REPUBLIQUE ALGERIENNE
DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
Université Mohamed Boudiaf de M'sila
Faculté des Mathématiques et de l'Informatique
Département des Mathématiques

Mémoire de Master

Domaine : Mathématiques et Informatique

Filière : Mathématiques

Option : Algèbre et Mathématiques Discrète

Thème

Fonctions génératrices des codes à longueurs variables

Présentée par :

M^r Djaidja Faris Badr

Soutenu publiquement le : 19/06/2023.

Devant le jury composé de :

MIHOUBI Douadi	Prof,	Université de M'sila	**Président**.
GHADBANE Nacer	M.C.A,	Université de M'sila	**Encadreur**.
BOUDAOUD Abdelmadjid	Prof,	Université de M'sila	**Examinateur**.

Année universitaire 2022/2023

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 amis.

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

- Avant tout, nous remercions ALLAH le tout Puissant de nous avoir accordé la force et les moyens afin de pouvoir accomplir ce mémoire.

- Nous exprimons nos vifs remerciements, nos profondes gratitudes et nos reconnaissances pour notre encadreur le Docteur Nacer GHADBANE, qui a dirigé ce mémoire. Son bonté et sa confiance nous ont permis de progresser régulièrement. Nous tenons à le remercier pour ses conseils avisés ses valeurs uniques ainsi que sa patience avec laquelle il a accompagné notre travail.

- Nous tenon à remercier le professeur Douadi MIHOUBI qui est aidé à réaliser ce travail et pour avoir accepté de juger ce travail et de présider le jury.

- Nous remercions très sincèrement, les membres de jury d'avoir bien voulu accepter de faire partie de la commission d'examinateur monsieur professeur Abdelmadjid BOUDAOUD. Nous espérons surtout qu'ils ont éprouvé du plaisir à lire ce travail.

- Nos remerciements vont également à tous les enseignants de Département mathématiques et informatiques.

- Je tiens ici à exprimer mes sentiments respectueux à mes chers parents à qui depuis de si longues années, m'ont encouragé et soutenu dans la poursuite de mes études.

- Un grand merci à ma famille, à mes proches et à mes collègues.

- Et enfin nous voulons remercier tous ceux qui nous ont aidés de près ou de loin on l'élaboration et la finalisation de ce travail.

Merci infiniment à tous.

M'sila, Juin 2023

Faris Badr Djaidja

Notations

A : Alphabet fini .

A^* : Monoïde libre sur A.

$|w|$: Longueur du mot w.

$|w|_\alpha$: Nombre d'occurrence de la lettre α dans le mot w.

L : Langage sur l'alphabet A.

$P(A^*)$: Ensemble des langages sur A.

X_M : Code de Morse.

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

Q^Q : Ensemble des applications d'un ensemble Q vers lui mème

$\overset{*}{\underset{\Re}{\longleftrightarrow}}$: Congruence engendrée par $\Re$.

$\Sigma^*/\overset{*}{\underset{\Re}{\longleftrightarrow}}$: Monoïde quotient.

Id_A : Morphisme identité sur A.

Table des matières

Introduction

Les séries génératrices sont des outils algébriques qui permettent 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 $(a_n)_{n\in\mathbb{N}}$ dont on cherche à déterminer le $n - ième$ terme.
En particulier, la série génératrice d'une suite finie est un polynôme. [16]

Les systèmes de traitement de l'information ont toujours utilisé des techniques de codage pour différents buts : protection contre les erreurs, représentation de l'information en mémoire d'un ordinateur, compression de l'information, cryptage, etc.

On distingue deux grandes familles de codes : les codes à longueurs constantes, les codes à longueurs variables.

Les codes à longueurs variables constituent une classe d'objets très importante, comme témoignent les différents domaines dans lesquels ils furent introduits : en théorie de l'information par SHANNON dans les années 1950, dans la théorie des évènements récurrents par FELLER (1950), dans la théorie des langages formels par SCHUTZENBERGER (1956). [12]

Ce travail est composé de trois chapitres :

Le premier chapitre, consiste à un rappel des notions élémentaires sur les monoïdes, les mots et langages.

Dans le second chapitre, nous allons étudier certaines propriétés des séries génératrices.

Dans le troisième chapitre, on fait une étude sur les codes à longueurs variables et leurs séries génératrices.

Chapitre 1

Préliminaires

1.1 Généralités sur les monoïdes

1.1.1 Monoïde

Définition 1.1.1.

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

— *L'opération"." est associative* :

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

— *Il existe un élément neutre* $1_M \in M$ *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* $m \cdot m' = m' \cdot m = 1_M$.

Exemple 1.1.1.

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

2- *L'ensemble des applications d'un ensemble* Q *vers lui même*
$Q^Q = \{f \ / \ Q \longrightarrow Q\}$ *muni de la composition des applications est un monoïde dont l'application identique noté* 1_Q *est l'élément neutre.*

Remarque 1.1.1.

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

Remarque 1.1.2.

Tout groupe est un monoïde mais l'inverse n'est pas toujours vrai.

Définition 1.1.2. *[10]*

Soit $(M, \cdot, 1_M)$ un monoïde. Un sous-monoïde de M est un triplet $(N, \cdot, 1_N)$ tel que :

1. $N \subseteq M$;

2. $1_M = 1_N$;

3. $\forall m, m' \in N : m \cdot m' \in N$.

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 monoïdes de $(M, \cdot, 1_M)$ qui contiennent Y.

Exemple 1.1.2.

Tout monoïde M admet deux sous monoïde, M et $\{1_M\}$, appelés sous monoïdes triviaux.

Exemple 1.1.3.

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 1.1.3. *[3]*

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 $y^{-1}X = \bigcup_{x \in X} y^{-1}x$.

Exemple 1.1.4.

Soit M le monoïde donné par la table ci-dessous

$\cdot$	1	0	x	y	t
1	1	0	x	y	t
0	0	0	0	0	0
x	x	0	x	0	0
y	y	0	t	y	t
t	t	0	t	0	0

On calcule les quotients à gauche : $y^{-1}x$ et $y^{-1}\{0,x,t\}$.

$y^{-1}\cdot x = \{z \in M; y \cdot z = x\} = \emptyset$,

$y^{-1}\{0,x,t\} = y^{-1}\cdot 0 \cup y^{-1}\cdot x \cup y^{-1}\cdot t$,

$y^{-1}\cdot 0 = \{0\}$, $y^{-1}\cdot x = \{\emptyset\}$, $y^{-1}\cdot t = \{x,t\}$,

Donc , $y^{-1}\{0,x,t\} = \{0,x,t\}$.

Définition 1.1.4. *[10]*

Soit $(M,\cdot,1_M)$ un monoïde, une congruence $\equiv$ sur $(M,\cdot,1_M)$ est une relation d'équivalence stable par la multiplication à droite et à gauche, i.e, :

$\forall a,b,c \in M : a \equiv b \Rightarrow a\cdot c \equiv b\cdot c$ *et* $c\cdot a \equiv c\cdot b$

L'ensemble quotient $M/\equiv$ est alors naturellement muni d'une structure de monoïde dit le monoïde quotient.

