

Dédicace

Je dédie ce modeste travail à ceux qui m'ont encouragé et soutenu moralement et matériellement pendant les moments les plus difficiles et durant toute ma vie , et qui me sont les plus chères sur cette planète : mon père et ma mère .

A tous mes amies

A tous ceux que j'aime

A tous les étudiants de ma promotion

Avec l'expression de tous mes sentiments de respect,

Je dédie ce mémoire.

Remerciements

*Merci avant tout au bon Dieu « ALLAH », le clément
le miséricordieux, le plus puissant.*

*En premier lieu j'adresse ma reconnaissance
à mon encadreur le Professeur :*

Mr. N. GHADBANE

*Son sérieux et sa compétence m'ont été très utiles
pour mener à bien ce travail.*

Je remercie les professeurs

Mr. D. MIHOUBI

Mr. L. MEBOUB

*qui m'ont faite l'honneur d'être membres
du jury.*

*Merci à tous les enseignants et les étudiants
de département mathématique
pour leurs aides judicieuses, les moyens qu'ils ont
met à notre disposition pour réaliser ce travail.*

*Enfin, je remercie toutes les personnes, famille, amis, qui
directement ou indirectement ont contribué
à la réalisation de ce travail.*

Table des matières

Notations	1
introduction	2
1 Notions élémentaires	4
1.1 Mots	4
1.2 Monoïde	7
1.3 Langage	9
1.4 Automate fini déterministe	13
2 Codes de longueurs variables	21
2.1 Code	21
2.2 Algorithme de reconnaissance des codes	25
2.3 Série génératrice	28
3 La représentation d'un code de longueur variable par un automate fini	33
3.1 Automate fini déterministe associé un mot de code	33
3.2 Représentation d'un code de longueur variable	34
Conclusion	38
Bibliographie	39

Notations

A : alphabet fini.

A^* : monoïde libre sur A .

$|w|$: la longueur du mot w .

$|w|_a$: le nombre d'occurrence de la lettre a dans le mot w .

L : langage sur l'alphabet A .

$\mathcal{P}(A^*)$: l'ensemble des langage sur A .

L^* : l'étoile (ou la fermeture) de kleene de L .

L^+ : l'étoile stricte de L .

$\text{Rat}(A^*)$: l'ensemble des langages rationnels (ou réguliers) sur A .

\mathcal{R}_A : l'ensemble des expressions rationnelles (ou régulières) sur A .

$\mathcal{A} = (Q, I, F, A, \delta)$: un automate.

AFD : automate fini déterministe.

$\mathcal{L}(\mathcal{A})$: le langage reconnu par un automate \mathcal{A} .

$\text{Rec}(A^*)$: l'ensemble des langages reconnaissables sur A .

\sim_L : une relation sur A^* associée un langage L .

$(\mathcal{R}_k)_{k \in \mathbb{N}}$: la suite de relation d'équivalence définit sur Q .

\approx : la relation d'équivalence sur Q associée à la suite de relation $(\mathcal{R}_k)_{k \in \mathbb{N}}$.

\mathcal{A}/\approx : automate quotient.

X_M : le code Morse.

$f_X(z)$: la fonction génératrice de l'ensemble X .

$\mathcal{A}(X)$: l'automate d'un code de longueur variable X .

Introduction

Le codage évoque un procédé de transformation d'un objet associé à un procédé inverse, appelé le décodage, qui permet de restituer l'objet initial.

Les débuts de la théorie du codage et de la théorie de l'information datent des travaux pionniers de Shannon de 1948. La théorie des codes s'est ultérieurement développée dans deux directions indépendantes. La première est l'étude des codes de longueur constante dans l'optique de la détection et de la correction d'erreurs. L'autre direction, initiée par Schützenberger en 1955, a conduit au développement de la théorie des codes de longueurs variables. Cette théorie est maintenant une branche de l'informatique théorique, qui a des liens importants avec les langages formels, la combinatoire sur les mots, la théorie des automates, la théorie des semigroupes et la dynamique symbolique.

Son objet est l'étude des propriétés des factorisations de mots en suites finies de mots appartenant à un ensemble donné. Plus précisément, on suppose donnés un alphabet source B et un alphabet de canal A . On cherche à coder des messages v écrits sur B en messages w sur l'alphabet A . Les lettres de B sont alors mises en correspondance avec les mots d'un code X sur l'alphabet A . Chaque lettre du message source v est remplacée par le mot qui lui est associé dans X . De plus, le codage doit être fait de telle sorte que le décodage du mot obtenu $w \in X^*$ ne conduise à aucune ambiguïté, quant au texte original.

L'objectif de ce travail est d'étudier des codes de longueurs variables et leurs représentations par un automate fini.

Ce travail est composé de trois chapitres :

Le premier chapitre consiste en un rappel des notions et notations utilisées par la suite: mots, monoïde, langage et automate fini déterministe.

Dans le second chapitre, on fait une étude sur les code de longueurs variables ainsi que certaines de leurs propriétés et on introduit aussi dans ce chapitre l'algorithme de reconnaissance des codes et la série génératrice.

Dans le troisième chapitre, on s'intéresse à la représentation d'un code de longueur variable par un automate fini.

Chapitre 1

Notions élémentaires

Ce premier chapitre contient les définitions et les propriétés des outils que nous utiliserons par la suite : mots, monoïde, langage et automate fini déterministe.

1.1 Mots

Définition 1.1.1

Un alphabet, noté A , est un ensemble fini non vide de symboles appelés également lettres de l'alphabet.

Exemples 1.1.2

1. $A_1 = \{a, b, c, \dots, z\}$, $A_2 = \{0, 1\}$.
2. Le biologiste intéressé par l'étude de l'ADN utilisera un alphabet à quatre lettres $\{A, C, G, T\}$ pour les quatre constituants des gènes : Adénine, Cytosine, Guanine et Thymine.

Définition 1.1.3

Soit A un alphabet. Un mot sur A est une suite finie de symboles. Par exemple, $abbac$ et ba 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, $|abbac| = 5$ et $|ba| = 2$.

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 ε .

L'ensemble de tous les mots formés à partir de l'alphabet A (resp. de tous les mots non-vides) est noté A^* (resp. A^+). Par exemple,

$$\{a, b, c\}^* = \{\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, \dots\}.$$

- Si a est une lettre de l'alphabet A , pour tout mot $w = w_1 w_2 \dots w_k$ de A^* , on note par :

$$|w|_a = \text{card} \{i \in \{1, 2, \dots, k\}, w_i = a\},$$

le nombre d'occurrences de la lettre a dans le mot w et $w(i)$ sa i -ème lettre. Par exemple, $|abbac|_a = 2$, $|abbac|_c = 1$ et $abbac(5) = c$.

- Le mot miroir d'un mot w est le mot formé par la suite des symboles composant w mais pris dans l'ordre inverse. Par exemple, sur l'aphabet $A_2 = \{0, 1\}$, si $u = 0101$ et $v = \varepsilon$, alors le miroir de u est 1010 et le miroir de v est ε .

Proposition 1.1.4

Soit A un alphabet.

1. L'ensemble A^* est infini.
2. L'ensemble A^* est dénombrable.

Démonstration

1. L'ensemble A^* est infini, en effet on a $A^* = \bigcup_{n=0}^{+\infty} A^n = A^0 \cup A \dots A^n \cup \dots$

2. On montre que A^* est dénombrable.

- Comme A est fini, on peut donc numéroté ses éléments, par exemple, si $A = \{\alpha, \beta, \gamma\}$, alors $n(\alpha) = 1, n(\beta) = 2, n(\gamma) = 3$.

- Ensuite, soit u un mot de A^* , on considère les longueurs $|u|$ premiers nombres premiers, par exemple si $|u| = 5$, on a les 5 premiers nombres premiers sont $p(1) = 2, p(2) = 3, p(3) = 5, p(4) = 7, p(5) = 11$.

- On forme le nombre $f(u) = \prod_{i=1}^{i=|u|} p(i)^{n(u(i))}$, où $u(i)$ désigne la i ème lettre de u . Par exemple si $u = \alpha\gamma\beta\alpha\alpha$, alors

$$f(u) = \prod_{i=1}^{i=|u|} p(i)^{n(u(i))} = \prod_{i=1}^{i=5} p(i)^{n(u(i))} = 2^1 \times 3^3 \times 5^2 \times 7^1 \times 11^1.$$

- Donc on peut définir une application

$$f : A^* \longrightarrow \mathbb{N}$$

$$u \longmapsto f(u) = \prod_{i=1}^{i=|u|} p(i)^{n(u(i))}$$

Par l'unicité de la décomposition d'un entier en facteurs premiers, l'application f est injective. Enfin, comme f est injective et l'ensemble \mathbb{N} est dénombrable, alors A^* est dénombrable. ■

Définition 1.1.5

Soit A un alphabet. On définit l'opération de concaténation (produit) sur A^* de la façon suivante. Pour tous mots $u = u_1 \dots u_k$ et $v = v_1 \dots v_l$, $u_i, v_i \in A$, La concaténation de u et v , notée $u.v$ ou simplement uv , est le mot

$$w = w_1 \dots w_{k+l} \text{ où } \begin{cases} w_i = u_i \text{ pour } 1 \leq i \leq k \\ w_{i+k} = v_i \text{ pour } 1 \leq i \leq l \end{cases}$$

On définit la puissance n -ième d'un mot w comme la concaténation de n copies de w ,

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

On pose $w^0 = \varepsilon$.

Par exemple, sur l'alphabet $A = \{a, b, c\}$, si $u = aabb$ et $v = cc$, alors $u.v = aabbcc$ et $u^2 = aabbaabb$.

Propriétés 1.1.6

La concaténation est une loi de composition interne de A^ , cette loi possède les propriétés suivantes :*

1. $\forall u, v, w \in A^*, (u.v).w = u.(v.w)$ (La loi est associative).
2. $\forall u \in A^*, u.\varepsilon = \varepsilon.u = u$ (Le mot vide ε est élément neutre du produit).
3. $\forall u, v \in A^*, |u.v| = |u| + |v|$.
4. $\forall u \in A^*, u.u = u \Leftrightarrow u = \varepsilon$ (Le mot vide ε est le seul mot idempotent).
5. *La concaténation n'est pas commutative en générale.*

Définition 1.1.7

Un mot v est un facteur d'un mot $u \in A^*$ s'il existe $x, y \in A^*$ tels que $u = xvy$. Si de plus $x = \varepsilon$, i.e, $u = vy$, alors on dit v est un préfixe, ou facteur gauche, de u . Si $y = \varepsilon$, i.e, $u = xv$, alors on dit v est un suffixe, ou facteur droit, de u .

