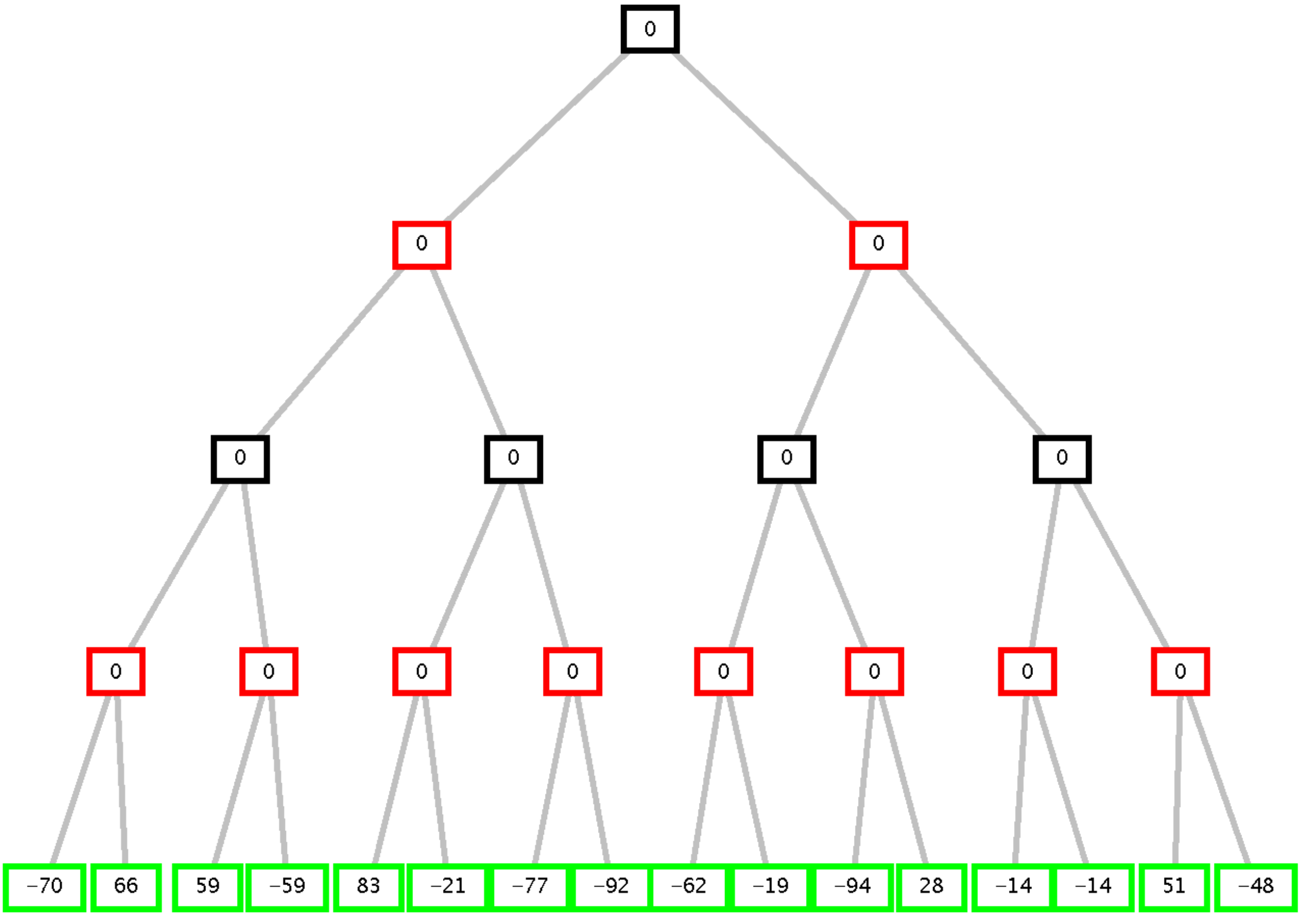
Fondamenti di Intelligenza Artificiale
11 Giugno 2020 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

NOTA: Consegnare la soluzione tramite un singolo file, che lo studente avrà cura di nominare come:
CognomeNomeDataAI
Ad esempio:
RossiMario20200611AI

Esercizio 1 (6 punti)
Si traducano in logica dei predicati le seguenti frasi:
1. Tutti gli atleti bravi ricevono un premio.
2. Per ogni squadra esiste almeno un atleta bravo.
3. Esiste almeno un squadra.

