DS2 (3 heures)

Lisez tout le texte avant de commencer. La plus grande importance sera attachée à la clarté, à la précision et à la concision de la rédaction. Toute réponse non justifiée ne sera pas prise en compte. Si vous repérez ce qui vous semble être une erreur d'énoncé, signalez la sur votre copie et poursuivez votre composition en expliquant les raisons de vos éventuelles initiatives. L'usage de tout dispositif électronique est interdit.

Le sujet comporte quatre exercices. Vous devez traiter les exercices 1 et 2 et, au choix, soit l'exercice 3, soit l'exercice 4. À titre indicatif, une estimation des durées de traitement et des niveaux de difficultés de chaque exercice est donnée ci-dessous.

- *Exercice 1 : 0,5 h (*)*
- *Exercice 2 : 1,5 h (**)*
- *Exercice 3 : 1,0 h (**)*
- *Exercice 4 : 1,0 h (***)*

Dans tout ce sujet, OCaml est le seul langage de programmation autorisé. Seules les fonctions incluses dans la bibliothèque standard du langage sont autorisées.

Exercice 1

Soit Σ un alphabet et $\mathcal{A}$ un automate défini par le quintuplet (Σ, Q, T, I, F), où :

- Q est *ensemble fini non vide des états* de $\mathcal{A}$;
- $T \subseteq Q \times \Sigma \times Q$ est l'*ensemble des transitions* ; étant donnée une transition $(p, x, q) \in T$, notée $p \xrightarrow{x} q$ également, on dit qu'elle va de l'état p à l'état q et qu'elle est d'étiquette x ;
- $I \subseteq Q$ est l'*ensemble des états initiaux* de $\mathcal{A}$;
- $F \subseteq Q$ est l'*ensemble des états acceptants* de $\mathcal{A}$.

$r_0 \xrightarrow{v}{}^*_{\mathcal{A}} r_n$ $r_0 \xrightarrow{a_1} r_1$

Un *calcul* c de $\mathcal{A}$ est une suite de la forme $p_0 \xrightarrow{x_1} p_1 \xrightarrow{x_2} p_2 \ldots \xrightarrow{x_k} p_k$ où, pour $1 \leqslant i \leqslant k$, $p_{i-1} \xrightarrow{x_i} p_i$ est une transition ; p_0 est l'origine du calcul, p_k son extrémité. L'étiquette de c est le mot $x_1 x_2 \ldots x_k$ formé par la suite des étiquettes des transitions successives. Un calcul d'origine p, d'extrémité q, d'étiquette m peut être noté $p \xrightarrow{m} q$. Il est dit *réussi* si $p \in I$ et $q \in F$. Un mot m de Σ^* est *reconnu* par $\mathcal{A}$ s'il est l'étiquette d'un calcul réussi. Le *langage reconnu* par $\mathcal{A}$, noté $\mathcal{L}(\mathcal{A})$, est l'ensemble des mots reconnus par $\mathcal{A}$. Un langage est dit *reconnaissable* s'il existe un automate qui le reconnaît.

Un état q d'un automate $\mathcal{A}$ est dit *accessible* s'il existe dans $\mathcal{A}$ au moins un calcul dont l'origine est un état initial et dont l'extrémité est q. L'automate $\mathcal{A}$ est dit *déterministe* si I contient exactement un élément et si, pour tout $p \in Q$ et tout $x \in \Sigma$, il existe au plus un état $q \in Q$ avec $(p, x, q) \in T$. L'automate $\mathcal{A}$ est dit *complet* si, pour tout $p \in Q$ et tout $x \in \Sigma$, il existe au moins un état $q \in Q$ avec $(p, x, q) \in T$.

Pour les exemples de ce problème, on utilisera l'alphabet $\Sigma_{ex} = \{a, b\}$.

Question 1. Soit $L_{1,ex}$ le langage sur Σ_{ex} des mots qui commencent par a. Représenter graphiquement un automate déterministe et complet noté $\mathcal{A}_{1,ex}$ qui reconnaît $L_{1,ex}$, c'est-à-dire vérifiant $\mathcal{L}(\mathcal{A}_{1,ex}) = L_{1,ex}$. Les états de cet automate sont nommés 1, 2, 3, l'état 1 étant l'unique état initial et l'état 2 l'unique état final.