1.2 Monoïde

Définition 1.2.1

Un monoïde est un ensemble muni d'une loi interne, i.e, d'une application $\cdot : M \times M \longrightarrow M$, qui satisfait aux conditions suivantes :

- L'opération " \cdot " est associative :

$$\forall x, y, z \in M, (x \cdot y) \cdot z = x \cdot (y \cdot z).$$

- Il existe un élément neutre (unique) $1_M \in M$ tel que

$$\forall x \in M, x \cdot 1_M = 1_M \cdot x = x.$$

Un élément $m' \in M$ est dit le symétrique de l'élément $m \in M$ si $m \cdot m' = 1_M$.

Exemples 1.2.2

- $(\mathbb{N}, +, 0)$, $(\mathbb{R}, \times, 1)$ et $(\mathbb{N} \cup \{+\infty\}, \min, +\infty)$ sont des monoïdes, où $+$ et \times dénotent respectivement l'addition et la multiplication usuelles.

- $(A^*, \cdot, \varepsilon)$ est un monoïde.

Remarque 1.2.3

Un monoïde $(M, \cdot, 1_M)$ qui est tel que tout élément de M possède un symétrique est un groupe .

Définition 1.2.4

Soit un monoïde $(M, \cdot, 1_M)$. Un sous monoïde est un triplet $(M', \cdot, 1_{M'})$ tels que

- $M \subseteq M'$;
- $1_M = 1_{M'}$;
- $\forall m_1, m_2 \in M', m_1 \cdot m_2 \in M'$.

Soit I un ensemble d'indice et $\forall i \in I, (M_i, \cdot, 1_M)$ est un sous monoïde de $(M, \cdot, 1_M)$ alors $(\bigcap_{i \in I} M_i, \cdot, 1_M)$ est un sous monoïde de $(M, \cdot, 1_M)$.

Soit Y une partie d'un monoïde M . on appelle sous monoïde engendré par Y , le plus petit sous monoïde de $(M, \cdot, 1_M)$ contenant Y , on le note Y^* . D'après ce qui précède Y^* est l'intersection de tous les sous monoïde de $(M, \cdot, 1_M)$ qui contiennent Y .

Exemple 1.2.5

Soit A l'ensemble des nombres pairs et B l'ensemble des nombres impairs. $(A, +, 0)$ est un sous monoïde de $(\mathbb{N}, +, 0)$ engendré par $\{2\}$ tandis que $(B, +, 0)$ n'est pas un sous monoïde de $(\mathbb{N}, +, 0)$.

Définition 1.2.6

Soit $(M, \cdot, 1_M)$ un monoïde. Pour tout couple (x, y) d'éléments de M , le quotient à gauche de x par y noté $y^{-1}x$ est l'ensemble $\{z \in M : y \cdot z = x\}$. Le quotient à gauche d'un sous ensemble de M par y est l'union des quotients des éléments du sous ensemble par y , i.e, si $X \subseteq M$, alors $y^{-1}X = \bigcup_{x \in X} y^{-1}x$.

Propriété 1.2.7

Soit A un alphabet quelconque. Le monoïde A^ possède les deux propriétés suivantes :*

1. *Tout élément de A^* est une suite d'éléments de A .*
2. *Deux suites distinctes d'éléments de A définissent deux éléments distincts de A^* .*

Définition 1.2.8

Soient $(M, \cdot, 1_M)$ et $(N, *, 1_N)$ deux monoïdes. Un morphisme (ou homomorphisme) de monoïdes $h : M \longrightarrow N$ est une application qui vérifie :

- $\forall x, y \in M, h(x \cdot y) = h(x) * h(y)$;
- $h(1_M) = 1_N$.

Un isomorphisme de monoïdes est un homomorphisme bijective de monoïdes.

Exemples 1.2.9

1. L'application l'ongueur $|\cdot| : A^* \longrightarrow \mathbb{N}$ est un morphisme de monoïdes entre $(A^*, \cdot, \varepsilon)$ et $(\mathbb{N}, +, 0)$. En effet, $\forall u, v \in A^*, |u.v| = |u| + |v|$ et $|\varepsilon| = 0$.
2. La fonction exponentielle représente un isomorphisme de $(\mathbb{R}, +)$ dans $(\mathbb{R}_+ - \{0\}, \times)$. Elle est bijective et vérifie $\forall x, y \in \mathbb{R}, \exp(x + y) = \exp(x) \times \exp(y)$ et $\exp(0) = 1$.
3. Soit $A = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ un alphabet, $n \in \mathbb{N} \setminus \{0, 1\}$. La fonction de Parikh

$$\begin{aligned} \Psi : A^* &\longrightarrow \mathbb{N}^n \\ w &\longmapsto \Psi(w) = (|w|_{\alpha_1}, \dots, |w|_{\alpha_n}), \end{aligned}$$

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

Proposition 1.2.10

Soit M un monoïde quelconque et f est une application d'un alphabet A dans M . Alors il existe un homomorphisme unique \tilde{f} de A^* dans M qui prolonge f , c'est-à-dire tel que $\forall a \in A, \tilde{f}(a) = f(a)$.

Démonstration

- L'existence : Posons

$$\tilde{f}(\varepsilon) = 1_M \text{ et } \tilde{f}(\alpha_1\alpha_2\ldots\alpha_n) = f(\alpha_1)f(\alpha_2)\ldots f(\alpha_n), \quad n \in \mathbb{N}, \quad \alpha_i \in A, \quad 1 \leq i \leq n.$$

Il est facile de voir que \tilde{f} est bien un homomorphisme

- L'unicité : Si \tilde{f} et \tilde{g} sont deux homomorphismes de A^* dans M tels que :

$\forall \alpha \in A, \tilde{f}(\alpha) = \tilde{g}(\alpha)$, alors $\tilde{f}(\varepsilon) = \tilde{g}(\varepsilon) = 1_M$ et pour tout mot $w = \alpha_1\alpha_2\ldots\alpha_n \in A^*$, on a $\tilde{f}(w) = \tilde{f}(\alpha_1\alpha_2\ldots\alpha_n) = f(\alpha_1)f(\alpha_2)\ldots f(\alpha_n) = \tilde{g}(\alpha_1\alpha_2\ldots\alpha_n) = \tilde{g}(w)$. ■

1.3 Langage

Définition 1.3.1

Soit A un alphabet. On appelle langage sur A tout partie (sous ensemble) L de A^* . L'ensemble des langage sur A est donc

$$\mathcal{P}(A^*) = \{L, L \subset A^*\}.$$

Un langage sur un alphabet est donc un ensemble de mots sur cet alphabet.

Remarque 1.3.2

A^* est le plus grand langage sur A au sens de l'inclusion.

Exemples 1.3.3

1. $L = \{aa, ab, ba, bb\}$ est le langage sur l'aphabet $A = \{a, b\}$ composé des mots de longueur 2.
2. $L = \{u \in \{a, b\}^*, |u|_a = |u|_b\}$ est le langage sur l'aphabet $A = \{a, b\}$ composé des mots contenant autant de a que de b .
3. $L = \emptyset$ est le langage vide, il ne contient pas aucun mot.
4. $L = \{\varepsilon\}$ est le langage contenant uniquement le mot vide.

Définition 1.3.4

Soient L_1, L_2 et L des langages définis sur un alphabet A . On définit les opérations ensemblistes (classiques) comme suivantes :

- L'union : $L_1 \cup L_2 = L_1 + L_2 = \{u \in A^*, u \in L_1 \vee u \in L_2\}$.
- L'intersection : $L_1 \cap L_2 = \{u \in A^*, u \in L_1 \wedge u \in L_2\}$.
- La Différence : $L_1 - L_2 = \{u \in A^*, u \in L_1 \text{ et } u \notin L_2\}$.
- Le complémentaire de L est le langage $L^c = \{u \in A^*, u \notin L\}$.

On définit des nouvelles opérations sur les langages sur A

- La concaténation (ou produit) des langages L_1 et L_2 est le langage

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

- En particulier, on peut définir la puissance n -ième d'un langage $L, n > 0$, par :

$$L^n = \{w_1 w_2 \dots w_n, \forall i \in \{1, 2, \dots, n\} \ w_i \in L\}.$$

Et on pose $L^0 = \{\varepsilon\}$.

Définition 1.3.7

Soit $L \subseteq A^*$, l'étoile (ou la fermeture) de Kleene de L est donné par :

$$L^* = \bigcup_{n \geq 0} L^n = \sum_{n \geq 0} L^n$$

Ainsi, les mots de L^* sont exactement les mots obtenus en concaténant un nombre arbitraire de mots de L .

On rencontre parfois l'opération L^+ définie par :

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

Exemples 1.3.8

Soient $A = \{a, b\}$ un alphabet, $L_1 = \{a\}$, $L_2 = \{ab\}$ et $L_3 = A$ trois langages sur A . on a

1. $L_1^* = \{a^n, n \geq 0\}$.
2. $L_2^* = \{(ab)^n, n \geq 0\}$.
3. $L_3^* = A^*$.

Définition 1.3.9

Soient L_1 et L_2 deux langages sur A . On appelle quotient à gauche de L_2 par L_1 et l'on note $L_1^{-1}.L_2$ le langage défini par :

$$L_1^{-1}.L_2 = \{u \in A^*, \exists v \in L_1, v.u \in L_2\}.$$

De même façons, on appelle quotient à droite de L_1 par L_2 et l'on note $L_1.L_2^{-1}$ le langage défini par :

$$L_1.L_2^{-1} = \{u \in A^*, \exists v \in L_2, u.v \in L_1\}.$$

Définition 1.3.10

Soit A un alphabet, l'ensemble des langages rationnels (ou réguliers) sur A noté $Rat(A^*)$ est définis inductivement par :

1. $\{\varepsilon\}$ et \emptyset sont des langages rationnels;
2. $\forall a \in A, \{a\}$ est un langage rationnel;
3. Si L_1 et L_2 sont des langages rationnels, alors $L_1 \cup L_2$, L_1L_2 et L_1^* sont également des langages rationnel.

Les trois opérations union, produit et étoile, qui interviennent dans la définition, sont qualifiées d'opérations rationnelles.

