Algo 2 : TD 2 : Np-complétude

Yoann Lemesle

Mathieu Chambard

11 février 2020

1 3-SAT

On souhaite montrer que 3-SAT est NP-complet.
Le problème 3-SAT est défini comme suit.
Instance : Une formule Φ du calcul propositionnel en forme normale conjonctive dont chaque clause possède au plus 3 littéraux.
Question : La formule Φ est-elle satisfiable,ie peut-on trouver une valuation ν des variables de Φ qui la rende vraie ?

1.1 Montrer que 3-SAT est dans NP

Correction :

Pour montrer cela, on va prendre une valuation ν et l'on va montrer que l'on peut vérifier qu'elle répond ou non au problème 3-SAT (cad satisfait-elle Φ ou non) en un temps polynomial.
La formule Φ du calcul propositionnel est sous la forme normale conjonctive donc on a :
$\Phi = \bigwedge_{i=1}^{n} (a_{i,1} \vee a_{i,2} \vee a_{i,3})$
Or étant donnée une valuation ν, chaque clause de Φ doit être vérifié. Au pire des cas, on doit vérifier les 3 variables propositionnelles. Donc on a, au pire des cas, 3*n comparaisons.
La vérification de cette valuation ν pour le problème 3-SAT a une compléxitée dans le pire des cas en O(n).
Le problème 3-SAT est donc bien NP.

On souhaite maintenant trouver une transformation polynomiale de SAT vers 3-SAT.

1.2 Soit C une clause contenant 2 littéraux. En introduisant une nouvelle variable, transformer C en une conjonction $C_1 \wedge C_2$ de clauses contenant 3 littéraux et telle que C et $C_1 \wedge C_2$ sont équisatisfiables.

Correction :

On a C une clause de 2 littéraux donc $C = p \vee q$.
On pose $C' = (p \vee q \vee i) \wedge (p \vee q \vee \neg i)$.
On va montrer que C et C' sont équisatisfiables et que les mêmes valuations les rendent vrai.
($\Rightarrow$)Pour cela, on prend une valuation qui satisfait C : on a donc p ou q a vrai et donc $(p \vee q \vee i)$ est à vrai et $(p \vee q \vee \neg i)$ est à vrai et finalement C' est à vrai. On a donc trouvé une valuation qui satisfait C'
($\Leftarrow$) On prend un valuation ν qui rend C' vrai.
Si i = vrai, on sait que $(p \vee q \vee \neg i)$ est vrai donc $(p \vee q)$ est vrai car $\neg i$ est faux.
Sinon, on sait que $(p \vee q \vee i)$ est vrai donc $(p \vee q)$ est vrai car i est faux Donc cette valuation satisfait C.
On a donc prouvé que C et C' sont équisatisfiables.

1.3 Soit $k \neq 4$ et C une clause contenant k littéraux. En introduisant k-3 nouvelles variables, transformer C en une conjonction $C_1 \wedge ... \wedge C_{k-2}$ de clauses contenant 3 littéraux et telle que C et $C_1 \wedge ... \wedge C_{k-2}$ sont équisatisfiables.

Correction :

C est cette fois-ci de la forme $C = l_1 \vee ... \vee l_k$. J'introduis les k-3 variables suivantes : $p_1, ..., p_{k-3}$ et je pose :
$C' = (l_1 \vee l_2 \vee p_1) \bigwedge_{i=3}^{k-2} (l_i \vee \neg p_{i-2} \vee p_{i-1}) \wedge (\neg p_{k-3} \vee l_{k-1} \vee l_k)$
On montre que C et C' sont équisatisfiables :
($\Rightarrow$) On prend une valuation ν qui satisfait C.
$\exists j \in [|1, k|]$ tel que l_j est vrai. On met p_1,..,p_{j-2} à Vrai et p_{j-1},...,p_{k-3} à Faux. Je dis que cette nouvelle valuation