Question 2. Soit $L_{2,ex}$ le langage sur Σ_{ex} des mots dont la longueur est multiple de 3. Représenter graphiquement un automate déterministe et complet noté $A_{2,ex}$ qui reconnaît $L_{2,ex}$, c'est-à-dire vérifiant $\mathcal{L}(A_{2_ex}) = L_{2,ex}$. Les états de cet automate sont nommés 4, 5, 6, l'état 4 étant l'unique état initial et l'unique état final.

Question 3. Les questions suivantes sont consacrées à une première méthode pour calculer un automate reconnaissant l'intersection de deux langages reconnaissables.

□ **3.1.** On considère deux langages L_1 et L_2 sur un alphabet quelconque Σ. On suppose que L_1 est reconnu par un automate $\mathcal{A}_1 = (\Sigma, Q_1, T_1, I_1, F_1)$; on suppose que L_2 est reconnu par un automate $\mathcal{A}_2 = (\Sigma, Q_2, T_2, I_2, F_2)$. Montrer que le langage $L = L_1 \cap L_2$ est reconnu par un automate $\mathcal{A} = (\Sigma, Q, T, I, F)$ pour lequel :

- $Q = Q_1 \times Q_2$
- $I = I_1 \times I_2$
- $F = F_1 \times F_2$.

En particulier, préciser l'ensemble T des transitions de $\mathcal{A}$.

□ **3.2.** En utilisant la construction de la question précédente, représenter graphiquement un automate noté $A_{3,ex}$ reconnaissant $L_{1,ex} \cap L_{2,ex}$. On ne représentera que les états accessibles de cet automate.

Question 4. La suite du problème propose d'appliquer une seconde façon de calculer un automate reconnaissant l'intersection de deux langages reconnaissables.

□ **4.1.** On considère un langage L sur un alphabet quelconque Σ. On suppose que L est reconnu par un automate déterministe et complet $\mathcal{A} = (\Sigma, Q, T, \{q_0\}, F)$, où q_0 est l'unique état initial de $\mathcal{A}$. Montrer que le langage complémentaire $\overline{L}$ de L, constitué des mots de Σ^* qui ne sont pas dans L, est reconnu par un automate $\mathcal{A}' = (\Sigma, Q, T', \{q_0\}, F')$ déterministe et complet. En justifiant votre réponse, préciser les ensembles T' et F'. En déduire un automate déterministe et complet $\mathcal{A}'_{1,ex}$ qui reconnaît $\overline{L}_1$ et un automate déterministe et complet $\mathcal{A}'_{2,ex}$ qui reconnaît $\overline{L}_2$. Ces deux automates seront décrits par leur représentation graphique.

□ **4.2.** Soient deux automates $\mathcal{A}_1 = (\Sigma, Q_1, T_1, I_1, F_1)$ et $\mathcal{A}_2 = (\Sigma, Q_2, T_2, I_2, F_2)$ reconnaissant respectivement des langages L_1 et L_2. On suppose que les ensembles Q_1 et Q_2 sont disjoints. Par définition, l'union de ces automates est l'automate $\mathcal{A} = (\Sigma, Q_1 \cup Q_2, T_1 \cup T_2, I_1 \cup I_2, F_1 \cup F_2)$. On admet que cet automate $\mathcal{A}$ reconnait le langage $L_1 \cup L_2$. On note $\mathcal{A}_{4,ex}$ l'automate obtenu par union de $\mathcal{A}_{1,ex}$ et $\mathcal{A}_{2,ex}$. Représenter graphiquement un automate déterministe et complet $\mathcal{A}_{5,ex}$ obtenu en appliquant à $\mathcal{A}_{4,ex}$ un algorithme classique de déterminisation. On ne représentera que les états accessibles de cet automate.

□ **4.3.** Déduire de l'automate $\mathcal{A}_{5,ex}$ un automate reconnaissant $L_{1,ex} \cap L_{2,ex}$.

Exercice 2

Un langage est régulier si et seulement s'il est accepté par un automate. Cependant, plusieurs automates peuvent accepter le même langage. Ce problème montre que tout langage reconnaissable L est reconnu par un unique automate, au renommage près des états, appelé *automate minimal*.

Dans ce problème, les automates sont déterministes et complets; on les appelle simplement automates. Un *automate déterministe* $\mathcal{A}$ est un quintuplet $(Q, \Sigma, q_0, F, \delta)$ avec :

- Q : ensemble fini non vide d'états;
- Σ : alphabet (ensemble fini non vide);
- q_0 : état initial;
- $F \subset Q$: ensemble d'états finaux;
- $\delta : Q \times \Sigma \to Q$: application de transition définie sur $Q \times \Sigma$ tout entier.

