Algorithmique I
Année 2008-2009

A. Benoit, B. Depardon
C. Mouilleron, C. Rezvoy

TD n°9

1 Variantes de 3-SAT

Montrer la $\mathcal{NP}$-complétude des deux variantes de 3-SAT suivantes :

Question 1.1 3-SAT NAE (*not all equal*), où l'on impose que les trois littéraux de chaque clause ne soient pas tous à la même valeur.

Solution : *Évidemment 3-SAT NAE est NP.*

Pour montrer qu'il est NP-complet, et contrairement à toute attente, nous allons effectuer une réduction à partir de 3-SAT de la manière suivante :

Pour chaque clause, $a \vee b \vee c$ du problème 3-SAT, on créé les instances $a \vee b \vee x$ et $c \vee \overline{x} \vee f$ où f est global (ie f est la même pour chaque clause créée ainsi)

S'il existe une instanciation des variables qui met toutes les clauses à vrai, il en existe aussi une pour sa réduction, pour cela, il suffit de prendre $x = \overline{a \vee b}$, avec les notations ci-dessus, de plus si a ou b sont à vrai, x sera à faux, et donc pour la seconde clause sera mis a vrai par $\overline{x}$, on fini en prenant f toujours à faux, on a donc une bonne instanciation pour 3-SAT NAE.

Réciproquement, si on a un bonne instanciation du problème 3-SAT NAE (réduction du problème 3-SAT) il existe un bonne instanciation pour le problème 3-SAT d'origine. En effet si f est à faux soit c est à vrai et donc la clause $a \vee b \vee c$ est vrai, soit c est à faux donc $\overline{x}$ est à vrai, x est donc à faux, d'où soit a soit b est à vrai et donc la clause $a \vee b \vee c$ est à vrai, par contre si f est à vrai, on prend toutes les variables du problème NAE et on prend leur négation, sans changer l'instance du problème. Cette instanciation, met toujours NAE à vrai, car comme "not all equal" une variable fausse au moins par clause qui est à vrai, et on recommence comme ci-dessus.

Ce qui précède montre que tout problème 3-SAT peut se réduire polynomialement (on multiplie le nombre de clause par deux, et on rajoute une variable plus une par clauses de départ) en un problème 3-SAT NAE équivalent, <u>3-SAT NAE est donc NP-complet</u>. □

Question 1.2 3-SAT OIT (*one in three*), où l'on impose qu'exactement un littéral soit à VRAI dans chaque clause.

Solution : *On fait comme le cas précédent, sauf qu'à partir de $a \vee b \vee c$ on créé les trois clauses $a \vee x \vee y$, $\overline{b} \vee x \vee x'$ et $\overline{c} \vee y \vee y'$*

Pour commencer on va montrer que s'il existe une bonne instanciation pour le problème 3-SAT de départ, il en existe une pour sa réduction à 3-SAT OIT.

C'est assez simple à montrer une fois qu'on connaît la transformation, par contre elle est un peu longue à trouver. En effet, pour chaque clause de $a \vee b \vee c$ 3-SAT , si a est à vrai on met les variables x et y (avec les notations précédentes) à faux, et on donne la valeur de b à x' et celle de c à y' (a, b et c restent inchangées d'un problème à l'autre) ; et si a est à faux, alors soit b, soit c est à vrai. Comme la construction est symétrique, on supposera que b est à vrai,

x' et y sont donc mis à faux, donc x doit être mis à vrai, et y' à c. On a alors exactement une variable à vrai dans les trois instances. On répète ce procédé pour chaque instance, et vu qu'on ne touche à aucune des variables directement associées à celle du problème 3-SAT, il n'y a pas de problème.

Dans l'autre sens, on va montrer que si on a une instanciation qui permet de répondre oui au problème 3-SAT OIT obtenu à partir d'un problème 3-SAT par la construction explicitée plus haut alors on en a une pour ce problème 3-SAT. En effet on ne peut avoir aucune clause $a \vee b \vee c$ à faux car cela signifie que a, b et c sont faux, ce qui signifie que dans les clauses du 3-SAT OIT équivalentes on a x et x' à faux à cause de $\overline{b}$ (dans la deuxième clause), et de même, on a y et y' à faux à cause de $\overline{c}$ (dans la troisième clause), ce qui donne que la première clause est fausse, il y a donc contradiction.