Si dimostri poi, tramite il principio di risoluzione, che esiste almeno un atleta che riceve un premio.
Si usino i seguenti predicati:
premio(X): "X riceve un premio" , **atleta(X)**: "X è un atleta", **bravo(X)**: "X è bravo", **squadra(X):** "X è una squadra".

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 -70) sarà denominato a15, mentre l'ultimo arco che connette il nodo foglia più a destra (valore -48) a30.



Esercizio 3 (6 punti)

Si considerino due variabili X e Y con domini interi Dx e Dy che vanno da 0 a 10 (estremi compresi) e soggette al vincolo:$X+4 \leq Y$

Si mostrino i passi dell'algoritmo di "arc-consistency" e si indichino i domini di X e Y (ridotti) ottenuti dopo la sua applicazione.

Ogni valore dei domini ridotti farà parte di almeno una possibile soluzione? Si motivi la risposta data.

Si mostri inoltre la prima soluzione trovata dallo standard backtracking a partire dai domini ridotti, considerando X prima di Y, e i valori nell'ordine crescente in cui appaiono nel dominio.

Esercizio 4 (5 punti)

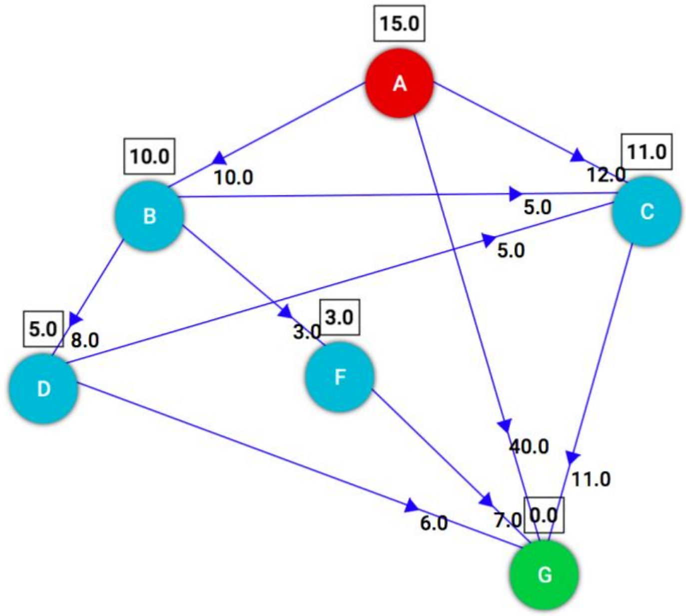
Si scriva una predicato PROLOG: `countList(N, P, N1)` che data una lista `P` di interi e un intero `N`, dia in uscita un numero `N1` che rappresenta quante volte compare l'elemento `N` in `P`.

Ad esempio:

```
?-countList(3, [3,3,2,3,1], X).
X = 3
?-countList(9, [3,3,2,3,1], X).
X = 0
```

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) L'euristica h(n) è ammissibile? (motivare la risposta)

Esercizio 6 (4 punti)

Dopo avere spiegato brevemente il comportamento del predicato predefinito `findall`, si indichi il risultato del seguente programma Prolog, quando interrogato col goal:

`?- test(W)` , motivando la risposta data:

```
p(0).
p(1).
p(2).
p(3).
p(4).
test(X) :- findall(Y,p(Y),L),sum(L,X).
sum([],0).
sum([X|Y],N):- sum(Y,N0), N is X + N0.
```