La projection naturelle (la surjection canonique) de M dans $M/\equiv$ est notée p.

Notons que si $h : M \longrightarrow N$ est un morphisme, alors la congruence associée à h, notée $\equiv_h$, est définie par : pour tous $u,v \in M, u \equiv_h v \Longleftrightarrow h(u) = h(v)$.

Notons que si $\Re$ une relation sur M qui vérifie $h(r) = h(s)$ pour tout $(r,s) \in \Re$, alors il existe un unique morphisme $\psi : M/\overset{*}{\underset{\Re}{\leftrightarrow}} \longrightarrow N$ tel que $\psi \circ p = h$ où $\overset{*}{\underset{\Re}{\leftrightarrow}}$ est la congruence engendré par $\Re$ et p est la surjection canonique.

Exemple 1.1.5.

Soit le monoïde $(\mathbb{N},+)$ et soit la relation $\equiv$ définie par $x \equiv y$ si et seulement si x et y ont même parité. La relation $\equiv$ est une congruence. Le quotient de $\mathbb{N}$ par cette relation donne un monoïde comprenant deux éléments, notés $\bar{0}$ et $\bar{1}$ correspondant respectivement aux entiers pairs et impairs.

Définition 1.1.5. *[10]*

Soit $(M,\cdot,1_M)$ un monoïde et A un ensemble de générateurs de M.

Le graphe de Cayley (gauche) de M par rapport á A est le graphe (M, E) óu $E = \{(x, a, y) \in M \times A \times M : y = a \cdot x\}$.

Exemple 1.1.6.

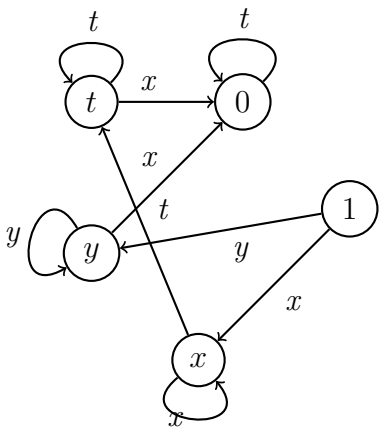
D'après l'exemple 1.1.4.

On a : $y^{-1} \cdot x = \emptyset$, $y^{-1} \cdot t = \{x, t\}$, $x^{-1} \cdot 0 = \{y, t, 0\}$,

$t^{-1}\{0, x, t\} = t^{-1} \cdot 0 \cup t^{-1} \cdot x \cup t^{-1} \cdot t$, $t^{-1} \cdot 0 = \{y, t, 0\}$, $t^{-1} \cdot x = \emptyset$, $t^{-1} \cdot t = \{1, x\}$,

$t^{-1}\{0, x, t\} = \{y, t, 0, 1, x\}$.

On trace le graphe de Cayley (gauche) de M par à A,



Définition 1.1.6.

Soit X un ensemble et M un monoïde, une application de $M \times X$ dans X est une action á gauche de M sur X si :

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

2. $\forall\ x \in X, \forall\ m_1, m_2 \in M : (m_1 \cdot m_2) \cdot x = m_1 \cdot (m_2 \cdot x)$.

Définition 1.1.7.

Soit $\equiv$ une congruence sur un monoïde M.

- *Une partie X de M est dite saturée par $\equiv$ si $\forall\ x \in X : \bar{x} \subseteq X$.*
- *On appelle partie saturée engendrée par une partie A de M et on note $Sat(A)$, la plus petite partie saturée par $\equiv$ de M contenant A.*

Proposition 1.1.1. *[10]*

Soit $\equiv$ une congruence sur un monoïde M.

1- *La réunion et l'intersection de parties saturées par $\equiv$ sont saturées.*

2- *Pour toute partie A de M, on a $Sat(A) = \bigcup_{x\in A} \bar{x}$.*

3- *Si $f : M \longrightarrow M'$ est un morphisme de monoïdes et si $\equiv$ la congruence sur M définie par : $x \equiv y \Leftrightarrow f(x) = f(y)$, alors $S \subseteq M$ est saturée par $\equiv$, si et seulement, si $f^{-1}(f(S)) = S$.*

Démonstration.

1- *Soient S et T deux parties de E saturées modulo R, i.e, $\bar{x} \subseteq S$ pour tout $x \subseteq S$ et $\bar{x} \subseteq T$ pour tout $x \subseteq T$. Il est claire que $S \cap T$ et $S \cup T$ sont des parties saturées modulo R.*

2- *Pour montrer que $Sat(A) = \bigcup_{x\in A} \bar{x}$, il suffit de vérifier que :*
- *$A \subseteq \bigcup_{x\in A} \bar{x}$;*
- *$\bigcup_{x\in A} \bar{x}$ est une partie de E saturée modulo R ;*
- *$\bigcup_{x\in A} \bar{x}$ est minimale au sens de l'inclusion.*

1.1.2 Homomorphisme de monoïdes

Définition 1.1.8.

Un morphisme (ou encore homomorphisme) d'un monoïde M dans un monoïde N est une application h tel que :

1- *$\forall\ u, v \in M : h(uv) = h(u)h(v)$.*

2- *$h(1_M) = 1_N$.*

-Un isomorphisme de monoïdes est un homomorphisme bijectif de monoïdes.

Exemple 1.1.7.

L'application $h : (\mathbb{R}, +) \longrightarrow (\mathbb{R}\backslash\{0\}, \times)$, $x \longrightarrow a^x$, tels que a un élément fixé de $\mathbb{R}\backslash\{0\}$ est un homomorphisme de $(\mathbb{R}, +, 0)$ dans $(\mathbb{R}\backslash\{0\}, \times, 1)$.

Exemple 1.1.8.

La fonction exponentielle représente un isomorphisme de $(\mathbb{R}, +)$ dans $(\mathbb{R}_+ - \{0\}, \times)$. Elle est bijective et vérifie : $exp(x + y) = exp(x) \times exp(y)$ et $exp(0) = 1$.

Définition 1.1.9.

Une partie P d'un monoïde M est appelé une base de M si tout élément de M admet une unique décomposition comme produit d'éléments de P. Une base est donc

en particulier une partie génératrice de M. Elle est unique à l'ordre prés.

Un monoïde est dit libre s'il admet une base. Dans ce cas, la base est unique.

Propriété 1.1.1.

Soit A un ensemble (appelé par fois alphabet).

1. *L'ensemble des suites finies d'éléments de A (appelées mots) muni de l'opération de concaténation est un monoïde libre noté A^* et appelé monoïde libre sur A.*

2. *Deux monoïdes libres sur des alphabets finis sont isomorphes si et seulement si leurs bases ont même cardinal.*

Remarque 1.1.3.

- Un morphisme de monoïdes libres est entièrement défini par l'image des lettres :

- $h(\epsilon) = \epsilon$.
- $h(a_1 \dots a_n) = h(a_1) \dots h(a_n)$.