ν' satisfait C'. Je le montre :
Si j = 1,2 alors $(l_1 \vee l_2 \vee p_1)$ est vrai et on met tout les p_i à Faux et le reste est vrai car chaque clause à un $\neg p_i$.
Si j = k-1,k alors $(\neg p_{k-3} \vee l_{k-1} \vee l_k)$ est vrai et on met tout les p_i à Vrai et le reste est vrai car chaque clause à un p_i.
Sinon on a $(l_j \vee \neg p_{i-2} \vee p_{i-1})$ qui est vrai et on met p_1,..,p_{j-2} à Vrai donc $(l_1 \vee l_2 \vee p_1)$ est vrai et toutes les clauses $(l_i \vee \neg p_{i-2} \vee p_{i-1})$ pour i allant de 3 à j-1 sont vrai grâce à p_{i-1}. De plus, on met p_{j-1},..,p_{k-3} à Faux donc $(\neg p_{k-3} \vee l_{k-1} \vee l_k)$ est vrai et toutes les clauses $(l_i \vee \neg p_{i-2} \vee p_{i-1})$ pour i allant de j+1 à k-2 sont vrai grâce à $\neg p_{i-2}$.
Donc cette nouvelle valuation satisfait C'.

($\Leftarrow$) Soit ν une valuation qui satisfait C'. On va montrer qu'il existe un l_i qui est vrai.
Pour cela, on suppose qu'il n'y a pas de l_i à vrai. C' est donc équivalent à $p_1 \bigwedge\limits_{i=3}^{k-2} (\neg p_{i-2} \vee p_{i-1}) \wedge \neg p_{k-3}$. Donc p_1 est vrai et C' est donc équivalent à $p_2 \bigwedge\limits_{i=4}^{k-2} (\neg p_{i-2} \vee p_{i-1}) \wedge \neg p_{k-3}$ Donc p_2 est vrai et de proche en proche on a p_{k-3} qui est vrai. Or la dernière clause est $\neg p_{k-3}$ et elle est vrai. C'est absurde!!
On a donc qu'il existe i tel que l_i est vrai. Donc la valuation satisfait C.
On a donc que C et C' sont équisatisfiables.

1.4 Conclure quant à la complexité du problème 3-SAT

Correction :

La question 1.3 nous donne une réduction du problème SAT vers 3-SAT.
On prend une formule Φ du calcul propositionnel en forme normale conjonctive. On regarde chaque clauses. Pour chaque clauses C, on pose k le nombre de littéraux de celle-ci. Avec la question 1.3, on peut transformer le problème de satisfiabilité de cette clause C en un problème de satisfiabilité d'un problème 3-SAT. Donc pour chaque clause C, on la transforme en k-2 clauses de 3 littéraux.
On a donc une réduction en un temps polynomial d'un problème SAT vers un problème 3-SAT. Or le problème SAT est un problème NP-difficile donc 3-SAT est un problème NP-difficile. Or 3-SAT est NP donc 3-SAT est un problème NP-complet.

2 MAX2SAT

Dans cet exercice, on souhaite prouver que le problème MAX2SAT est NP-Complet. Ce problème est défini ainsi :
Instance : Une formule Φ du calcul propositionnel en forme normale conjonctive dont chaque clause possède au plus 2 littéraux, et un entier k.
Question : La formule Φ est-elle k-satisfiable,ie peut-on trouver une valuation ν des variables de Φ qui rende k clauses de Φ vraies ?

2.1 Etudier la satisfaction des clauses de Φ' en fonction de celle des clauses de Φ

Correction :

Chacune des clauses ($\mathrm{a_i} \vee \mathrm{b_i} \vee \mathrm{c_i}$) de φ est satisfaite si un littéral ou plus parmis $\{\mathrm{a_i}, \mathrm{b_i}, \mathrm{c_i}\}$ est vrai.

$\rightarrow$ 1 littéral vrai : L_1

- Les 3 clauses ($\neg$X $\vee$ $\neg$Y) sont vraies car il ne peut pas exister de couple (X,Y) de littéraux vrais.
- Une des clauses (a_i) , (b_i) , (c_i) est vraie.
- d_i vrai : (d_i) et $(L_1 \vee \neg d_i)$ vraies $\rightarrow$ 5 clauses satisfaites.

d_i faux : les 3 clauses (X $\vee$ $\neg d_i$) sont vraies $\rightarrow$ 7 clauses satisfaites.

