



### Esercizio 3 (punti 6)

Andrea, Bebo e Carlo (A, B, C) sono tre bambini, rispettivamente di età compresa tra 2 e 10 anni (Andrea, estremi inclusi), e di età compresa tra 1 e 10 anni (Bebo e Carlo, estremi inclusi). Andrea ha il doppio degli anni di Carlo. Bebo ha un'età pari alla metà della differenza in anni tra Andrea e Carlo. Le loro mamme (MA, MB, MC) sono molto amiche e hanno più o meno la stessa età (compresa tra 36 e 40, estremi inclusi). La mamma di Carlo ha dieci volte l'età di Andrea. La mamma di Andrea ha l'età che aveva la mamma di Carlo quando è nato Carlo. La mamma di Bebo ha un'età che è venti volte la differenza di età della mamma di Carlo e della mamma di Andrea.

Si definisca il problema come CSP, e lo si risolva applicando la tecnica del Forward Checking (FC) dopo ogni passo di labeling. Si parta scegliendo per il primo passo di labeling la variabile A. Si scelga sempre la variabile successiva da istanziare secondo il principio MRV (Minimum Remaining Value). Nei casi non deterministici si scelga prima una variabile "bambino" rispetto a una "mamma". I valori del dominio siano provati secondo l'ordine crescente. Nel passo di propagazione FC si propagano solo i vincoli nei quali resta una sola variabile futura non istanziata.

### Esercizio 4 (punti 5)

Definire il programma Prolog e il goal con cui si può modellare e risolvere, con un approccio del tipo Generate&Test, il CSP dell'Esercizio 3.

Nota bene: per generare le possibili istanziazioni per ciascuna variabile è sufficiente utilizzare la versione generativa del predicato **member/2** (visto a lezione, e da riportare nella soluzione proposta); per verificare se le variabili istanziate soddisfano i vincoli, si definisca un opportuno predicato **test/6** con le condizioni espresse dai vincoli aritmetici sulle variabili.

Si mostri infine un possibile goal che, invocando member/2 (per ogni variabile) e test/6 in maniera opportuna, fornisca la soluzione al problema.

### Esercizio 5 (punti 6)

Si consideri il seguente problema di "mondo a blocchi", nel quale si vuole passare dallo stato iniziale allo stato goal:



Le mosse consentite sono spostare uno o più blocchi "liberi" sul piano o sopra un altro blocco (o pila di blocchi). Si mostri come l'algoritmo A\* trova una soluzione, considerando l'euristica che conta il numero di blocchi "fuori posto". Un blocco è "fuori posto" se la colonna di blocchi sottostanti ad esso non è corretta rispetto alla configurazione da raggiungere). Ad esempio, nello stato iniziale, la funzione euristica vale 3, perché tutti i blocchi sono "fuori posto" eccetto A.

Questa euristica sovrastima o meno la distanza dallo stato goal (non è o è ammissibile)? Trova la soluzione ottima?

### Esercizio 6 (punti 4)

Si spieghi cosa si intende per pianificazione classica e per pianificazione reattiva, illustrandone le differenze, i vantaggi e gli svantaggi.

# FONDAMENTI DI INTELLIGENZA ARTIFICIALE

13 Febbraio 2015 – Soluzioni

## Esercizio 1

In logica dei predicati del I ordine:

$\forall X \forall Y$  acquista(X,Y) and rubato(Y)  $\rightarrow$  compie\_atto\_illegale(X)

$\exists X$  possiede(giovanni,X) and rubato(X).