Définition 1.3.11

• Soit A un alphabet, l'ensemble des expressions rationnelles (ou régulières) sur A noté \mathcal{R}_A est définies inductivement par

1. ε et \emptyset sont des des expressions rationnelles;
2. $\forall a \in A, a$ est une expression rationnelle;
3. Si e_1 et e_2 sont deux expressions rationnelles, alors $(e_1 + e_2)$, (e_1e_2) , (e_1^*) sont également sont des expressions rationnelles .

• On définit alors l'application $\mathcal{L}_R : \mathcal{R}_A \rightarrow Rat(A^*)$, qui à toute expression rationnelle associée le langage rationnel de la manière suivante :

1. $\mathcal{L}_R(\emptyset) = \emptyset$ et $\mathcal{L}_R(\varepsilon) = \varepsilon$.
2. $\forall a \in A, \mathcal{L}_R(a) = \{a\}$.
3. $\mathcal{L}_R(e_1 + e_2) = \mathcal{L}_R(e_1) \cup \mathcal{L}_R(e_2)$.
4. $\mathcal{L}_R(e_1 e_2) = \mathcal{L}_R(e_1) \mathcal{L}_R(e_2)$.
5. $\mathcal{L}_R(e^*) = \mathcal{L}_R(e)^*$.

Exemples 1.3.12

Soit $A = \{a, b\}$ un alphabet, voici quelques exemples d'expressions rationnelles et les langages associés.

1. $e_1 = (\varepsilon + (a.b))$.
2. $e_2 = (((a.b).a) + b^*)^*$.
3. $e_3 = ((a + b)^* . (a.b))$.
4. $e_4 = (a + b) aa (b + \varepsilon)$.

Par conséquent :

1. $\mathcal{L}_R(e_1) = \{\varepsilon, ab\}$.
2. $\mathcal{L}_R(e_2) = (\{aba\} \cup \{b\}^*)^*$.
3. $\mathcal{L}_R(e_3) = \{a, b\}^* \{ab\}$.
4. $\mathcal{L}_R(e_4) = \{a, b\} aa \{b, \varepsilon\} = \{aaa, aab, baa, baab\}$.

Définition 1.3.13

Un langage L sur A est rationnel s'il existe une expression rationnelle $e \in \mathcal{R}_A$ telle que $L = \mathcal{L}_R(e)$, et on dit que e dénote L .

Si e_1 et e_2 sont deux expressions rationnelles telles que $\mathcal{L}_R(e_1) = \mathcal{L}_R(e_2)$ alors on dit que e_1 et e_2 sont équivalentes.

Exemple 1.3.14

On considère le langage $L = \{a^n, n > 0\}$, on peut définir trois expressions simples e_1 , e_2 et $e_3 \in \mathcal{R}_A$ dénotant L c'est-à-dire $\mathcal{L}_R(e_1) = \mathcal{L}_R(e_2) = \mathcal{L}_R(e_3) = L$, telles que.

1. $e_1 = (a.a^*)$.
2. $e_2 = (a^*.a)$.
3. $e_3 = a^+$.

Alors e_1 , e_2 et e_3 sont expressions équivalentes.

1.4 Automate fini déterministe

Définition 1.4.1

Un automate est un 5-uplet $\mathcal{A} = (Q, I, F, A, \delta)$ où,

- A est un ensemble fini dit l'alphabet d'entrée.
- Q est un ensemble dit l'ensemble des états.
- I est l'ensemble des états initiaux.
- $F \subseteq Q$ est l'ensemble des états finals.
- $\delta \subseteq Q \times A \times Q$ est l'ensemble des transitions.

Un automate fini est un automate $\mathcal{A} = (Q, I, F, A, \delta)$ dont l'ensemble d'états Q est fini.

Un automate fini $\mathcal{A} = (Q, I, F, A, \delta)$ est un automate fini déterministe et l'on note AFD si :

- $|I| = 1$;
- δ est une fonction c'est-à-dire, pour tout état $q \in Q$ et toute lettre $a \in A$ il existe au plus un état $q' \in Q$ tel que $\delta(q, a) = q'$.

Si δ est une fonction totale c'est-à-dire, que δ est défini pour tout couple $(q, a) \in Q \times A$ (on parle alors d'AFD complet).

Nous représentons un automate fini $\mathcal{A} = (Q, I, F, A, \delta)$ de la manière suivante :

- 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 d'origine q et de d'extrémité q' étiqueté par la lettre a ,

$$q \xrightarrow{a} q'.$$

- Les états finals sont représentés à un double cercle et l'état initial est désigné par une flèche entrante sans étiquette.

- Enfin si deux lettres a, b sont telles que $\delta(q, a) = q'$ et $\delta(q, b) = q'$, on s'autorise à dessiner un unique arc portant deux étiquettes séparés par une virgule

$$q \xrightarrow{a,b} q'.$$

Cette convention s'adapte à plus de deux lettres.

Exemple 1.4.2

Soit $\mathcal{A} = (Q, q_0, F, A, \delta)$ où $Q = \{1, 2, 3\}$, $q_0 = \{1\}$, $F = \{1, 2\}$, $A = \{a, b\}$ et où la fonction de transition donnée par :

δ	a	b
1	1	2
2	1	3
3	3	2

est représenté à la figure 1.4.1

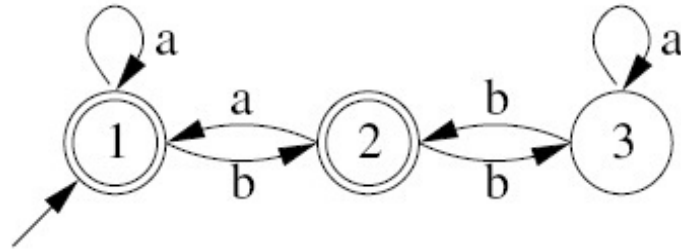


FIGURE 1.4.1 Un AFD

Définition 1.4.3

Soit $\mathcal{A} = (Q, q_0, F, A, \delta)$ un automate fini déterministe. La fonction $\delta : Q \times A \longrightarrow Q$ se prolonge en une fonction $\delta^* : Q \times A^* \longrightarrow Q$ défini par :

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

Définition 1.4.4

Soient $\mathcal{A} = (Q, I, F, A, \delta)$ un automate fini et $u = u_0 \dots u_n$ un mot sur A^* .

On appelle calcul dans \mathcal{A} tout suite de transition $(q_i \xrightarrow{u_i} q_{i+1})_{i \in [0, n]}$ tel que l'extrémité d'une transition est l'origine de la suivante, et l'on note

$$c = q_0 \xrightarrow{u_1} q_1 \xrightarrow{u_2} q_2 \xrightarrow{u_3} \dots \xrightarrow{u_n} q_n \text{ ou } q_1 \xrightarrow{u} q_n$$

q_0 est son origine, q_n son est extrémité et le mot $u = u_0 u_1 \dots u_n$ est son étiquette.

Un calcul dans \mathcal{A} est réussi lorsque son origine appartient à I et son extrémité appartient à F .

$$c \text{ est un calcul réussi} \iff c = q_1 \xrightarrow{u} q_n, \quad q_1 \in I, \quad q_n \in F, \quad u \in A^*$$

Un mot $u \in A^*$ est reconnu (ou accepté) par \mathcal{A} s'il existe un calcul réussi d'étiquette u .

Définition 1.4.5

Le langage reconnu par un automate \mathcal{A} , noté $\mathcal{L}(\mathcal{A})$, est l'ensemble des mots reconnus par cet automate.

$$\mathcal{L}(\mathcal{A}) = \{u \in A^*, \exists q_0 \in I, \delta^*(q_0, u) \in F\}.$$

Un langage $L \subseteq A^*$ est reconnaissable, s'il existe un automate fini \mathcal{A} tel que $L = \mathcal{L}(\mathcal{A})$.

On note $\text{Rec}(A^*)$ l'ensemble des langages reconnaissables sur A^* .

Exemple 1.4.6

L'automate \mathcal{A} accepte exactement le langage formé des mots sur l'alphabet $\{0, 1\}$ qu'ils contiennent un nombre pair de 0 et un nombre pair de 1, i.e,

$$\mathcal{L}(\mathcal{A}) = \{w \in \{0, 1\}^*, |w|_0 = 0 \pmod{2} \text{ et } |w|_1 = 0 \pmod{2}\}.$$

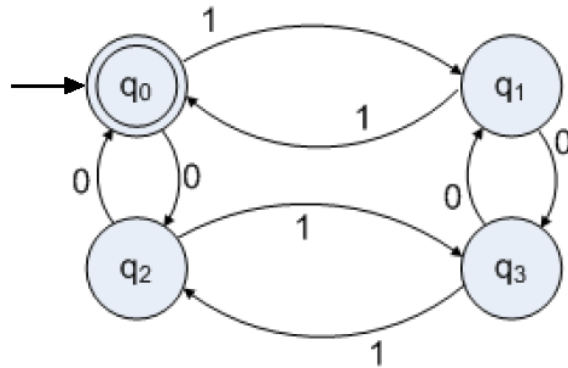


FIGURE 1.4.1 L'automate \mathcal{A} qui reconnaît L .

- L'état q_0 correspond à un nombre pair de 0 et de 1.
- L'état q_1 correspond à un nombre impair de 1.
- L'état q_2 correspond à un nombre impair de 0.
- L'état q_3 correspond à un nombre impair de 0 et de 1.

Théorème 1.4.7

L'ensemble des langages reconnaissables est exactement l'ensemble des langages rationnels, i.e,

$$\text{Rat}(A^*) = \text{Rec}(A^*).$$

Proposition 1.4.8

Soit $L_1, L_2 \subseteq A^$. On considère l'équation*

$$X = L_1.X + L_2 \quad (1)$$

Où l'inconnue X est un langage sur A .

- $\varepsilon \notin L_1$ alors l'équation (1) admet une solution unique $X = L_1^*.L_2$,
- $\varepsilon \in L_1$ l'ensemble des solutions de l'équation est $\{L_1^*.L, L_2 \subset L\}$.

Proposition 1.4.9

Soit $A = (Q, q_0, F, A, \delta)$ un automate fini déterministe, pour déterminer $L(\mathcal{A})$ le langage accepté par l'automate A , on résoudre le système des équations suivantes :

$$\begin{cases} X_q = \sum_{(q,a,q') \in \delta} aX_{q'} + \varepsilon \text{ si } q \in F \\ X_q = \sum_{(q,a,q') \in \delta} aX_{q'} \text{ sinon.} \end{cases}$$

Où $\mathcal{L}(\mathcal{A}) = X_{q_0}$.

