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

Année académique 2012 - 2013

Table des matières

1 Énumération 3,2 **2**

1.1 Problème et notation 2

1.1.1 Grille 2

1.1.2 Variables 3

1.1.3 Contraintes 3

1.1.4 Codage 4

1.2 Parcours et représentation par énumération 4

2 Les Bandes **7**

2.1 Problème et notation 7

2.1.1 Variables 8

2.1.2 Contraintes 8

2.1.3 Codage 8

2.2 Représentation par bandes 8

2.2.1 Observation : position minimale, maximale et sous-espace 8

2.2.2 Énumération récursive 9

Chapitre 1

Énumération 3,2

1.1 Problème et notation

1.1.1 Grille

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

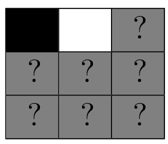
Soit une matrice 3×3,

$$\begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{1,0} & x_{1,1} & x_{1,2} \\ x_{2,0} & x_{2,1} & x_{2,2} \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

où chacune des cases $x_{i,j}$ prendra potentiellement une des 2 valeurs : $\{-1, 1\}$, respectivement le blanc, le noir. Les cases sont initialisées avec la valeur 0.

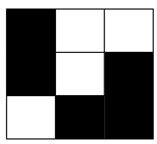
On aura par exemple comme problème à résoudre :

$$\begin{array}{cc} & \begin{array}{ccc} 2 & 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 1\ 1 \\ 2 \end{array} & \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$



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

$$\begin{array}{cc} & \begin{array}{ccc} 2 & 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 1\ 1 \\ 2 \end{array} & \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \end{array}$$



1.1.2 Variables

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 $x_{i,j}$.

On a que $(i, j) \in \{1, 2, 3\}^2$ et que chaque variable peut prendre comme valeur $v \in \{-1, 0, 1\}$ soit :

$$X = \{x_{i,j} | (i, j) \in \{0, 1, 2\}^2, v \in \{-1, 0, 1\}\}$$

On a pour sémantique :
- $x_{i,j}$ est **vrai** *si et seulement si* la case (i, j) a pour valeur $v = 1$
- $x_{i,j}$ est **faux** *si et seulement si* la case (i, j) a pour valeur $v = -1$

1.1.3 Contraintes

1. (trivial) **Pour chaque** ligne, s'il existe une bande, **toutes** les cases ne sont pas fausses.

$$\bigwedge_{i \in \{0,1,2\}} \neg \left(\bigwedge_{j \in \{0,1,2\}} \neg x_{i,j} \right)$$

soit,

$$\bigwedge_{i \in \{0,1,2\}} \left(\bigvee_{j \in \{0,1,2\}} x_{i,j} \right)$$

idem pour chaque colonne.

$$\bigwedge_{j \in \{0,1,2\}} \left(\bigvee_{i \in \{0,1,2\}} x_{i,j} \right)$$

2. (trivial) **Pour toutes** lignes, **pour toutes** colonnes, **pour toutes** bandes, **il existe** une case qui satisfait à la fois une bande de la ligne et une de la colonne.

$$\bigwedge_{i \in \{0,1,2\}} \bigwedge_{j \in \{0,1,2\}} \bigwedge_{b_i^{(k)}, k \in \{1,2\}} \neg (\neg x_{i,j} \wedge x_{i,j})$$

soit,

$$\bigwedge_{i \in \{0,1,2\}} \bigwedge_{j \in \{0,1,2\}} \bigwedge_{b_i^{(k)}, k \in \{1,2\}} (x_{i,j} \vee \neg x_{i,j})$$

3. **Pour toutes** lignes i, **pour toutes** bandes $b_i^{(k)}, k \in \{1, 2\}$ associées à la ligne, **il existe** *autant* de cases vraies $x_{i,j}$ que ne l'exprime la taille de la bande.
   Cela donne au cas par cas :
   (a) Une seule bande :

i. Pour toutes lignes i où $b = 3$,

$$\bigwedge_{j \in \{0,1,2\}} \left(\bigvee_{l=1}^{n=1} x_{i,j} \right)$$

$\top$	$\top$	$\top$

ii. Pour toutes lignes i où $b = 2$,

$$(\neg x_{i,0} \vee \neg x_{i,2}) \wedge (x_{i,0} \vee x_{i,2}) \wedge x_{i,1}$$

