ROYAUME DU MAROC
UNIVERSITE ABDELMALEK ESSAADI
Ecole Nationale des Sciences Appliquées

Tanger

Année universitaire 2019-2020

Cours : Théorie des Automates / Chapitre III

Langages réguliers & Automates

Version 2.0

Introduction

Dans ce chapitre, qui fait le lien entre les deux derniers chapitres, nous allons montrer que l'ensemble des langages réguliers coïncide exactement avec l'ensemble des langages acceptés par automate fini.

I. Des expressions aux Automates :

a. Définition :

A toute expression régulière φ, on peut associer un automate fini A de telle sorte que :

$$\mathcal{L}(\varphi)=L(A)$$

On procède par **récurrence** sur la longueur de φ :

- Si φ = 0, les automates suivants acceptent le langage Ø,

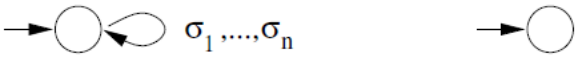

Figure III.1 AFD et AFND acceptant Ø

- Si φ = e, les automates suivants acceptent le langage {Ɛ}

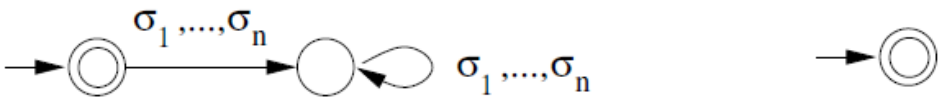

Figure III.2 AFD et AFND acceptant {Ɛ}

- Si φ = σ , σ ϵ Σ, les automates suivants acceptent le langage {σ }

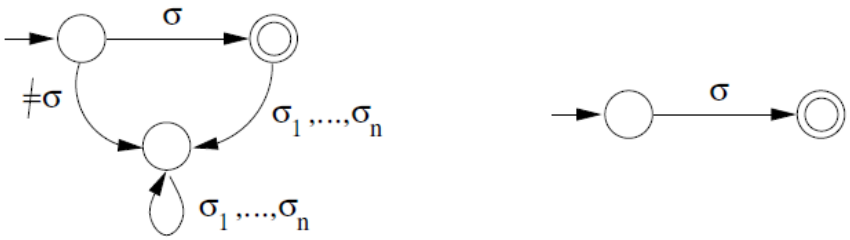

Figure III.3 AFD et AFND acceptant {σ}

- Si φ = Ψ + μ, avec Ψ et μ des expressions régulières de longueurs inférieure à celle de φ, alors par hypothèse de **récurrence**, on dispose de deux automates finis A_ψ et A_μ acceptant respectivement $\mathcal{L}(\Psi)$ et $\mathcal{L}(\mu)$. Donc il existe un automate fini qui accepte $\mathcal{L}(\Psi) \cup \mathcal{L}(\mu)$
- Si φ = Ψ.μ, même raisonnement que précédemment : Donc il existe un automate fini qui accepte $\mathcal{L}(\Psi) \cap \mathcal{L}(\mu)$
- Si $\varphi = \Psi^*$, On tire la même conclusion en utilisant cette fois la proposition : « Si L est accepté par un automate fini alors L^* l'est aussi ».

Remarque : Ainsi de proche en proche, on peut, étant donné une expression régulière, construire un automate acceptant le langage généré par l'expression.

b. Exemple :

Soit l'expression régulière $\varphi = (a^* ba^*b)^* a^*$.

Des automates acceptant $\mathcal{L}(a) = \{a\}$ et $\mathcal{L}(b) = \{b\}$

Figure III.4 AFND acceptant {a} et {b}.

En utilisant la proposition II.3.3 on construit un automate acceptant a^*.

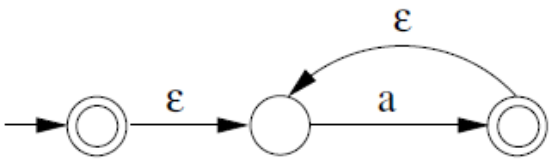

Figure III.5 AFND acceptant $\{a\}^*$.

Pour des raisons de simplifications évidentes, nous allons considérer un automate équivalent acceptant aussi a^* :

Figure III.6 AFND acceptant $\{a\}^*$.

En utilisant la proposition II.3.2, on construit un automate acceptant a^*b :

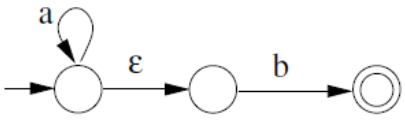

Figure III.7 AFND acceptant $\{a\}^*\{b\}$.

