CERCA PER A SATISFACCCIÓ DE RESTRICCIONS
Resolució de problemes de presa de decisió
per exploració d'alternatives

Exemple: Assignar valors a les següents lletres, de manera que es verifiqui la següent suma:

$$\begin{array}{cccccc} & & S & E & N & D \\ + & & M & O & R & E \\ \hline & M & O & N & E & Y \end{array}$$

s'han de trobar valors del conjunt: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

que assignats a les variables: {D, E, Y, N, R, O, S, M}

Verifiquin la suma donada

Solució: {D=4, E=5, Y=9, N=5, R=0, O=0, S=9, M=1}

Problema de satisfacció de restriccions: és un problema en el que a partir d'un conjunt de variables:

$$\text{Variables} = \{X_1, X_2, \cdots, X_n\}$$

i a partir d'un domini de valors que poden prendre aquestes variables:

$$\text{Domini} = \{v_1, v_2, \cdots, v_k\}$$

es vol trobar un conjunt d'assignacions de totes les variables amb valors del domini que verifiquin un conjunt de

Restriccions del problema

Arbre de satisfacció de restriccions: és una representació de coneixement que és un arbre semàntic amb les següents particularitats:

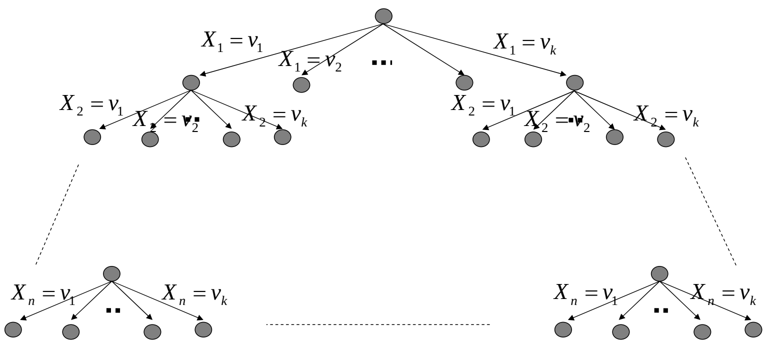
Lèxic: restriccions, conjunt de variables, variables assignades i no-assignades, domini de les variables,

Estructural:
b=#Domini de les variables (factor d'expansió)
d=#Conjunt de variables (profunditat de l'arbre)

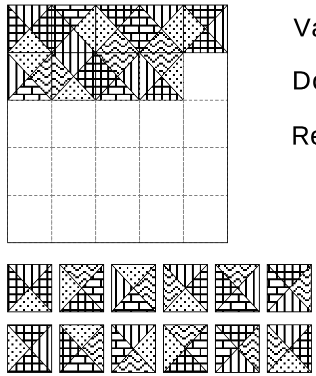
Semàntica: Node= Comprovació de restriccions sobre les variables assignades.
Branca= Assignació d'un valor del domini a una variable.

Procedimental:
- Proc. Backtracking.
- Proc. Backtracking amb forward checking.
- Proc. comproven les restriccions sobre un conjunt de variables assignades.

Arbre de satisfacció de restriccions: b=k, prof.=n



Exercici: Descriviu el problema de resoldre un puzzle com un problema de satisfacció de restriccions.



$\text{Variables} = \{X_1, X_2, \cdots, X_n\} = ?$

$\text{Domini} = \{v_1, v_2, \cdots, v_k\} = ?$

Restriccions ?

Exercici: Descriviu el problema de la construcció d'un quadrat màgic, com un problema de satisfacció de restriccions

6	1	8	$\sum = 15$
7	5	3	$\sum = 15$
2	9	4	$\sum = 15$

$\sum = 15$

1	14	14	4	$\sum = 33$
11	7	6	9	$\sum = 33$
8	10	10	5	
13	2	3	15	

$\sum = 33$ $\sum = 33$

$\text{Variables} = \{X_1, X_2, \cdots, X_n\} = ?$

$\text{Domini} = \{v_1, v_2, \cdots, v_k\} = ?$

Restriccions ?