Σ^* est l'ensemble des mots construits sur Σ, ε est le mot vide, $|u|_a$ est le nombre d'occurrences de $a \in \Sigma$ dans u.

Soit $L \subset \Sigma^*$ un langage et $u \in \Sigma^*$. Le *résiduel à gauche de L par rapport à u* est le langage

$$u^{-1}L = \{v \in \Sigma^*,\ uv \in L\}$$

Question 1. Pour $L = \{ab, ba, aab\}$, donner le résiduel de L par rapport à a.

Question 2. On définit la relation $\sim_L$ sur Σ^*, dite *congruence de Nérode*, de la manière suivante.

$$\forall u, v \in \Sigma^* \qquad u \sim_L v \iff u^{-1}L = v^{-1}L$$

Montrer que $\sim_L$ est une relation d'équivalence et que :

$$\forall u, v, w \in \Sigma^* \qquad (u \sim_L v) \Rightarrow (uw \sim_L vw)$$

Question 3. Posons $L = \{u \in \Sigma^*,\ |u|_a = 0 \bmod 3\}$ le langage sur $\Sigma = \{a, b\}$ composé des mots de Σ^* ayant un nombre de a multiple de 3. Pour chacun des cas suivants, déterminer si les deux mots sont équivalents par $\sim_L$: b et ab, aba et bab, $abbaba$ et aaa.

Soit $\mathcal{A} = (Q, \Sigma, q_0, F, \delta)$ un automate. La fonction de transition δ^* étendue aux mots est définie récursivement par :

$$\begin{cases} \forall q \in Q & \delta^*(q, \varepsilon) = q \\ \forall a \in \Sigma,\ \forall m \in \Sigma^*,\ \forall q \in Q & \delta^*(q, m.a) = \delta(\delta^*(q, m), a) \end{cases}$$

Pour $q \in Q$ et $G \subset Q$, on note :

$$q^{-1}G = \{u \in \Sigma^*,\ \delta^*(q, u) \in G\}$$

$q^{-1}G$ est l'ensemble des mots correspondant à des chemins qui débutent en q et qui aboutissent dans un état de G. En particulier, $q^{-1}F$ est l'ensemble menant de q à un état terminal. On dit que $q^{-1}F$ est l'ensemble des mots *acceptés par l'automate depuis q*.

Une relation d'équivalence $\sim_{\mathcal{A}}$ est également définie sur Q par :

$$p \sim_{\mathcal{A}} q \iff p^{-1}F = q^{-1}F$$

L'automate $\mathcal{A}$ accepte le langage $L = q_0^{-1}F$. Si $q \in Q$ et $u \in \Sigma^*$ sont tels que $\delta^*(q_0, u) = q$ alors on peut montrer que $q^{-1}F = u^{-1}L$. Ainsi la congruence de Nérode peut définir un automate particulier appelé *automate minimal* de L.

Soit L un langage. L'*automate minimal* de L est défini par le quintuplet $\mathcal{A}_L = (Q_L, \Sigma, q_L, F_L, \delta_L)$ avec :

- $Q_L = \{u^{-1}L,\ u \in \Sigma^*\}$
- $q_L = \varepsilon^{-1}L = L$
- $F_L = \{u^{-1}L,\ u \in L\} = \{q \in Q_L,\ \varepsilon \in q\}$
- $\forall q \in Q_L,\ \forall a \in \Sigma,\ \delta_L(q,a) = a^{-1}q$

Question 4. Montrer que si L est un langage régulier alors Q_L est fini. Dans ce cas, l'automate minimal est un automate au sens de notre énoncé (ensemble fini d'états, caractère déterministe et complet).

Un automate $\mathcal{A} = (Q, \Sigma, q_0, F, \delta)$ est *accessible* si pour tout $q \in Q$, il existe $u \in \Sigma^*$ tel que $\delta^*(q_0, u) = q$. Un automate $\mathcal{A} = (Q, \Sigma, q_0, F, \delta)$ est *réduit* si pour tous $p, q \in Q$, $(p^{-1}F = q^{-1}F) \Rightarrow p = q$. $\mathcal{A}$ est donc réduit si les langages acceptés depuis deux états distincts sont distincts, ou encore si chaque classe d'équivalence pour la relation $\sim_{\mathcal{A}}$ sur Q est un singleton. Pour $L \subset \Sigma^*$ régulier, il est possible de montrer que l'automate minimal $\mathcal{A}_L$ de L est accessible et réduit. L'automate minimal du langage reconnu par un automate fini $\mathcal{A}$ peut être construit en exploitant cette propriété.