Exemple 1.4.10

Soit $\mathcal{A} = (Q, q_0, F, A, \delta)$ où $Q = \{q_0, q_1, q_2\}$, $q_0 = \{q_2\}$, $F = \{q_2\}$, $A = \{\alpha, \beta\}$ et où la fonction de transition donnée par :

δ	α	β
q_0	q_1	—
q_1	q_1	q_2
q_2	q_1	q_2

Le langage accepté par l'automate \mathcal{A} est $\mathcal{L}(\mathcal{A}) = X_{q_0}$

Pour trouver X_{q_0} , on résout le système des équations suivants :

$$\begin{cases} X_{q_0} = \alpha X_{q_1} & (1) \\ X_{q_1} = \alpha X_{q_1} + \beta X_{q_2} & (2) \\ X_{q_2} = \alpha X_{q_1} + \beta X_{q_2} + \varepsilon \quad (q_2 \in F) & (3) \end{cases}$$

On substitue (2) dans (3) on trouve $X_{q_2} = X_{q_1} + \varepsilon$ (4)

On substitue (4) dans (2) on trouve $X_{q_1} = (\alpha + \beta) X_{q_1} + \beta$, on a $\varepsilon \notin (\alpha + \beta)$ donc d'après proposition 1.4.8 $X_{q_1} = (\alpha + \beta)^* \cdot \beta$ (5)

On substitue (5) dans (1) on trouve $X_{q_0} = \alpha \cdot (\alpha + \beta)^* \cdot \beta$

Finalement $\mathcal{L}(\mathcal{A}) = X_{q_0} = \alpha \cdot (\alpha + \beta)^* \cdot \beta$

Définition 1.4.11

Soit $L \subseteq A^*$ un langage, si w est un mot sur A , on note par $w^{-1}L$ l'ensemble des mots qui concaténés avec w appartiennent à L , i.e.,

$$w^{-1}L = \{u \in A^*, wu \in L\}.$$

On appelle résiduel du langage L , tout langage de la forme $w^{-1}L$, on note

$$Q(L) = \{w^{-1}L, w \in A^*\}.$$

Pour tout langage $L \subseteq A^*$, on peut définir une relation sur A^* , notée \sim_L comme suit :

$$w_1 \sim_L w_2 \iff w_1^{-1}L \sim_L w_2^{-1}L.$$

En d'autres termes, $w_1 \sim_L w_2$ si, et seulement si,

$$\forall u \in A^*, w_1 u \in L \iff w_2 u \in L.$$

Exemples 1.4.12

Soient $A = \{a, b\}$ et $L = \{w \in \{a, b\}^*, |w|_a \equiv 0 \pmod{3}\}$. Pour ce langage, on a par exemple, $b \sim_L ab$ car pour $u = aa$, $bu \notin L$ et $abu \in L$.

Par contre $a \not\sim_L ababaa$ car $a^{-1}L = (ababaa)^{-1}L = \{w \in \{a, b\}^*, |w|_a \equiv 2 \pmod{3}\}$.

Lemme 1.4.13

Soient L un langage et u, v deux mots sur A . On a

$$(uv)^{-1}.L = v^{-1}.(u^{-1}.L).$$

Démonstration

Si w appartient à $(uv)^{-1}.L$, cela signifie que uvw appartient à L . En d'autres termes, vw appartient à $u^{-1}.L$ et ainsi w appartient à $v^{-1}.(u^{-1}.L)$. La démonstration de l'autre inclusion est identique.

Définition 1.4.14

Soient A un alphabet et L un langage rationnel sur A . On définit l'automate minimal de L $\mathcal{A}_L = (Q_L, q_o, F_L, A, \delta_L)$ comme suit :

- $Q_L = \{w^{-1}L, w \in A^*\}$.
- $q_o = \varepsilon^{-1}L = L$.
- $F_L = \{w^{-1}L, w \in L\}$.
- $\delta_L(q, a) = a^{-1}q$, pour tous $q \in Q_L, a \in A$.

La fonction de transition δ_L s'étend à $Q_L \times A^*$ par $\delta_L(q, w) = w^{-1}q, \forall q \in Q_L, \forall w \in A^*$.

Proposition 1.4.15

L'automate minimal d'un langage $L \subseteq A^*$ accepte L .

Démonstration

En effet, soit $w \in A^*$,

$$w \in L(\mathcal{A}_L) \iff \delta_L(q_o, w) \in F_L \iff w^{-1}L \in F_L \iff w \in L$$

Définition 1.4.16

Soit $\mathcal{A} = (Q, q_o, F, A, \delta)$ un automate fini déterministe complet, on définit sur Q la suite de relation d'équivalence $(\mathcal{R}_k)_{k \in \mathbb{N}}$ comme suit :

$$\begin{aligned} (q\mathcal{R}_0q') &\iff (q \in F \iff q' \in F), \\ (q\mathcal{R}_{k+1}q') &\iff ((q\mathcal{R}_kq') \text{ et } \forall a \in A, \delta(q, a)\mathcal{R}_k\delta(q', a)), \forall k \in \mathbb{N}. \end{aligned}$$

La relation d'équivalence sur Q notée \approx associe à la suite de relation $(\mathcal{R}_k)_{k \in \mathbb{N}}$ est définie par :

$$\forall k \in \mathbb{N}, \mathcal{R}_{k+1} = \mathcal{R}_k \implies \approx = \mathcal{R}_k.$$

On considère l'automate quotient \mathcal{A}/\approx de \mathcal{A} , $\mathcal{A}/\approx = (Q_\approx, q_o, F_\approx, A, \delta_\approx)$ où,

$$Q_{\approx} = \{[q]_{\approx}, q \in Q\}.$$

$$q_o = [q_0]_{\approx}.$$

$$F_{\approx} = \{[q]_{\approx}, q \in F\}.$$

$$\delta_{\approx}([q]_{\approx}, a) = [\delta(q, a)]_{\approx}.$$

L'automate quotient \mathcal{A}/\approx est l'automate minimal de \mathcal{A} .

Exemple 1.4.17

Soit l'automate $\mathcal{A} = (Q, q_o, F, A, \delta)$ où, $Q = \{1, 2, 3, 4, 5, 6, 7\}$, $A = \{a, b\}$, $q_o = \{1\}$, $F = \{3, 4, 5\}$ et la fonction de transition δ est définie par :

δ	a	b
1	2	4
2	3	7
3	7	2
4	5	7
5	7	6
6	5	7
7	7	7

On a $(q\mathcal{R}_0q') \iff (q \in F \iff q' \in F)$, donc $Q/\mathcal{R}_0 = \{\{1, 2, 6, 7\}, \{3, 4, 5\}\}$

La relation \mathcal{R}_1 est définie comme suit :

$(q\mathcal{R}_1q') \iff ((q\mathcal{R}_0q') \text{ et } \forall a \in A, \delta(q, a) \mathcal{R}_0 \delta(q', a))$, i.e, $((q\mathcal{R}_0q') \text{ et } \delta(q, a) \mathcal{R}_0 \delta(q', a) \text{ et } \delta(q, b) \mathcal{R}_0 \delta(q', b))$.

Donc $Q/\mathcal{R}_1 = \{\{1\}, \{2, 6\}, \{7\}, \{3, 5\}, \{4\}\}$.

La relation \mathcal{R}_2 est définie comme suit :

$(q\mathcal{R}_2q') \iff ((q\mathcal{R}_1q') \text{ et } \forall a \in A, \delta(q, a) \mathcal{R}_1 \delta(q', a))$, i.e, $((q\mathcal{R}_1q') \text{ et } \delta(q, a) \mathcal{R}_1 \delta(q', a) \text{ et } \delta(q, b) \mathcal{R}_1 \delta(q', b))$.

On a $\mathcal{R}_2 = \mathcal{R}_1$ et par conséquent $Q/\mathcal{R}_2 = Q/\mathcal{R}_1$. Finalement la relation \approx est égale à la relation \mathcal{R}_1 .

L'automate minimal de \mathcal{A} est l'automate $\mathcal{A}/\approx = (Q_{\approx}, q_o, F_{\approx}, A, \delta_{\approx})$ où,

$$Q_{\approx} = \{[q]_{\approx}, q \in Q\} = \{\{1\}, \{2, 6\}, \{7\}, \{3, 5\}, \{4\}\}.$$

$$q_o = [q_0]_{\approx} = \{\{1\}\}.$$

$$F_{\approx} = \{[q]_{\approx}, q \in F\} = \{\{3, 5\}, \{4\}\}.$$

La fonction de transition δ_{\approx} est définie par :

δ_{\approx}	a	b
$\{1\}$	$\{2, 6\}$	$\{4\}$
$\{2, 6\}$	$\{3, 5\}$	$\{7\}$
$\{7\}$	$\{7\}$	$\{7\}$
$\{3, 5\}$	$\{7\}$	$\{2, 6\}$
$\{4\}$	$\{3, 5\}$	$\{7\}$

Chapitre 2

Codes de longueurs variables

Coder, c'est-à-dire remplacer des lettres par d'autres lettres ou ensembles de lettres, est non seulement utile en cryptologie où il sert à cacher le sens des messages, mais est aussi la base de la compression de données. Ce chapitre contient la définition de code ainsi que certaines propriétés qui les consernent, algorithme de reconnaissance des codes et la série génératrice.

2.1 Code

Définition 2.1.1

Soit A un alphabet. Un sous-ensemble X de monoïde libre A^* est un code sur A si pour tout $n, m \geq 1$ et $x_1x_2...x_n, x'_1x'_2...x'_m \in X$ on a :

$$x_1x_2...x_n = x'_1x'_2...x'_m \implies n = m \text{ et } x_i = x'_i \text{ pour } i = 1...n.$$

D'autre terme, un ensemble X est un code si chaque mot dans X^+ a une unique factorisation en produit de mot de X .

Les mots de X sont appelés mots de code. Les éléments de X^* sont des messages.

Exemples 2.1.2

1. L'ensemble $\{aa, baa, ba\}$ est un code sur l'alphabet $A = \{a, b\}$. Par contre, l'ensemble $\{a, ab, ba\}$ n'est pas un code car le mot $w = aba$ a les deux décompositions $(ab)a = a(ba)$. L'ensemble $X = a^*b$ est un code infini.