$\forall X$  possiede(giovanni,X)  $\rightarrow$  acquista(mario,X)

Goal: compie\_atto\_illegale(mario)

In forma a clausole:

C1: not acquista(X,Y) or not rubato(Y) or compie\_atto\_illegale(X)

C2: possiede(giovanni,c1)

C3: rubato(c1)

C4: not possiede(giovanni,X) or acquista(mario,X)

Goal negato: C5 not compie\_atto\_illegale(mario)

Risoluzione

Da C5 e C1, C6: not acquista(mario,Y) or not rubato(Y)

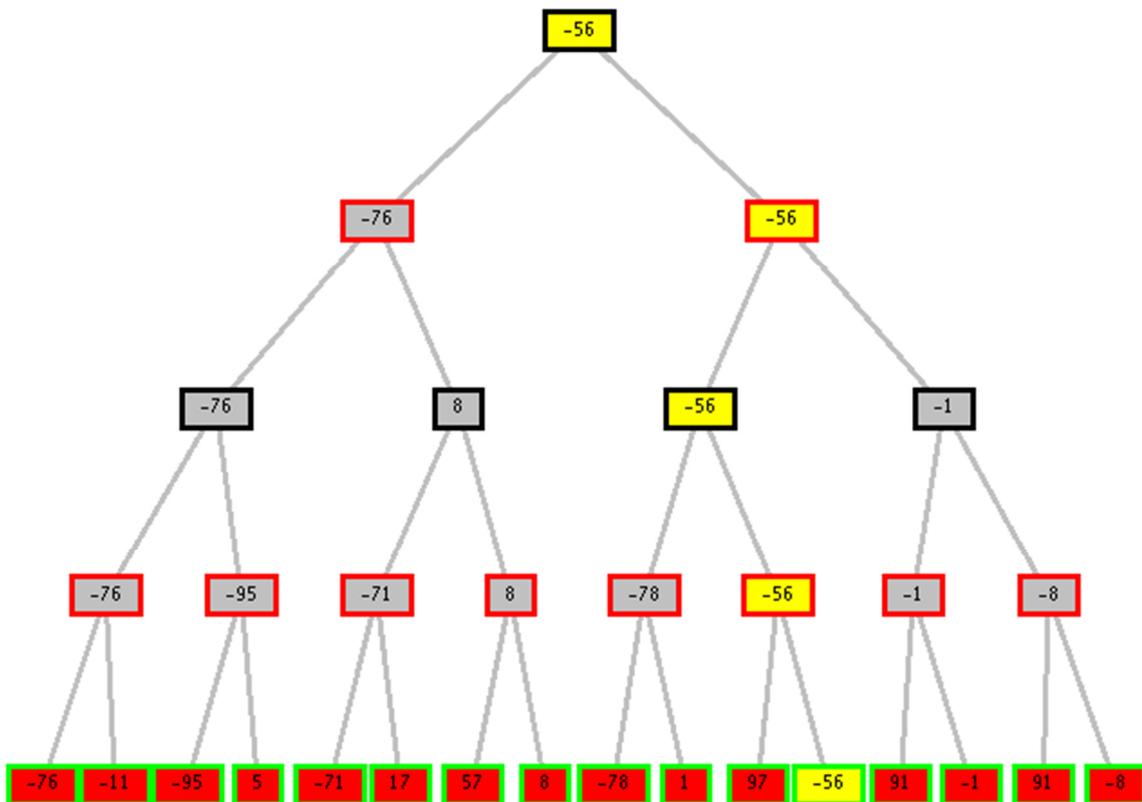
Da C6 e C3, C7: not acquista(mario,c1)

Da C7 e C4, C8: not possiede(giovanni,c1)