$\rightarrow$ 2 littéraux vrais : L_1 et L_2

• Seule une des 3 clauses (¬X ∨ ¬Y) est fausse, quand (X,Y) = (L_1, L_2), 2 clauses sont dont satisfaites.
• Deux des clauses (a_i) , (b_i) , (c_i) sont vraies.
• d_i vrai : (d_i) , $(L_1 \lor \neg d_i)$ et $(L_2 \lor \neg d_i)$ vraies → 7 clauses satisfaites.
d_i faux : les 3 clauses (X ∨ $\neg d_i$) sont vraies → 7 clauses satisfaites.

→ 3 littéraux vrais

• Toutes les clauses (¬X ∨ ¬Y) sont fausses.
• Les 3 clauses (a_i) , (b_i) , (c_i) sont vraies.
• d_i vrai : (d_i) , $(a_i \lor \neg d_i)$, $(b_i \lor \neg d_i)$ et $(c_i \lor \neg d_i)$ vraies → 7 clauses satisfaites.
d_i faux : les 3 clauses (X ∨ $\neg d_i$) sont vraies → 6 clauses satisfaites.

Dans tous les cas, pour une clause satisfaite de φ, 7 clauses de φ' sont satisfiables car d_i n'est pas fixé par la valuation qui satisfait $(a_i \lor b_i \lor c_i)$. Si φ contient P clauses vraies, alors φ' contient 7 * P clauses satisfiable.

2.2 Proposer une valeur de k telle que Φ est satisfiable si et seulement su k clauses de Φ' sont simultanément satisfiables.

Correction :

Pour une formule φ à P clauses, on propose k = 7 * P.

2.3 Montrons que MAX2SAT est NP-complet.

Correction :

Un verificateur pour MAX2SAT prend en entrée une instance du problème (une formule normale conjonctive dont les clauses contiennent au plus 2 littéraux, un entier k) et une valuation. Vérifier que cette valuation rend vraies au moins k clauses se fait en temps polynomial sur le nombre de clause, avec une complexité temporelle constante pour chaque clause (du fait du nombre borné de littéraux).

MAX2SAT appartient donc bien à la classe des problèmes NP.

Montrons que MAX2SAT ∈ NP-Difficile :

Soit une formule φ 3-CNF, transformons la en une formule φ' via le procédé définit par l'énoncé.
On applique à φ' le problème MAX2SAT avec k = 7*P, P le nombre de clauses de φ'.
D'après 2.2, si le résultat est positif pour 7*P, alors la formule φ est satisfiable, ce qui répond au problème 3-SAT.
Ainsi, le problème 3-SAT se réduit au problème MAX2SAT. Or 3-SAT est NP-Difficile, il en est donc de même pour MAX2SAT.

MAX2SAT est NP et NP-Difficile, il est donc NP-Complet.

3 Couverture par sommets :

On définit le problème de la couverture par sommets d'un graphe
Instance : Un graphe non orienté G = (V,E), un entier naturel k
Question : Le graphe G admet-il une couverture par k sommets, i.e. existe-t-il $V' \subseteq V$, avec $|V'| \leq k$ tel que $\forall \{u, v\} \in E, u \in V' \lor v \in V'$

3.1 Montrons que VC est dans NP.

Correction :

Soit V' un ensemble de sommets. On doit vérifier si cet ensemble recouvre notre graphe. Pour cela, On prend chaque arrête et on vérifie que un de ces sommets est dans V'.
On a donc au maximum 2 * |V'| * |A| comparaisons(|V'| est pour vérifier que le sommet est dans V'). La vérification est donc bien en temps polynomial.
Le problème VC est donc bien dans NP.

3.2 Construire G et exhiber une couverture minimale de G.

Correction :