Satisfiabilité

Épreuve pratique d'algorithmique et de programmation
Concours commun des écoles normales supérieures

Durée de l'épreuve: 3 heures 30 minutes

Juin/Juillet 2010

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0
à l'emplacement prévu sur votre fiche réponse

Important.

Sur votre table est indiqué un numéro u_0 qui servira d'entrée à vos programmes. Les réponses attendues sont généralement courtes et doivent être données sur la fiche réponse fournie à la fin du sujet. À la fin du sujet, vous trouverez en fait deux fiches réponses. La première est un exemple des réponses attendues pour un $\widetilde{u_0}$ particulier (précisé sur cette même fiche et que nous notons avec un tilde pour éviter toute confusion!). Cette fiche est destinée à vous aider à vérifier le résultat de vos programmes en les testant avec $\widetilde{u_0}$ au lieu de u_0. Vous indiquerez vos réponses (correspondant à votre u_0) sur la seconde et vous la remettrez à l'examinateur à la fin de l'épreuve.

En ce qui concerne la partie orale de l'examen, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures!
Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, par exemple: $O(n^2)$, $O(n \log n)$,...
Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de ***tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie en annexe***. Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

1 Définitions

Un littéral est soit une variable booléenne v_i soit sa négation $\overline{v_i}$. Quand la valeur 1 est assignée à une variable booléenne v_i, sa négation $\overline{v_i}$ prend la valeur 0, et inversement. Une clause est un ensemble de littéraux. Elle représente la disjonction (ou logique) entre ces littéraux. Par exemple,

$$v_3 \vee \overline{v_5} \vee v_6 \vee v_9 \vee \overline{v_{10}}.$$

Une formule logique est un ensemble de clauses $(C_j)_j$. Elle représente leur conjonction (et logique).

$$(v_1 \vee v_2 \vee \overline{v}_3) \wedge (v_1 \vee v_2 \vee v_1).$$

L'objectif est d'assigner des valeurs 0 ou 1 à toutes les variables booléennes afin que toutes les clauses soient vraies, c'est-à-dire que chaque clause contienne au moins un littéral prenant la valeur 1. La formule est dite satisfiable s'il existe une telle assignation des variables. Elle est dite insatisfiable si, pour toute assignation des variables, il existe une clause fausse, c'est-à-dire une clause dont tous les littéraux prennent la valeur 0.

La taille d'une formule est définie comme la somme pour chaque clause du nombre de littéraux différents. Autrement dit, les littéraux redondants sont ignorés dans les clauses. Par exemple, la formule ci-dessus a une taille $3 + 2 = 5$.

2 Génération de formules logiques

Considérons la suite d'entiers $(u_k)_k$ définie par

$$u_{k+1} = 15091 \times u_k \mod 64007.$$

On s'assurera de précalculer et stocker suffisamment de valeurs de u_k de manière à pouvoir y accéder en temps constant par la suite.

Question 1 *Quelles sont les valeurs de* ***a)*** u_{17}, ***b)*** u_{42} *et* ***c)*** u_{8000} *?*

La formule $F(p, q, r)$ contient q clauses de r littéraux (potentiellement redondants). Ces littéraux sont choisis parmi p variables et leur négation. Plus formellement, pour tout couple (j, k) tel que $0 \leq j < q$ et $0 \leq k < r$, on définit l'entier $\alpha_{j,k} = u_{j\times r+k} \mod p$. Si $u_{j\times r+k} < 32003$, le littéral $v_{\alpha_{j,k}}$ appartient à la clause C_j. Dans le cas contraire, c'est $\overline{v_{\alpha_{j,k}}}$ qui appartient à la clause C_j.

Pour les questions suivantes (exceptées celles de la Section 6), on s'intéressera à la satisfiabilité des formules $F(4, 15, 2)$, $F(3, 30, 3)$, $F(8, 60, 3)$ et $F(10, 100, 4)$.

Question 2 *Quelle est la taille des formules logiques suivantes ?* ***a)*** $F(4, 15, 2)$, ***b)*** $F(3, 30, 3)$, ***c)*** $F(8, 60, 3)$, ***d)*** $F(10, 100, 4)$.

Question à développer pendant l'oral : Quelles structures de données ont été utilisées pour représenter les formules ? Quelle est la complexité de l'algorithme de création ?

3 Clauses triviales

Si une clause contient à la fois une variable v_i et sa négation $\overline{v_i}$, elle est trivialement vraie et elle peut être supprimée. Elle n'apporte en effet aucune information sur les valeurs que peuvent prendre les variables.