2. Le code de Morse X_M est un code sur l'alphabet $\{., \wedge, -\}$ dont les mots sont mis en correspondance avec les lettres a, b, \dots, z . Le symbole \wedge n'apparaît qu'à la fin des mots de X_M , ce qui assure que X_M est un code.

a .- \wedge	j .- - - \wedge	s ... \wedge
b -... \wedge	k -. - \wedge	t - \wedge
c -. - . \wedge	l .-.. \wedge	u ..- \wedge
d -.. \wedge	m - - \wedge	v ...- \wedge
e . \wedge	n -. \wedge	w .- - \wedge
f ..- \wedge	o - - - \wedge	x -.. \wedge
g --. \wedge	p .--. \wedge	y -. - - \wedge
h \wedge	q --.- \wedge	z --.. \wedge
i .. \wedge	r .-. \wedge	

On peut noter que les mots du code de Morse sont de longueurs variables.

Propriétés 2.1.3

Soit X un code sur A

- i. $\varepsilon \notin X$.
- ii. $\forall Y \subset X, Y$ est un code.
- iii. Soit B un alphabet, tout morphisme $\phi : B^* \rightarrow A^*$ qui induit une injection de B sur X est injectif.

Réciproquement, s'il existe un morphisme injectif $\phi : B^ \rightarrow A^*$ tel que $X = \phi(B)$ alors X est un code.*

Cette dernière propriété (qui pourrait aussi servir de définition aux codes) traduit la notion intuitive de code. En effet le morphisme ϕ de codage permet de coder les mots de B^* dans A^* et l'injectivité permet d'assurer que le décodage est possible.

Démonstration

- i. $\varepsilon = \varepsilon\varepsilon$ donc ε n'a pas une unique factorisation.
- ii. Toute factorisation d'un mot w dans Y est une factorisation dans X et est donc unique.

iii. Soit $\phi : B^* \rightarrow A^*$ qui induit une bijection de B sur X . Soit $u, v \in B^*$ tels que $\phi(u) = \phi(v)$.

- Si $u = \varepsilon$, supposons $v \neq \varepsilon$ alors v contient au moins une lettre b et par hypothèse $\phi(b) \in X$ or $\varepsilon \notin X$ donc $|\phi(b)| > 0$. On en déduit que $|\phi(v)| > 0$ ce qui est absurde car $\phi(u) = \varepsilon$.

- Sinon $u = b_1 \dots b_n$ et $v = b'_1 \dots b'_m$. On a alors $\phi(b_1) \dots \phi(b_n) = \phi(b'_1) \dots \phi(b'_m)$ avec $\phi(b_i), \phi(b'_j) \in X$. Or X est un code donc $n = m$ et $\forall i \phi(b_i) = \phi(b'_i)$ or ϕ induit une injection de B sur X donc $\forall i b_i = b'_i$, i.e, $u = v$. Donc ϕ est injective.

Réciproquement, soit $\phi : B^* \rightarrow A^*$ morphisme injectif, supposons qu'on a $n, m \in \mathbb{N}$ et $(x_i)_{i=1 \dots n}, (x'_j)_{j=1 \dots m} \in X = \phi(B)$ tels que $x_1 x_2 \dots x_n = x'_1 x'_2 \dots x'_m$.

Soit $(b_i)_{i=1 \dots n}, (b'_j)_{j=1 \dots m} \in B$ tels que $\forall i x_i = \phi(b_i)$ et $\forall j x_j = \phi(b'_j)$. On a donc $\phi(b_1 \dots b_n) = \phi(b'_1 \dots b'_m)$ or ϕ est injective donc $b_1 \dots b_n = b'_1 \dots b'_m$. D'où $n = m$ et $\forall i b_i = b'_i$ et donc $\forall i x_i = \phi(b_i) = \phi(b'_i) = x'_i$. ■

Corollaire 2.1.4

Soit $\phi : B^* \rightarrow A^*$ un morphisme injectif. Si $Z \subseteq B^+$ est un code, alors $\phi(Z)$ est un code. Si $X \subseteq A^+$ est un code, alors $\phi^{-1}(X)$ est un code.

Proposition 2.1.5

Pour tout $X \subset A^*$, on a

$$X \text{ est un code} \iff (\forall f \in A^* : (X^* f \cap X^* \neq \emptyset) \text{ et } (f X^* \cap X^* \neq \emptyset)) \longrightarrow f \in X^* \quad (1)$$

Démonstration.

Supposons que X vérifie la condition (1) et X ne soit pas un code.

Il existe $a_{i_1} \dots a_{i_n}, a_{j_1} \dots a_{j_m} \in X$ tels que $a_{i_1} \dots a_{i_n} = a_{j_1} \dots a_{j_m}$ avec $a_{i_1} \neq a_{j_1}$, n ou $m \neq 1$ sans quoi X ne serait pas un système minimal de générateurs pour X^* . On peut toujours supposer qu'il n'existe pas $n' < n$ et $m' < m$ tels que $a_{i_1} \dots a_{i_{n'}} = a_{j_1} \dots a_{j_{m'}}$, et que $a_{j_1} = a_{i_1} h$, $h \in A^*$. J'ai alors

$X^* h \cap X^* \neq \emptyset$ puisque $a_{i_1} h = a_{j_1}$, mais aussi

$h X^* \cap X^* \neq \emptyset$ puisque $a_{i_1} h_{j_2} \dots a_{j_m} = a_{i_1} \dots a_{i_n}$ on peut déduire $h a_{j_2} \dots a_{j_m} = a_{i_2} \dots a_{i_n}$.

Or $h \notin X^*$ ce qui contredit la condition (1).

Réciproquement, supposons que X ne vérifie pas la condition (1). Il existe alors $f \notin X^*$ tel que $a_{i_1} \dots a_{i_n} f = a_{j_1} \dots a_{j_m}$, $a_{i_1} \neq a_{j_1}$ et $f a_{k_1} \dots a_{k_p} = a_{l_1} \dots a_{l_q}$, mais alors

$$a_{i_1} \dots a_{i_n} f a_{k_1} \dots a_{k_p} = a_{j_1} \dots a_{j_m} a_{k_1} \dots a_{k_p} = a_{i_1} \dots a_{i_n} a_{l_1} \dots a_{l_q} \text{ avec } a_{i_1} \neq a_{j_1},$$

et ceci entraîne que X n'est pas un code.

Définition 2.1.6

Un sous-ensemble X de A^* est dit ensemble préfixe (resp. suffixe) si aucun mot de X n'est préfixe (resp. suffixe) propre d'un mot de X , i.e, pour tous mots u et v dans X ,

$$u \leq v \implies u = v \text{ où } \leq \text{ signifie être un préfixe (resp. suffixe).}$$

X est bipréfixe s'il est à la fois préfixe et suffixe.

Par exemple, sur l'alphabet $A = \{a, b\}$.

$\{a, ba\}$ est un ensemble préfixe alors que $\{a, ab\}$ n'en est pas un.

$\{a, ab, bb\}$ est un ensemble suffixe.

Proposition 2.1.7

$$\forall X \subset A^*, X \text{ préfixe} \implies X \text{ est un code.}$$

Démonstration

Supposons que X n'est pas un code. Soit w de longueur minimale tel que w ait deux factorisations dans X . On a donc $n, m \in \mathbb{N}$ et $(x_i)_{i=1 \dots n}, (x'_j)_{j=1 \dots m} \in X$ tels que $w = x_1 x_2 \dots x_n = x'_1 x'_2 \dots x'_m$. Comme w est de longueur minimale, on a $x_1 \neq x'_1$ et donc $x_1 < x'_1$ ou $x'_1 < x_1$ ce qui rentre en contradiction avec X préfixe. ■

Exemple 2.1.8

Soit $A = \{a, b\}$ et $X = \bigcup_{n \geq 0} a^n b A^n$. X est un préfixe car $a^n b u = a^m b v \implies m = n$ et donc $u = v$. C'est donc un code sur A .

Définition 2.1.9

Un code X est maximal sur A si X n'est pas strictement inclus dans un autre code sur A , i.e, si

$$X \subset X', \quad X' \text{ code} \implies X = X'.$$

Exemples 2.1.10

$X_1 = \{aa, ab, bb, ba\}$ est un code maximal fini (en effet, si un surcode contient un autre mot w alors ww admet deux décompositions car étant de longueur paire).

$X_2 = ba^*$ est un code maximal infini.

$X_3 = \{a, ba\}$ est un code mais n'est pas maximal (cela reste un code si on lui ajoute bb).

2.2 Algorithme de reconnaissance des codes

Reconnaitre si un ensemble donné est un code n'est pas toujours chose facile, mais il existe un algorithme de Sardinas et Patterson qui permet de le décider. Les deux propositions qui suivent sont la preuve de correction et de terminaison de cet algorithme.

Proposition 2.2.1

Soit $X \subset A^*$. On définit par récurrence la suite $(U_n)_{n \in \mathbb{N}^*}$ comme suite :

$$\begin{cases} U_1 = X^{-1}X \setminus \{\varepsilon\} \\ U_{n+1} = X^{-1}U_n \cup U_n^{-1}X \text{ pour tout } n \geq 1 \end{cases}$$

On a alors :

$$X \text{ est un code} \iff \forall n \geq 1 \ \varepsilon \notin U_n.$$

La démonstration nécessite le lemme suivant :

Lemme 2.2.2

Soit $X \subset A^+$. $\forall n \geq 1 \ \forall k = \{1 \dots n\}$ on a

$$\varepsilon \in U_n \iff \exists u \in U_k \ \exists i, j \in \mathbb{N}^2 \ uX^i \cap X^j \neq \emptyset,$$

avec $i + j + k = n$

Démonstration

On prouve le lemme à n fixé par récurrence descendante sur k .

Si $k = n$, on a évidemment $i = j = 0$. Si $\varepsilon \in U_n$ on pose $u = \varepsilon$ et on a bien $\varepsilon X^0 \cap X^0 = \{\varepsilon\}$.

Réciproquement si on a $u \in U_n$ tel que $uX^0 \cap X^0 = \{u\} \cap \{\varepsilon\} \neq \emptyset$ alors $u = \varepsilon$ et donc $\varepsilon \in U_n$.

