IFT503/711 – Théorie du calcul

Université de Sherbrooke

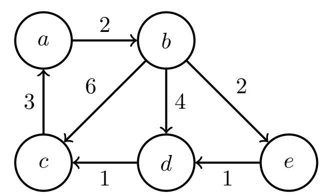
Devoir 4

Enseignants:	Michael Blondin (page 1) et Manuel Lafond (page 2)
Date de remise:	jeudi 1er avril 2021 à 10h29
À réaliser:	individuellement ou à deux au 1er cycle individuellement aux cycles supérieurs
Modalités:	remettre en ligne sur Turnin
Pointage:	sur 40 points au 1er cycle (+ 5pts bonus pour ★) sur 50 points aux cycles supérieurs

Question 1. 10 pts

La *distance* d'un sommet s vers un sommet t dans un graphe dirigé pondéré est le coût d'un plus court chemin de s vers t. Par exemple, dans le graphe ci-dessous, la distance de a vers c est de 6.

Montrez que le problème PCC appartient à NC:



PCC

ENTRÉE: un graphe dirigé pondéré $\mathcal{G} = (V, E)$ décrit par une matrice de coûts positifs, deux sommets $s, t \in V$, et un entier positif k sous représentation binaire;

QUESTION: la distance de s vers t dans $\mathcal{G}$ est-elle égale à k?

Il n'est *pas* nécessaire d'argumenter que votre famille de circuits est uniforme sous espace logarithmique, mais il faut brièvement expliquer pourquoi elle est uniforme sous temps polynomial.

Question 2. 10 pts

Soit X un ensemble fini. Nous disons que $S \subseteq X$ *engendre* $t \in X$ sous une *opération binaire* $\star \colon X \times X \to X$ s'il est possible d'appliquer $\star$ à partir d'éléments de S jusqu'à l'obtention de t. Par exemple, pour l'opération décrite par la table ci-dessous, $S = \{a, c\}$ engendre e car $(a \star a) \star (c \star a) = b \star d = e$.

Montrez que le problème GÉNÉRATION est P-complet:

$\star$	**a**	**b**	**c**	**d**	**e**
a	b	b	a	a	a
b	b	d	a	e	b
c	d	a	a	a	a
d	c	c	a	d	c
e	c	c	a	d	e

GÉNÉRATION

ENTRÉE: un ensemble fini X, une opération binaire $\star \colon X \times X \to X$ représentée sous forme de table, un sous-ensemble $S \subseteq X$, et un élément $t \in X$;

QUESTION: est-ce que S engendre t sous l'opération $\star$? *Indice: voyez une porte comme deux éléments.*

Il n'est *pas* nécessaire d'expliquer pourquoi votre réduction se calcule en espace logarithmique.

★ Question 3. (cycles supérieurs)

Montrez que REG $\subseteq$ NC^1, où REG est l'ensemble des langages réguliers. ★ 5 pts

Indice: afin de déterminer si un automate accepte un mot w, considérez des sous-mots de w de taille $1, 2, 4, \ldots$

Question 4.

(a) Soit $G = (V, E)$ un graphe. Un sous-ensemble $X \subseteq V$ est appelé *dominant* si, pour tout sommet $u \in V \setminus X$, il existe $v \in X$ tel que $uv \in E$. Dans le problème MIN-k-DOMSET, on reçoit un graphe dans lequel chaque sommet a exactement k voisins, et on cherche un ensemble dominant de G de taille minimum. 3 pts

Montrez qu'il existe une $(k+1)$-approximation pour le problème MIN-k-DOMSET.

(b) Nous avons vu que MAX-SAT, qui cherche à maximiser le nombre de clauses satisfaites, admet une $\frac{1}{2}$-approximation. Considérez MIN-UNSAT, le problème analogue de minimisation. Dans le problème MIN-UNSAT, on reçoit une formule booléenne CNF ϕ, et on cherche une assignation qui minimise le nombre de clauses de ϕ non-satisfaites. 3 pts

Montrez que si P $\neq$ NP, il n'existe pas de c-approximation pour MIN-UNSAT pour toute constante $c > 1$.

(c) Dans le problème MAX-TRUE-SAT, on reçoit en entrée une formule booléenne ϕ qui est satisfaisable en assignant toutes les variables à *false* (ceci se vérifie facilement). On veut trouver une assignation qui satisfait ϕ et qui maximise le nombre de variables assignées à *true*. 4 pts

Montrez que si P $\neq$ NP, il existe une constante ρ telle qu'il n'existe pas de ρ-approximation pour MAX-TRUE-SAT.

Suggestion: Pas trivial. Ma réduction utilise IND-SET et les clauses ne contiennent que des variables négatives.

Question 5.

(a) Montrez que PCP$(0, 0)$ = P. 2 pts

(b) Montrez que PCP$(\log n, 0)$ = P (donc que $\log n$ bits aléatoires n'ajoutent pas de puissance à P). 4 pts

(c) Montrez que s'il existe un vérificateur $(\log(\log n), q)$-PCP pour SAT, où q est une constante, alors P $=$ NP. Notez qu'il n'y a aucune constante qui affecte le $\log\log n$. 4 pts

★ Question 6. (cycles supérieurs)

Nous allons démontrer que MAX-CLIQUE n'admet pas de (n^ε)-approximation pour une certaine constante ε. Étant donné un graphe G, on dénote par $\omega(G)$ la taille maximum d'une clique de G. Vous pouvez (et devriez) utiliser le théorème suivant en boîte noire (le démontrer dépasserait largement les objectifs du cours). 5 pts

Théoreme. *Soit $G = (V, E)$. Il existe un algorithme en temps polynomial en $|V|$ qui, sur entrée G, donne en sortie un graphe H qui satisfait*

$$n^{100}\left(\frac{\omega(G)}{n} - \frac{1}{100}\right)^{\log n} \leq \omega(H) \leq n^{100}\left(\frac{\omega(G)}{n} + \frac{1}{100}\right)^{\log n}$$

On sait qu'il est difficile de distinguer si un graphe G a une clique d'au plus $n/6$ sommets, ou une clique d'au moins $n/3$ sommets. Montrez qu'il est difficile de distinguer si un graphe a une clique d'au plus $f(n)$ sommets, ou une clique d'au moins $n^\varepsilon f(n)$ sommets pour une une certaine fonction f et une certaine constante ε.

Suggestion: réduisez MAX-CLIQUE à MAX-CLIQUE en utilisant H pour amplifier le "gap". Aussi, notez que $c^{\log n} = n^{\log c}$.