Question 3 *Combien de clauses triviales sont retirées des formules ?* ***a)*** $F(4,15,2)$, ***b)*** $F(3,30,3)$, ***c)*** $F(8,60,3)$, ***d)*** $F(10,100,4)$.

4 Variables forcées et propagation

Note : les clauses triviales ont maintenant été retirées des formules. En particulier, elles ne doivent pas être prises en compte lors du calcul de la taille des formules simplifiées.

Si une clause se réduit à une seule variable v_i, la valeur 1 doit être attribuée à v_i pour espérer satisfaire la formule. De même, une clause $\overline{v_i}$ force la valeur de v_i à 0.

Question 4 *Combien de variables ont des valeurs forcées dans les formules ?* ***a)*** $F(4,15,2)$, ***b)*** $F(3,30,3)$, ***c)*** $F(8,60,3)$, ***d)*** $F(10,100,4)$.

Une valeur forcée peut ensuite être propagée dans toutes les clauses où la variable (ou sa négation) apparaît. Si le littéral présent dans la clause vaut 1, elle est trivialement vraie et peut donc être supprimée de la formule. Si le littéral vaut 0, il peut être supprimé de la clause.

Question 5 *Quelle est la taille des formules une fois qu'a été propagée la première valeur forcée (celle qui constitue la clause C_j d'indice j minimal) ?* ***a)*** $F(4,15,2)$, ***b)*** $F(3,30,3)$, ***c)*** $F(8,60,3)$, ***d)*** $F(10,100,4)$. *Note : il se peut qu'aucune clause ne soit réduite à un seul littéral ; la taille de la formule reste alors la même.*

Parce que les littéraux valant 0 sont retirés, certaines clauses peuvent se trouver réduites à un unique littéral. Cela fait donc apparaître de nouvelles valeurs forcées qui vont pouvoir à leur tour être propagées dans la formule, et ainsi de suite.

Si les simplifications ont supprimé toutes les clauses de la formule, cela signifie qu'il existe une assignation des variables les rendant toutes vraies ; la formule initiale était donc satisfiable. Par contre, si les simplifications font disparaître tous les littéraux d'une clause, cette clause est fausse pour toute assignation des variables ; la formule initiale était donc insatisfiable. Dans ces deux cas, la formule simplifiée est considérée comme étant de taille nulle.

Question 6 *Quelle est la taille des formules une fois simplifiées le plus possible ?* ***a)*** $F(4,15,2)$, ***b)*** $F(3,30,3)$, ***c)*** $F(8,60,3)$, ***d)*** $F(10,100,4)$. *Pour les formules de taille nulle, indiquer si elles sont satisfiables ou non.*

Question à développer pendant l'oral : Quelle est la complexité de l'algorithme ?

5 Résolution complète

Les simplifications précédentes ne permettent pas toujours de décider si la formule est ou non satisfiable. Une manière de progresser est alors de choisir une variable apparaissant dans la formule et de tester séparément ses deux valeurs possibles. Comme précédemment, la valeur testée est propagée dans la formule, ce qui peut forcer les valeurs d'autres variables qui sont à leur tour propagées. On obtient ainsi deux nouvelles formules de taille plus petite et contenant moins de variables différentes. Ce processus peut être appliqué récursivement à chacune des deux formules.
Si les deux formules obtenues en propageant les valeurs 0 et 1 pour la variable choisie sont insatisfiables, alors la formule initiale est insatisfiable. Si l'une des formules est satisfiable, alors la formule initiale est satisfiable.

Question 7 *Pour chacune des formules, indiquer si elle est satisfiable ou non.* ***a)*** $F(4,15,2)$, ***b)*** $F(3,30,3)$, ***c)*** $F(8,60,3)$, ***d)*** $F(10,100,4)$.

Question à développer pendant l'oral : Quelle est la complexité de l'algorithme ?

6 Cas particulier : clauses à deux littéraux

Si l'on fait abstraction des simplifications, la méthode précédente revient donc à tester toutes les valeurs possibles des variables apparaissant dans une formule. Dans le cas où les clauses n'ont pas plus de deux littéraux chacune, il existe une méthode plus efficace.

Considérons une clause $l_a \vee l_b$. Si le littéral l_a prend la valeur 0, il faut que l_b prenne la valeur 1 pour que la clause soit vraie. Cette condition peut être représentée par l'implication $\overline{l_a} \to l_b$: si $\overline{l_a}$ à la valeur 1, l_b l'a aussi. Les deux littéraux jouant des rôles symétriques, l'implication inverse $\overline{l_b} \to l_a$ est elle aussi nécessaire. Ces deux implications sont suffisantes pour représenter la clause.
Une clause à un seul littéral l est quant à elle interprétée par l'implication $\overline{l} \to l$.