Soit $1 \leq k < n$, supposons la propriété vérifiée pour $k + 1$. Si $\varepsilon \in U_n$ par hypothèse de récurrence, $\exists v \in U_{k+1}$, $\exists i, j \in \mathbb{N}^2$ tel que $i + j + k + 1 = n$ et $\exists x, y \in X^i \times X^j$ tel que $vx = y \in uX^i \cap X^j$. Comme $U_{k+1} = X^{-1}U_k \cup U_k^{-1}X$, on a $z, u \in X \times U_k$ tel que soit $zv = u$ soit $z = uv$.

- Dans le premier cas, on a $ux = zvx = zy$ comme $z, y \in X \times X^j$, $zy \in X^{j+1}$ et $uX^i \cap X^{j+1} \neq \emptyset$.

- Dans le deuxième cas $uy = uvx = zx \in X^{i+1}$ donc $uX^j \cap X^{i+1} \neq \emptyset$, avec à chaque fois $u \in U_k$.

Réciproquement, soient $w \in uX^i \cap X^j$ où $i + j + k = n$. Si $j = 0$, alors l'intersection est vide à moins d'avoir $u = \varepsilon$ et $i = 0$ car $\varepsilon \notin X$, on a alors $k = n$ mais on a supposé $k < n$ donc $j \geq 1$. On a donc $v, x, v' \in X^i \times X \times X^{j-1}$ tels que $uv = xv'$. On distingue ensuite 2 cas suivant les longueurs comparées de u et x .

- Si $|u| \leq |x|$ alors $\exists u' \in A^*$ $uu' = x$ et alors $u' \in U_k^{-1}X \subset U_{k+1}$. De plus $v = u'v'$ donc $u'X^{j-1} \cap X^i \neq \emptyset$ et par hypothèse de récurrence $\varepsilon \in U_n$.

- Sinon $\exists x' \in A^+$ $u = xx'$ avec $x' \in X^{-1}U_k \subset U_{k+1}$ et $x'v = v' \in x'X \cap X^{j-1}$. D'après l'hypothèse de récurrence $\varepsilon \in U_n$. ■

Démonstration

On peut maintenant démontrer la proposition 2.2.1

Supposons que X ne soit pas un code. Soit w de longueur minimale tel que w ait deux factorisations dans X .

On a $n, m \in \mathbb{N}$ et $(x_i)_{i=1\dots n}, (x'_j)_{j=1\dots m} \in X$ tels que $w = x_1x_2\dots x_n = x'_1x'_2\dots x'_m$ avec $x_1 \neq x'_1$. On peut supposer sans perdre de généralité que $|x_1| < |x'_1|$ car w est de longueur minimale. On a alors $\exists u \in A^+$ $x_1u = x'_1$ avec $u \in X^{-1}X \setminus \{\varepsilon\} = U_1$ et $uX^{m-1} \cap X^{n-1} \neq \emptyset$ et donc $\varepsilon \in U_{n+m+1}$ d'après le lemme 2.2.2.

Réciproquement, Si $\varepsilon \in U_n$ pour un certain n , on applique le lemme avec $k = 1$. $\exists u \in U_1$, $i, j \in \mathbb{N}^2$ et $v, w \in X^i \times X^j$ tels que $uX^i \cap X^j \neq \emptyset$. De plus comme $U_1 = X^{-1}X \setminus \{\varepsilon\}$ $\exists x, y \in X$ tels que $xu = y$ avec $x \neq y$ car $u \neq \varepsilon$. d'où $yX^i \cap xX^j = xuX^i \cap xX^j \neq \emptyset$ ce qui fait que X ne peut être un code. ■

Exemples 2.2.3

1. Soient $A = \{0, 1\}$ et $X = \{00, 010, 101, 11\}$ on a,

$$U_1 = X^{-1}X \setminus \{\varepsilon\} \text{ tel que } X^{-1}X = \bigcup_{x \in X} x^{-1}X \text{ avec } x^{-1}X = \{y \in \{0, 1\}^* : xy \in X\}.$$

- $(00)^{-1}X = \{y \in \{a, b\}^* : (00)y \in X\} = \{\varepsilon\}.$
- $(010)^{-1}X = \{y \in \{a, b\}^* : (010)y \in X\} = \{\varepsilon\}.$

- $(101)^{-1}X = \{y \in \{a, b\}^* : (101)y \in X\} = \{\varepsilon\}.$
- $(11)^{-1}X = \{y \in \{a, b\}^* : (11)y \in X\} = \{\varepsilon\}.$

Alors $U_1 = X^{-1}X \setminus \{\varepsilon\} = \emptyset.$

$U_2 = X^{-1}U_1 \cup U_1^{-1}X$ tel que $X^{-1}U_1 = \bigcup_{x \in X} x^{-1}U_1$ et $U_1^{-1}X = \bigcup_{u_1 \in U_1} u_1^{-1}X.$

On a $U_2 = U_1 = \emptyset.$ Donc l'ensemble X est un code de longueur variable.

2. Soient $A = \{a, b\}$ et $X = \{a, ab, ba\}$ on a,

$U_1 = X^{-1}X \setminus \{\varepsilon\}$ tel que $X^{-1}X = \bigcup_{x \in X} x^{-1}X$ avec $x^{-1}X = \{y \in \{a, b\}^* : xy \in X\}.$

- $a^{-1}X = \{y \in \{a, b\}^* : ay \in X\} = \{b\}.$
- $(ab)^{-1}X = \{y \in \{a, b\}^* : (ab)y \in X\} = \{\varepsilon\}.$
- $(ba)^{-1}X = \{y \in \{a, b\}^* : (ba)y \in X\} = \{\varepsilon\}.$

Alors $U_1 = X^{-1}X \setminus \{\varepsilon\} = \{b\}.$

$U_2 = X^{-1}U_1 \cup U_1^{-1}X$ tel que $X^{-1}U_1 = \bigcup_{x \in X} x^{-1}U_1$ et $U_1^{-1}X = \bigcup_{u_1 \in U_1} u_1^{-1}X.$

On a $X^{-1}U_1 = \bigcup_{x \in X} x^{-1}U_1$ avec $x^{-1}U_1 = \{y \in \{a, b\}^* : xy \in U_1\}.$

- $a^{-1}U_1 = \{y \in \{a, b\}^* : ay \in U_1\} = \emptyset.$
- $(ab)^{-1}U_1 = \{y \in \{a, b\}^* : (ab)y \in U_1\} = \emptyset.$
- $(ba)^{-1}U_1 = \{y \in \{a, b\}^* : (ba)y \in U_1\} = \emptyset.$

Alors $X^{-1}U_1 = \emptyset.$

On a $U_1^{-1}X = \bigcup_{u_1 \in U_1} u_1^{-1}X = b^{-1}X = \{y \in \{a, b\}^* : by \in X\} = \{a\}$

Alors $U_1^{-1}X = \{a\}.$

Donc $U_2 = X^{-1}U_1 \cup U_1^{-1}X = \{a\}.$

$U_3 = X^{-1}U_2 \cup U_2^{-1}X$, on a

$U_2^{-1}X = a^{-1}X = \{y \in \{a, b\}^* : ay \in X\} = \{\varepsilon, b\}.$

Puisque $\varepsilon \in U_3$ alors X n'est pas un code de longueur variable.

La construction des U_n donne donc un algorithme pour déterminer si un ensemble est un code. La propriété suivante conclut quant à la terminaison de l'algorithme dans le cas où X est rationnel.

Propositions 2.2.4

Si X est rationnel l'ensemble de ses U_n est fini.

Démonstration

On rappelle qu'un langage est rationnel est équivalent au fait que le nombre de ses quotients à gauche est fini. On montre par récurrence sur n que les U_n sont des unions finies de quotients à gauche de X auxquelles on peut avoir retiré ε .

$$\text{Pour } n = 1, U_1 = X^{-1}X \setminus \{\varepsilon\} = \left(\bigcup_{x \in X} x^{-1}X \right) \setminus \{\varepsilon\}.$$

Supposons que c'est vrai au rang $n - 1$, alors comme le quotients à gauche d'un quotient à gauche de X est un quotient à gauche de $X^{-1}U_n$ est bien une union de quotients à gauche de X (l'absence ou la présence de ε ne change pas grand chose juste de nombreuses disjonctions de cas si on veut rentrer dans les détails). De plus $(U_n)^{-1}X$ est bien sur une union de quotient à gauche de X .

Comme le nombre de quotients à gauche de X est fini leurs unions sont aussi en nombre fini (retirer ε ne fait que doubler ce nombre au pire).

Exemple 2.2.5

Soit $A = \{a, b\}$, considérons $X = \{aa, ba, bb, baa, bba\}$. X n'est pas préfixe et considérer des factorisations dans X n'est pas vraiment envisageable. Les U_n permettent pourtant de conclure très vite. En effet $U_1 = \{a\} = U_2$.

2.3 Série génératrice

Les séries génératrices sont on outil algébrique qui permet de reformuler des problèmes de combinatoire afin de les transformer en des problèmes de manipulation d'expressions algébriques. En particulier, en combinatoire, il s'agit souvent de déterminer le nombre d'objets d'un certain type qui sont de taille n , ce qui donne lieu à une suite $(a_n)_{n \geq 0}$ dont on cherche à déterminer le n -ième terme.

Définition 2.3.1

La fonction génératrice associée à la suite $(a_n)_{n \geq 0}$ est la série (somme infinie) formelle

$$a_0 + a_1x + a_2x^2 + \dots = \sum_{k \geq 0} a_k x^k.$$

En particulier, la série génératrice d'une suite finie est un polynôme.

Exemples 2.3.2

1. Soit n un entier strictement positif, la suite (finie) des coefficients binomiaux $\left(\binom{n}{k}\right)_{k \in \{0, \dots, n\}}$ a pour série génératrice le polynôme

$$\sum_{k=0}^n \binom{n}{k} x^k = (1+x)^n.$$

2. Soit n un entier, si la suite $a_n = 1$, on a alors $f(x) = \sum_{n \geq 0} x^n = 1 + x + x^2 + \dots$

On remarque alors

$$xf(x) = x \sum_{n \geq 0} x^n = x(1 + x + x^2 + \dots) = x + x^2 + x^3 + \dots = 1 - f(x), \text{ i.e,}$$

$$xf(x) = 1 - f(x). \text{ Alors } (1-x)f(x) = 1. \text{ Donc}$$

$$\sum_{n \geq 0} x^n = \frac{1}{1-x}.$$

Définition 2.3.3

La somme de deux séries génératrices se définit de manière assez évidente en sommant les suites correspondantes.

