Complexité et Calculabilité 2019/20,
Devoir maison (projet)

Le problème de graphes *Distance commune* décrit ci-dessous est un problème NP-complet. Vous allez dans ce devoir (1) réduire le problème *Distance commune* vers le problème *SAT* (satisfaisabilité de formules booléennes en CNF) et (2) utiliser un SAT-solveur pour résoudre *Distance commune*.

1 Définition du problème *Distance commune*

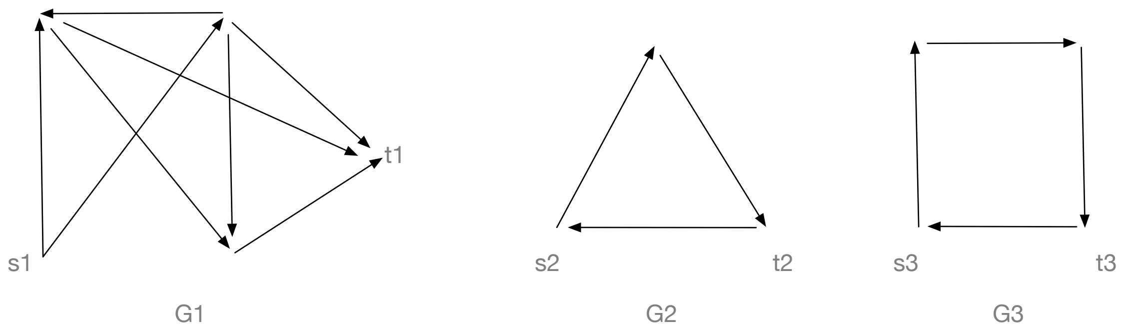
Un graphe pointé est un tuple $G = (V, E, s, t)$ tel que (V, E) est un graphe orienté, $s \in V$ est le sommet *source* et $t \in V$ est le sommet *destination*. Un chemin est une séquence (non-vide) de sommets $v_1, \ldots, v_n$ t.q. $(v_i, v_{i+1}) \in E$ pour tout $0 \leq i < n$ (la longueur du chemin est $n-1$). Il est dit *valide* si $v_1 = s$ et $v_n = t$. Il est dit *simple* si il ne contient qu'au plus une seule fois chaque sommet.

Le problème *Distance commune* est :

Entrée : $G_1 = (V_1, E_1, s_1, t_1), \ldots, G_N = (V_N, E_N, s_N, t_N)$ graphes pointés.

Question : Est-ce qu'il existe un entier n tel que chacun des graphes $G_1, \ldots, G_N$ possède un chemin simple et valide de longueur n ?

Exemples : si l'entrée du problème sont les graphes G_1, G_2, G_3 ci-dessous, alors la réponse est « non ». Si l'entrée est G_1, G_2, la réponse est « oui » (avec $n = 2$). Si l'entrée est G_1, G_3, la réponse est « oui » (avec $n = 3$). Si l'entrée est G_2, G_3, la réponse est « non ».



1. Justifiez que ce problème appartient à la classe NP, en fournissant (informellement) un vérificateur dont vous justifierez la complexité polynomiale et en justifiant que les témoins des instances positives sont de taille polynomiale.

2 Cas particulier acyclique

Un graphe orienté qui ne contient aucun circuit (cycle) est appelé *graphe acyclique*.

1. On suppose ici que les graphes $G_1, \ldots, G_N$ en entrée sont acycliques. Proposez un algorithme polynomial pour le problème Distance commune. Précisez le temps de calcul de votre algorithme.

   Indication : on peut calculer par induction sur k l'ensemble des sommets reliés à une source par un chemin de longueur k.
2. Pourquoi l'algorithme précédent ne fonctionne-t'il pas dans le cas général (avec cycles) ?
3. (bonus points) Trouvez un exemple témoignant que si on modifie le problème pour demander l'existence de chemin valides non nécéssairement simples, alors il existe des cas où le plus petit n est exponentiel en N.

