Suites automatiques et morphiques de grande complexité le long des sous-suites

Pierre Popoli

UNIVERSITÉ DE LORRAINE

École Doctorale IAEM Lorraine

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LORRAINE

Spécialité de doctorat : Mathématiques

Présentée et soutenue publiquement le 16/12/2022, par :

Pierre Popoli

Suites automatiques et morphiques de grande complexité le long des sous-suites

Devant le jury composé de :

Valérie Berthé Directrice de recherches, CNRS, IRIF	**Examinatrice**
Émilie Charlier Chargée de cours, Université de Liège	**Rapportrice**
Damien Jamet Maître de conférences, Université de Lorraine, Loria	**Directeur de thèse**
Irène Marcovici Maîtresse de conférences, Université de Lorraine, IECL	**Examinatrice**
Edita Pelantová Professor, Czech Technical University in Prague	**Examinatrice**
Jeffrey O. Shallit Professor, University of Waterloo	**Examinateur**
Thomas Stoll Professeur des universités, Université de Lorraine, IECL	**Directeur de thèse**
Arne Winterhof Senior Fellow, ÖAW, RICAM	**Président du Jury et rapporteur**

Institut Élie Cartan de Lorraine (IECL) – UMR 7502
Laboratoire lorrain de recherche en informatique et ses applications (Loria) – UMR 7503

À mes frères.

Remerciements

Cette thèse a pu être réalisée grâce à l'aide de nombreuses personnes, comme tout projet de recherche. Je souhaite ici remercier toutes celles qui m'ont apporté du soutien, sur le plan personnel ou professionnel, sans forcément s'en rendre compte.

Mes premiers remerciements sont pour mes deux directeurs de thèse, Damien et Thomas. Je souhaite vous remercier pour votre accompagnement personnel et scientifique durant ces trois dernières années. Vous avez toujours su être présents, patients et bienveillants avec moi. Vous m'avez toujours laissé une grande liberté dans la manière de mener à bien mes travaux et pour cela je vous en remercie. Je m'estime très chanceux d'avoir pu travailler avec chacun de vous de manière privilégiée aussi longtemps et j'espère que cela sera encore possible par la suite.

J'aimerais remercier Digitrust pour avoir financé ces trois années de recherche.

Je tiens à remercier tous les membres de mon jury. L'intérêt que vous portez à mes travaux m'honore au plus haut point. Dans un premier temps, merci à Émilie Charlier qui a gentiment accepté de relire ce manuscrit et pour ton rapport très détaillé. I would also like to thank Arne Winterhof for his report of my PhD and his support throughout the past years. Merci à Jeffrey Shallit d'avoir accepté d'être dans mon jury et pour les échanges que nous avons pu avoir sur mes travaux. Merci à Valérie Berthé, Irène Marcovici et Edita Pelantová d'avoir accepté d'être dans mon jury.

Je souhaite remercier Romain, Ronan et Samuel, mes amis du Lycée Jean Vigo à Millau. Pendant toutes ces années, nous sommes restés en contact et votre amitié m'est très chère aujourd'hui. Merci également à Clara, Lucas, Marc-André et Théo O. avec qui nous avons passé des vacances mémorables ces derniers étés.

Merci à Valentin H. pour toutes nos années passés ensemble à Paris, depuis la prépa Turgot où nous avons développé une addiction à un certain Bo Bun.

Je souhaite ensuite remercier mes amis de la fac de Jussieu. Merci à Dylan, Thibault, Thomas G. et Victor R. pour nos discussions, toujours objectives et très sensées. Merci aussi à Théo L. pour m'avoir redonné goût à Warhammer à Paris.

Je tiens à remercier les personnes que j'ai rencontrées à Nancy. Dans un premier temps, je veux remercier mes coburéaux, ceux qui ont eu le privilège de me voir tous les jours sur une période prolongée. Alors merci à Pierre-Adrien, pour toutes tes réponses à mes questions, ta bonne humeur constante et d'avoir permis que je puisse rentrer chez moi un soir sur deux. Ensuite, merci à Johann, avec qui on a tout de suite lié une grande amitié. Enfin, merci à Thomas M., pour ton soutien pendant ces trois années, mais qui ne suffiront pas à me faire lire HPMOR.

Je remercie également tous les membres de l'IECL avec qui j'ai pu discuter, principalement à la machine à café. Merci tout particulièrement à Anouk, Benoit, Christophe, Clémence, Coralie, Jimmy, Maxime, Nicolas, Rémi, Rodolphe, Valentin S., Victor D., Vincent. Merci aussi à Yann.