Pour le produit, c'est un peu plus compliqué. Il se fait par analogie avec le produit des polynômes :

$$\left(\sum_{m \geq 0} a_m x^m\right) \left(\sum_{n \geq 0} b_n x^n\right) = \sum_{m, n \geq 0} a_m b_n x^{m+n} = \sum_{n \geq 0} \left(\sum_{k=0}^n a_k b_{n-k}\right) x^n.$$

Le produit est donc également une série génératrice, correspondant à la suite

$$c_n = \sum_{k=0}^n a_k b_{n-k}.$$

La dérivée au sens formel d'une série génératrice se définit sans trop de problèmes par analogie avec les polynômes :

$$\left(\sum_{n \geq 0} a_n x^n\right)' = \sum_{n \geq 1} n a_n x^{n-1}.$$

Définition 2.3.4

Soient r un réel et k un entier naturel. Alors on définit le coefficient binomial généralisé $\binom{r}{k}$ par :

$$\binom{r}{k} = \frac{r(r-1) \dots (r-k+1)}{k!}.$$

Par exemple, $\binom{-1}{k} = \frac{(-1)(-2)\dots(-k)}{k!} = (-1)^k$.

Proposition 2.3.5

Soient n et k des entiers naturels. Alors

$$\binom{-n}{k} = (-1)^k \binom{n+k-1}{k}.$$

Définition 2.3.6

Pour chaque ensemble $X \subset A^*$, la fonction génératrice ou série génératrice de X est la série formelle.

$$f_X(z) = \sum_{n \geq 0} u_n z^n \text{ tel que } u_n = \text{Card}(X \cap A^n).$$

Proposition 2.3.7

Si X est un code, alors $f_{X^*} = \frac{1}{(1 - f_X)}$.

Démonstration

Soient X un code et $f_X = \sum_{n \geq 0} u_n z^n$ sa série génératrice. La série génératrice de $f_{X^*}(z)$ du monoïde libre $X^* = \sum_{n \geq 0} X^n$, engendré par X , est, par définition,

$$f_{X^*} = \sum_{n \geq 0} f_{X^n}(z).$$

Où $f_{X^n}(z)$ est la série génératrice de X^n et comme X est un code alors $f_{X^n}(z) = (f_X(z))^n$.

On obtient alors

$$f_{X^*}(z) = \sum_{n \geq 0} (f_X(z))^n = \frac{1}{1 - f_X} = \frac{1}{1 - \sum_{n \geq 0} u_n z^n}. \quad \blacksquare$$

Exemples 2.3.8

1. L'ensemble $X = \{b, ab\}$ est un code préfixe sur l'alphabet $A = \{a, b\}$. La série f_{X^*} est

$$f_{X^*}(z) = \frac{1}{1 - z - z^2}.$$

En effet, on a $f_X(z) = \sum_{n \geq 0} u_n z^n$ tel que $u_n = \text{Card}(X \cap A^n)$.

- $u_0 = \text{Card}(X \cap A^0) = |\{b, ab\} \cap \{\varepsilon\}| = |\{\emptyset\}| = 0$.
- $u_1 = \text{Card}(X \cap A^1) = |\{b, ab\} \cap \{a, b\}| = |\{b\}| = 1$.
- $u_2 = \text{Card}(X \cap A^2) = |\{b, ab\} \cap \{aa, ab, ba, bb\}| = |\{ab\}| = 1$.

$$\bullet u_n = 0, \forall n \geq 3.$$

$$\text{Alors } f_X(z) = \sum_{n=0}^2 u_n z^n = 0 + z + z^2 = z + z^2.$$

$$\text{Donc } f_{X^*} = \frac{1}{1 - f_X} = \frac{1}{1 - (z + z^2)} = \frac{1}{1 - z - z^2}.$$

2. L'ensemble de mots sur $A = \{a, b\}$ qui a le même nombre d'occurrence de a et b est un sous monoïde de A^* engendré par un code préfixe D , i.e,

$$D^* = \{w \in \{a, b\}^*, |w|_a = |w|_b\}.$$

Alors $\forall w \in D^*, |w| = 2n$ (La longueur de w est paire)

La série génératrice de l'ensemble D^* est $f_{D^*}(z) = \sum_{n \geq 0} u_n z^n$ où $u_n = \text{Card}(D^* \cap A^n)$, mais si n est impair alors $u_n = \text{Card}(D^* \cap A^n) = 0$. Et puisque chaque mot de D^* de longueur $2n$ s'obtient par on chose n positions parmi $2n$, Donc

$$f_{D^*}(z) = \sum_{n \geq 0} \binom{2n}{n} z^{2n}.$$

On a

$$\binom{-\frac{1}{2}}{n} = \frac{1}{(-4)^n} \binom{2n}{n} \text{ alors } \binom{2n}{n} = \binom{-\frac{1}{2}}{n} \times (-4)^n.$$

Donc

$$f_{D^*}(z) = \sum_{n \geq 0} \binom{2n}{n} z^{2n} = \sum_{n \geq 0} \binom{-\frac{1}{2}}{n} (-4)^n z^{2n} = \sum_{n \geq 0} \binom{-\frac{1}{2}}{n} (-4z^2)^n = (1 - 4z^2)^{-\frac{1}{2}}.$$

D est un code alors $f_D(z) = \frac{1}{1 - f_{D^*}(z)}$, donc

$$f_D(z) = 1 - \frac{1}{f_{D^*}(z)} = 1 - \frac{1}{(1 - 4z^2)^{-\frac{1}{2}}} = 1 - (1 - 4z^2)^{\frac{1}{2}}.$$

On calcule la série génératrice de l'ensemble D .

On a

$$f_D(z) = 1 - (1 - 4z^2)^{\frac{1}{2}}.$$

En utilisant la formule de Newton généralisée, on obtient

$$f_D(z) = 1 - \sum_{n \geq 0} \binom{\frac{1}{2}}{n} (-4z^2)^n.$$

Où $\binom{\frac{1}{2}}{0} = 1$ et $\binom{\frac{1}{2}}{n} = \frac{\frac{1}{2}(\frac{1}{2}-1)(\frac{1}{2}-2)\dots(\frac{1}{2}-n+1)}{n!}$.

Alors

$$\begin{aligned}
f_D(z) &= 1 - \left(\binom{\frac{1}{2}}{0} (-4z^2)^0 + \sum_{n \geq 1} \binom{\frac{1}{2}}{n} (-4z^2)^n \right) \\
&= 1 - \left(1 + \sum_{n \geq 1} \binom{\frac{1}{2}}{n} (-4z^2)^n \right) \\
&= - \sum_{n \geq 1} \binom{\frac{1}{2}}{n} (-4z^2)^n \\
&= - \sum_{n \geq 1} \frac{\frac{1}{2}(\frac{1}{2}-1)(\frac{1}{2}-2)\dots(\frac{1}{2}-n+1)}{n!} (-4)^n (z^2)^n \\
&= - \sum_{n \geq 1} \frac{\frac{1}{2} \left(\frac{-1}{2}\right) \left(\frac{-3}{2}\right) \left(\frac{-5}{2}\right) \left(\frac{-7}{2}\right) \dots \left(\frac{-2n+3}{2}\right)}{n!} (-1)^n \times 2^{2n} \times z^{2n} \\
&= - \sum_{n \geq 1} \frac{\frac{1}{2} \left(\frac{-1}{2}\right) \left(\frac{-3}{2}\right) \left(\frac{-5}{2}\right) \left(\frac{-7}{2}\right) \dots \left(\frac{-(2n-3)}{2}\right)}{n!} (-1)^n \times 2^n \times 2^n \times z^{2n} \\
&= \sum_{n \geq 1} \frac{(1)(1)(3)(5)(7)\dots(2n-3)}{n! \times n!} 2^n \times n! \times z^{2n}.
\end{aligned}$$

On a $n! \times 2^n = (1 \times 2 \dots \times n)(2 \times 2 \dots \times 2) = 2 \times 4 \times 6 \dots \times 2n$.

Alors $1 \times 3 \times 5 \times 7 \dots \times (2n-3) \times 2 \times 4 \dots \times (2n-2) \times 2n = (2n-2)!2n$.

Donc

$$f_D(z) = \sum_{n \geq 1} \frac{(2n-2)!2n}{n!n!} z^{2n} = \sum_{n \geq 1} \frac{(2n-2)!2n}{n(n-1)!n(n-1)!} z^{2n} = \sum_{n \geq 1} \frac{2}{n} \binom{2n-2}{n-1} z^{2n}$$

Chapitre 3

La représentation d'un code de longueur variable par un automate fini

Ce chapitre contient : l'automate fini déterministe associé un mot de code et la représentation d'un code par un automate fini.

3.1 Automate fini déterministe associé un mot de code

Définition 3.1.1

Soient A un alphabet et X un code de longueur variable sur A . La représentation de X est basée en la théorie des automates. On construit un automate pour X par l'union des automates de mots de code.

Si le mot de code $w = x_1...x_n$ alors l'automate $\mathcal{A}(w)$ de w est $\mathcal{A} = (Q_w, q_i, F_w, A, \delta_w)$ tels que :

- $Q_w = \{q_i, q_{x_1}, q_{x_1x_2}, \dots, q_{x_1x_2\dots x_{n-1}}\}$, avec $\text{Card}(Q) = \text{longueur}(w)$.
- $F_w = \{q_i\}$.
- La fonction $\delta_w : Q_w \times A \longrightarrow Q_w$ définit par :

$$\delta_w(q_i, x_1) = q_{x_1}, \delta_w(q_{x_1}, x_2) = q_{x_1x_2}, \dots, \delta_w(q_{x_1x_2\dots x_{n-2}}, x_{n-1}) = q_{x_1x_2\dots x_{n-1}},$$

$$\delta_w(q_{x_1x_2\dots x_{n-1}}, x_n) = q_i.$$

Ainsi $\mathcal{A}(w)$ peut reconnaître $w^* = \bigcup_{k \geq 0} w^k$.

Exemple 3.1.2

Figure 3.1.1 représente l'automate du mot de code $w_1 = 0100$.

