**FONDAMENTI DI INTELLIGENZA ARTIFICIALE**

**28 Gennaio 2016 – Tempo a disposizione: 2 h – Risultato: 32/32 punti**

## Esercizio 1 (6 punti)

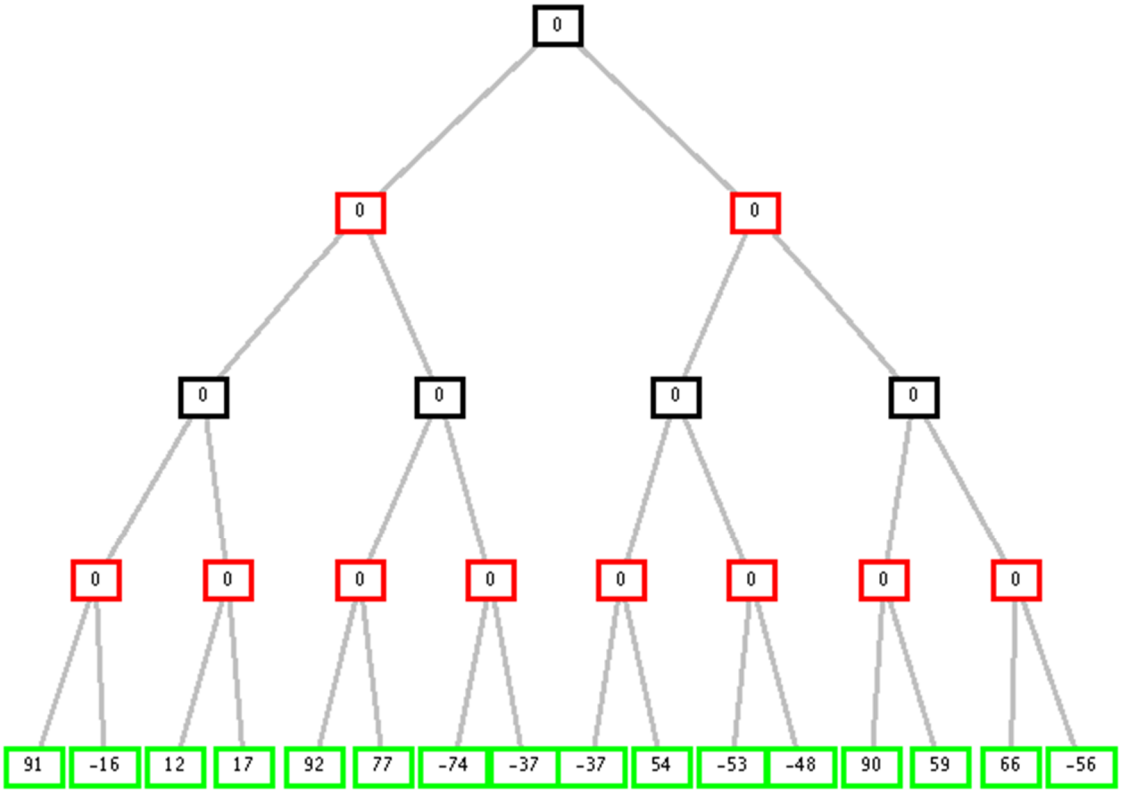
Date le seguenti frasi in linguaggio naturale:

1. Anna ama gli stessi dolci che ama Luca
2. Luca ama gli stessi dolci che ama Anna (*si noti che 1) e 2) sono l'una il viceversa dell'altra*)
3. Chi ama la torta al cioccolato , ingrassa
4. La torta al cioccolato è un dolce.
5. Anna non ingrassa

Dimostrare, tramite il principio di risoluzione, che ci sono dolci che Luca non ama.

## Esercizio 2 (5 punti)

Si consideri il seguente albero di gioco in cui la valutazione dei nodi terminali è dal punto di vista del primo giocatore (*MAX*). Si mostri come l'algoritmo *min-max* e l'algoritmo *alfa-beta* risolvono il problema e la mossa selezionata dal giocatore.




