

Projet M1 : Création d'emplois du temps

4 novembre 2014

Le but de ce projet est d'écrire un programme pour créer des emplois du temps étant donné un certain nombre de contraintes : les salles à disposition, les cours à dispenser, les différents professeurs et les différentes filières. Vous devrez commencer par lire ces contraintes depuis un fichier dont la syntaxe est décrite à la section 2. Puis vous devrez traduire ces contraintes-là en un problème SAT et utiliser un SAT solver (Minisat par exemple) pour résoudre le problème (voir le TP5 pour les explications techniques). Enfin, dans le cas où Minisat répond positivement, vous devez pouvoir réinterpréter la solution qu'il fournit pour afficher un emploi du temps lisible à l'écran.

1 Modalités

Ce projet peut être réalisé *en binôme* (c'est d'ailleurs un bon exercice de mener un projet de programmation à deux). Il est à rendre par mail (fcapelli@math.univ-paris-diderot.fr) avant le **04 Janvier 2015 à 23h59**. Il comportera le code source de vos programmes, *indenté et commenté*. Ce code source doit compiler avec `gcc -Wall`. La lisibilité et l'organisation du code seront un facteur important de la note finale. Il doit être accompagné d'un rapport de 4 à 8 pages présentant votre traduction vers un problème SAT, des exemples d'emplois du temps générés à partir de fichiers de contraintes bien choisis (et aussi des exemples de fichiers de contraintes n'ayant pas de solutions). Vous pouvez aussi analyser l'efficacité de votre programme en fonction du nombre de contraintes (et expliquer alors comment vous générez ce genre de fichiers). Vous pouvez aussi expliquer comment vous avez amélioré le programme si vous avez le temps d'en faire un peu plus (voir section 3).

Votre projet doit pouvoir au minimum (éventuellement en plusieurs programmes distincts, dans ce cas, bien expliquer cela dans votre rapport) :

- prendre en entrée un fichier de contraintes tel que décrit dans la section 2.
- générer un fichier pour Minisat qui sera satisfiable si et seulement si il existe un emploi du temps remplissant les contraintes précédentes.
- être capable d'analyser la sortie de Minisat pour afficher un emploi du temps lisible c'est-à-dire de la forme :

```
Salle 1 :  
Jour 1 :  
8–10 : Cours Bla avec Prof Toto  
10–12 : Cours Bibi avec Prof Tartempion  
14–18 : /
```

```
Jour 2 : ...
```

```
Salle 2:  
Lundi :  
...
```

- Il est aussi fortement conseillé d'écrire un programme qui prend en entrée un fichier de contraintes tel que décrit à la section 2 et qui essaie de générer un emploi du temps sans utiliser Minisat, en essayant tout simplement toutes les possibilités. Cela vous permettra de vérifier que votre traduction vers Minisat est correcte sur de petits exemples.

Si certaines des consignes ci-dessus ne sont pas remplies par votre programme, n'hésitez pas à commenter cela dans le rapport, à expliquer et à mettre en évidence certains bugs. L'analyse honnête que vous ferez de votre programme est aussi importante que le programme en lui-même.

2 Syntaxe des fichiers d'entrée

On suppose qu'on veut créer un emploi du temps sur n jours et on dispose d'une plage de m heures par jour.

Les contraintes du fichiers d'entrée spécifieront, toujours dans cet ordre :

- le nombre n de jour et le nombre m d'heures disponible par jour
- les salles dont on dispose et leur capacité maximale
- les professeurs et leurs noms
- une liste de filières avec leurs effectifs
- les cours qui doivent avoir lieu, leur durée, le professeur qui les dispense ainsi que la filière concernée par ce cours

Bien entendu, un professeur ne peut pas donner deux cours en même temps, de même qu'une même filière ne peut pas suivre deux cours en même temps ou que deux cours ne peuvent avoir lieu en même temps dans la même salle etc. Votre traduction vers un fichier CNF devra tenir compte de ces contraintes. Il est conseillé d'écrire des exemples de contraintes pour tester chacun des différents aspects (par exemple, un fichier avec une seule filière et un seul professeur, et beaucoup de salle pour vérifier qu'aucun cours ne sera mis en parallèle).

Voici comment se présentera un tel fichier :

n m

SALLE Nb_salle

Id Capacite

PROF Nb_prof

Id Nom

FILIERE Nb_filiere

Id Nom Effectif

COURS Nb_cours

Id Nom Duree Id_Prof Id_Filiere

Par exemple, si on veut créer un emploi du temps sur 2 jours, avec 8h par jour et 3 salles pour la filière L1-Math de 30 élèves ayant les cours Algebre1 de 4h dispensé par M. Toto et Analyse1 de 4h dispensé par M. Tartempion et pour la filière M1-Math de 12 élèves ayant le cours ProgC de 2h dispensé par M. Tartempion, on écrira :

2 8

SALLE 3

1 10

2 20

3 40

PROF 2

1 Toto

2 Tartempion

FILIERE 2

1 M1-Math 12

2 L1-Math 30

COURS 3

1 Algebre1 4 1 2
2 Analyse1 4 2 2
3 ProgC 2 2 1

On spécifie toujours le nombre de jours et d'heures disponibles puis le nombre de salles, de professeurs, de filière et de cours qu'on va lister ensuite. Chaque objet est précédé d'un identifiant unique et chaque propriété de l'objet est séparée par une espace. Un saut de ligne sépare différents objets. Les noms des professeurs, des filières et des cours ne doivent pas comporter d'espace. Pour un cours, on indique son identifiant, son nom suivi de la durée (un nombre entier, le nombre d'heure), de l'identifiant du professeur qui le dispense et de l'identifiant de la filière qui le suit.

3 Améliorations possibles

Si vous avez un programme qui remplit toutes les modalités de la section 1 et que vous l'avez déjà bien analysé, vous pouvez encore l'améliorer. Voici quelques idées que vous pouvez mettre en application. Bien entendu, vous pouvez aussi proposer des améliorations différentes tant qu'elles sont expliquées et justifiées dans le rapport.

3.1 Amélioration des contraintes

Comme idée d'amélioration, on peut enrichir le langage des contraintes. On peut par exemple ajouter des préférences pour les professeurs (pour certains jours). Vous devez ensuite essayer de respecter au mieux ces préférences.

On peut aussi imaginer qu'un cours doit être suivi par plusieurs filières différentes et peut être dispensé par différents professeur. Si deux filières doivent suivre le même cours, on peut soit les rassembler si on dispose d'une salle suffisamment grande, soit faire deux sessions pour le même cours. On pourra chercher à minimiser le nombre de sessions pour un même cours.

3.2 Amélioration de la sortie

Au lieu d'afficher l'emploi du temps dans la console, on peut aussi exporter cela vers un langage de formatage de texte plus agréable. Si vous connaissez, vous pouvez par exemple générer un fichier HTML ou un fichier Latex contenant un tableau récapitulatif de l'emploi du temps.

3.3 Comparaison avec d'autres SAT solvers

Minisat n'est pas le seul ni le meilleur SAT solver existant. Il en existe beaucoup d'autres, par exemple Glucose ou Picosat. Vous pouvez essayer d'utiliser ces autres SAT solvers pour résoudre votre problème et comparer leurs performances.