

PROBLEMI DI SODDISFACIMENTO DI VINCOLI

ESERCIZIO

Ci sono 5 scatole (S_1, \dots, S_5 nell'ordine) che possono contenere numeri da 1 a 5.

Si inseriscano i 5 numeri nelle scatole rispettando il vincolo:

Si può contenere il numero N solo se sia S_{i-1} sia S_{i+1} non contengono i numeri $N+1$ o $N-1$.

Per la soluzione si utilizzi la strategia di forward checking e si mostrino i passi eseguiti.

SOLUZIONE

A ogni scatola $S_1..S_5$ associo un dominio che inizialmente contiene i valori 1,2,3,4,5

$$D1 = [1,2,3,4,5]$$

$$D2 = [1,2,3,4,5]$$

$$D3 = [1,2,3,4,5]$$

$$D4 = [1,2,3,4,5]$$

$$D5 = [1,2,3,4,5]$$

Vincoli: $S_i \neq S_{i\pm 1} \neq 1$ $S_i \neq S_j$ se $i \neq j$

La tecnica del forward checking, dopo ogni istanziazione, propaga i vincoli eliminando dai domini associati alle variabili coinvolte nei vincoli stessi, i valori incompatibili con l'assegnazione appena effettuata.

Prima assegnazione $S_1 = 1$

$$D1 = 1$$

$$D2 = [3,4,5]$$

$$D3 = [2,3,4,5]$$

$$D4 = [2,3,4,5]$$

$$D5 = [2,3,4,5]$$

Seconda assegnazione $S_2 = 3$

$$D1 = 1$$

$$D2 = 3$$

$$D3 = 5$$

$$D4 = [2,4,5]$$

$$D5 = [2,4,5]$$

La terza assegnazione è obbligata $S_3 = 5$

$$D1 = 1$$

$$D2 = 3$$

$$D3 = 5$$

$$D4 = 2$$

$$D5 = [4,2]$$

La quarta assegnazione è obbligata $S_4 = 2$

$$D1 = 1$$

$$D2 = 3$$

$$D3 = 5$$

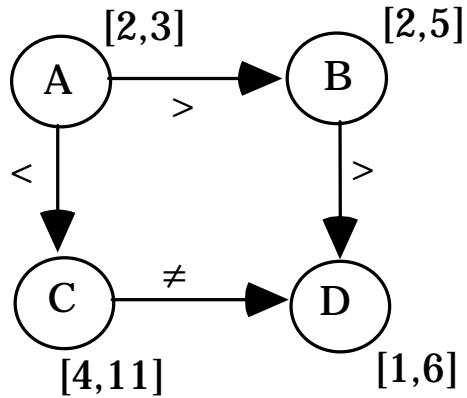
$$D4 = 2$$

$$D5 = 4$$

L'ultima assegnazione è obbligata $S_5 = 4$

ESERCIZIO

È dato il seguente grafo di vincoli:



in cui i nodi rappresentano le variabili del problema il cui dominio è rappresentato dai valori entro parentesi quadra.

$A :: [2, 3]$, $B :: [2, 5]$, $C :: [4, 11]$, $D :: [1, 6]$

Gli archi rappresentano i vincoli tra le variabili.

$A > B$, $B > D$, $A < C$, $C \neq D$

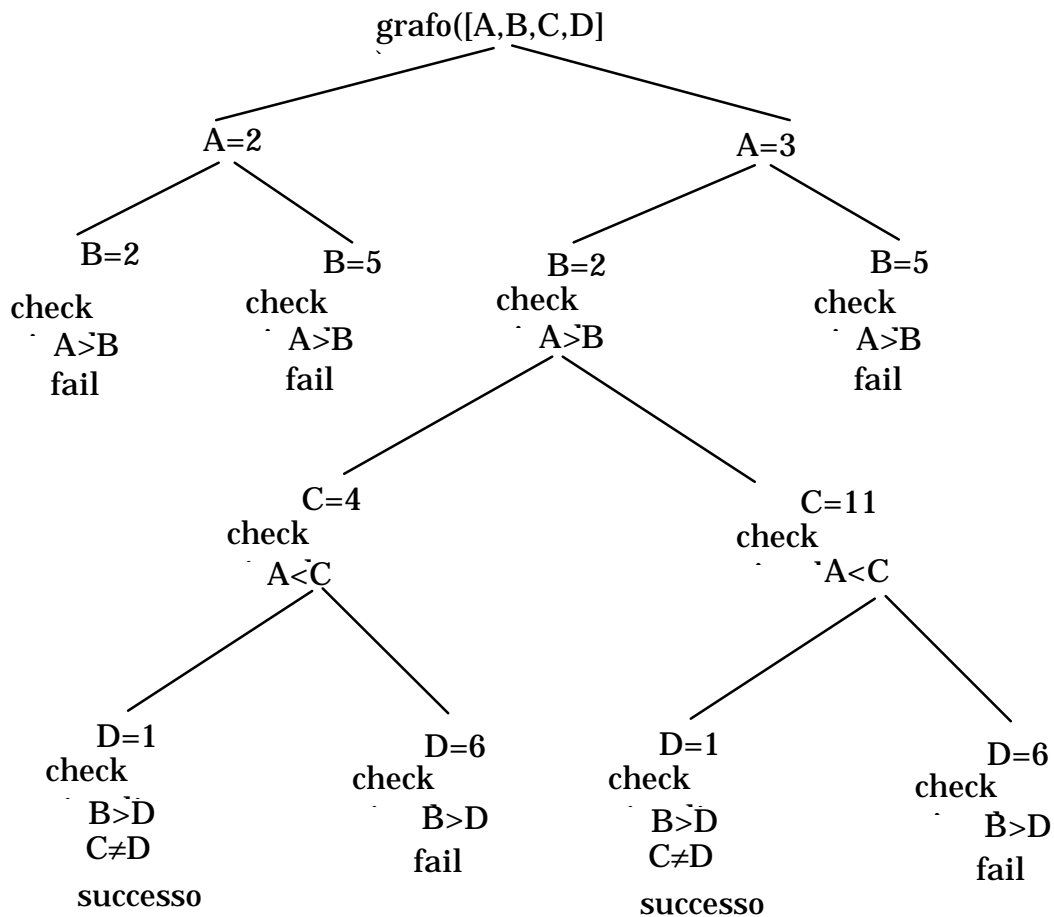
Si descriva il comportamento del programma, lo spazio di ricerca esplorato in caso di strategia standard backtracking e forward checking.