$\boxed{\top \mid \top \mid \bot}$ ou $\boxed{\bot \mid \top \mid \top}$

iii. Pour toutes lignes i où $b = 1$,

$$\bigwedge_{\substack{j,j' \in \{0,1,2\} \\ j<j', j'' \neq j,j'}} (\neg x_{i,j} \vee \neg x_{i,j'} \vee x_{i,j''})$$

$\boxed{\top \mid \bot \mid \bot}$ ou $\boxed{\bot \mid \top \mid \bot}$ ou $\boxed{\bot \mid \bot \mid \top}$

iv. Pour toutes lignes i où $b = 0$,

$$\bigwedge_{j \in \{0,1,2\}} \left(\bigvee_{l=1}^{n=1} \neg x_{i,j} \right)$$

$\bot$	$\bot$	$\bot$

(b) Deux bandes, pour toutes lignes i :

$$\left(\bigvee_{l=1}^{n=1} x_{i,0} \right) \wedge \left(\bigvee_{l=1}^{n=1} \neg x_{i,1} \right) \wedge \left(\bigvee_{l=1}^{n=1} x_{i,2} \right)$$

$\top$	$\bot$	$\top$

Idem pour toutes colonnes.

1.1.4 Codage

Pour miniSAT, il faut coder les propositions par des entiers.
La taille d'une grille est de 3×3, on codera en base 3.
Pour (i,j,v) on aura le codage : $(i \times n^2) + (j \times n) + (v + 1)$

1.2 Parcours et représentation par énumération

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 connait pas encore la nature.

Pour chaque case contenue dans une ligne i donnée :

1. Si elle contient une information (1 ou -1), on en créer une clause à part entière qui sera jointe ($\wedge$) à toutes les autres.
2. Si elle ne contient pas d'informations (0) et conditionnellement au contraintes $c_i^{(k)}, k \in \{1, 2\}$, on énumère les possibilités sous forme disjonctive.

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

Algorithm 1 Énumération selon la valeur de(s) bande(s) de chaque lignes

for ligne $i \to n$ **do**
    for colonne $j \to 2$ **do** ▷ encodage de l'information connue
        if $x_{i,j} == \perp$ **then**
            Créer une nouvelle *clause* avec $-x_{i,j}$
        else if $x_{i,j} == \top$ **then**
            Créer une nouvelle *clause* avec $x_{i,j}$
        end if
    end for
    for colonne $j \to 2$ **do** ▷ application de la contrainte 1
        if $\nexists\, clause_i$ **then**
            Créer $clause_i$
        end if
        Ajouter $x_{i,j}$ à $clause_i$
    end for
    if nombre de *bandes* $== 1$ **then**
        if $bande_i{==}0$ **then** ▷ si la taille de la bande est 0
            for colonne $j \to 2$ **do**
                Créer une nouvelle *clause* avec $\neg x_{i,j}$
            end for
        else if $bande_i{==}1$ **then** ▷ si la taille de la bande est 1
            $J = \{0, 1, 2\}$
            for colonne $j'' \to 2$ **do**
                $j = min(J \setminus j'')$
                $j' = max(J \setminus j'')$
                Créer une nouvelle *clause* avec $(\neg x_{i,j} \vee \neg x_{i,j'} \vee x_{i,j''})$
            end for
        else if $bande_i{==}2$ **then** ▷ si la taille de la bande est 2
            Joindre la FNC $(\neg x_{i,0} \vee \neg x_{i,2}) \wedge (x_{i,0} \vee x_{i,2}) \wedge x_{i,1}$
        else if $bande_i{==}3$ **then** ▷ si la taille de la bande est 3
            for colonne $j \to 2$ **do**
                Créer une nouvelle *clause* avec $x_{i,j}$
            end for
        end if
    else if nombre de *bandes* $== 2$ **then**
        Créer une nouvelle *clause* avec $(x_{i,0} \vee \neg x_{i,1} \vee x_{i,2})$
    end if
    Joindre $(\wedge)$ toutes les *clauses*
end for

Chapitre 2

Les Bandes

2.1 Problème et notation

Une suite de cases noires représente une *bande*.
Une bande appartient à une ligne ou colonne et commence à une position donnée. Comme il peut y avoir plusieurs bandes sur une même ligne ou colonne, chaque bande doit être identifiable.