<

En utilisant cette proposition une seconde fois, on obtient un automate acceptant $a^*b\,a^*b$:

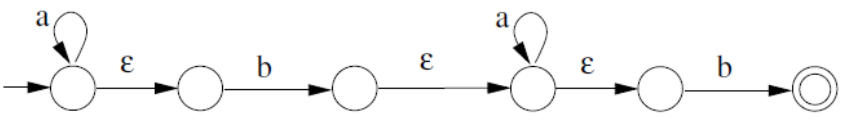

Figure III.8 AFND acceptant $\{a\}^*\{b\}\{a\}^*\{b\}$.

Et en simplifiant quelque peu, on a même :

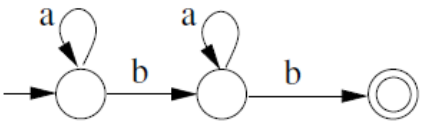

Figure III.9 AFND acceptant $\{a\}^*\{b\}\{a\}^*\{b\}$.

Appliquons à présent la proposition II.3.3 à ce dernier automate pour obtenir :

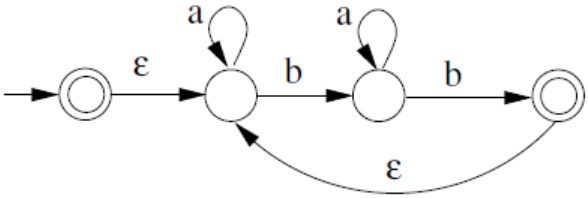

Figure III.10 AFND acceptant $(\{a\}^*\{b\}\{a\}^*\{b\})^*$.

La dernière étape consiste à combiner l'automate ci-dessus avec celui acceptant a^* au moyen de la proposition II.3.2.

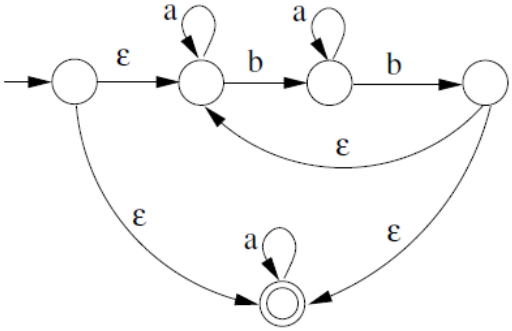

Figure III.11 AFND acceptant $(\{a\}^*\{b\}\{a\}^*\{b\})^*\{a\}^*$.

II. Des automates aux expressions régulières

Nous définissons tout d'abord des automates généralisés dont les arcs ont comme label non pas des lettres de l'alphabet Σ mais des expressions régulières. Pour rappel, on note R_Σ, l'ensemble des expressions régulières sur Σ.

a. Définition :

Un automate fini étendu1 (AFE) est la donnée d'un quintuple : A = (Q; q_0; F; Σ; δ) Où

- Q est un ensemble fini d'états,
- $q_0 \in Q$ est l'état initial,
- $F \subseteq Q$ est l'ensemble des états finals,
- $\delta : Q \times Q \rightarrow R_\Sigma$ est la fonction d'étiquetage des transitions.

Si aucune transition entre q et q' n'est explicitement pas définie, on pose :

- $\delta(q, q') = 0$ si $q \neq q'$
- et $\delta(q, q') = e$ si $q = q'$.

b. Exemple :

L'automate présenté à la figure III.12 est un AFE, on a :

- $\delta(1,2) = ab^*$
- $\delta(2,2) = a + ba$
- $\delta(2,3) = bab$
- $\delta(1,1) = \delta(3,3) = e$
- $\delta(i,j) = 0 \; pour \; i,j \in \{(1,3),(2,1),(3,1),(3,2)\}$

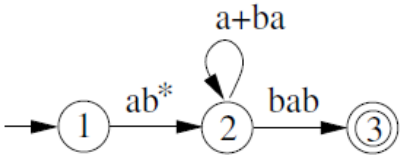

Figure III.12 : Un automate Fini étendu (AFE).

c. Définition :

Un mot w est accepté par un AFE : A = (Q; q_0; F; Σ; δ) s'il existe des mots $w_1, \ldots, w_n \in \Sigma^*$ et des états $q_1, \ldots, q_n \in Q$ tels que $w = w_1 \ldots w_n$

$$w_1 \in L(\delta(q_0, q_1)), \ldots, w_n \in L(\delta(q_{n-1}, q_n)) \text{ et } q_n \in F$$