SOLUZIONE

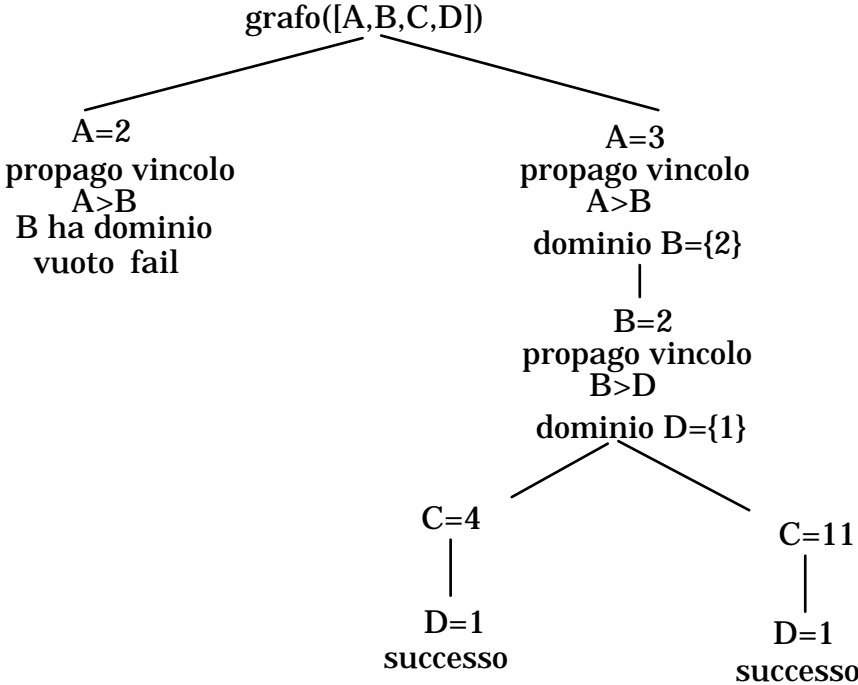
Comportamento del programma:

i vincoli $A > B$, $B > D$, $A < C$, $C \neq D$ vengono addormentati perché le variabili non sono sufficientemente istanziate. La propagazione dei vincoli cambia nel caso di strategia standard backtracking e forward checking.

Standard Backtracking:

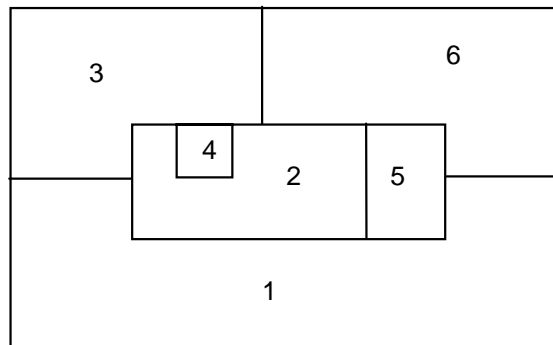


Forward Checking:



ESERCIZIO

Si consideri il problema di colorare la mappa seguente usando solo 4 colori in modo che nessuna regione adiacente condivida lo stesso colore.



Si scriva la conoscenza e i vincoli coinvolti e si mostri come si arriva alla soluzione nel caso si applichino le seguenti strategie:

Generate and Test
Standard Backtracking
Forward Checking.

SOLUZIONE

Associamo ad ogni zona della mappa (corrispondente ad un numero) un dominio che, inizialmente, contiene 4 colori, ad esempio red (r), blue (b), green (g), yellow (y):
 $D1=[r,b,g,y]$, $D2=[r,b,g,y]$, $D3=[r,b,g,y]$, $D4=[r,b,g,y]$,
 $D5=[r,b,g,y]$, $D6=[r,b,g,y]$.

Vincoli:

$X_i \in D_i$ per $\forall i$

$X_i \neq X_j$ per $\forall i \neq j$ e $\text{near}(i,j)$

Generate and Test:

Genera una permutazione di valori per tutte le variabili tale che $X_i \in D_i$ per $\forall i$ e, solo successivamente testa se l'assegnamento è una soluzione:

$X_1=r, X_2=r, X_3=r, X_4=r, X_5=r, X_6=r$ -> fallimento

$X_1=b, X_2=r, X_3=r, X_4=r, X_5=r, X_6=r$ -> fallimento

$X_1=g, X_2=r, X_3=r, X_4=r, X_5=b, X_6=r$ -> fallimento

$X_1=y, X_2=r, X_3=r, X_4=r, X_5=b, X_6=r$ -> fallimento

Standard Backtracking:

Ogni volta che genera un assegnamento di un valore ad una variabile, testa che tale assegnamento sia compatibile con i vincoli che coinvolgono la variabile appena istanziata e le variabili istanziate precedentemente.

$X_1=r,$

$X_2=r$ fallimento: viola il vincolo con X_1

$X_2=b$

$X_3=r$ fallimento viola il vincolo con X_1

$X_3=b$ fallimento viola il vincolo con X_2

$X_3=g$

$X_4=r$

$X_5=r$ fallimento viola il vincolo con X_1

$X_5=b$ fallimento viola il vincolo con X_2

$X_5=g$

$X_6=r$ fallimento viola il vincolo con X_1

$X_6=b$ fallimento viola il vincolo con X_2

$X_6=g$ fallimento viola il vincolo con X_5

$X_6=y$

soluzione con $X_1=r, X_2=b, X_3=g, X_4=r, X_5=g, X_6=y$

Forward Checking:

Ogni volta che genera un assegnamento di un valore ad una variabile, verifica che tale assegnamento sia compatibile con i vincoli che coinvolgono la variabile appena istanziata e i valori contenuti nei domini delle variabili successive (non ancora istanziate).

$X1=r \rightarrow D2=[b,g,y], D3=[b,g,y], D4=[r,b,g,y], D5=[b,g,y], D6=[b,g,y]$

$X2=b \rightarrow D3=[g,y], D4=[r,g,y], D5=[g,y], D6=[g,y]$

$X3=g \rightarrow D4=[r,y], D5=[g,y], D6=[y]$

$X4=r \rightarrow D5=[g,y], D6=[y]$

$X5=g \rightarrow D6=[y]$

$X6=y$

soluzione con $X1=r, X2=b, X3=g, X4=r, X5=g, X6=y$

Se si utilizzasse il first-fail la quarta istanziazione assegnerebbe a $X6$ (var. più vincolata) il valore y . Questa assegnazione provocherebbe una propagazione su $D5$ ($D5=[g]$) che quindi conterrebbe un solo valore (soluzione)

