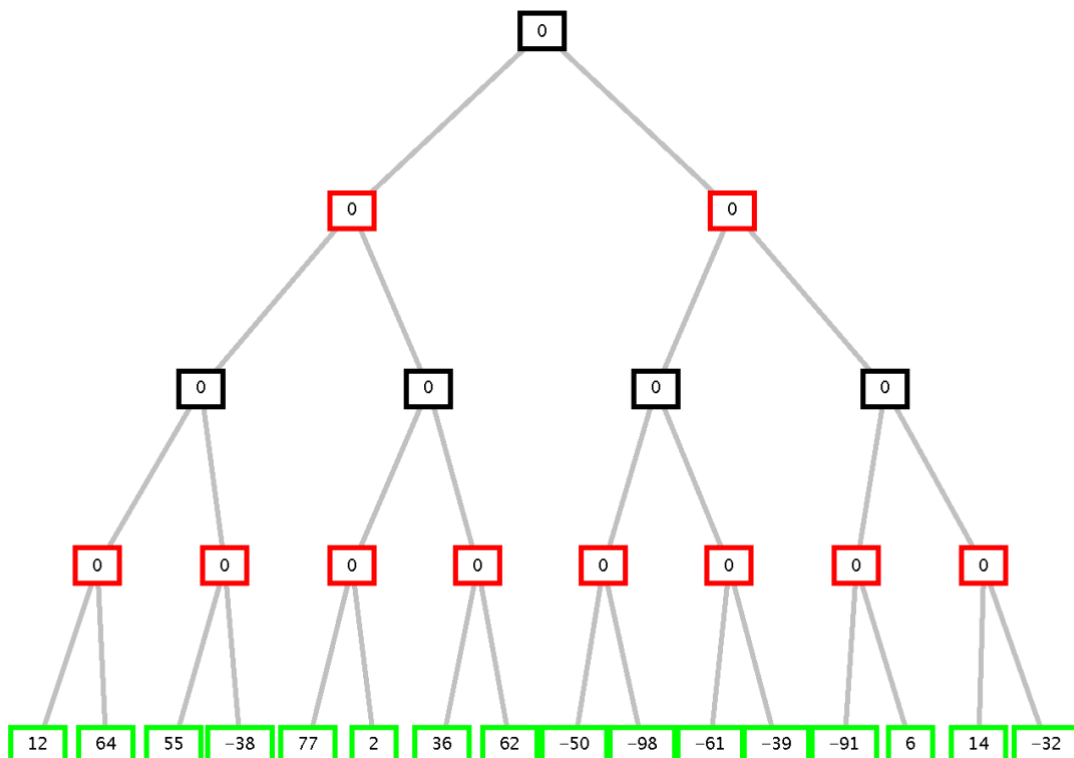
Si dimostri, per refutazione, tramite l'applicazione della risoluzione, che *Giorgio è più forte di Mario*.

Si usi il seguente vocabolario:

Predicati: **sciatore(X)**, **atleta(X)**, **slalomista(X)**, **teme(X,Y)** con significato “X teme Y”, **forte(X,Y)** con significato “X è più forte di Y”, Costanti: **mario**, **giorgio**.

Esercizio 2 (4 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 12) sarà denominato a15, mentre l'ultimo arco che connette il nodo foglia più a destra (valore -32) a30.



Esercizio 3 (6 punti)

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

A::[5, 6, 7, 8, 9]

B::[5, 6, 7, 8, 9]

C::[1, 2, 3, 4, 5, 6, 7, 8, 9]

D::[5, 6, 7, 8, 9]

$C \leq 20 - D$

$A \neq B$

$C \neq D - 3$

Durante la ricerca, si applichi il forward checking dopo ogni passo di labeling, e ci si fermi alla prima soluzione. Nella scelta della prossima variabile da istanziare si applichi l'euristica Minimum Remaining Value (e a parità di cardinalità di dominio, si scelga in base all'ordine alfabetico dei nomi delle variabili). Nella scelta dei valori si preferiscano quelli minori. Si mostri come si raggiunge la soluzione indicando ad ogni passo i valori della variabili via via istanziate (labeling) e i domini delle variabili non ancora istanziate (eventualmente ridotti a causa del forward checking).

