

Esercizio 1 (6 punti)

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

- Ogni persona lavora o è disoccupata (xor)
- Tutte le persone che lavorano sono maggiorenni
- Tutte le persone sono maggiorenni o minorenni (xor)
- Giorgio è maggiorenne e Maria è minorenne e sono entrambi persone

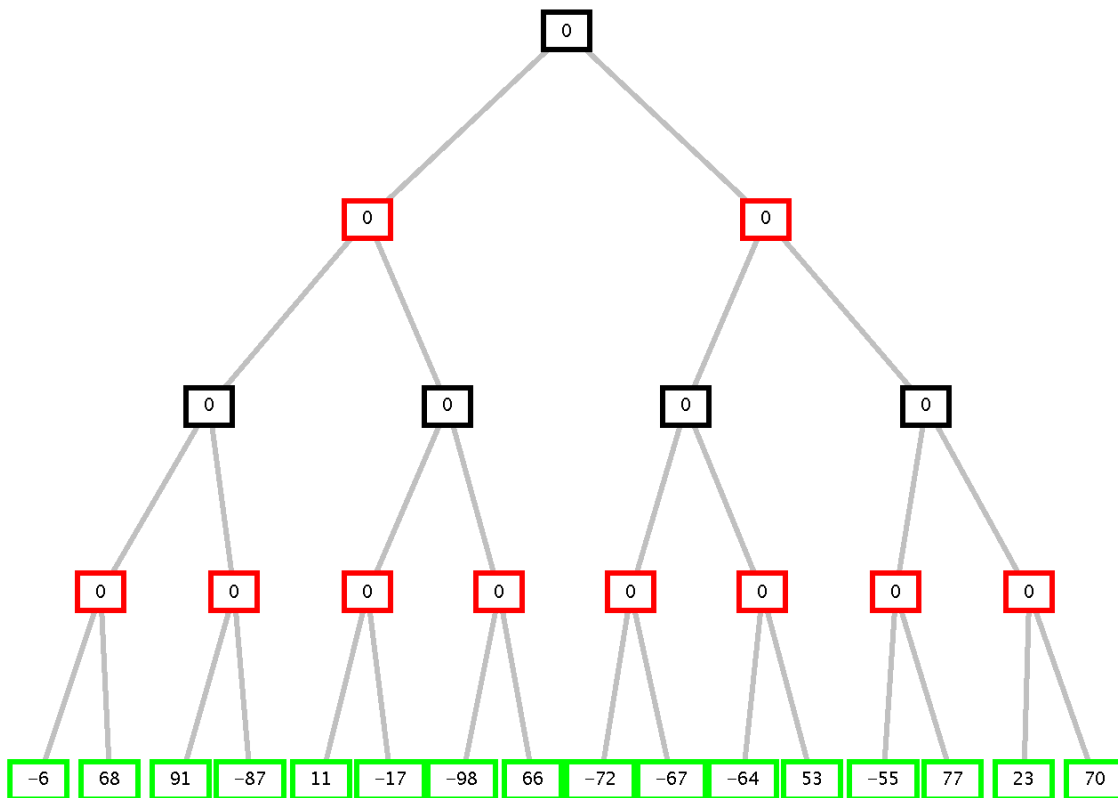
Le si trasformi in clausole, utilizzando i seguenti predicati:

lavora(X) persona(X) disoccupata(X)
 maggiorenne(X) minorenne(X)

con l'ovvio significato, e si applichi poi la risoluzione per dimostrare che Maria è disoccupata.

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 (arco a sinistra o a destra).
- Si mostrino poi i tagli che l'algoritmo alfa-beta consente, indicando gli archi che verranno tagliati.

Esercizio 3 (5 punti)

Si scriva un predicato Prolog **vicini(X,Y,Z)** che è vero se X e Y sono adiacenti nella lista Z indipendentemente dall'ordine in cui compaiono. Ad esempio:

?-vicini(3,4,[1,2,3,4,5,6]).

true

?-vicini(3,4,[1,2,4,3,5,6]).

true

?-vicini(3,4,[1,2,3,5,4]).

false

?-vicini(3,4,[]).

false

?-vicini(X,Y,[1,2,3]).