Pour les questions suivantes, on s'intéressera à la satisfiabilité des formules $G(3,6)$, $G(5,20)$, $G(30,40)$ et $G(3000,3300)$ avec $G(p,q) = F(p,q,2)$.

Question 8 *Pour chacune des formules, combien de littéraux différents sont directement impliqués par* $\overline{v_{\alpha_{3,1}}}$ *?* ***a)*** $G(3,6)$, ***b)*** $G(5,20)$, ***c)*** $G(30,40)$, ***d)*** $G(3000,3300)$.

Un <u>chemin</u> menant de l_1 à l_n est un ensemble d'implications $l_1 \to l_2, l_2 \to l_3, \ldots, l_{n-1} \to l_n$ fournies par les clauses de la formule. On le notera $l_1 \to l_2 \to \ldots \to l_n$. On peut remarquer que, s'il existe un chemin de l_1 à l_n, assigner la valeur 1 à l_1 force la valeur de l_n à 1.

Supposons maintenant que la formule contienne des chemins de v_i à $\overline{v_i}$ et de $\overline{v_i}$ à v_i. Si l'on assigne la valeur 1 à v_i, le premier chemin mène à une contradiction puisqu'il force $\overline{v_i}$ à 1 aussi. Si l'on assigne la valeur 0, c'est le deuxième qui mène à une contradiction. La formule est donc insatisfiable.
S'il n'existe qu'un seul des deux chemins, alors la valeur de v_i est forcée : 0 dans le premier cas, 1 dans le second. Enfin, si aucun des deux chemins n'existe, la satisfiabilité de la formule ne dépend pas de la valeur assignée à v_i.

Question 9 *Quelles valeurs en déduire pour la variable* $v_{\alpha_{0,0}}$ *dans les formules suivantes ?* ***a)*** $G(3,6)$, ***b)*** $G(5,20)$, ***c)*** $G(30,40)$, ***d)*** $G(3000,3300)$.

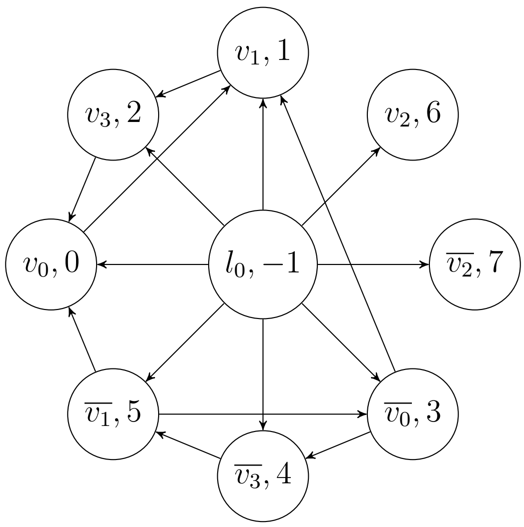
Question à développer pendant l'oral : Quelle est la complexité de l'algorithme ?

La formule initiale contient au plus $2p$ littéraux différents : les v_k et $\overline{v_k}$ avec $0 \leq k < p$. En fonction des tirages aléatoires, certains littéraux peuvent ne pas apparaître explicitement dans une formule ; on supposera néanmoins leur existence.
On ajoute maintenant un littéral spécial l_0 qui n'est aucun des $2p$ littéraux précédents. Ce littéral est construit de telle sorte qu'il implique chacun d'entre eux. Aucune autre que ces $2p$ implications n'est ajoutée au problème. Ce nouveau littéral l_0 ne change donc pas la satisfiabilité ou non de la formule initiale. En effet, il suffit de lui assigner la valeur 0 pour faire neutraliser les $2p$ nouvelles implications.

On attribue des entiers consécutifs $N(l)$ aux littéraux de la formule. On appellera chemin croissant menant de l_1 à l_n un chemin $l_1 \rightarrow \ldots \rightarrow l_n$ tel que $N(l_1) < N(l_2) < \ldots < N(l_n)$.

Le littéral l_0 reçoit le numéro $N(l_0) = -1$. Les autres littéraux reçoivent des numéros consécutifs tous différents à partir de 0 de telle sorte que la propriété suivante soit vraie : pour tous littéraux l_a, l_b et l_c tels que $l_a \rightarrow l_c$ et $N(l_a) < N(l_b) < N(l_c)$, il existe un chemin croissant menant de l_a à l_b.
Pour obtenir un choix déterministe des numéros, on s'assurera, quand plusieurs littéraux peuvent porter les mêmes numéros, d'attribuer le plus petit numéro possible au littéral le plus petit. L'ordre sur les littéraux est arbitrairement défini comme suit : $\forall i < j,\ v_i \prec \overline{v_i} \prec v_j \prec \overline{v_j}$.