ESERCIZIO

Si scriva un programma PROLOG che risolva il problema dell'esercizio precedente nel caso in cui si applichi la strategia standard backtracking.

```
quattro_colori(X) :-
    colori([1,2,3,4,5,6], X, [], [], [r,b,g,y]).

colori([], [], [6,5,4,3,2,1], Placed, _).
colori([N1|Ntail], [X|Tail], NPlaced,
Placed, Values) :-
    member(X, Values),
    compatible(X, N1, Placed, NPlaced),
    colori(Ntail, Tail, [N1|NPlaced],
           [X|Placed], Values).

compatible(_, _, [], []).
compatible(X, N1, [P|Tail], [NP|NTail]) :-
    (near(N1, NP); near(NP, N1)), !,
    X \== P,
    compatible(X, N1, Tail, NTail).
compatible(X, N1, [P|Tail], [NP|NTail]) :-
    compatible(X, N1, Tail, NTail).
```

Altra soluzione

Fatti necessari:

```
adiac(1, [2, 3, 5, 6]).   adiac(2, [1.3.4.5, 6]).
adiac(3, [1, 2, 4, 6]).   adiac(4, [2, 3]).
adiac(6, [1, 2, 3, 5]).
```

```
quadro(1). quadro(2). quadro(3). quadro(4).
quadro(5). quadro(6).
```

```
vincoli_soddisfatti([]).
vincoli_soddisfatti([(Quadro, Colore) | Tail])
:-
    vincoli_soddisfatti(Tail),
    member(Colore, [c1, c2, c3, c4]),
    not(member((Quadro1, Colore), Tail),
         adiacente(Quadro, Quadro1)).
```

```
adiacente(Quadro, Quadro1) :-
    adiac(Quadro, Adiacenti),
    member(Quadro1, Adiacenti).
```

Query:

```
trova_soluzione(S) :-
    setof((Quadro, Colore), quadro(Quadro), S),
    vincoli_soddisfatti(S).
```

ESERCIZIO

Si risolva il seguente problema di soddisfacimento di vincoli. Si hanno 6 gatti e 4 ceste disposte in fila. Ogni cesta può contenere al più due gatti. I gatti hanno le seguenti caratteristiche:

- gatto1: adulto rosso
- gatto2: adulto tigrato
- gatto3: cucciolo nero
- gatto4: adulto nero
- gatto5: cucciolo tigrato
- gatto6: adulto tigrato

I vincoli sono i seguenti:

1. I gatti neri adulti devono stare in ceste vicino a gatti rossi.
2. I gatti neri adulti non possono stare né nella stessa cesta né in ceste vicine ai tigrati adulti.
3. I cuccioli possono stare in ceste contenenti solo altri cuccioli.
4. Gli adulti di un certo colore devono stare in ceste vicine a ceste contenenti cuccioli dello stesso colore.

Si definiscano le variabili, i domini e si risolva il problema adottando la strategia del *forward checking*. Per la selezione delle variabili si adotti la strategia *first fail* anche detta MRV (Minimum remaining values)

SOLUZIONE

Le variabili rappresentano i gatti. Il dominio delle variabili contiene i numeri corrispondenti alle ceste.

$G_1, G_2, G_3, G_4, G_5, G_6 :: [1, 2, 3, 4]$

I vincoli sono:

- 1) $G_4 = G_1 \pm 1$
- 2) $G_4 \neq G_2, G_4 \neq G_6, G_4 \neq G_2 \pm 1, G_4 \neq G_6 \pm 1$
- 3) $G_3 \neq G_1, G_3 \neq G_2, G_3 \neq G_4, G_3 \neq G_6$
 $G_5 \neq G_1, G_5 \neq G_2, G_5 \neq G_4, G_5 \neq G_6$
- 4) $G_3 = G_4 \pm 1, G_5 = G_6 \pm 1, G_5 = G_2 \pm 1$

Istanziando la prima variabile:

$G_1 = 1$
e propago.
 $G_2 :: [1, 2, 3, 4]$
 $G_3 :: [2, 3, 4]$
 $G_4 :: [2]$
 $G_5 :: [2, 3, 4]$
 $G_6 :: [1, 2, 3, 4]$

Istanziando la seconda variabile:

$G_4 = 2$
e propago
 $G_2 :: [4]$
 $G_3 :: [3]$
 $G_5 :: [3, 4]$
 $G_6 :: [4]$

Istanzio la terza variabile:

$$G2=4$$

e propago

$$G3::[3]$$

$$G5::[3]$$

$$G6::[4]$$

Poi gli assegnamenti non causano ulteriori riduzioni nei domini:

$$G3=3$$

$$G5::[3]$$

$$G6::[4]$$

$$G5 = 3$$

$$G6::[4]$$

$$G6 = 4$$

Soluzione

ESERCIZIO

Si hanno 4 insegnanti a, b, c e d che devono tenere complessivamente 10 lezioni. Ogni lezione può essere tenuta da un solo insegnante. Delle 10 lezioni si conoscono l'inizio S_j e la durata $D_j = 2$ ore. Inoltre, si sa che nessun insegnante può tenere due lezioni consecutive e nemmeno lezioni sovrapposte temporalmente. Si rappresenti il problema come problema di soddisfacimento di vincoli definendo le variabili, i domini delle variabili e i vincoli tra queste. Si supponga inoltre che le lezioni abbiano i seguenti inizi:

$$S_1 = 7$$

$$S_2 = 8$$

$$S_3 = 9$$

$$S_4 = 10$$

$$S_5 = 11$$

$$S_6 = 12$$

$$S_7 = 13$$

$$S_8 = 14$$

$$S_9 = 15$$

$$S_{10} = 16$$

e durata di due ore. Si imposti la soluzione del problema utilizzando come tecnica il forward checking.

SOLUZIONE

Variabili: ore di lezione $X_1 \dots X_{10}$

Dominio: insegnanti $[a,b,c,d]$

Le ore di lezione sono caratterizzate da un inizio S_j e da una durata D_j nota.

