Corrigé TD 3
Complexité

Luc Lapointe
luc.lapointe@ens-paris-saclay.fr
home.lmf.cnrs.fr/LucLapointe/

Vous pouvez reprendre les exercices du TD précédent si vous le souhaitez.

La classe coNP Soit une classe $\mathcal{C}$ de problèmes de décision. La classe co$\mathcal{C}$ correspond à l'ensemble des langages L tels que $\overline{L} \in \mathcal{C}$.

Exercice 1 – Clôture

Supposons que le langage L soit complet pour la classe $\mathcal{C}$. Exhiber un langage complet pour la classe co$\mathcal{C}$.

Idée de solution

Considérons L^c le complémentaire de L. C'est, par définition, un langage qui est dans co$\mathcal{C}$. Montrons qu'il est co$\mathcal{C}$-dur.

Soit K un langage dans co$\mathcal{C}$. Alors K^c est un langage dans $\mathcal{C}$, donc il existe une réduction de K^c vers L. En conséquence, la même fonction est une réduction de K vers L^c.

Exercice 2 – Tautologie

Prouver que le problème de décision Tautologie suivant est coNP-complet :

Entrée: Une formule propositionnelle φ sous forme normale disjonctive.
Question: Est-ce que toute valuation satisfait φ ?

Idée de solution

Le problème complémentaire est le suivant :
Entrée: Une formule propositionnelle φ sous forme normale disjonctive.
Question: Est-ce qu'il existe une valuation qui ne satisfait pas φ ?

C'est un problème interréductible avec SAT, en prenant la négation de la formule. D'où la NP-complétude : un sens de la réduction donne que le problème est dans NP, l'autre qu'il est NP-dur.

Exercice 3 – Changement de format

1. Le problème SAT reste-t-il NP-complet si la formule est en forme normale disjonctive (au lieu de conjonctive) ?
2. Le problème Tautologie reste-t-il coNP-complet si la formule est en forme normale conjonctive (au lieu de disjonctive) ?

Idée de solution

1. cf TD précédent.
2. Interréductible avec SAT en FND, donc non.

Exercice 4 – Combien de patates

Un problème coNP est-il, a priori, dans NP ?

Idée de solution

Non. L'égalité de NP et coNP est un problème ouvert. Il est conjecturé que ces classes sont différentes.

Problèmes NP-complets sur les graphes

Note : les trois problèmes qui suivent sont des problèmes d'optimisation qui ont été transformés en problème de décision, via l'ajout d'un seuil m.

Exercice 5 – Ensemble indépendant

Un *ensemble indépendant* dans un graphe non orienté $G = (S, A)$ est un ensemble $C \subseteq S$ de sommets dont aucun n'est relié à aucun autre par une arête de G, c'est à dire que $u, v \in C$ implique $uv \notin A$. Démontrer que le langage INDEPENDANT SET défini comme suit est NP-complet :

Entrée: Un graphe non orienté $G = (S, A)$, un entier $m \in \mathbb{N}$
Question: G a-t-il un ensemble indépendant de cardinal au moins m ?

Idée de solution

Dans NP :
1. Deviner un ensemble de m sommets du graphe.
2. Vérifier en temps polynomial que c'est un ensemble indépendant.

Pour la NP-difficulté, réduction depuis 3-SAT. L'idée va être de représenter les valuations possibles des variables avec des sommets, et les arêtes vont représenter des incompatibilités.

Soit φ une entrée de 3-SAT, avec k clauses, et des variables étiquetés $x_1, ..., x_n$. On considère comme entier en entrée de INDEPENDANT SET l'entier k. Pour le graphe, on considère un ensemble de sommets comportant autant de sommets que de littéraux présents dans la formule, avec répétitions. Plus précisément, si la clause i est $l_a \vee l_b \vee l_c$, alors le graphe aura des somemts l_a^i, l_b^i, l_c^i. Pour les arêtes :
- Une arête tous les littéraux au sein d'une même clause. Les 3-clause forment donc des triangles.
- Pour tout j, une arête entre toutes les copies de x_j et toutes les copies de $\neg x_j$.

Cette réduction peut se faire en espace logarithmique : il suffit de construire le graphe au fur et à mesure que la formule est lue. Deux compteurs suffisent pour construire les arêtes entre les x_j et $\neg x_j$, le reste peut se faire sans mémoire.

Cette réduction conserve les solutions : si φ est satisfiable, alors chaque valuation qui la satisfait donne au moins un ensemble indépendant de taille k, en considérant pour chaque clause un sommet dans le graphe qui est le pendant d'un littéral satisfait dans la formule. Réciproquement, si le graphe a un ensemble indépendant E de taille k, alors du fait des triangles il y a au plus un sommet de E dans chaque groupe de sommets qui correspond à une clause - donc exactement k, car il y a k tels groupes. Un tel ensemble ne comprend par ailleurs jamais une variable et sa négation. On peut donc en déduire une valuation, qui s'avère satisfaire φ.

Exercice 6 – Recouvrement

Un *recouvrement* C d'un graphe non orienté $G = (S, A)$ est un ensemble $C \subseteq S$ de sommets tel que toute arête de A est incidente à C, c'est à dire à au moins un élément de C. Démontrer que le langage Vertex Cover défini comme suit est NP-complet :

Entrée: Un graphe non orienté $G = (S, A)$, un entier $m \in \mathbb{N}$
Question: G a-t-il un recouvrement de cardinal au plus m ?