On représente une bande par

$$LBande_{pos,ID,start}$$

où,

L|C précise s'il s'agit d'une ligne ou d'une colonne.

pos désigne le numéro de la ligne ou la colonne où est située la bande.

ID désigne, s'il existe plusieurs bandes sur la ligne ou colonne, la bande que l'on considère.

start désigne le numéro de colonne ou ligne à partir de laquelle commence la bande.

On défini deux fonctions donnant la longueur d'une bande :

$\texttt{L}(pos, ID)$ donnant la longueur de la ID^{eme} bande à la ligne pos.

$\texttt{C}(pos, ID)$ donnant la longueur de la ID^{eme} bande à la colonne pos.

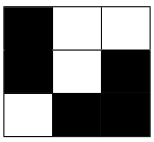


TABLE 2.1 – Grille 3×3

On aura pour la grille 2.1 :

$$\texttt{LBande}_{3,1,2} = \top$$

$$L(3, 2) = 2$$

et

$$x_{3,j} = \top, \forall j \in [3, 2 + \mathtt{L}(3,2)]$$
$$\wedge$$
$$x_{3,2+\mathtt{L}(3,2)+1} = \bot$$

2.1.1 Variables

2.1.2 Contraintes

2.1.3 Codage

2.2 Représentation par bandes

2.2.1 Observation : position minimale, maximale et sous-espace

Soit une colonne d'une longueur de 8 cases avec pour contrainte 3 bandes de taille 1, 2 et 1.
La première bande pourra prendre la position minimum $x_{1,0}$ et maximum $x_{1,\alpha}$.
La valeur de α peut être obtenue à partir du nombre de contraintes, de la taille des différentes bandes et de la règle selon laquelle deux bandes sont séparés d'au moins une case.

Dans l'exemple, on obtient :
$c = 3$, où c est le nombre de contraintes
$l = 8$ est la longueur de la colonne
$L_{total} = 4$, est la longueur cumulée des bandes dans la colonne
$\alpha = l - (L_{total} + (c - 1))$

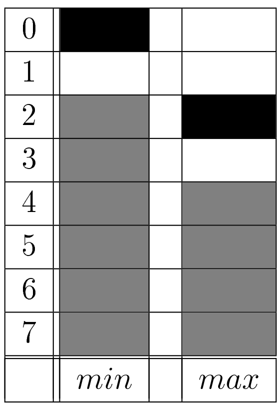


TABLE 2.2 – position *max* et *min* sur une colonne 8×1 avec la contrainte 1 2 1

On voit dans le tableau 2.2 qu'il est possible de connaitre *a priori* le nombre restreint de positions possibles pour la première bande.
On voit aussi que chaque possibilité délimite un *sous-espace* (zone grisée) au sein duquel devront être placées les bandes restantes.

Il est alors possible d'appliquer la même analyse quant à la détermination de la position de la bande suivante au sein du sous-espace.
Une fois les positions de la seconde bande déterminées, elles délimiteront l'espace restant pour la dernière bande.
Suite à cela, il est possible d'élaborer une méthode récursive, ou faussement récursive, au nombre d'itérations réduit, permettant d'énumérer les possibilités de placement des bandes.

2.2.2 Énumération récursive

			$\updownarrow$ $\mathtt{L}(1,1) = 1$					$x_{1,1} = \top$
			$+1$					$x_{1,2} = \bot$
					# itération			$x_{1,3} = \bot$
					$\updownarrow$ $\mathtt{L}(1,2) = 2$			$x_{1,4} = \top$
								$x_{1,5} = \top$
					$+1$			$x_{1,6} = \bot$
							$\updownarrow$ $\mathtt{L}(1,3) = 1$	$x_{1,7} = \top$
								$x_{1,8} = \bot$
$\bullet$		r_1		r_1		r_3	récursion	

TABLE 2.3 – Énumération récursive sur une colonne 8×1 avec la contrainte 1 2 1

Algorithm 2 Énumération des bandes

for ligne $i \rightarrow n$ **do**
  for colonne $j \rightarrow n$ **do** nop() ;
  end for
end for