Esercizio 4 (5 punti)

Si scriva un predicato Prolog `nuovaLista(L, Start, End)` che, dati due valori interi `Start` ed `End`, restituisca una lista `L` contenente i valori interi a partire da `Start` fino a `End`. Se $Start > End$ dovrà restituire la lista vuota.

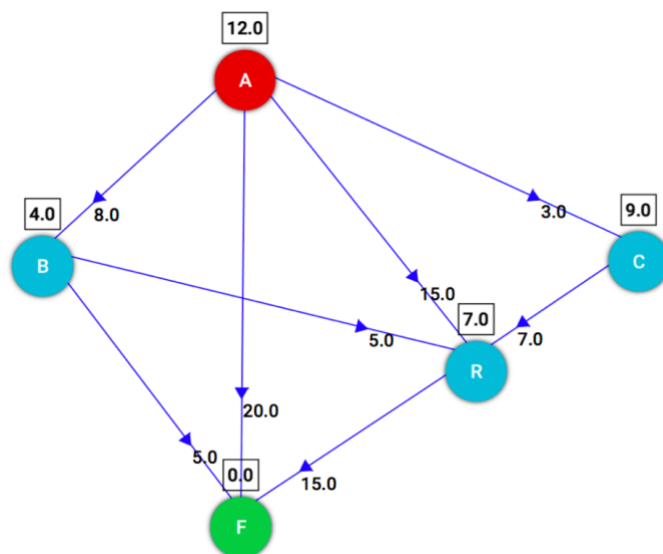
Ad esempio il goal:

?- `nuovaLista(L, 3, 6)` produrrà la risposta: `L=[3, 4, 5, 6]`.

?- `nuovaLista(L, 6, 3)` produrrà la risposta: `L=[]`.

Esercizio 5 (7 punti)

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



- Si applichi la ricerca **Best-first**, e si scrivano i nomi dei 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.
- Si applichi poi la ricerca **A*** e si scrivano i nomi dei 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.
- Qual è il costo di cammino trovato per arrivare al goal F a partire dal nodo iniziale A con le due strategie? (Best-first e A*?)
- La soluzione trovata nel caso A* è ottimale? (motivare la risposta).

Esercizio 6 (4 punti)

Si introduca brevemente il metodo di ricerca locale di Hill-climbing, sottolineandone le caratteristiche. Se ne descriva poi l'algoritmo in pseudo-codice.

Esercizio 1

Formule in FOL:

- $\text{sciatore}(\text{mario})$
- $\forall X (\text{sciatore}(X) \rightarrow \text{atleta}(X))$
- $\text{slalomista}(\text{giorgio})$
- $\forall X (\text{slalomista}(X) \rightarrow \text{sciatore}(X))$
- $\forall X \forall Y (\text{atleta}(X) \wedge \text{atleta}(Y) \wedge \text{teme}(X,Y) \rightarrow \text{forte}(Y,X))$
- $\text{teme}(\text{mario}, \text{giorgio})$.

Goal: $\text{forte}(\text{giorgio}, \text{mario})$

Traduzione in clausole:

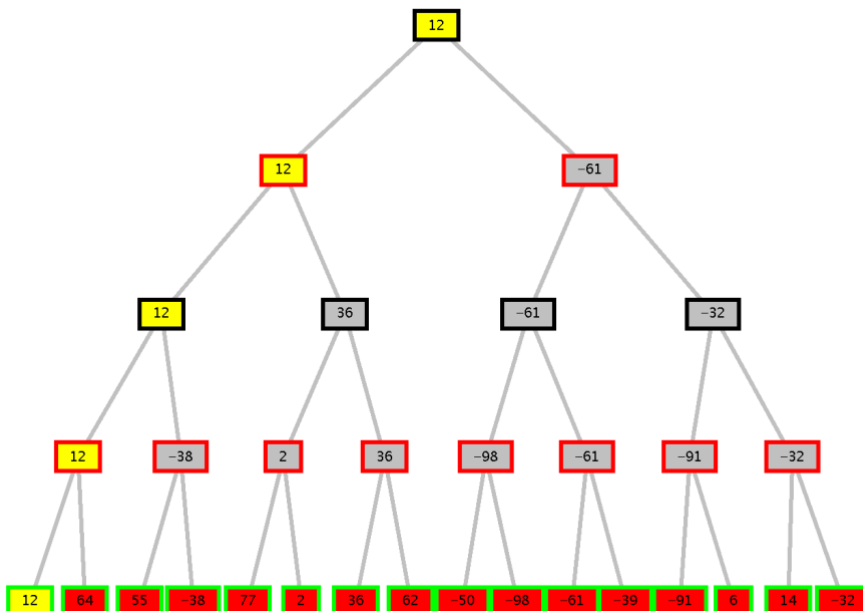
- C1: $\text{sciatore}(\text{mario})$
 C2: $\neg \text{sciatore}(X) \vee \text{atleta}(X)$
 C3: $\text{slalomista}(\text{giorgio})$
 C4: $\neg \text{slalomista}(X) \vee \text{sciatore}(X)$
 C5: $\neg \text{atleta}(X) \vee \neg \text{atleta}(Y) \vee \neg \text{teme}(X,Y) \vee \text{forte}(Y,X)$
 C6: $\text{teme}(\text{mario}, \text{giorgio})$.
 Gneg: $\neg \text{forte}(\text{giorgio}, \text{mario})$

Risoluzione:

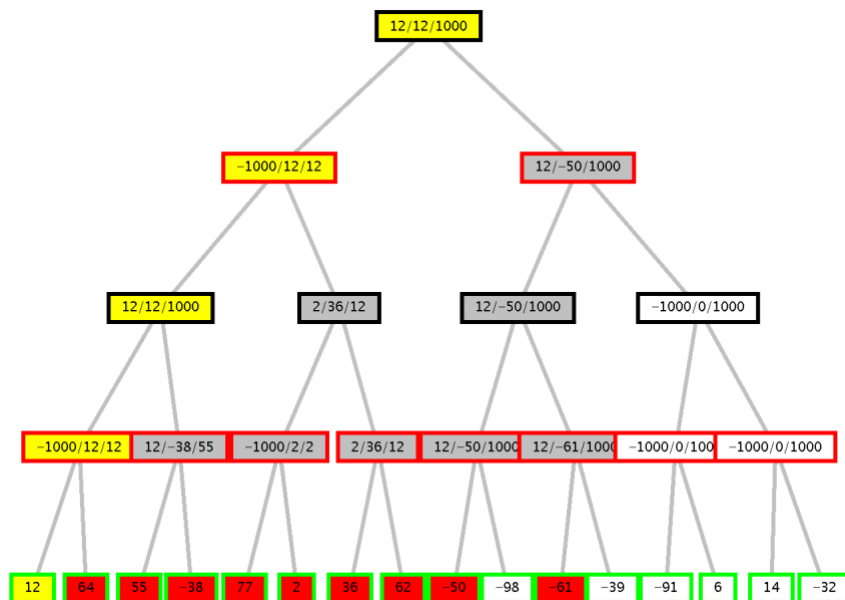
- C7 da C6+C5: $\neg \text{atleta}(\text{mario}) \vee \neg \text{atleta}(\text{giorgio}) \vee \text{forte}(\text{giorgio}, \text{mario})$
 C8 da C7+Gneg: $\neg \text{atleta}(\text{mario}) \vee \neg \text{atleta}(\text{giorgio})$
 C9 da C8+C2: $\neg \text{sciatore}(\text{mario}) \vee \neg \text{atleta}(\text{giorgio})$
 C10 da C9+C1: $\neg \text{atleta}(\text{giorgio})$
 C11 da C10+C2: $\neg \text{sciatore}(\text{giorgio})$
 C12 da C11+C4: $\neg \text{slalomista}(\text{giorgio})$
 Clausola vuota da C12+C3

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 a24, a26, a6.

