

I – Langage local, linéaire et automate de Glushkov

1. Construire l'automate de Glushkov reconnaissant $(ab \mid b)^*b$.
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 à la concaténation vues en cours).
3. Montrer qu'il existe un nombre fini de langages locaux sur un alphabet fixé.
4. Soient $\mathcal{L}_{\text{reg}}$, $\mathcal{L}_{\text{loc}}$, $\mathcal{L}_{\text{lin}}$ les ensembles des langages réguliers, locaux et linéaires (c'est-à-dire décrits par une expression régulière linéaire). Quelles sont les inclusions entre ces trois ensembles ? Ces inclusions sont-elles strictes ?

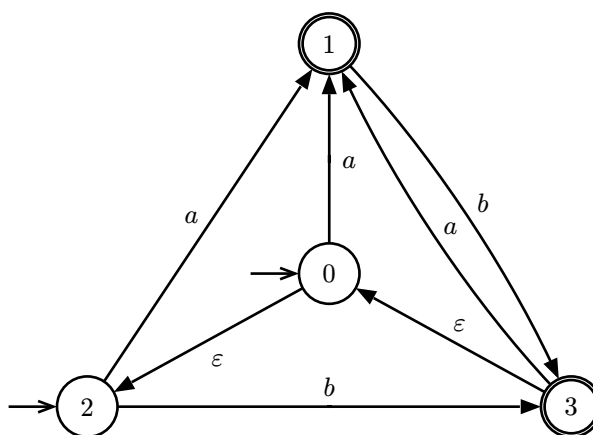
II – Sujet 0 CCINP

CCINP 2022 MPI oral

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

- (a) $L_1 = \{a^n b a^m \mid n, m \in \mathbb{N}\}$ (c) $L_3 = \{a^n b a^m \mid n, m \in \mathbb{N}, n > m\}$
 (b) $L_2 = \{a^n b a^m \mid n, m \in \mathbb{N}, n \leq m\}$ (d) $L_4 = \{a^n b a^m \mid n, m \in \mathbb{N}, n + m \equiv 0 \pmod{2}\}$

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 $L \subset L'$ alors L' est régulier.
2. Si L' est régulier et $L \subset L'$ 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 régulière ($L_1, \dots, L_n$ réguliers $\Rightarrow \bigcup_{k=1}^n L_k$ régulier).
7. Une union dénombrable de langages réguliers est régulière ($L_1, \dots, L_n, \dots$ réguliers $\Rightarrow \bigcup_{k=1}^\infty L_k$ régulier).
8. Une intersection finie de langages réguliers est régulière ($L_1, \dots, L_n$ réguliers $\Rightarrow \bigcap_{k=1}^n L_k$ régulier).
9. Une intersection dénombrable de langages réguliers est régulière ($L_1, \dots, L_n, \dots$ réguliers $\Rightarrow \bigcap_{k=1}^\infty L_k$ régulier).

IV – Algorithme de McNaughton-Yamada

On présente une alternative à la méthode d'élimination des états pour convertir un automate en expression régulière, avec un algorithme par programmation dynamique similaire à Floyd-Warshall.

Soit $A = (\Sigma, Q, 0, F, \delta)$ un automate déterministe tel que $Q = \{0, \dots, n-1\}$. Pour $i, j \in Q$ et $k \in \{0, \dots, n\}$, on note $L(i, j, k)$ le langage des étiquettes des chemins de i à j dont tous les états intermédiaires – sauf les extrémités – sont strictement inférieurs à k .

1. Montrer que $L(i, j, 0)$ est régulier, pour tout $i, j \in Q$.
2. Donner une équation de récurrence sur $L(i, j, k)$.
3. En déduire, par récurrence, que tout langage reconnaissable est régulier.

4. Implémenter cet algorithme en `to_regex` : 'a automate -> 'a regexp. Préciser la complexité, en temps et mémoire.

```

type 'a regexp =
  | Vide | Epsilon | L of 'a
  | Union of 'a regexp * 'a regexp
  | Concat of 'a regexp * 'a regexp
  | Etoile of 'a regexp

```

```

type 'a automate = {
  n : int;
  finaux : int list;
  transitions : (int * 'a * int) list;
}

```

V – Résiduels et théorème de Myhill-Nerode

Soit Σ un alphabet fini et $L \subset \Sigma^*$. Un *ADC* est un automate déterministe complet, noté $A = (Q, \Sigma, q_0, F, \delta)$. On rappelle qu'un langage est régulier si et seulement s'il est reconnu par un ADC. Pour les exemples, on fixe $\Sigma = \{a, b\}$ et $H = \mathcal{L}((a \mid b)^* ab (a \mid b)^*)$.

Un mot w est un *suffixe distinguant* pour u et v si exactement l'un des mots uw et vw appartient à L . Un ensemble $D \subset \Sigma^*$ est *distinguant* pour L si deux mots distincts de D ont toujours un suffixe distinguant.

II – Résiduels et théorème de Myhill-Nerode

Le *résiduel* de L par un mot u est le langage $u^{-1}L = \{w \in \Sigma^* \mid uw \in L\}$ des continuations acceptées après u . On note $\mathcal{R}_L = \{u^{-1}L \mid u \in \Sigma^*\}$ l'ensemble des résiduels de L .

1. Calculer $\varepsilon^{-1}L$ et exprimer $v^{-1}(u^{-1}L)$ comme un résiduel de L . Déterminer tous les résiduels de H , en distinguant les mots contenant déjà ab , ceux qui se terminent par a , et les autres.
2. On définit $u \equiv_L v$ par $u^{-1}L = v^{-1}L$. Montrer que $\equiv_L$ est une relation d'équivalence, appelée *congruence de Nerode*, stable par concaténation à droite :

$$u \equiv_L v \implies \forall t \in \Sigma^*, ut \equiv_L vt.$$

Relier cette relation aux suffixes distinguants. En déduire une bijection entre ses classes et les résiduels, puis caractériser les ensembles distinguants à l'aide de ces classes.

3. On suppose $\mathcal{R}_L$ fini. Construire un ADC M_L d'ensemble d'états $\mathcal{R}_L$ tel que la lecture de tout mot m depuis l'état initial conduise à $m^{-1}L$. Préciser l'état initial, les états finaux et les transitions, puis montrer que M_L est accessible et reconnaît L .
4. Dessiner M_H et donner un ensemble distinguant certifiant qu'aucun ADC reconnaissant H n'a moins d'états.
5. Soit $A = (Q, \Sigma, q_0, F, \delta)$ un ADC accessible reconnaissant L . Montrer que $\varphi : \delta^*(q_0, m) \mapsto m^{-1}L$ est bien définie et surjective. Vérifier qu'elle préserve l'état initial, les transitions et l'appartenance aux états finaux.

Un ADC reconnaissant L est *minimal* s'il a le moins d'états possible parmi les ADC reconnaissant L . En déduire la minimalité de M_L , puis l'unicité de l'ADC minimal à renommage des états près.

6. Établir le *théorème de Myhill-Nerode* : les propriétés suivantes sont équivalentes.
 - L est régulier
 - L possède un nombre fini de résiduels
 - la congruence de Nerode a un nombre fini de classes

Pour L régulier, relier le nombre d'états de l'ADC minimal à la taille maximale d'un ensemble distinguant. Montrer enfin que tout langage non régulier possède un ensemble distinguant infini.