UNIVERSITÉ LIBRE DE BRUXELLES

INFO-F-302 Informatique Fondamentale

Projet : Logique Propositionnelle et Utilisation de l'Outil MiniSat

---

L'objectif de ce projet est de modéliser le *problème d'emploi du temps* en logique propositionnelle, et de le résoudre avec l'outil MiniSat.

1 L'Outil MiniSat

L'outil MiniSat est un programme qui décide le problème SAT. Si la formule est satisfaisable, une interprétation qui la satisfait est retournée.

`http://minisat.se/`

MiniSat prend en entrée une formule en FNC de la logique propositionnelle. Une version de MiniSat permet aussi de résoudre le problème MAX-SAT (voir cours). Vous avez déjàá un exemple du code nécessaire pour utiliser MiniSat.

`http://www.ulb.ac.be/di/info-f-302/2015/123path.tar.gz`

2 Le problème d'emploi du temps

On va travailler sur les horaires d'examens de l'ULB, vus de manière générale comme une fonction d'un ensemble fini d'examens vers un ensemble fini de salles et d'instant dans le temps.

Considérons un ensemble fini de salles S et une fonction de capacité $c : S \to \mathbb{N}$ qui nous donne la capacité maximale d'une salle. On se donne aussi un ensemble fini E d'étudiants, un ensemble fini P de professeurs, et un ensemble fini X d'examens. Nous avons aussi deux fonctions qui représente les examens de chaque étudiant et chaque professeur :

$$a : E \to 2^X \setminus \emptyset \qquad b : P \to 2^X \setminus \emptyset.$$

Finalement, on considère un borne du temps $T \in \mathbb{N}$.

Dans ce problème, il vous est demandé de trouver un horaire d'examens correcte. Formellement, l'horaire est donné par une fonction $\mu : X \to S \times \{1, \ldots, T\}$: pour tout $x \in X$, si $\mu(x) = (s, t)$, alors cela veut dire que l'examen x aura lieu dans la salle s à l'instant t. La durée d'un examen est toujours considérée comme étant une unité de temps. C'est une fonction μ satisfaisant certaines contraintes (que vous devrez définir) que vous devez trouver.

Exemple 1 Si on vous donne

— $T = 4$

— $S = \{s_1, s_2, s_3\}$

— $c(s_1) \mapsto 10$, $c(s_2) \mapsto 30$, $c(s_3) \mapsto 100$

— $E = \{e_1, \ldots, e_{100}\}$

— $P = \{p_1, p_2\}$

— $X = \{x_1, \ldots, x_4\}$

— $a(e_i) \mapsto \{x_1\}$, $1 \leq i \leq 30$; $a(e_i) \mapsto \{x_2\}$, $31 \leq i \leq 60$; $a(e_i) \mapsto \{x_3, x_4\}$, $61 \leq i \leq 100$

— $b(p_1) \mapsto \{x_1, x_3\}$, $b(p_2) \mapsto \{x_2, x_4\}$

l'horaire défini par μ, telle que $\mu(x_1) \mapsto (s_1, 1)$, $\mu(x_2) \mapsto (s_2, 1)$ et $\mu(x_3) \mapsto \mu(x_4) \mapsto (s_3, 1)$, n'est pas correcte (voir les capacités des salles). Une solution possible est donnée par ν, telle que $\nu(x_1) \mapsto (s_2, 1)$, $\nu(x_2) \mapsto (s_3, 1)$, $\nu(x_3) \mapsto (s_3, 2)$ et $\nu(x_4) \mapsto (s_4, 3)$.

Exemple 2 Avec les données suivantes :

— $S = \{s_1, \ldots\}$

— $c(s_i) \mapsto |E|$, $i \geq 1$

— $P = \{p_1\}$

— $X = \{x_1, \ldots\}$

— $T = |X|$

les horaires possibles sont définis par des fonctions μ, telles que $\mu(x_1) \mapsto (s, j_0)$, pour n'importe quelle salle $s \in S$ et telles que $\mu(x_i) \mapsto (s_i, j_i)$, $1 < i \leq |X|$ pour n'importe quelle salle $s_i \in S$ avec la contrainte suivante : $j_k < j_{k+1}$ pour $1 \leq k < |X|$.

3 Questions [20pts]

[2pts] Q1. Définir la notion de correction pour une fonction $\mu : X \to S \times \{1, \ldots, T\}$, i.e., définir en en Français un ensemble de contraintes que doit satisfaire μ permettant de dire si oui ou non μ est correcte. On vous laisse le choix des contraintes mais elle doivent bien sûr être "naturelles". Inutile d'inventer des contraintes du type "un étudiant doit avoir chaque examen dans une salle différente", contrainte qui n'aurait pas de sens...