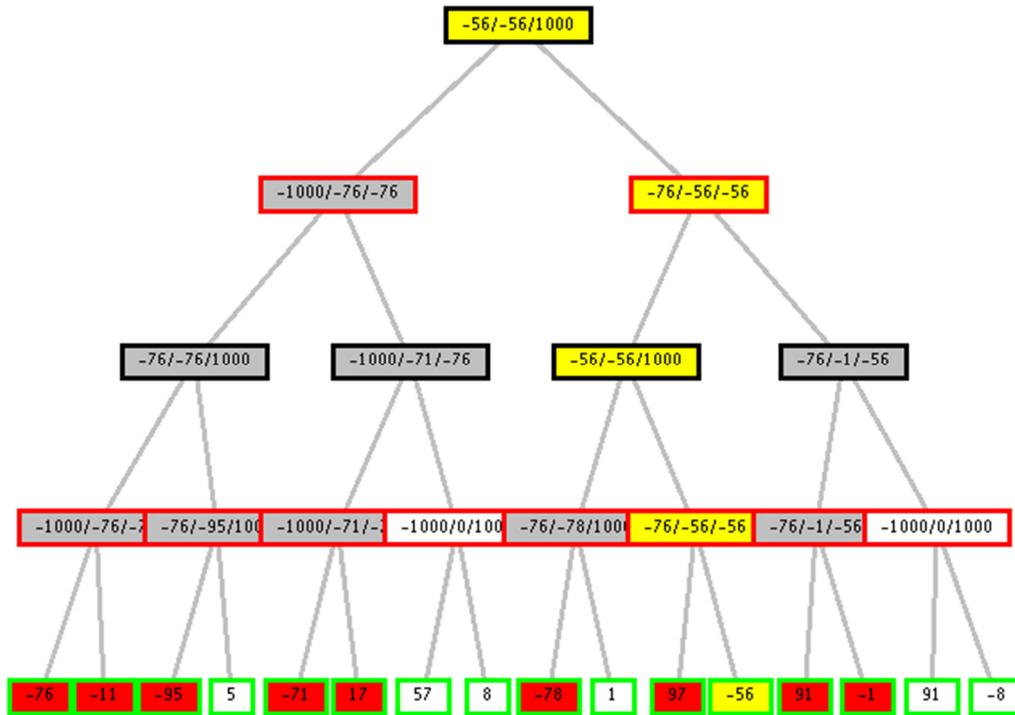
Da C8 e C2 segue la contraddizione.

## Esercizio 2

Min-Max:



Alfa-Beta:



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

### Esercizio 3

Nonostante dal problema si possa evincere facilmente che A e C sono pari, e quindi anche MC ed MB, tali conclusioni non si dovrebbero applicare nell'uso della tecnica FC, poiché sarebbero equivalenti ad un passo di arc-consistency. Quindi i domini sono:

A::[2,3,4,5,6,7,8,9,10]

B,C::[1,2,3,4,5,6,7,8,9,10]

MA,MB,MC::[36,37,38,39,40]

$A = 2 * C$

$B = (A - C) / 2$

$MC = 10 * A$

$MA = MC - C$

$MB = 20 * (MC - MA)$

Considerando le variabili a partire da A (e scegliendo la variabile successiva secondo il MRV nei passi di labeling, e nei casi non deterministici scegliendo prima una variabile "bambino" rispetto a una "mamma") e applicando il FC dopo ogni passo di labeling:

	A	B	C	MA	MB	MC
	[2,3,4,5,6,7,8,9,10]	[1,2,3,4,5,6,7,8,9,10]	[1,2,3,4,5,6,7,8,9,10]	[36,37,38,39,40]	[36,37,38,39,40]	[36,37,38,39,40]
Labeling e FC	A=2	[1,2,3,4,5,6,7,8,9,10]	[1]	[36,37,38,39,40]	[36,37,38,39,40]	<b>Fail</b>
Backtracking						
Labeling e FC	A=3	[1,2,3,4,5,6,7,8,9,10]	<b>Fail</b>	[36,37,38,39,40]	[36,37,38,39,40]	<b>Fail</b>
Backtracking						
Labeling e FC	A=4	[1,2,3,4,5,6,7,8,9,10]	[2]	[36,37,38,39,40]	[36,37,38,39,40]	[40]
Labeling e FC		[1]	C=2	[36,37,38,39,40]	[36,37,38,39,40]	[40]
Labeling e FC		B=1		[36,37,38,39,40]	[36,37,38,39,40]	[40]
Labeling e FC				[38]	[36,37,38,39,40]	MC=40
Labeling e FC				MA=38	[40]	
Labeling					MB=40	

### Esercizio 4

%predicato member generativo:

member( X, [ X| \_]).

