

AUTOMATES ET LANGAGES RATIONNELS

Lycée Louis-le-Grand

Antoine Goudey

Mars 2026

1 Définitions fondamentales

Définition (Alphabet). Un **alphabet** Σ est un ensemble fini de symboles.

Définition (Mot). Un **mot** est une suite finie d'éléments de Σ . Le mot vide est noté ϵ .

Définition (Langage). Un **langage** L est un sous-ensemble de Σ^* .

Proposition. On peut faire les opérations suivantes sur les langages :

- Union : $L_1 + L_2$
- Concaténation : $L_1 \cdot L_2$
- Étoile de Kleene $L^* = \bigcup_{n \geq 0} L^n$
- Intersection : $L_1 \cap L_2$

2 Langages locaux

Définition (Langage local). Un langage est dit **local** s'il existe $P, S \subset \Sigma$ et $N \in \Sigma^2$ tels que $L \setminus \{\epsilon\} = (P\Sigma^* \cap \Sigma^*S) \setminus (\Sigma^*N\Sigma^*)$

Proposition. On a les résultats suivants :

- L'ensemble des langages locaux est stable par **intersection** et par passage à l'**étoile de Kleene**.
- Si L_1 et L_2 sont des langages locaux sur des alphabets **disjoints**, $L_1 + L_2$ et $L_1 \cdot L_2$ sont des langages locaux.

3 Expressions régulières

Définition (Expression régulière). L'ensemble des expressions régulières sur Σ est un langage sur l'alphabet $\Sigma \cup \{\emptyset, \epsilon, *, |, \cdot, (,)\}$ construit par induction.

Définition (Langage régulier). Une expression régulière peut être interprétée en un langage. Un langage est dit **régulier** s'il existe une expression régulière dont l'interprétation est ce langage.

Définition (Expression linéaire). Une expression régulière est dite **linéaire** si chaque symbole $s \in \Sigma$ apparaît au plus une fois.

Proposition. L'interprétation de toute expression régulière linéaire est un langage local. (Attention, réciproque fausse : aa^*)

4 Automates Finis Déterministes

Définition (Automate Fini Déterministe). Sur Σ , un **automate fini déterministe** est un quadruplet (Q, q_0, F, δ) avec $\delta : K \rightarrow Q$ une fonction de transition ($K \subset Q \times \Sigma$). Un mot w est reconnu s'il existe un chemin de q_0 jusqu'à un état $q \in F$.

Définition (Blocage). Un couple $(q, s) \in Q \times \Sigma$ est un **blocage** si la fonction de transition n'est pas définie pour ce couple. Un automate sans blocage est **complet**.

Définition (Automate standard). Un automate est dit **standard** s'il n'existe pas $(q, s) \in Q \times \Sigma$ tels que $\delta(q, s) = q_0$.

Définition (Automate local). Un automate **local** est tel que pour tout $s \in \Sigma$, il existe $q_s \in Q$ tel que si $q \in Q$, $\delta(q, s) = q_s$, ou (q, s) est un blocage.

5 Reconnaissance d'un langage

Définition. Un langage est **reconnaissable** s'il existe un automate dont l'ensemble des mots reconnus est exactement ce langage.

Lemme (d'Arden). Soit A et B deux langages sur Σ . L'équation $L = AL + B$ d'inconnue le langage L admet pour solution $L = A^*B$. De plus, cette solution est unique si $\epsilon \notin A$.

Proposition. Tout langage local est reconnaissable par un automate local.

Proposition. L'ensemble des langages reconnaissables est stable par :

- Complémentation (on inverse états finaux et états non finaux)
- Concaténation (On "colle" le début du second automate à la fin du premier)
- Étoile de Kleene (On construit un AFN avec des transitions des états finaux vers les états voisins de q_0)
- Intersection (On construit un automate "produit" sur $Q \times Q'$)
- Union (On détermine l'ensemble composé des deux automates, cela revient à faire un "produit")

6 Opérations sur les automates

Définition (État accessible). Un état est accessible s'il existe un chemin de l'état initial jusqu'à cet état.

Définition (État co-accessible). Un état est co-accessible s'il existe un chemin de cet état à un état final.

