

Feuille de T. P. 1 : Utilisation d'un solveur SAT

Préliminaires : MiniSAT

- MiniSAT est installé sur votre compte sous Linux. Pour une utilisation sur votre machine personnelle, URL : <http://minisat.se/MiniSat.html>.
- Documentation, URL : <https://www.dwheeler.com/essays/minisat-user-guide.html>
- Pour lancer MiniSAT:
 - se connecter sous Linux
 - ouvrir un terminal
 - taper :

```
minisat
```
 - pour l'aide en ligne :

```
minisat --help
```

Format DIMACS Les solveurs SAT utilisent des formules et des ensembles de formules au format DIMACS (pour en savoir plus, vous pouvez consulter: <http://www.satlib.org/Benchmarks/SAT/satformat.ps>) Les CNF sont codées dans un fichier d'extension `.cnf` structuré ainsi :

- le fichier commence par une entête qui est suivie de la liste des clauses de votre formule cnf. A tout moment, vous pouvez insérer des commentaires, une ligne de commentaire commençant par `c`.
- la ligne d'entête est:

```
p cnf nbr_de_propositions nbr_de_clauses
```

où les paramètres `nbr_de_propositions` et `nbr_de_clauses` doivent être respectivement remplacés par le nombre de propositions utilisées et le nombre de clauses présentes dans votre formule cnf.

- un littéral positif est codé par un entier positif et un littéral négatif est codé par un entier négatif.
- une clause est représentée par une suite de littéraux séparés par un espace
- les clauses sont séparées par un 0.

Exemple : la formule $(x_1 \vee x_3 \vee \neg x_4) \wedge (x_4) \wedge (x_2 \vee \neg x_3)$ peut être codée par un fichier `exemple.cnf` contenant :

```
c exemple de fichier au format CNF
p cnf 4 3
1 3 -4 0
4 0
2 -3
0
```

Dans cet exemple, les propositions x_1 , x_2 , x_3 et x_4 sont respectivement codées par les entiers 1, 2, 3 et 4.

Exercice 1 : Format DIMACS

Ecrire l'exemple précédent dans un fichier `test.cnf` à l'aide d'un éditeur. Déterminer ensuite si la formule est satisfaisable en lançant la commande `minisat test.cnf`

Pour obtenir les modèles, il faut ajouter un fichier texte en deuxième argument de la commande. Les modèles seront alors écrits dans ce fichier.

```
minisat test.cnf result.txt
```

Exercice 2 : Test des exercices du cours

Tester avec le solveur MiniSat les exercices suivants vus en cours :

- $S = \{p \vee q, p \vee r, \neg q \vee \neg r, \neg p\}$
- $S = \{\neg a \vee b, a, \neg b\}$
- $F = ((p \rightarrow q) \wedge p) \rightarrow q$
- $\{p \rightarrow r, q \rightarrow r\} \models (p \vee q) \rightarrow r$

Exercice 3 : Coloriage de graphe

- représenter en logique propositionnelle le problème du coloriage d'un graphe un peu plus simple que celui vu au cours 1 : coloriage avec 3 couleurs d'un graphe comportant 4 sommets : S_1, S_2, S_3, S_4 et 5 arêtes comme suit : une arête entre S_1 et S_2 , une arête entre S_1 et S_4 , une arête entre S_2 et S_3 , une arête entre S_3 et S_4 , une arête entre S_1 et S_3 .
- coder le problème au format DIMACS
- tester le problème avec MiniSat

Exercice 4 : Les témoins

- représenter en logique propositionnelle le problème des témoins vu au cours 1.
- coder le problème au format DIMACS
- tester le problème avec MiniSat