X = 1

Y = 2;

X = 2
Y = 1;
X = 2
Y = 3;
X = 3
Y = 2;
false

Esercizio 4 (6 punti)

Il seguente programma Prolog determina il massimo valore in una lista di interi:

max([X],X).

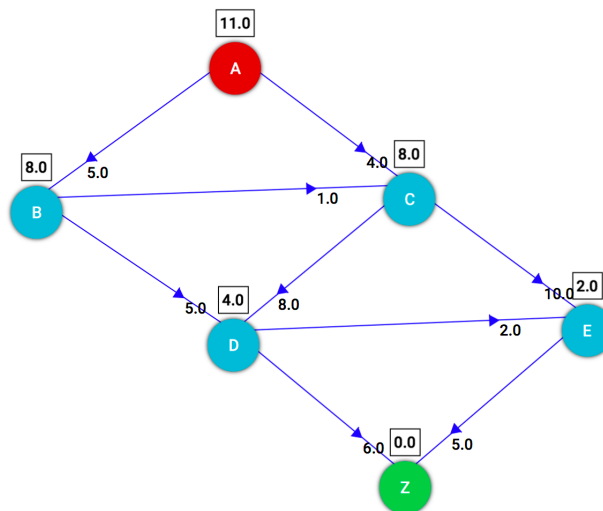
max([X|T],X):-max(T,E1), X>E1, !.

max([_|T],E1):-max(T,E1).

Si mostri l'albero SLD relativo al goal: $?- \text{max}([9,12], X)$.

Esercizio 5 (6 punti)

Si consideri il seguente grafo, dove A è il nodo iniziale e Z 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 quadratino, è indicata inoltre la stima euristica della sua distanza dal nodo goal:



Si applichi la ricerca **A*** su alberi (non tenendo quindi traccia dei nodi già visitati) **disegnando l'albero generato dinamicamente**. In caso di non determinismo si selezionino i nodi da espandere secondo l'ordine alfabetico.

Si indichino:

- i nodi espansi nell'ordine di espansione;
- i nodi sulla strada della soluzione e il costo della soluzione;
- la condizione sulla stima euristica $h(n)$ che garantisce l'ottimalità della ricerca su alberi e se è soddisfatta o meno in questo caso.

Esercizio 6 (4 punti)

Dopo avere brevemente introdotto l'algoritmo di Arc-Consistency, se ne mostri l'esecuzione su questo esempio:

A::[1, 2, 3, 4]

B::[1, 2, 3]

C::[1, 2, 3]

$A = 3 * B + 1$

$C \geq B + 2$

$A \leq 10 - C$

considerando gli archi nel seguente ordine:

A --> B, B --> A, C --> B, B --> C, A --> C, **C --> A**

Fino alla quiescenza della rete, mostrando - per ogni iterazione - i domini di ogni variabile.

Indicare se si può già identificare la soluzione del CSP o no, al termine dell'applicazione di Arc-Consistency.

Esercizio 1

Traduzione in predicati in logica del primo ordine:

- *Ogni persona lavora o è disoccupata* (xor)
 $\forall X \text{ persona}(X) \rightarrow \text{lavora}(X) \text{ xor } \text{disoccupata}(X)$
- *Tutte le persone che lavorano sono maggiorenni.*
 $\forall X \text{ persona}(X) \text{ and } \text{lavora}(X) \rightarrow \text{maggiorenne}(X)$
- *Tutte le persone sono maggiorenni o minorenni* (xor)
 $\forall X \text{ persona}(X) \rightarrow \text{maggiorenne}(X) \text{ xor } \text{minorenne}(X)$
- *Giorgio è maggiorenne e Maria è minorenne e sono entrambi persone.*
- $\text{maggiorenne}(\text{giorgio}).$
 $\text{minorenne}(\text{maria}).$
 $\text{persona}(\text{giorgio}).$
 $\text{persona}(\text{maria}).$
- Negazione del Goal: *Maria non è disoccupata* $\text{not disoccupata}(\text{maria})$