On a donc équivalence entre entre les deux problèmes. Il y a un certificat pour un problème 3-SAT si et seulement il y en a une pour sa réduction vue précédemment, en un problème 3-SAT OIT. De plus la réduction est polynomiale (on multiplie le nombre de clauses par trois, et on rajoute six variables par clause de départ).

Ce qui précède montre que tout problème 3-SAT peut se réduire polynomialement en un problème 3-SAT OIT équivalent, <u>3-SAT OIT est donc un problème NP-complet.</u>

Remarque :*une fois qu'on à montré que 3-SAT OIT est NP-complet, le problème* **??** *deviens plus facile, car on démontre que SUBSET-SUM est NP-complet en faisant la même construction, sans les s_i et s'_i, et en prenant t=1. . . 11 . . . 1.* □

2 Encore une variante : SAT-N

On appelle SAT-N le problème SAT restreint aux formules qui n'ont pas plus de N occurrences de la même variable.

Question 2.1 Montrer que SAT-3 est au moins aussi dur que SAT. En déduire que pour tout $N \geq 3$, SAT-N est $\mathcal{NP}$-complet.

Solution : *SAT-3 est trivialement NP.*

Pour montrer qu'il est NP-complet, on part d'un problème de SAT et pour chaque littéral x_i apparaissant k fois avec k avec $k > 3$, on applique l'algorithme suivant : on remplace les occurrences de x_i par k nouveaux littéraux $y_{i_1}, \dots, y_{i_k}$, et on rajoute les clauses $y_{i_1} \vee \overline{y_{i_2}}, \dots, y_{i_{k-1}} \vee \overline{y_{i_k}}$ et $y_{i_k} \vee \overline{y_{i_1}}$.

Les nouveaux littéraux apparaissent alors 3 fois chacun (une fois à la place de x_i, et deux fois dans les nouvelles clauses). De plus les nouvelles clauses entraînent que les y_{i_j} prennent tous la même valeur : si y_{i_j} est vrai, alors $y_{i_{j-1}}$ également, et inversement, si y_{i_j} est faux, alors $y_{i_{j+1}}$ doit l'être aussi (en toute rigueur on devrait rajouter des modulos k), et comme les clauses sont cycliques, on obtient que tous ces littéraux ont la même valeur.

Enfin, si n est le nombre de littéraux, et m le nombre de clauses, le nouveau problème obtenu comporte au plus nm littéraux et $(m + nm)$ clauses, donc une instance de SAT se réduit polynomialement en une instance SAT-3 équivalente, et comme SAT-3 est NP, <u>SAT-3 est NP-complet.</u>

Et comme pour tout $N \geq 3$, SAT-3 est un cas particulier de SAT-N, <u>SAT-N est NP-complet.</u>
□

Question 2.2 Soit x une variable apparaissant dans une formule F de SAT-2. Trouver une formule équivalente à F et dans laquelle la variable x n'apparaisse plus. En déduire un algorithme polynomial pour SAT-2.

Solution : *Soit x une variable d'un système de clauses F de SAT-2, on transforme F selon les cas de sorte que le système obtenu soit satisfiable si et seulement si le système initial l'était :*

- *si x (ou $\overline{x}$) n'apparaît qu'une seule fois, on supprime la clause dans laquelle il apparaît, car on est libre de mettre x (ou $\overline{x}$), donc toute la clause à vrai.*
- *si x (ou bien $\overline{x}$) apparaît 2 fois dans une même clause, ou dans 2 clauses distinctes, on supprime la ou les clauses, pour la même raison.*
- *si x et $\overline{x}$ apparaissent dans la même clause, on supprime la clause car elle est toujours à vrai.*
- *si x et $\overline{x}$ apparaissent dans 2 clauses distinctes, $x \vee C_1$ et $\overline{x} \vee C_2$, alors le système de clauses est satisfiable si et seulement si au moins une des deux clauses C_1 ou C_2 peut être mise à vrai en même temps que les autres clauses (pour l'autre on choisit x pour compléter), ainsi :*
  - *$\rightarrow$ si $C_1 = C_2 = \emptyset$ alors le système n'est pas satisfiable.*
  - *$\rightarrow$ sinon on remplace les 2 clauses par la nouvelle clause $C_1 \vee C_2$.*

On a ainsi défini un algorithme en n étapes (où n est le nombre de littéraux), pour lequel on décide à chaque étape de la valeur d'un littéral avec une complexité m (où m est le nombre de clauses) : <u>on peut donc résoudre SAT-2 en $O(nm)$</u>. □

3 Complexité de SUBSET-SUM

Montrer que le problème suivant SUBSET-SUM est $\mathcal{NP}$-complet.