Funcio Backtracking(LVA,LVNA,R,D)
1. Si (LVNA és buida) llavors Retornar(LVA) fSi
2. Var=Cap(LVNA);
3. Per a cada (valor del Domini(D) que podem assignar a Var) fer
   - Si (SatisfaRestriccions('(Var valor),LVA,R)) llavors
     - Res=Backtracking(Insertar('(Var valor),LVA),Cua(LVNA),R,D);
     - Si (Res és una solució completa) llavors
       - Retornar(Res);
     - Fsi
   - Fsi
4. Fper
5. Retornar(Falla)

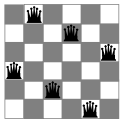
FFuncio

SatisfaRestriccions(A,LVA,R): Retorna cert si l'assignació A afegida la llista d'assignacions LVA satisfan les restriccions R.
Domini(V,D): Retorna un valor del domini D per a la variable V.
Insertar(e,L):Retorna la llista resultant d'afegir e al principi de L.
Cua(L) i Cap(L): Retornen la cua i el cap de L, respectivament.

Exemple: Descripció del problema de les N reines sobre una taulell de NxN com un problema de satisfacció de restriccions.

Exemple del cas N=6

$X_1 = 2$
$X_2 = 4$
$X_3 = 6$
$X_4 = 1$
$X_5 = 3$
$X_6 = 5$



$$\text{Variables} = \{X_1, X_2, \cdots, X_N\} = V$$

X_i : Representa la reina de la fila i

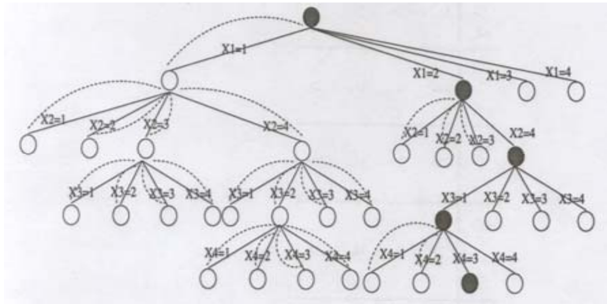
$$\text{Domini} = \{1,...,N\} = D$$

k : Columna on es posa la reina

Restriccions:

$$\forall X_i, X_j \in V : j > i \Rightarrow \begin{cases} X_i \neq X_j \\ X_j \neq X_i + (j - i) \\ X_j \neq X_i - (j - i) \end{cases}$$

L'arbre que generarà el *backtracking* pel problema de les 4 reines



En Prolog l'algorisme de backtracking ja ve donat, aleshores:

```
problema(LVNA,LVA,Restriccions,Domini,Resultat):
       LVNA: Llista de variables no assignades.
       LVA: Llista de variables assignades.
       Domini: Nom del predicat que dona valors a les variables del
          domini.
       Resultat: Llista de totes les variables assignades amb els
          valors que satisfan totes les restriccions.
Domini(Var,Valor): Var es la variable que volem assignar i Valor
   s'unifica amb el valor del domini que li podem donar.
Restriccions(LVA): Retorna cert si la llista d'assignacions de LVA
   satisfan totes les restriccions del problema.
```

```
problemacsp([X|Y],LVA,R,D,Res):- Domini=..[D,X,K],
                           call(Domini),
                           Restriccions=..[R,[[X,K]|LVA]],
                           call(Restriccions);
                           problemacsp(Y,[[X,K]|LVA],R,D,Res).
problemacsp([],Res,R,D,Res).
```

<u>**Funcio**</u> **BackForwardChecking**(LVA,LVNA,R,DA)
1. <u>Si</u> (LVNA és buida) llavors Retornar(LVA) <u>fSi</u>
2. Var=Cap(LVNA);
3. <u>Per a cada</u> (valor del Domini(DA) que podem assignar a Var) <u>fer</u>
   - <u>Si</u> (SatisfaRestriccions('(Var valor),LVA,R)) i
     (DA=ActualitzarDominis(('(Var valor), LVNA,R)<>fals) <u>llavors</u>
     - R= **BackForwardChecking** (Insertar('(Var valor),LVA),Cua(LVNA),R,DA);
     - <u>Si</u> (R és una solució completa) <u>llavors</u>
       - Retornar(Resultat);
     - <u>Fsi</u>
   - <u>Fsi</u>
4. <u>Fper</u>
5. Retornar(Falla)