Soit $\mathcal{A} = (Q, \Sigma, q_0, F, \delta)$ un automate fini acceptant le langage $L \subset \Sigma^*$. Trouver l'automate minimal $\mathcal{A}_L$ de L revient à trouver un automate fini accessible et réduit équivalent à $\mathcal{A}$, c'est-à-dire reconnaissant le même langage.

Pour touver un automate accessible, il suffit par exemple de visiter les états qui peuvent être atteints par $\mathcal{A}$ depuis q_0 et d'éliminer les autres états. Il reste donc à trouver une méthode pour rendre $\mathcal{A}$ réduit. Par définition de la relation $\sim_{\mathcal{A}}$ sur Q, $\mathcal{A}$ est réduit si pour tout couple $(p,q) \in Q^2$ avec $p \neq q$, $p \not\sim_{\mathcal{A}} q$. Notons que $p \not\sim_{\mathcal{A}} q$ a lieu s'il existe $u \in \Sigma^*$ tel que $\delta^*(p,u) \in F$ et $\delta^*(q,u) \notin F$ ou $\delta^*(q,v) \in F$ et $\delta^*(p,u) \notin F$. On dit alors que u *distingue* p et q ou que le couple (u,v) est *distingué* par u.

L'algorithme qui suit est un algorithme de réduction du langage associé à $\mathcal{A}$ utilisant ces notions. Dans la suite, N_k désigne l'ensemble des couples d'états de Q qui sont distingués par un mot de longueur k et qui ne sont distingués par aucun autre mot strictement plus court.

Algorithme : recherche des états équivalents

Entrée : un automate déterministe $\mathcal{A} = (Q, \Sigma, q_0, F, \delta)$
Sortie : les ensembles d'états équivalents
$k \leftarrow 0$
*********Initialisation************
$N_0 \leftarrow \varnothing$
pour tous $p \in F$ et $q \in Q \setminus F$, **faire**
  La paire (p,q) est distinguée
  $N_0 \leftarrow N_0 \cup \{(p,q)\}$.
Tant que $N_k \neq \varnothing$, **faire**
  ****Construction de N_{k+1}****
  $N_{k+1} \leftarrow \varnothing$
  pour chaque paire $(p,q) \in N_k$, **faire**
    pour chaque $a \in \Sigma$, **faire**
      pour chaque $(r,s) \in Q^2$ tel que $\delta(r,a) = p$ et $\delta(s,a) = q$, **faire**
        si $(r,s) \notin \bigcup_{i=0}^{k} N_i$, **alors**
          $N_{k+1} \leftarrow N_{k+1} \cup \{(r,s),(s,r)\}$
  $k \leftarrow k+1$

Question 5. Dans la phase d'initialisation, pourquoi la paire (p,q) est-elle distinguée ?

Question 6. Pourquoi, si $N_i = \varnothing$, alors $\forall j > i,\ N_j = \varnothing$?

Soit alors $\mathcal{A} = (Q, \Sigma, q_0, F, \delta)$ l'automate fini défini ci-dessous.

- $Q = \{1,2,3,4,5,6\}$
- $\Sigma = \{a,b\}$
- $q_0 = \{4\}$
- $F = \{2,5\}$
-

δ	1	2	3	4	5	6
a	2	4	6	5	1	6
b	1	3	2	1	6	2

Question 7. Représenter graphiquement l'automate $\mathcal{A}$.

Question 8. Appliquer l'algorithme pour trouver l'ensemble des états équivalents. Pour chaque itération k, la trace de l'algorithme sera donnée par une matrice T de taille $|Q| \times |Q|$, avec $T(i,j)$ valant la longueur d'un chemin, s'il existe, qui distingue le couple (i,j) ($T(i,j)$ vide sinon). En déduire les classes d'équivalence des états de $\mathcal{A}$.