Par exemple, pour l'AFE donné dans l'exemple précédent, le mot :

- w = abbbbbabab est accepté car si on pose :
  - w1 = abbbb,
  - w2 = ba,
  - w3 = bab.
- On s'aperçoit que :
  - $w_1 \in L(ab^*)$,
  - $w_2 \in L(a + ba)$,
  - $w_3 \in L(bab)$.

d. Définition :

Le langage accepté par un AFE est l'ensemble des mots qu'il accepte. Deux AFE sont dits équivalents s'ils acceptent le même langage.

e. Remarque :

Un AFD est un cas particulier d'AFE où toutes les transitions sont des expressions régulières de la forme σ, σ ∈ Σ. Ainsi les techniques décrites ci-après peuvent s'appliquer au départ d'un AFD.

Dans les lignes qui suivent, nous allons expliquer comment, au départ d'un AFE arbitraire, obtenir un AFE équivalent possédant uniquement deux états (un état initial et un état final). De cette manière, il sera aisé d'en déduire une expression régulière du langage accepté.

Le pivotage (élimination des états qui ne sont ni initial, ni final).

Soit A = (Q, q_0, F, Σ, δ) un AFE. Pour tout p, q ∈ Q, on note r_{pq} l'expression régulière δ (p, q). Soit q un état de A tel que q ≠ q_0 et q ∉ F.

Définissons l'A. F. E. A'= (Q', q_0, F, Σ, δ')

Où : Q' = Q \ {q} et pour tout p, s ∈ Q'on :

$$\delta'(p, s) = r_{ps} + r_{pq} r_{qq}^{*} r_{qs}$$

Par construction, il est clair que A' est équivalent à A.

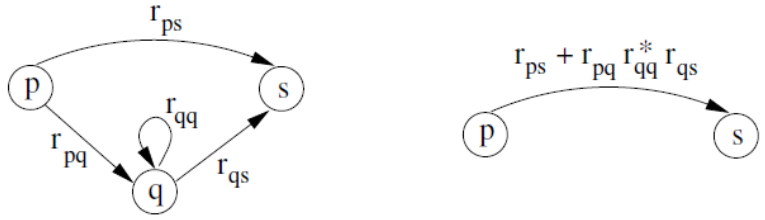

Figure III.13 : Le pivotage

f. Exemple :

Considérons l'AFE donné à la figure III.14.

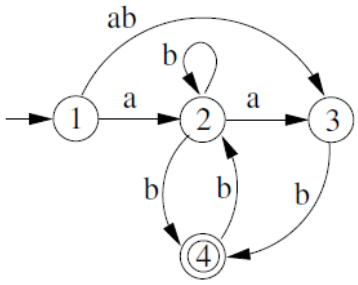

Figure III.14 : Un AFE avant élimination de l'état 2.

Avec les notations précédentes, si on désire éliminer l'état 2, on obtient :

- $\delta'(1,1) = r_{11} + r_{12}\, r22^{*}\, r_{21} = e + a\, b^{*}\, 0 = e$
- $\delta'(1,3) = r_{13} + r_{12}\, r_{22}^{*}\, r_{23} = ab + a\, b^{*}\, a$
- $\delta'(1,4) = r_{14} + r_{12}\, r_{22}^{*}\, r_{24} = 0 + a\, b^{*}\, b = ab^{*}\, b$

- $\delta'(3,1) = r_{31} + r_{32}r_{22}^{*}r_{21} = 0 + 0\,b^{*}\,0 = 0$
- $\delta'(3,3) = r_{33} + r_{32}r_{22}^{*}r_{23} = e + 0\,b^{*}\,a = e$
- $\delta'(3,4) = r_{34} + r_{32}r_{22}^{*}r_{24} = b + 0\,b^{*}\,b = b$
- $\delta'(4,1) = r_{41} + r_{42}r_{22}^{*}r_{21} = 0 + bb^{*}\,0 = 0$
- $\delta'(4,3) = r_{43} + r_{42}r_{22}^{*}r_{23} = 0 + bb^{*}\,a = bb^{*}\,a$
- $\delta'(4,4) = r_{44} + r_{42}r_{22}^{*}r_{24} = e + bb^{*}\,b = bb^{*}\,b + e$