Idée de solution

Dans NP :

1. Deviner un ensemble de m sommets du graphe.
2. Vérifier en temps polynomial que c'est un recouvrement.

Pour la NP-difficulté, réduction depuis Independant Set. On commence par prouver que dans un graphe $G = (S, A)$, E est un ensemble indépendant si et seulement si son complémentaire est un recouvrement.

E est un ensemble indépendant de G ssi pour $x, y \in E$, il n'y a pas d'arête entre x et y ssi pour toute arête xy, soit x soit y est dans E^c ssi E^c est un recouvrement.

La réduction est ensuite simple : pour G un graphe comportant n sommet, il suffit de conserver G et remplacer m par $n - m$.

Exercice 7 – Clique

Une *clique* C d'un graphe non orienté $G = (S, A)$ est un ensemble $C \subseteq S$ de sommets induisant un sous-graphe complet de G, c'est à dire tel que pour $u \neq v \in C$ on a $uv \in A$. Démontrer que le langage Vertex Cover défini comme suit est NP-complet :

Entrée: Un graphe non orienté $G = (S, A)$, un entier $m \in \mathbb{N}$
Question: G a-t-il un recouvrement de cardinal au plus m ?

Idée de solution

Dans NP :

1. Deviner un ensemble de m sommets du graphe.
2. Vérifier en temps polynomial que c'est une clique.

Pour la NP-difficulté, réduction depuis Vertex Cover. Pour G un graphe, on note G^c son *graphe complémentaire* défini comme comportant les mêmes sommets, et où il y a une arête entre x et y dans G si et seulement si il n'y en a pas entre x et y dans G^c.

On commence par prouver que dans un graphe $G = (S, A)$, E est une clique si et seulement si son complémentaire est un recouvrement dans le graphe complémentaire de G.

E est une clique de G ssi pour $x, y \in E$, il y a une arête entre x et y dans G ssi pour $x, y \in E^c$, il n'y a pas d'arête entre x et y dans G^c ssi pour toute arête xy dans G^c, soit x soit y est dans E^c ssi E^c est un recouvrement dans G^c.

La réduction consiste ensuite à recopier m, et à inverser les arêtes dans G. Ceci peut se faire en espace logarithmique quel que soit l'encodage de G.

Exercice 8 – 3-Coloration

Prouver que le problème 3-Coloration est NP-complet. Pour prouver la NP-dureté de ce problème, on pourra effectuer une réduction depuis le problème 3-SAT, également NP-complet.

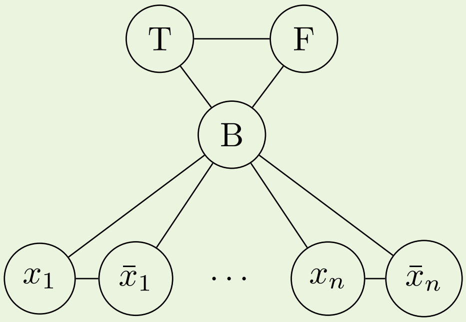
Idée de solution

Dans NP :

1. Deviner une fonction qui à chaque sommet du graphe associe un entier entre 1 et 3.
2. Vérifier en temps polynomial que c'est une 3-coloration.

Pour la NP-difficulté, réduction depuis 3-SAT. Soit φ une formule en 3-FNC. On commence par faire une base de graphe avec :

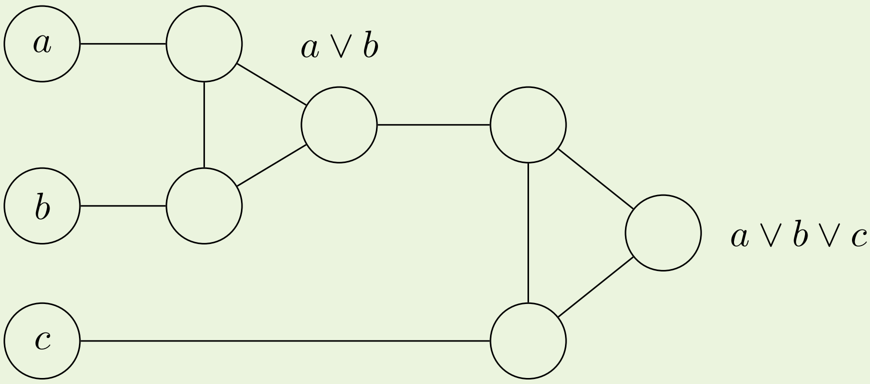
- Pour chaque variable, un sommet avec et un sommet sans négation,
- Trois sommets B, F, V pour Base, Faux, Vrai, connectés en triangle,
- Le sommet B est connecté à chaque sommet de variable, et chaque variable est connectée à sa négation.



Dans l'idée, la couleur des sommets F et V correspondra à une valeur de vérité d'une variable.

Ensuite, pour chaque clause, on rajoute un gadget comme ci-dessous :

- Un triangle dont un sommet est relié à la première variable a de la clause, et le deuxième à une autre variable b.
- Un autre triangle dont un sommet est relié au troisième sommet du premier triangle, et un autre à la troisième variable c de la clause.



Dans l'idée, le troisième du premier triangle sera nécessairement affecté à la couleur de faux si a et b sont fausses. Par contre, si une des deux est vraie, *il existe* une coloration qui colore comme vrai le troisième sommet. Ainsi, le troisième sommet du premier triangle décrit la valeur de vérité de $a \vee b$. De même, le troisième sommet du deuxième triangle décrit la valeur de vérité de $a \vee b$.

Ensuite, il suffit de relier le troisième sommet du deuxième triangle à B et F.