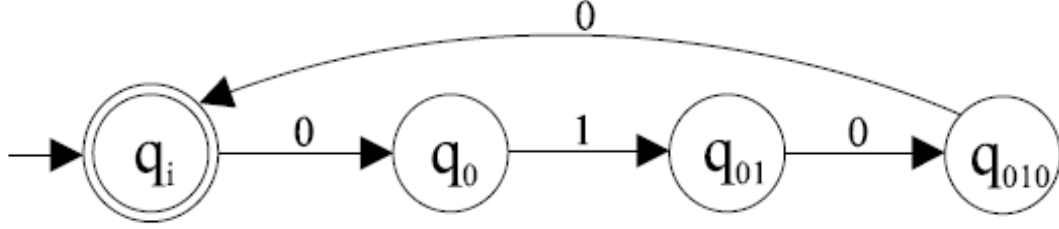


FIGURE 3.1.1 L'automate $\mathcal{A}(0100)$

Si w_1 est un préfixe de w_2 alors leurs automates auront des états communs, i.e, $Q_{w_1} \subset Q_{w_2}$, cette propriété apparaît dans la figure 3.1.2, tel que w_1 est préfixe de $w_2 = 010011$.

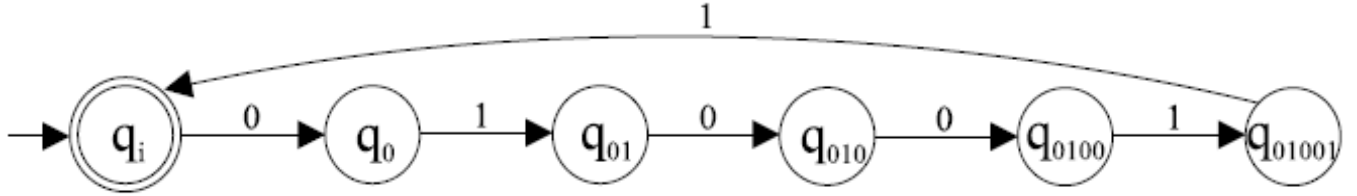


FIGURE 3.1.2 L'automate $\mathcal{A}(010011)$

3.2 Représentation d'un code de longueur variable

Définition 3.2.1

L'automate $\mathcal{A}(w_1, w_2, \dots, w_3)$ du code de longueur variable $X = \{w_1, w_2, \dots, w_n\}$ noté $\mathcal{A}(X)$.

$\mathcal{A}(X) = (Q_X, q_i, F_X, A, \delta_X)$ avec

- $Q_X = Q_{w_1} \cup Q_{w_2} \dots \cup Q_{w_n}$.
- $F_X = \{q_i\}$.
- $\delta_X = \delta_{w_1} \cup \delta_{w_2} \dots \cup \delta_{w_n}$.

L'automate $\mathcal{A}(X)$ accepte $X^* = \bigcup_{k \geq 0} X^k$.

Exemple 3.2.2

1. Soient $A = \{0, 1\}$ et $X_1 = \{00, 01, 10, 11\}$, on a

$\mathcal{A}(X_1) = (Q_{X_1}, q_i, F_{X_1}, A, \delta_{X_1})$ avec

- $Q_{X_1} = \{q_i, q_0, q_1\}$.
- $F_{X_1} = \{q_i\}$.
- δ_{X_1} est donnée par le tableau suivant :

δ_{X_1}	0	1
q_i	q_0	q_1
q_0	q_i	q_i
q_1	q_i	q_i

Et représenté à la figure 3.2.1

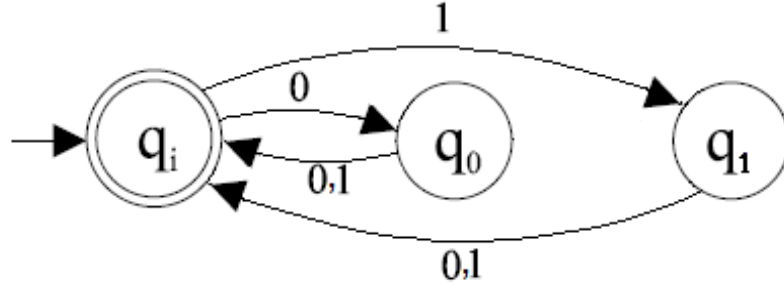


FIGURE 3.2.1 L'automate $\mathcal{A}(X_1)$

Le langage accepté par $\mathcal{A}(X_1)$ est :

$$\begin{aligned}
 L(\mathcal{A}(X_1)) &= \{w \in A^* : \delta(q_i, w) \in F_{X_1}\} \\
 &= \{w \in A^* : \delta(q_i, w) = q_i\} \\
 &= X_1^* \\
 &= \{00, 01, 10, 11\}^*
 \end{aligned}$$

L'équation caractéristique pour les états q_i, q_0 et q_1 respectivement, sont :

$$\begin{cases} x_i = 0x_0 + 1x_1 + \varepsilon \\ x_0 = (0 + 1)x_i \\ x_1 = (0 + 1)x_i \end{cases}$$

Avec $L(\mathcal{A}(X_1)) = x_i$, on a $x_i = 0x_0 + 1x_1 + \varepsilon = 0(0 + 1)x_i + 1(0 + 1)x_i + \varepsilon$.

Alors $x_i = [0(0 + 1) + 1(0 + 1)]x_i + \varepsilon$.

Finalement $x_i = [0(0 + 1) + 1(0 + 1)]^* = X_1^*$.

2. Soient $A = \{0, 1\}$ et $X_2 = \{00, 010, 101, 11\}$, on a

$\mathcal{A}(X_2) = (Q_{X_2}, q_i, F_{X_2}, A, \delta_{X_2})$ avec

- $Q_{X_2} = \{q_i, q_0, q_1, q_{01}, q_{10}\}$.
- $F_{X_2} = \{q_i\}$.
- δ_{X_2} est donnée par le tableau suivant :

δ_{X_2}	0	1
q_i	q_0	q_1
q_0	q_i	q_{01}
q_1	q_{10}	q_i
q_{01}	q_i	—
q_{10}	—	q_i

Et représenté à la figure 3.2.2

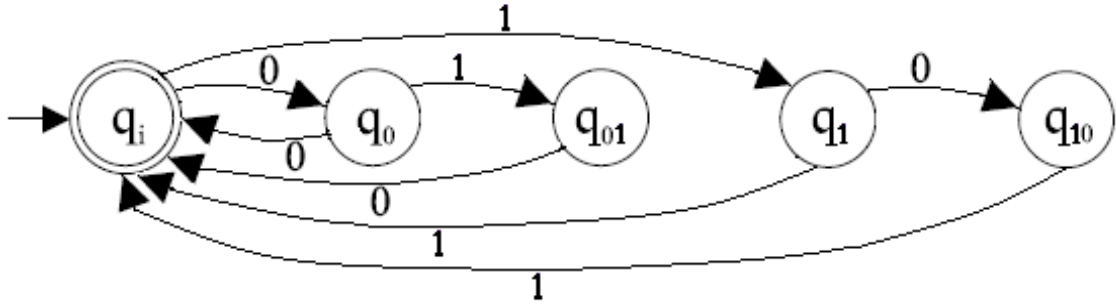


FIGURE 3.2.2 L'automate $\mathcal{A}(X_2)$

Le langage accepté par $\mathcal{A}(X_2)$ est :

$$\begin{aligned}
 L(\mathcal{A}(X_2)) &= \{w \in A^* : \delta(q_i, w) \in F_{X_2}\} \\
 &= \{w \in A^* : \delta(q_i, w) = q_i\} \\
 &= X_2^* \\
 &= \{00, 010, 101, 11\}^*
 \end{aligned}$$

L'équation caractéristique pour les états q_i , q_0 , q_1 , q_{01} et q_{10} respectivement, sont :

$$\left\{ \begin{array}{l} x_i = 0x_0 + 1x_1 + \varepsilon \\ x_0 = 0x_i + 1x_{01} \\ x_1 = 1x_i + 0x_{10} \\ x_{01} = 0x_i \\ x_{10} = 1x_i \end{array} \right.$$

Avec $L(\mathcal{A}(X_2)) = x_i$, on a $x_0 = 0x_i + 1x_{01} = 0x_i + 10x_i = (0 + 10)x_i$,

et $x_1 = 1x_i + 0x_{10} = 1x_i + 01x_i = (1 + 01)x_i$.

Alors $x_i = 0x_0 + 1x_1 + \varepsilon = 0(0 + 10)x_i + 1(1 + 01)x_i + \varepsilon = (00 + 010 + 101 + 11)x_i + \varepsilon$.

Finalement $x_i = L(\mathcal{A}(X_2)) = (00 + 010 + 101 + 11)^*$.

Conclusion

Dans ce mémoire, on a fait une étude sur les codes de longueurs variables ainsi que certaines de leurs propriétés, et on utilise un algorithme de Sardinas et Patterson qui permet de les reconnaître, enfin on donne la représentation des codes de longueurs variables par un automate fini.

Bibliographie

- [1] **F. Bassino**, *Séries rationnelles et distributions de longueurs*, Thèse de Doctorat, Université de Marne-La-Vallée, 1996.
- [2] **J. Berstel et D. Perrin**, *Theory of codes*, Université de paris VI, Université de paris VII, 1985.
- [3] **P. Berlioux, M. Echenim et M. Lévy**, *Théorie des langages*, Ecole nationale supérieure d'informatique et de mathématiques appliquées de France, 2009.
- [4] **T. Bourdier**, *Mathématiques Discrètes 1 & Informatique Théorique*, École Supérieure d'Informatique et Applications de Lorraine, 2007-2008
- [5] **N. Daviaud**, *Théorie des code*, 2006
- [6] **N. Ghabbane**, *A construction and representation of some variable length codes*, Anale. Seria Informatică, vol 2, 2017.
- [7] **N. Ghabbane**, *Systèmes de réécriture et le problème du mot dans un monoïde*, Thèse de Doctorat, Université de M'sila, 2017.
- [8] **N. Ghabbane**, Cours Master1, *Semi groupes et automates finis*, Université de M'sila, 2017-2018.
- [9] **C. Moulin**, *Théorie des langages*, Université de Technologie de Compiègne, 2013.
- [10] **B. Margaret**, *Séries génératrices*, 2013
- [11] **M. Nivat**, *Elements de la théorie générale des codes*, Université de paris, 1965.

- [12] **F. Olive**, *Automates & Langages*, 2010-2011.
- [13] **S. Rideau**, *Théorie des codes*, 2008.
- [14] **M. Rigo**, *Théorie des automates et langages formels*, Université de Liège, 2009.
- [15] **F. Yvon, A. Demaille et P. Senellart**, *Théorie des langages*, 2016.