Leçon 5.
Le problème SAT et ses variantes

Denis TRYSTRAM
Master Informatique Grenoble – 2011

Motivation. Le problème de satisfabilité *SAT* est le premier problème qui a été montré NP-complet (théorème de Cook en 1971) (voir la leçon **??** pour les détails de cette démonstration). L'étude des variantes de ce problème permet d'une part d'introduire des techniques de preuves et d'autre part de pointer sur la frontière entre problèmes faciles et difficiles.

1 Préliminaires : Présentation du problème SAT

Le problème SAT est un problème de logique propositionnelle, encore appelée logique d'ordre 0 ou logique des prédicats. Rappelons qu'un prédicat est défini sur un ensemble de variables logiques, à l'aide des 3 opérations élémentaires que sont la négation NON ($\neg x$ que nous noterons aussi $\bar{x}$), la conjonction ET ($x \wedge y$) et la disjonction OU ($x \vee y$). Un littéral est un prédicat formé d'une seule variable (x) ou de sa négation $\bar{x}$.

Une clause est un prédicat particulier, formée uniquement de la disjonction de littéraux, par exemple $C = x \vee \bar{y} \vee z$. Une formule est sous forme normale conjonctive si elle s'écrit comme la conjonction de clauses. Le problème SAT consiste à décider si une formule en forme normale conjonctive est satisfiable, c'est-à-dire si il existe une assignation τ de valeur de vérité ($\{true, false\}$) aux variables telle que toutes les clauses sont *true*.

Définition 1 *SAT (Satisfiability)*
Instance : m clauses C_i formées à l'aide de n variables logiques (littéraux)
Question : la formule $\Phi = C_1 \wedge C_2 \wedge ... \wedge C_m$ est-elle satisfiable ?[1]

Par exemple la formule $(x \vee \bar{y} \vee z) \wedge (\bar{x} \vee \bar{y}) \wedge (x \vee \bar{z})$ est satisfaite avec l'assignation $\tau(x) = true$ et $\tau(y) = false$. Il est facile de se convaincre que le problème SAT est dans $\mathcal{NP}$. Un certificat de positivité consiste à donner une assignation des variables, codable sur un vecteur de n bits (n est le nombre de littéraux). La vérification que toutes les clauses sont satisfaites est alors clairement polynomiale (plus précisément, dans $\mathcal{O}(nm)$).

Cook a montré en 1971 que ce problème est $\mathcal{NP}$-complet en codant l'exécution de tout algorithme sur une machine de Turing non-déterministe par une expression en forme normale conjonctive.

[1] ou encore : existe-t-il une fonction d'interprétation qui rende la formule vraie ?

2 Une première série de variante : kSAT

On note kSAT la variante du problème SAT où toutes les clauses ont exactement k littéraux ($k \geq 2$). La variante obtenue pour $k = 3$ est particulièrement utile, on peut établir le résultat simple suivant.

Theorem 1 *3SAT est NP-complet.*

Preuve. L'idée est simplement de réécrire chaque clause comme une clause de cardinalité 3.

- Pour les clauses de deux littéraux, on rajoute une variable et on introduit deux nouvelles clauses qui ne changent pas l'évaluation de la formule.

  Par exemple pour le cas d'une clause de cardinalité 2, $(x \vee y)$, on rajoute la variable z, et on réécrit la clause comme $(x \vee y \vee z) \wedge (x \vee y \vee \bar{z})$.

  Pour les clauses formées d'une seule variable, on rajoutera de même deux variables...

- Pour les clauses de cardinalité p strictement supérieure à 3, on procède avec la même technique.

  Par exemple pour une clause formée de 5 littéraux $(x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5)$, on introduit 2 nouvelles variables y_1 et y_2 (dans le cas général on en rajoute $p-3$ pour des clauses de cardinalité p) et on rajoute autant de clauses ($p-3$ dans le cas général) :

  $(x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5) = (x_1 \vee x_2 \vee y_1) \wedge (\bar{y_1} \vee x_3 \vee y_2) \wedge (\bar{y_2} \vee x_4 \vee x_5)$

  La réduction est alors simple à vérifier.

CQFD

D'une manière générale, tous les problèmes kSAT sont NP-complets pour $k \geq 3$ (en tant que généralisation de 3SAT...).

3 Variantes autour de 2SAT

D'une certaine manière, SAT est un problème frontière, comme nous allons le voir ci-dessous avec les deux variantes : 2SAT et max2SAT.

Définition 2 *2SAT*
Instance : m clauses C_i formées de au plus deux littéraux à l'aide de n variables
Question : la formule $\Phi = C_1 \wedge C_2 \wedge ... \wedge C_m$ est-elle satisfiable ?

Theorem 2 *2SAT est polynomial*

La preuve est constructive : on considère une variable logique quelconque x parmi les n variables. Prenons x= true et remplaçons x dans toutes les clauses où cette variable apparait.