Je souhaite remercier Emmanuel Jeandel pour son accompagnement tout le long de ma thèse et de m'avoir fait confiance pour enseigner en informatique.

Je souhaite remercier tout le personnel administratif de l'IECL, en particulier Laurence, Paola et Virginie. Nous avons tous ici beaucoup de chance de vous avoir, merci beaucoup pour votre accompagnement, votre gentillesse et votre disponibilité.

Pendant la thèse, j'ai aussi fait la rencontre d'André et Séréna, deux amis de Pierre-Adrien et Johann. Je tiens à vous remercier tous les deux pour les bons moments passés ensemble. Vous avez été essentiels à ma survie mentale durant les deux dernières années. J'espère de tout cœur que vous pourrez poursuivre dans les mathématiques plus tard.

Je remercie tous les professeurs qui m'ont enseigné les mathématiques à un moment donné dans mon parcours. Merci tout particulièrement à Marc Matas, mon professeur de terminale, avec qui nous avons continué de discuter pendant plusieurs années. Merci à Régis de la Bretèche pour son enseignement de théorie des nombres analytique en M2 et de m'avoir conseillé de me renseigner sur le laboratoire de Nancy pour la thèse. Une pensée également pour Jan Nekovář, décédé récemment, pour son enseignement de la théorie des nombres dont j'ai adoré suivre le cours en M1.

Je souhaite remercier Anaïs pour les neuf années passées ensemble. Si nous avons pris des chemins distincts aujourd'hui, il n'empêche que je garde un très bon souvenir de notre histoire commune.

Mes derniers remerciements sont destinés à ma famille, sans qui cette thèse n'aurait pas vu le jour. Un grand merci à mes deux frères, Antoine et Julien. Je chéris plus que tous les liens indescriptibles qui nous unissent tous les trois. J'espère pouvoir passer plus de temps avec vous par la suite et cette thèse vous est dédiée pour tout ce que vous faites pour moi. Un grand merci à mes parents, Didier et Laurence, pour votre éducation et pour votre soutien sans faille. Vous avoir eu comme parents est la plus grande chance que j'ai eue de ma vie. Merci à mes grands-parents, Camille, Gérard, Henri et Marie-Claire, pour votre incroyable tendresse depuis que je suis enfant et d'avoir permis que je puisse réaliser mes études à Paris. Merci à ma grande tante, Christiane, pour nos nombreux rendez-vous durant mes années d'études parisiennes, à discuter de la vie et de politique. Je vous remercie tous, ainsi que l'ensemble de ma famille, pour avoir rendu ma vie si plaisante.

Table des matières

Liste des figures

Liste des tableaux

Introduction

> Random numbers should not be generated with a method chosen at random.
>
> Donald E. Knuth (The Art of Computer Programming, vol.2)

Introduction en français

L'aléatoire dans la nature.

Le hasard, ou l'aléatoire, est l'absence de motifs ou d'organisation apparente. Ainsi, un phénomène aléatoire est intrinsèquement imprévisible. Nous trouvons dans la nature un certain nombre de phénomènes pour lesquels une structure claire ne peut être déterminée. Par exemple, la radioactivité est un phénomène physique imprévisible. En effet, la vitesse de désintégration est calculable mais il est impossible de prédire quel atome spécifique se désintègre pour un intervalle de temps fixé. D'un point de vue biologique, l'héritage de l'ADN est aléatoire. Nous obtenons 50% de notre ADN de chacun de nos parents et il est imprévisible de savoir quelle partie est héritée. En mathématiques, les décimales du nombre π ne semblent pas suivre de schéma particulier. Il est conjecturé que π est un nombre univers, c'est-à-dire que toute suite finie de chiffres apparaît dans les décimales de π.

Mesures de complexité.

Les travaux de cette thèse s'inscrivent dans la volonté de créer des suites aléatoires. Pour pouvoir affirmer qu'une suite est *aléatoire*, nous disposons de plusieurs critères théoriques. Une interprétation possible est qu'une suite aléatoire est une suite "sans caractéristique particulière". Ainsi, si un critère est vérifié pour presque toutes les suites, une suite aléatoire doit satisfaire ce critère. Les critères utilisés dans cette thèse sont appelés *mesures de complexité*. Si une mesure pour une suite donnée est trop éloignée de ce qui est attendu de la mesure d'une suite aléatoire, nous ne qualifions pas cette suite d'aléatoire.

Génération de nombres aléatoires.