Traduzione in clausole:

1. $\text{not persona}(X) \text{ or } \text{lavora}(X) \text{ or } \text{disoccupata}(X).$
2. $\text{not persona}(X) \text{ or } \text{not lavora}(X) \text{ or } \text{not disoccupata}(X).$
3. $\text{not persona}(X) \text{ or } \text{not lavora}(X) \text{ or } \text{maggiorenne}(X).$
4. $\text{not persona}(X) \text{ or } \text{maggiorenne}(X) \text{ or } \text{minorenne}(X).$
5. $\text{not persona}(X) \text{ or } \text{not maggiorenne}(X) \text{ or } \text{not minorenne}(X).$
6. $\text{maggiorenne}(\text{giorgio}).$
7. $\text{minorenne}(\text{maria}).$
8. $\text{persona}(\text{giorgio}).$
9. $\text{persona}(\text{maria}).$
10. $\text{not disoccupata}(\text{maria})$

Risoluzione:

$10 + 1 = 11$ $\text{not persona}(\text{maria}) \text{ or } \text{lavora}(\text{maria})$

$11 + 3 = 12$ $\text{not persona}(\text{maria}) \text{ or } \text{maggiorenne}(\text{maria})$

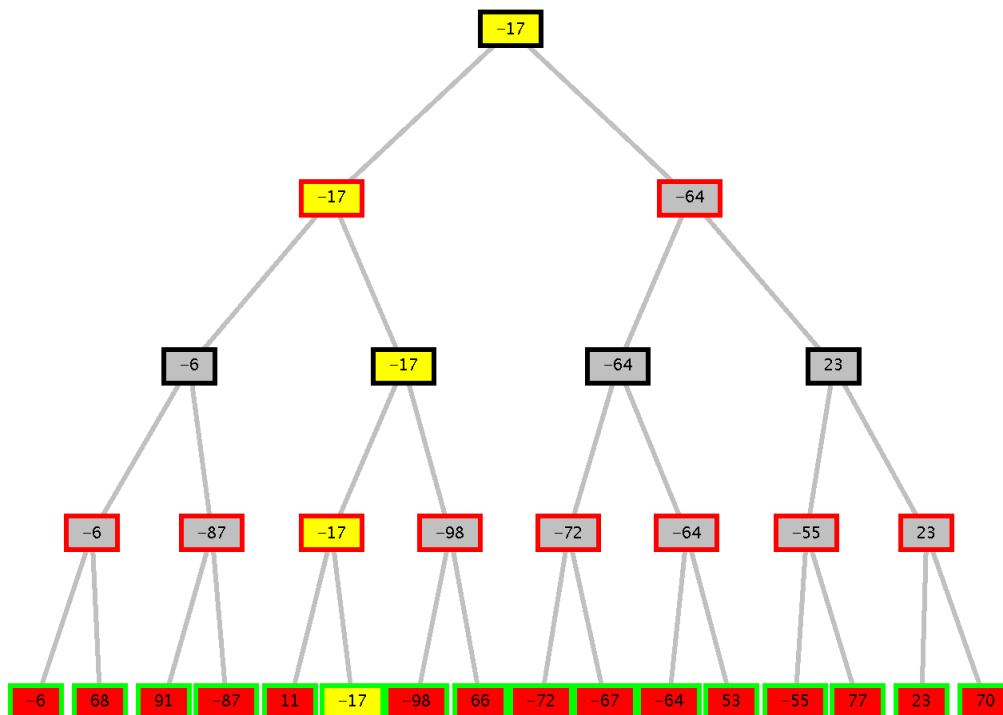
$12 + 5 = 13$ $\text{not persona}(\text{maria}) \text{ or } \text{not minorenne}(\text{maria})$

