DM3: Conception et utilisation d'un SAT-solver

1 Conception du SAT-solver

Q10.

Soit n le nombre d'opérateur.

Dans la configuration $(...((a_0|a_1)|a_2)|...)|a_n$, il y a bien n opérateur.

De plus, la complexité de l'algorithme dans cette configuration est

$$\begin{aligned} C_n &= C_{n-1} + C_0 + \Theta(n) \\ &\geq C_{n-1} + nA \\ &\geq C_0 + \sum_{i=0}^{n-1}(n-i)A \\ C_n &= \boxed{\Omega(n^2)} \end{aligned}$$

Donc dans le pire cas, la complexité est au moins en $\Omega(n^2)$.

Pour améliorer la fonction il faut passer par une variable intermédiaire puis trier la liste.

Q11.(Bonus)

Dans la nouvelle fonction, on a

$$\begin{aligned} C_n &= \Theta(n) + \Theta(n\log(n)) \\ C_n &= \boxed{\Theta(n\log(n))} \end{aligned}$$

Q19.

Dans la configuration $\sim(...(\sim T)...)$, la complexité de l'algorithme dans cette configuration est

$$\begin{aligned} C_n &= C_{n-2} + \Theta(n) \\ C_n &= \boxed{\Theta(n^2)} \end{aligned}$$

Q20.(Bonus)

Dans la nouvelle fonction, on simplifie les enfants avant le noeud.

La complexité est

$$\begin{aligned} C_n &= C_i + C_{n-1-i} + \Theta(1), \quad i \in [\![0, n-1]\!] \\ C_n &= \boxed{\Theta(n)} \end{aligned}$$

Q25.

Q26.(Bonus)

Le satsolver spécialiser en FNC a été implémenter dans "fnc_solver.ml". De plus, la propagation unitaire a été rajouter.

En terme d'efficacité, le satsolver en FNC est meilleur que le simple. En effet, dans la résolution du problème a n dames (voir 2^{nd} partie), le satsolver en FNC permet de résoudre plus de dames et en moins de temps.

2 Résolution de problèmes

Q31.

La formule $\boxed{\bigwedge_{1\leq i<j\leq n} (\neg a_i \vee \neg a_j)}$ est sous FNC.

Pour n variables différentes, la formule contient $\frac{n(n-1)}{2}$ variables.

Q38.

La taille de la formule de "gen_formule_n_dames" est

$$\begin{aligned} C_n &= C_{ligne}(n) + C_{col}(n) + C_{diag}(n) + \Theta(1) \\ &= nE_x(n) + nE_p(n) + 2\sum_{i=-n+2}^{n-2} E_p(|n-i|) + \Theta(n) \end{aligned}$$

Avec E_x la fonction "exactement_une" et E_p "au_plus_une".

Or $nE_x(n) = \Theta(n^3)$, $nE_p(n) = \Theta(n^3)$ et $2\sum_{i=-n+2}^{n-2} E_p(|n-i|) \leq 4nE_p(n) = O(n^3)$,

$$\begin{aligned} C_n &= \Theta(n^3) + \Theta(n^3) + O(n^3) + \Theta(n) \\ C_n &= \boxed{O(n^3)} \end{aligned}$$

Q40.

Pour le problème a 5 dames, on obtient

```
fred@mp2:~/DM3$ problemes/n_dames 5
Fichier '5_dames.txt' généré.
Taille du fichier : 3471 octets.
fred@mp2:~/DM3$ satsolver/fnc_solver '5_dames.txt'
La formule est sous FNC.
La formule est satisfiable en assignant 1 aux variables suivantes et 0 aux autres :
X_0_4
X_1_2
X_2_0
X_3_3
X_4_1
Temps d'exécution : 0.001377 s
```

Soit encore

	0	1	2	3	4
0			X		
1					X
2		X			
3				X	
4	X				

Pour le problème a 8 dames, on obtient

```
fred@mp2:~/DM3$ satsolver/fnc_solver '8_dames.txt'
La formule est sous FNC.
La formule est satisfiable en assignant 1 aux variables suivantes et 0 aux autres :
X_0_7
X_1_3
X_2_0
X_3_2
X_6_6
X_5_1
X_4_5
X_7_4
Temps d'exécution : 0.005287 s
```