11 Giugno 2020 - Soluzioni

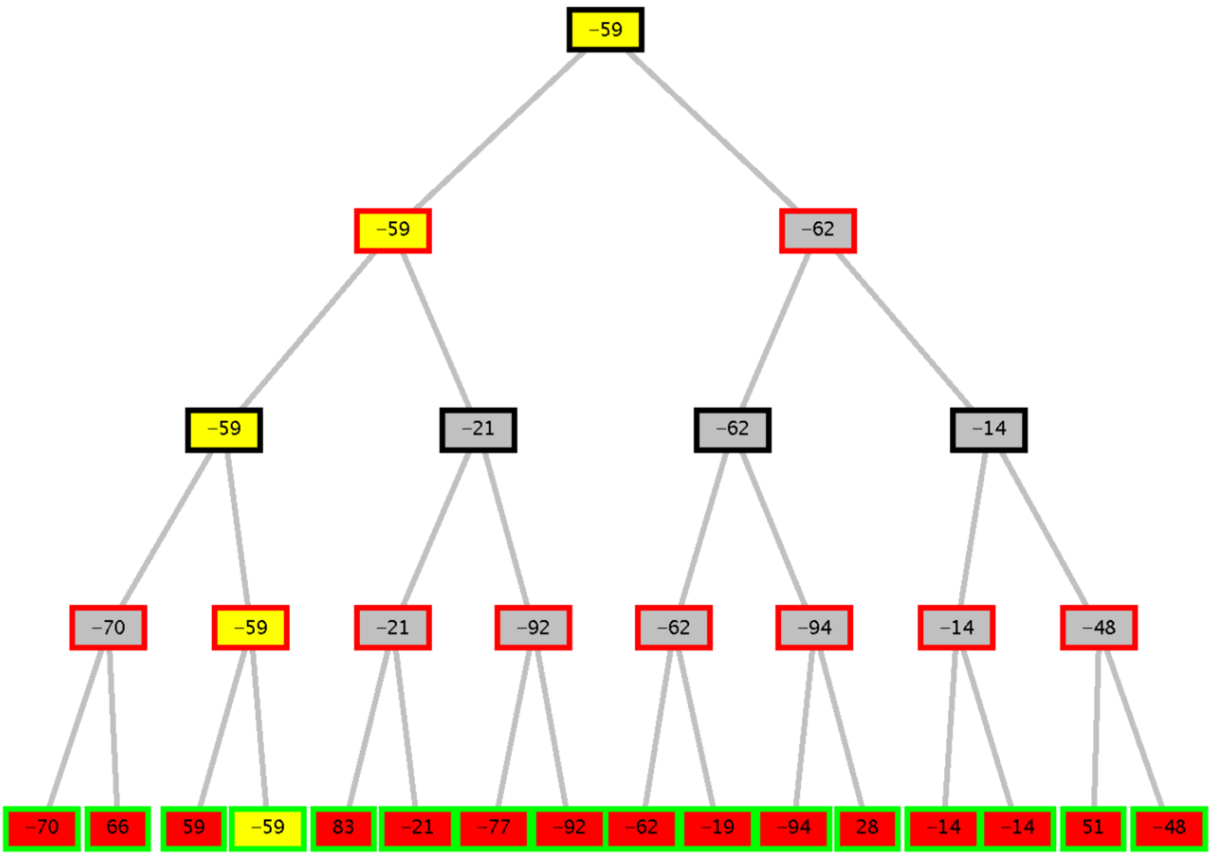
Esercizio 1

1. $\forall X \;\; atleta(X) \wedge bravo(X) \longrightarrow premio(X)$.
   $\neg\, atleta(X) \;\vee\; \neg\, bravo(X) \;\vee\; premio(X)$
2. $\forall X \;(squadra(X) \;\longrightarrow\; \exists Y\; \text{atleta}(Y) \;\wedge bravo(Y))$
   $\forall X\, \exists \text{Y}\; \neg squadra(X) \;\vee\; (\text{atleta}(Y) \;\wedge bravo(Y))$
   $(\neg squadra(X) \;\vee\; atleta(f(X))) \quad \wedge \quad (\neg squadra(X) \;\vee\; bravo(f(X)))$ funz. skolem f().
   2a. $\neg \text{squadra}(X) \vee\; atleta(\, f(X)\,)$.
   2b. $\neg squadra(X) \;\vee\; bravo(\, f(X)\,)$.
3. $\exists X\;\, squadra(X)$.
   $squadra(c)$ costante Skolem c

