I Langage local, linéaire et automate de Glushkov

1. Construire l'automate de Glushkov reconnaissant $(ab|b)^*b$.
2. Montrer que si L_1 et L_2 sont locaux alors $L_1 \cap L_2$ est local (même si les alphabets ne sont pas disjoints, contrairement à l'union et concaténation vue en cours).
3. Montrer qu'il existe un nombre fini de langages locaux sur un alphabet fixé.
4. Soient $\mathcal{L}_{\text{reg}}$, $\mathcal{L}_{\text{loc}}$, $\mathcal{L}_{\text{lin}}$ l'ensemble des langages réguliers, locaux et linéaires (c'est à dire décrits par une expression régulière linéaire). Quelles sont les inclusions entre ces 3 ensembles ? Ces inclusions sont-elles strictes ?

II Exercice CCP (élimination des états)

Dans cet exercice, on utilise une flèche sortante pour indiquer un état final.

1. Rappeler la définition d'un langage régulier.
2. Les langages suivants sont-ils réguliers ? Justifier.

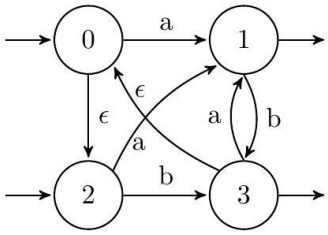
(a) $L_1 = \{a^n b a^m \mid n, m \in \mathbb{N}\}$

(b) $L_2 = \{a^n b a^m \mid n, m \in \mathbb{N}, n \leq m\}$

(c) $L_3 = \{a^n b a^m \mid n, m \in \mathbb{N}, n > m\}$

(d) $L_4 = \{a^n b a^m \mid n, m \in \mathbb{N}, n + m \equiv 0 \bmod 2\}$

3. On considère l'automate non déterministe suivant :



(a) Déterminiser cet automate.

(b) Construire une expression régulière dénotant le langage reconnu par cet automate.

(c) Décrire simplement avec des mots le langage reconnu par cet automate.

III Stabilité des langages réguliers

Pour chacune des propositions suivantes, expliquer pourquoi elle est vraie ou donner un contre-exemple :

1. Si L est régulier et $L \subseteq L'$ alors L' est régulier.
2. Si L' est régulier et $L \subseteq L'$ alors L est régulier.
3. Si L est régulier alors L^* est régulier.
4. Si L^* est régulier alors L est régulier.
5. Si L est régulier sur un alphabet Σ alors $\Sigma^* \setminus L$ (complémentaire de L) est régulier.
6. Une union finie de langages réguliers est régulier (L_1, ..., L_n réguliers $\implies \bigcup_{k=1}^{n} L_k$ régulier).
7. Une union dénombrable de langages réguliers est régulier (L_1, ..., L_n, ... réguliers $\implies \bigcup_{k=1}^{\infty} L_k$ régulier).
8. Une intersection finie de langages réguliers est régulier (L_1, ..., L_n réguliers $\implies \bigcap_{k=1}^{n} L_k$ régulier).
9. Une intersection dénombrable de langages réguliers est régulier (L_1, ..., L_n, ... réguliers $\implies \bigcap_{k=1}^{\infty} L_k$ régulier).

IV Reconnaissable $\Longrightarrow$ régulier par programmation dynamique ($\approx$ Floyd-Warshall)

Cet exercice est une alternative à la méthode d'élimination des états pour obtenir une expression régulière à partir d'un automate.
Soit $(\Sigma, Q, 0, F, \delta)$ un automate déterministe tel que $Q = \{0, ..., n-1\}$.
Soit $L(i, j, k)$ le langage des étiquettes des chemins de i à j n'utilisant que des états intermédiaires strictement inférieurs à k.

1. Montrer que $L(i, j, 0)$ est régulier, pour tous les états i, j.
2. Donner une équation de récurrence sur $L(i, j, k)$, en la démontrant.
3. En déduire, par récurrence, que tout langage reconnaissable est régulier.
4. Écrire le pseudo-code d'un algorithme ayant pour entrée un automate déterministe et renvoyant une expression régulière dénotant le langage reconnu par cet automate. Préciser la complexité.

V Résiduels et minimisation d'automate

Si $u \in \Sigma^\star$ et L un langage sur Σ, on appelle résiduel de L par rapport à u le langage $u^{-1}L$ tel que :

$$u^{-1}L = \{m \in \Sigma^\star \mid um \in L\}$$

Intuitivement le résiduel de L par rapport à u est l'ensemble des mots avec lesquels on peut compléter u pour obtenir un mot de L. L'ensemble des résiduels de L est noté $\mathcal{R}_L = \{u^{-1}L \mid u \in \Sigma^\star\}$.

1. Si L est un langage, quel est le résiduel de L par rapport à ε ?
2. Montrer que pour tout langage L sur Σ et tous mots $u, v \in \Sigma^\star$, $u^{-1}(v^{-1}L) = (vu)^{-1}L$.
3. a) Soit $L_1 = (a|b)^\star ab(a|b)^\star$. Déterminer $(bab)^{-1}L_1$ et $(aaaba)^{-1}L_1$. Que constate-t-on ?
   b) Expliciter $\mathcal{R}_{L_1}$.

On veut montrer qu'un langage L est reconnaissable si et seulement s'il admet un nombre fini de résiduels.

4. Soit L un langage reconnaissable par un automate $A = (\Sigma, Q, q_0, F, \delta)$ qu'on peut supposer déterministe, complet et dont tous les états sont accessibles sans perte de généralité.
   a) Montrer que la fonction $\varphi : \begin{cases} Q \longrightarrow \mathcal{R}_L \\ q \longmapsto u^{-1}L \text{ où } u \text{ est l'un des mots tels que } \delta^\star(q_0, u) = q \end{cases}$ est correctement définie.
   b) Montrer que L admet un nombre fini de résiduels.
5. Soit à présent L un langage ayant un nombre fini de résiduels. On définit alors un automate $M(L) = (\Sigma, Q, I, F, \delta)$, appelé automate des résiduels associé à L, où :
   - $Q = \mathcal{R}_L$.
   - $I = \{\varepsilon^{-1}L\}$.
   - $F = \{u^{-1}L \mid u \in L\}$.
   - pour toute lettre $a \in \Sigma$ et tout résiduel $u^{-1}L \in Q$, $\delta(u^{-1}L, a) = (ua)^{-1}L$.

   a) Montrer que la fonction δ est correctement définie, c'est-à-dire que l'image donnée par δ d'un résiduel $u^{-1}L$ ne dépend que du résiduel et pas de u.
   b) Montrer que $M(L)$ est un automate déterministe et complet.
   c) Montrer que $M(L)$ reconnaît le langage L.
6. A l'aide des questions précédentes, déterminer l'automate des résiduels associé au langage L_1.

Un automate reconnaissant un langage L est dit minimal s'il est déterministe, complet et possède le plus petit nombre d'états possible parmi les automates déterministes et complets reconnaissant L.

7. Montrer que l'automate des résiduels est un automate minimal.

On peut montrer que cet automate minimal est en fait unique (à renommage des états près) ce qui fournit une façon de déterminer si deux automates reconnaissent le même langage : il suffit de les minimiser et comparer les deux automates obtenus (une autre méthode consistant à tester si $L_1 \Delta L_2 = \emptyset$: cf TP 2).
Il est possible de déterminer l'automate minimal équivalent à un automate donné grâce à l'algorithme de Moore.