Soit encore

	0	1	2	3	4	5	6	7
0			X					
1						X		
2				X				
3		X						
4								X
5					X			
6							X	
7	X							

Et pour le problème a 3 dames, on obtient

```
fred@mp2:~/DM3$ satsolver/fnc_solver '3_dames.txt'
La formule est sous FNC.
La formule est insatisfiable.
Temps d'exécution : 0.000985 s
```

3 Problème des cinq maisons

3.1 Description

Le problème des cinq maisons requiert certaines contraintes :

1. φ_1 : Chaque caractéristique (« anglais », « poisson rouge », ...) se retrouve dans exactement une des cinq maisons
2. φ_2 : Chaque maison doit avoir exactement une caractéristique de chaque catégorie (nationalité, boisson, couleur, ...). Par exemple, la maison 1 doit avoir exactement une couleur.
3. φ_3 : Contraintes de l'énoncé

3.2 Nomenclature

Les variables utilisées dans les formules sont de la forme `numéro_caractéristique`. Par exemple :

— `1_anglais` représente « l'anglais habite dans la maison 1 »
— `5_poisson_rouge` représente « la personne habitant la maison 5 a pour animal de compagnie un poisson rouge »
— `3_yop` représente « la personne habitant la maison 3 boit du yop »
— etc.

Dans la suite de cette section, notons C l'ensemble des caractéristiques ($C = \{\text{anglais}, \text{lait}, \text{escalade}, \dots\}$), $\mathscr{C}$ l'ensemble des catéogries de caractéristiques (nationalité, couleur, ...) (donc $C = \bigsqcup_{\mathcal{C} \in \mathscr{C}} \mathcal{C}$) et utilisons des indices de maison dans $[\![1, 5]\!]$.

3.3 Modélisation de la contrainte 1

Pour modéliser la contrainte « Chaque caractéristique se retrouve dans exactement une des cinq maisons », on peut d'abord modéliser la contrainte « La caractéristique c se retrouve dans exactement une maison », en utilisant la formule suivante :

$$\bigvee_{j=1}^{5} \left(\texttt{j\_c} \wedge \bigwedge_{\substack{i=1 \\ i \neq j}}^{5} \neg \texttt{i\_c} \right)$$

(i.e. « Soit cette caractéristique est dans la maison 1 et aucune autre, soit dans la 2 et aucune autre, etc. »)

Ainsi, on peut modéliser la contrainte sur toutes les caractéristiques en faisant un « et » logique :

$$\varphi_1 \equiv \bigwedge_{c \in C} \bigvee_{j=1}^{5} \left(\texttt{j\_c} \wedge \bigwedge_{\substack{i=1 \\ i \neq j}}^{5} \neg \texttt{i\_c} \right)$$

3.4 Modélisation de la contrainte 2

Pour modéliser la contrainte « Chaque maison doit avoir exactement une caractéristique de chaque catégorie », on peut d'abord modéliser la contrainte « La maison i a exactement une caractéristique de la catégorie $\mathcal{C} \in \mathscr{C}$ », en utilisant la formule suivante :

$$\bigvee_{c' \in \mathcal{C}} \left(\texttt{i\_c'} \wedge \bigwedge_{\substack{c \in \mathcal{C} \\ c \neq c'}} \neg \texttt{i\_c} \right)$$

On peut alors modéliser la contrainte sur toutes les catégories de caractéristiques avec un « et » logique :

$$\bigwedge_{\mathcal{C} \in \mathscr{C}} \bigvee_{c' \in \mathcal{C}} \left(\texttt{i\_c'} \wedge \bigwedge_{\substack{c \in \mathcal{C} \\ c \neq c'}} \neg \texttt{i\_c} \right)$$