$13 + 9 = 14$ $\text{not minorenne}(\text{maria})$

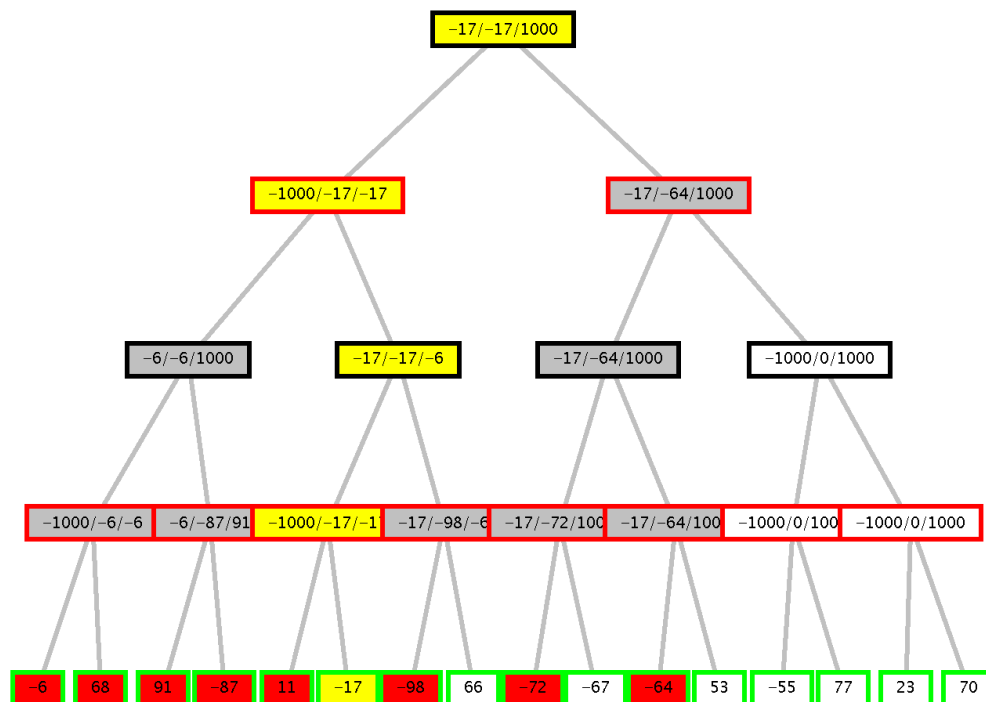
$14 + 7 =$ Contraddizione!!

Esercizio 2

Min-Max: in giallo la strada selezionata.



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.



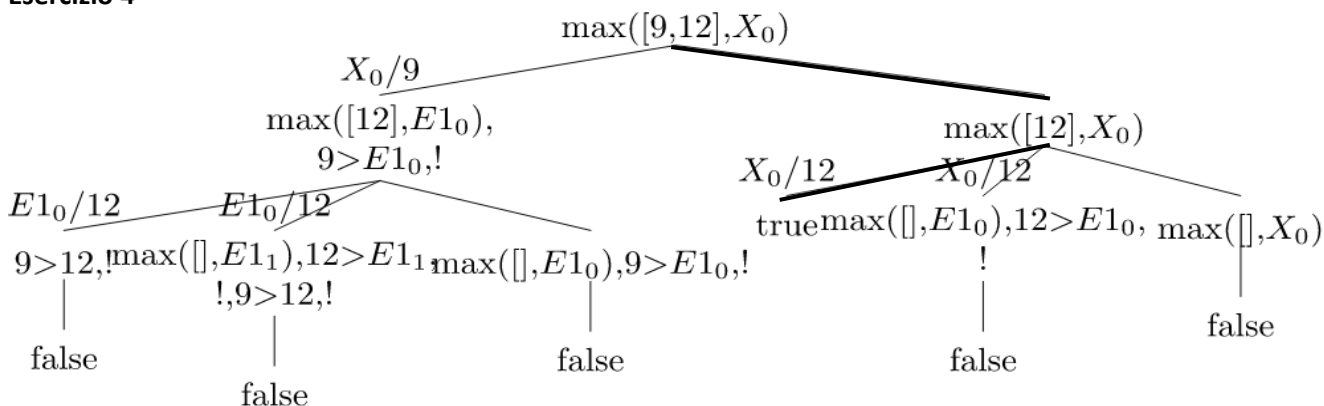
Esercizio 3

vicini(X,Y,[X,Y|_]).

vicini(X,Y,[Y,X|_]).