Question 9. Un théorème, non détaillé ici, permet alors de projeter $\mathcal{A}$ sur $\mathcal{A}_L$ et de préciser états et transitions de cet automate minimal. Il permet en particulier de définir les états de $\mathcal{A}_L$ comme étant les classes d'équivalence issues de l'algorithme précédent. Il permet également d'affirmer que si un état de $\mathcal{A}_L$ correspond à une classe d'équivalence $[q]_A$ pour la relation $\sim_{\mathcal{A}}$, alors la lecture d'un symbole $a \in \Sigma$ depuis cet état dans $\mathcal{A}_L$ conduit à l'état correspondant à la classe $[\delta(q,a)]_A$. Représenter graphiquement l'automate minimal de la question précédente, avec ses états et ses transitions.

Exercice 3

Un langage défini sur un alphabet $\Sigma = \{a\}$ est dit *unaire*. Un automate reconnaissant un tel langage est également dit *unaire*. Lorsqu'on représente un tel automate, il n'est pas utile de faire figurer les étiquettes des transitions, toutes ces étiquettes étant l'a. Cette convention est adoptée dans cet énoncé.
Dans un automate unaire, un *chemin* d'un état q_1 à l'état q_p est une suite $q_1, \dots, q_p$ d'états telle que, pour i compris entre 2 et p, il existe une transition de q_{i-1} vers q_i. Si en outre il existe une transition de q_p vers q_1, le chemin est appelé un *circuit*.
Dans cet exercice, tous les automates considérés sont finis et ont un seul état initial. Un automate est *émondé* si, pour tout état q, il existe d'une part un chemin de l'état initial à q et d'autre part un chemin de q à un état acceptant.
On rappelle qu'un langage non vide est *régulier* si et seulement s'il est reconnu par un automate ou encore si et seulement s'il est reconnu par un automate déterministe émondé.
Soient α et β deux entiers positifs ou nuls. On note $L(\alpha, \beta)$ le langage unaire défini par :

$$L(\alpha, \beta) = \{a^{\alpha k+\beta} \mid k \in \mathbb{N}\}$$

Question 1.
□ **1.1.** Sans justification, donner une condition nécessaire et suffisante pour que $L(\alpha, \beta)$ soit fini.
□ **1.2.** Dans le cas où cette condition est satisfaite, donner sans justification le cardinal de $L(\alpha, \beta)$.

Question 2. On considère l'automate $\mathcal{A}_1$ ci-dessous. Sans justification, trouver deux entiers α_1, β_1 tels que $\mathcal{A}_1$ reconnaisse le langage $L(\alpha_1, \beta_1)$.

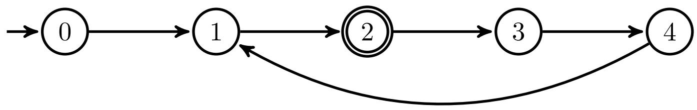

Figure 1 – Automate $\mathcal{A}_1$

Question 3. On considère l'automate $\mathcal{A}_2$ ci-dessous. On note L_2 le langage reconnu par $\mathcal{A}_2$. Indiquer sans justification quatre entiers $\alpha_2, \beta_2, \alpha_3, \beta_3$ tels que $\mathcal{A}_2$ reconnaisse le langage $L_2 = L(\alpha_2, \beta_2) \cup L(\alpha_3, \beta_3)$.

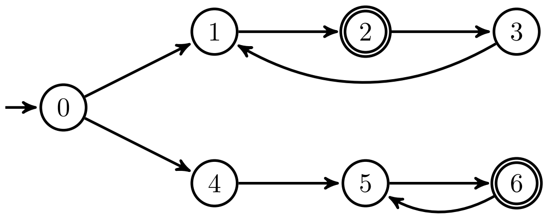

Figure 2 – Automate $\mathcal{A}_2$

Question 4. Construire un automate *déterministe émondé* $\mathcal{A}_3$ en déterminisant l'automate $\mathcal{A}_2$.

Question 5. À l'aide de l'automate $\mathcal{A}_3$, indiquer sans justification cinq entiers α_4, β_4,β_5, β_6, β_7 tels que $\mathcal{A}_3$ reconnaisse le langage $L_3 = L(\alpha_4, \beta_4) \cup L(\alpha_4, \beta_5) \cup L(\alpha_4, \beta_6) \cup L(\alpha_4, \beta_7)$

Un automate est dit *de la forme F* si, en omettant les états acceptants, il peut se tracer selon le schéma ci-dessous.

