Remerciements

Ce mémoire est toujours une version provisoire. Malgré tout le soin apporté à sa rédaction, il est possible que quelques erreurs soient toujours présentes dans le mémoire.

A terme de cette étude, nous avons le plaisir et le devoir de remercier en premier lieu **Allah** le tout puissant de nous avoir donné la volonté et le courage pour accomplir cette mémoire.

L'occasion nous tenons à adresser notre sincère remerciement à l'encadreur :

Monsieur **MIHOUBI DOUADI,** et **GHADBANE NACER** pour sa bienveillance, et pour son aide précieux qu'il nous a apportée.

Je remercier les membres du jury qui ont accepté de juger mon travail.

Mr. A. AMROUNE. Mr. N. GHADBANE.

Mr. D. MIHOUBI. Mr. L. HEBOUB.

Avec remerciements les plus profonds à nos enseignants qui nous guident durant toutes nos années de formation.

Sans oublier de tenir à remercier tous mes parents, mes familles et sur tout mon mari hakim et mes amis, pour leur soutient tout au long de nos étude.

Table des matières

Introduction générale

Les langages rationnels sont le premier niveau de la hiérarchie de Chomsky. Cette famille est constituée des langages acceptés par les automates finis qui sont le modèle le plus simple de machines. Cette famille de langages jouit de propriétés remarquables. Elle est en particulier close par de très nombreuses opérations.

La théorie des automates est une théorie relativement riche et certains de ses résultats sont de véritables perles. Elle entretient aussi des liens avec beaucoup d'autres domaines comme la dynamique symbolique, la combinatoire, l'algèbre, la topologie, la théorie des jeux, l'arithmétique, la logique et l'algorithmique.

Un langage rationnel peut être vu comme les étiquettes des chemins dans un automate. Sa hauteur d'étoile est une mesure de la complexité en boucles d'automates associés au langage. La notion de hauteur d'étoile d'une expression rationnelle exprimée au moyen de l'union, de la concaténation et de l'étoile, a été introduite par Eggan. La hauteur d'étoile d'un langage formel est le nombre minimal d'étoiles superposées dans une expression décrivant ce langage. La difficulté de ce problème vient du fait que, dans le cas général, il existe une infinité d'expressions associées à un langage donné.

Dans ce mémoire, on s'intéresse à une étude sur les langages rationnels ainsi que l'étude du problème de la hauteur de l'étoile et son interprétation sur les automates finis.

Nos références, dans l'étude du problème de hauteur de l'étoile sont : [5] et [10].

Dans le premier chapitre, on donne un aperçu général sur le monoïde libre, puis on exposera des propriétés fondamentales concernant les mots et les langages, de plus on donne quelques opérations importantes sur les mots et les langages dans la fin de ce chapitre, on donnera le lemme d'Arden qui joue un rôle important pour résoudre le système d'équation pour la suite de ce mémoire.

Le deuxième chapitre présent les notions fondamentales de langages réguliers (ou langages rationnels), d'expressions régulières et d'automates finis, puis on peut être décrire les langages réguliers de plusieurs façons équivalentes :

• Ce sont les langages décrits par les expressions régulières ou rationnelles.

• Ce sont les langages obtenus, à partir des lettres et de l'ensemble vide, par les opérations rationnelles, à savoir l'union, le produit et l'étoile (ou fermeture de Kleene).

• Ce sont les langages reconnus par des automates finis, d'où le nom de langages reconnaissables.

L'équivalence entre l'expression rationnelles et l'automate finis exprime en théorème de Kleene, et aussi on étudie l'automate minimale qui joue de propriétés intéressantes, de plus on donne la définition de monoïde syntaxique.

Dans le troisième chapitre, on étudie le lien entre la hauteur d'étoile d'un langage rationnel et l'automate fini qui reconnait ce langage. D'abord, on présente une introduction générale et on donne une partie historique sur ce problème, puis on introduit la notion de rang cyclique d'un automate. Eggan a montré que la hauteur d'étoile d'un langage rationnel est égale au minimum des rangs cycliques des automates reconnaissant ce langage, enfin McNaughton a montré que la hauteur d'étoile du langage peut être déterminée en utilisant son automate minimal.

Chapitre 1

Préliminaires

Dans ce chapitre on donne un aperçu général sur les mots et les langages. Dans la première partie, on définit quelques notions de base sur : Semigroupe, monoïde. La deuxième partie est consacrée à l'étude un monoïde particulier (monoïde libre) et ces propriétés, les mots et langages, cette partie qui contient : généralités sur les mots et les langages, ensuit des résultats sur les mots, ainsi quelques opérations importantes, enfin on s'intéresse tout particulièrement au lemme de Arden.

1.1 Semigroupe, sous-semigroupe, monoïde

Définition 1.1.1

Un semigroupe $(S, .)$ est un ensemble S muni d'une opération binaire interne. Cette opération est appelée le produit et le note $m.n$ pour m et n éléments de S et satisfait l'axiome d'associativité :

$$\forall m_1,\, m_2,\, m_3 \in S,\, (m_1 m_2)m_3 = m_1(m_2 m_3).$$

Exemple 1.1.1

1. $(\mathbb{N}, +)$; $(\mathbb{N}, .)$; $(\mathbb{Z}, +)$; $(\mathbb{Z}, .)$; $(\mathbb{R}, +)$; $(\mathbb{R}, .)$ sont des semigroupes, puisque l'opération $(+)$ est associative.

2. $(\mathbb{Z}, -)$ n'est pas un semigroupe car l'opération " $-$ " n'est pas associative, par exemple: $2 - (3 - 5) \neq (2 - 3) - 5$.

Définition 1.1.2

Soit S un semigroupe, soit S' une partie de S tel que :

$$\forall a,\ b \in S' : a.b \in S'.$$

Donc S' est un sous-semigroupe de S.

Définition 1.1.3

Un monoïde $(M, ., 1_M)$ est un semigroupe avec un élément remarquable $1_M \in M$ qui a la propriété d'être un élément neutre pour le produit:

$$\forall m \in M,\ 1_M m = m = m 1_M.$$

Remarque 1.1.1 *Un monoïde $(M, ., 1_M)$ qui est tel que tout élément de M possède un inverse est un groupe.*

Exemple 1.1.2

1. Tout groupe est un monoïde; $(\mathbb{N}, +, 0)$ est un monoïde qui n'est pas un groupe.
2. L'ensemble $\mathbb{N}$ muni de la multiplication $\times$ et de l'élément neutre 1 est aussi un monoïde.
3. L'ensemble des application $E \to E$, avec l'opération de composition (la composition $g \circ f$ de f et g est notée $g.f$ ou gf, est définie par $(gf)(x) = g(f(x))$ pour $x \in E$) d'élément neutre l'application identique est un monoïde.

1.2 Mots et langages

On introduit dans cette partie quelques définitions, propriétés et notations concernant les mots et les langages. (Pour plus de détails voir [8] et [9] et [10]).

Définition 1.2.1

Un alphabet est un ensemble fini non vide. Un alphabet sera en général désigné par une lettre grecque majuscule. Ainsi $A = \{a,\ b,\ c,\ d\}$, $\triangle = \{\bigstar,\ \heartsuit,\ \emptyset,\ \measuredangle\}$, $\Omega = \{0,\ 1\}$ sont des alphabets. Les éléments d'un alphabet sont appelés lettres ou symboles. En général, il est noté Σ ou A.

Exemple 1.2.1

Le biologiste intéressé par l'étude l'ADN (Acide désoxyribo Nucléique) utilisera un alphabet à quatre lettres $\{A, C, G, T\}$ pour les quatre constituants des gènes : Adénine, Cytosine, Guanine et Thymine.

Définition 1.2.2

Soit A un alphabet. Un mot sur A est une suite finie de symboles. Par exemple, abbabc et abc sont deux mots sur l'alphabet $\{a, b, c\}$, La longueur d'un mot w est le nombre de symboles constituant ce mot; on la note $|w|$. Ainsi, $|abbabc| = 6$ et $|abc| = 3$. L'unique mot de longueur 0 est le mot correspondant à la suite vide. Ce mot s'appelle le mot vide et on le note 1, ou bien ε. L'ensemble des mots sur A est noté A^*. Par exemple,

$\{0, 1, 2\}^* = \{\varepsilon, 0, 1, 2, 00, 01, 02, 10, 11, 12, 20, 21, 22, 000, 001, \ldots\}$ (ε *le mot vide*).

Exemple 1.2.2

Si $A = \{a, b\}$, il y a un mot de longueur 0, ε, deux mots de longueur 1, (a) et (b) et plus généralement, 2^n mots de longueur n.

Définition 1.2.3

Si a est une lettre de l'alphabet A, pour tout $w = a_1 a_2 \ldots a_k \in A^*$, on note par :

$$|w|_a = card\ \{i \in \{1, 2, \ldots, k\} : a_i = a\}.$$

Exemple 1.2.3

Soit $X = \{a, b, c\}$ et le mot abb. $|abb| = 3$; $|abb|_a = 1$, $|abb|_b = 2$, $|abb|_c = 0$.

Définition 1.2.4

Soit A un alphabet. On définit l'opération de concaténation sur A^* de la façon suivante: pour tout mots $u = a_1 \cdots a_n$ et $v = b_1 \cdots b_m$, où $1 \leq i \leq n$, $1 \leq j \leq m$, $a_i, b_j \in A$, la concaténation de u et v, noté $u.v$ ou uv est le mot :

$$C = c_1 c_2 \ldots c_{n+m} \text{ où } c_i = a_i,\ 1 \leq i \leq n \text{ et } c_{n+i} = b_i,\ 1 \leq i \leq m.$$

Ainsi, A^* muni de l'opération de concaténation est un monoïde de neutre 1. *En particulier, on définit la puissance n-iéme d'un mot de w comme la concaténation de n copies de w,*

$$w^n = \underbrace{w \dots w}_{n\ fois}$$

On pose $w^0 = 1$.

Par exemple, sur l'alphabet $A = \{a,\ b,\ c\}$, si $u = aabb$ et $v = cc$, alors $u.v = aabbcc$ et $u^3 = aabbaabbaabb$.

Remarque 1.2.1 *L'ensemble A^* muni de la concaténation est un monoïde libre.*

Proposition 1.2.1

La loi produit (ou concaténation) possède les propriétés suivantes :
(1) $\forall \alpha,\ \beta \in A^*,\ |\alpha.\beta| = |\alpha| + |\beta|$.
(2) $\forall \alpha,\ \beta,\ \gamma \in A^*,\ (\alpha.\beta).\gamma = \alpha.(\beta.\gamma)$ (la loi est associative).
(3) $\forall \alpha \in A^*,\ \alpha.\varepsilon = \varepsilon.\alpha = \alpha$ (le mot vide ε est l'élément neutre du produit).
(4) $\forall \alpha \in A^*,\ \alpha.\alpha = \alpha \Leftrightarrow \alpha = \varepsilon$ (le mot vide ε est le seul mot idempotent).

Exemple 1.2.4

Si $u = abaa$ et $v = bab$, on a $uv = abaabab$ et $vu = bababaa$. La concaténation n'est pas commutative.