- Un mot ω est un point fixe d'un morphisme f, si $f(\omega) = \omega$.

- Le morphisme identité sur A est noté Id_A.

Exemple 1.1.9.

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

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

Exemple 1.1.10. *[21]*

Fonction de Parikh : Soit $A = \{a_1, a_2, \dots, a_n\}$ un alphabet, de cardinal $n \geq 1$, et ordonné (avec $a_1 \leq a_2 \leq \dots \leq a_n$). On définit alors la fonction de Parikh par :

$$\begin{aligned} \Psi &: A^* \longrightarrow N^n \\ \Psi(\omega) &= (|\omega|_{a_1}, \dots, |\omega|_{a_n}) \end{aligned}$$

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

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

Proposition 1.1.2. *[2]*

Tout fonction $\tilde{\psi} : A \longrightarrow M$ de A dans un monoïde M se prolonge de façon unique en un morphisme de A^* dans M.

Démonstration.

• *L'existence* : *Posons* $\tilde{\psi}(\epsilon) = 1_M$ *et* $\tilde{\psi}(a_1a_2\dots a_n) = \psi(a_1)\psi(a_2)\dots\psi(a_n)$, $n \in N$, $1 \leq i \leq n$, $a_i \in A$.

Il est facile de voir que $\tilde{\psi}$ *est bien un homomorphisme.*

• *L'unicité* : *Soient* $\tilde{\psi}$ *et* $\tilde{\vartheta}$ *deux homomorphismes de* A^* *dans* M *tels que* :

$\forall a \in A, \tilde{\psi}(a) = \tilde{\vartheta}(a)$.

Alors $\tilde{\psi}(\epsilon) = \tilde{\vartheta}(\epsilon) = 1_M$ *et pour tout mot* $\omega = a_1a_2\dots a_n$

On a $\tilde{\psi}(\omega) = \tilde{\psi}(a_1a_2\dots a_n) = \psi(a_1)\psi(a_2)\dots\psi(a_n) = \tilde{\vartheta}(a_1a_2\dots a_n) = \tilde{\vartheta}(\omega)$.

1.2 Mots et langage

On introduit dans ce paragraphe quelques définitions, propriétés et notations concernant les mots et les langages.

1.2.1 Mots

Définition 1.2.1.

- *Un alphabet est un ensemble fini* Σ *, les éléments de* Σ *sont appelés lettres ou symboles. Ainsi* $T = \{a, b, c, d\}$ *,* $\Omega = \{0, 1\}$ *sont des alphabets.*

- *Un mot* ω *sur l'alphabet* Σ *est une suite fini* $\sigma_1\sigma_2...\sigma_n$ *de lettres de* Σ.

L'entier n est appelé la longueur de ω *notée* $|\omega|$. *on note* $|\omega|_\alpha$ *le nombre de d'occurrence de la lettre* α *dans le mot* ω. *Si l'on note* $\omega = \sigma_1\sigma_2...\sigma_n$:

$$|\omega|_\sigma = card\{i \in \{1, 2, ..., k\} : \sigma_i = \sigma\}.$$

- 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.

- La concaténation de deux mots $u = u_1\,u_2...u_m$ et $v = v_1\,v_2...v_n$ est le mot noté $u.v$ où uv et égal à $u_1\,u_2...u_mv_1\,v_2...v_n$ obtenu simplement par juxtaposition.

Propriété 1.2.1.

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

1. $\forall u, v, w \in A^*, (u\cdot v)\cdot w = u\cdot(v\cdot w)$ (*La loi est associative*).

2. $\forall u \in A^*, u\cdot \epsilon = \epsilon \cdot u = u$ *(Le mot vide* ϵ *est élément neutre).*

3. $\forall u, v \in A^*, |u\cdot v| = |u| + |v|$.

4. $\forall u \in A^*, u\cdot u = u \Leftrightarrow u = \epsilon$ *(Le mot vide* ϵ *est le seul mot idempotent).*

5. *La concaténation n'est pas commutative.*

Exemple 1.2.1.

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.

Proposition 1.2.1.

Soit X *une partie de* A^*. *Le sous-monoïde de* A^* *engendré par* X, *noté* X^* *est défini par* :

$$X^* = \{x_1 \dots x_n : n \in \mathbb{N}, \forall\ 1 \leqslant i \leqslant n, x_i \in X\}.$$

Proposition 1.2.2.

Soit Σ *un alphabet quelconque le monoïde* Σ^* *possède les deux propriétés suivantes* :

1. *Tout élément de* Σ^* *est une suite finie d'éléments de* Σ.

2. *Deux suites distinctes d'éléments de* Σ *définissent deux éléments distincts de* Σ^*.

- La propriété (1) *distingue le monoïde* Σ^* *par exemple du monoïde* $(\Sigma \cup \{\sigma\})^*$, *avec* $\sigma \notin \Sigma$.

- La propriété (2) *distingue le monoïde* $\{\alpha, \beta\}^*$ *par exemple du monoïde commutatif* M *obtenu en postulant que l'opération de concaténation est commutative* :
Les deux mots $\alpha\beta$ *et* $\beta\alpha$ *définissent alors le même élément de* M.

- Σ^* *est le seul monoïde satisfaisant les propriétés* (1) *et* (2), *on dit que* Σ^* *est le monoïde libre sur* Σ. *On dit que* Σ *est une base de* Σ^*.

1.2.2 Facteurs et sous mots

Si u et v sont deux mots, on dit que u est un :

- préfixe (ou facteur gauche) de v s'il existe un mot $w \in \Sigma^*$ tel que $v = uw$;
- suffixe (ou facteur droit) de v s'il existe un mot $w \in \Sigma^*$ tel que $v = wu$;
- facteur de v lorsqu'il existe deux mots $x \in \Sigma^*$ et $y \in \Sigma^*$ tel que $v = xuy$. [1]

Les préfixes et suffixes sont dits propres lorsque $w \neq \epsilon$, les facteurs sont dits propres lorsque $x \neq \epsilon$ ou $y \neq \epsilon$.

Enfin, un sous-mot d'un mot $u = a_1 \dots a_n$ de longueur n (les a_i désignant ses lettres) est un mot $v = a_{\varphi(1)} \dots a_{\varphi(p)}$ de longueur p, où $\varphi : [1, p] \longrightarrow [1, n]$ est une application strictement croissante.

Exemple 1.2.2.

- *"bon" est un facteur gauche du mot "bonjour".*
- *"jour" est un facteur droit du mot "bonjour".*
- *"on" est un facteur propre du mot "bonjour".*

1.2.3 Quelques propriétés combinatoires élémentaires

Lemme (Levi)

Soient x, y, u et $v \in \Sigma^*$ tel que $uv = xy$. Alors il existe un unique mot $t \in \Sigma^*$ tel que l'une des deux conditions suivantes soit réalisée :

$u = xt$ et $y = tv$.

$x = ut$ et $v = ty$.