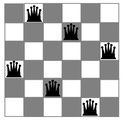
<u>**FFuncio**</u>

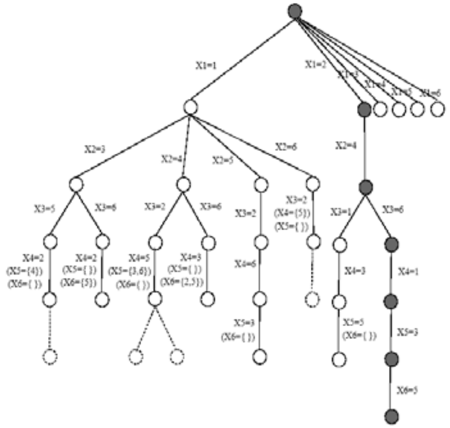
ActualitzarDominis('(X v),L,R): Retorna la llista dels dominis per a les variables no assignades de L considerant les restriccions de R despres d'assignar X amb v, retorna fals si algun domini actualitzat és buit.

Exemple: Aplicació del Backtracking amb *forward-checking* al problema de les 6 reines.

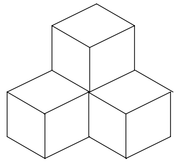
Exemple del cas N=6

$X_1 = 2$
$X_2 = 4$
$X_3 = 6$
$X_4 = 1$
$X_5 = 3$
$X_6 = 5$

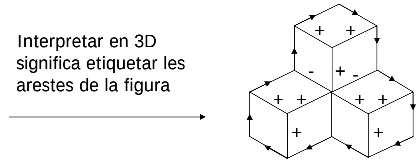




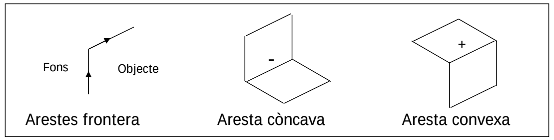
Exemple: Descripció del problema d'interpretar tridimensionalment les imatges dibuixades amb traços sobre un paper, com un problema de satisfacció de restriccions



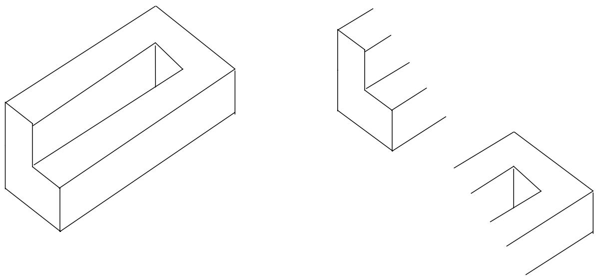
Interpretar en 3D significa etiquetar les arestes de la figura



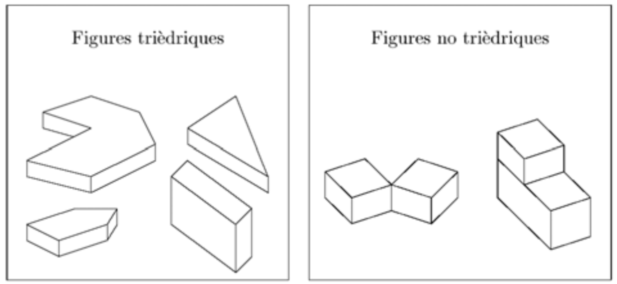
Interpretació de les arestes



Cas particular: L'algorisme hauria de permetre trobar les interpretacions impossibles, tal com ara

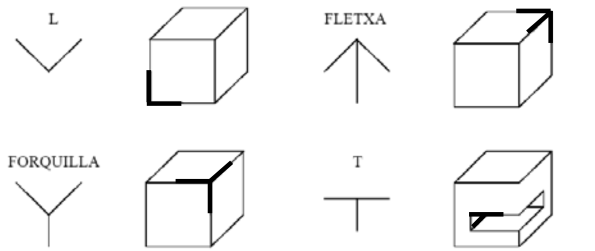


<u>Assumpció:</u> per simplificar suposarem que totes les figures són tetraèdriques, això és, tot vèrtex és la intersecció de 3 cares.