Définition 1.2.5 *Si A est un alphabet et $k \in \mathbb{N}$, on note A^k l'ensemble des mots de longueur k construits à partir de A.*

Exemple 1.2.5

Soit $A = \{0,\ 1\}$
(1) $A^0 = \{\varepsilon\}$.
(2) $A^1 = \{0,\ 1\}$.
(3) $A^2 = \{00,\ 01,\ 10,\ 11\}$.
(4) $A^3 = \{000,\ 001,\ 100,\ 010,\ 110,\ 011,\ 101,\ 111\}$.

Remarque 1.2.2 *Il est utile de remarquer que si* $card(A) > 1$*, alors* A^* *est un monoïde non commutatif, i.e., il existe* $u, v \in A^*$ *tels que* $uv \neq vu$*.*

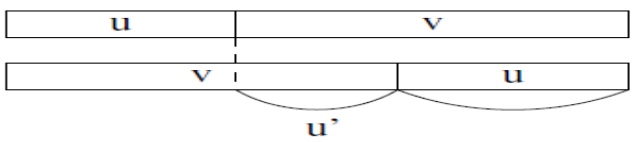
Proposition 1.2.2

Deux mots u *et* v *commutent s'ils sont puissances d'un même troisième, i.e., s'il existe un mot* w *et des entiers* i, j *tels que* $u = w^i$ *et* $v = w^j$*.*

Preuve.

On procède par récurrence sur la longueur de uv. Si $|uv| = 0$, le résultat est immédiat. Supposons à présent le résultat satisfait pour $|uv| < n$. Soient u, v tels que $|uv| = n$. On peut même considérer que $u \neq 1$ et $v \neq 1$ car sinon, le résultat serait trivial.

Si $|u| = |v|$, alors il est immédiat que $u = v$. Sinon, on peut supposer que $|u| < |v|$ (voir la figure ci-dessous),



Donc il existe u' tel que $v = uu'$ et $|u'| < |v|$. Ainsi, $uv = uu'u = vu = u'uu$ et donc on trouve $u'u = uu'$. Puisque $|uu'| < |uv|$, on peut appliquer l'hypothèse de récurrence. Il existe un mot w et des entiers p, q tels que $u = w^p$ et $u' = w^q$. Pour conclure, on remarque que $v = u'u = w^{p+q}$. ■

Remarque 1.2.3 *Noter que la réciproque du résultat ci-dessus est triviale.*

Proposition 1.2.3 ***(de Levy)***

Soient x, y, z, t *des mots tels que* $xy = zt$*, alors il existe un mot* w *tel que :*

$(xw = z$ avec $y = wt)$ *ou* $(x = zw$ avec $wy = t)$*.*

Il en résulte en particulier que si $|x| = |z|$*, le mot* w *est vide donc* $x = z$ *et* $y = t$*.*

Preuve.

Posons $x = a_1 a_2 \dots a_n$, $y = a_{n+1} \dots a_m$ avec $a_i \in A$ et $1 \leq i \leq m$, de même $z = b_1 b_2 \dots b_k$, $t = b_{k+1} \dots b_q$ avec $b_i \in A$ et $1 \leq i \leq q$, comme $xy = zt$, nous avons $m = q$ (mais pas

nécessairement $n = k$ et $a_i = b_i$ pour $i = 1, 2, \ldots m$) de sorte que $z = a_1 a_2 \ldots a_n$ et $t = a_{n+1} \ldots a_m$.

Si $|z| = k \leq n = |x|$, posons $w = a_{n+1} \ldots a_n$, alors $x = zw$ avec $wy = t$.

Si $|z| > |x|$ posons $w = a_{n+1} \ldots a_k$ alors $xw = z$ et $y = wt$. ■

Définition 1.2.6

Un morphisme d'un monoïde M dans un monoïde N est une application ϕ telle que :

(1) $\phi(uv) = \phi(u)\phi(v)$.

(2) $\phi(\varepsilon_M) = \varepsilon_N$ l'image de l'élément neutre de M est l'élément neutre de N.

Exemple 1.2.6

L'application longueur $|.| : A^* \to \mathbb{N}$ est un morphisme de monoïdes entre $(A^*, .)$ et $(\mathbb{N}, +)$. En effet et,

$$\forall u, v \in A^* : |uv| = |u| + |v| \text{ et } |1| = 0.$$

Exemple 1.2.7

Soit $A = \{a_1, a_2, \ldots, a_n\}$ un alphabet, $n \in \mathbb{N} \setminus \{0, 1\}$.

La fonction de Parikh :

$$\Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_n}).$$

est un morphisme de monoïdes entre $(A^*, .)$ et $(\mathbb{N}^n, +)$.

▶ La proposition suivante justifie le fait que le monoïde A^* soit appelé monoïde libre. Cette propriété caractérisé le monoïde libre engendré par A.

Proposition 1.2.4

Toute fonction $\mu : A \to M$ de A dans un monoïde M se prolonge de façon unique en un morphisme de monoïde de A^* dans M.

Preuve. [2]

Existence **:** Posons $\widetilde{\mu}(1) = e_M$ et $\widetilde{\mu}(a_1 \cdots a_n) = \mu(a_1) \ldots \mu(a_n)$, $n \in \mathbb{N}$, $a_i \in A$, $1 \leq i \leq n$.

Il est facile de voir que $\widetilde{\mu}$ est bien un homomorphisme.

Unicité **:** Soient $\widetilde{\mu}$ et $\widetilde{\lambda}$ deux homomorphismes de A^*dans M tels que :

$\forall a \in A, \widetilde{\mu}(a) = \widetilde{\lambda}(a)$.

Alors $\widetilde{\mu}(1) = \widetilde{\lambda}(1) = e_M$ et pour tout mot $w = a_1 \cdots a_n$,

$\widetilde{\mu}(w) = \widetilde{\mu}(a_1 \cdots a_n)$

$= \mu(a_1)\mu(a_2) \ldots \mu(a_n)$

$= \widetilde{\lambda}(a_1) \ldots \widetilde{\lambda}(a_n)$

$= \widetilde{\lambda}(a_1 \cdots a_n) = \widetilde{\lambda}(w)$. ■

Exemple 1.2.8

Considérons l'alphabet $A = \{a,\ b,\ c\}$ et le morphisme $\varphi : A^* \longrightarrow A^*$ défini par :

$\varphi(a) = abc$, $\varphi(b) = ac$ et $\varphi(c) = b$. En effet, pour définir un tel morphisme, on remarquer qu'il suffit de se donner l'image des lettres de l'ensemble de départ. On a, par exemple,

$\varphi(abbc) = \varphi(a)\varphi(b)\varphi(b)\varphi(c) = abcacacb$.

Proposition 1.2.5 *[7, p.02]*

Les conditions nécessaires et suffisantes pour qu'un monoïde M soit un monoïde libre :

1) Il existe un homomorphisme λ de M sur $\mathbb{N}$ ensemble des entiers positifs avec $\lambda^{-1}(0) =1$ (1 l'élément neutre de M).

2) *Quelque soit $f_1,\ f_2,\ f_3,\ f_4 \in M$ tels que $f_1 f_2 = f_3 f_4$ on a l'une des deux situations suivantes :*

$$\exists f_5 \in M : f_1 = f_3 f_5 \text{ et } f_5 f_2 = f_4.$$
$$\exists f_6 \in M : f_3 = f_1 f_6 \text{ et } f_6 f_4 = f_2.$$

Exemple 1.2.9

Soit $X \subseteq A^*$, le monoïde engendré par X est définit par :

$X^* = \{w = x_1 x_2 \ldots x_n$, oú pour $1 \leq i \leq n$, $x_i \in X$, $n \in \mathbb{N}\ \}$.

Montrons que X^*est un monoïde libre.

1) On définit l'homomorphisme λ comme suite :

$\lambda : X^* \longrightarrow \mathbb{N}$

$w \longmapsto |w|$ avec $\lambda^{-1}(0) = \{1\}$

2) Quelque soit f_1, f_2, f_3, $f_4 \in X^*$ tels que $f_1 f_2 = f_3 f_4$ d'après le lemme de Levy on a l'une des deux situations suivantes :

$$\exists f_5 \in X^* : f_1 = f_3 f_5 \text{ et } f_5 f_2 = f_4.$$
$$\exists f_6 \in X^* : f_3 = f_1 f_6 \text{ et } f_6 f_4 = f_2.$$

1.2.1 Opérations sur les mots

Définition 1.2.7

Soit $w = a_1 a_2 \dots a_k$ un mot sur A. Les mots,

$$1, a_1, a_1 a_2, \ \dots, \ a_1 a_2 \dots a_{k-1}, a_1 a_2 \dots a_k = w.$$

sont les préfixes de w. Un préfixe de w différent de 1 et w est dit propre.

De façon semblable,

$$1, a_k, a_{k-1} a_k, \ \dots, \ a_2 \dots a_k, a_1 a_2 \dots a_k = w.$$

sont les suffixes de w. Un suffixe de w est qualifié de propre s'il diffère de 1 et de w.

Soient $1 \leq i \leq j \leq k$, le mot $a_i \dots a_j$ est un facteur du mot w, on parle de facteur propre lorsque ce dernier diffère de w et de 1, l'ensemble des préfixes (resp. suffixes, facteurs) de w est noté $Pref(w)$ (resp. $Suff(w)$, $Fac(w)$).

Si w se factorise en $u_1 v_1 u_2 v_2 ... u_n v_n u_{n+1}$, où tous les u_i et v_i sont des mots de A^*, alors $v = v_1 v_2 ... v_n$ est sous-mot de w.

Définition 1.2.8

On appelle image de miroir d'un mot $w = a_1 a_2 \dots a_k$ le mot :

$$w^R = a_k \dots a_1.$$

▶ *Un mot w est primitif (non-décomposable) si $w = y^n$ alors $n = 1$(n'admet pas de solution pour $n > 1$).*

▶ *Un mot w est périodique (décomposable) si $\exists k > 1$ et y primitive lel que $w = y^k$(y est dit la période).*

▶ *Un mot w est un palindrome s'il est égal à son miroir c'est-dire $w^R = w$.*

Par exemple, soit $A = \{a, b\}$ un alphabet et soit $u, v \in A^*$,

- $u = bab, u$ est primitif.

- $v = babbab = (bab)^2, v$ est périodique dans la période est bab.

Exemple 1.2.10

1. Si abb est facteur de $babba$, et ba est à la fois préfixe et suffixe, mais aa n'est pas facteur.
2. Soit $w = aabccaa$. Alors bcc est un facteur de w; $aabc$ est un préfixe de w; caa est un suffixe de w; aa est à la fois préfixe et suffixe de w.
3. $abbaa^R = aabba$; $100^R = 001$.
4. Soit $w = 10101$. On a $w^R = 10101$ est un miroir de w alors $w^R = w$ (même chose pour le mot $xyyx$) sont des palindromes.

