

Examen du cours Complexité (L3)

Les documents (notes, photocopiés, ..) 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, $*$: $X \times X \rightarrow X$ 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 $S_{0,*} \subseteq S_{1,*} \subseteq S_{2,*} \subseteq \dots \subseteq X$ la suite donnée par :

$$S_{0,*} \stackrel{\text{def}}{=} S, \quad \forall i \in \mathbb{N} : S_{i+1,*} \stackrel{\text{def}}{=} S_{i,*} \cup \{x * y \mid x, y \in S_{i,*}\}, \quad (\dagger)$$

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

$$S_{0, *_1, *_2} \stackrel{\text{def}}{=} S, \quad S_{i+1, *_1, *_2} = S_{i, *_1, *_2} \cup \{x *_1 y, x *_2 y \mid x, y \in S_{i, *_1, *_2}\}. \quad (\ddagger)$$

Pour de tels problèmes, on peut supposer sans perte de généralité que l'input X est de la forme $\{1, \dots, n\}$ 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 $\text{BinOpGen} \leq \text{2BinOpGen}$ ainsi qu'une réduction montrant $\text{2BinOpGen} \leq \text{BinOpGen}$ (c.-à-d. que les deux problèmes sont « inter-réductibles »). Justifiez votre définition.

Solution:

La réduction montrant $\text{BinOpGen} \leq \text{2BinOpGen}$ est facile : $r : (X, *, S, t) \mapsto (X, *, *_1, S, t)$ convient. C'est évidemment correct et logspace.

Pour $\text{2BinOpGen} \leq \text{BinOpGen}$, on peut p.ex. définir

$$r : (X, *_1, *_2, S, t) \mapsto (X', *', S', t')$$

où $X' \stackrel{\text{def}}{=} X \times \{1, 2\}$ et où $*', S'$ et t' sont donnés par

$$(x, k) *' (y, l) \stackrel{\text{def}}{=} (x *_k y, l), \quad S' \stackrel{\text{def}}{=} S \times \{1, 2\}, \quad t' \stackrel{\text{def}}{=} (t, 1).$$

En d'autres termes, X' contient deux copies de X « coloriées avec la couleur 1 ou la couleur 2 » et le produit $(x, k) *' (y, l)$ de deux valeurs coloriées combine x et y via l'opération associée à la couleur de (x, k) et colorie le résultat comme (y, l) . Pour la correction il suffit de vérifier que $S'_{i, *} = S_{i, *_1, *_2} \times \{1, 2\}$ pour tout $i \in \mathbb{N}$, ce qui se fait facilement par induction sur i (omis). On peut alors affirmer que $t \in \langle S \rangle_{*_1, *_2}$ ssi $(t, 1) \in \langle S' \rangle_{*}'$, ce qui est exactement la correction recherchée. Enfin, r est évidemment logspace puisque on produit une table pour $*'$ en émettant 4 entrées pour $*'$ à partir de chaque entrée de la table pour $*$ (qu'il suffit de parcourir).

TernOpGen est une autre variante où on se donne une opération ternaire, $\phi : X^3 \rightarrow X$, et où on demande, étant donné (X, ϕ, S, t) , si $t \in \langle S \rangle_\phi$. Ici on remplace (\dagger) 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.

Solution:

Montrer $\text{BinOpGen} \leq \text{TernOpGen}$ est très facile. On peut p.ex. définir $r : (X, *, S, t) \mapsto (X, \phi, S, t)$ avec $\phi(x, y, z) \stackrel{\text{def}}{=} x * y$. On obtient alors $x \in S_{i, *}$ ssi $x \in S_{i, \phi}$ pour tous $x \in X$ et $i \in \mathbb{N}$, ce qui donne la correction.

Pour montrer $\text{TernOpGen} \leq \text{BinOpGen}$, on peut proposer une réduction de la forme

$$r : (X, \phi, S, t) \mapsto (X', *, S, t)$$

avec $X' \stackrel{\text{def}}{=} X \cup X^2$ (on suppose que X et X^2 sont disjoints). Maintenant $*$ doit combiner à la fois des éléments de X , mais aussi des paires tirées de X^2 . On pose, pour tous $x, y, z, u \in X$,

$$x * y \stackrel{\text{def}}{=} (x, y), \quad x * (y, z) \stackrel{\text{def}}{=} \phi(x, y, z), \quad (x, y) * (z, u) \stackrel{\text{def}}{=} (x, y) * z \stackrel{\text{def}}{=} (x, y).$$

On obtient alors $S_{i, \phi} \subseteq S_{2i, *}$ ainsi que $S_{i, *} \subseteq S_{i, \phi} \cup (S_{i, \phi} \times S_{i, \phi})$, ceci pour tout i (preuve par induction omise). De là on déduit $t \in \langle S \rangle_*$ ssi $t \in \langle S \rangle_\phi$, ce qui est la correction de la réduction. Enfin r est logspace (on peut produire X' et $*$ au moyen de deux et, respectivement, quatre boucles imbriquées sur X).

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.

Solution:

Pour $\text{GAP} \leq \text{BinOpGen}_{\text{assoc}}$ on renvoie au corrigé du TD6.

Pour montrer $\text{BinOpGen}_{\text{assoc}} \leq \text{GAP}$ on peut donner un algorithme montrant $\text{BinOpGen}_{\text{assoc}} \in \text{NL}$, ce qui suffira puisque GAP est NL-complet. (NB : le corrigé du TD6 ne donne pas explicitement le critère de terminaison de l'algorithme proposé).

Une réduction plus explicite a la forme $r : (X, *, S, t) \mapsto (G = (V, E), \$, t)$ avec $V \stackrel{\text{def}}{=} X \cup \{\$\}$ (on suppose $\$ \notin X$) et où les arêtes sont données par

$$E \stackrel{\text{def}}{=} \{(x, x * y) \mid x \in V, y \in S\} \text{ en étendant } * \text{ à } V \text{ via } \$ * x = x * \$ \stackrel{\text{def}}{=} x.$$

On prouve alors la correction :

— Si $t \in \langle S \rangle_*$, et puisque $*$ est associative, on peut écrire $t = x_1 * x_2 * \dots * x_\ell$ pour des éléments $x_1, \dots, x_\ell \in S$. En notant p_i le produit partiel $x_1 * x_2 * \dots * x_i$, on voit alors que G contient le chemin

$$\$ \rightarrow \$ * x_1 = p_1 \rightarrow p_1 * x_2 = p_2 \rightarrow p_2 * x_3 = p_3 \rightarrow \dots \rightarrow p_{\ell-1} \rightarrow p_{\ell-1} * x_\ell = p_\ell = t.$$

— Réciproquement tout chemin $\$ \xrightarrow{*} t$ de G avec $t \in X$ est de la forme $\$ = x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_m = t$. Par définition de E tout élément x_i (avec $i \geq 1$) est de la forme $x_i = x_{i-1} * y_i$ pour un y_i de S . Donc $x_1 = y_1$ et $x_i = y_1 * y_2 * \dots * y_i$ avec au final $t = x_m = y_1 * \dots * y_m \in \langle S \rangle_*$.

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.

Solution:

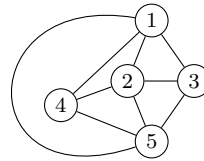
La réduction $\text{HornSat} \leq \text{BinOpGen}$ vue en cours est déjà une réduction montrant $\text{HornSat} \leq \text{BinOpGen}_{\text{comm},1}$, d'où l'on déduit la PTIME-difficulté. L'appartenance à PTIME est héritée de BinOpGen . (NB : Pour être complet il faut ajouter qu'on peut effectivement vérifier en temps polynomial si une instance est de la bonne forme, c.-à-d. si $*$ est bien commutative et si S est bien un singleton.)

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 :

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

$$E = \{\{1,2\}, \{1,3\}, \{1,4\}, \{1,5\}, \dots\}$$



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 $V_n = \{1, 2, \dots, n\}$ et $\frac{n(n-1)}{2}$ arêtes $E_n = \{\{i, j\} \mid 1 \leq i < j \leq n\}$).

Solution:

K_3 est dessiné via un triangle ABC, donc est planaire, et tous les dessins possibles sont topologiquement équivalents.

Pour passer de K_3 à K_4 on peut positionner le 4^e sommet D à l'intérieur du triangle ABC et le connecter aux 3 autres sommets via des arêtes rectilignes. On peut aussi le positionner à l'extérieur et utiliser des arêtes courbes mais alors un des 3 sommets d'origine devient intérieur à la figure et, à permutation des sommets près, on obtient une figure topologiquement isomorphe à celle obtenue par le choix précédent. Donc K_4 est nécessairement représenté par un "triangle" ABC extérieur contenant un sommet intérieur D ainsi que trois arêtes intérieures divisant le triangle en trois sous-triangles.

Quand on veut ajouter un 5^e sommet E, soit on le place à l'extérieur d'ABC et il n'est pas possible de dessiner une arête DE sans intersecter le triangle extérieur, soit on le place à l'intérieur, donc dans un des sous-triangles, disons dans ABD, et alors il n'est pas possible de relier E à C sans intersecter le sous-triangle.

Le problème PIGrClique (pour « Planar Graph Clique ») consiste, étant donné 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 $(G_{\text{ex}}, 4) \in \text{PIGrClique}$?

Solution:

Oui. Par exemple le sous-graphe induit par $\{1, 2, 4, 5\}$ est une clique.

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

Solution:

Il n'y a pas de solution puisque un graphe G' qui contiendrait une clique de taille 5 ne serait pas planaire, comme vu à la question 5.

On peut considérer deux versions de **PIGrClique** : 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.

Solution:

C'est un raisonnement classique. Rappelons que la réduction « unaire \mapsto binaire » est facile (au moyen d'un compteur) et que dans l'autre direction il est impossible de transformer une écriture binaire en une écriture unaire en temps polynomial, et a fortiori par une réduction logspace. La solution pour contourner cette impossibilité utilise le fait qu'une clique dans un graphe G ne peut pas avoir de taille supérieure à $|V|$ (et même à 4 si G est planaire). Donc la réduction peut se contenter de produire une écriture unaire de $\max(k, 1 + |V|)$, ou de $\max(k, 5)$, ce qui est réalisable en logspace, et n'affectera pas la correction de la réduction.

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 **PIGrClique** est-il **NP**-difficile ou bien **PTIME**-complet ou bien **NL**-complet ou bien dans **LOGSPACE**? Justifiez.

Solution:

Comme vu en TD le problème de la clique est **NP**-complet pour des graphes arbitraires. Mais pour des graphes planaires il est logspace!

Un algorithme simple commence par tester si G est bien planaire et si $k < 5$, sans quoi on rejette immédiatement (cf. question 5). Quand $k < 5$ on peut énumérer tous les k -uplets de sommets (il y en a $|V|^k$) et tester si les $\frac{k(k-1)}{2}$ arêtes correspondantes sont présentes dans E . On a donc $O(n^{k+2})$ tests sur E . Puisque on ne considère que les situations où $k \leq 4$, cet algorithme utilise un nombre fixe de compteurs et est dans **LOGSPACE**.

10. Expliquez comment votre réponse à la question 9 dépend de l'hypothèse posée juste en amont.

Solution:

La définition du problème **PIGrClique** demande, en lisant un input x , de rejeter si x n'est pas une paire de la forme (G, k) pour G un graphe planaire et k un entier naturel. Comme expliqué en cours, les détails de la façon dont un graphe G est codé sous forme d'un mot sont assez indifférents du moment qu'on se réfère (implicitement) à des codages qui sont tous logspace-équivalents. On sait ainsi qu'il est facile de rejeter x s'il ne contient pas un codage de G en bonne et due forme. Mais il faut aussi rejeter si x contient (le codage d') un graphe G qui ne serait pas planaire. Tout algorithme pour **PIGrClique** doit donc être capable de décider si un graphe G est planaire (de fait le problème de la planarité se réduit à **PIGrClique** : il suffit d'adjoindre $k = 0$!). Donc notre réponse « **PIGrClique** est dans **LOGSPACE** » à la question 9 suppose qu'un test de planarité en espace logarithmique existe et nous avons effectivement utilisé cette hypothèse dans la 1^e instruction de l'algorithme donné plus haut! (NB : Un tel test existe bien, mais c'est une question d'algorithmique hors-sujet pour cet examen.)

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 **PIGr** des graphes planaires soit de la forme

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

pour un polynôme p et un langage **CPIGr** (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.

Solution:

Une solution connue consiste à lister les régions du plan (les « facettes ») délimitées par un dessin planaire fixé : pour chaque sommet on liste les arêtes incidentes dans l'ordre où on les voit (sur le dessin planaire) quand on tourne autour du sommet. Ce certificat est de taille linéaire. On n'expliquera pas ici quel critère (de correction) permettrait de dire quand une telle liste de facettes est réalisable, ni pourquoi ce critère est décidable en espace logarithmique (voir p.ex. Wikipedia pour en savoir plus).

Pour ceux qui le connaissent, des réponses basées sur le théorème de Kuratowski sont envisageables. Il faut alors expliquer comment on certifie que G ne contient ni K_5 ni $K_{3,3}$ comme « mineur », ce qui est plus compliqué que de certifier que K_5 et $K_{3,3}$ ne sont pas des sous-graphes induits.