université de BORDEAUX

Collège Sciences et technologies

Devoir maison
Partie Théorique
Calculabilité et Complexité
3 novembre 2019

ANEAS Alan et CATALDI Alexis
Master - Année 1
Spécialité informatique

Professeur de TD :
SENHAJI Mohammed

Professeur en amphithéâtre :
MUSCHOLL Anca

Sommaire

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 ∈ V est le sommet *source* et t ∈ V est le sommet *destination*. Un chemin est une séquence (non-vide) de sommets v_1, ..., v_n t.q. (v_i, v_{i+1}) ∈ 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* s'il ne contient qu'au plus une seule fois chaque sommet.

Le problème *Distance commune* est :

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

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

1.1 Classe NP

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.

R : En entrée, il nous est donné un chemin par graphe.
Il suffit de vérifier que chaque chemin est de la même taille, et que chaque chemin est valide, en vérifiant que chaque arrête existe et qu'on ne passe pas deux fois par le même sommet.
Il faut aussi que le sommet de départ soit le sommet *source*, et le sommet d'arrivée soit le sommet *destination* de chaque graphe.

O(1) pour vérifier les tailles des chemins et que les sommets sources et destinations soient corrects.
O(n) pour parcourir le graphe en prenant les arcs mentionnés.
O(n*N) pour parcourir chaque graphe de cette façon.
Cela est bien polynomial.

2 Cas particulier acyclique

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

2.1 Algorithme

1. On suppose ici que les graphes G_1, ..., G_N en entréee 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.

R : On fait le parcours du plus petit des graphes en partant du point s et l'on note la longueur des chemins arrivant à t en mettant à *vrai* une case d'un tableau dont l'indice correspond à la longueur du chemin. Ensuite on parcourt les autres graphes en notant s'ils possèdent eux aussi des cases à *vrai*, à la fin de l'algorithme, si tous les tableaux ont une case avec le même indice à *vrai* alors cela veut dire que tous les graphes ont un chemin de même taille avec cette taille étant l'indice de la case.

Complexité : Cela correspond à un parcours en profondeur de tous les graphes, soit la somme des n et m, avec n et m étant respectivement le nombre de sommets et d'arcs dans chaque graphe.

$$O(\sum_{i=1}^{N} n + m) = O(N * (m + n))$$

Soit une complexité polynomiale en l'entrée.

2.2 Cas général (avec cycles)

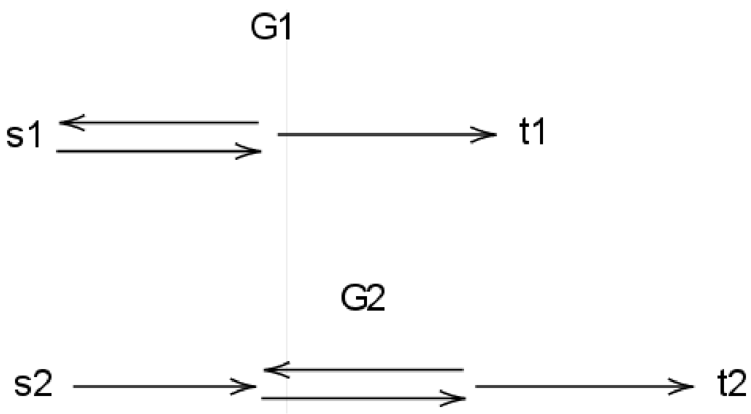
2. Pourquoi l'algorithme précédent ne fonctionne-t-il pas dans le cas général (avec cycles) ?

R : Dans le cas général, les boucles peuvent nous faire tourner à l'infini dans le premier graphe ce qui fausse notre tableau des tailles des chemins et surtout ne nous fait pas finir l'algorithme.

2.3 Bonus : n exponentiel

3. (bonus points) Trouvez un exemple témoignant que si on modifie le problème pour demander l'existence de chemin valides non nécessairement simples, alors il existe des cas où le plus petit n est exponentiel en N.

R : Il suffit d'avoir un cycle, et l'algorithme pourrait boucler infiniment. S'il l'on programme quelque chose pour passer à un autre graphe si l'un fait boucler infiniment, une fois qu'il aurait trouvé un n il devrait revenir au graphe bouclant, pour voir s'il trouve un chemin de cette taille aussi, et s'il ne le trouve pas, il réessayera avec l'autre graphe, cela pourrait prendre un temps exponentiel, voire infini dans le cas où l'on ne trouve que des tailles paires pour un graphe et impaires pour l'autre, par exemple.



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 satisfaisabilité est testée une par une.

3.1 $x_{i,j,k,q}$

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^{eme} sommet d'un chemin simple de longueur k du graphe pointé G_i, issu du sommet source s_i.

3.1.1 Valeur maximale de k

(a) Quelle est la valeur maximale du paramètre k qu'on doit considérer ?

R : Il s'agit du nombre de sommet du plus petit des graphes.

3.1.2 Formule propositionnelle

(b) Donnez une formule propositionnelle qui est satisfaisable 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.

