

UNIVERSITÉ LIBRE DE BRUXELLES  
FACULTÉ DES SCIENCES  
DÉPARTEMENT D'INFORMATIQUE

INFO-F-302 - Logique Informatique  
Projet : Le jeu Pattern et Utilisation de  
MiniSAT

OOMS Aurélien, SONNET Jean-Baptiste

# Table des matières

<b>1</b>	<b>Énumération</b>	<b>3,2</b>	<b>2</b>
1.1	Problème et notation . . . . .		2
1.1.1	Grille . . . . .		2
1.1.2	Cases . . . . .		2
1.2	Parcours . . . . .		3

# Chapitre 1

## Énumération 3,2

### 1.1 Problème et notation

#### 1.1.1 Grille

Le problème est présenté sous forme d'une grille  $3 \times 3$  contenant au *max* 2 contraintes par ligne ou colonne.

Soit une matrice  $3 \times 3$ ,

$$\begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{1,0} & x_{1,1} & x_{1,2} \\ x_{2,0} & x_{2,1} & x_{2,2} \end{bmatrix}$$

où chacune des cases  $x_{i,j}$  prendra potentiellement une des 3 valeurs :  $\{0, 1, -1\}$ , respectivement l'inconnu, le noir, le blanc.

On aura par exemple comme problème à résoudre :

$$\begin{array}{ccc} & 2 & 1 & 2 \\ \begin{array}{c} 1 \\ 1 \ 1 \\ 2 \end{array} & \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{array}{|c|c|c|} \hline \blacksquare & \square & ? \\ \hline ? & ? & ? \\ \hline ? & ? & ? \\ \hline \end{array} \end{array}$$

Selon les contraintes précisées, la solution devra donner pour toutes les cases inconnues une valeur de 1 ou  $-1$  :

$$\begin{array}{ccc} & 2 & 1 & 2 \\ \begin{array}{c} 1 \\ 1 \ 1 \\ 2 \end{array} & \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{bmatrix} & \begin{array}{|c|c|c|} \hline \blacksquare & \square & \square \\ \hline \blacksquare & \square & \blacksquare \\ \hline \square & \blacksquare & \blacksquare \\ \hline \end{array} \end{array}$$

#### 1.1.2 Cases

Chaque case est représentée par une variable  $x$ , de sorte que la case à la  $i^{eme}$  ligne et à la  $j^{eme}$  colonne se note  $x_{i,j}$

## 1.2 Parcours

On va parcourir la grille et énumérer les combinaisons possibles premièrement suivant les contraintes de lignes ensuite selon les contraintes de colonnes.

Par ligne (respectivement colonne), on crée si nécessaire une clause représentant les cas possibles quant aux cases dont on ne connaît pas encore la nature.

Pour chaque case contenue dans une ligne donnée :

1. Si elle contient une information, on en crée une clause à part entière qui sera jointe aux autres.
2. Si elle ne contient pas d'informations et conditionnellement à la contrainte, on énumère les possibilités sous forme disjonctive (clause).

On effectue de même par colonne et relativement aux contraintes de la colonne.

L'énumération des cas possibles se formule naturellement à l'aide de la disjonction exclusive (XOR ou  $\oplus$ ), à chaque étape, il convient de la traduire en forme normale conjonctive (FNC).

$$a \oplus b \equiv (a \wedge \neg b) \vee (\neg a \wedge b) \equiv (a \vee b) \wedge \neg(a \wedge b)$$

$a$	$b$	$a \oplus b$	$(a \wedge \neg b) \vee (\neg a \wedge b)$	$(a \vee b) \wedge \neg(a \wedge b)$
0	0	0	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	0	0

---

**Algorithm 1** Énumération selon les contraintes de lignes et de colonnes

---

```
for ligne  $i \rightarrow n$  do
  for colonne  $j \rightarrow n$  do
    if  $x_{i,j} == -1$  then
      Créer une nouvelle clause avec  $-x_{i,j}$ 
    end if
    if  $x_{i,j} == 1$  then
      Créer une nouvelle clause avec  $x_{i,j}$ 
    end if
    if  $x_{i,j} == 0$  then
      if  $\nexists clause_i$  then
        Créer  $clause_i$ 
      end if
      Ajouter  $(\vee) x_{i,j}$  à la  $clause_i$ 
    end if
    Joindre  $(\wedge)$  les clauses
  end for
  if  $\exists clause_i$  then
    Joindre  $(\wedge)$  la  $clause_i$ 
  end if
end for
for colonne  $j \rightarrow n$  do
  for ligne  $i \rightarrow n$  do
    if  $x_{i,j} == -1$  then
      Créer une nouvelle clause avec  $-x_{i,j}$ 
    end if
    if  $x_{i,j} == 1$  then
      Créer une nouvelle clause avec  $x_{i,j}$ 
    end if
    if  $x_{i,j} == 0$  then
      if  $\nexists clause_j$  then
        Créer  $clause_j$ 
      end if
      Ajouter  $(\vee) x_{i,j}$  à la  $clause_j$ 
    end if
    Joindre  $(\wedge)$  les clauses
  end for
  if  $\exists clause_j$  then
    Joindre  $(\wedge)$  la  $clause_j$ 
  end if
end for
```

---