Définition 1.2.9

Le quotient droit d'un mot u par le mot v, dénoté uv^{-1}ou encore $u_{/v}$, est défini par :

$$u_{/v} = uv^{-1} = \{w \in A^* : u = wv\}.$$

Par exemple $abcde(cde)^{-1} = ab$, et $abd(abc)^{-1} = \emptyset$. Il ne faut pas voir uv^{-1} comme un produit : v^{-1} ne représente pas un mot. On peut définir de la même façon le quotient gauche de v par u : $u^{-1}v$ ou ${}_{u\backslash}v$.

Définition 1.2.10

Les relations de préfixe, suffixe, facteur et sous-mot induisent autant de relations d'ordre sur A^*: ce sont, en effet, des relations réflexives, transitives et antisymétriques. Ainsi pourra-t-on dire que $u \preccurlyeq_p v$ si u est un préfixe de v. Deux mots quelconques ne sont pas nécessairement comparables pour ces relations : ces ordres sont partiels.

Remarque 1.2.4

Le relation précédant est une relation d'ordre noté comme sa $\preccurlyeq$ et définit :

(1) $\forall u \in A^*$, $u \preccurlyeq u$ (réflexivité).

(2) $\forall u, v, w \in A^*$, si $u \preccurlyeq w$ et $w \preccurlyeq v$ alors $u \preccurlyeq v$ (transitivité).

(3) $\forall u, w \in A^*$, si $u \preccurlyeq w$ et $w \preccurlyeq u$ alors $u = w$ (antisymétrie).

Définition 1.2.11

Il est possible de définir des ordres totaux sur A^*, à la condition de disposer d'un ordre total $\leqslant$ sur A. À cette condition, l'ordre lexicographique sur A^* noté $\leqslant_l$ est défini par $u \leqslant_l v$ ssi;

(1) Soit u est un préfixe de v.

(2) Soit sinon $u = tu'$, $v = tv'$ avec $u' \neq 1$ et $v' \neq 1$ et le premier symbole de u' précède celui de v' pour $\leqslant$.

Définition 1.2.12

Un langage sur un alphabet A est simplement un ensemble (fini ou infini) de mots sur A. En d'autres termes, un langage est une partie (sous ensemble) de A^*. L'ensemble des langages sur A est donc : $P(A^*) = \{L,\, L \subset A^*\}$. On distingue en particulier le langage vide $\emptyset$ qui ne contient aucun mot.

Classes de langages

Différentes manières de décrire des langages :

- Description verbale : L est l'ensemble des mots qui sont de longueur paire.
- Description ensembliste : $L = \{u \ / \ |u| \in 2\mathbb{N}\}$.
- Description par une expression : $L = ((a+b)^2)^*$.
- Description par une machine (automate, machine de Turing).

Exemple 1.2.11

▶ Considérons l'alphabet $A = \{a, b, c\}$.

L'ensemble $\{a, aa, bbc, ccca, ababab\}$ est un langage fini.

L'ensemble L_{2a} des mots sur A comprenant un nombre pair de a est aussi un langage (infini), $L_{2a} = \{\varepsilon, b, c, aa, bb, bc, cb, cc, aab, aac, aba, aca, \ldots, abaacaaa, \ldots\}$.

▶ Soit l'alphabet $\Delta = \{0, 1\}$. L'ensemble constitué des écritures binaires 10 des entiers positifs pairs est un langage sur, $\{10, 100, 110, 1000, 1010, 1100, 1110, \ldots\}$.

de même que le langage formé des écritures binaires des nombres premiers,

$$\{10, 11, 101, 111, 1011, 1101, 10001, \ldots\}.$$

▶ Le langage des nombres est défini sur un alphabet $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ .02, 00310 , 3200 sont alors des mots sur A. On définira le langage des nombres comme les mots sur A qui ne commencent pas par 0. Ainsi, 1233 et 3200 seront des mots du langage mais pas 00310.

Définition 1.2.13

Le miroir d'un langage L est ; $L^R = \{u^R : u \in L\}$.

On peut avoir $L = L^R$ sans pour autant que les mots de L soient tous des palindromes.

1.2.2 Opérations sur les langages

Définition 1.2.14

Soient L_1 et L_2 deux langages sur A. On appelle union de L_1 et de L_2 et l'on note $L_1 \cup L_2$ (ou $L_1 + L_2$) le langage défini par :

$$L_1 \cup L_2 = L_1 + L_2 = \{w \in A^*, w \in L_1 \vee w \in L_2\}.$$

Proposition 1.2.6

L'union est associative, commutative, d'élément neutre $\emptyset$ et d'élément absorbant A^*. Autrement dit, $\forall L$, L_1, L_2, $L_3 \in P(A^*)$ (langages sur A) :

- $(L_1 \cup L_2) \cup L_3 = L_1 \cup (L_2 \cup L_3)$.
- $L_1 \cup L_2 = L_2 \cup L_1$.
- $\emptyset \cup L = L \cup \emptyset = L$.
- $L \cup A^* = A^* \cup L = A^*$.

Définition 1.2.15

Soient L_1 et L_2 deux langages sur A. On appelle intersection de L_1 et de L_2 et l'on note $L_1 \cap L_2$ le langage défini par :

$$L_1 \cap L_2 = \{w \in A^*, w \in L_1 \wedge w \in L_2\}.$$

Considérons par exemple les deux langages $L = \{ab, bc, ac\}$ et $L' = \{aa, ac\}$ définis sur $\{a, b, c\}$: $L \cap L' = \{ac\}$.

Définition 1.2.16

Soient L_1 et L_2 deux langages sur A. On appelle différence symétrique (ou réunion disjointe) entre L_1 et L_2 et l'on note $L_1 \triangle L_2$ le langage défini par :

$$L_1 \triangle L_2 = (L_1 - L_2) \cup (L_2 - L_1) = L_1 \cup L_2 \setminus L_1 \cap L_2.$$

Définition 1.2.17

Soit L un langage sur A. On appelle complémentaire de L et l'on note $\overline{L}$ le langage défini par :

$$\overline{L} = A^* \setminus L = \{w \in A^*, w \notin L\}.$$

Définition 1.2.18

Soient L_1 et L_2 deux langages sur A. On appelle concaténation de L_1 et de L_2 et l'on note $L_1.L_2$ le langage défini par :

$$L_1.L_2 = \{uv : u \in L_1,\ v \in L_2\}.$$

Considérons par exemple les deux langages $L_1 = \{00, 11\}$ et $L_2 = \{0, 1, 01\}$ définis sur $\{0, 1\}$:

$$L_1.L_2 = \{000, 001, 0001, 110, 111, 1101\}.$$

Exemple 1.2.12

Soient les deux langages $L_1 = \{u \in A^*/\ |u|$ pair$\}$ et $L_2 = \{u \in A^*/\ |u|$ impair$\}$. On a alors les égalités suivantes.

(1) $L_1 + L_2 = L_1 \cup L_2 = A^*$.

(2) $L_1 L_2 = L_2 = L_2 L_1$.

(3) $L_2 L_2 = L_1 / \{1\}$.

(4) $L_1 L_1 = L_1$.

Définition 1.2.19

Soient L un langage sur A. On appelle $n^{ième}$ puissance de L et l'on note L^n le langage défini par :
$$\begin{cases} \{1\} \ \textit{si } n = 0 \\ L \ \textit{si } n = 1 \\ L.L^{n-1} \ \textit{sinon} \end{cases}$$

Proposition 1.2.7 *[8, p.12]*

La concaténation de langages est une opération associative, elle posséde $\{1\}$ *pour neutre,* $\emptyset$ *pour absorbant et distributive à droite et à gauche pour l'union, i.e. si* L_1, L_2, L_3 *sont des langages on a :*

- $(L_1.L_2).L_3 = L_1.(L_2.L_3)$.
- $L_1.\{1\} = \{1\}.L_1 = L_1$.
- $L_1.\emptyset = \emptyset.L_1 = \emptyset$.
- $L_1.(L_2 \cup L_3) = L_1.L_2 \cup L_1.L_3$.
- $(L_1 \cup L_2)L_3 = L_1.L_3 \cup L_2.L_3$.

Exemple 1.2.13

$$\text{Si } L = \{a, ab, ba, ac\}, \text{ alors}$$

$$L^2 = \{aa, aab, aba, aac, aba, abab, abba, abac, baa, baab, baba, baac, aca, acab, acba, acac\}.$$

Définition 1.2.20

Soient L un langage sur A. On appelle fermeture de Kleene ou étoile (ou itéré) de L et l'on note L^* *le langage défini par :*

$$L^* = \sum_{n \geq 0} L^n = L^0 + L^1 + L^2 + \ldots = \cup_{n \geq 0} L^n.$$

avec $L^0 = \{1\}$. *Il s'agit du plus petit langage sur A contenant L et le mot vide et stable par l'opération produit.*

Proposition 1.2.8 *[8, p.12 et 13]*

Soit $L \subseteq A^*$ *un langage. Le langage* L^* *est le plus petit langage M tel que* $1 \in M$, $L \subseteq M$ *et* $M^2 \subseteq M$.

Preuve. Il est clair que L^* vérifie les trois propriétés. Si M satisfait les propriétés indiquées, nous devons montrer que $L \subseteq M$ puisque $L \subseteq M$ et $M^2 \subseteq M$, on en conclut que $L^i \subseteq M$. De proche en proche, on s'aperçoit que $L^i \subseteq M$, $\forall i > 0$. Ceci conclut la preuve. ■

Définition 1.2.21

Soient L un langage sur A. On appelle étoile stricte (ou itéré strict) de L et l'on note L^+ le langage défini par :

$$L^+ = \cup_{n>0} L^n = \sum_{n>0} L^n.$$

Il s'agit du plus petit langage sur A contenant L et stable par l'opération produit.

Exemple 1.2.14

Si A est un alphabet, alors $A^+ = A^* / \{1\}$. D'une manière générale, si L est un langage ne contenant pas le mot vide, alors $L^+ = L^* / \{1\}$.

► Si $L = \{a\}$, $a \in A^*$, $L^* = \{1,\ a,\ aa,\ aaa,\ \ldots\}$ et $L^+ = \{a,\ aa,\ aaa,\ \ldots\}$.

Remarque 1.2.5

(1) $L^0 \neq 0$.

(2) $L^* = L^0 + L^+$.

(3) $L^+ = L^*.L = L.L^*$.

Définition 1.2.22

Le quotient droit d'un langage L par le mot u est défini par :

$$L_{/u} = Lu^{-1} = \cup_{v \in L} \left\{ vu^{-1} \right\} = \left\{ w \in A^* : wu \in L \right\}.$$

Le quotient droit de L par u est donc l'ensemble des mots de A^* dont la concaténation par u est dans L. De même, le quotient gauche de L par u, $u^{-1}L =_{u\backslash} L$, est l'ensemble des mots de A^* qui, concaténés à u, produisent un mot de L.