Vincoli:

Unari: appartenenza delle variabili ai domini

Binari:

- Ogni ora di lezione può essere tenuta da un solo insegnante: ogni variabile viene istanziata con un solo valore.
- Ogni insegnante non può tenere due ore consecutive di lezione:

$$S_i + D_i = S_j \rightarrow X_i \neq X_j$$

- Ogni insegnante non può tenere due ore sovrapposte di lezione:

$$S_i + D_i > S_j \quad \text{and} \quad S_i + D_i \leq S_j + D_j \rightarrow X_i \neq X_j .$$

Soluzione:

Inizialmente $X_1 \dots X_{10} :: [a,b,c,d]$

Istanzio $X_1 = a$

Propago: $X_2 \ X_3 :: [b,c,d]$, $X_4 \dots X_{10} :: [a,b,c,d]$

Istanzio $X_2 = b$

Propago: $X_3 :: [c,d]$, $X_4 :: [a,c,d]$, $X_5 \dots X_{10} :: [a,b,c,d]$

Istanzio $X_3 = c$

Propago: $X_4 :: [a,d]$, $X_5 :: [a,b,d]$, $X_6 \dots X_{10} :: [a,b,c,d]$

Istanzio $X_4 = a$

Propago: $X_5 :: [b,d]$, $X_6 :: [b,c,d]$, $X_7 \dots X_{10} :: [a,b,c,d]$

Istanzio $X_5 = b$

Propago $X_6 :: [c,d]$, $X_7 :: [a,c,d]$, $X_8 \dots X_{10} :: [a,b,c,d]$

Istanzio $X_6 = c$

Propago $X_7 :: [a,d]$, $X_8 :: [a,b,d]$, $X_9 \dots X_{10} :: [a,b,c,d]$

Istanzio $X_7 = a$

Propago $X_8 :: [b,d]$, $X_9 :: [b,c,d]$, $X_{10} :: [a,b,c,d]$

Istanzio $X_8 = b$

Propago $X_9 :: [c,d]$, $X_{10} :: [a,c,d]$

Istanzio $X_9 = c$

Propago $X_{10} :: [a,d]$

Istanzio $X_{10} = a$ soluzione

OSSERVAZIONE

Si noti che esiste un'altra possibile rappresentazione del problema che associa agli insegnanti una variabile il cui dominio contiene inizialmente tutte le lezioni. I vincoli potrebbero eliminare dai domini quelle lezioni che sono incompatibili con i vincoli sull'insegnante. Tuttavia, questa rappresentazione non rientra nell'ottica dei problemi di soddisfacimento di vincoli in quanto in una soluzione deve essere assegnato UNO E UN SOLO valore a ogni variabile e non un insieme di valori come nel caso di questa seconda rappresentazione. Infatti, in quest'ultimo caso una possibile soluzione avrebbe

$X_a::[1,4,7]$

che significa che il maestro a può tenere le lezioni 1, 4 e 7.

ESERCIZIO

Si supponga di avere a disposizione i seguenti vincoli:

$$X < Y, X \neq K, Y + 5 \leq K, Y + 7 > Z, X \leq Z$$

definiti sulle variabili X, Y, Z, K il cui dominio di definizione è $[1..20]$.

Si risolva il problema applicando la strategia di full look ahead.

$$X < Y, X \neq K, Y + 5 \leq K, Y + 7 > Z, X \leq Z$$
$$X, Y, Z, K :: [1..20]$$

Istanzio $X=1$

Propagazione full look ahead: $Y::[2..15], K::[7..20], Z::[1..20]$

Istanzio: $Y=2$

Propagazione: $K::[7..20], Z::[1..8]$

Istanzio: $K=7$

Propagazione: $Z::[1..8]$

Istanzio: $Z = 1$

ESERCIZIO

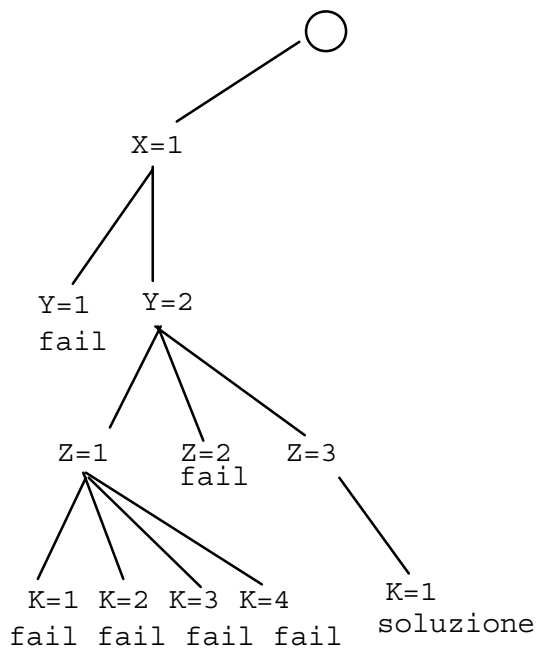
Si consideri il seguente insieme di vincoli.

$$X < Y, Y \neq Z, Z > K$$

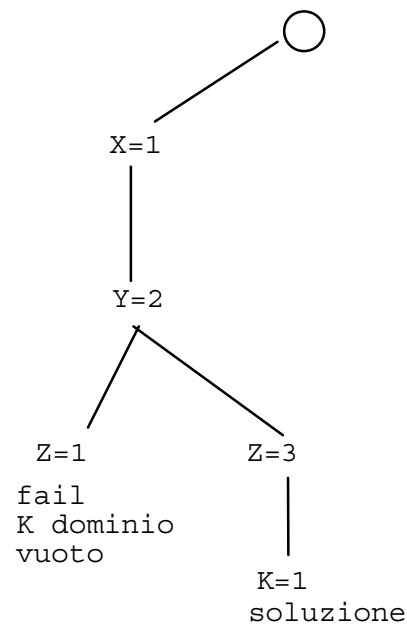
in cui le variabili sono definite su un dominio finito di valori interi [1,2,3,4]

Si mostrino gli alberi di ricerca fino alla prima soluzione relativo alla strategia standard backtracking e quello relativo al forward checking e si commentino i risultati, utilizzando, nella costruzione dell'albero, per le variabili l'ordine X,Y,Z,K.

STANDARD BACKTRACKING

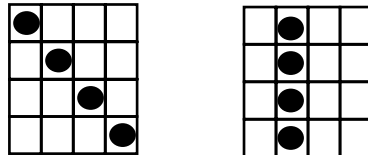


FORWARD CHECKING



ESERCIZIO

Si consideri il problema chiamato *inverse 4-queens*. Si ha a disposizione una scacchiera 4 x 4 e si devono disporre 4 regine in modo tale che ciascuna coppia sia su una stessa colonna o su una stessa diagonale. Ad esempio, due possibili soluzioni al problema sono:



Si discuta la rappresentazione del problema (variabili, domini e vincoli), e si descriva una strategia di ricerca della soluzione di tipo standard backtracking e forward checking.