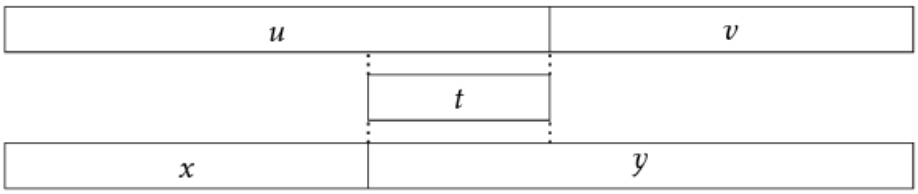


FIGURE 1.1 – Illustration du lemme de Levi lorsque |u|⩾|x|.

Démonstration.

Supposons par exemple $|u| \geqslant |x|$. x est un préfixe de uv donc de u, ce qui justifie l'existence d'un mot t tel que $u = xt$. Dans ce cas, l'égalité $uv = xy$ peut encore s'écrire $xtv = xy$ puis en simplifiant $tv = y$.

Le cas $|x| \geqslant |u|$ se traite de la même façon.

Ce résultat est utilisé pour démontrer deux autres résultats élémentaires qui découlent de la non-commutativité de la concaténation. [20]

1.2.4 Distances entre les mots

Il existe plus d'une façon de munir l'ensemble Σ^* d'une distance ; l'une d'entre-elles consiste à considérer le plus long préfixe commun à deux mots u et v, préfixe que nous allons désormais noter $\text{plpc}(u, v)$. [20]

$$d(u, v) = |uv| - 2|plpc(u, v)|.$$

On obtient ainsi une distance, appelée distance préfixe. On vérifie en effet que :

- $d(u, v) \geqslant 0$;
- $d(u, v) = 0 \Longleftrightarrow u = v$;
- $d(u, w) \leqslant d(u, v) + d(v, w)$.

Démonstration.

De ces trois propriétés seule la troisième demande peut-être une justification.

Après simplification, celle-ci revient à prouver que :

$$|\text{plpc}(u, v)|+|\text{plpc}(v, w)|\leqslant|\text{v}|+|\text{plpc}(u, w)|$$

Le plus court des deux préfixes $\text{plpc}(u, v)$ et $\text{plpc}(v, w)$ est commun à u,v et w donc

$$\min(|\text{plpc}(u, v)|,|\text{plpc}(v, w)|)\leqslant |\text{plpc}(u, w)|.$$

Les deux préfixes $\text{plpc}(u, v)$ et $\text{plpc}(v, w)$ sont préfixes de $|v|$ donc

$$\max(|\text{plpc}(u, v)|,|\text{plpc}(v, w)|)\leqslant |v|.$$

D'où le résultat, en additionnant ces deux inégalités. [20]

1.2.5 Langage

Définition 1.2.2.

On appelle langage défini sur un alphabet A, tout sous-ensemble (fini ou infini) de A^*. L'ensemble des langages sur A^* est donc : [2]

$$P(A^*) = \{L, L \subseteq A^*\}$$

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

Deux langages particuliers sont indépendants de l'alphabet A

- *Le langage vide* ($L = \emptyset$).

- *Le langage contenant le seul mot vide* ($L = \{\epsilon\}$).

Exemple 1.2.3.

On considère l'alphabet $A = \{a, b\}$.

- $L_1 = \{ab, a, ba, bb\}$;
- $L_2 = \{\omega \in \{a, b\}^* / |\omega| \geqslant 3\}$;
- $L_3 = \{\omega \in \{a, b\}^* / |\omega| \equiv 0\,[5]\}$;
- $L_4 = \{\omega \in \{a, b\}^* / |\omega|_a \equiv 0\,[3]\}$;
- $L_5 = \{a^n / n \geqslant 0\}$;
- $L_6 = \{a^i b^j / i \geqslant 0, j \geqslant 1\}$;
- $L_7 = \{a^i b^i / i \geqslant 0\}$;
- $L_8 = \{a^i b^j a^j / i \geqslant j \geqslant 1\}$.

Sont des langages sur $A = \{a, b\}$. *[2]*

Définition 1.2.3.

Soient L,$M \subseteq \Sigma^*$ *deux langages. La concaténation des langages* L *et* M *est le langage,*

$$LM = \{uv, u \in L, v \in M\}$$

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

$$L^n = \{\omega_1\omega_2 \dots \omega_n, \forall\ i \in \{1, 2 \dots n\}, \omega_i \in L\}.$$

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

Remarque 1.2.1.

On voit facilement que L^* *est le petit langage contenant* L *et le mot vide et qui soit stable par concaténation.*

Exemple 1.2.4.

Soient les deux langages $L = \{u \in \Sigma^* : |u|$ *est paire* $\}$ *et* $K = \{u \in \Sigma^* : |u|$ *est impaire* $\}$.

On a alors les égalités suivantes :

$LK = KL = K; LL = L; KK = L - \{\epsilon\}$.

Proposition 1.2.3.

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

- *$P(A^*)$ est un monoïde libre pour la concaténation où $\{\epsilon\}$ est l'élement neutre.*

Propriété 1.2.2.

En plus des propriétés des opérateurs ensemblistes qui sont toujours valables pour les langages, nous définissons les propriétés supplémentaires suivantes : [1]

- *La concaténation des langages n'est pas idempotente. cela signifie on toujours que $\exists L \neq \epsilon / L.L \neq L$;*
- *La concaténation des langages est associative ;*
- *La concaténation des langages n'est pas commutative ;*
- *La concaténation des langages est distributive par rapport à l'union ;*
- *La concaténation des langages n'est pas distributive par rapport à l'intersection ;*
- $L^* = (L^*)^*$;
- $L^* = L^*.L^*$;
- $(L_1 + L_2)^* = (L_1^*.L_2^*)^* = (L_1^* + L_2^*)^*$.

1.2.6 Opérations sur les langages

Les langages sont des ensembles, par conséquent on peut leur appliquer toutes les opérations appliquées sur les ensembles ; toute fois, il existe des opérations qui leurs sont spécifiques, il s'agit d'une extension des opérations définies sur les mots.[1] Soient L,L_1 et L_2 trois langages définis respectivement sur les trois alphabets X,X_1 et X_2 :

- L'union : $L_1 \cup L_2 = L_1 + L_2 = \{w / w \in L_1 \vee w \in L_2\}$;
- L'intersection : $L_1 \cap L_2 = \{w / w \in L_1 \wedge w \in L_2\}$;
- Le complément : $\bar{L}_1 = \{w / w \in X_1^* \wedge w \notin L_1\}$;
- La concaténation des langages : $L_1.L_2 = \{w = w_1.w_2 / w_1 \in X_1 \wedge w_2 \in X_2\}$;
- La puissance concaténative : $L^n = L.L...L$ (n fois L). On peut le définir par induction comme suit : $L^0 = \{\epsilon\}, L^1 = L$ et $L^n = L.L^{n-1}$;

- La fermeture itérative ou l'étoile de Kleen : $L^* = L^0 \cup L^1 \cup \ldots \cup L^n$ (n tend vers l'infini) ;
- La fermeture itérative propre (l'étoile propre) : $L^+ = L^1 \cup L^2 \cup \ldots \cup L^n$ (n tend vers l'infini) ;

Exemple 1.2.5.