Par exemple, voici l'ensemble des implications associées à la formule $(v_0 \vee v_1) \wedge (v_0 \vee \overline{v_3}) \wedge (\overline{v_0} \vee v_1) \wedge (\overline{v_1} \vee v_3)$ ainsi que les valeurs de N pour chacun des littéraux :



Question 10 *Quels sont les numéros* $N(l)$ *des littéraux suivants dans la formule* $G(20,80)$ *?* ***a)*** $v_{\alpha_{0,0}}$, ***b)*** $v_{\overline{\alpha_{1,0}}}$, ***c)*** $v_{\alpha_{3,1}}$.

Soit $\mathcal{T}(l)$ l'ensemble des littéraux accessibles depuis l par un chemin croissant, ainsi que l lui-même. On appellera $\mathcal{T}(l)$ le sous-arbre enraciné en l. On dira de $\mathcal{T}(l)$ qu'il est borné si, pour tous littéraux l_a et l_b tels que $l_a \in \mathcal{T}(l)$ et $l_a \to l_b$, on a $N(l) \leq N(l_b)$. Il sera borné minimal s'il est borné et si aucun $\mathcal{T}(l')$ n'est borné pour $l' \in \mathcal{T}(l) - \{l\}$.

Question à développer pendant l'oral : Justifier les propriétés suivantes :
- Pour tout littéral l_a dans $\mathcal{T}(l)$, si on a $l_a \to l_b$ et $N(l_a) \leq N(l_b)$, alors l_b appartient aussi à $\mathcal{T}(l)$.
- Soit $\mathcal{T}(l)$ un sous-arbre borné. Tout littéral l' accessible par un chemin partant de l appartient à $\mathcal{T}(l)$.
- Soit $\mathcal{T}(l)$ un sous-arbre borné minimal. Pour tout littéral l' accessible par un chemin partant de l, il existe un chemin menant de l' à l.
- Soit $\mathcal{T}(l)$ un sous-arbre borné minimal. Si v_i et $\overline{v_i}$ sont deux littéraux de $\mathcal{T}(l)$, alors la formule n'est pas satisfiable.

Question 11 *Quel est le littéral l de numéro $N(l)$ minimal pour lequel $\mathcal{T}(l)$ est borné minimal pour les formules suivantes ?* ***a)*** $G(3,6)$, ***b)*** $G(5,20)$, ***c)*** $G(30,40)$, ***d)*** $G(3000,3300)$.

Question à développer pendant l'oral :
- Si un sous-arbre borné minimal $\mathcal{T}(l)$ ne contient pas à la fois v_i et $\overline{v_i}$, prouver qu'il est possible de supprimer toutes les implications concernant les littéraux de $\mathcal{T}(l)$ sans changer la satisfiabilité de la formule.
- En déduire un algorithme décidant si une formule ne contenant que des clauses à deux littéraux au plus est satisfiable.

Question 12 *Pour chacune des formules, indiquer si elle est satisfiable ou non.* ***a)*** $G(3,6)$, ***b)*** $G(5,20)$, ***c)*** $G(30,40)$, ***d)*** $G(3000,3300)$.

Question à développer pendant l'oral :
- Quel algorithme permettrait d'attribuer les numéros $N(l)$ avec une complexité temporelle en $O(p+q)$? Remarque : le surcoût causé par le choix déterministe des numéros pourra être ignoré, si nécessaire.
- Qu'en est-il de la recherche d'un sous-arbre borné minimal ?
- Qu'en est-il de la satisfiabilité d'une formule ?

Fiche réponse type: Satisfiabilité

$\widetilde{u_0}$: 241

Question 1

a) 18325

b) 37695

c) 62662

Question 2

a) 27

b) 75

c) 159

d) 373

Question 3

a) 1

b) 11

c) 9

d) 27

Question 4

a) 3

b) 1

c) 1

d) 0

Question 5

a) 16

b) 13

c) 105

d) 272

Question 6

a) 0 (unsat)

b) 0 (sat)

c) 0 (unsat)

d) 272

Question 7

a) unsat

b) sat

c) unsat

d) sat

Question 8

a) 1

b) 4

c) 4

d) 1

Question 9

a) 0 ou 1

b) aucune

c) 1

d) 1

Question 10

a) 9

b) 10

c) 23

Question 11

a) $\overline{v_0}$

b) $\overline{v_3}$

c) $\overline{v_9}$

d) v_{556}

Question 12

a) sat

b) unsat

c) sat

d) unsat