UNIVERSITÉ LIBRE DE BRUXELLES

INFO-F-302 Informatique Fondamentale

Projet : Logique Propositionnelle et Utilisation de l'Outil MiniSat

A rendre le **20 Mai 2016** !

Attention : lire tout le sujet avant de commencer !

L'objectif de ce projet est de modéliser le problème du *orthogonal packing* 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 déjà un exemple du code nécessaire pour utiliser MiniSat.

`http://www.ulb.ac.be/di/info-f-302/2016/123path.tar.gz`

2 Le problème de orthogonal packing

Le problème consiste à essayer d'enchâsser un ensemble de rectangles dans la surface d'un grand rectangle.

Dans la suite on appelle *largeur* toute dimension verticale et *longueur* toute dimension horizontale. Considérons un rectangle R de largeur et longueur $n, m \in \mathbb{N}_{>0}$. On se donne aussi un ensemble $\{r_1, \ldots, r_k\}$ de k rectangles, chacun plus petit que celui avec taille $n \times m$, vus de manière générale comme deux fonctions $\mathcal{X}, \mathcal{Y} : \{1, \ldots, k\} \to \mathbb{N}_{>0}$. Intuitivement, la fonction $\mathcal{X}$ nous donne la largeur de chaque rectangle et $\mathcal{Y}$ sa longueur. Par exemple, la taille du deuxième rectangle est $\mathcal{X}(2) \times \mathcal{Y}(2)$.

Dans ce problème, il vous est demandé de trouver une assignation de chacun des k rectangles à une position valide dans R. Formellement, l'assignation est donnée par une fonction μ : $\{1, \ldots, k\} \to \{1, \ldots, m\} \times \{1, \ldots, n\}$: pour tout rectangle r_i, si $\mu(i) = (a, b)$, alors cela veut dire que le rectangle sera defini par les sommets (a, b), $(a, \mathcal{Y}(i) + b)$, $(\mathcal{X}(i) + a, \mathcal{Y}(i) + b)$, et $(\mathcal{X}(i) + a, b)$, où tous ces points sont contenus dans R. Les rectangles ne doivent pas se superposer.

Exemple 1 Si on vous donne

- $n = 4$,
- $m = 4$,
- $k = 2$,
- $\mathcal{X}(1) \mapsto 4$, $\mathcal{X}(2) \mapsto 1$,
- $\mathcal{Y}(1) \mapsto 1$, $\mathcal{Y}(2) \mapsto 4$.

Il n'y a pas d'assignation valide.

Exemple 2 Avec les données suivantes :

- $n = 4$,
- $m = 4$,
- $k = 3$,
- $\mathcal{X}(1) \mapsto 4$, $\mathcal{X}(2) \mapsto 2$, $\mathcal{X}(3) \mapsto 2$,
- $\mathcal{Y}(1) \mapsto 1$, $\mathcal{Y}(2) \mapsto 2$, $\mathcal{Y}(3) \mapsto 2$

une fonction μ possible est $\mu(1) \mapsto (0,0)$, $\mu(2) \mapsto (0,1)$, $\mu(3) \mapsto (2,2)$.

3 Questions [20pts]

[2pts] Q1. Écrire en langage mathématique les contraintes que doit satisfaire μ permettant de dire si oui ou non μ est correcte. I.e., formaliser ces contraintes en utilisant les données n, m, k, $\mathcal{X}$, et $\mathcal{Y}$, des quantificateurs, etc.

[4pts] Q2. Pour toute instance du problème orthogonal packing (i.e. la donnée de $I = (n, m, k, \mathcal{X}, \mathcal{Y})$), 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.

[5pts] Q3. Implémenter et tester sur les exemples 1 et 2, et proposer éventuellement d'autres exemples.

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 variable d'entrée ensemble on donne sa valeur et pour chaque fonction, pour chaque élément dans son domaine on donne l'image. Par exemple,

```
3
5
8
```

```
1  10  50
2  6   4
3  9   8
```

représente le fait que (i) $k = 3$, (ii) $n = 5$, (iii) $m = 8$, (iv) $\mathcal{X}(1) = 10$, (v) $\mathcal{Y}(1) = 50$, (vi) $\mathcal{X}(2) = 6$, (vii) $\mathcal{Y}(2) = 4$, (viii) $\mathcal{X}(3) = 9$, (ix) $\mathcal{Y}(3) = 8$. Remarquez que la longueur de l'entrée dépend de valeurs dans l'entrée même.

Vous pouvez utiliser des exemples d'ici : `http://or.dei.unibo.it/sites/or.dei.unibo.it/files/instances/2sp.zip` en remarquant qu'il faut ajouter une valeur pour la largeur de R.

La sortie doit être une liste de k pairs de valeurs Exemple :

```
1  0  0
2  6  3
3  9  8
```

nous donne une fonction μ telle que $\mu(1) \mapsto (0, 0)$, $\mu(2) \mapsto (6, 8)$ et $\mu(3) \mapsto (9, 8)$. S'il n'y a pas de fonction μ qui satisfait toutes les contraintes vous devez donner comme sortie juste 0.

[3pts] Q4. Maintenant, on suppose que R est un carré. Écrire un programme qui, étant donnés les rectangles $\{r_1, \ldots, r_k\}$ calcule la plus petite dimension du carré R admettant une solution.

[2pts] Q5. Écrire un programme qui étant donné un entier n, calcule la dimension du plus petit carré R admettant une solution pour les carrés $\{r_1, \ldots, r_n\}$ avec r_i de dimension $i \times i$ pour tout $i \leq n$. Voir `http://math.stackexchange.com/questions/1451362/method-for-optimally-packing-a-group-of-squares-from-1-x-1-to-n-x-n-into-a-l`.

[2pts] Q6. Maintenant, on ajoute au problème une troisième dimension (parallélépipède rectangle aussi appelés pavés droits). Autrement dit, on se donne une fonction $\mathcal{Z} : \{1, \ldots, k\} \to \mathbb{N}_{>0}$ qui nous donne la hauteur des pavés droits et une variable h qui nous donne la hauteur du R. Répondre aux même questions que 3 et 4.

Format d'entrée Les valeurs données par h et $\mathcal{Z}$ sont ajoutées au debut du format original et comme troisième colonne, respectivement.

Format de sortie Ajoutez une troisième colonne a la sortie aussi.

[2pts] Q7. Maintenant, avec trois dimensions, ajoutez des contraintes pour trouver uniquement des solutions qui ne fassent pas "flotter" les pavés droits dans l'espace.

[3pts Bonus] Q8. Nous remarquons que pour l'exemple 1 il existe une solution qui consiste à tourner (pivoter) un des rectangles. Ajoutons à l'output du problème la direction choisie pour chaque rectangle. I.e. la sortie est maintenant une liste de triplets (a, b, c) où $c \in \{0, 1\}$ est égal à 1 ssi nous voulons faire tourner le rectangle. Répondre aux mêmes questions que 3 et 4.

Format de sortie Ajoutez une autre colonne a la sortie.

[2pts Bonus] Q9. On se donne un paramètre $p \in \mathbb{N}_{>0}$. On voudrait qu'il y ait au minimum p unités de contact entre les rectangles et les bords de R. Modifiez votre modélisation. Implémenter et expliquer.

[2pts Bonus] Q10. Utiliser le mode MAX-SAT dans MINISAT pour maximiser le contact entre rectangles et le bord de R.