Soient $A = \{a, b\}$ un alphabet, $L_1 = \{a\}$,$L_2 = \{ab\}$ et $L_3 = A$ trois langages sur A. On a $L_1^* = \{a^n, n \geq 0\}, L_2^* = \{(ab)^n\}$ et $L_3^* = A^*$.

Définition 1.2.4.

On dit qu'un mot $u \in \Sigma^*$ est facteur de $w \in \Sigma^*$ s'il existe deux mots $f, g \in \Sigma^*$ tels que $w = fug$.

Exemple 1.2.6.

Soit l'alphabet $\Sigma = \{a, b, c\}$, le mot $w = aabc$, alors ab est un facteur de w, mais ac ne l'est pas.

Chapitre 2

Séries génératrices

2.1 Définitions et notations

Définition 2.1.1.

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

$$a_0 + a_1 x + a_2 x^2 + ... = \sum_{k\geq 0}^{\infty} a_k x^k.$$

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

Définition 2.1.2.

Une fonction génératrice est une série formelle à coefficients dans un corps $\mathbb{F}$*, avec* $\mathbb{F} = \mathbb{R}$ *ou* $\mathbb{C}$*.*

$$f(x) = \sum_{k\geq 0}^{\infty} a_k x^k$$

Définition 2.1.3.

Soient r *un réel et* k *un entier naturel. Alors on définit le coefficient binomial généralisé* $\binom{r}{k}$ *par : [16]*

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

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

Exemple 2.1.1.

Soit la somme $C(n,k) = \sum_{a_1+\cdots+a_k=n} a_1 \ldots a_k$*, avec* $S(x) = \sum_{n\geq 1} nx^n$*.*
On calculer $C(n,k)$*, donc comme* $S(x)^k = \sum_{n\geq 1} C(n,k)x^n$*, d'ou* $C(n,k) = C_{2k-1}^{n+k-1}$*.*

Proposition 2.1.1.

Soient n et k des entiers naturels. Alors [16]

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

Exemple 2.1.2.

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* $f(x) = \sum\limits_{n\geq 0} x^n = 1+x+x^2+\dots+\dots$

On remarque alors

$xf(x) = x\sum\limits_{n\geq 0} x^n = x(1+x+x^2+\dots+\dots) = x+x^2+x^3+\dots+\dots = f(x)-1.$

i.e, $xf(x) = f(x)-1$. *Alors* $(1-x)f(x) = 1$. *Donc*

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

3. *Soit la suite* $(1, a, a^2, a^3, \dots)$, *on a* $\sum\limits_{n\geq 0} x^n = 1+x+x^2+\cdots = \frac{1}{1-x}$ (1)

On substitue x par ax dans (1), *on obtient* :

$\sum\limits_{n\geq 0} (ax)^n = 1+ax+(ax)^2+\cdots+(ax)^n+\cdots = \frac{1}{1-ax}$

$$\sum_{n\geq 0} (ax)^n = \frac{1}{1-ax}$$

En particulier, si $a = -1$, *soit la suite* $(1, -1, 1, -1, \dots)$.

On obtient $\sum\limits_{n\geq 1} (-1)^n (x)^n = \frac{1}{1+x}$

Remarque 2.1.1.

Les fonctions génératrices transforment les problémes des séquences en probleme de fonctions, c'est une excellente chose car nous disposons d'un grand nombre des machines mathématiques pour manipuler les fonctions.

2.1.1 Opérations sur les séries génératrices

Soient $f(x) = \sum_{n\geq 0} a_n x^n$ et $g(x) = \sum_{n\geq 0} b_n x^n$

• La somme de deux séries génératrices se définit de manière assez évidente en sommant les suites correspondantes. [16]

$$\left(\sum_{n\geq 0} a_n x^n\right) + \left(\sum_{n\geq 0} b_n x^n\right) = \sum_{n\geq 0}(a_n + b_n)x^n.$$

• Le produit, 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 :

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

Exemple 2.1.3.

Soient $A(x) = 1 + x + x^2 + x^3 + \ldots$, *et* $B(x) = 2 + 3x + 4x^2 + 5x^3 + \ldots$

1- *L'Addition de séries génératrices* :

$C(x) = A(x) + B(x) = (1+2) + (1+3)x + (1+4)x^2 + (1+5)x^3 + \ldots$

Ainsi, $C(x) = 3 + 4x + 5x^2 + 6x^3 + \ldots$

2- *La Multiplication de séries génératrices* :

$C(x) = A(x) * B(x) = (1*2) + (1*3+2*1)x + (1*4+2*3+3*1)x^2 + (1*5+2*4+3*3+4*1)x^3 + \ldots$

Nous obtenons, $C(x) = 2 + 5x + 10x^2 + 17x^3 + \ldots$

3- *La Dérivation de série génératrice* :

$A'(x) = 0 + 1 + 2x + 3x^2 + \ldots$

Ainsi, $A'(x) = 1 + 2x + 3x^2 + \ldots.$

Notation 2.1.1.

Soit $(g_n)_{n\geq 0}$ *une séquence de nombres, la fonction génératrice associée à cette séquence est la série* : $G(x) = \sum_{n\geq 0} g_n x^n$.
La correspondance entre une séquence et sa fonction génératrice avec une flèche double face comme suit :

$$\langle g_0, g_1, g_2, g_3, \ldots\rangle \longleftrightarrow G(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 \ldots.$$

Propriété 2.1.1.

Soit $\langle g_0, g_1, g_2, g_3, \ldots\rangle \longleftrightarrow G(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 \ldots..$ *On a* :

$$\left\langle \overbrace{0, 0, \ldots, 0}^{k\ zeros}, g_0, g_1, g_2, g_3, \ldots, \right\rangle \longleftrightarrow x^k G(x).$$

Exemple 2.1.4. *[18]*

Le décalage de la séquence $(1, 1, 1, 1, 1, 1, \ldots)$ *de* k *positions donne la séquence* :
$(0, 0, \ldots, 0, 1, 1, 1, 1, 1, 1, \ldots) \longleftrightarrow x^k + x^{k+1} + x^{k+2} + \cdots = x^k(1 + x + x^2 + \ldots) = \frac{x^k}{1-x}$.