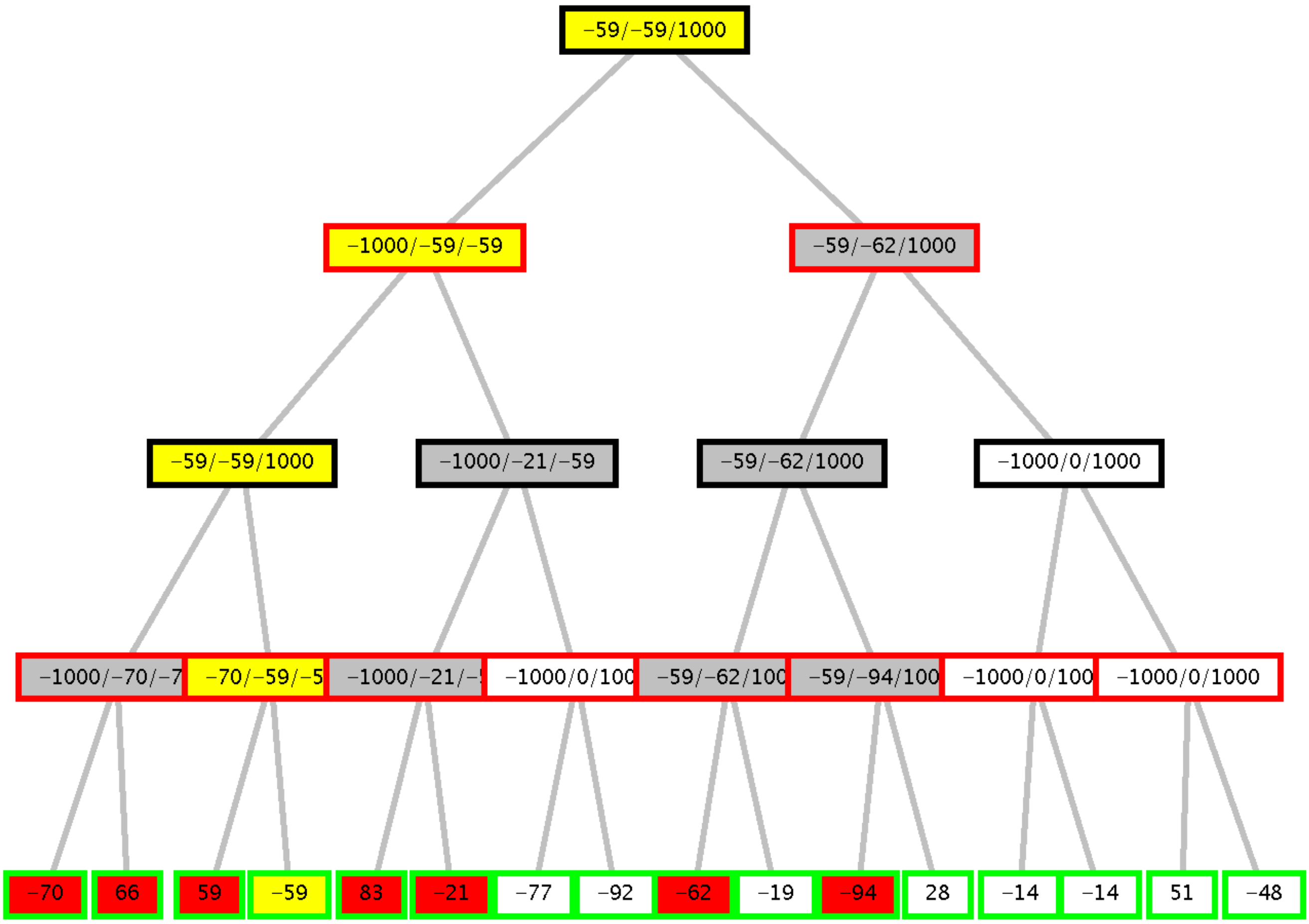
G. $\exists X\;\, \text{atleta}(\text{X}) \wedge\; \text{premio}(X)$
Gneg. $\neg premio(X) \quad \vee \quad \neg atleta(\text{X})$

<u>Risoluzione</u>
5: Gneg + 1: $\neg\, atleta(X) \;\vee\; \neg\, bravo(X)$
6: 5 + 2b: $\neg squadra(X1) \vee \neg\, atleta(\text{f}(\text{X1}))$
7: 6+2a: $\neg squadra(X1)$
8: 7+3: [] contraddizione !!!