Variabili: righe della scacchiera X_1, X_2, X_3, X_4

Domini: colonne della scacchiera:

$X_1, X_2, X_3, X_4 :: [1, 2, 3, 4]$

Vincoli: $\forall i, j (X_i = X_j) \text{ or } (|i - j| = |X_i - X_j|)$

Standard backtracking:

$X_1=1$ $X_2=1$ $X_3=1$ $X_4=1$

Forward checking:

$X_1=1$ $X_2::[1,2]$ $X_3::[1,3]$ $X_4::[1,4]$

$X_2=1$ $X_3::[1]$ $X_4::[1]$

Soluzione $X_1=1, X_2=1, X_3=1, X_4=1$

ESERCIZIO

Sono dati 9 numeri interi che devono essere inseriti in una scacchiera 3 X 3 rispettando il vincolo che i numeri su righe dispari siano sempre maggiori di quelli adiacenti inseriti sulle diagonali, mentre quelli su righe pari sempre minori di quelli adiacenti inseriti sulle diagonali.

Inoltre per ogni riga l'elemento adiacente a destra deve essere minore e quello a sinistra maggiore. Si esprima formalmente il problema come CSP e si scrivano i vincoli coinvolti.

Si indichi poi come trovare una soluzione utilizzando la strategia di forward checking e si mostrino i passi eseguiti.

Esempio di soluzione corretta:

9 8 7

3 2 1

6 5 4

NOTA: per semplicità si consideri che l'EURISTICA di selezione dei valori scelga sempre i valori maggiori contenuti nel dominio delle variabili

Variabili $X_{i,j} :: [1..9]$

Vincoli

A) $X_{i,j} > X_{i,j+1}$

B) $X_{i,j} < X_{i,j-1}$ % può essere sottinteso (contrario del primo)

per i dispari

C) $X_{i,j} > X_{i-1,j+1}$

D) $X_{i,j} > X_{i-1,j-1}$

E) $X_{i,j} > X_{i+1,j-1}$

F) $X_{i,j} > X_{i+1,j+1}$

per i pari

G) $X_{i,j} < X_{i-1,j+1}$

H) $X_{i,j} < X_{i-1,j-1}$

I) $X_{i,j} < X_{i+1,j-1}$

L) $X_{i,j} < X_{i+1,j+1}$

Inoltre

M) $X_{i,j} \neq X_{k,l} \forall (i,j) \neq (k,l)$

Inizialmente $X_{1,1}$ $X_{1,2}$ $X_{1,3}$ $X_{2,1}$ $X_{2,2}$ $X_{2,3}$ $X_{3,1}$ $X_{3,2}$ $X_{3,3}$
definiti sul dominio $[1..9]$

Assegno a $X_{1,1}$ il valore 9

$X_{1,2}$ $X_{1,3}$ $X_{2,1}$ $X_{2,2}$ $X_{2,3}$ $X_{3,1}$ $X_{3,2}$ $X_{3,3} :: [1..8]$

Assegno a $X_{1,2}$ il valore 8

$X_{1,3}$ $X_{2,1}$ $X_{2,2}$ $X_{2,3}$ $X_{3,1}$ $X_{3,2}$ $X_{3,3} :: [1..7]$

Assegno a $X_{1,3}$ il valore 7

$X_{2,1}$ $X_{2,2}$ $X_{2,3}$ $X_{3,1}$ $X_{3,2}$ $X_{3,3} :: [1..6]$

Assegno a $X_{2,1}$ il valore 6

$X_{2,2}$ $X_{2,3}$ $X_{3,1} :: [1..5]$

$X_{3,2} :: []$ per il vincolo L

Assegno a $X_{2,1}$ il valore 5

$X_{2,2} :: [1..4]$

$X_{2,3}$ $X_{3,1}$ $X_{3,3} :: [1..4,6]$

$X_{3,2} :: [6]$

Assegno a $X_{2,2}$ il valore 4

$X_{2,3} :: [1..3]$

$X_{3,1}$ $X_{3,2}$ $X_{3,3} :: [6]$

Assegno a $X_{2,3}$ il valore 3

$X_{3,1}$ $X_{3,2}$ $X_{3,3} :: [6]$

Assegno a $X_{3,1}$ il valore 6

Fallisco

La stessa cosa succede se assegno a $X_{2,3}$ i valori 2 e 1

-> $X_{3,1}$ $X_{3,2}$ $X_{3,3} :: [6]$

Assegno a $X_{2,2}$ il valore 3

$X_{2,3} ::= [1, 2]$

$X_{3,1} X_{3,3} ::= [4, 6]$

$X_{3,2} ::= [6]$

Assegno a $X_{2,3}$ il valore 2

$X_{3,1} X_{3,3} ::= [4, 6]$

$X_{3,2} ::= [6]$

Assegno a $X_{3,1}$ il valore 6

$X_{3,2} ::= []$ Fallisco

Assegno a $X_{3,1}$ il valore 4

$X_{3,2} X_{3,3} ::= [6]$

Assegno a $X_{3,2}$ il valore 6

Fallisco

Assegno a $X_{2,2}$ il valore 2

$X_{2,3} ::= [1]$

$X_{3,1} X_{3,3} ::= [3, 4, 6]$

$X_{3,2} ::= [6]$

Assegno a $X_{2,3}$ il valore 1

$X_{3,1} X_{3,3} ::= [3, 4, 6]$

$X_{3,2} ::= [6]$

Assegno a $X_{3,1}$ il valore 6

Fallisco

Assegno a $X_{3,1}$ il valore 4

$X_{3,2} ::= []$

Fallisco causa vincolo A

Assegno a $X_{2,2}$ il valore 1
 $X_{2,3} ::= []$
Fallisco causa vincolo A

Assegno a $X_{2,1}$ il valore 4
 $X_{2,2} ::= [1..3]$
 $X_{2,3} \ X_{3,1} \ X_{3,3} ::= [1..3,5,6]$
 $X_{3,2} ::= [6,5]$

Assegno a $X_{2,2}$ il valore 3
 $X_{2,3} ::= [1,2]$
 $X_{3,1} \ X_{3,2} \ X_{3,3} ::= [5,6]$

Assegno a $X_{2,3}$ il valore 2
 $X_{3,1} \ X_{3,2} \ X_{3,3} ::= [5,6]$

Assegno a $X_{3,1}$ il valore 6
 $X_{3,2} \ X_{3,3} ::= [5]$

Assegno a $X_{3,2}$ il valore 5
Fallisco

Stessa cosa se assegno a $X_{3,1}$ il valore 5

Assegno a $X_{2,2}$ il valore 2
 $X_{2,3} ::= [1]$
 $X_{3,2} ::= [5,6]$
 $X_{3,1} \ X_{3,3} ::= [3,5,6]$

Assegno a $X_{2,3}$ il valore 1
 $X_{3,1} \ X_{3,3} ::= [3,5,6]$
 $X_{3,2} ::= [5,6]$

Assegno a $X_{3,1}$ il valore 6

$X_{3,2}::[5]$

$X_{3,3}::[3,5]$

Assegno a $X_{3,2}$ il valore 5

$X_{3,3}::[3]$

Soluzione: $X_{1,1} = 9$ $X_{1,2} = 8$ $X_{1,3} = 7$ $X_{2,1} = 4$ $X_{2,2} = 2$ $X_{2,3} = 1$
 $X_{3,1} = 6$ $X_{3,2} = 5$ $X_{3,3} = 3$

ESERCIZIO

Data una scacchiera 4 x 4 e 4 colori [r,b,g,y], si deve collocare un colore in ciascuna cella della scacchiera modo che ogni riga, ogni colonna e le due diagonali principali della scacchiera contengano colori diversi.

