

2-SAT est NL, CLIQUE est NP-Complet

Julien Devevey

2018-2019

Ref : Cormen - Introduction to Algorithms p.1087

Papadimitriou - Computational Complexity p.184

Théorème 1. 2-SAT $\in$ NL.

Démonstration.

Soit ϕ une instance de 2-SAT, c'est à dire que ϕ est une formule sous forme CNF, donc un ensemble de clauses de deux littéraux chacune. On construit le graphe orienté $G(\phi)$ suivant : soient $x_1, \dots, x_n$ les variables de ϕ . Alors $G(\phi)$ a $2n$ sommets, chaque sommet représentant soit une variable, soit sa négation. De plus, l'ensemble des arêtes de $G(\phi)$ est défini par : $E(\phi) = \{(\alpha, \beta) \mid \text{il existe une clause } (\neg\alpha \vee \beta) \text{ ou } (\neg\beta \vee \alpha) \text{ dans } \phi\}$.

Lemme 2. ϕ est insatisfiable si et seulement si il existe une variable x telle qu'il existe un chemin de x à $\neg x$ et de $\neg x$ à x dans $G(\phi)$.

Démonstration.

Supposons que de tels chemins existent et que ϕ soit satisfiable par une affectation T . Quitte à remplacer tous les $\neg x$ par x et vice-versa dans ϕ , on peut supposer que $T(x) = \text{Vrai}$. Or $T(\neg x) = \text{Faux}$, et il y a un chemin entre les deux dans $G(\phi)$, il existe donc une arête (α, β) dans le graphe telle que $T(\alpha) = \text{Vrai}$ et $T(\beta) = \text{Faux}$. Alors $(\neg\alpha \vee \beta)$ s'évalue à Faux. Or c'est une clause de ϕ par construction de $G(\phi)$: c'est une contradiction, et ϕ est alors insatisfiable.

Réciproquement, supposons qu'aucune variable n'admette un tel chemin dans $G(\phi)$. On va construire une affectation qui satisfait ϕ , c'est à dire une affectation telle qu'aucune arête de $G(\phi)$ n'aille du Vrai vers le Faux (ou vice-versa). On répète le processus suivant jusqu'à avoir une affectation de toutes les variables : on choisit α une variable qui n'a pas encore été affectée. On affecte α et tous les littéraux qu'on peut atteindre depuis α à Vrai, et leur négation à Faux (en remarquant que $(\alpha, \beta) \in E(\phi) \Leftrightarrow (\neg\beta, \neg\alpha) \in E(\phi)$, on voit que ceci correspond à mettre à faux tous les sommets qui peuvent atteindre $\neg\alpha$). Cette étape est bien définie, car s'il y a un chemin de α vers β et $\neg\beta$, on aurait des chemins de β et $\neg\beta$ à $\neg\alpha$, donc de α à $\neg\alpha$, ce qui contredit l'hypothèse. De plus, s'il y a un chemin de α vers un noeud qui est déjà assigné à Faux, alors il y a un chemin de la négation de ce noeud vers $\neg\alpha$, donc $\neg\alpha$ aurait déjà dû être affecté à Vrai et α à Faux : ce cas n'arrive pas.

Au final, on obtient une affectation valide de toutes les variables, et de plus on sait par construction qu'aucune arête du graphe ne part du Vrai pour aller vers le Faux ou vice-versa : l'affectation satisfait donc ϕ . $\square$

Retour à la démonstration du théorème.

Etant donné que NL est clos par passage au complémentaire, on choisit de montrer qu'on peut reconnaître les expressions insatisfiables en espace logarithmique non-déterministe. Pour ça, il suffit de choisir une variable x et de tester s'il y a un chemin de x à $\neg x$ et inversement. Or le problème *PATH* est dans NL, ce qui conclut la preuve. $\square$

Remarque 3. On peut aussi montrer que 2-SAT est NL-Complet en effectuant une réduction depuis *PATH*, qui est NL-Complet.

Théorème 4. CLIQUE $\in$ NP-Complet.

Démonstration.

Etant donné un graphe $G = (V, E)$ et $k \in \mathbb{N}$, on vérifie s'il y a une clique en utilisant comme certificat le sous-ensemble $V' \subset V$ sensé être la clique, et il suffit de vérifier qu'il y a une arête entre chaque élément de V' et que $|V'| = k$. Tout ceci se fait en temps polynomial. Donc CLIQUE $\in$ NP.

Il reste maintenant à montrer que CLIQUE est NP-difficile. Pour ça, on va montrer que 3-CNF-SAT $\leq_P$ CLIQUE. Soit alors ϕ une formule booléenne sous forme CNF, et soit $C_i, i \in \llbracket 1, k \rrbracket$ ses clauses, contenant exactement trois littéraux, qu'on note l_1^i, l_2^i, l_3^i . On construit à nouveau un graphe $G(\phi)$ tel que ϕ est satisfiable si et seulement si $G(\phi)$ a une clique de taille k .

On pose alors le graphe non orienté $G(\phi) = (V, E) : |V| = 3k$ et chaque sommet v_j^i de V correspond à un littéral $l_j^i, (i, j) \in \llbracket 1, k \rrbracket \times \llbracket 1, 3 \rrbracket$. De plus, on crée une arête entre deux sommets v_i^r et v_j^s si et seulement si $r \neq s$ et l_i^r n'est pas la négation de l_j^s . Un tel graphe se construit en temps polynomial.

Supposons que ϕ soit satisfiable et que T est une affectation satisfaisant ϕ . Alors chaque clause contient un littéral évalué à Vrai : choisir un tel littéral (et donc sommet associé) pour chaque clause correspond à une clique dans le graphe associé car on choisit toujours dans des clauses différentes, et chaque littéral n'est pas la négation d'un autre (étant donné que chaque littéral choisi est évalué à vrai dans T et T est valide).

Réciproquement, si on trouve une clique de taille k dans le graphe, alors comme aucun des sommets dans la clique n'est relié à un sommet issu de la même clause, et qu'aucun sommet de la clique n'est relié à sa négation, on peut choisir d'évaluer à Vrai tous les littéraux des sommets de la clique, et à Faux les autres : l'évaluation est bien valide pour la seconde raison, et pour cette évaluation, toutes les clauses à Vrai pour la première raison (au moins un littéral dans chaque clause est évalué à Vrai, car il y a k clauses et la clique est de taille k).

Au final, on a donc bien prouvé que CLIQUE est NP-difficile donc NP-complet. $\square$