Définition 1.2.23

Soit L un langage de A^*, on définit l'ensemble des préfixes de L, noté $Pref(L)$ par :

$$Pref(L) = \{u \ / \ \exists v \in A^*, uv \in L\}.$$

Définition 1.2.24

Soit L un langage de A^*, on définit l'ensemble des suffixes de L, noté $Suff(L)$ par :

$$Suff(L) = \{u \ / \ \exists v \in A^*, vu \in L\}.$$

Définition 1.2.25

Soit L un langage de A^*, on définit l'ensemble des facteurs de L, noté $Fac(L)$ par :

$$Fac(L) = \{v \ / \ \exists u, w \in A^*, uvw \in L\}.$$

Définition 1.2.26

Une congruence droite de A^* est une relation d'équivalence R de A^* qui vérifie :

$$\forall w, w' \in A^*, wRw' \Longrightarrow (\forall u, wuRw'u).$$

Définition 1.2.27

Soit f un morphisme de monoïdes entre A^* et Γ^*. On remarque que f est complétement caractérisé par les images de f sur les symboles de A. Si L est un langage sur A, alors l'image de L par le morphisme f est;

$$f(L) = \{f(u) \in \Gamma^* / u \in L\}.$$

De la même manière, si M est un langage sur Γ, alors l'image inverse de M par le morphisme f est;

$$f^{-1}(M) = \{u \in A^* / f(u) \in M\}.$$

Exemple 1.2.15

Soient $A = \{a, b, c\}$, $\Gamma = \{\mu, \nu\}$ et f le morphisme défini par $f(a) = \mu$, $f(b) = \nu$, $f(c) = \nu$.

Si $L = \{ab, bc, cb, aaab, aaac\}$, alors $f(L) = \{\mu\nu, \nu\nu, \mu\mu\mu\nu\}$.

Si $M = \{\mu\nu, \nu\mu, \nu\mu\nu\}$, alors $f^{-1}(M) = \{ab, ac, ba, ca, bab, bac, cab, cac\}$.

1.2.3 Equation associée aux langages (lemme d'Arden)

Lemme 1.2.1 *[10, p.36] Soient A et B deux langages sur Σ, et l'équation $Y = AY + B$ d'inconnue Y à valeur dans 2^{Σ^*}. Alors :*

(1) $Y = A^*B$ est la solution minimale de l'équation.

(2) Si $\varepsilon \notin A$, $Y = A^*B$ est l'unique solution .

(3) Si $\varepsilon \in A$ alors pour tout $C \subseteq \Sigma^*$, $Y = A^*B + A^*C$ est aussi solution.

La démonstration de ce lemme dans la partie 2.2 [voir chapitre 2 page (28)].

Chapitre 2

Langages rationnels et les automates finis

Ce chapitre contient les définitions et notions usuelles de la théorie des automates qui seront nécessaires pour la suite de mémoire. Dans une première partie nous présenterons les langages rationnels, les expressions rationnelles. On introduit par la suite dans la deuxième partie, les automates finis qui permet de représenter d'une manière fini certaines ensembles (les ensembles rationnels) du monoïde libre A^*, nous finirons en rappelant théorème de Kleene qui affirme l'ensemble des langages acceptés par automate fini coïncide avec l'ensemble des langages rationnels. La dernière section de ce chapitre est consacrée à l'étude de l'automate minimale et son monoïde syntaxique.

2.1 Langages rationnels

Parmi les opérations définies dans $P(A^*)$ à la section 1.2, trois sont distinguées et sont qualifiées de rationnelles : il s'agit de l'union, de la concaténation et de l'étoile. Par contraire, notez que la complémentation et l'intersection ne sont pas des opérations rationnelles. Cette distinction permet de définir une famille importante de langages : les langages rationnels. (Pour plus de détails Voir [4] et [9]).

▶ Commençons par rappeler la définition de ces opérations rationnelles :

On peut définir deux opérations binaires et une opération unaire sur les langages :

• L'union des langages est définie comme d'habitude (union ensembliste).

• La concaténation des langages est défini de la manière suivante :

$$L_1.L_2 = \{uv \ / \ u \in L_1 \ et \ v \in L_2\}.$$

• L'étoile (ou fermeture de kleene) d'un langage est définie ainsi : $A^* = \cup_{n\geq 0} A^n$.

Définition 2.1.1

Un langage rationnel sur un alphabet A est un sous-ensemble de A^* défini inductivement de la façon suivante :

• *Le langage vide $\emptyset$ et $\{\varepsilon\}$ (le langage composé du mot vide) sont des langages rationnels.*

• *Pour tout $a \in A$, le singleton $\{a\}$ est un langage rationnel.*

• *Si L_1 et L_2 sont des langages rationnels, alors $L_1 \cup L_2$, L_1L_2, L_1^* sont également des langages rationnels.*

Définition 2.1.2

On appelle ensemble des langages rationnels (ou réguliers) sur l'alphabet A le plus petit ensemble de langages contenant $\emptyset$ et $\{a\}$ pour $a \in A$ et stable pour les opérations rationnelles (l'union, le produit et l'étoile) et l'on note $Rat(A^*)$; Autrement dit, $Rat(A^*)$ est le plus petit sous-ensemble de $P(A^*)$.

Remarque 2.1.1 *Tous les langages finis sont rationnels; A^* est rationnel.*

Exemple 2.1.1

Quelques exemples de langages rationnels :

• Le langage $\{1\}$ est rationnel car il s'écrit $\emptyset^*$.

• Le langage A est rationnel puisqu'il s'écrit $A = \cup_{a\in A} \{a\}$.

• Le langage L des mots de longueur paire est rationnel puisqu'il s'écrit :

$$L = (AA)^* = (A^2)^*.$$

• Le langage L' des mots de longueur impaire est rationnel puisqu'il s'écrit $L' = AL$.

Soit $L = \{a^p / \ p$ premier$\}$ un langage n'est pas rationnel.

2.1.1 Expression rationnelle

Définition 2.1.3

Soit A un alphabet. Supposons que 0, e, $+$, $.$, $(,)$, $*$ sont des symboles n'appartenant pas à A. L'ensemble $\Re_A$ des expressions régulières sur A est défini récursivement par :

▶ 0 *appartient à* $\Re_A$.

▶ *pour tout* $a \in A$, a *appartient à* $\Re_A$.

▶ *si* E *et* F *appartiennent à* $\Re_A$ *, alors :*

• $(E + F)$ *appartient à* $\Re_A$.

• (EF) *appartient à* $\Re_A$.

• E^* *appartient à* $\Re_A$.

Exemple 2.1.2

Si $A = \{a, b\}$, voici quelques exemples d'expressions régulières :

$\alpha_1 = (e + (a.b))$,

$\alpha_2 = (((a.b).a) + b^*)^*$,

$\alpha_3 = ((a + b)^*.(ab))$.

Définition 2.1.4

L'expression rationnelle est la plus petite famille d'expression Γ *telle que :*

▶ $\{a\} \in \Gamma$ *pour toute lettre* a.

▶ *Pour tout couple d'expressions* (E, E') *de* Γ*, l'expressions* $E + E'$, EE' *et* E^* *sont encore dans* Γ.

Exemple 2.1.3

Quelques exemples d'expressions rationnelles

• A^* tous les mots.

• aA^* mots commençant par a.

• A^*a mots finissant par a.

• $a^* + b^*$ mots n'ayant que des a ou que des b.

• $(aa + b)^*$ mots avec des blocs de a de longueur paire.

• $(ab^*a + b)^*$ mots ayant un nombre pair de a.

Définition 2.1.5

On définit alors une application $\mathcal{L} : \Re_A \to 2^{A^*}$ *qui à toute expression rationnelle associe le langage rationnel décrit par l'expression :*

- ▶ $\mathcal{L}(0) = \emptyset$, $\mathcal{L}(e) = \{1\}$.
- ▶ *si* $a \in A$, *alors* $\mathcal{L}(a) = \{a\}$.
- ▶ *si* E *et* F *sont des expressions régulières,*
- $\mathcal{L}[(E + F)] = \mathcal{L}(E) + \mathcal{L}(F)$.
- $\mathcal{L}[(EF)] = \mathcal{L}(E)\mathcal{L}(F)$.
- $\mathcal{L}(E^*) = (\mathcal{L}(E))^*$.

Exemple 2.1.4

D'aprés l'exemple au dessus (2) On a:

$\mathcal{L}(\alpha_1) = \{1, ab\}$.

$\mathcal{L}(\alpha_2) = (\{aba\} \cup \{b\}^*)^*$.

$\mathcal{L}(\alpha_3) = \{a, b\}^* \{ab\}$.

Identités rationnelles Soit R, S, T expressions régulières. On a les formules suivant qui expriment (par le signe =) un certain nombre d'équivalences élémentaires. (Voir [1], p. 65 et 66).

1. $R + R = R$, $R + \emptyset = R$.
2. $R + S = S + R$.
3. $(R + S) + T = R + (S + T)$.
4. $(RS)T = R(ST)$.
5. $R\varepsilon = \varepsilon R = R$, $R\emptyset = \emptyset R = \emptyset$.
6. $(R + S)T = RT + ST$.
7. $T(R + S) = TR + TS$.
8. $R^*R^* = R^*$, $(R^*)^* = R\ ^*$.

Définition 2.1.6

Un langage L sur A est régulier s'il existe une expression régulière $E \in \Re_A$ telle que : $L = \mathcal{L}(E)$, si E et F sont deux expressions rationnelles telles que $\mathcal{L}(E) = \mathcal{L}(F)$, alors on dit que et sont équivalentes.

Exemple 2.1.5

Soit $A = \{a, b\}$. L'ensemble L de tous les mots de A^* qui se terminent par la lettre a, peut être représenté par l'expression rationnelle $L = (a + b)^*a$.

▶ Cependant, plusieurs expressions rationnelles distinctes peuvent décrire un même langage rationnel. Ainsi l'ensemble des mots qui s'écrivent sur l'alphabet $A = \{a, b\}$ peut être décrit par les expressions $(a + b)^*$ et $(a^*b)^*a^*$.

▶ De façon plus générale, quelles que soient les expressions e et f, les interprétations des expressions rationnelles associées à $(e + f)^*$ et $(e^*f)^*e^*$ sont égales, cette relation est appelée une identité et est alors notée $(e + f)^* \equiv (e^*f)^*e^*$.

Cette propriété est la source du problème de la hauteur d'étoile qui sera explicité dans ce qui suit.

Proposition 2.1.1 *[8. p.17]*

L'ensemble $\mathcal{L}(\Re_A)$ des langages réguliers sur A est la plus petite famille de langages contenant le langage vide, les langages $\{a\}$ réduits à une lettre ($a \in A$) et qui est stable pour les opérations d'union, de concaténation et d'étoile de Kleene.

2.2 Automates finis et langages reconnaissables

Les automates finis sont des « machines abstraites » qui savent reconnaître l'appartenance ou la non-appartenance d'un mot à un langage régulier donné. (Voir [8] et [9]).