Proposition 2.1.2. *[7]*

La suite des nombres de Fibonacci est générée par la fonction :

$$F(t) = \frac{t}{1 - t - t^2}.$$

Démonstration

On sait que la suite de Fibonacci est définie par :

$$\begin{cases} f_0 = 0; \\ f_1 = 1; \\ f_{n+2} = f_{n+1} + f_n, n \geqslant 2. \end{cases}$$

$$\begin{aligned} F(t) &= \sum_{n\geq 0} F_n t^n = \sum_{n\geq 0}(F_{n+2} - F_{n+1})t^n \\ &= \sum_{n\geq 0} F_{n+2}t^n - \sum_{n\geq 0} F_{n+1}t^n \\ &= \frac{1}{t^2}\sum_{n\geq 0} F_{n+2}t^{n+2} - \frac{1}{t}\sum_{n\geq 0} F_{n+1}t^{n+1} \\ &= \frac{1}{t^2}\sum_{n\geq 2} F_n t^n - \frac{1}{t}\sum_{n\geq 1} F_n t^n \\ &= \frac{1}{t^2}(F(t) - t) - \frac{1}{t}F(t) \end{aligned}$$

Ainsi :
$$\left(1 + \frac{1}{t} - \frac{1}{t^2}\right) F(t) = \frac{-1}{t},$$

$$\frac{t^2 + t - 1}{t^2}\ F(t) = \frac{-1}{t}.$$

Enfin :
$$F(t) = \frac{t}{1 - t - t^2}.$$

Théorème 2.1.1.

Soit la suite $(G_n)_{n\in\mathbb{N}}$ *définie par la relation de récurrence suivante* :

$$\begin{cases} G_n = pG_{n-1} + qG_{n-2}, \quad n \geq 2 \\ G_0 = \alpha, G_1 = \beta. \end{cases}$$

Avec $p,q \in \mathbb{R}_+^*$ *et* $\alpha, \beta \in \mathbb{C}$. *[6]*

Alors la fonction génératrice associée á $(G_n)_{n\in\mathbb{N}}$ *est donnée par* :

$$G(t) = \frac{\alpha + (\beta - p\alpha)^t}{1 - pt - qt^2}.$$

Démonstration

On a :

$$\begin{aligned} G(t) &= \sum_{n=0}^{\infty} G_n t^n \\ &= G_0 + G_1 t + \sum_{n=2}^{\infty} G_n t^n \\ &= \alpha + \beta t + \sum_{n=2}^{\infty} (pG_{n-1} + qG_{n-2}) t^n \\ &= \alpha + \beta t + pt \sum_{n=2}^{\infty} G_{n-1} t^{n-1} + qt^2 \sum_{n=2}^{\infty} G_{n-2} t^{n-2} \\ &= \alpha + \beta t + pt \sum_{n=1}^{\infty} G_n t^n + qt^2 \sum_{n=0}^{\infty} G_n t^n \\ &= \alpha + \beta t + pt \left(\sum_{n=0}^{\infty} G_n t^n - \alpha \right) + qt^2 \sum_{n=0}^{\infty} G_n t^n \\ &= \alpha + (\beta - \alpha p)t + ptG(t) + qt^2 G(t). \end{aligned}$$

Alors : $G(t)(1 - pt - qt^2) = \alpha + (\beta - p\alpha)t.$

D'où : $$G(t) = \frac{\alpha + (\beta - p\alpha)t}{1 - pt - qt^2}.$$

Corollaire 2.1.1.

D'après le théorème précédent on déduit la fonction génératrice des nombres de k Fibonacci :

• *Pour* $\alpha = k$, $\beta = q = 1$, $p = k$, *On obtient :*

$$G(t) = \frac{1}{1 - kt - t^2}.$$

• *Pour* $k = 1$ *on obtient la fonction génératrice des nombres de Fibonacci est donnée par :*

$$G(t) = \frac{1}{1 - t - t^2}.$$

Chapitre 3

Codes à longueurs variables

3.1 Définitions et Notations

Définition 3.1.1.

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 $x_1, x_2, ..., x_n, y_1, y_2, ..., y_m \in X$ on a :

$$x_1x_2...x_n = y_1y_2...y_m \Longrightarrow n = m \text{ et } x_i = y_i \text{ pour } i = 1...n.$$

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

Exemple 3.1.1. *[3]*

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 $w = aba$, les deux décompositions $(ab)a = a(ba)$.*
2. *Le code de Morse X_M est un code sur l'alphabet $\{., \wedge, -\}$ dont les mots sont mis en correspondance avec les lettres $a, b, ...z$. 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.

Propriété 3.1.1.

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 $\phi : B^* \longrightarrow A^*$ qui induit une injection de B sur X est injectif .*

Réciproquement, s'il existe un morphisme injectif $\phi : B^* \longrightarrow A^*$ 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 [3]

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 que est unique.

iii. Soit $\phi : B^* \longrightarrow A^*$ qui induit une bijection de B sur X.
Soient $u, v \in (B^*)^2$ 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 $u = b_1...b_n$ et $v = b'_1...b'_m$. On a alors $\phi(b_1)...\phi(b_n) = \phi(b'_1)...\phi(b'_m)$ avec $\phi(b_i), \phi\left(b'_j\right) \in X$. Or X est un code donc $n = m$ et $\forall i\ \phi(b_i) = \phi\left(b'_j\right)$ or ϕ induit une injection de B sur X donc $\forall i\ b_i = b'_j$ i.e., $u = v$. Donc ϕ est injective.

Réciproquement, soit $\phi : B^* \to A^*$ morphisme injectif, supposons qu'on a $n, m \in \mathbb{N}$ et $(x_i)_{i=1...n}, \left(x'_j\right)_{j=1...m} \in X = \phi(B)$ tels que $x_1x_2...x_n = x'_1x'_2...x'_m$.
Soient $(b_i)_{i=1...n}, \left(b'_j\right)_{j=1...m} \in B$ tels que $\forall i\ x_i = \phi(b_i)$ et $\forall j\ x_j = \phi\left(b'_j\right)$.
On a donc $\phi(b_1...b_n) = \phi(b'_1...b'_m)$ or ϕ est injective donc $b_1...b_n = b'_1...b'_m$.
D'ou $n = m$ et $\forall i\ b_i = b'_j$ et donc $\forall i\ x_i = \phi(b_i) = \phi\left(b'_j\right) = x'_j$.

Corollaire 3.1.1. *[3]*

Soit $\phi : B^* \longrightarrow A^*$ 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 3.1.1.

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

$$X \text{ est un code} \Longleftrightarrow (\forall f \in A^* : (X^*f \cap X^* \neq \emptyset) \text{ et } (fX^* \cap X^* \neq \emptyset)) \Rightarrow f \in X^*$$

Démonstration.