En ne représentant que les transitions différentes de 0 et différentes de δ (q, q)=e, on obtient l'AFE équivalent représenté à la figure III.15 :

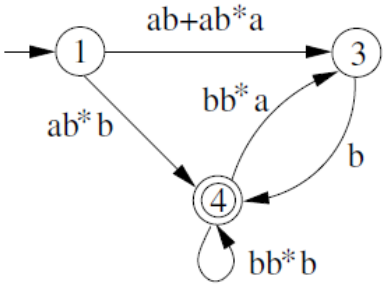

Figure III.15 : AFE équivalent après élimination de l'état 2.

g. L'algorithme complet : (Algorithme de McNaughton-Yamada)

Soit A = (Q, q_0, F, Σ, δ) un AFE.

1. Obtention d'un état initial/final :
   a. Obtention d'un **état initial non final** et auquel on ne peut aboutir :
      i. Si l'état initial q_0 est final ou s'il existe q ∈Q tel que : $\delta(q, q_0) \neq 0$,
      ii. Alors on ajoute un nouvel état $q_0' \in$ Q d'états et on pose : $\delta(q_0', q_0) = e$,
      iii. On redéfinit q_0' comme le nouvel état initial.
   b. Obtention d'un unique état final :
      i. Si $cardinal(F) > 1$, c-à-d, s'il y a plus d'un état final,
      ii. On ajoute un nouvel état f' et on pose $\delta(q, f') = e$ pour tout q ∈F,
      iii. Ensuite, on redéfinit {f'} comme nouvel ensemble d'états finals
   c. Ainsi à la fin de l'étape 1, on peut supposer disposer de :
      i. Un AFE équivalent A' = (Q', q'_0, F', Σ, δ')
      ii. Possédant :
         1. Un unique état initial q'_0 (non final et auquel n'aboutit aucune transition), et
         2. Un unique état final f'
2. Fin ?
   a. Si $Q' = \{q_0', f'\}$,
   b. Alors une expression régulière du langage accepté par A' est : $r_{q_0'f'}(r_{f'f'})^{*}$, où :

i. $r_{q_0'f'} = \delta'(q_0', f')$

ii. $r_{f'f'}{}^{*} = \delta'(f', f')$

c. $L'algoritmes'achève$ sinon on passe à l'étape 3

3. Elimination d'un état :

a. Il existe $q \in Q'\backslash\{q_0', f'\}$,

b. On élimine q de A' par la méthode du pivot présentée ci-dessus.

c. Après pivotage :

i. L'ensemble d'états est Q \ {q}

ii. On recommence le point 2,

iii. A chaque étape, le nombre d'états décroît strictement,

iv. Par conséquent l'algorithme s'achève toujours.

h. III.2.7 : Exemple :

Poursuivons l'exemple III.2.6. Si on élimine le sommet 3 de l'AFE de la figure III.15, il vient :

- δ' (1,1) = $r_{11} + r_{13}r_{33}^{*}r_{31}$ = e + (ab + ab^{*} a) e 0 = e
- δ' (1,4) = $r_{14} + r_{13}r_{33}^{*}r_{34}$ = ab^{*} b + (ab + ab^{*} a) e b
- δ' (4,1) = $r_{41} + r_{43}r_{33}^{*}r_{31}$ = 0 + (bb^{*} b) e 0 = 0
- δ' (4,4) = $r_{44} + r_{43}r_{33}^{*}r_{34}$ = bb^{*} b + (bb^{*} a) e b

On obtient l'automate représenté à la figure III.16 :

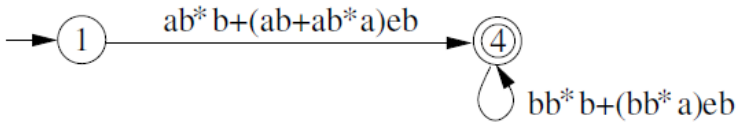

Figure III.16 : AFE équivalent après élimination de l'état 3.

Finalement une expression régulière du langage accepté par l'automate de départ est :

(ab^{*} b + (ab + ab^{*} a) b)(bb^{*} b + (bb^{*} a) b)*.

Puisqu'à toute expression régulière φ, correspond un automate acceptant le langage $\mathcal{L}(\varphi)$ et qu'à tout langage L accepté par un automate correspond une expression régulière Ψ telle que L = $\mathcal{L}(\Psi)$, nous avons le résultat suivant.

i. III.2.8. Théorème : (Klenne)

Un langage est régulier si et seulement si il est accepté par un automate fini.

j. III.2.9. Remarque :

D'une certaine manière, on peut dire que les expressions régulières sont les générateurs des langages réguliers, alors que les automates finis en sont les accepteurs.