Si formalizzi il problema come CSP, e lo si risolva fino alla prima soluzione tramite la tecnica del forward checking con euristica first-fail (MRV)

SOLUZIONE

Si considera una scacchiera 4 x 4. Ogni cella della scacchiera rappresenta una variabile da X_{11} a X_{44} . I domini iniziali delle variabili sono composti dai quattro colori a disposizione.

I vincoli sono

per ogni i $X_{ij} \neq X_{ik}$ per $j \neq k$

per ogni i $X_{ji} \neq X_{ki}$ per $j \neq k$

per ogni i e j $X_{ii} \neq X_{jj}$ con $i \neq j$

per ogni i e j $X_{i,4-i+1} \neq X_{j,4-j+1}$ con $i \neq j$

Rappresentiamo i domini all'interno delle celle della scacchiera. Con il forward checking si arriva ad una soluzione senza mai fallire.

r,b	r,b	r,b	r,b
y,g	y,g	y,g	y,g
r,b	r,b	r,b	r,b
y,g	y,g	y,g	y,g
r,b	r,b	r,b	r,b
y,g	y,g	y,g	y,g
r,b	r,b	r,b	r,b
y,g	y,g	y,g	y,g

X11=r

r	b	b	b
y,g	y,g	y,g	y,g
b	b	r,b	r,b
y,g	y,g	y,g	y,g
b	r,b	b	r,b
y,g	y,g	y,g	y,g
b	r,b	r,b	b
y,g	y,g	y,g	y,g

X12=b

r	b	y,g	y,g
b		r,b	r,b
y,g	y,g	y,g	y,g
b	r	b	r,b
y,g	y,g	y,g	y,g
b	r	r,b	b
y,g	y,g	y,g	y,g

X13=y

r	b	y	g
b		r,b	r,b
y,g	y,g	g	y,g
b	r	b	r,b
y,g	y,g	g	y,g
b	r	r,b	b
y,g	y,g	g	y,g

X14=g

r	b	y	g
b		r,b	r,b
y,g	y,g	r,b	y
b	r	b	r,b
y,g	y	g	y
b	r	r,b	b
y	y,g	g	y

X22=y

r	b	y	g
b		r,b	r,b
g	y	r,b	r,b
b	r	b	r,b
y,g		g	y
b	r	r,b	b
y	g	g	b

X32=r

r	b	y	g
b		b	r,b
g	y	b	r,b
b	r	b	b
y,g		g	y
b	g	r,b	b
y		g	b

X23=b

r	b	y	g
g	y	b	r
b	r	g	b
y,g			y
y	g	r	b
		g	b

X24=r

r	b	y	g
g	y	b	r
b	r	g	b
y,g			y
y	g	rg	b

X21=g

r	b	y	g
g	y	b	r
b	r	g	b
y			y
y	g	r	b
		g	b

X33=g

r	b	y	g
g	y	b	r
b	r	g	b
y			y
y	g	r	b

X41=y

r	b	y	g
g	y	b	r
b	r	g	b
y			y
y	g	r	b

X31=b

r	b	y	g
g	y	b	r
b	r	g	y
y	g	r	b

Più quattro stati identici all'ultimo relativi alle istanziazioni di X34 a y, X42 a g, X43 a r e X44 a b. SOLUZIONE CONSISTENTE

ESERCIZIO

Si consideri il seguente problema di soddisfacimento di vincoli.

Si devono visitare 6 clienti A, B, C, D, E, F nell'arco della giornata lavorativa (dalle 9 alle 19).

- Due clienti non possono essere visitati contemporaneamente.
- Si sa che i clienti C ed F devono essere visitati prima del cliente D.
- A è un cliente fuori città, mentre B, C, D, E ed F sono tutti in centro. Quindi, per spostarsi da A a ogni altro cliente si impiegano due ore, mentre qualunque spostamento in centro città viene effettuato a tempo trascurabile ($=0$).
- Il cliente C può essere visitato solo dalle 15 alle 17.

Si modelli il problema in termini di variabili e vincoli. Si mostri l'albero di ricerca fino alla prima soluzione relativo alla strategia standard backtracking e quello relativo al forward checking e si commentino i risultati.

SOLUZIONE

Variabili: clienti

Domini: possibili orari di visita A, B, C, D, E, F::[9..18]

Vincoli:

- Due clienti non possono essere visitati contemporaneamente:
per $\forall X, Y \quad X \neq Y$
- I clienti C ed F devono essere visitati prima del cliente D:
 $C < D \quad F < D$
- A è un cliente fuori città, mentre B, C, D, E e F sono tutti in centro. Quindi, per spostarsi da A ad ogni altro cliente si impiegano due ore, mentre qualunque spostamento in centro città viene effettuato a tempo trascurabile (=0).

