Examen du cours Complexité (L3)

Les documents (notes, polycopiés, ..) et calculatrices (téléphone, tablette, ..)
ne sont pas autorisés.

Date : 13 janv. 2025 à 10h45 / Durée : 2 heures

Exercice : Fonctions polylogspace

Pour un entier $k > 0$ et un alphabet fini A, une fonction totale $f : A^* \to A^*$ est dite $\log^k$space s'il existe une machine de Turing déterministe qui calcule $f(x)$ pour tout $x \in A^*$ en utilisant un espace de travail $O(\log^k n)$, c.-à-d. $\leq c(\log |x|)^k$, pour un entier c. On dit que f est *logspace* quand $k = 1$, et *polylogspace* si elle est $\log^k$space pour un $k \in \mathbb{N}$. **Note :** pour simplifier les calculs, on parle ici d'une fonction discrète $\log : \mathbb{N} \to \mathbb{N}$ définie par $\log(0) = 0$ et, pour $n > 0$, $\log(n) = \lfloor \log_2 n \rfloor$. Toujours pour simplifier, on convient que l'adjectif *logspace* et ses dérivés sont invariables.

1. Donnez une fonction polylogspace qui n'est pas calculable en temps polynomial. Justifiez.
2. Montrez que si f est $\log^k$space alors $|f(x)|$ est en $|x|^{O(\log^{k-1} |x|)}$.
3. Est-ce que la composition de deux fonctions $\log^k$space est elle-même $\log^k$space ? Justifiez.

Problème : Sous-mots et sous-suites

On dit qu'un mot u est sous-mot d'un mot v, noté $u \preccurlyeq v$, si u est une sous-suite de v, c.-à-d., obtenue en retirant un nombre arbitraire de lettres. P.ex. `examen` $\preccurlyeq$ `inexorablement` et `examen` $\not\preccurlyeq$ `existentialisme`. En particulier, $\epsilon \preccurlyeq u \preccurlyeq u$ pour tout u, où ϵ dénote le mot vide.
On s'intéresse au problème LCS (pour "Long Common Subword"). Le problème LCS prend en entrée un alphabet A, une liste $u_1, \ldots, u_k$ de mots dans A^*, ainsi qu'un entier N. La question à résoudre est « existe-t-il un mot $v \in A^*$ tel que $|v| = N$ et $v \preccurlyeq u_i$ pour $i = 1, \ldots, k$? »

4. Parmi les deux instances suivantes

$$(A = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\},\ u_1 = \mathtt{aabc},\ u_2 = \mathtt{bbca}, u_3 = \mathtt{ccab},\ N = 2)\,,$$
$$(A = \{\mathtt{0}, \mathtt{1}, \ldots, \mathtt{9}\},\ u_1 = \mathtt{31415926535897932384 6},\ u_2 = \mathtt{14142135623730950488 0},\ N = 8)\,,$$

   dites laquelle est positive et justifiez brièvement.

On précise que le problème LCS est un langage sur un alphabet Σ fixé. Donc la donnée d'un alphabet A (qui peut être n'importe quel alphabet fini) est codée sur Σ et la donnée des u_i reprend ce codage. Par ailleurs l'entier N est donné en base 1, p.ex. sous la forme $\mathtt{I}^N = \mathtt{II}\cdots\mathtt{I}$ avec $\mathtt{I} \in \Sigma$. En fin de compte, la taille d'une instance sera en $O\big(k+N+(|A|+|u_1|+\cdots+|u_k|)\log|A|\big)$ où le facteur multiplicatif $\log |A|$ rend compte du codage des lettres de A sur l'alphabet Σ.

5. Montrez que LCS est dans NP.

Soit LCS_b le problème qui est comme LCS sauf que N est écrit en base 2, de sorte que la taille d'une instance est en $O(k + \log N + (|A| + |u_1| + \cdots + |u_k|) \log |A|)$.