Supposons que X vérifie la condition (1) et X ne soit pas un code.
Il existe $a_{i_1}...a_{i_n}$, $a_{j_1}...a_{j_m} \in X$ tels que $a_{i_1}...a_{i_n} = a_{j_1}...a_{j_m}$ avec $a_{i_1} \neq a_{j_1}$,
n ou $m \neq 1$ sans quoi X ne serait pas un système minimal de générateur pour X^*.
On peut toujours supposer qu'il n'existe pas $n' < n$ et $m' < m$ tels que,
$a_{i_1}...a_{i_{n'}} = a_{j_1}...a_{j_{m'}}$, 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
$hX^* \cap X^* \neq \emptyset$ puisque $a_{i_1}h_{j_2}...a_{j_m} = a_{i_1}...a_{i_n}$ on peut déduire $ha_{j_2}...a_{j_m} = a_{i_2}...a_{i_n}$.
Or $h \notin X^*$ce qui contredit la condition (1).
Réciproquement, supposons que X ne vérifie pas la condition (1). Il existe alors, $f \notin X^*$ tels que $a_{i_1}...a_{i_n}f = a_{j_1}...a_{j_m}, a_{i_1} \neq a_{j_1}$ et $fa_{k_1}...a_{k_p} = a_{l_1}...a_{l_q}$, mais alors,

$$a_{i_1}...a_{i_n}fa_{k_1}...a_{k_p} = a_{j_1}...a_{j_m}a_{k_1}...a_{k_p} = a_{i_1}...a_{i_n}a_{l_1}...a_{l_q} \text{ avec } a_{i_1} \neq a_{j_1}.$$

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

Définition 3.1.2.

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,

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

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 3.1.2.

$\forall X \subset A^*$, X préfixe $\Longrightarrow$ X 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 $(x_i)_{i=1...n}, \left(x'_j\right)_{j=1...m} \in X$ tels que $w = x_1x_2...x_n = x'_1x'_2...x'_m$. 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 3.1.2.

Soit $A = \{a, b\}$ et $X = \underset{n\geq 0}{\cup} a^n b A^n$, X est un préfixe car $a^n bu = a^m bv \Longrightarrow m = n$ et donc $u = v$. Par conséquent X est un code sur A. [3]

Définition 3.1.3.

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

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

Exemple 3.1.3.

$X_1 = \{aa, ab, bb, ba\}$ est un code maximal fini (en effet, si un code 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).

3.2 Série génératrice d'un code

Définition 3.2.1.

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

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

Proposition 3.2.1. *[9]*

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

Démonstration

Soient X un code et $f_X = \sum_{n\geq 0} u_n z^n$ sa sérié génératrice. La sérié 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

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

Exemple 3.2.1. *[11]*

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

- $u_0 = Card(X \cap A^0) = |\{b, ab\} \cap \{\varepsilon\}| = |\{\emptyset\}| = 0.$
- $u_1 = Card(X \cap A^1) = |\{b, ab\} \cap \{a, b\}| = |\{b\}| = 1.$
- $u_2 = Card(X \cap A^2) = |\{b, ab\} \cap \{aa, ab, ba, bb\}| = |\{ab\}| = 1.$
- $u_n = 0. \quad \forall n \geq 3.$

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

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

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* $u_n = Card(D^* \cap A^n)$*, mais si* n *est impair alors* $u_n = Card(D^* \cap A^n) = 0$*. Et puisque chaque mot de* D^* *de longueur* $2n$ *s'obtinet 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 $$f_{D^*}(z) = \sum_{n\geq 0} \binom{2n}{n} z^{2n} = \sum_{n\geq 0} \binom{-\frac{1}{2}}{n} (-4)^n z^{2n} = \sum_{n\geq 0} \binom{-\frac{1}{2}}{n} \left(-4z^2\right)^n = \left(1 - 4z^2\right)^{-\frac{1}{2}}.$$

D *est un code alors* $f_{D^*}(z) = \dfrac{1}{1 - f_D(z)}$*. Donc*

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

On calcule la série génératrice de l'ensemble D*.*

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

En utilisant la formule de Newton généralisée, on obtient

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

Où $\binom{\frac{1}{2}}{0} = 1$ *et* $\binom{\frac{1}{2}}{n} = \dfrac{\frac{1}{2}\left(\frac{1}{2}-1\right)\left(\frac{1}{2}-2\right)...\left(\frac{1}{2}-n+1\right)}{n!}$*. Alors*

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

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

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

Donc

$$\begin{aligned}
f_D(z) &= \sum_{n\geq 1}\frac{(2n-2)!2n}{n!n!}z^{2n}=\sum_{n\geq 1}\frac{(2n-2)!2n}{n(n-1)!n(n-1)!}z^{2n}=\sum_{n\geq 1}\frac{2}{n}\binom{2n-2}{n-1}z^{2n}\\
&= 2\sum_{n\geq 1}\frac{1}{n}\binom{2n-2}{n-1}z^{2n}.
\end{aligned}$$

3.3 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.

Proposition 3.3.1. *[22]*

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

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

On a alors : X *est un code* $\Longleftrightarrow\forall n\geq 1\ \varepsilon\notin U_n.$

Exemple 3.3.1. *[12]*

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

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

- $(00)^{-1} X = \{y \in \{a,b\}^* : (00)\, y \in X\} = \{\varepsilon\}$.
- $(010)^{-1} X = \{y \in \{a,b\}^* : (010)\, y \in X\} = \{\varepsilon\}$.
- $(101)^{-1} X = \{y \in \{a,b\}^* : (101)\, y \in X\} = \{\varepsilon\}$.
- $(11)^{-1} X = \{y \in \{a,b\}^* : (11)\, y \in X\} = \{\varepsilon\}$.

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

$U_2 = X^{-1}U_1 \cup U_1^{-1}X$ *tels que* $X^{-1}U_1 = \underset{x\in X}{\cup} x^{-1}U_1$ *et* $U_1^{-1}X = \underset{u_1\in U_1}{\cup} u_1^{-1}X$.

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

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

$U_1 = X^{-1}X \setminus \{\varepsilon\}$, $X^{-1}X = \underset{x\in X}{\cup} x^{-1}X$, *tels que* $x^{-1}X = \{y \in \{a,b\}^* : xy \in X\}$.

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

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

$U_2 = X^{-1}U_1 \cup U_1^{-1}X$ *tels que* $X^{-1}U_1 = \underset{x\in X}{\cup} x^{-1}U_1$ *et* $U_1^{-1}X = \underset{u_1\in U_1}{\cup} u_1^{-1}X$.

On a, $X^{-1}U_1 = \underset{x\in X}{\cup} x^{-1}U_1$ *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, $X^{-1}U_1 = \emptyset$.

$U_1^{-1}X = \underset{u_1\in U_1}{\cup} u_1^{-1}X = b^{-1}X$.

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

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 à longueur variable.*

3.4 Codes à groupes

Proposition 3.4.1. *[8]*

Soient G un groupe et H un sous groupe de G, $\psi : \Sigma^* \longrightarrow G$ un morphisme.

Posons $X^* = \psi^{-1}(H)$ avec X l'ensemble minimal générateur de X^*. Alors :

1- *X est un code bipréfixe.*

2- *Si ψ est surjectif, alors X est un code maximal bipréfixe.*

Remarque 3.4.1.

1- *Dans le cas où le morphisme ψ est surjectif, la base X de X^* (qui est donc toujours un code bipréfixe maximal) est nommée code à groupe. Nous dirons que c'est le code à groupe défini à partir de G, H et ψ, nous le noterons $X(G,H)_\psi$.*

2- *Soit $X(G,H)_\psi$ un code à groupe. Nous dirons que X est de degré d si $[G:H] = d$. Nous dirons que X est régulier si $H = \{1_G\}$.*

Exemple 3.4.1. *[11]*

1- *Considérons le morphisme de monoïdes $\psi : \{a,b\}^* \longrightarrow (\mathbb{Z},+)$ défini par :*

$\psi(a) = 1, \psi(b) = -1, \psi(\epsilon) = 0.$

Donc, $\forall w \in \{a,b\}^* : \psi(w) = |w|_a - |w|_b$.