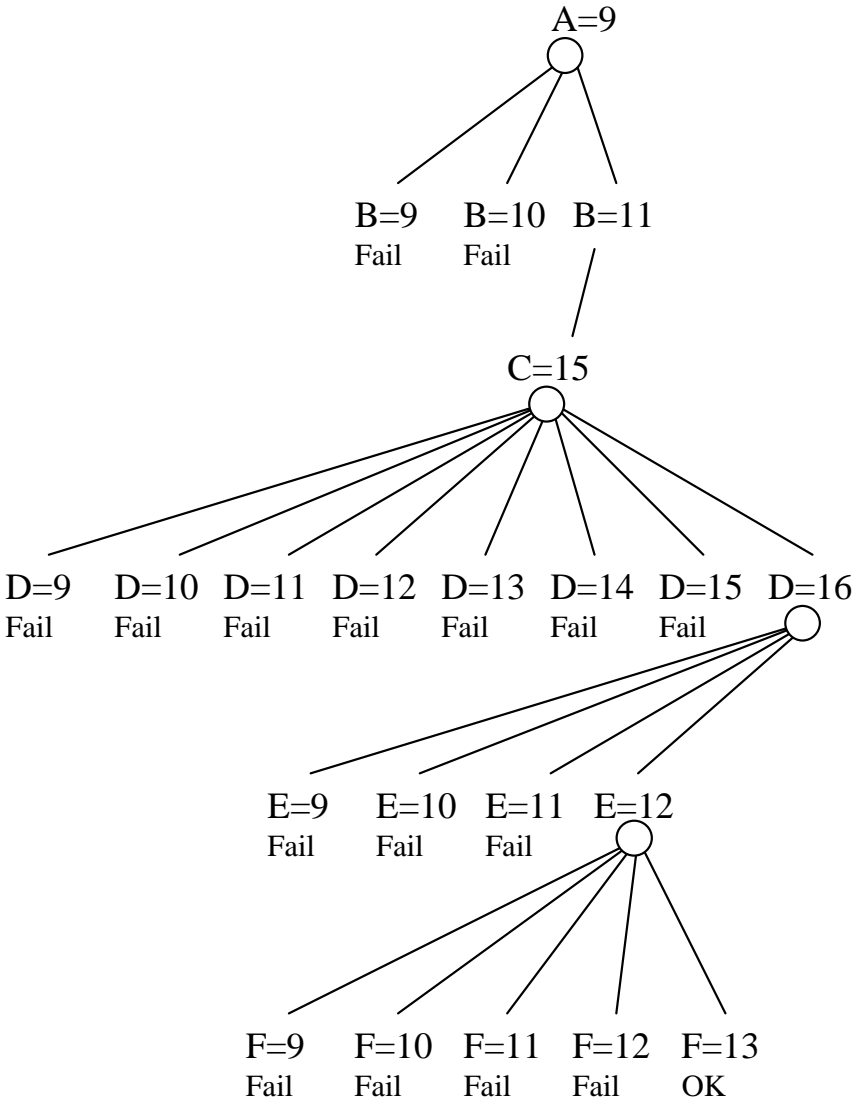
$\forall X \in [B, C, D, E, F] \quad A \geq X + 2 \quad \text{OR} \quad X \geq A + 2$

- Il cliente C può essere visitato solo dalle 15 alle 17.

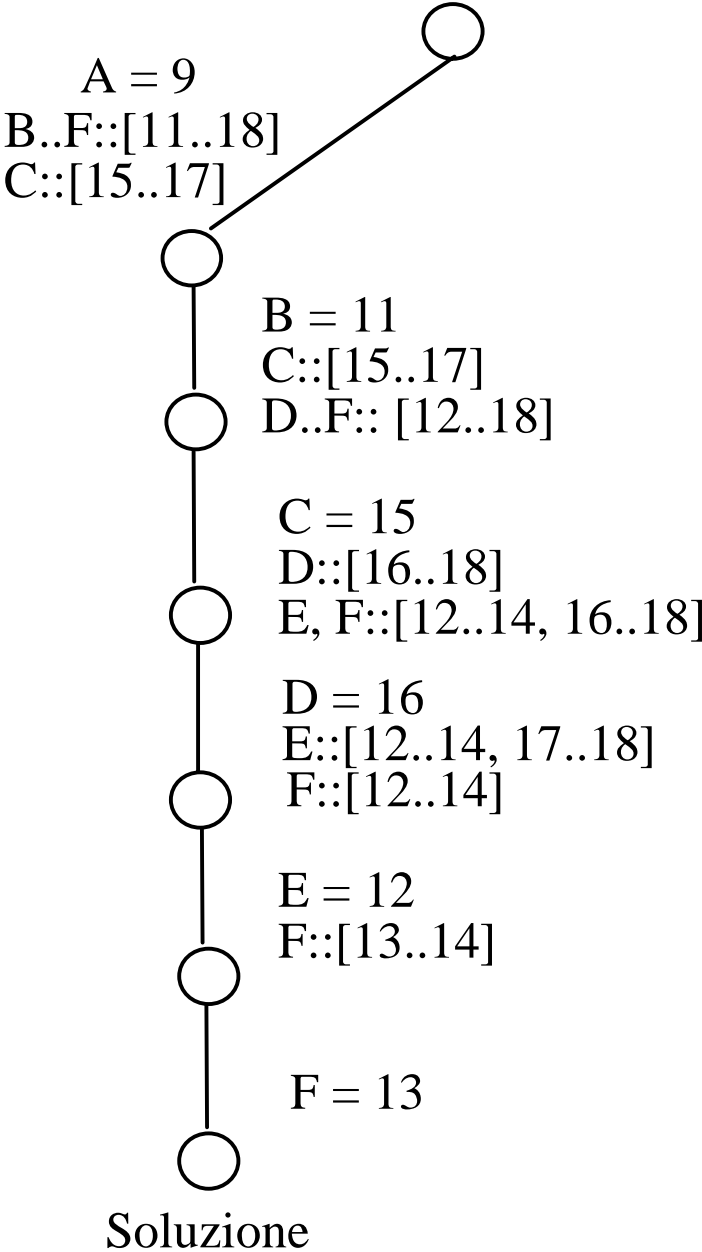
Vincolo unario su C che riduce il suo dominio

$$C \geq 15 \quad \text{e} \quad C \leq 17$$

Standard backtracking

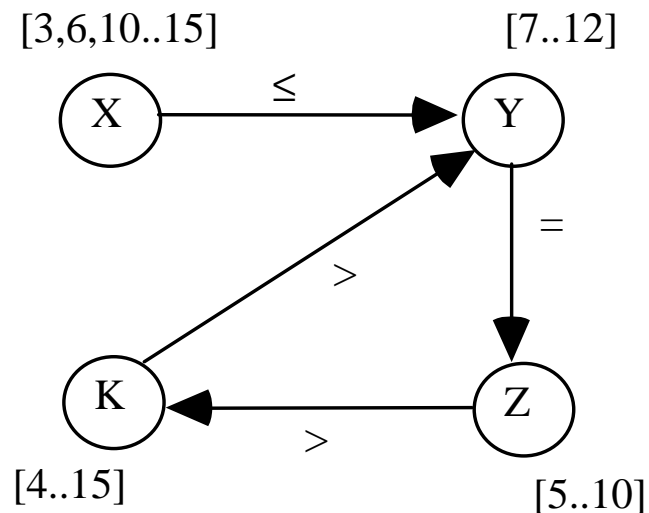


Forward Checking



ESERCIZIO

Si consideri la seguente rete di vincoli:



$X :: [3, 6, 10..15]$, $Y :: [7..12]$, $K :: [4..15]$,
 $Z :: [5..10]$, $Y = Z$, $Z < K$, $K > Y$, $X \leq Y$

E si applichi l'arc-consistenza. Si discuta inoltre cosa accade applicando l'arc-consistenza alla stessa rete se si introduce un vincolo ulteriore $X = K$.