2.2.1 Définitions et représentations

Définition 2.2.1

Un automate fini (AF) est défini par 5-uplet $\mathcal{A} = (Q, I, F, A, \delta)$ *où,*

- A *est un ensemble fini, l'alphabet d'entrée.*
- Q *est un ensemble fini, l'ensemble fini des états.*
- $I \subseteq Q$ *est l'ensemble des états initiaux.*
- $F \subseteq Q$ *est l'ensemble des états acceptants (finaux) .*
- $\delta \subseteq (Q \times A \times Q)$ *est l'ensemble des transitions.*

Définition 2.2.2

Un chemin est un chemin fini dans le graphe;

▶ *un chemin acceptant est un chemin dont l'origine est dans* I *et l'(autre) extrémité est dans* F*.*

▶ *l'étiquette d'un chemin est la concaténation des étiquettes des transitions qui forment le chemin.*

▶ *un mot est accepté ou reconnu par l'automate* $\mathcal{A}$ *si c'est l'étiquette d'un chemin acceptant dans* $\mathcal{A}$*.*

Plus formellement, et pour introduire quelques notations, un mot $u = u_1 \dots u_n$ *est accepté par* $\mathcal{A} = (Q, q_0, F, A, \delta)$ *s'il existe une suite d'états,*

$q_1, \dots, q_n \in Q$ *tels que* $\forall i \in \{1, \dots, n\}$,

$$q_i \in \delta(q_{i-1}, u_i) \text{ et } q_n \in F.$$

Définition 2.2.3

Un automate fini déterministe (ou AFD) est la donnée d'un 5-uple $\mathcal{A} = (Q, q_0, F, A, \delta)$ *où,*

- Q *est un ensemble fini dont les éléments sont les états de* $\mathcal{A}$.
- $q_0 \in Q$ *est un état privilégié appelé état initial.*
- $F \subseteq Q$ *désigne l'ensemble des états finals.*
- A *est l'alphabet de l'automate.*
- $\delta : Q \times A \to Q$ *est la fonction de transition de* $\mathcal{A}$.

► *Nous supposerons que* δ *est une fonction totale, i.e., que* δ *est définie pour tout couple* $(q, a) \in Q \times A$, $\exists q' \in Q : \delta(q, a) = q'$ *(on parle alors AFD complet).*

► *Nous représentons un AFD par deux manière suivante :*

- *Un diagramme de transition représentant* $\mathcal{A}$ *est défini comme suit :*

Les états de $\mathcal{A}$ *sont les sommets d'un graphe orienté et sont représentés par des cercles.*

Si $\delta(q, a) = q'$, $q, q' \in Q$, $a \in A$, *alors on trace un arc orienté de* q *vers* q' *et de label* a, $q \xrightarrow{a} q'$.

Les états finales sont repérés grâce à un double cercle ou par une flèche sortante et l'état initial est désigné par une flèche entrante sans label. Enfin si deux lettres a *et* a' *sont telles que* $\delta(q, a) = q'$ *et* $\delta(q, a') = q'$, *on s'autoriser un unique arc portant deux labels séparés par une virgule* $q \xrightarrow{a,\, a'} q'$. *Cette convention s'adapte aisément à plus de deux lettres.*

- *Table de transition : table représentant l'action de* δ*; les rangées correspondent aux états, les colonnes aux symboles d'entrée.*

L'état de départ est reconnu grâce à une flèche, les états acceptants grâce à une étoile.

Exemple 2.2.1

L'automate $\mathcal{A}$= (Q, q_0, F, A, δ) où $Q = \{1, 2, 3\}$, $q_0 = 1$, $F = \{1, 2\}$ $A = \{a, b\}$ et où la fonction de transition est donnée par :

	a	b
$* \longrightarrow 1$	1	2
$*2$	1	3
3	3	2

est représenté à la figure **2.1**

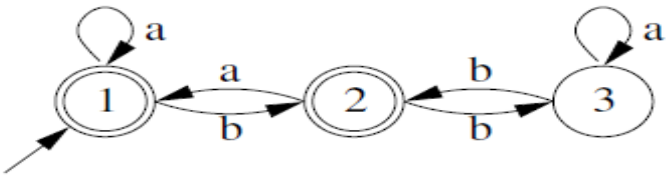


Figure .2.1. Un AFD

Exemple 2.2.2

Tous les nombres divisible par 3 froment un ensemble infini. Dans cet exemple, 0 est un état à la fois initial et final ($q_0 = 0$ et $F = \{0\}$) et chaque état correspond à une valeur du reste dans la division par 3 ($Q = \{0, 1, 2\}$) avec l'alphabet binaire $A = \{0, 1\}$. Donc, on va représenter cet ensemble par automate fini : (voir la figure 2.2).

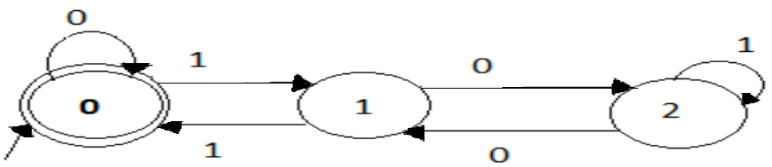


Figure .2.2

Donc, on peut lire les nombres qui sont divisible par 3 de l'automate comme : 101101, 1110101, ...

Exemple 2.2.3

Soit $A = \{a, b\}$ un alphabet, on a :

- $Q = \{1, 2\}$.
- $q_0 = 1$, état noté avec une petite flèche entrante.
- $F = \{2\}$, état noté avec deux cercles.

$\delta : Q \times A \to Q$

$(1, a) \to 2$

$(1, b) \to 1$

$(2, b) \to 1$

Cet automate n'est pas complet, car $\delta(2, a)$ n'est pas défini (voir la figure 2.3) :

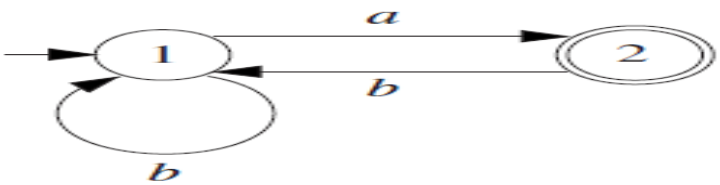


Figure .2.3

Définition 2.2.4

Soit $\mathcal{A} = (Q,\ q_0,\ F,\ A,\ \delta)$ *un automate fini déterministe. On étend naturellement la fonction de transition* δ *à* $Q \times A^*$ *de la manière suivante :*

$$\delta(q, 1) = q.$$

$$\delta(q, aw) = \delta(\delta(q, a), w), a \in A, w \in A^*.$$

Le langage accepté par $\mathcal{A}$ *est alors,*

$$L(\mathcal{A}) = \{w \in A^* : \delta(q_0, w) \in F\}.$$

Si $w \in L(\mathcal{A})$*, on dit encore que* $\mathcal{A}$ *accepté le mot* w *(ou que* w *est accepté par* $\mathcal{A}$*). Ainsi, le rôle fondamental d'un automate est d'accepter ou de rejeter des mots.*

Définition 2.2.5

$\mathcal{A} = (Q,\ q_0,\ F,\ A,\ \delta)$ *est un automate fini non déterministe (AFND);*

seule différence :

δ *fonction de transition* $\delta : Q \times A \longrightarrow 2^Q$.

La fonction δ *étendue aux mots :*

▶ $\delta(q,\ 1) = q$, *pour tout* $q \in Q$.

▶ $\delta(q,\ wa) = \{p : \exists r \in \delta(q,\ w)$ *et* $p \in \delta(r,\ a)\}$ *pour tout* $q \in Q$, $w \in A^*$, $a \in A$.

les deux coïncide sur les lettres.

▶ $\delta(p,\ w) = \cup\delta(q,\ w)$.

▶ *Le langage associé à* $\mathcal{A}$ *noté* $L(\mathcal{A})$ *est défini par* $L(\mathcal{A}) = \{w\ /\ \delta(q_0,\ w) \cap F \neq \emptyset\}$.

Proposition 2.2.1 *Tout langage accepté par un AFN, alors il est accepté par un AFD.*

Preuve.

Donnée : un AFN$\langle N\rangle = (Q, q_0, F, A, \delta)$ accepté le langage L.

Résultat : un AFD$\langle M\rangle$ accepté le même langage L.

Les états de M sont des ensembles d'états de N.

L'état initial de M est $\{q_0\}$.

Si P est un ensemble d'états de N, on définit $\delta(P, a) = \cup_{p\in P}\{q : q \in \delta(p, a)\}$. ■

Exemple 2.2.4

Considérons l'automate fini non déterministe $N = (Q, q_0, F, A, \delta)$;

- $Q = \{0, 1, 2\}$.
- $A = \{a, b\}$, $q_0 = \{0\}$, $F = \{2\}$.
- $\delta(0, a) = 0$, $\delta(0, b) = 0$, $\delta(0, a) = 1$, $\delta(1, b) = 2$, $\delta(2, a) = 2$, $\delta(2, b) = 2$.

Il accepte le langage A^*abA^*des mots qui contiennent au moins un facteur ab.

L'automate fini déterministe M tel que $L(M) = L(N)$ est définit par :

$$M = (B(Q), q_0, F, A, \delta).$$

- $A = \{a, b\}$.
- $q_0 = \{q_0\} = \{0\}$.
- $F = \{P \subseteq Q \mid P \cap F \neq \emptyset\}$.
- $\delta(p, x) = P'$ tel que $P' = \{q \mid \exists p \in P\ p \xrightarrow{x} q\}, x \in \{a, b\}$.

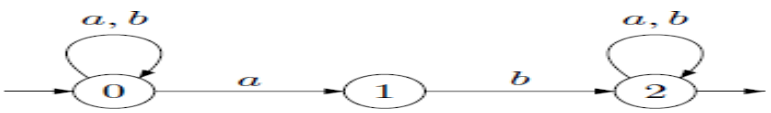
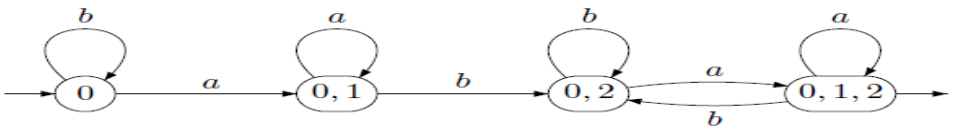


Figure .2.4. Automate non déterministe



Déterminisation de l'automate de **la figure.2.4**

Définition 2.2.6

On dit qu'un langage est reconnaissable par un AF s'il existe un automate fini qui le reconnaît. On note Rec(A^*) l'ensemble des langages sur l'alphabet A reconnaissables par un automate fini.

Proposition 2.2.2

1. *$Rec(A^*)$ est fermée par produit de concaténation.*
2. *$Rec(A^*)$ est fermée par étoile.*
3. *$Rec(A^*)$ est fermée par union.*

▶ Le théorème suivant énonce l'équivalence du pouvoir de description des expressions rationnelles et des automates finis.

