

Projet AMR : problème 2-SAT

Université Bordeaux I, Master-2 d'Informatique, 2009–2010

7 octobre 2009

A. Travail. Le projet est à réaliser par équipes de trois étudiants. Dans le cadre de ce projet il faut fournir :

1. Un rapport, de 5 à 10 pages (en pdf) présentant les algorithmes, les estimations de leurs complexités et les résultats d'expérimentation.
2. Les programmes réalisant tous les algorithmes, avec le Makefile, le fichier README expliquant comment travailler avec les programmes, et les fichiers test.
3. La soutenance de tous les projets AMR se tiendra début janvier 2010. Les modalités de la soutenance seront précisées ultérieurement.

Une archive contenant le rapport et les programmes doit être rendue avant le jeudi 12 novembre 2009. Il faut l'envoyer à *chacun* des quatre enseignants du cours (Paul Dorbec, Cyril Gavaille, Anca Muscholl, Alexandre Zvonkine).

B. Énoncé du problème. Le problème 2-SAT s'énonce comme suit :

Instance : n variables booléennes x_1, x_2, \dots, x_n et une formule booléenne ayant la forme

$$\varphi = \varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_m$$

où chaque clause φ_k est de la forme $\varphi_k = u_k \vee v_k$ où chaque littéral u_k et v_k est, soit une variable x_i , soit une négation $\neg x_i$ de telle variable.

Question : La formule φ est-elle satisfaisable? Si oui, trouver une valuation des variables x_1, \dots, x_n qui rend la formule vraie.

C. Idée de la solution. À chaque clause $\varphi_k = u_k \vee v_k$ on associe deux implications qui lui sont équivalentes : $\neg u_k \rightarrow v_k$ et $\neg v_k \rightarrow u_k$. On construit un graphe orienté $G = (V, E)$ ayant $2n$ sommets et $2m$ arcs : les sommets correspondent aux variables x_1, \dots, x_n et à leurs négations $\neg x_1, \dots, \neg x_n$, tandis que les arcs correspondent aux implications $\neg u_k \rightarrow v_k$ et $\neg v_k \rightarrow u_k$, $k = 1, \dots, m$. On cherche une valuation des "sommets" du graphe qui rend tous les "arcs" vrais.

1. Montrer que pour chaque composante fortement connexe du graphe G les valeurs booléennes correspondant aux sommets de cette composante doivent être égales. (Une *composante fortement connexe* d'un graphe orienté est un sous-graphe maximal pour lequel il existe un chemin *orienté* de tout sommet à tout autre sommet de ce sous-graphe.)
2. Présenter (et implémenter) un algorithme pour calculer les composantes fortement connexes d'un graphe orienté.

Indication : utiliser le *parcours en profondeur*.

Important : utiliser le même codage des formules booléennes que celui du SAT-solver `minisat` qui se trouve sur le site <http://minisat.se>. Voir aussi une documentation

<http://www.dwheeler.com/essays/minisat-user-guide.html>

3. Quelle est la complexité de cet algorithme? Justifier.
4. Expliquer comment vérifier la satisfaisabilité de φ .

Dans ce qui suit, on suppose que la formule φ soit satisfaisable et on cherche une valuation des variables x_i , $i = 1, \dots, n$, qui rend φ vraie.

D. Graphe des composantes. Soit $G = (V, E)$ un graphe orienté, et soient V_1, \dots, V_p ses composantes fortement connexes, $V = V_1 \cup \dots \cup V_p$. Le *graphe des composantes* est le graphe H ayant p sommets qui correspondent aux composantes du graphe G , et il existe un arc allant de V_i à V_j dans H si et seulement si il existe au moins un arc dans G allant d'un sommet appartenant à V_i à un sommet appartenant à V_j .

1. Montrer que le graphe H est un graphe orienté *sans cycles*.
2. Présenter (et implémenter) un algorithme qui calcule le graphe des composantes pour un graphe orienté donné.
3. Quelle est la complexité de cet algorithme? Justifier.

E. Recherche d'une valuation. Soit G le graphe de la partie **C** du projet, et soit H son graphe des composantes.

1. Démontrer les propriétés suivantes des composantes du graphe G :
 - Pour chaque composante U il existe une composante \overline{U} qui contient les négations de tous les littéraux appartenant à U (et rien d'autre).
 - Un arc $V_i \rightarrow V_j$ dans H existe si et seulement si il existe un arc $\overline{V_j} \rightarrow \overline{V_i}$.
2. L'algorithme de recherche d'une valuation (qu'il faudra présenter en détail dans le rapport et implémenter) procède comme suit :
 - Trouver une composante U qui n'a pas d'arcs entrants dans H .
 - Attribuer la valeur 0 à tous les littéraux de U , et la valeur 1 à tous les littéraux de \overline{U} .
 - Éliminer U et \overline{U} , ainsi que tous les arcs incidents, du graphe H , et continuer avec le graphe restant.

Remarque : cette solution peut ne pas être unique. Donner un exemple.
3. La complexité de l'algorithme ci-dessus est très sensible au choix des structures des données : avec un bon choix on peut le rendre de complexité linéaire. Il faudra donc présenter clairement le choix utilisé (même s'il n'est pas le meilleur possible).

F. Expérimentation.

1. Implémenter un générateur aléatoire des formules booléennes de type 2-SAT. Étudier la probabilité d'obtention d'une formule satisfaisable en fonction du rapport entre les paramètres m et n .
2. Comparer la performance, sur les instances du problème 2-SAT, de votre algorithme (qui est spécifique pour le 2-SAT) et du `minisat` (qui est générique pour toute formule booléenne ayant la forme normale conjonctive).