SUBSET-SUM
Instance : un ensemble fini S d'entiers positifs et un entier objectif t.
Question : existe-t-il un sous-ensemble $S' \subseteq S$ tel que $\sum_{x \in S'} x = t$?

Indication : vous pouvez par exemple effectuer une réduction à partir de 3-SAT. A partir d'un ensemble de clauses $C_0, ..., C_{m-1}$ sur les variables $x_0, ..., x_{n-1}$, considérer S l'ensemble des entiers $v_i = 10^{m+i} + \sum_{j=0}^{m-1} b_{ij} 10^j$ et $v'_i = 10^{m+i} + \sum_{j=0}^{m-1} b'_{ij} 10^j$, $0 \leq i \leq n-1$, où b_{ij} (resp. b'_{ij}) vaut 1 si le littéral x_i (resp. $\overline{x_i}$) apparaît dans C_j et 0 sinon, et des entiers $s_j = 10^j$ et $s'_j = 2.10^j$, $0 \leq j \leq m-1$. Trouver alors un entier objectif t tel qu'il existe un sous-ensemble $S' \subseteq S$ de somme t si et seulement si l'ensemble initial de clauses est satisfiable. Conclure. Quels autres entiers auraient aussi marché ?

Solution : *SUBSET-SUM est évidemment NP.*

Pour montrer qu'il est NP-complet nous allons suivre l'indication et effectuer une réduction à partir de 3-SAT. Pour ce faire nous allons commencer en transformant les données de 3-SAT en chiffres.

Soit $C_0, \ldots, C_{m-1}$ les clauses et $x_0, \ldots, x_{n-1}$ les variables de 3-SAT. On construit v_i et v'_i de la manière suivante :

$$v_i = 10^{m+i} + \sum_{j=0}^{m-1} (b_{ij} 10^j) \text{ et } v'_i = 10^{m+i} + \sum_{j=0}^{m-1} (b'_{ij} 10^j),\ 1 \leq i \leq n-1 \text{ avec}$$

$$b_{ij} = \begin{cases} 1 & \textit{si } x_i \textit{ apparaît dans } C_j \\ 0 & \textit{sinon} \end{cases} \quad \textit{et } b'_{ij} = \begin{cases} 1 & \textit{si } \overline{x_i} \textit{ apparaît dans } C_j \\ 0 & \textit{sinon} \end{cases}$$

Pour mieux comprendre ce qu'on fait, on peut imaginer qu'on crée des $(m+n)$-upplets dont les n premières cases il n'y à que des zéros sauf à un endroit i qui permet de savoir de quelle variable on parle, que ce soit x_i ou $\overline{x_i}$, et les m suivants permettent de savoir si la variable est dans les clauses correspondantes.

Entier	*n variables* $x_{n-1}\ x_{n-2}\ \dots\ x_0$	*m clauses* $C_{m-1}\ C_{m-2}\ \dots\ C_0$	*Commentaires*
v_0	$00\dots1$	$\dots1\dots1\dots$	*1 : clauses dans lesquelles x_0 apparaît*
v_{n-2}	$01\dots0$	$\dots1\dots$	*1 : clauses dans lesquelles x_{n-2} apparaît*
v'_0	$0\dots1$	$\dots1\dots1\dots$	*1 : clauses dans lesquelles $\overline{x_0}$ apparaît*
v'_i	$\dots010\dots$	$\dots1\dots1\dots$	*1 : clauses dans lesquelles $\overline{x_i}$ apparaît*
s_0	$00\dots0$	$00\dots1$	*Uniquement 1 dans colonne C_0*
s_i	$00\dots0$	$\dots1\dots$	*Uniquement 1 dans colonne C_i*
s'_1	$00\dots0$	$00\dots02$	*Uniquement 2 dans colonne C_0*
s'_i	$00\dots0$	$\dots020\dots$	*Uniquement 2 dans colonne C_i*
t	$11\dots1$	$44\dots4$	*Entier objectif*