- si x apparait telle quelle, la clause est simplement satisfaite
- si la variable apparait comme $\bar{x}$, on fixe la valeur de l'autre variable de la clause pour qu'elle soit satisfaite. Cela détermine un ensemble de variables X. Deux cas se présentent : soit on réussit à satisfaire ainsi toutes les clauses, on recommence alors avec les variables restantes qui ne sont pas dans X. Soit les clauses ne sont pas toutes satisfaites, on recommence avec la valeur x= *false. Si les clauses sont de nouveau non satisfaites, la réponse à 2SAT est NON.*

CQFD

Il existe une autre construction qui transforme la formule Φ en un graphe $G(\Phi)$ à $2n$ sommets (chaque sommet correspondant aux variables dans les deux états x et $\bar{x}$). On place un arc du sommet $\bar{a}$ au sommet b pour toute clause de type $(a \vee b)$. Ceci traduit l'implication "si l'assignation de a est fausse, alors, celle de b doit être vraie" (ou en terme d'expression logique : $\bar{a} \rightarrow b$). Ainsi, les chemins de ce graphes représentent les implications logiques valides. On peut alors établir le théorème suivant :

Theorem 3 *Φ est non-satisfiable si et seulement si il existe une variable x pour laquelle il existe simultanément un chemin de x à $\bar{x}$ et de $\bar{x}$ à x.*

La preuve est par contradiction : supposons qu'un tel chemin existe et que Φ soit quand même satisfiable. Sans perte de généralité, on suppose qu'elle est satisfaite par une assignation où la variable x est true. Le long du chemin de x à $\bar{x}$ il doit exister un arc α–β tel que α soit true et β soit false. Or, cet arc appartient à $G(\Phi)$ ce qui signifie que la clause $(\bar{\alpha}, \beta)$ est une clause de Φ qui n'est pas satisfaite, ce qui est une contradiction.

Réciproquement, on montre la contraposée : supposons qu'il n'existe pas de variable avec de tels chemins dans $G(\Phi)$.

Notons pour finir que cette formulation permet également de montrer que 2SAT est NL-complet.

Une autre variante de 2SAT permet de mieux caractériser encore la frontière entre Pet NP[2] :

Définition 3 *Max2SAT*

Instance : m clauses C_i formées d'au plus deux littéraux à l'aide de n variables, un entier K

Question : existe-t-il une assignation des variables qui satisfasse au moins K clauses ?

[2]si elle existe...

Theorem 4 *max2SAT est NP-complet.*

Tout d'abord, il est simple de vérifier que Max2SAT est dans NP.

La preuve s'obtient par une réduction à partir de 3SAT, elle illustre la technique de construction à l'aide d'un *gadget* (c'est-à-dire, d'un sous-ensemble particulier de l'instance pour lequel une propriété reste vraie). Ici, le gadget est le problème 2SAT suivant à 10 clauses pour lequel au maximum 7 clauses sont satisfaites lorsqu'au moins un littéral est true (la vérification est facile en explicitant tous les cas possibles) :

Soit une clause $x_1 \vee x_2 \vee x_3$, on introduit un nouveau littéral p et les 10 clauses suivantes :

- (x_1), (x_2), (x_3), (p)
- $(\bar{x_1} \vee \bar{x_2})$, $(\bar{x_2} \vee \bar{x_3})$, $(\bar{x_1} \vee \bar{x_3})$
- $(x_1 \vee \bar{p})$, $(x_2 \vee \bar{p})$, $(x_3 \vee \bar{p})$

Il n'y a aucun moyen de satisfaire toutes ces clauses simultanément. par exemple, si l'on suppose que les 4 premières sont satisfaites, on perd les trois suivantes, etc.. Comme ces 10 clauses sont symétriques par rapport à x_1, x_2 et x_3, nous devons vérifier les cas suivants :

- x_1, x_2 et x_3 sont *true, alors on perd une clause dans la seconde ligne et une dans la troisième. Il reste deux cas selon les valeurs de p (si p est true, on perd les trois dernières, mais on gagne une clause dans la première ligne, si p est false, c'est la première ligne que l'on perd. On satisfait au plus* 7 *clauses.*
- *SI deux des trois variables sont true, ... on satisfait également* 7 *clauses.*
- *Si une seule des trois est true, ... on satisfait* 7 *clauses.*
- *si les trois sont false, on ne peut en satisfaire plus de* 6.

Ces 10 clauses ont la propriété intéressante que toute assignation qui satisfait une clause du type $x_1 \vee x_2 \vee x_3$ peut être étendue pour satisfaire au plus 7 clauses. Ceci suggère une réduction immédiate de Max2SAT à partir de 3SAT, ce qui montre ainsi que ce problème est NP-complet. En effet, étant donnée une instance Φ de 3SAT, on construit une instance $R(\Phi)$ de Max2SAT de la façon suivante : pour chaque clause $C_i = (\alpha \vee \beta \vee \gamma)$ de Φ on rajoute les 10 clauses précédentes à $R(\Phi)$ où α, β et γ remplacent x_1, x_2 et x_3. p est remplacée par une variable supplémentaire p_i associée à la clause C_i. Ainsi, $R(\Phi)$ a $10m$ clauses si Φ en possède m et l'objectif est $K = 7m$. La réduction proposée est bien polynomiale.

On vérifie facilement que l'objectif est atteint pour $G(\Phi)$ ssi Φ est satisfiable.

CQFD

4 Aller plus loin : SAT et la complexité en espace

Classe NL (problèmes qui s'exécutent en espace poly-logarithmique en la taille de l'instance sur une TM non-déterministe). L'opérateur de réduction s'entend ici au sens ...

De même que pour NP, on montre qu'il existe un problème NL-complet (c'est le Graph accessibility Problem – ou GAP).

Définition 4 *GAP Instance : un graphe G=(V,E) orienté et deux sommets x et y appartenant à V. question : existe-t-il un chemin de x à y ?*

On peut montrer que 2SAT appartient à NL car peut être réduit (au sens de la réduction poly-log) à partir du problème GAP. La réduction utilise la transformation d'une formule en graphe vue dans ce chapitre.

5 Ce qu'il faut retenir

- SAT est NP-complet(Cook 1971)
- kSAT est NP-completpour $k \geq 3$(en particulier 3SAT)
- 2SAT est polynomial
- 2SAT est NL-complet
- max2SAT est NP-complet