Rapport TP3 : Deux réductions vers SAT

Auteurs : Boudermine Antoine
Pelletier Sébastien

Au cours de ce TP, nous allons travailler sur les réductions polynomiales de deux problèmes :
- Un graphe admet – il un noyau
- Une grille de sudoku amet – elle une solution.

Pour ce faire nous allons réduire ces problèmes à SAT et tester ces réductions avec le solveur Minisat.

Kernel à SAT

Un graphe $G = (V, E)$ admet un noyau $K \subseteq V$ si :
- Deux sommets de K ne sont pas reliés par un arc de *E*
- $\forall\, v \in V \backslash K$ il existe un sommet une arête aller de v à un sommets de K.

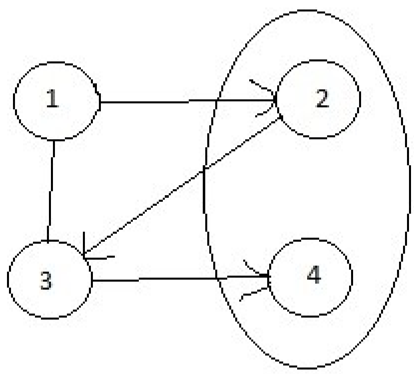
Voici une réduction en temps polynomial de KERNEL à SAT :

$$\varphi = \left(\bigcap_{(i,j \in E)} (\overline{V_i} \vee \overline{V_j}) \right) \wedge \bigcap_{i \in V} \left(V_i \vee \bigcup_{(i,j) \in E} V_j \right)$$

Ici Vi = Vrai Si et seulement si le sommet i fait partis du noyau.

Nous avons développé un programme en C++ qui à partir d'un graphe donné en paramètre sous la forme vu dans le TP précédent (3 3 1 2 2 3 3 1 pour une clique de taille trois) produit une formule propositionnel en format Dimacs. Pour l'utiliser il faut faire : ./exo1_test.bash "3 3 1 2 2 3 3 1" (Ce scripte utilise le programme en c++ et gère l'affichage en fonction de la sortie de minisat)

Pour ce graphe : 4 4 1 2 2 3 3 4 ce visualise comme suit (4 sommet et 4 arcs) :



./exo1_test.bash "4 4 1 2 1 3 2 3 3 4" donne
Graphe : 4 4 1 2 1 3 2 3 3 4
Fomule :
p cnf 4 8
-1 -2 0
-1 -3 0
-2 -3 0
-3 -4 0
1 2 3 0
2 3 0
3 4 0
4 0

Il existe un noyau composé de(s) sommet(s) : 2 4

Réduction de Sudoku à SAT

Nous allons réduire le problème du Sudoku (une grille donné amet – elle une solution) à SAT.
Pour se faire nous allons traduire chaque règles qu'implique les Sudokus à SAT:

1. Au moins une valeur par case.
2. Au moins une fois chaque chiffre sur chaque ligne.
3. Au moins une fois chaque chiffre sur chaque colonne.
4. Au moins une fois chaque chiffre dans chaque région.
5. Au plus une valeur par case.
6. Au plus une fois chaque chiffre sur chaque colonne.
7. Au plus une fois chaque chiffre sur chaque ligne.
8. Au plus une fois chaque chiffre sur chaque colonne.
9. Au plus une fois chaque chiffre dans chaque région.