3 Réduction de ***Distance commune*** vers ***SAT***

Le but de cette section est une réduction polynomiale de *Distance commune* vers *SAT*. Vous allez proposer une réduction, et la justifier dans votre rapport (voir détails dans la Section 4).

Remarques : l'efficacité d'un solveur dépend bien-sûr de la taille de la formule, mais aussi du nombre des variables. Pour le problème *Distance commune* vous pouvez soit générer une seule formule à partir de l'entrée, ou plusieurs formules, dont la satisfiabilité est testée une par une.

1. Dans le cas d'une seule formule, les variables de la formules seront $x_{i,j,k,q}$ avec $1 \leq i \leq N$, $0 \leq j \leq k$, $q \in V_i$, pour une certaine valeur de k. Une telle variable sera vraie si et seulement si le sommet q est le j-ème sommet d'un chemin simple de longueur k du graphe pointé G_i, issu du sommet source s_i.
   (a) Quelle est la valeur maximale du paramètre k qu'on doit considérer ?
   (b) Donnez une formule propositionnelle qui est satisfiable si et seulement si chacun des graphes en entrée dispose d'un chemin simple et valide de longueur k. Vous découperez cette formule en sous-formules simples dont vous décrirez intuitivement le rôle.
   (c) Démontrez que si il existe un chemin simple et valide de longueur k dans chacun des graphes, alors votre formule est satisfiable (vous décrirez la valuation satisfaisant votre formule en fonction de ces chemins).
   (d) Démontrez que si il existe une valuation satisfaisant votre formule, alors il existe un chemin simple et valide de longueur k dans chaque graphe (vous décrirez ce k et ces chemins en fonction de la valuation).
   (e) Déduisez-en la réduction de Distance commune vers SAT en découpant votre réduction en sous-formules simples, dont vous expliciterez le rôle.
2. Vous pouvez aussi générer des formules pour chaque valeur de $k = 1, 2, \ldots,$ et les tester une par une.

(a) Décrivez l'algorithme permettant d'obtenir la solution au problème Distance commune suivant cette approche, en précisant quelle formule est testée pour chaque k.

(b) Pouvez-vous avoir moins de variables sur l'ensemble de l'algorithme qu'à la question précédente ? Pourquoi ?

(c) Quel est le gain pour cette approche ?

(d) Justifiez que cette approche est équivalente à la précédente en commentant la forme de la formule obtenue à la question précédente par rapport à celle de celle-ci.

3. (bonus points) Mettez votre formule en forme normale conjonctive.

4 Évaluation

Vous réaliserez *obligatoirement* le travail par binômes. Vous déclarerez ces binômes sur la plateforme moodle [1] avant le 1/10/2019, en vous répartissant *à l'intérieur* de votre groupe de TD.

Le devoir consistera en deux parties :

- Un programme réalisant la dite réduction vers un SAT-solver (z3). Un squelette de code, un parseur de graphes et un exemple d'utilisation de z3 vous seront fournis ultérieurement.
- Un document PDF répondant aux questions présentées dans le présent sujet, qui vous permettront notamment de démontrer la correction de la réduction du problème vers SAT. Ce document contiendra également un rapport de votre programme détaillant son fonctionnement, ainsi que les éventuelles optimisations apportées et difficultés rencontrées.

Vous remettrez au plus tard le 3 novembre 2019 sur la plateforme moodle une archive contenant votre code, ainsi que les documents précédemment décrits. Les évaluations auront lieu durant les TDs après les vacances de Toussaint. Votre code sera testé sur les machines du CREMI et devra donc pouvoir s'y executer. Étant donné qu'un squelette fonctionnel vous a été fourni, tout programme ne compilant pas se verra attribuer une note nulle sur la partie du code.

1. `https://moodle1.u-bordeaux.fr/course/view.php?id=4861`