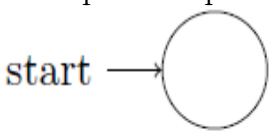
Théorème 2.2.1 (Théorème de Kleene) *[3] Soit A un alphabet. Un langage de A^* est rationnel si et seulement si il est reconnu par un automate fini sur A :*

$$Rat(A^*) = Rec(A^*).$$

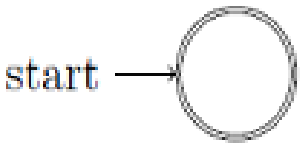
Démonstration. Montrons la double inclusion. ■

$Rat(A^*) \subseteq Rec(A^*)$:

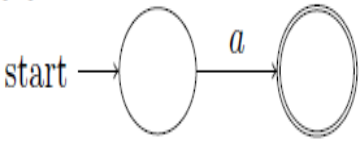
- $\emptyset$ est représenté par :



- $\{\varepsilon\}$ est représenté par :



- $\{a\}$ $\forall a \in A$ est représenté par :



$Rec(A^*)$ est stable par $\cup$, ., * et $Rat(A^*)$ est la plus petite classe vérifiant ces propriétés.

$Rec(A^*) \subseteq Rat(A^*)$:

Réciproquement, à partir d'un AFD, peut on construire une expression rationnelle qui décrit le langage reconnu par $\mathcal{A}$.

▶ Les systèmes d'équations linéaires à droite permettent de ramener le calcul d'une expression rationnelle à la résolution d'un système d'équations.

▶ A chaque état q on associe une expression régulière Y_q dénotant le langage $L_q(\mathcal{A})$ associe à cet état.

▶ On obtient un système d'équations dont les inconnues sont des expressions régulières dénotant des langages.

▶ Si q est l'état initial, Y_q décrit le langage reconnu par $\mathcal{A}$.

Exemple 2.2.5

Soit le système d'équations suivantes :

$$\begin{cases} Y_1 = bY_2 + aY_3 \\ Y_2 = bY_1 + aY_4 \\ Y_3 = \varepsilon + aY_4 + bY_2 \\ Y_4 = \varepsilon + (a+b)Y_4 \end{cases}$$

Résoudre un tel système revient à calculer Y_1, car il est associé à l'état initial 1.

▶ On procédé par substitutions et on va avoir de résoudre l'équation : $Y = AY + B$ (lemme d'Arden).

• A^*B est bien solution : $A(A^*B) + B = A^+B + B = (A^+ + \{\varepsilon\})B = A^*B$.

▶ Si Y est une solution, alors $A^*B \subseteq Y$:

• Récurrence sur la hauteur d'étoile :

- Si $i = 0$, $A^0B = B \subseteq Y$ car $Y = AY + B$.

• Par hypothèse de récurrence :

- pour $i = n$: $A^nB \subseteq Y$.

- pour $i = n + 1$: $A^{n+1}B = AA^nB \subseteq AY \subseteq AY + B = Y$, cqfd.

▶ Si $\varepsilon \notin A$ alors A^*B est l'unique solution.

On suppose la non-unicité de la solution A^*B. Soit X un autre solution et soit un mot w de longueur minimale tel que $w \in X \setminus A^*B$.

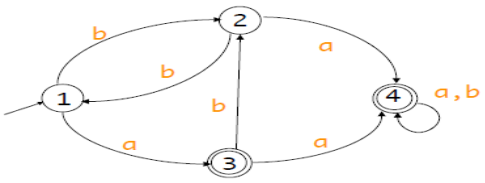
$w \in X = AX + B$ et $w \notin B$ donc $w = uv$ avec $u \in A$ $(u \neq \varepsilon)$ et $v \in X$. Or $v \notin A^*B$ (sinon w aussi) donc $v \in X \setminus A^*B$.

Contradiction : la longueur de w n'était donc pas minimale.

▶ Si $\varepsilon \in A$ alors pour tout $C \subseteq \Sigma^*$, $Y = A^*B + A^*C$ est aussi solution :

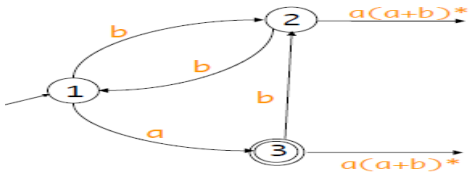
$A(A^*B + A^*C) + B = A^+B + A^+C + B = A^*B + A^+C = A^*B + A^*C.$

▶ Si $\varepsilon \notin A$ alors A^*B est l'unique solution de $Y = AY + B$.



(1)
$$\begin{cases} Y_1 &= bY_2 + aY_3 \\ Y_2 &= bY_1 + aY_4 \\ Y_3 &= \varepsilon + aY_4 + bY_2 \\ Y_4 &= \varepsilon + (a+b)Y_4 \end{cases}$$

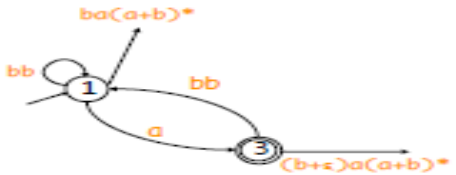
$\Longrightarrow Y_4 = (a+b)^*\varepsilon = (a+b)^*$ [Y_4 est alors éliminé].



(2)
$$\begin{cases} Y_1 &= bY_2 &+ & aY_3 \\ Y_2 &= bY_1 &+ & a(a+b)^* \\ Y_3 &= \varepsilon &+ & a(a+b)^* + bY_2 \end{cases}$$

$$\Longrightarrow \begin{cases} Y_1 &= bbY_1 + ba(a+b)^* + aY_3 \\ Y_3 &= \varepsilon + a(a+b)^* + bbY_1 + ba(a+b)^* \end{cases}$$

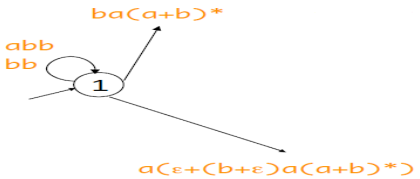
[Y_2 est alors éliminé].



(3)
$$\begin{cases} Y_1 &= bbY_1 &+ & ba(a+b)^* + aY_3 \\ Y_3 &= \varepsilon &+ & bbY_1 + (b+\varepsilon)a(a+b)^* \end{cases}$$

$\Longrightarrow Y_1 = bbY_1 + ba(a+b)^* + a\varepsilon + abbY_1 + a(b+\varepsilon)a(a+b)^*$ [Y_3 est alors éliminé].

▶ Si $\varepsilon \notin A$ alors A^*B est l'unique solution de $Y = AY + B$.



(4) $Y_1 = (abb+bb)Y_1 + ba(a+b)^* + a(\varepsilon + (b+\varepsilon)a(a+b)^*).$

$\Longrightarrow Y_1 = (abb + bb)^* + (ba(a+b)^* + a(\varepsilon + (b+\varepsilon)a(a+b)^*)).$

Conséquence : le théorème de Kleene

Ce théorème dit que l'ensemble des langages rationnels, qui sont définis à partir des singletons, et des unions, produits et étoiles de langages rationnels, coïncide avec l'ensemble des langages reconnus par des automates finis.

2.3 Automate minimal et monoïde syntaxique

Dans cette partie, on montre que tout langage rationnel est accepté par un automate minimal qui est le quotient de tout automate déterministe acceptant ce langage, en suite nous terminons par le monoïde syntaxique. (Voir [10] et [6] et [3]).

2.3.1 Quotient d'un language

Les quotients à gauche constituent un outil indispensable à l'étude des automates déterministes acceptant un langage L puisque le nombre de quotients donne un minorant du nombre d'états. Ils fournissent également une caractérisation très utile des langages rationnels (voir proposition 2.3.2).

Définition 2.3.1

Soit $L \subseteq A^*$ un langage. Le quotient à gauche (ou résiduel) de L par un mot $u \in A^*$ est le langage :

$$u^{-1}L = \{v \in A^* \mid uv \in L\}.$$

De manière symétrique, on peut aussi définir les quotients à droite d'un langage.

Proposition 2.3.1 *Pour toute lettre a, tous mots u, v et w et tous langages K et L, on a les relations suivantes :*

1. $u^{-1}(K + L) = u^{-1}K + u^{-1}L$.
2. $a^{-1}(KL) = (a^{-1}K)L + \varepsilon(a^{-1}K)L$.
3. $a^{-1}(L^*) = (a^{-1}L)L^*$ *et plus généralement* $w^{-1}L^* = \sum_{w=uv} uv = \varepsilon(u^{-1}L^*)(v^{-1})LL^*$.
4. $(uv)^{-1}L = v^{-1}(u^{-1}L)$.

Les relations précédentes permettent de calculer les quotients à gauche.

▶ Le lemme suivant établit le lien fondamental entre les quotients à gauche d'un langage et les automates déterministes acceptant ce même langage.

Lemme 2.3.1 *Soit $\mathcal{A} = (Q, A, \delta, q_0, F)$ un automate déterministe, q un état de Q, et u un mot de A^*. Si $\delta(q_0, u) = q$, alors $L_q = u^{-1}L$.*

Preuve. Par récurrence sur $|u|$: initialement, $L_{q_0} = L = \varepsilon^{-1}L$, puis pour $u = va$ avec v dans A^* et a dans A, et $q_0 \xrightarrow{v} q' \xrightarrow{a} q$, par hypothèse de récurrence $L_{q'} = v^{-1}L$ et on vérifie bien $L_q = a^{-1}L_{q'} = a^{-1}v^{-1}L = (va)^{-1}L$ puisque l'automate est déterministe. ■

Proposition 2.3.2 *Un langage L est rationnel si et seulement si il a un nombre fini de quotients à gauche.*

Pour la preuve de la proposition, on introduit la définition suivante.

Définition 2.3.2

Soit L un langage rationnel. L'automate minimal de L est automate :

$$\mathcal{A}_L = (Q_L, A, \delta_L, I, F_L).$$

où,

1) $Q_L = \{u^{-1}L \mid u \in A^*\}$.

2) $I = \{L\}$.

3) $F_L = \{u^{-1}L \mid u \in L\}$.

4) $\delta_L = \{(u^{-1}L, a, (ua)^{-1}L) \mid u \in A^* \text{ et } a \in A\}$.

Exemple 2.3.1

Soit le langage $L = (ab)^*$. Ses différents quotients à gauche sont les langages suivants : $a^{-1}L = bL$, $b^{-1}L = \emptyset$, $a^{-1}(bL) = \emptyset$ et $b^{-1}(bL) = L$. En appliquant la construction donnée dans la preuve de la proposition précédente, on obtient l'automate minimal de L de la figure 2.5.

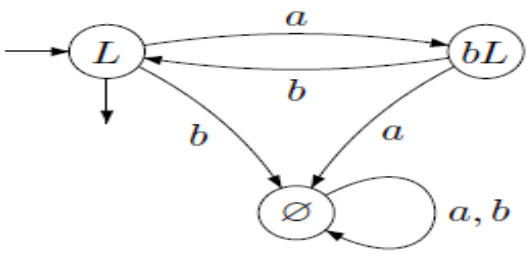


Figure .2.5. Automate minimal de $L=(ab)^*$

Le nombre d'états de l'automate minimal est bien sûr minimal puisqu'il est égal au nombre de quotients à gauche.

