

Projet Programmation Logique - partie n° 1

Julien REICHERT

LSV, ENS Cachan

Vendredi 7 février 2014

La première partie du projet de programmation logique est consacrée aux SAT-solvers. Le devoir associé est à rendre pour le retour des vacances d'hiver.

1 Glucose

Nous allons travailler sur un tel outil : Glucose¹. La première séance permettra de se familiariser avec l'outil, ainsi qu'avec le format qu'il emploie, à savoir dimacs².

Glucose résout le problème de satisfaisabilité pour des formules sous forme normale conjonctive (le principe de l'outil est dans le nom : « glue clauses »). Ces formules s'écrivent dans le format dimacs de la façon suivante :

- La première ligne hors commentaires commence par 'p' (« problème »), précise le type (« cnf » pour les formules en CNF, que nous utiliserons, par exemple), le nombre de variables et le nombre de clauses.
- Ensuite, on décrit des clauses, sous la forme de suites d'entiers strictement positifs, séparés par des espaces, indiquant l'identifiant des variables propositionnelles, et précédés d'un '-' si les variables apparaissent négativement. Les clauses sont séparées par des '0'³.
- Les lignes commençant par 'c' sont des commentaires.

Par exemple, la formule $(x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee \neg x_1)$ s'écrira :

```
p cnf 3 2
1 -3 0
2 3 -1 0
```

1. <http://www.labri.fr/perso/lSimon/glucose/>
2. Les fichiers doivent aussi avoir pour extension .cnf - Conjunctive Normal Form - voir aussi <http://people.sc.fsu.edu/~jburkardt/data/cnf/cnf.html>.
3. Et il serait bienvenu pour la lisibilité d'utiliser une ligne par clause.

2 Utilisation

Vous pouvez télécharger et décompresser l'archive sur le lien en note de bas de page, puis l'installer via la commande `make` dans le dossier `core`. Ensuite, l'exécutable s'appelle en faisant `./glucose` avec éventuellement le nom du fichier CNF qui contient votre formule, autrement une interface en ligne de commande vous permet de la saisir, en finissant la saisie par la commande `Ctrl+D`. Si vous souhaitez trouver une assignation des variables qui satisfait la formule, en cas de satisfaisabilité, ajoutez l'option `-model` en appelant l'exécutable.

3 Premier projet

L'exercice ci-dessous se compose de plusieurs étapes et vise à étudier une situation où le problème de satisfaisabilité est illustré. Il s'agit de trouver des déductions à faire dans une instance du démineur, à partir d'un sous-ensemble de cases révélées. Les étapes sont les suivantes :

1. Transformer une grille partiellement résolue en une formule propositionnelle.
2. Mettre cette formule au format d'entrée de Glucose.
3. Trouver des cases pour lesquelles on déduit qu'il y a, ou qu'il n'y a pas, une mine.

Les instances de démineur en entrée sont des grilles partiellement (ou totalement) résolues, écrites sous le format suivant :

```
<nb lignes> <nb colonnes> <nb mines>
<ligne> <colonne> <info>
...
<ligne> <colonne> <info>
```

Toutes les cases apparaissent une fois exactement dans la grille ; l'information est -1 si on ne sait rien, 9 si la case contient une mine, ou sinon un entier entre 0 et 8 indiquant le nombre de mines parmi les 8 cases voisines.

Étape préliminaire : Vérification

Étant donné une grille dans le format ci-dessus et sans le moindre -1, écrivez un programme qui vérifie qu'il n'y a pas d'incohérence. Le programme peut se contenter de compter les mines voisines de chaque case, il s'agit surtout de pouvoir comparer ses résultats avec ceux donnés par Glucose sur la formule que vous construisez.

Étape 1 : De la grille à la formule

Vous écrirez en OCaml (avec une certaine tolérance pour un autre langage) un programme qui transforme une grille partiellement résolue en une formule propositionnelle dans le format de votre choix, où l'on associe à chaque case de la grille une variable indiquant si elle contient une mine ou non. Une interprétation de toutes les variables correspond à la répartition des mines dans la grille.

Étape 2 : De la formule au Glucose

Vous écrirez en OCaml (même remarque) un programme qui transforme une formule propositionnelle du format que vous aurez choisi en une formule en CNF dont on peut tester la satisfaisabilité avec Glucose⁴.

Étape 3 : Déductions

À partir de la formule en CNF obtenue, vous chercherez un moyen de déduire la présence ou non d'une mine dans au moins une case dont l'information était -1 dans la grille partielle.

Bonus : Résolution du jeu

Considérant une grille totalement résolue, créez un programme qui répète les trois premières étapes pour arriver à cette grille totalement résolue à partir d'une grille vierge en minimisant le nombre de cases révélées « en cliquant au hasard ». Votre programme devra évidemment reposer sur Glucose⁵.

4. N'hésitez pas à ajouter du sucre syntaxique. . .

5. Pour votre santé, mangez au moins cinq fruits et légumes par jour. <http://mangerbouger.fr>