Risultato dell'arc-consistency

$X = [3, 6, 10]$

$Z = [7..10]$

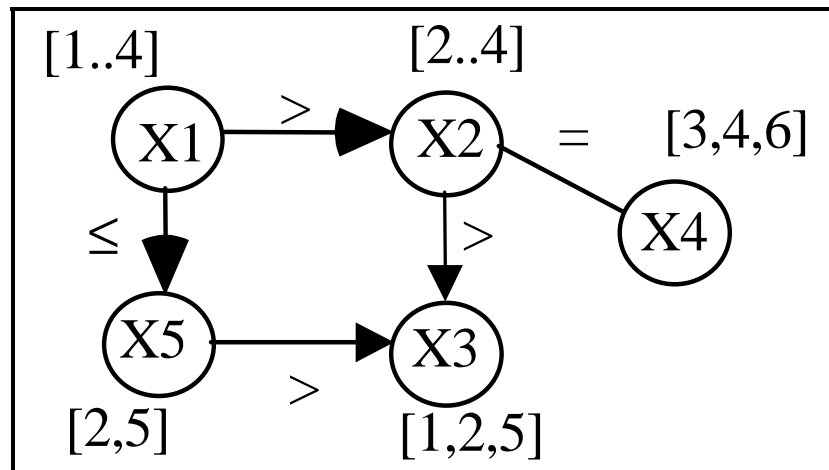
$K = [8..15]$

$Y = [7..10]$

Introducendo il nuovo vincolo si riporta fallimento.

ESERCIZIO

Si consideri il problema a vincoli rappresentato tramite il seguente grafo:



$X1 :: [1..4]$, $X2 :: [2..4]$, $X4 :: [3, 4, 6]$,
 $X3 :: [1, 2, 5]$, $X5 :: [2, 5]$, $X1 > X2$, $X2 = X4$,
 $X1 \leq X5$, $X3 < X5$, $X2 > X3$,

Si applichi l'arc-consistenza alla rete.

I domini delle variabili ridotti al termine dell'arc-consistenza sono:

$$X4 = 3$$

$$X2 = 3$$

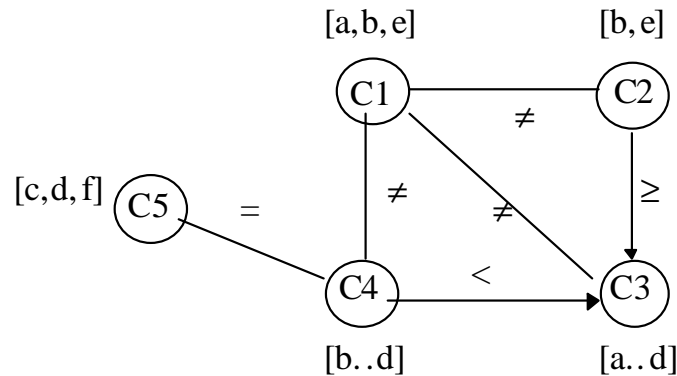
$$X3 :: [1, 2]$$

$$X1 = 4$$

$$X5 = 5$$

ESERCIZIO

Si consideri il problema a vincoli rappresentato tramite il seguente grafo:



dove $C1 \neq C2$, $C2 \geq C3$, $C3 \neq C1$, $C1 \neq C4$, $C4 < C3$, $C5 = C4$, e i domini delle variabili sono rappresentati tramite notazione a intervallo $[A..B]$ che denota tutti le lettere comprese tra A e B nell'ordine alfabetico, oppure la notazione enumerativa. Si applichi l'arc-consistenza alla rete, e se ne indichino i risultati.

SOLUZIONE

I domini delle variabili ridotti al termine dell'arc-consistenza sono:

$C1 :: [a, b]$

$C2 = e$

$C3 = d$

$C4 = c$

$C5 = c$

Esercizio

Dati i seguenti vincoli:

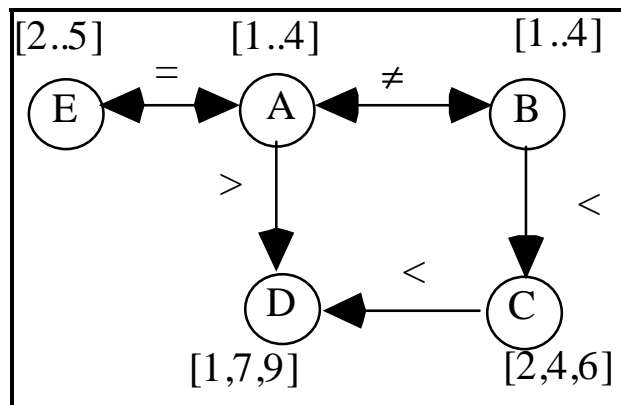
$A :: [1..4]$, $B :: [1..4]$, $C :: [2, 4, 6]$,

$D :: [1, 7, 9]$, $E :: [2..5]$,

$A > D$, $A \neq B$, $B < C$, $C > D$, $E = A$

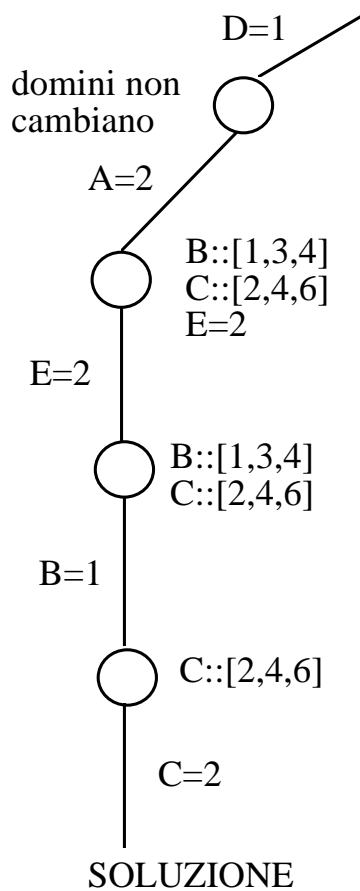
Si disegni il grafo corrispondente al problema di soddisfacimento di vincoli e si applichi l'arc-consistenza. Si disegni l'albero per arrivare alla prima soluzione usando come euristica di assegnamento di valori alle variabili il first-fail e ad ogni istanziazione si riapplichino l'arc-consistenza alla rete residua.

Grafo corrispondente al CSP



Dopo l'applicazione dell'arc-consistenza al problema originale si ottiene

A::[2..4]
B ::[1..4]
C ::[2, 4, 6]
D = 1
E ::[2..4]



A = 2, B = 1, C = 2, D = 1, E = 2