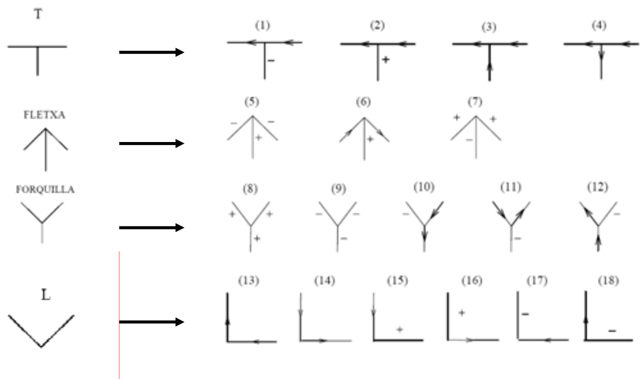


Per poder tractar el problema com a satisfacció de restriccions necessitem conèixer totes les **maneres possibles d'etiquetar les arestes d'un vèrtex en 3D**.

Haurem de tenir en compte que hi ha 4 tipus de vertex 2D:

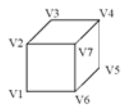


Despres d'un estudi ... només existeixen 18 maneres diferents d'etiquetar vèrtexs 3D:



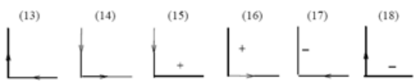
Per tant, el problema de l'etiquetatge es pot veure com un problema de satisfacció de restriccions en el que:

Variables = { Conjunt de vèrtex de la figura }, Exemple:

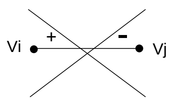


Domini = { Possibles maneres d'etiquetar cada vèrtex de la figura en funció del seu tipus }

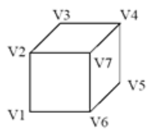
Exemple: V1, és de tipus L, llavors el Domini d'aquest vèrtex són totes les maneres d'etiquetar-lo:



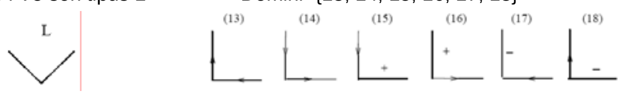
Restriccions = Una aresta només es pot etiquetar d'una manera a partir dels seus dos vèrtexs.



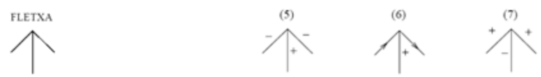
Exemple:



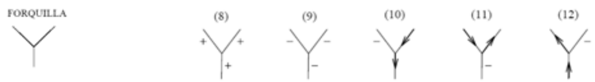
- V1,V3 i V5 són tipus L ⟶ Domini={13, 14, 15, 16, 17, 18}



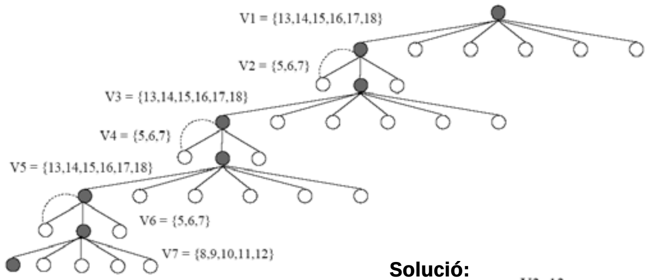
- V2, V4 i V6 són tipus fletxa ⟶ Domini={5, 6, 7}



- V7 és tipus forquilla ⟶ Domini={8, 9, 10, 11, 12}



Arbre de satisfacció de restriccións:



Solució: