Sujet M1 - Réécriture et automates - Corrigé

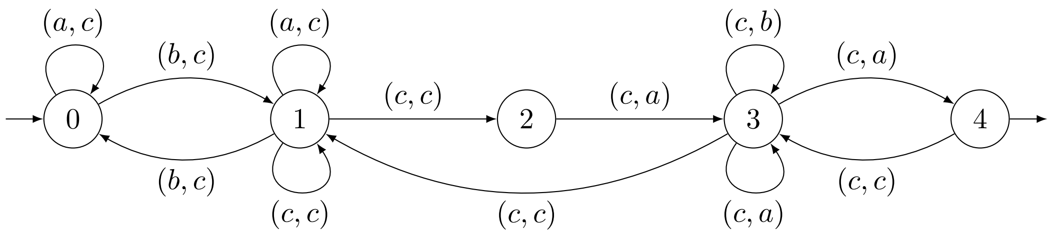
Dans ce sujet, on va étudier la décidabilité d'un problème sur des automates lisant des paires de mots : on s'interrogera alors de savoir s'il existe un mot de la forme $(uc^n, c^n u)$ accepté par l'automate. **Toutes les questions sont indépendantes et peuvent donc être traitées dans un ordre arbitraire.**

On précise tout d'abord la définition des automates finis que nous allons considérer. Un **automate** fini $\mathcal{A}$ sur l'alphabet A est la donnée d'un tuple $\mathcal{A} = \langle Q, E, I, T\rangle$ où Q est un ensemble fini d'états, $E \subseteq Q \times A \times Q$ est l'ensemble des transitions, $I \subseteq Q$ est l'ensemble des états initiaux et $T \subseteq Q$ est l'ensemble des états terminaux. Une transition $(p, a, q) \in E$ sera dénotée par une flèche $p \xrightarrow{a} q$. Un **calcul** c de $\mathcal{A}$ est une séquence finie de transitions qui forment un chemin dans le graphe de $\mathcal{A}$, $p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \cdots \xrightarrow{a_n} p_n$: on note un tel calcul $c = p_0 \xrightarrow{a_1a_2\cdots a_n} p_n$. L'étiquette de c est le mot $a_1a_2\cdots a_n$ de A^*. Le calcul est acceptant si $p_0 \in I$ et $p_n \in T$. Le **langage** de $\mathcal{A}$ est le sous-ensemble $\mathcal{L}(\mathcal{A})$ de A^* qui contient l'ensemble des étiquettes des calculs acceptants de $\mathcal{A}$. L'automate $\mathcal{A}$ est dit **déterministe** si I possède un unique élément et pour tout $p \in Q$ et $a \in A$, il existe au plus un état $q \in Q$ tel que $(p, a, q) \in E$.

Ici, on va considérer des automates sur un alphabet de la forme $A = (\Gamma \cup \{c\})^2$, avec $c \notin \Gamma$. Un mot w sur l'alphabet A s'écrit $w = (x_1, y_1)(x_2, y_2)\cdots(x_n, y_n)$ avec $x_1, y_1, x_2, y_2, \ldots, x_n, y_n \in \Gamma \cup \{c\}$. On notera un tel mot $w = (x_1x_2\cdots x_n, y_1y_2\cdots y_n)$. Le problème qu'on étudie est le suivant, qu'on appelle **problème du décalage** :

Donnée : un alphabet Γ, une lettre $c \notin \Gamma$ et un automate fini $\mathcal{A}$ sur l'alphabet $(\Gamma \cup \{c\})^2$.
Question : est-ce qu'il existe un mot de la forme $(uc^n, c^n u)$, avec $n \geq 0$ et $u \in \Gamma^*$, dans le langage $\mathcal{L}(\mathcal{A})$ reconnu par $\mathcal{A}$?

1. Montrer que la réponse du problème du décalage est « oui » pour l'alphabet $\Gamma = \{a, b\}$, la lettre $c \notin \Gamma$ et l'automate fini ci-dessous, dans lequel $I = \{0\}$ et $T = \{4\}$? *[Indication : il suffit d'exhiber un mot $u \in \Gamma^*$ et un entier $n \geq 0$ tel que l'automate accepte la paire $(uc^n, c^n u)$.]*



Le mot $(abacccc, ccccaba)$ est reconnu par l'automate, la réponse au problème est donc « oui ».

2. Montrer que le problème du décalage est décidable (en précisant la complexité de l'algorithme que vous trouverez) si on demande de plus que le mot u soit de longueur $|u|$ inférieure ou égale à n :

Donnée : un alphabet Γ, une lettre $c \notin \Gamma$ et un automate fini $\mathcal{A}$ sur l'alphabet $(\Gamma \cup \{c\})^2$.
Question : est-ce qu'il existe un mot de la forme $(uc^n, c^n u)$, avec $n \geq 0$ et $u \in \Gamma^*$ tel que $|u| \leq n$, dans le langage $\mathcal{L}(\mathcal{A})$ reconnu par $\mathcal{A}$?

Soit $u = a_1 \cdots a_m \in \Gamma^*$ avec $m \leq n$. Le mot $w = (uc^n, c^n u)$ est donc de la forme

$$w = (a_1, c)(a_2, c) \cdots (a_m, c)(c, c)^{m-n}(c, a_1)(c, a_2) \cdots (c, a_m)$$

À partir de l'automate $\mathcal{A} = \langle Q, E, I, T \rangle$ sur l'alphabet $(\Gamma \cup \{c\})^2$, on construit un automate $\mathcal{A}' = \langle Q^3, E', I', T' \rangle$ sur l'alphabet Γ qui lit un mot d'entrée $u \in \Gamma^*$ et initie deux calculs en parallèle simulant le comportement de $\mathcal{A}$ sur les entrée (u, c^m) et (c^m, u). Pour l'entrée (u, c^m), on commence le calcul dans un état de I et on continue le calcul dans la première composante de Q^3. On devine également un état $q \in Q$ dès le début, qui permet d'initier le calcul sur l'entrée (c^m, u). On stocke cet état q comme une constante dans la deuxième composante de Q^3, alors que la troisième composante sert à simuler le calcul de $\mathcal{A}$ sur l'entrée (c^m, u). Arrivé à la fin, on accepte le mot u si la troisième composante a atteint un état final et s'il existe un chemin dans $\mathcal{A}$, lisant un mot de $(c, c)^*$ entre l'état p de la première composante et l'état q de la seconde composante. On peut en effet aisément précalculer l'ensemble $\mathcal{P}$ de telles paires d'état (p, q) en temps quadratique en $|Q|$. Finalement, il ne reste plus qu'à décider le vide du langage $\mathcal{L}(\mathcal{A}')$ en temps linéaire en $|Q^3|$: on obtient donc un algorithme cubique pour résoudre le problème initial.

Précisément, l'automate $\mathcal{A}'$ est défini par :
— $I' = \{(i, q, q) \mid i \in I, q \in Q\}$;
— $T' = \mathcal{P} \times T$;
— $E' = \{((p, q, r), a, (p', q, r')) \mid (q, (a, c), q') \in E, (r, (c, a), r') \in E\}$.

On va montrer que le problème est indécidable sans cette précision sur la longueur de u. Pour cela, on va réduire un problème sur les systèmes de réécriture, qu'on montrera être indécidable par la suite. Un **système de réécriture** S sur un alphabet Σ consiste en un ensemble fini de règles hors-contexte de la forme $g \to d$ avec $g, d \in \Sigma^*$. Une telle règle s'applique à un mot $\alpha g \beta \in \Sigma^*$ et le réécrit en $\alpha d \beta$, ce qu'on indique en écrivant $\alpha g \beta \Longrightarrow \alpha d \beta$. On utilise la notation $\Longrightarrow^*$ pour la clôture réflexive et transitive de la relation $\Longrightarrow$. On dit qu'un système de réécriture S **préserve les longueurs** si toutes ses règles $g \to d$ vérifient que les longueurs des mots g et d sont identiques. Pour l'instant, on suppose démontré que le problème suivant est indécidable, qu'on appelle **problème de la puissance** :

Donnée : un alphabet Σ contenant, entre autres, deux symboles a et b, et un système de réécriture S préservant les longueurs.
Question : est-ce qu'il existe un entier $n \geq 1$ tel que $a^n \Longrightarrow^* b^n$?

3. On réduit le problème de la puissance au problème du décalage. On suppose donc fixée une instance du problème de la puissance, c'est-à-dire un alphabet Σ contenant a et b et un système de réécriture S préservant les longueurs, et on considère l'alphabet $\Gamma = \Sigma \cup \{d\}$, avec $d \notin \Sigma$.

a) Donner un langage régulier C sur l'alphabet $\Delta = (\Gamma \cup \{c\})^2$ tel que $x \Longrightarrow y$ si et seulement si $(y, x) \in C$.

On prend $C = E^* R E^*$ avec $E = \{(e, e) \mid e \in \Sigma\}$ et $R = \{(d, g) \mid g \to d \in S\}$. Du fait que S préserve la longueur, il vient immédiatement que $x \Longrightarrow y$ si et seulement si $(y, x) \in C$.

b) En déduire un langage régulier D sur l'alphabet Δ tel que $x_0 \Longrightarrow x_1 \Longrightarrow \cdots \Longrightarrow x_m$ si et seulement si $(dx_1dx_2 \cdots dx_md, dx_0dx_1 \cdots dx_{m-1}d) \in D$.

À cause de la préservation des longueurs, $(dx_1dx_2 \cdots dx_md, dx_0dx_1 \cdots dx_{m-1}d)$ peut s'écrire comme $(d, d)(x_1, x_0)(d, d)(x_2, x_1) \cdots (d, d)(x_m, x_{m-1})(d, d)$. On pose donc $D = (d, d)[C(d, d)]^*$.

c) Construire alors un automate fini $\mathcal{A}$ sur l'alphabet Δ tel que $a^{n-1} \Longrightarrow^* b^{n-1}$ si et seulement si $\mathcal{L}(\mathcal{A})$ contient un mot de la forme (uc^n, c^nu) et conclure.

On considère le langage rationnel $L = (d, c)(a, c)^+ D(c, b)^+(c, d)$. Grâce à la question précédente, on sait que si $a^{n-1} = x_0 \Longrightarrow x_1 \Longrightarrow \cdots \Longrightarrow x_m = b^{n-1}$ avec $n \geq 2$, on a $w = (d, d)(x_1, x_0)(d, d)(x_2, x_1) \cdots (d, d)(x_m, x_{m-1})(d, d) \in D$. Ainsi, $(d, c)(x_0, c^{n-1})w(c^{n-1}, x_m)(c, d) \in L$. Ce mot est de la forme $(dx_0dx_1d \cdots dx_mdc^n, c^ndx_0dx_1 \cdots dx_{m-1}dx_md)$. Donc il existe un mot (uc^n, c^nu) dans le langage L.

Réciproquement, supposons qu'il existe un mot (uc^n, c^nu) dans L, avec $u \in \Gamma^*$. D'après la question précédente, tout mot de L a la forme

$$w = (d, c)(a, c)^i(d, d)(y_0, x_0)(d, d)(y_1, x_1)(d, d) \cdots (y_m, x_m)(d, d)(c, b)^j(c, d)$$

avec $i, j \geq 1$ et $x_t \Longrightarrow y_t$ pour tout $1 \leq t \leq m$. On peut réécrire w sous la forme :

$$w = (da^idy_0dy_1d \cdots y_mdc^{j+1}, c^{i+1}dx_0dx_1d \cdots x_mdb^jd)$$

Puisque $w = (uc^n, c^nu)$ avec $u \in \Gamma^*$, on sait que $i = j = n - 1$, $n \geq 2$ et $u = da^idy_0dy_1d \cdots y_md = dx_0dx_1d \cdots x_mdb^jd$. Puisque d est un symbole frais, qui n'appartient pas à Σ, il s'en suit que : $x_0 = a^i$, $x_1 = y_0$, $x_2 = y_1$, ..., $x_m = y_{m-1}$ et $b^j = y_m$. Ceci implique donc que $a^{n-1} \Longrightarrow^* b^{n-1}$, avec $n - 1 \geq 1$.
On peut bien sûr produire un automate fini $\mathcal{A}$ reconnaissant le langage L, ce qui conclut la réduction du problème de la puissance au problème du décalage, et prouve l'indécidabilité du problème du décalage (si on admet l'indécidabilité du problème de la puissance).

On prouve maintenant l'indécidabilité du problème de la puissance, en réduisant le problème de l'arrêt des machines de Turing initialement sur bande vide. Pour fixer les notations, on supposera qu'une **machine de Turing** est un tuple $M = \langle Q, \Gamma, \delta, q_0, \mathsf{B}, q_f \rangle$, où Q est un ensemble fini d'états, $q_0 \in Q$ est l'état initial, $q_f \in Q$ est l'unique état final, Γ l'alphabet de la bande qui contient le symbole blanc B et δ la fonction (partielle) de transition $Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$, où L (respectivement R) signifie que la machine se déplace d'un cran vers la gauche (respectivement droite). Sans perte de généralité, on suppose que si M s'arrête, c'est dans l'état q_f : q_f est donc le seul état sans successeur dans δ.

4. Soit M une machine de Turing. On pose $\Sigma = \Gamma \cup Q \cup \{a, b, \$\}$, avec $a, b, \$$ des symboles n'apparaissant pas dans Γ et Q. On construit le système de réécriture S contenant, entre autres, les règles suivantes :

$$aa \to \$q_0 \tag{1}$$
$$a \to \mathsf{B} \tag{2}$$
$$\$q_f \to bb \tag{3}$$

a) Quelles sont toutes les possibilités de réécrire le mot a^n en plusieurs étapes dans S ?

Le mot a^n peut se réécrire en la concaténation de $\$q_0$ et B, un nombre total respectif de fois i et j, tels que $2i + j = n$. En particulier, l'une des réécritures possibles est $a^n \Longrightarrow^* \$q_0\mathsf{B}^{n-2}$.

b) Ajouter des règles au système S de sorte que tout mot de la forme uq_fv avec $u, v \in \Gamma^*$ puisse se réécrire (entre autres) dans S en un mot de la forme b^n.

On ajoute les règles :

$$q_fc \to cq_f \qquad \text{pour tout } c \in \Gamma \tag{4}$$
$$cq_f \to q_fb \qquad \text{pour tout } c \in \Gamma \tag{5}$$

c) Ajouter finalement des règles pour simuler un pas de calcul de la machine M et montrer que M s'arrête à partir de la bande vide si et seulement s'il existe $n \geq 1$ tel que $a^n \Longrightarrow^* b^n$.

On ajoute les règles pour simuler un pas de calcul :

$$q_ic \to dq_j \qquad \text{pour tout } (q_j, d, R) = \delta(q_i, c) \tag{6}$$
$$fq_ic \to q_jfd \qquad \text{pour tout } (q_j, d, L) = \delta(q_i, c) \text{ et } f \in \Gamma \tag{7}$$

Supposons que M s'arrête quand elle est lancée sur la bande vide et montrons que $a^n \Longrightarrow^* b^n$ pour un certain $n \geq 1$. Puisque M s'arrête, elle utilise au total un nombre k de cases de la bande. Soit $n = k + 2$. Grâce à la question a), on peut réécrire a^n en $\$q_0B^k$. On utilise alors les règles précédentes pour simuler la machine jusqu'à ce qu'elle s'arrête, dans l'état q_f. On utilise alors la question b) pour réécrire la bande en b^n.

Réciproquement, supposons que $a^n \Longrightarrow^* b^n$ pour un certain $n \geq 1$ et montrons que la machine M s'arrête. Pour que le mot se réécrive à la fin en b^n, tous les a doivent disparaitre par l'application des règles (1) et (2). Sans perte de généralité, on peut donc supposer qu'on commence par appliquer au maximum ces règles. Grâce à la question a), on sait que a^n se réécrit alors comme un mot de la forme $\$q_0B^{i_1}\$q_0B^{i_2}\cdots\$q_0B^{i_m}$ avec $i_1 + i_2 + \cdots + i_m = n - 2m$. Puisque \$ est uniquement dans le membre gauche de la règle (3), on voit qu'un tel mot va permettre de simuler m exécutions de la machine M en parallèle. La seule manière de finalement réécrire les $i_j + 2$ cases de chaque exécution en b^{i_j+2} est d'utiliser la règle (3) et les règles de la question b). Ceci n'est possible que si l'état q_f a été atteint, c'est-à-dire si la machine M s'est arrêtée dans cette exécution.

On a donc réduit le problème de l'arrêt au problème de la puissance, ce qui prouve que ce dernier est un problème indécidable.

On s'intéresse finalement à une autre variante du problème du décalage, basé sur la conjugaison de mots. On dit que deux mots x et y de Σ^* sont conjugués s'il existe deux mots u et v tels que $x = uv$ et $y = vu$. Le **problème du décalage général** est alors le suivant :

Donnée : un alphabet Σ et un automate fini $\mathcal{A}$ sur l'alphabet Σ^2.
Question : est-ce qu'il existe un mot (x, y), avec x et y deux mots conjugués de Σ^*, dans le langage $\mathcal{L}(\mathcal{A})$ reconnu par $\mathcal{A}$?

5. Montrer que le problème du décalage général est indécidable par réduction du problème du décalage. *[Indication : montrer qu'il est décidable de savoir si un mot de la forme (x, x) est reconnu par un automate $\mathcal{A}$ sur Γ^2, sinon construire un automate reconnaissant le langage $\mathcal{L}(\mathcal{A}) \cap \{(sc^n, c^m t) \mid (s,t) \in \Gamma^*, n, m \geq 0\}$ et l'utiliser pour procéder à la réduction.]*

On considère une instance du problème du décalage, c'est-à-dire un alphabet Γ, une lettre $c \notin \Gamma$ et un automate $\mathcal{A}$ sur l'alphabet Σ^2, avec $\Sigma = \Gamma \cup \{c\}$. Tout d'abord, on peut tester l'acceptation par $\mathcal{A}$ d'un mot de la forme (x, x), en testant le vide de l'intersection de $\mathcal{A}$ avec un automate reconnaissant le langage régulier $\{(x, x) \mid x \in \Gamma^*\} = \{(a, a) \mid a \in \Gamma\}^*$. Si $\mathcal{A}$ n'accepte aucun mot de la forme $(x, x) \in (\Gamma^2)^*$, on construit l'automate $\mathcal{A}'$ reconnaissant $\mathcal{L}(\mathcal{A}) \cap \{(sc^n, c^m t) \mid (s,t) \in \Gamma^*, n, m \geq 0\}$: en effet, le langage $\{(sc^n, c^m t) \mid (s,t) \in \Gamma^*, n, m \geq 0\}$ est régulier puisque c'est l'intersection de $\{(sc^n, u) \mid s \in \Gamma^*, u \in \Sigma^*, n \geq 0\}$ avec $\{(u, c^m t) \mid t \in \Gamma^*, u \in \Sigma^*, m \geq 0\}$, tous deux réguliers.

On considère alors $\mathcal{A}'$ comme instance du problème du décalage général. S'il n'existe aucun mot (x, y), avec x et y deux mots conjugués de Σ^*, dans le langage $\mathcal{A}'$, alors la réponse au problème de décalage initial est également négative. Sinon, un certain mot $w = (uv, vu)$ est accepté par $\mathcal{A}'$. Montrons alors que w s'écrit $(zc^n, c^n z)$ avec $z \in \Gamma^*$ et $n \geq 1$, ce qui permet de répondre à la positive au problème de décalage, et donc de conclure la réduction.

Par construction, uv termine par c, vu commence par c et deux occurrences de c dans uv ou vu n'ont que des c entre elles. Ainsi, si u ou v est vide, alors $w = (c^n, c^n)$ pour $n \geq 1$, et on peut prendre $z = \varepsilon$. Sinon, ni u ni v ne sont vides. Ainsi, v commence et termine par c, de sorte que $v \in c^+$. Il s'ensuit que si u contient un c alors u commence et finit également par c et donc que $u \in c^+$, si bien que $w = (c^n, c^n)$ pour $n \geq 1$: on peut donc à nouveau prendre $z = \varepsilon$. Le dernier cas correspond à celui où u ne contient pas la lettre c, mais alors w s'écrit $(uc^n, c^n u)$ avec $u \in \Gamma^+$ et $n = |y|$: on peut alors choisir $z = u$.

Ceci conclut la réduction et prouve que le problème du décalage général est indécidable.

Pour finir, on s'intéresse à ce dernier problème :

Donnée : un alphabet Σ et un automate fini déterministe $\mathcal{A}$ sur l'alphabet Σ.
Question : est-ce qu'il existe deux mots u et v de Σ^* tels que uv et vu soient deux mots distincts du langage $\mathcal{L}(\mathcal{A})$ reconnu par $\mathcal{A}$?

6. On va montrer que ce problème est décidable. Au long de cette question, on pourra utiliser sans preuve le fait que si deux mots u et v vérifient $uv = vu$, alors il existe un mot non vide t tel que $u = t^i$ et $v = t^j$ (on peut même supposer que t est primitif, c'est-à-dire qu'il n'est la puissance d'aucun mot de Σ^*).

a) Commencer par montrer que si un automate fini déterministe $\mathcal{A}$ à n états acceptent deux mots distincts conjugués uv et vu, il accepte une telle paire de mots distincts uv et vu avec u ou v de longueur strictement inférieure à n^2.

Considérons une paire de mots (u, v) telle que $uv \in \mathcal{L}(\mathcal{A})$, $vu \in \mathcal{L}(\mathcal{A})$, $uv \neq vu$ et $|uv|$ est minimale parmi ces paires de mots. Supposons par l'absurde que $|u| \geq n^2$ et $|v| \geq n^2$. On considère le calcul acceptant de $\mathcal{A}$ d'étiquette uv : $q_0 \xrightarrow{u} q_1 \xrightarrow{v} p_1$ avec $q_1 \in Q$, q_0 unique état de I et $p_1 \in T$. De manière similaire, il existe un calcul $q_0 \xrightarrow{v} q_2 \xrightarrow{u} p_2$ avec $q_2 \in Q$, $p_2 \in T$, d'étiquette vu.
Dans l'automate $\mathcal{A}' = \langle Q \times Q, \Sigma, E', \{(q_0, q_1)\}, \{(q_2, p_1)\}\rangle$ avec $E' = \{((r, s), a, (r', s')) \mid a \in \Sigma, (r, a, r') \in E, (s, a, s') \in E\}$, le mot v est accepté. Comme $\mathcal{A}'$ a n^2 états et $|v| \geq n^2$, le calcul de v dans $\mathcal{A}'$ visite plus de $n^2 + 1$ états et on peut donc écrire $v = v_1v_2v_3$ avec $v_2 \neq \varepsilon$ et $\mathcal{A}'$ accepte v_1v_3. On en déduit donc les calculs $q_1 \xrightarrow{v_1v_3} p_1$ et $q_0 \xrightarrow{v_1v_3} q_2$ dans $\mathcal{A}$, ce qui implique que uv_1v_3 et v_1v_3u sont deux mots conjugués acceptés par $\mathcal{A}$. Par minimalité de $|uv|$, on en déduit que

$$uv_1v_3 = v_1v_3u \tag{8}$$

On peut répéter le même argument pour le mot u pour obtenir une décomposition $u = u_1u_2u_3$ avec $u_2 \neq \varepsilon$ et

$$vu_1u_3 = u_1u_3v \tag{9}$$

Finalement, les calculs de $\mathcal{A}$ qu'on a ainsi créés peuvent être découpés dans u et v simultanément, indiquant que $u_1u_3v_1v_3$ et $v_1v_3u_1u_3$ sont tous deux acceptés par $\mathcal{A}$ et donc que

$$u_1u_3v_1v_3 = v_1v_3u_1u_3$$

Grâce au résultat donné en préliminaire de cette question, il existe un mot t tel que $u_1u_3 = t^i$ et $v_1v_3 = t^j$, avec t primitif. Une fois remplacé dans (9), on a $vt^i = t^jv$, de sorte que $v = t^k$. Finalement, grâce à (8), on obtient $u = t^\ell$. On a alors $uv = vu$ ce qui contredit l'hypothèse.

b) En déduire que le problème précédent est décidable. *[Indication : on pourra commencer par démontrer que le langage $L_x = \{y \in \Sigma^* \mid xy \in L, yx \in L, xy \neq yx\}$ est un langage rationnel pour tout $x \in \Sigma^+$, si L est un langage rationnel.]*

On fixe $L = \mathcal{L}(\mathcal{A})$ le langage d'un automate déterministe sur Σ. On définit, pour tout mot $x \in \Sigma^+$, le langage $L_x = \{y \in \Sigma^* \mid xy \in L, yx \in L, xy \neq yx\}$. On vérifie que $L_x = L_1 \cap L_2 \cap L_3$ avec

$$\begin{aligned} L_1 &= \{y \in \Sigma^* \mid xy \in L\} \\ L_2 &= \{y \in \Sigma^* \mid yx \in L\} \\ L_3 &= \{y \in \Sigma^* \mid xy \neq xy\} \end{aligned}$$

L_1 et L_2 sont tous les deux rationnels. Pour L_3, grâce au préliminaire de la question, on sait que $xy = yx$ avec $x \neq \varepsilon$ implique que $y \in t^*$ où t est la racine primitive de x. Ainsi, $L_3 = \Sigma^* \setminus t^*$ est rationnel aussi.
Pour résoudre le problème initial, grâce à la question précédente, on sait qu'il suffit de vérifier l'existence d'une paire de mots (u, v) avec $|u| < n^2$ telle que $v \in L_u$. Ainsi, il suffit de tester le vide de tous les langages L_u avec $|u| < n^2$, ce qui est possible puisque L_u est rationnel.

Tiré de *Undecidability and Finite Automata* de Jörg Endrullis, Jeffrey Shallit et Tim Smith, arXiv :1702.01394v1 (2017).