L'application ψ est surjective car $\forall m \in \mathbb{Z}, \exists w \in \{a,b\}^*$ tels que $\psi(w) = m$.

On distingue les cas suivants :

1. *Si $m = 0$, alors $\psi(\epsilon) = 0$.*
2. *Si $m > 0$, alors $\psi(a^m) = m \cdot \psi(a) = m \cdot 1 = m$.*
3. *Si $m < 0$, alors $\psi(b^{-m}) = -m \cdot \psi(b) = -m \cdot (-1) = m$.*

Soit $H = \{0\}$ le sous groupe trivial de $(\mathbb{Z},+)$. Alors

$X^* = \psi^{-1}(\{0\}) = \{w \in \{a,b\}^* : \psi(w) = |w|_a - |w|_b = 0\} = \{w \in \{a,b\}^* : |w|_a = |w|_b\}$.

Donc X est infini car X contient les mots de la forme $a^n b^n, n > 0$, finalement, d'aprés la proposition 3.4, X est un code maximal bipréfixe.

2- *Soit $\psi : \Sigma^* \longrightarrow (\mathbb{Z}/n\mathbb{Z}, \oplus)$ le morphisme de monoïdes défini par :*

$\psi(\sigma) = \overline{1}$ pour tout $\sigma \in \Sigma$ et $\psi(\epsilon) = \overline{0}$.

Donc pour tout $w \in \Sigma^* : \psi(w) = |w| \; mod\,(n)$. L'application ψ est surjective car pour tout $\overline{m} \in \mathbb{Z}/n\mathbb{Z}$, il existe $w = \sigma^m \in \Sigma^*$, $\sigma \in \Sigma$, tels que $\psi(\sigma^m) = \overline{m}$.

$X^* = \psi^{-1}(\{\overline{0}\}) = \{w \in \Sigma^* : |w| \equiv 0 \, mod \;\; (n)\}$ alors,

$X = \Sigma^n$. D'aprés la proposition 3.4, X est un code maximal bipréfixe.

Conclusion

Nous avons présenté dans ce travail une étude sur les codes à longueurs variables et leurs séries génératrices.

Elle est basée sur les notions de monoïde, homomorphisme de monoïde et les mots et langages.

Dans ce mémoire on présenté deux méthodes pour déterminer des codes à longueurs variables,

Les codes à groupes qui basés sur les notions des groupes.

L'algorithme de Sardinas et Patterson qui basé sur les notions des suites et les quotients à gauches.

Bibliographie

[1] S. Aissani, cours Théorie des langages, Université de Béjaia, 2019.

[2] P. Berlioux , Mnacho. Echenim et Michel Lévy. Théorie des langages, Ecole nationale supérieure d'informatique et de mathématique appliquées de France, 2009.

[3] J. Berstel and D. Perrin, Theory of Codes. Academic Press, 1984.

[4] J. Berstel and D. Perrin and C. Reutenauer, Codes And Automata, Cambridge University Press, 2009.

[5] N. Biggs. Coding natural languages. In Codes : An Introduction to Information Communication and Cryptography. Springer, 2008.

[6] Kh. Boubellouta, Fonction symétriques Et Leurs Applications à Certains Nombres Et Polynômes, (Doctoral dissertation). Université Mohamed Seddik Ben Yahia, Jijel, 2020.

[7] P. Catarino, On Some Identities For k-Fibonacci Sequence, Int. J. Contemp. Math. Sciences. 9(1), 37-42, 2014.

[8] P. Dominique and G. Rindone, On syntactic groups, Bulletin of the Belgian Mathematical Society Simon Stevin, 3 (2004), 749-760.

[9] B. Frédérique, Generating functions of Circular Codes, Advances In Applied Mathematics, 22.1 (1999) 1-24.

[10] N. Ghadbane, Cours Master1, Semi groupes et automates finis, Université de M'sila 2017-2018.

[11] N. Ghadbane, Generating Function of Group Codes, International J. Math. Combin. Vol.3(2022), 61-68.

[12] N. Ghadbane, A construction and representation of some varaiable length codes, anale. Seria Informatică, vol 2, 2017.

[13] N. Ghadbane and D. Mihoubi, A construction of some group codes, International Journal of Electronics and Information Engineering, 4 (2) (2016) 55-59.

[14] R. John, Durbin, Modern Algebra An Introduction.

[15] R. Lidl, G. Pilz, Applied Abstract Algebra, 1998.

[16] B. Margaret, Séries génératrices, 2013.

[17] V. Mark, V. Guba, Combinatorial Algebra Syntax And Semantics, 2014.

[18] D. Mihoubi, cours Master1, Combinatoire 2, Université de m'sila, 2022.

[19] M. Nivat, Eléments de la théorie général des codes, Université de Paris, (1965-1966).

[20] J. Pierre Becirspahic, Mot et Langages , Lycée Louis Le Grand de France, 2007.

[21] P. Rannou, Réécriture de diagrammes et de $\sum$-diagrammes, Thèse de doctorat, Université d'Aix-marseille, 2013.

[22] A. A. Sardinas, C. W. Patterson – A necessary and sufficient condition for the unique decomposition of codes messages, IRE internat. Conv. Rec, vol 08, p.104-108, 1953.

ملخص

هذه المذكرة، تندرج في اطار نظرية الشفرات ذات الطول المتغير.

في الفصل الأول، ذكرنا بعض المفاهيم الأساسية حول أنصاف الزمر، و تماثل أنصاف الزمر، والكلمات واللغات، ثم درسنا بعض الخصائص حول السلاسل المولدة

أخيراً، نهتم بالسلاسل المولدة للشفرات ذات الطول المتغير.

كلمات مفتاحية

نصف الزمرة، تماثل نصف الزمرة، الكلمات واللغة، السلاسل المولدة، الشفرات ذات الطول المتغير.

Abstract

In this thesis, we study the variable length codes and its generating series.

In chapter one, we begin with some elementary material concerning of monoids,

homomorphisms of monoids and words and languages.

In chapter 2, we recall some results of generating series.

In chapter 3, we focus on generating series of variable length codes.

Key words

Monoid, monoid morphism, words and languages, generating series, variable length code.

Résumé

Ce mémoire s'inscrit dans le cadre de la théorie des codes à longueurs variables.

Au premier chapitre, on rappelle des notions élémentaires sur les monoïdes,

homomorphismes de monoïdes et les mots et langages.

À la suit, on étude quelques propriétés sur les séries génératrices.

En fin, on s'intérèsse aux séries génératrices des codes à longueurs variables.

Mot-clés

Monoïde, morphisme de monoïdes, mots et langage, série génératrice, code à longueur variable.