On introduit une variable Ci,j,k telle que Ci,j,k = Vrai Si et seulement si le casse i,j contient la valeur k.
On a donc :

$$\varphi_1 = \bigcap_{(i,j)\in(1..9)} \left(\bigcup_{k\in(1..9)} C_{i,j,k} \right)$$

$$\varphi_2 = \bigcap_{i\in(1..9)} \left[\bigcap_{k\in(1..9)} \left(\bigcup_{j\in(1..9)} C_{i,j,k} \right) \right]$$

$$\varphi_3 = \bigcap_{j\in(1..9)} \left[\bigcap_{k\in(1..9)} \left(\bigcup_{i\in(1..9)} C_{i,j,k} \right) \right]$$

$$\varphi_4 = \bigcap_{w\in(0,2)} \left[\bigcap_{w'\in(0,2)} \left(\bigcap_{k\in(1..9)} \bigcup_{i\in(1+3w..3+3w)} \left(\bigcup_{j\in(1+3w'..3+3w')} C_{i,j,k} \right) \right) \right]$$

$$\varphi_5 = \bigcap_{(i,j)\in(1..9)} \left(\bigcap_{k\in(1..9)} \left(\bigcap_{k'\in(1..9)} \overline{C_{i,j,k}} \vee \overline{C_{i,j,k'}} \right)\right)$$

$$\varphi_6 = \bigcap_{i\in(1..9)} \left[\bigcap_{k\in(1..9)} \left(\bigcap_{j\in(1..9)} \left(\bigcap_{j'\in(1..9)\neq j} \overline{C_{i,j,k}} \vee \overline{C_{i,j',k}} \right)\right)\right]$$

$$\varphi_7 = \bigcap_{i\in(1..9)} \left[\bigcap_{k\in(1..9)} \left(\bigcap_{i\in(1..9)} \left(\bigcap_{i'\in(1..9)\neq j} \overline{C_{i,j,k}} \vee \overline{C_{i',j,k}} \right)\right)\right]$$

$$\varphi_8 = \bigcap_{w\in(0,2)} \left[\bigcap_{w'\in(0,2)} \left(\bigcap_{k\in(1..9)} \left(\bigcap_{i\in(1+3w,3+3w)} \left(\bigcap_{i'\in(1+3w,3+3w)\neq i} \left(\bigcap_{j\in(1+3w,3+3w)} \left(\bigcap_{j'\in(1+3w,3+3w)\neq j} \overline{C_{i,j,k}} \vee \overline{C_{i',j',k}} \right)\right)\right)\right)\right)\right]$$

$$\varphi_{sudoku} = \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4 \wedge \varphi_5 \wedge \varphi_6 \wedge \varphi_7 \wedge \varphi_8$$

Pour les cas 6, 7 ,8 on vérifie qu'il n'existe aucun couple de case dans chaque ligne/colonne/région qui contient la même la valeur de k.

à virer :

\ \varphi_{8} = \bigcap_{w \in (0,2)} \left [\bigcap_{w'\in (0,2)}^{ } \left (\bigcap_{k\in (1..9)}^{ } \left (\bigcap_{i\in (1+3w, 3+3w)}^{ } \left (\bigcap_{i'\in (1+3w, 3+3w) \neq i}^{ } \left (\bigcap_{j\in (1+3w, 3+3w)}^{ } \left (\bigcap_{j'\in (1+3w, 3+3w)\neq j}^{ } \overline{C_{i,j,k}} \vee \overline{C_{i',j',k}} \right) \right) \right) \right) \right) \right] \right]

\ \varphi_{4} = \bigcap_{w \in (0,2)} \left[\bigcap_{w' \in (0,2)} \left (\bigcap_{k \in (1..9)} \bigcup_{i \in (1+3w..3+3w)} \left (\bigcup_{j \in (1+3w'..3+3w')} C_{i,j,k} \right)\right)\right]

\ \varphi_{2} = \bigcap_{i \in (1..9)} \left[\bigcap_{k \in (1..9)} \left (\bigcup_{j \in (1..9)} C_{i,j,k} \right) \right]

\ \varphi_{3} = \bigcap_{j \in (1..9)} \left[\bigcap_{k \in (1..9)} \left (\bigcup_{i \in (1..9)} C_{i,j,k} \right) \right]

\ \varphi_{8} = \bigcap_{w \in (0,2)} \left [\bigcap_{w'\in (0,2)}^{ } \left (\bigcap_{k\in