6. Donnez une réduction logspace de LCS à LCS_b.
   Donnez ensuite une réduction logspace de LCS_b à LCS.

   Pour ces deux questions on justifiera la correction et on détaillera (sans forcément écrire un programme) les algorithmes utilisés par les réductions de façon à bien comprendre comment un espace logarithmique est suffisant.

On veut montrer que LCS est NP–difficile. Pour cela on part du problème NODECOVER vu en TD et connu pour être NP-complet. On rappelle qu'une instance de NODECOVER est constituée d'un graphe simple (i.e., non orienté, sans arêtes multiples ni boucles) $G = (V, E)$ et d'un entier K et qu'on se demande si G admet un recouvrement de cardinal au plus K, sachant qu'un recouvrement est un ensemble de sommets $C \subseteq V$ tel que chaque arête de E a au moins une de ses extrémités dans C.

Soit une instance $G = (V, E), K$ avec $|V| = \ell$ sommets et $|E| = m$ arêtes. On va considérer des mots $u_0, u_1, \ldots, u_m$ sur l'alphabet V. On pose d'abord $u_0 = v_1 v_2 \ldots v_l$ en fixant une énumération de V. On fixe ensuite une énumération $e_1, e_2, \ldots, e_m$ des arêtes et pour, $i \in \{1, \ldots, m\}$, on pose $e_i = \{v_r, v_s\}$ de sorte que $r < s$. On pose alors

$$u_i = v_1 v_2 \ldots v_{r-1} v_{r+1} \ldots v_l v_1 v_2 \ldots v_{s-1} v_{s+1} \ldots v_l$$

7. Prouvez que G admet un recouvrement de taille $\ell - N$ si et seulement si il existe un mot w de longueur N qui soit sous-mot de chacun des u_i pour $i = 0, \ldots, m$.
8. Donnez une réduction logspace de NODECOVER à LCS. Justifiez soigneusement (sans écrire de code) que votre réduction est bien calculable en espace logarithmique.
9. Conclure en donnant la complexité de LCS et celle de LCS_b.

On s'intéresse à deux versions de LCS. Pour deux mots u, v, on note $u \preccurlyeq_{\text{no}} v$, et on dit que « u est un sous-mot non orienté de v », ssi $u \preccurlyeq v$ ou $\widetilde{u} \preccurlyeq v$. Ici $\widetilde{u}$ est le mot miroir de u : p.ex. $\widetilde{\texttt{engager}} = \texttt{regagne}$. De même on note $u \preccurlyeq_{\text{cy}} v$, et on dit que « u est cycliquement sous-mot de v », si $u_2 u_1 \preccurlyeq v$ pour une factorisation $u = u_1 u_2$ de u. P.ex., $\texttt{avril} \preccurlyeq_{\text{cy}} \texttt{invraisemblable}$ puisque avril est sous-mot de $\texttt{able} \cdot \texttt{invraisembl}$.

10. Montrez que $\preccurlyeq_{\text{no}}$ et $\preccurlyeq_{\text{cy}}$ sont des préordres, c.-à-d., des relations réflexives et transitives, sur les mots ?

Le problème LCS_{no} est une version de LCS où on demande s'il existe un sous-mot non orienté v de longueur N tel que $v \preccurlyeq_{\text{no}} u_i$ pour $i = 1, \ldots, k$?

11. Montrez que LCS_{no} est NP-complet.
12. Même question pour le problème LCS_{cy} qui demande si les u_is admettent un sous-mot commun de longueur N au sens de $\preccurlyeq_{\text{cy}}$.

On note LCS_2 la restriction de LCS où les instances contiennent exactement deux mots, i.e., $k = 2$.

13. Donnez un algorithme en PTIME qui résout LCS_2. Justifiez sa correction et votre analyse de complexité.

Plus généralement, pour $k \in \mathbb{N}$ fixé, le problème LCS_k est la restriction de LCS où les instances contiennent exactement k mots $u_1, \ldots, u_k$.

14. Soit $k \in \mathbb{N}$. Est-ce que LCS_k est dans PTIME ?