Et de même, on modélise la contrainte sur toutes les maisons ainsi :

$$\varphi_2 \equiv \bigwedge_{i=1}^{5} \bigwedge_{\mathcal{C} \in \mathscr{C}} \bigvee_{c' \in \mathcal{C}} \left(\texttt{i\_c'} \wedge \bigwedge_{\substack{c \in \mathcal{C} \\ c \neq c'}} \neg \texttt{i\_c} \right)$$

3.5 Modélisation de la contrainte 3

Les contraintes de l'énoncé se modélisent assez facilement. Par exemple, « L'Anglais vit dans une maison rouge » se modélise ainsi :

$$\bigvee_{i=1}^{5} \texttt{i\_anglais} \wedge \texttt{i\_rouge}$$

« La personne qui fait de l'escalade vit à côté de celle qui a des chats » se modélise avec cette formule :

$$\left(\bigvee_{i=1}^{4} \texttt{i\_escalade} \wedge \texttt{(i+1)\_chats} \right) \vee \left(\bigvee_{i=1}^{4} \texttt{i\_chats} \wedge \texttt{(i+1)\_escalade} \right)$$

Encore plus simple, « La personne qui vit dans la maison du centre boit du lait » se modélise avec la formule suivante :

$$\texttt{3\_lait}$$

Il suffit ensuite de faire un « et » logique sur toutes les formules données par les contraintes pour obtenir φ_3.

3.6 Solution

Le satsolver résoud le problème en 0.4 secondes. À noter que l'ordre des contraintes est important : écrire dans le fichier $\varphi_1 \wedge \varphi_2 \wedge \varphi_3$ prend au moins 10 minutes (peut-être beaucoup plus, on l'a arrêté avant), tandis que $\varphi_3 \wedge \varphi_2 \wedge \varphi_1$ est beaucoup plus rapide.

La solution est la suivante :

Maison	1	2	3	4	5
Couleur	Jaune	Bleu	Rouge	Vert	Blanc
Nationalité	Norvégien	Danois	Anglais	Allemand	Suédois
Animal	Chats	Cheval	Oiseaux	Poisson rouge	Chiens
Boisson	Eau	Thé	Lait	Café	Yop
Sport	Danse	Escalade	Vélo	Karaté	Basket

Ainsi, le poisson rouge est l'animal de compagnie de l'Allemand.

4 Problème du calendrier

Description

Pour la résolution du problème du calendrier, il faut respecter ces deux règles :

1. Chaque case a une pièce (voir 0 pour certaine case).
2. Chaque pièce n'est utiliser qu'une seul fois.

Pour représenter le problème, chaque case aura 10 variables, indiquant si la pièce correspondante est dessus.

Elle seront de la forme "X_0_0" avec X le nom de la pièces, (0,0) les coordonnées de la cases. Exemples : "l_4_5", "T_2_7"

(le nom des pièces sont : I, L, S, b, C, l, s, Z, T et V)

Pour optimiser, les cases qui doivent avoir 0 pièce ne seront pas crée et utiliser.

1. Contrainte d'une case

Pour $p_{i,j}$ les variables "p est en (i,j)" et $\mathbb{P}$ l'ensembles des pièces, la contrainte sur la case (i, j) est

$$F_{i,j} = \left(\bigvee_{p \in \mathbb{P}} p_{i,j} \right) \wedge \bigwedge_{\substack{(p,p') \in \mathbb{P}^2 \\ p < p'}} (\neg p_{i,j} \vee \neg p'_{i,j})$$

Cette formule est sous FNC.

2. Contrainte de toutes les cases

Pour $C \subset [\![0, 7]\!]^2$ l'ensemble des cases qui doivent être remplie, la contrainte de toutes les cases est

$$F = \bigwedge_{(i,j) \in C} \left(\left(\bigvee_{p \in \mathbb{P}} p_{i,j} \right) \wedge \bigwedge_{\substack{(p,p') \in \mathbb{P}^2 \\ p < p'}} (\neg p_{i,j} \vee \neg p'_{i,j}) \right)$$

Cette formule est sous FNC.