## Esercizio 3 (5 punti)

Si considerino matrici quadrate NxN, rappresentate come liste di liste (la lista ha N elementi lista, ciascuna di N elementi). Scrivere un programma PROLOG `prodScal(A,B,C)` che, data la matrice quadrata (lista di liste) A, e lo scalare B, calcoli la matrice C come prodotto di A per lo scalare B.
Ad esempio alla query:

```
?-  prodScal([[1,2],[1,3]],5, C).
```

la risposta sarà:

```
yes C=[[5,10], [5,15]]
```

## Esercizio 4 (5 punti)

Dato il programma Prolog:

```
setdiff(L,[],L):-!.
setdiff([],_,[]):-!.
setdiff([H|T1],L2,[H|T2]):- not(member(H,L2)),!,
                            setdiff(T1,L2,T2).
setdiff([_|T],L2,T2):- setdiff(T,L2,T2).

member(X,[X|_]):-!.
```

```
member(X,[_|T]):-member(X,T).
```

Si rappresenti l'albero di derivazione SLDNF relativo al goal
```
?-setdiff([3],[2,3],L).
```
e si indichi qual è la risposta calcolata.

## Esercizio 5 (6 punti)

Si consideri il problema 3-puzzle (una versione semplificata dell' 8-puzzle, o gioco del filetto) in cui la griglia è 2 x 2 e ci sono tre piastrelle, numerate 1 , 2 , e 3 , e uno spazio vuoto.
Si hanno quattro operatori, che muovono il vuoto su, giù, a sinistra , o a destra.
Lo stato iniziale (Start) e obiettivo (Goal) sono indicati di seguito:

Start

| | |
|---|---|
| 2 | 3 |
| 1 | |

Goal

| | |
|---|---|
| 1 | 2 |
| | 3 |

Mostrare come si individua la sequenza di azioni che porta da Start a Goal, sviluppando lo spazio di ricerca utilizzando le seguenti strategie:

a) breadth-first
b) depth-first
c) A* con funzione euristica h(...) data dal numero di tessere "fuori posto" rispetto al goal. Ogni mossa effettuata ha costo unitario.

Si assuma che le strategie di ricerca non "ricordino" i nodi visitati in precedenza. Inoltre, si utilizzino gli operatori (su, giù, sinistra, destra) nell'ordine indicato, a meno che il metodo di ricerca non stabilisca in diverso modo.
Nello sviluppo dell'albero, etichettare ogni nodo visitato con un numero che indica l'ordine di visita.
Si indichino inoltre quale/quali strategia/e porta ad una situazione di loop.

## Esercizio 6 (3 punti)

Gordon Ramsey, nel suo ristorante londinese, sta predisponendo un menu per un evento speciale. Ci sono diverse portate, ciascuna rappresentata da una variabile:

(A)ppetizer, (B)everage, main (C)ourse, and (D)essert.

Gordon deve decidere cosa servire, avendo diverse scelte per ogni portata. I domini delle variabili sono i seguenti:

A: (v)eggies, (e)scargot
B: (w)ater, (s)oda, (m)ilk
C: (f)ish, (b)eef, (p)asta
D: (a)pple pie, (i)ce cream, (ch)eese

Affinché tutti gli ospiti abbiano lo stesso menu, Gordon deve inoltre soddisfare i seguenti vincoli:
(i) Se l'Appetizer è veggies, il main Course non è pasta, e viceversa (se il main Course è pasta, l'Appetizer non è veggies);
(ii) Se si servono escargot come Appetizer, il Beverage può essere solo water (questo per limitare il costo totale del menu).
(iii) Se si serve milk, il Dessert non è né ice-cream né cheese (per non avere troppo calcio nel menu).
Aiutiamo Gordon a individuare i possibili menu.
Si risponda, quindi, ai seguenti punti:

(a) Modellare i vincoli precedenti tra le variabili A, B, C, e D.
(b) Si assegni inizialmente A=e. Si utilizzi la propagazione del Forward Checking, mostrando come si riducono i domini delle altre variabili.

## Esercizio 7 (2 punti)

Si definisca le proprietà di ammissibilità di una funzione euristica. Che proprietà implica nell'algoritmo di ricerca A*?

# FONDAMENTI DI INTELLIGENZA ARTIFICIALE

# 28 gennaio 2016 – Soluzioni

## Esercizio 1

Conversione in clausole

1. *Anna ama gli stessi dolci che ama Luca*.