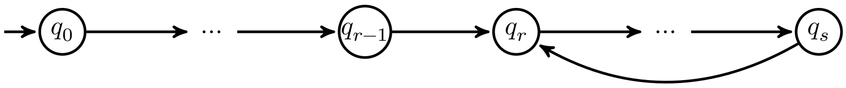

Figure 3 – Automate de la forme F

Le chemin $q_0, \dots, q_{r-1}$ peut être vide, auquel cas on a $r = 0$. Le circuit $q_r, \dots, q_s$ ne doit pas être vide mais on peut avoir $r = s$ avec une transition de l'état q_r vers lui même (*boucle*). On constate que les automates $\mathcal{A}_1$ et $\mathcal{A}_3$ sont de la forme F, mais non $\mathcal{A}_2$.

Question 6. Sans justification, dessiner un automate de la forme F qui reconnaît le langage $L(1, 2)$. On fera figurer le ou les états acceptants.

Question 7. Sans justification, dessiner un automate de la forme F qui reconnaît le langage $L(2,3) \cup L(5,2)$. On fera figurer le ou les état(s) acceptant(s).

Question 8. En s'inspirant de la réponse à la question précédente, décrire sans justification un automate de la forme F qui reconnaît le langage $L(2,3) \cap L(5,2)$. Indiquer deux entiers α et β tels que $L(2,3) \cap L(5,2) = L(\alpha, \beta)$.

Question 9. Montrer qu'un automate déterministe émondé qui reconnaît un langage unaire régulier infini est de la forme F. Donner une condition nécessaire et suffisante portant sur les états acceptants pour qu'un automate de la forme F reconnaisse un langage infini.

Question 10. Soit L un langage régulier unaire infini. Montrer qu'il existe deux entiers $\alpha \geqslant 1$ et $\beta \geqslant 0$ tels que L contient $L(\alpha, \beta)$.

Question 11. On considère une suite $(u_n)_{n\geqslant 0}$ de nombres entiers positifs ou nuls. On suppose que la suite $(u_{n+1} - u_n)_{n\geqslant 0}$ est positive et strictement croissante. Soit L le langage défini par

$$L = \{a^{u_n} /\ n \geqslant 0\}$$

En utilisant la question précédente, montrer que L n'est pas régulier.

Question 12. Montrer que le langage L défini par $L = \{a^{n^2} /\ n \geqslant 0\}$ n'est pas régulier.

Exercice 4

Soit Σ un alphabet et deux mots $w, w' \in \Sigma^*$. On dit que w' est un *sur-mot* de w, noté $w \preccurlyeq w'$, s'il existe une fonction strictement croissante ϕ de $\{1, \dots, |w|\}$ dans $\{1, \dots, |w'|\}$ telle que pour tout $1 \leqslant i \leqslant |w|$, $w_i = w'_{\phi(i)}$ où $|w|$ dénote la longueur de w et w_i dénote la i-ème lettre de w. L étant un langage, on note $\overline{L}$ le langage des sur-mots de mots de L.

$$\overline{L} = \{w' \in \Sigma^* \mid \exists w \in L, w \preccurlyeq w'\}$$

Question 1. On pose L_0 le langage défini par l'expression régulière ab^*a et L_1 le langage défini par l'expression régulière $(ab)^*$. Donner une expression régulière pour $\overline{L_0}$ et pour $\overline{L_1}$.

Question 2. Montrer que, pour tout langage L, on a $\overline{\overline{L}} = \overline{L}$.

Question 3. Existe-t-il des langages L' pour lesquels il n'existe aucun langage L tel que $\overline{L} = L'$?

Question 4. Montrer que, pour tout langage régulier L, le langage $\overline{L}$ est également régulier.

Question 5. Pour cette question, on admet le résultat suivant : pour toute suite $(w_n)_{n\in\mathbb{N}}$ de mots de Σ^*, il existe $i < j$ tels que $w_i \preccurlyeq w_j$. Montrer que, pour tout langage L (non nécessairement régulier), il existe un langage fini $F \subseteq L$ tel que $\overline{F} = \overline{L}$.

Question 6. Un langage L est *clos par sur-mots* si, pour tout $u \in L$ et $v \in \Sigma^*$ tel que $u \preccurlyeq v$, on a $v \in L$. Déduire de la question précédente que tout langage clos par sur-mots est régulier.