Lemme 2.3.2 *L'automate minimal $\mathcal{A}_L$ accepte le langage L.*

Preuve. On montre facilement par récurrence sur la longueur que pour tout mot u, on a un chemin $L \xrightarrow{u} u^{-1}L$ dans l'automate $\mathcal{A}_L$. La définition des états finaux donne immédiatement le résultat. ■

2.3.2 Congruence de Nerode

Définition 2.3.3

Soit $\mathcal{A} = (Q, A, \delta, q_0, F)$ un automate déterministe et complet, et $\sim$ une relation d'équivalence sur Q. Cette relation est une congruence si elle est compatible avec les transitions de $\mathcal{A}$:

$$q \sim q' \implies \forall a \in A,\ \delta(q,a) \sim \delta(q',a).$$

$$q \sim q' \implies (q \in F \iff q' \in F).$$

pour tous états q, q' de Q.

La seconde propriété signifie que chaque classe d'une congruence ne contient que des états finaux ou que des états qui ne sont pas finaux.

Définition 2.3.4

Soit $\mathcal{A} = (Q, A, \delta, q_0, F)$ un automate déterministe et $\sim$ une relation d'équivalence sur Q. L'automate quotient de $\mathcal{A}$ par $\sim$ est l'automate $\mathcal{A}/\sim = (Q/\sim, A, \delta/\sim, [q_0], \{[q_f] | q_f \in F\})$ avec $\delta/\sim = \{([q], a, [\delta(q,a)])\ | q \in Q$ et $a \in A\}$. Cet automate est encore déterministe car la classe $[\delta(q,a)]$ ne dépend que de la classe de q.

Lemme 2.3.3 *Soit $\sim$ une congruence sur un automate $\mathcal{A}$, alors l'automate quotient $\mathcal{A}/\sim$ accepte le langage $L(\mathcal{A})$.*

Preuve. Soit un chemin :

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n.$$

d'étiquette $w = a_1...a_n$ dans l'automate $\mathcal{A}$. On en déduit que :

$$[q_0] \xrightarrow{a_1} [q_1] \xrightarrow{a_2} \cdots \xrightarrow{a_n} [q_n].$$

est un chemin étiqueté par w dans $\mathcal{A}/\sim$. Si de plus q_0 est initial et q_n est final dans $\mathcal{A}$, alors $[q_0]$ est initial et $[q_n]$ est final dans $\mathcal{A}/\sim$. Ceci prouve l'inclusion $L(\mathcal{A}) \subset L(\mathcal{A}/\sim)$.

Soit maintenant un chemin :

$$[q_0] \xrightarrow{a_1} [q_1] \xrightarrow{a_2} \cdots \xrightarrow{a_n} [q_n].$$

d'étiquette $w = a_1...a_n$ dans l'automate $\mathcal{A}/\sim$. On montre par récurrence sur n que pour tout état q'_0 de la classe $[q_0]$, il existe un chemin :

$$q'_0 \xrightarrow{a_1} q'_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q'_n.$$

d'étiquette w tel que $q'_i \sim q_i$ pour tout $0 \leq i \leq n$. Si le chemin dans $\mathcal{A}/\sim$ est acceptant, on peut choisir $q'_0 = i$ et l'état q'_n est alors final. Ceci montre l'inclusion $L(\mathcal{A}/\sim) \subset L(\mathcal{A})$. ■

▶ On introduit maintenant la congruence de Nerode qui permet le calcul de l'automate minimal d'un langage à partir de n'importe quel automate déterministe le reconnaissant.

Définition 2.3.5 *Soit $\mathcal{A} = (Q, A, \delta, q_0, F)$ un automate déterministe et complet. La congruence de Nerode est définie pour tous états q et q' par :*

$$q \sim q' \iff L_q = L_{q'}.$$

aussi définit comme suivant :

$$q \sim q' \overset{déf}{\iff} \forall w \ (q.w \in F \overset{déf}{\iff} q'.w \in F).$$

Notons $[q]$ pour la classe d'équivalence de l'état q.

▶ La proposition suivante donne comment obtenir l'automate minimal à partir de n'importe quel automate déterministe.

Proposition 2.3.3 *Soit $\mathcal{A}$ un automate déterministe et complet acceptant un langage L. L'automate minimal $\mathcal{A}_L$ est égal à $\mathcal{A}/\sim$ où $\sim$ est la congruence de Nerode de $\mathcal{A}$.*

Preuve. Pour un état q, on note L_q le langage $\{w \mid q.w \in F\}$. Deux états q et q' vérifient $q \sim q'$ si et seulement si $L_q = L_{q'}$. D'après le lemme 2.3.1, chaque langage L_q est de la forme $u^{-1}L$ pour un mot u qui étiquette un chemin de l'état initial à q. On peut donc identifier les classes de la congruence de Nerode avec les quotients à gauche de L. ■

2.3.3 Monoïde syntaxique

A tout automate déterministe, on peut associer un monoïde de la manière suivante.

▶ Soit $\mathcal{A} = (Q, A, \delta, q_0, F)$ un automate déterministe et soit Γ le monoïde des fonctions partielles de Q dans Q, la loi de composition étant définie par :

$$\forall q \in Q,\ \forall \gamma_1,\ \gamma_2 \in \Gamma,\ q(\gamma_1\gamma_2) = (q\gamma_1)\gamma_2.$$

▶ Soit ϕ la fonction qui à tout mot w de A^* associe la fonction partielle $\phi(w)$ de Q dans Q définie par :

$$q\phi(w) = q.w.$$

Alors ϕ est un morphisme de A^* dans le monoïde des fonctions partielles de Q. Le sous monoïde $\phi(A^*)$ est appelé le monoïde des transitions de l'automate $\mathcal{A}$.

▶ On introduit la notion de monoïde syntaxique d'un langage. Il est isomorphe au monoïde des transitions de l'automate minimal (Proposition 2.3.4).

Définition 2.3.6 *[10]*

Pour tout langage $L \subset A^*$, on appelle contexte de $w \in A^*$ l'ensemble :

$$C_L(w) = \{(u, v) \in A^* \times A^* | uwv \in L\}.$$

La relation $\equiv_L$ définie par :

$$w \equiv_L w' \Leftrightarrow C_L(w) = C_L(w') \Leftrightarrow \forall u, v \in A^*(uwv \in L \Leftrightarrow uw'v \in L).$$

Cette dernière est appelée congruence syntaxique de L. Autrement dit, deux mots sont équivalents s'ils ont mêmes contextes dans L. Comme une congruence est une relation d'équivalence compatible avec la structure de monoïde, le monoïde quotient de A^* par $\equiv_L$ est appelé le monoïde syntaxique de L.

Proposition 2.3.4 *Soit $L \subset A^*$. Le monoïde syntaxique d'un langage L est isomorphe au monoïde des transitions de l'automate minimal de L.*

Chapitre 3

Etude du lien entre la hauteur d'étoile d'un langage rationnel et l'automate fini qui reconnait ce langage

En 1963, Eggan a introduit la notion de hauteur d'étoile des expressions rationnelles définies à l'aide des opérateurs de l'union, de la concaténation et de l'étoile. Cette hauteur est égale au nombre d'étoiles superposées dans l'expression considérée.

Par définition, la hauteur d'étoile d'un langage rationnel est égale au plus petit nombre d'étoiles superposées dans une expression rationnelle qui le représente. Cependant, il peut exister une infinité d'expressions rationnelles qui décrivent un même langage rationnel.

La notion de hauteur d'étoile peut être interprétée en termes d'automates : la hauteur d'étoile d'un langage rationnel est égale au nombre minimal de cycles imbriqués dans un automate qui reconnaît ce langage. Mais, la hauteur d'étoile n'étant pas une propriété syntaxique, elle ne peut être calculée à partir de l'automate minimal.

Eggan a montré que, pour tout entier naturel k, il existe un langage rationnel de hauteur d'étoile k et a soulevé le problème de la calculabilité de la hauteur d'étoile. En 1966, Dejean et Schützenberger ont prouvé que, pour tout entier naturel k, il existe des langages rationnels de hauteur d'étoile exactement k sur un alphabet à deux lettres. En 1967, McNaughton a donné un algorithme pour déterminer la hauteur d'étoile d'un langage rationnel dont le monoïde syntaxique est un groupe. Enfin, Hashigushi a trouvé en 1982 un algorithme

permettant de décider si un langage rationnel est de hauteur d'étoile 1, et en 1989 un algorithme permettant de décider la hauteur d'étoile dans le cas général.(Pour plus de détails voir [5] et [10]).

3.1 La hauteur de l'étoile

Définition 3.1.1

La hauteur d'étoile h d'une expression rationnelle est définie sur l'ensemble des expressions rationnelles non nulles, comme une fonction à valeurs dans $\mathbb{N}$ par :

1) $h(\emptyset) = h(1) = 0$.

2) $\forall a \in A,\ h(a) = 0$.

3) $\forall \alpha, \beta \in \Re_A,\ h(\alpha + \beta) = \max(h(\alpha), h(\beta))$.

4) $\forall \alpha, \beta \in \Re_A,\ h(\alpha\beta) = \max(h(\alpha), h(\beta))$.

5) $\forall \alpha \in \Re_A,\ h(\alpha^*) = 1 + h(\alpha)$.

Exemple 3.1.1

D'aprés la définition ci-dessus on a:

$$\begin{aligned}
h(ab+ba) &= \max(h(ab), h(ba)) \\
&= \max(\max((h(a), h(b)), \max(h(b), h(a))) \\
&= \max(0,0) = 0, \text{ alors } h(ab+ba) = 0. \\
h((a+b)^*) &= 1 + h(a+b) \\
&= 1 + \max(h(a), h(b)) \\
&= 1 + \max(0,0) = 1, \text{ alors } h((a+b)^*) = 1. \\
h((a^*b)^*a^*) &= \max(h((a^*b)^*), h(a^*)) \\
&= \max((1 + h(a^*b)), (1 + h(a))) \\
&= \max((1 + \max(h(a^*), h(b))), 1) \\
&= \max((1 + \max(1,0)), 1) \\
&= \max((1+1), 1) \\
&= \max(2,1) = 2, \text{ alors } h((a^*b)^*a^*) = 2.
\end{aligned}$$

Il faut remarquer qu'un même langage peut être décrit par plusieurs expressions rationnelles ayant des hauteurs d'étoile différentes. Par exemple, les deux expressions $(a+b)^*$

et $(a^*b)^*a^*$ sont de hauteurs d'étoile 1 et 2 mais décrivent le même langage de tous les mots sur l'alphabet $\{a, b\}$ (d'aprés l'exemple 2.1.5).

Définition 3.1.2

La hauteur d'un langage rationnel L est le minimum des hauteurs des expressions régulières qui le décrivent :

$$h(L) = \min\{h(\alpha)/\ \mathcal{L}(\alpha) = L\}.$$

Exemple 3.1.2

1. L'ensemble des mots sur l'alphabet $\{a, b\}$ qui se terminent par la lettre a, représenté, entre autres, par les expressions rationnelles $(a+b)^*a$ et $(a^*b)^*a$, est de hauteur d'étoile 1.

2. $h(L) = 0$ si et seulement si L est fini.

3. Le langage $(a + b)^*$ est de hauteur d'étoile 1 puisqu'il est décrit par une expression rationnelle de hauteur 1 et qu'il ne peut être décrit par une expression de hauteur 0 parce qu'il est infini. En effet, $\mathcal{L}((a + b)^*) = \{(a + b)^i/\ i \geq 0\}$.