Scelta per il ramo a sinistra, valore propagato 12.

Esercizio 3

	A	B	C	D
Labeling	A=5	[5...9]	[1...9]	[5...9]
FC	A=5	[6...9]	[1...9]	[5...9]
Labeling	A=5	B=6	[1...9]	[5...9]
FC	A=5	B=6	[1...9]	[5...9]
Labeling	A=5	B=6	[1...9]	D=5
FC	A=5	B=6	[1, 3, 4, 5, 6, 7, 8, 9]	D=5
Labeling	A=5	B=6	C=1	D=5

Esercizio 4

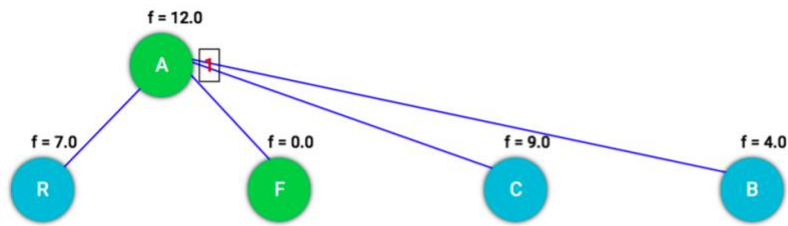
```

nuovaLista([],Start, End):- Start > End, !.
nuovaLista([Start|T], Start, End):- Start1 is Start + 1,
                                   nuovaLista(T,Start1,End).

```

Esercizio 5

Best-first trova per prima la strada rappresentata dall'arco diretto tra A e F:



Operations

Show operations

1) A [f = 12.0]

/) R [f = 7.0]

/) F [f = 0.0]

/) C [f = 9.0]

/) B [f = 4.0]

Path cost: 20.0

Nodes expanded: 1

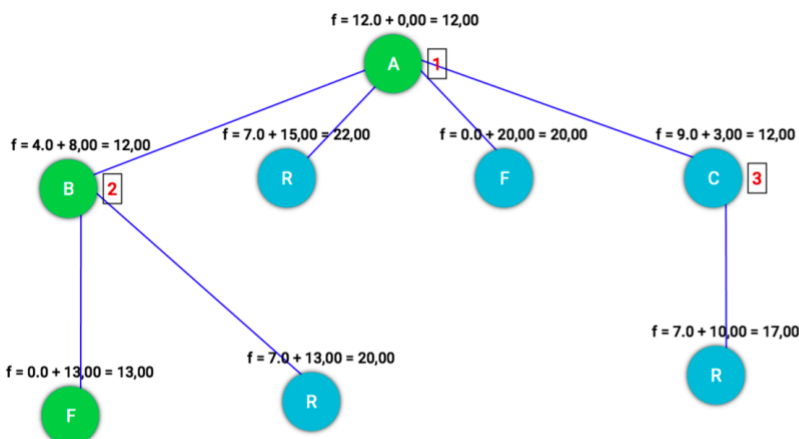
Queue size: 3

Max queue size: 4

Nodi Espansi: solo il nodo A (poi raggiunge il goal F): Non è la soluzione ottimale perché AF ha costo 20

Con ricerca A*, invece, data l'euristica che è ammissibile, si trova la soluzione ottimale: ABF che ha costo 13.

Nodi Espansi: ABC .



Operations

Show operations

1) A [f = 12.0 + 0,00 = 12,00]

2) B [f = 4.0 + 8,00 = 12,00]

3) C [f = 9.0 + 3,00 = 12,00]

/) R [f = 7.0 + 15,00 = 22,00]

/) F [f = 0.0 + 20,00 = 20,00]

/) R [f = 7.0 + 10,00 = 17,00]

/) R [f = 7.0 + 13,00 = 20,00]

/) F [f = 0.0 + 13,00 = 13,00]

Path cost: 13.0

Nodes expanded: 3

Queue size: 4

Max queue size: 5

Esercizio 6

Si vedano le slides del corso.