Examen du cours Complexité (L3)

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

Date : 12 janv. 2026 à 10h45 / Durée : 2 heures

1 Diverses lois pour une clôture

On rappelle que le problème BinOpGen vu en cours, aussi appelé ClotureDUneLoiBinaire dans le polycopié, est le langage de tous les $(X, *, S, t)$ tels que $t \in \langle S \rangle_*$, où X est un ensemble fini, $* : X \times X \to X$ est une opération binaire sur X, $S \subseteq X$ est une partie de X et $t \in X$ un élément. Ici $\langle S \rangle_*$ désigne la *clôture* de S par $*$, c.-à-d. le plus petit sous-ensemble de X contenant S et fermé pour $*$. Formellement on définit $S_{0,*} \subseteq S_{1,*} \subseteq S_{2,*} \subseteq \cdots \subseteq X$ la suite donnée par :

$$S_{0,*} \stackrel{\text{def}}{=} S\,, \qquad \forall i \in \mathbb{N} :\ S_{i+1,*} \stackrel{\text{def}}{=} S_{i,*} \cup \{x * y \mid x, y \in S_{i,*}\}\,, \tag{†}$$

de sorte que $\langle S \rangle_* \stackrel{\text{def}}{=} \bigcup_{i \in \mathbb{N}} S_{i,*}$. Il a été vu en cours que BinOpGen est PTIME-complet. On considère aussi 2BinOpGen qui s'intéresse à la clôture par 2 opérations binaires, notées $*_1$ et $*_2$, sur X. Formellement 2BinOpGen demande si $t \in \langle S \rangle_{*_1,*_2}$ où on définit $\langle S \rangle_{*_1,*_2}$ comme $\bigcup_{i \in \mathbb{N}} S_{i,*_1,*_2}$, basé sur

$$S_{0,*_1,*_2} \stackrel{\text{def}}{=} S\,, \qquad S_{i+1,*_1,*_2} = S_{i,*_1,*_2} \cup \{x *_1 y, x *_2 y \mid x, y \in S_{i,*_1,*_2}\}\,. \tag{‡}$$

Pour de tels problèmes, on peut supposer sans perte de généralité que l'input X est de la forme $\{1, \ldots, n\}$ et qu'une opération telle que $*$ est donnée sous la forme d'une table ou d'une matrice $n \times n$ à valeur dans X.

1. Donnez une réduction (logspace) montrant BinOpGen $\leq$ 2BinOpGen ainsi qu'une réduction montrant 2BinOpGen $\leq$ BinOpGen (c.-à-d. que les deux problèmes sont « inter-réductibles »). Justifiez votre définition.

TernOpGen est une autre variante où on se donne une opération *ternaire*, $\phi : X^3 \to X$, et où on demande, étant donné (X, ϕ, S, t), si $t \in \langle S \rangle_\phi$. Ici on remplace (†) par $S_{i+1,\phi} \stackrel{\text{def}}{=} S_{i,\phi} \cup \{\phi(x, y, z) \mid x, y, z \in S_{i,\phi}\}$.

2. Montrez que BinOpGen et TernOpGen sont inter-réductibles.

On considère ensuite $\text{BinOpGen}_{\text{assoc}}$ qui est BinOpGen restreint au cas où l'opération $*$ est associative, c.-à-d. qu'elle vérifie $x * (y * z) = (x * y) * z$ pour tous $x, y, z \in X$.

3. Montrez que $\text{BinOpGen}_{\text{assoc}}$ et GAP (le *Graph Accessibility Problem* vu en cours) sont inter-réductibles.

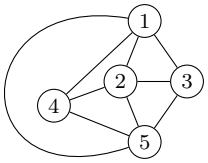
On considère enfin le problème $\text{BinOpGen}_{\text{comm},1}$ qui est BinOpGen restreint à des opérations $*$ *commutatives*, et à des ensembles $S = \{s\}$ ne contenant qu'un élément.

4. Montrez que $\text{BinOpGen}_{\text{comm},1}$ est PTIME-complet.

2 Cliques planaires

Un graphe non orienté est dit *planaire* s'il peut être dessiné dans le plan de telle sorte que ses arêtes ne s'intersectent pas. Les arêtes peuvent être dessinées par des lignes courbes arbitraires (continues). Voici un exemple $G_{\text{ex}} = (V, E)$ de graphe planaire et une façon possible de le dessiner dans le plan :

$$V = \{1, 2, 3, 4, 5\}$$
$$\text{E}=\{\{1,2\},\{1,3\},\{1,4\},\{1,5\},\ldots\}$$



5. Expliquer pourquoi le graphe complet K_4 à 4 sommets est un graphe planaire et pourquoi K_5 ne l'est pas. (On rappelle que K_n a n sommets $V_n = \{1, 2, \ldots, n\}$ et $\frac{n(n-1)}{2}$ arêtes $E_n = \{\{i, j\} \mid 1 \leq i < j \leq n\}$).

Le problème PlGrClique (pour « Planar Graph Clique ») consiste, étant donnés un graphe planaire $G = (V, E)$ et un entier k, à décider s'il existe dans G un sous-ensemble de k sommets qui forment une clique (c.-à-d. que E contient toutes les arêtes entre ces k sommets).

6. Est-ce que $(G_{\text{ex}}, 4) \in$ PlGrClique ?
7. Quelles arêtes faut-il ajouter à G_{ex} pour obtenir un nouveau graphe G' tel que $(G', 5) \in$ PlGrClique ?

On peut considérer deux versions de PlGrClique : l'une où l'input k est écrit en base 10 ou 2, donc est de taille $O(\log k)$, et l'autre où k est donné en unaire, donc est de taille $O(k)$.

8. Montrez que ces deux versions sont inter-réductibles.

On suppose dans un premier temps qu'on sait tester efficacement (disons en temps linéaire et en espace logarithmique) si un graphe donné est planaire.

9. Sous cette hypothèse, le problème PlGrClique est-il NP-difficile ou bien PTIME-complet ou bien NL-complet ou bien dans LOGSPACE ? Justifiez.
10. Expliquez comment votre réponse à la question 9 dépend de l'hypothèse posée juste en amont.

On voudrait lever en partie l'hypothèse sur le test de planarité. Pour ce faire on peut supposer que l'input G du problème est fourni avec un *certificat de planarité* w facile à valider. On voudrait donc que le langage PlGr des graphes planaires soit de la forme

$$\text{PlGr} = \{G \mid \exists w : |w| \leq p(|G|) \wedge (G, w) \in \text{CPlGr}\}$$

pour un polynôme p et un langage CPlGr (pour « graphes à planarité certifiée ») qui soit dans LOGSPACE.

11. Proposez un tel langage CPlGr.

    Cette dernière question est assez ouverte, elle s'adresse aux étudiants qui ont complètement répondu aux questions précédentes. Le correcteur considérera avec bienveillance des propositions intéressantes même si vous ne savez pas (ou n'avez pas le temps de) justifier la correction (« w existe ssi G est planaire »), la borne sur la taille de w, ou l'appartenance de CPlGr à LOGSPACE.