R :
"Point de départ s"

$$\phi_1 = x_{i,0,k,s}$$

"Point d'arrivée t"

$$\phi_2 = x_{i,k,k,t}$$

"Au moins 1 sommet par position"

$$\phi_3 = \bigwedge_{j=0}^{k} \bigvee_{u \in V(G)} x_{i,j,k,u}$$

"Au plus 1 sommet par position"

$$\phi_4 = \bigwedge_{j=0}^{k} \bigwedge_{u \in V(G)} \bigwedge_{\substack{v \in V(G) \\ v \neq u}} \neg x_{i,j,k,u} \vee \neg x_{i,j,k,v}$$

"Chaque sommet apparaît au plus une fois"

$$\phi_5 = \bigwedge_{u \in V(G)} \bigwedge_{j=0}^{k} \bigwedge_{\substack{j2=0 \\ j2 \neq j}}^{k} x_{i,j,k,u} \vee x_{i,j2,k,u}$$

"Chemin de taille k continue"

$$\phi_6 = \bigwedge_{j=0}^{k-1} \bigwedge_{u \in V(G)} x_{i,j,k,u} \wedge \bigvee_{v \in N(u)} x_{i,j+1,k,v}$$

"Distance commune"

$$\phi = \bigwedge_{i=1}^{N} \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5 \wedge \phi_6$$

3.1.3 Formule satisfaisable

(c) Démontrez que s'il existe un chemin simple et valide de longueur k dans chacun des graphes, alors votre formule est satisfaisable (vous décrirez la valuation satisfaisant votre formule en fonction de ces chemins).

R : Supposons qu'il existe un chemin simple et valide de longueur k allant de s à t dans chacun des graphes.
Le chemin allant de s à t, pour tous les graphes on a bien $x_{i,0,k,s}$ et $x_{i,k,k,t}$ donc ϕ_1 et ϕ_2 sont satisfaisables pour tous les graphes.
Le chemin étant simple et de taille k, alors ϕ_3, ϕ_4 et ϕ_5 sont satisfaisables.
Le chemin étant continu, alors ϕ_6 est bien satisfaisable.
Alors par construction, s'il existe un chemin simple et valide de k allant de s à t, ϕ est donc satisfaisable.

3.1.4 Chemin valide

(d) Démontrez que s'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).

R : Si ϕ_1 et ϕ_2 sont satisfaisables, alors il doit exister un chemin allant s à t.
Si ϕ_3 et ϕ_4 sont satisfaisables, alors le chemin en question est de taille k.
Si ϕ_5 est satisfaisable, alors le chemin ne passe pas deux fois par le même sommet, il est donc simple.
Si ϕ_6 est satisfaisable, alors le chemin est continu.

Par construction, si ϕ est satisfaisable, alors il existe un chemin simple et continu detaille k allant de s à t.

3.1.5 Réduction vers *SAT*

(e) Déduisez-en la réduction de Distance commune vers SAT en découpant votre réduction en sous-formules simples, dont vous expliciterez le rôle.

R :

$$\phi' = \bigvee_{k=1}^{kmax} \bigwedge_{i=1}^{N} \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5 \wedge \phi_6$$

où $kmax$ correspond au k maximal défini à la question **3.1.a**.

3.2 Formules pour chaque *k*

2. Vous pouvez aussi générer des formules pour chaque valeur de k = 1,2, ..., et les tester une par une.

3.2.1 Algorithme

(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.

R : On crée la formule pour $k = 1$, si elle est satisfaisable on s'arrête, sinon on teste pour $k = 2$ et ainsi de suite jusqu'à $kmax$.

3.2.2 Moins de variables

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

R : On peut obtenir moins de variables sur l'ensemble de l'algorithme, car nous n'avons que pour les chemins de taille k que nous vérifions et non par pour tout les k inférieurs à $kmax$.

3.2.3 Gain

(c) Quel est le gain pour cette approche ?

R : On ne fait pas les formules pour tous les k, on s'arrête à la première satisfaisable, s'il y en a une.

3.2.4 Équivalence

(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.

R : Il suffit de faire un $\bigvee$ sur les formules pour passer de chemins de taille k avec un k fixé à un chemin de taille au plus $kmax$.

3.3 Bonus : CNF

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

R : Soit *kmax*, le k maximal défini à la question **3.1.a** :

"Point de départ s"

$$\phi''_1 = \bigvee_{k=1}^{kmax} x_{i,0,k,s}$$

"Point d'arrivée t"

$$\phi''_2 = \bigvee_{k=1}^{kmax} x_{i,k,k,t}$$

"Au moins 1 sommet par position"

$$\phi''_3 = \bigwedge_{j=0}^{k} \bigvee_{u \in V(G)} \bigvee_{k=1}^{kmax} x_{i,j,k,u}$$

"Au plus 1 sommet par position"

$$\phi''_4 = \bigwedge_{j=0}^{k} \bigwedge_{u \in V(G)} \bigwedge_{\substack{v \in V(G) \\ v \neq u}} \bigvee_{k=1}^{kmax} \neg x_{i,j,k,u} \vee \neg x_{i,j,k,v}$$

"Chaque sommet apparaît au plus une fois"

$$\phi''_5 = \bigwedge_{u \in V(G)} \bigwedge_{j=0}^{k} \bigwedge_{\substack{j2=0 \\ j2 \neq j}}^{k} \bigvee_{k=1}^{kmax} x_{i,j,k,u} \vee x_{i,j2,k,u}$$

"Chemin de taille k continue"

$$\phi''_6 = \bigwedge_{j=0}^{k-1} \bigwedge_{u \in V(G)} (\bigvee_{k=1}^{kmax} x_{i,j,k,u}) \wedge (\bigvee_{v \in N(u)} \bigvee_{k=1}^{kmax} x_{i,j+1,k,v})$$

"Distance commune CNF"

$$\phi'' = \bigwedge_{i=1}^{N} \phi''_1 \wedge \phi''_2 \wedge \phi''_3 \wedge \phi''_4 \wedge \phi''_5 \wedge \phi''_6$$