Désormais on va sommer certains v_i et certains v'_i, en fait on somme n entiers, un par littéral, $T = \sum_{j=0}^{n-1}(w_i)$ où w_i vaut soit x_i soit $\overline{x_i}$. T est un nombre qui commence par n "1" et est suivit de m chiffres compris entre 0 et 3 (comme on est parti de 3-SAT il y a au plus 3 variables par clause et ces m chiffres correspondent exactement au nombre de littéraux qui mettent cette clause à vrai). Si on part d'une solution du problème 3-SAT, et qu'on prend v_i si x_i est vrai et v'_i si x_i est faux, on remarque que c'est m chiffres sont compris entre 1 et 3.

A partir de là on se demande comment faire la différence entre les sous-ensembles donnant une somme qui contient un "0" de ceux donnant une somme sans "0" qui sont exactement les sous-ensembles correspondant à un certificat du problème 3-SAT par la bijection évidente $v_i \in S \leftrightarrow x_i$ est vrai.

C'est à cela que servent les $s_j = 10^j$ et $s'j = 2.10^j$. En prenant $t = 1\dots14\dots4$ et en ne réfléchissant que sur les m chiffres de droite de T on voit que :
- *à partir de 3 on peut avoir 4,$3+1$*
- *à partir de 2 on peut avoir 4,$2+2$*
- *à partir de 1 on peut avoir 4,$1+1+2$*
- *à partir de 0 on ne peut avoir 4.*

(Cette réflection est la même qui si on considère les nombres comme des $n+m$-upplets)

Le problème 3-SAT est donc réduit en SUBSET-SUM avec $S = \{v_i | 1 \leq i \leq n-1\} \cup \{v'_i | 1 \leq i \leq n-1\} \cup \{s_j | 1 \leq j \leq m-1\} \cup \{s_j | 1 \leq j \leq m-1\}$ et $t = 1\dots14\dots4$

On a ainsi que, si on trouve une solution du problème 3-SAT, on a une solution du problème SUBSET-SUM en prenant la bijection précédente, et en complétant avec les $s_j = 10^j$ et $s'j = 2.10^j$.

Réciproquement, si on trouve un sous-ensemble S' de S tel que si $\sum_{x \in S'}(x) = t = 1\dots14\dots4$, en prenant $\tilde{S} = S \cap (\{v_i | 1 \leq i \leq n-1\} \cup \{v'_i | 1 \leq i \leq n-1\})$ on a nécessairement $T = \sum_{x \in \tilde{S}}(x)$ qui est nombre commençant par n "1" et suivit de m chiffres compris entre 1 et 3 (ni 0 ni 4 : au plus 3 peut venir des s_i et s'_i, il y a nécessairement un 1 qui vient des v_i ou v'_i), ce qui montre que *comme aucun "0" n'apparaît, les instances de 3-SAT peuvent toutes êtres mises à vrai, trouvé à l'aide de la bijection citée plus haut, donc à partir du certificat de ce problème SUBSET-SUM on retrouve un certificat du problème 3-SAT de départ.*

La réduction précédente est polynomiale car on passe de n variables et m clauses à $2n+2m$ entiers de m chiffres, SUBSET-SUM est donc NP-complet.

Remarque : *On peut faire le même travail dans toutes les autres bases qui n'introduisent pas de problèmes dus aux phénomènes de retenue.*

Remarque : *On peut aussi remplacer les "4" dans t par n'importe quel autre chiffre supérieur à 4, à condition de rajouter des "s" en nombre suffisant.*

Remarque : *On peut aussi faire des $\tilde{s}_i = 10^{m+i}$ sans changer t, ça remplace les variables pour lesquelles on peut choisir n'importe que booléen.* □

4 Sous-chaîne transitive

Dans un graphe orienté, on dit que $\{x_1, \ldots, x_k\}$ est une chaîne transitive de longueur k si et seulement si pour tout $1 \leq i < j \leq k$, $(x_i, x_j) \in E$. Montrer que le problème suivant est $\mathcal{NP}$-complet.

Sous-chaîne transitive :
Instance : Un graphe orienté $D = (V, E)$.
Question : D contient-il une sous-chaîne transitive de longueur au moins $\left\lfloor \frac{|V|}{2} \right\rfloor$?