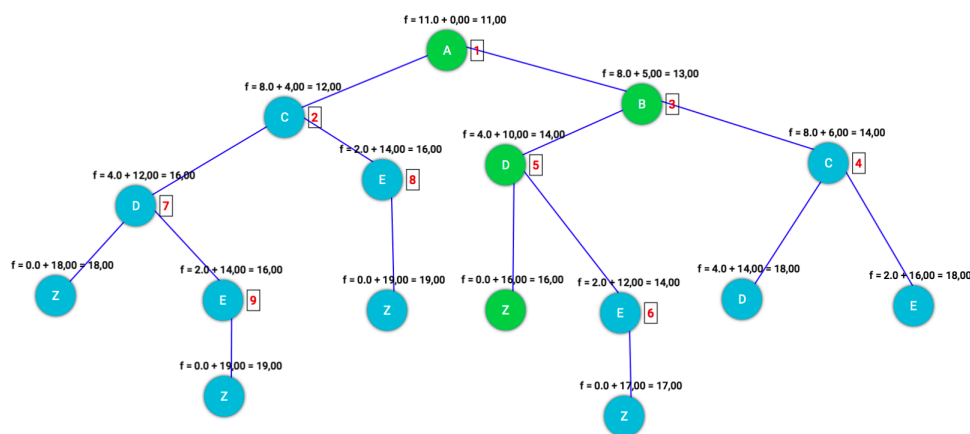
vicini(X,Y,[_|Zs]):-vicini(X,Y,Zs).

Esercizio 4



Esercizio 5

Con A* i nodi espansi sono indicati in figura (ACBCDEEEZ), la soluzione ABDZ ha costo 16 (ottimale):



Operations

☒ Show operations

1) A [f = 11.0 + 0.00 = 11.00]
2) C [f = 8.0 + 4.00 = 12.00]
3) B [f = 8.0 + 5.00 = 13.00]
4) C [f = 8.0 + 6.00 = 14.00]
5) D [f = 4.0 + 10.00 = 14.00]
6) E [f = 2.0 + 12.00 = 14.00]
7) D [f = 4.0 + 12.00 = 16.00]
8) E [f = 2.0 + 14.00 = 16.00]
9) E [f = 2.0 + 14.00 = 16.00]
/) Z [f = 0.0 + 18.00 = 18.00]
/) Z [f = 0.0 + 19.00 = 19.00]
/) Z [f = 0.0 + 19.00 = 19.00]
/) Z [f = 0.0 + 16.00 = 16.00]
/) Z [f = 0.0 + 17.00 = 17.00]
/) D [f = 4.0 + 14.00 = 18.00]
/) E [f = 2.0 + 16.00 = 18.00]

Path cost: 16.0
Nodes expanded: 9
Queue size: 6
Max queue size: 7

La condizione sulla funzione euristica stimata $h^*(n)$ che garantisce l'ottimalità della ricerca è la condizione di ammissibilità che deve valere per ogni nodo dell'albero e che è verificata se la $h^*(n)$ è ottimista cioè $h^*(n) \leq h(n)$. Tale condizione è soddisfatta in questo caso.

Esercizio 6

Vedi slide del corso per spiegare AC.

1a iterazione

Archi	Vincolo	Dominio ridotto
A --> B	A=3*B+1	A:: [4]
B --> A	B=(A-1)/3	B:: [1]
C --> B	C>=B+2	C:: [3]
B --> C	B<=C-2	B:: [1] non varia
A --> C	A<=10-C	A:: [4] non varia
C --> A	C<=10-A	C:: [3] non varia

2a iterazione: nessun dominio varia (quiescenza della rete).

Soluzione: A=4, B=1, C=3.