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

% predicato di test:

test( A, B, C, MA, MB, MC) :- A is 2\*C, B is ((A-C)/2), MC is 10\*A, MA is MC-C, MB is 20\*(MC-MA).

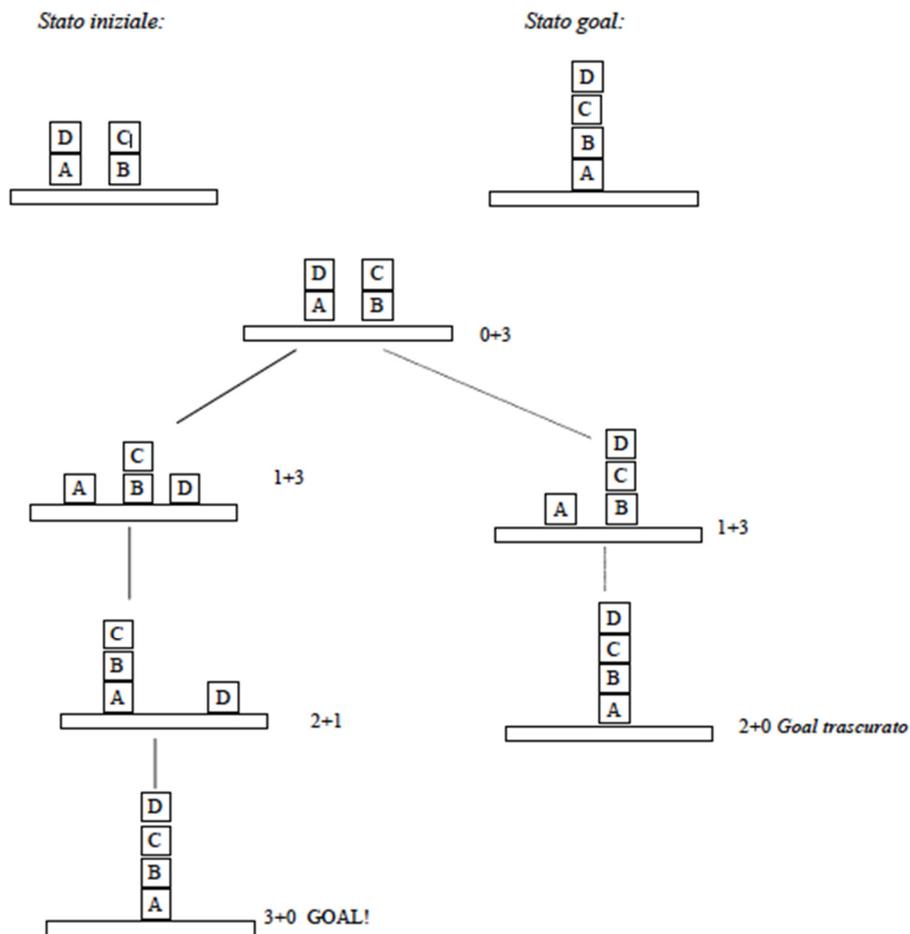
Goal:

```
?- member( A, [2,3,4,5,6,7,8,9,10]),
   member( B, [1,2,3,4,5,6,7,8,9,10]),
   member( C, [1,2,3,4,5,6,7,8,9,10]),
   member( MA, [36,37,38,39,40]),
   member( MB, [36,37,38,39,40]),
   member( MC, [36,37,38,39,40]),
   test( A, B, C, MA, MB, MC).
```

Risposta:

A = 4, B = 1, C = 2, MA = 38, MB = 40, MC = 40.

### Esercizio 5



L'algoritmo trova la soluzione a profondità 3, prima di quella ottima a profondità 2, poiché la funzione euristica non è ammissibile (sovrastima la distanza dal goal).

### Esercizio 6

Vedi slide del corso.