   $\forall P, dolce(P) \wedge ama(luca, P) \Rightarrow ama(anna, P)$
   $\forall P, \sim dolce(P) \vee \sim ama(luca, P) \vee ama(anna, P)$

   $\sim dolce(P) \vee \sim ama(luca, P) \vee ama(anna, P)$

2. *Luca ama gli stessi dolci che ama Anna*

   $\sim dolce(P) \vee \sim ama(anna, P) \vee ama(luca, P)$

3. *Chi ama il dolce torta al cioccolato ingrassa.*

   $\forall X\ ama(X,tortacioc) \Rightarrow ingrassa(X)$

$\sim ama(X, tortacioc) \vee ingrassa(X)$

3.a $dolce(tortacioc)$

4. *Anna non ingrassa.*

   $\sim ingrassa(anna)$.

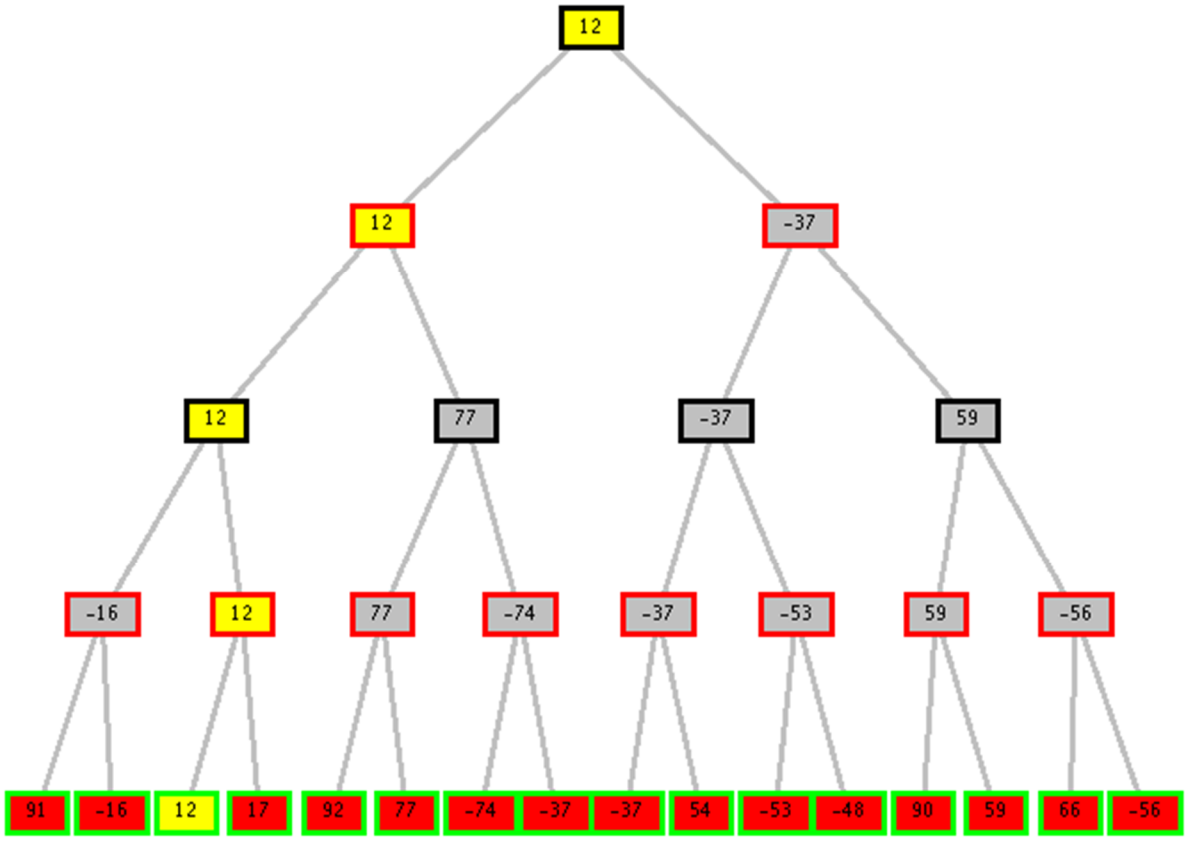
Q. *ci sono dolci che Luca non ama.*
$\exists P, dolce(P) \wedge \sim ama(luca, P)$
$\sim Q: \sim(\exists P, dolce(P) \wedge \sim ama(luca, P))$
$\forall P, \sim dolce(P) \vee ama(luca, P)$
$\sim dolce(P) \vee ama(luca, P)$