III. Stabilité de la régularité

a. III.3.1. Théorème :

L'ensemble des langages réguliers est stable par union, concaténation, étoile de klenne, image par morphisme, miroir, passage au complémentaire, intersection et shuffle.

Le résultat suivant est souvent utilisé pour vérifier que certains langages ne sont pas réguliers. Il s'agit simplement d'une redite du corollaire I.3.10.

b. III.3.2. Corollaire :

Si L est un langage régulier sur $\Sigma = \{\sigma_1, \dots, \sigma_n\}$, alors $|L| = \{|\omega| : \omega \epsilon L\} \subseteq \mathbb{N}$ est une union finie de progression arithmétique.

IV. Les critères de non-régularité

Introduction : aaa

a. III.4.1. Lemme : (Le lemme de la pompe)

Soit $L \subseteq \Sigma^* = \{\sigma_1, \dots, \sigma_n\}$, un langage régulier. Il existe un entier k tel que pour tout mot ω de L satisfaisant $|\omega| \geq k$, il existe $x, y, z \epsilon \Sigma^*$ tels que $\omega = xyz$ et

- $|xyz| \leq k$
- $y \neq \varepsilon$
- $xy^*z \subset L$

b. Exemple :

Considérons une fois encore le langage :

$$L = \{a^{n^2} : n \epsilon \mathbb{N}\}$$

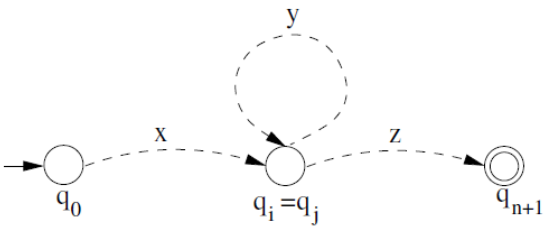

Figure III.14 l lemme de la pompe.

Nous avons déjà montré dans l'exemple I.3.13 que ce langage n'était pas régulier (en utilisant la proposition I.3.9). Utilisons ici le lemme de la pompe.

Si le langage L était régulier, il serait accepté par un AFD A ayant k états.

Dès lors, le mot a^{k^2} est accepté par A et cet automate comprend donc une boucle de longueur i > 0 (car k^2 > ou = k). Par conséquent tout mot de longueur $K^2 + ni, n \epsilon \mathbb{N}$ est accepté par A.

Or, l'ensemble des carrés parfaits ne contient aucune progression arithmétique infinie. On en tire que le langage L ne peut être régulier.

c. III.4.3. Remarque :

Attirons l'attention du lecteur sur le fait que des langages non réguliers peuvent néanmoins satisfaire la condition du lemme de la pompe.

En effet, soit $L \subset b^*$un langage non régulier arbitraire. Le langage $\{a\}^+L \cup \{b\}^*$ Satisfait le lemme de la pompe. Il suffit de prendre avec les notations du lemme, k=1.

La version suivante du lemme de la pompe fournit une condition nécessaire et suffisante pour qu'un langage soit régulier.

d. III.4.4. Lemme : (Lemme de la pompe version forte)

Soit $L \subseteq \Sigma^* = \{\sigma_1, \dots, \sigma_n\}$, un langage.

Le langage L est régulier si et seulement si :

- Il existe une constante k > 0 telle que :
  - Pour tout mot $\omega \epsilon \Sigma^*$,
  - Si $|\omega| \geq k$,
  - Alors il existe $x, y, z \epsilon \Sigma^*$ tels que :
    - $\omega = xyz$
    - $y \neq \varepsilon$
    - Et pour tout $i \geq 0$
    - Et pour tout $\vartheta \epsilon \Sigma^*$, $\omega\vartheta \epsilon L \leftrightarrow$ xy^izϑ ϵ L
- $|xyz| \leq k$
- $y \neq \varepsilon$
- $xy^*z \subset L$

e. Remarque :

Nous voulons faire observer au lecteur que cette dernière proposition nécessite une décomposition de ω en xyz qui doit pouvoir être appliquée pour <u>tout</u> mot : $\omega\vartheta$, $\vartheta \epsilon \Sigma^*$