[2pts] Q2. Écrire en langage mathématique ces contraintes, i.e., formaliser ces contraintes en utilisant les ensembles S,E,P etc., des quantificateurs, etc. Par exemple, si vous avez défini la contrainte qu'il existe des examens x_1, x_2 qui ont lieu en même temps (contrainte qui n'a pas de sens en pratique, mais c'est pour vous donner un exemple), alors vous écrivez :

$$\exists x_1, x_2 \in X, \exists t \in \{1, \ldots, T\}, \exists s_1, s_2 \in S \cdot \mu(x_1) = (s_1, t) \ \wedge \ \mu(x_2) = (s_2, t)$$

[2pts] Q3. Pour toute instance du problème d'emploi du temps (i.e. la donnée de $I = (S, E, P, X, c, a, b, T)$), construire une formule Φ_I en FNC de la logique propositionnelle telle que Φ_I est SAT si et seulement si il existe μ correcte, i.e. qui satisfait les contraintes que vous avez définies.

[4pts] Q4. Implémenter et tester sur les exemples, et proposer éventuellement d'autres exemples (notamment des exemples sans solution).

Format d'entrée/sortie Pour faciliter les tests de vos outils nous fixons un format d'entrée et sortie. D'abord, l'entrée va se faire par *stdin* et la sortie sera attendu dans *stdout*. Pour chaque ensemble on donne sa cardinalité et pour chaque fonction, pour chaque élément dans son domaine on donne l'image. Intuitivement on va encoder toutes les données sous forme de valeurs séparées par des point-virgules et de virgules (avec la possibilité d'écrire des commentaires). Par exemple,

```
2;3;1;10;50;2;2;2;1,2;1;1;2;
```

représente le fait que (i) $T = 2$, (ii) $|S| = 3$, (iii) $c(s_1) = 1$, (iv) $c(s_2) = 10$, (v) $c(s_3) = 50$, (vi) $|E| = 2$, (vii) $|P| = 2$, (viii) $|X| = 2$, (ix) $a(e_1) = \{1, 2\}$, (x) $a(e_2) = \{1\}$, (xi) $b(p_1) = \{1\}$, (xii) $b(p_2) = \{2\}$. Voici l'entrée annotée avec les valeurs qu'elle défini.

```
2;  // T = 2
3;  // |S| = 3
1;  // c(s1) = 1
10; // c(s2) = 10
50; // c(s3) = 50
2;  // |E| = 2
2;  // |P| = 2
2;  // |X| = 2
1,2;// a(e1) = {1,2}
1;  // a(e2) = {1}
1;  // b(p1) = {1}
2;  // b(p2) = {2}
```

Remarquez que la longueur de l'entrée dépend de valeurs dans l'entrée même.

La sortie doit être une liste de $|X|$ valeurs séparées par des point-virgules. Chaque valeur est une paire (salle, temps) séparées par des virgules (sans la possibilité d'écrire de commentaires). Exemple :

```
1,1;2,2;3,3;
```

nous donne une fonction μ telle que $\mu(x_1) \mapsto (s_1, 1)$, $\mu(x_2) \mapsto (s_2, 2)$ et $\mu(x_3) \mapsto (s_3, 3)$. S'il n'y a pas de fonction μ qui satisfait toutes les contraintes vous devez donner comme sortie juste 0.

[3pts] Q5. Maintenant, on ajoute au problème d'emploi du temps une durée pour chaque examen. Autrement dit, on se donne une fonction $d : X \to \mathbb{N}$ qui nous dit, pour chaque examen, combien d'unités de temps dure l'examen. Répondre aux même questions que 3 et 4.

Format d'entrée Les valeurs données par d sont ajoutées à la fin du format original.

[3pts] Q6. Dans la vraie vie on ne fait pas d'examens la nuit. On suppose qu'on a un ensemble de paires $I = \{(d_1^1, d_1^2), \ldots, (d_n^1, d_n^2)\}$ telle que $d_i^1 < d_i^2$, $1 \leq i \leq n$, telle que on ne peut pas avoir d'examens dans ces intervalles de temps. Ajouter les nouvelles contraintes induites par I à votre modélisation (i.e. dans Q3 et Q4). Implémenter et expliquer.

[2pts] Q7. On se donne un paramètre $k \in \mathbb{N}$. On voudrait qu'il existe au maximum k changements de salle pour tous les étudiants (i.e. la somme, pour tout étudiant e, du nombre de changements de salle de e, est bornée par k). Modifiez votre modélisation. Implémenter et expliquer.

[2pts Bonus] Q8. Utiliser le mode MAX-SAT dans MINISAT pour minimiser les changements de salle sans avoir k comme donnée.

[2pts] Q9. On veut maintenant s'assurer qu'entre deux examens de chaque étudiant, il y a au moins une unité de temps, sauf s'ils ont lieu dans la même salle. Modifier votre modélisation pour prendre en compte cette contrainte. Implémenter et expliquer.

[1pts Bonus] Q10. Proposer d'autres contraintes naturelles et donner leur modélisation.

[2pts Bonus] Q11. Est-ce que le problème d'emploi du temps est NP-dur ? Démontrer votre réponse.