Risoluzione

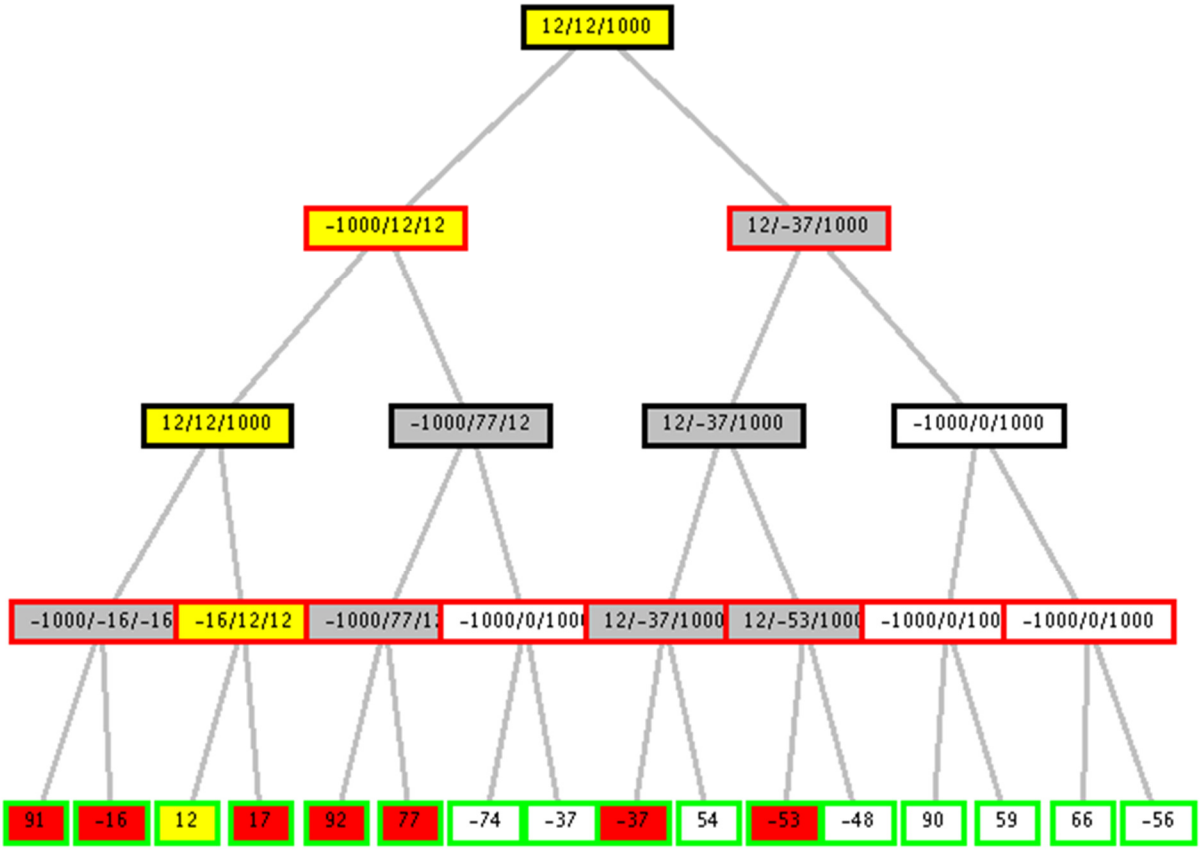
5. (~Q+1) $\sim dolce(P) \vee ama(anna, P)$
6. (5+3a) $ama(anna,tortacioc)$
7. (6+3) $ingrassa(anna)$
8. (7+4) []

## Esercizio 2

Min-Max:




Alfa-beta:



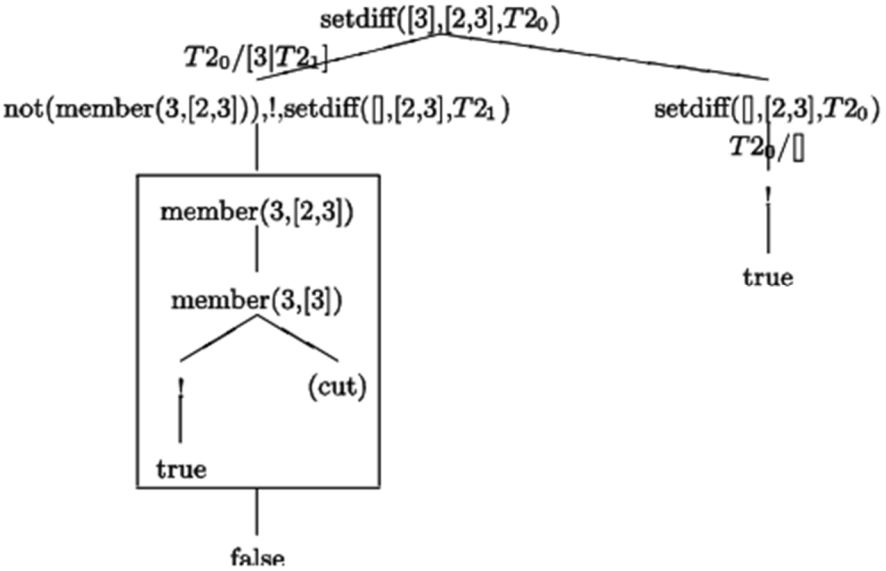

I nodi che portano alla soluzione sono in giallo, quelli tagliati in bianco.

## Esercizio 3

```
prodScal([],_,[]).
prodScal([H|T],B,[H1|T1]):-prodAux(H,B,H1),
                           prodScal(T,B,T1).
```

```
prodAux([],_,[]).
prodAux([H|T],B,[H1|T1]):-H1 is H*B, prodAux(T,B,T1).
```

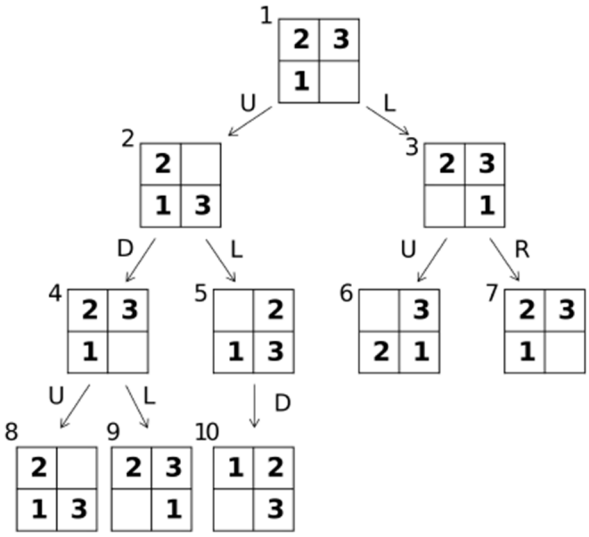
## Esercizio 4



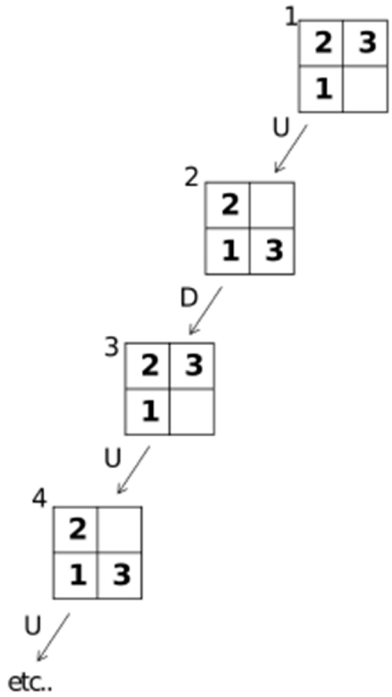

La risposta calcolata è L=[].