Indication : Vous pouvez par exemple effectuer une réduction à partir de 3-SAT. Si on se donne un ensemble de clauses $C_1, ..., C_k$, avec $C_i = (x_i^1 \vee x_i^2 \vee x_i^3)$, on peut construire une instance du problème de sous-chaîne transitive en posant $V = \{C_0\} \bigcup_{1 \leq i \leq k} \{C_i, x_i^1, x_i^2, x_i^3\}$ et en choisissant avec soin les arêtes à mettre dans E.

Solution : *On construit un graphe orienté $D = (V, E)$ avec pour sommets $V = \{C_0\} \bigcup_{1 \leq i \leq k} \{C_i, x_i^1, x_i^2, x_i^3\}$. Les 3 littéraux d'une clause C_i sont indépendants. Pour tout $1 \leq i \leq k$ et tout $1 \leq j \leq 3$, (x_i^j, c_i) est un arc de E. De plus, pour $1 \leq i < j \leq k$ et $1 \leq h, h' \leq 3$, $(x_i^h, x_j^{h'})$ appartient à E si et seulement si $x_i^h \neq \overline{x}_j^{h'}$. Finalement, pour tout $v \neq C_0$, on ajoute les arcs (C_0, v). La figure 1 présente un exemple d'une telle construction pour l'instance $(x_1^1 \vee x_1^2 \vee x_1^3) \wedge (x_2^1 \vee x_2^2 \vee x_2^3) \wedge (x_3^1 \vee x_3^2 \vee x_3^3)$ où $x_1^1 = \overline{x}_2^2$, $x_1^3 = \overline{x}_3^3$ et $x_2^3 = \overline{x}_3^1$.*

On va maintenant prouver qu'une instance $\mathcal{I}$ de 3-SAT est satisfiable si et seulement si le graphe dirigé associé contient une chaîne transitive de longueur égale à $\left\lfloor \frac{|V|}{2} \right\rfloor$.

On suppose $\mathcal{I}$ satisfiable. On a donc au moins un littéral l_i par clause qui a été satisfait, et pour tout $i \neq j$, $l_i \neq \overline{l_j}$. Il existe alors une chaîne dans D de C_0 à C_k passant par les l_i, telle que le sous-graphe induit par les sommets de la chaîne soit transitif. Une telle chaîne contient $2k+1$ sommets, donc sa longueur est $2k$, qui est égale à $\left\lfloor \frac{|V|}{2} \right\rfloor$ puisque $|V| = 4k+1$.

Inversement, si une chaîne transitive de taille $\left\lfloor \frac{|V|}{2} \right\rfloor$ existe dans G, alors elle doit contenir chaque C_i $(0 \leq i \leq k)$ plus un littéral l_i par clause. La transitivité nous assure que pour aucun $i \neq j$ nous n'avons $l_i = \overline{l_j}$. On a donc bien une assignation qui satisfasse l'instance. □

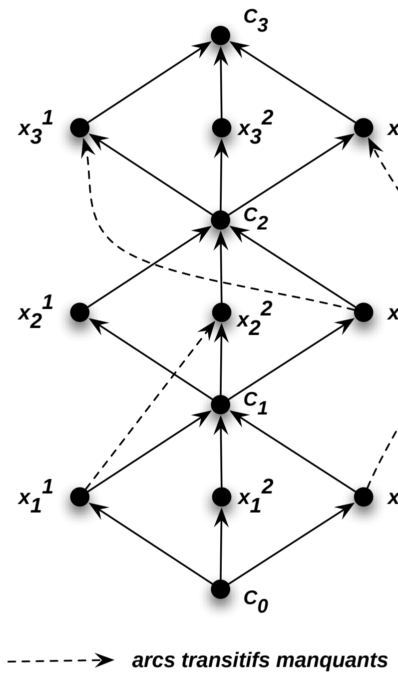

FIG. 1: Graphe associé avec l'instance de 3-SAT $(x_1^1 \vee x_1^2 \vee x_1^3) \wedge (x_2^1 \vee x_2^2 \vee x_2^3) \wedge (x_3^1 \vee x_3^2 \vee x_3^3)$ où $x_1^1 = \overline{x}_2^2$, $x_1^3 = \overline{x}_3^3$ et $x_2^3 = \overline{x}_3^1$. Les arcs transitifs ne sont pas représentés, et les arcs pointillés ne font pas partie du graphe.