3. Contrainte d'une pièce

Pour $\mathbb{V}_p$ l'ensemble des positions valides de la pièce p, et

$$N_v(p_{i,j}) = \begin{cases} p_{i,j} & \text{si } p \text{ dans la position } v \text{ est sur la case } (i,j) \\ \neg p_{i,j} & \text{sinon} \end{cases}$$

Exemple pour la pièce T de position $v = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

On a $N_v(T_{1,2}) = T_{1,2}$ et $N_v(T_{3,1}) = \neg T_{3,1}$

La contrainte d'une pièce p est

$$F_p = \bigvee_{v \in \mathbb{V}_p} \left(\bigwedge_{(i,j) \in C} N_v(p_{i,j}) \right)$$

Mais cette formule n'est pas sous FNC, on doit passer par des variables intermédiaire que l'on nomme $Z_{p,v}$.

$$\begin{aligned}
F_p &= \bigvee_{v \in \mathbb{V}_p} \left(\bigwedge_{(i,j) \in C} N_v(p_{i,j}) \right) \\
&= \left(\bigvee_{v \in \mathbb{V}_p} Z_{p,v} \right) \wedge \bigwedge_{v \in \mathbb{V}_p} \left(Z_{p,v} \leftrightarrow \bigwedge_{(i,j) \in C} N_v(p_{i,j}) \right) \\
&= \left(\bigvee_{v \in \mathbb{V}_p} Z_{p,v} \right) \wedge \bigwedge_{v \in \mathbb{V}_p} \left(Z_{p,v} \rightarrow \bigwedge_{(i,j) \in C} N_v(p_{i,j}) \right) \wedge \bigwedge_{v \in \mathbb{V}_p} \left(\neg Z_{p,v} \rightarrow \neg \bigwedge_{(i,j) \in C} N_v(p_{i,j}) \right)
\end{aligned}$$

Que l'on peut simplifié par

$$\begin{aligned}
F_p &= \left(\bigvee_{v \in \mathbb{V}_p} Z_{p,v} \right) \wedge \bigwedge_{v \in \mathbb{V}_p} \left(Z_{p,v} \rightarrow \bigwedge_{(i,j) \in C} N_v(p_{i,j}) \right) \\
&= \left(\bigvee_{v \in \mathbb{V}_p} Z_{p,v} \right) \wedge \bigwedge_{v \in \mathbb{V}_p} \left(\neg Z_{p,v} \vee \bigwedge_{(i,j) \in C} N_v(p_{i,j}) \right) \\
&= \left(\bigvee_{v \in \mathbb{V}_p} Z_{p,v} \right) \wedge \bigwedge_{\substack{(i,j) \in C \\ v \in \mathbb{V}_p}} \left(\neg Z_{p,v} \vee N_v(p_{i,j}) \right)
\end{aligned}$$

qui est sous FNC.

4. Contrainte de toutes les pièces

La contrainte de toutes les pièces est

$$F = \bigwedge_{p \in \mathbb{P}} \left(\left(\bigvee_{v \in \mathbb{V}_p} Z_{p,v} \right) \wedge \bigwedge_{\substack{(i,j) \in C \\ v \in \mathbb{V}_p}} \left(\neg Z_{p,v} \vee N_v(p_{i,j}) \right) \right)$$

Cette formule est sous FNC.

5. Formule générale

La formule générale est donc

$$F = \bigwedge_{(i,j) \in C} \left(\left(\bigvee_{p \in \mathbb{P}} p_{i,j} \right) \wedge \bigwedge_{\substack{(p,p') \in \mathbb{P}^2 \\ p < p'}} \left(\neg p_{i,j} \vee \neg p'_{i,j} \right) \right) \wedge \bigwedge_{p \in \mathbb{P}} \left(\left(\bigvee_{v \in \mathbb{V}_p} Z_{p,v} \right) \wedge \bigwedge_{\substack{(i,j) \in C \\ v \in \mathbb{V}_p}} \left(\neg Z_{p,v} \vee N_v(p_{i,j}) \right) \right)$$

Implémentation