## Esercizio 5

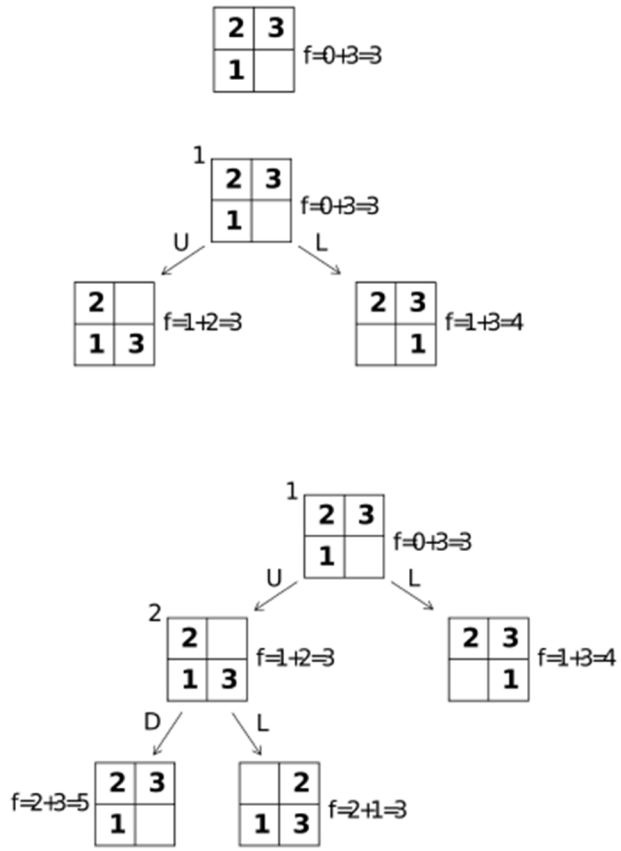
a) Breadth first:

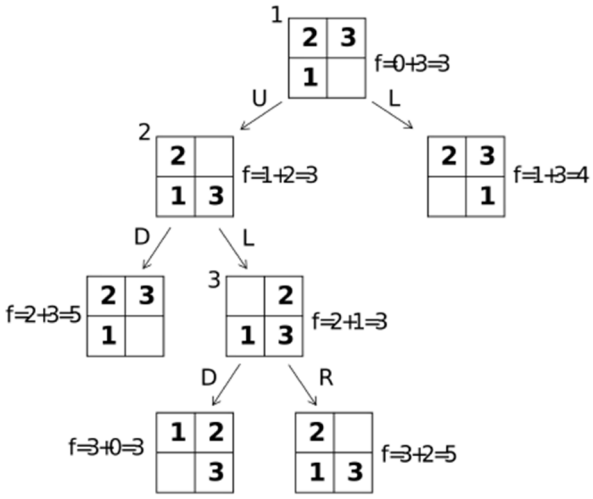



b) Depth first:


1
2 3
1
U
2
2
1 3
D
3
2 3
1
U
4
2
1 3
U
etc..

c) A-star:


2 3
1
f=0+3=3
1
2 3
1
f=0+3=3
U
L
2
1 3
f=1+2=3
2 3
1
f=1+3=4
1
2 3
1
f=0+3=3
U
L
2
2
1 3
f=1+2=3
2 3
1
f=1+3=4
D
L
f=2+3=5
2 3
1
2
1 3
f=2+1=3




## Esercizio 6

(a)

(i) Se l'Appetizer è veggies il main Course non è pasta, e viceversa (se il main Course è pasta, l'Appetizer non è veggies):

A=w => C≠p

C=p => A≠w

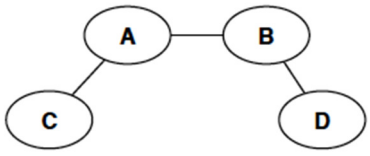
(ii) Se si servono escargot come Appetizer, il Beverage può essere solo water (questo per limitare il costo totale del menu):

A=e => B=w

(iii) Si si serve milk, il Dessert non è né ice-cream né cheese (per non avere troppo calcio nel menu). Aiutiamo Gordon a individuare i possibili menu.

B=m => D≠i

B=m => D≠c




(b)

| VARIABILI | DOMINIO_iniziale | DOMINIO_FC | Motivo |
|---|---|---|---|
| A | e | e | |
| B | w, s, m | w | Vincolo (ii) |
| C | f, b, p | f, b, p | |
| D | a, i, ch | a, i, ch | |

## Esercizio 7

Vedi slide del corso.