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.
In particolare saranno tagliati gli archi **a10**, **a24**, **a26** e **a6**.

Esercizio 3
Dx = 0..10
Dy = 0..10
$X+4 \leq Y$.
L'arco arc(X,Y) non è arc-consistent.
Si consideri l'applicazione dell'algoritmo di arc-consistency partendo dall'arco con nodo X verso nodo Y:

- per i valori di Dx che vanno da 0..6 esiste almeno un valore in Dy per cui il vincolo è soddisfatto;
- per i valori da 7..10 non esiste alcun valore in Dy che soddisfi il vincolo per cui vengono rimossi da Dx che ora è ridotto a 0..6.

Si consideri ora l'arco che va da Y a X, e il suo dominio Dy:

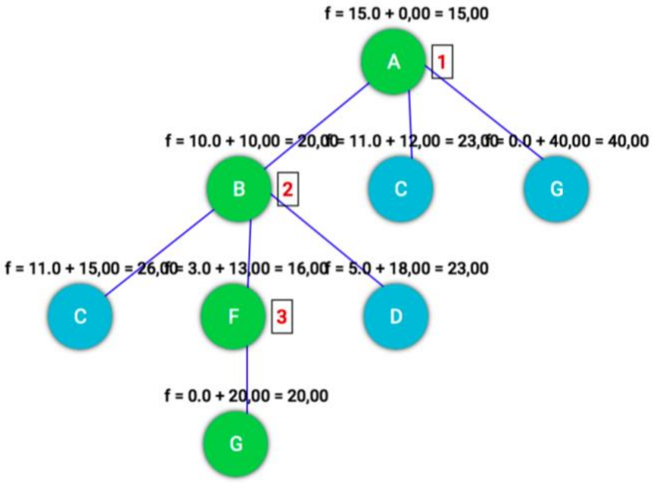
- per i valori di Dy che vanno da 4..10 esiste almeno un valore in Dx per cui il vincolo è soddisfatto;
- per i valori da 0..3 non esiste alcun valore in Dx che soddisfi il vincolo per cui vengono rimossi da Dy che ora è ridotto a 4..10.

Essendo stato ridotto il dominio si ri-verifica la consistenza del primo arco che va da X a Y che risulta comunque soddisfatto senza ulteriore riduzione dei domini.
I domini finali, dopo l'applicazione dell'arc-consistency saranno quindi:
Dx = 0..6
Dy = 4..10.
Ogni valore farà parte di una soluzione (che va trovata ovviamente con un algoritmo di ricerca e labeling), in quanto abbiamo una rete con consistenza di grado 2 e 2 soli nodi.
In particolare la prima soluzione, applicando lo standard backtracking a partire dai domini ridotti Dx = 0..6, Dy = 4..10, e considerando X prima di Y e i valori nell'ordine in cui appaiono nel dominio sarà:
labelling X=0, Y= 4.
Nessun backtracking.

Esercizio 4

```
countList(_, [] ,0) :-!.
countList(H, [H|M] ,N) :-!, countList(H, M ,N1), N is N1 +1.
countList(H, [_|L] ,N):- countList(H,L,N).
```

Esercizio 5



a) Nodi espansi: ABF (G)
b) Costo del cammino: 20
c) La stima euristica è ammissibile perchè ottimista per ogni nodo (h'(n) <=h(n)).

Esercizio 6

Per il comportamento del predicato PROLOG predefinito findall si consulti il materiale didattico.
Risultato yes W = 10.