3.2 Le lien entre la hauteur de l'étoile d'un langage rationnel et les automates finis

Ce paragraphe présente une interprétation de la hauteur d'étoile en termes d'automates.

▶ On met en évidence dans ce qui suit le lien entre la notion de hauteur d'étoile d'un langage rationnel et celle de complexité en cycles du graphe des états des automates reconnaissant le langage.

▶ L'objet de ce qui suit est le suivant : on introduit la notion de rang cyclique d'un automate et on montre que la hauteur d'étoile d'un langage rationnel est égale au minimum des rangs cycliques des automates reconnaissant ce langage.

▶ On définit récursivement le rang d'une composante fortement connexe (est un sous graphe fortement connexe maximal) de la façon suivante :

• Une composante fortement connexe est de rang 0 si elle est réduite à un seul sommet et si le seul chemin qu'elle contient est le chemin associé au mot vide.

• Une composante fortement connexe est de rang 1 s'il existe un sommet par lequel passent tous les cycles de la composante fortement connexe.

• Une composante fortement connexe est de rang k s'il existe un sommet tel que le graphe obtenu en supprimant ce sommet et les transitions ayant pour extrémité ce sommet contient un cycle de rang $k-1$ et si le graphe obtenu, à partir du cycle initial, en supprimant tout autre sommet, contient un cycle de rang supérieur ou égal à $k-1$.

Définition 3.2.1

On définit la notion de rang cyclique d'un automate de la façon suivante : si le graphe des états de l'automate est acyclique (s'il ne contient aucune boucle), ce rang est nul, sinon il est égal au maximum des rangs de ses composantes fortement connexes.

Exemple 3.2.1

Les automates en pétales étant exactement les automates finis dont tous les cycles passent par un même sommet, ce sont donc les automates de rang cyclique 1. Les deux cycles de l'automate de la *(*figure 3.6*)* passent par l'état 1 : l'automate est de rang 1.

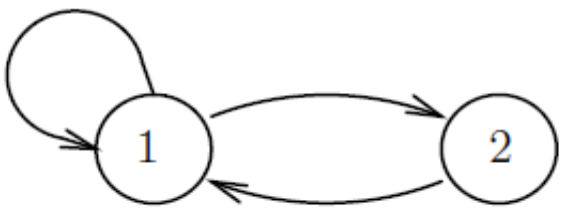


Figure .3.6. Automate de rang cyclique 1

Exemple 3.2.2

Automate de rang cyclique 2 *(*figure 3.7*)*. Du fait des propriétés de symétrie du graphe, tous les états jouent le même rôle. De plus, en supprimant l'un quelconque des états de l'automate, on obtient un automate en pétales (exemple 3.6). L'automate est donc de rang 2.

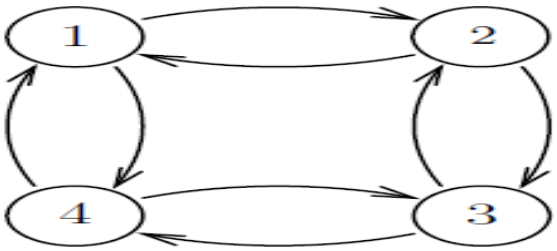


Figure .3.7. Automate de rang cyclique 2.

Théorème 3.2.1 (Eggan) *La hauteur d'étoile d'un langage rationnel est égal au minimum des rangs cycliques des automates finis reconnaissant ce langage.*

Démonstration. ■

Le minimum r des rangs cycliques des automates reconnaissant un langage rationnel L est inférieur à sa hauteur d'étoile. En effet, soit E une expression rationnelle représentant le langage L, en utilisant les méthodes usuelles pour construire un automate à partir d'une expression rationnelle, on crée une composante fortement connexe quand opère une étoile et on augmente éventuellement le rang d'une composante fortement connexe d'une unité si l'étoile opère sur une expression contenant déjà cet opérateur. La hauteur d'étoile de l'expression E est donc supérieure à r. Comme, par définition, la hauteur d'étoile d'un langage rationnel est égale au minimum des hauteurs d'étoile des expressions rationnelles décrivant ce langage, elle est supérieure au minimum des rangs cycliques des automates finis reconnaissant ce langage. Il reste à établir l'inégalité $r \geq h(L)$. On va prouver (*Lemme* 3.2.1) que si i et f sont deux états d'un automate de rang cyclique k, il existe une expression rationnelle de hauteur d'étoile au plus k qui décrit l'ensemble des étiquettes des chemins allant de l'état i à l'état f. Comme le langage L est reconnu par un automate de rang cyclique r, il existe une expression rationnelle de hauteur d'étoile au plus r qui décrit L. Cette expression est obtenue par union finie des étiquettes des chemins allant de i à f quand i et f décrivent respectivement l'ensemble des états initiaux et finaux de l'automate, ce qui montre que $h(L) \leq r$.

Lemme 3.2.1 *Si i et f sont deux états d'un automate $\mathcal{A}$ de rang cyclique k, il existe une expression rationnelle de hauteur au plus k qui décrit l'ensemble des étiquettes des chemins allant de l'état i à l'état f.*

Voir la démonstration de ce lemme dans [5].

► La hauteur d'étoile n'est pas une propriété syntaxique, il est facile de s'en convaincre en considérant un langage fini sur un alphabet à deux lettres : sa hauteur d'étoile est nulle mais celle de son complémentaire ne l'est pas alors qu'ils ont le même monoïde syntaxique.

► Une des conséquences de cette remarque est la suivante : l'automate minimal du langage ne donne qu'une borne supérieure de la hauteur d'étoile du langage (on rappelle que le monoïde de transition est isomorphe au monoïde syntaxique). De plus, selon le sens de lecture du mot, le rang cyclique de l'automate minimal associé peut être différent.

Exemple 3.2.3

On considère l'alphabet $A = \{a, b\}$ et L l'ensemble des mots de A^* qui se terminent par ab. Alors L n'est pas fini et est décrit par l'expression rationnelle $(a + b)^*ab$, il est donc de hauteur d'étoile 1. L'automate minimal *(*figure 3.8*)*, associé à une lecture de gauche à droite, est de rang cyclique 2.

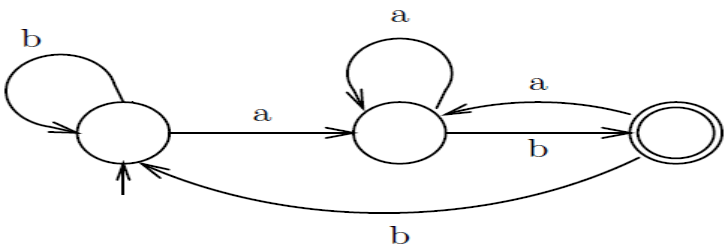


Figure.3.8. Automate minimal de rang cyclique 2

L'automate minimal (Figure 3.9), associé à une lecture de droite à gauche, est lui de rang cyclique 1.

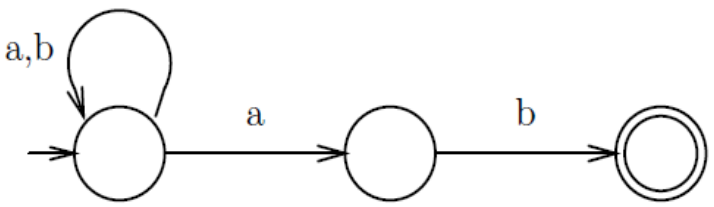


Figure.3.9. Automate minimal de rang cyclique 1

Théorème 3.2.2 (McNaughton) *Soit L un langage rationnel. Si le monoïde syntaxique de L est un groupe et si l'automate minimal de L a un seul état final, alors la hauteur d'étoile du langage L est égale au rang cyclique de cet automate.*

Exemple 3.2.4

Soit $A = \{a, b\}$ et soit $L = (a^* + ba^*b)^*$. On construit l'automate minimal de L (figure 3.10). Comme le monoïde syntaxique de L est isomorphe au monoïde des transitions de l'automate minimal, et que la lecture d'une lettre est une permutation des états de l'automate, le monoïde syntaxique est un groupe. On en déduit que la hauteur d'étoile de L est égale au rang cyclique de son automate minimal, c'est-à-dire 2.

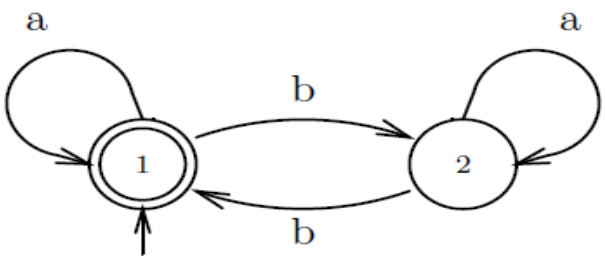


Figure .3.10. Automate minimal de L=$(a^*+ba^*b)^*$

Conclusion

Dans cette conclusion, nous résumons les principaux résultats obtenus lors de ce travail.

Nous avons présenté dans un premier temps les langage rationnels sont des parties de monoïde libre A^* et fermé pour les opérations rationnelles (l'union, la concaténation et l'étoile).

Ensuite nous avons présenté le théorème de Kleene qui établir l'égalité de la classe des langages reconnaissables (définis par les automates finis) et des langages rationnels (définis par des expressions rationnelles).

Finalement nous avons présenté :

- La hauteur d'étoile d'un langage rationnel est égale au plus petit nombre d'étoiles superposées dans une expression rationnelle qui le représente.

- La notion de hauteur d'étoile peut être interprétée en termes d'automates.

- Il existe une lien entre la hauteur d'étoile d'un langage rationnel et l'automate fini qui reconnait ce langage.

Bibliographie

[1] ABRAHAM GINZBURG. *Algebraic théorie of automata*, New York, London 1968.

[2] ANDRÉ ARNOLD. *Mathématiques pour l'informatique, Avec exercices corrigés*, Masson, Paris, 1992-1997.

[3] BENJAMIN MONMEGE ET SYLVAIN SCHMITZ. *Automates et langages*, Année 2011.

[4] FRANÇOIS YVON ET AKIM DEMAILLE. *Théorie des Langages Rationnels*, *Notes de cours, Travaux Dirigés et Travaux Pratiques Annales*, (2010).

[5] FRÉDERIQUE BASSIONO (1996). *Thèse de Séries rationnelles et distributions de longueures*; doctorat, Université de Marne-la-Vallée, 2012.

[6] JEAN-MICHEL AUTEBERT. *Théorie des langages et des automates*, Paris Milan Barcelone 1994.

[7] MAURICE NIVAT. *Éléments de la théorie générale des codes*, Université de Paris, Laboratoire de calcule Numérique, Année 1965-1966.

[8] MICHEL RIGO. *Théorie des automates et langages formels*, Université de liège. Année 2009-2010.

[9] NACER G. (2010). *Etude sur les groupes syntaxiques de petits degrés*; Magistère, Université de M'sila, 2007-2008.

[10] OLIVIER CARTON. *Langages formels, Calculabilité et Complexité*. Année 2007-2008.