Définition (Émondage). L'**émondage** d'un automate c'est la suppression des états non accessibles ou non co-accessibles.

Proposition. Pour tout automate fini déterministe, on peut trouver un automate fini déterministe **complet** équivalent (en ajoutant un état poubelle) et un automate fini **standard** équivalent (en ajoutant un nouvel état initial "clone").

7 Automates finis non-déterministes

Définition (Automate fini non-déterministe). Pour un automate fini **non-déterministe**, la fonction de transition est $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$. L'automate accepte un mot s'il existe *au moins un* chemin valide menant à un état final.

Proposition. On peut **déterminiser** un automate : Pour tout automate fini non déterministe, il existe un automate fini déterministe reconnaissant le même langage.

8 Théorème de Kleene

Théorème (Kleene). Un langage L sur un alphabet Σ est régulier si et seulement si il est reconnaissable par un automate fini.

Démonstration. Pour le sens direct, on utilise les propriétés de stabilité et on construit des automates qui reconnaissent chaque symbole. On fait une induction structurelle. On peut aussi utiliser l'algorithme suivant. $\square$

Algorithme (Berry-Sethi). Permet de construire un automate à partir d'une expression régulière. Il faut :

- Linéariser l'expression en étiquetant les symboles
- Calculer les ensembles P, S, F du langage local qu'est l'interprétation de cette expression régulière numérotée
- Construire l'automate associé au langage local
- Supprimer les étiquettes

9 Automates asynchrones

Définition (Automates asynchrones). On généralise les automates finis non-déterministes. On ajoute des ϵ -transitions (ou transitions spontanées) qui permettent de sauter instantanément aux états suivants.

Remarque. Les ϵ -transitions permettent facilement de créer un automate qui reconnaît L^* ou $L_1 \cdot L_2$.

Proposition. Il est possible de transformer un automate asynchrone en un automate équivalent sans ϵ -transition (fermeture avant) :

- Pour chaque transition de q vers q' étiquetée par s , pour chaque ϵ -chemin de q' vers q'' , on ajoute une transition de q vers q'' étiquetée par s .
- Pour chaque ϵ -chemin d'un état initial q vers q' , on ajoute q' à l'ensemble des états initiaux.
- On supprime toutes les ϵ -transitions.

10 Des automates aux expressions régulières

C'est le principe de la démonstration de la réciproque du théorème de Kleene.

Algorithme. On suit ces étapes :

- On crée un seul état initial en ajoutant un nouvel état et des ϵ -transitions vers tous les états initiaux
- Idem pour les états finaux
- On répète ensuite les étapes suivantes dans un ordre quelconque :
 - Si $q \xrightarrow{f} q' \xrightarrow{g} q''$, et si ce sont les seules transitions de q' , on transforme cela en $q \xrightarrow{fg} q''$
 - Si $q \xrightarrow[f]{f} q'$, on transforme cela en $q \xrightarrow{f|g} q'$
 - Si $q \xrightarrow[e_1]{e_2} q \xrightarrow[g_1]{g_2}$ et $q \xrightarrow{f} q$, on transforme cela en $q \xrightarrow[e_1 f^*]{e_2 f^*} q \xrightarrow[g_1]{g_2}$
 - Si $q \xrightarrow[f]{e} q \xrightarrow{g} q'$, on transforme cela en $q \xrightarrow{e} q \xrightarrow{g}$ et $q \xrightarrow{f} q \xrightarrow{g}$

11 Lemme de l'étoile

Lemme. Si L est un langage rationnel, il existe un entier $N \geq 1$ tel que pour tout mot $w \in L$ vérifiant $|w| \geq N$, il existe une décomposition $w = xyz$ satisfaisant :

1. $|xy| \leq N$
2. $y \neq \epsilon$ ($|y| \geq 1$)

3. $\forall k \in \mathbb{N}, xy^kz \in L$.

Démonstration. Idée de la preuve : Un langage régulier est reconnaissable, on construit un automate, et si $N = |Q|$, on peut trouver une boucle dans un chemin acceptant. $\square$

Le lemme s'utilise surtout par contraposée pour prouver qu'un langage **n'est pas** rationnel. (Par exemple : $\{a^n b^n \mid n \in \mathbb{N}\}$).