Un *générateur de nombres aléatoires*, ou *random number generator* (RNG) en anglais, est un procédé qui produit une suite de nombres dont il est impossible de prédire un

terme en fonction de ses prédécesseurs. Historiquement introduits dans les jeux de hasard, les générateurs de nombres aléatoires sont désormais utilisés dans de nombreux domaines scientifiques. Par exemple, l'utilisation de l'aléatoire est un des fondements de la cryptographie pour engendrer des clés de session. Plus la clé est aléatoire, plus le système cryptographique est sécurisé. En intégration numérique, les suites aléatoires sont utilisées pour les méthodes de Monte-Carlo. En théorie des jeux, l'aléatoire sert également à décrire un comportement non rationnel d'un joueur. Notons que certains de ces domaines ne nécessitent pas un aléa parfait et qu'une suite qui s'en rapproche est suffisant.

Un simple dé à six faces est un générateur de nombres aléatoires qui produit des nombres entre 1 et 6 de manière uniforme : en effet, il n'est pas possible de déterminer le résultat d'un lancer en fonction du lancer précédent. Cependant, avec suffisamment de lancers, la proportion de chaque chiffre est proche de 1/6. En déformant cette distribution uniforme de $\{1, \ldots, 6\}$, nous obtenons un générateur de nombres aléatoires pour tout intervalle. Pour certaines applications, comme l'équilibrage d'instruments, il est également intéressant de chercher des générateurs de nombres aléatoires qui produisent une distribution normale plutôt qu'une distribution uniforme, comme par exemple la Planche de Galton.

À l'inverse, certaines structures mathématiques ou naturelles, bien que paraissant complexes, voire "aléatoires", sont totalement prévisibles. Les plus célèbres d'entre elles sont probablement celles impliquant un comportement fractal comme le chou romanesco ou des spirales dans les coquillages. Ces agencements sont entièrement déterministes, par opposition à une organisation aléatoire. Il convient alors de ne pas confondre aléa et structure complexe.

Les meilleurs générateurs de nombres aléatoires reposent sur des propriétés physiques, appelés *générateurs de nombres aléatoires matériels*, ou *true random number generator* (TRNG). Par exemple, les générateurs fondés sur la radioactivité ou les bruits thermiques sont hautement imprévisibles. Il est possible d'utiliser ces générateurs en cryptographie pour générer des clés cryptographiques aléatoires pour le chiffrement des données par exemple. Cependant, ces générateurs possèdent un débit de nombres aléatoires relativement faible, les suites obtenues ne sont pas reproductibles et sont relativement difficiles à mettre en place pour une utilisation quotidienne. Par conséquent, pour corriger ces défauts, une question naturelle, quoique paradoxale, se pose : est-il possible de créer de l'aléatoire à partir d'un algorithme déterministe ? Une suite *pseudo-aléatoire* est une suite engendrée par un algorithme déterministe dont le comportement est "proche" d'une suite aléatoire. Un *générateur de nombres pseudo-aléatoires*, ou *Pseudo Random Number Generator* (PRNG) en anglais, est un algorithme qui engendre des suites pseudo-aléatoires. Le terme *pseudo* renvoie directement au fait que l'on utilise un algorithme déterministe pour tenter de simuler de l'aléatoire.

L'un des premiers générateurs de nombres pseudo-aléatoires, la *méthode carré médian*, utilise exclusivement des opérations arithmétiques élémentaires. Il a été publié par J. v. Neumann en 1951 [vN51] : soit un nombre m, appelé *graine*, de 6 chiffres en base

10. On calcule m^2 et soit m' le nombre correspondant aux 6 chiffres du milieu de m^2. Puis on itère la méthode avec m'. Par exemple,

- $m = 923456$,
- $m^2 = 852\,\underline{770983}\,936$,
- $m' = 770983$.

La suite obtenue est la suite $923456, 770983, 414786, 047425, \ldots$. Ce procédé itératif est un générateur simple à implémenter et calcule une suite pseudo-aléatoire rapidement. En effet, il parait assez difficile de prédire un élément en fonction des ces prédécesseurs si la méthode de calcul est inconnue. En pratique, la méthode carré médian n'est pas employée pour la génération de nombres aléatoires, car la suite ainsi obtenue contient éventuellement de nombreuses répétitions et sa période est généralement courte.

Le choix de la graine conditionne la qualité de la suite pseudo-aléatoire obtenue, c'est-à-dire sa période, ses propriétés statistiques, etc. Un autre exemple de générateur où le choix de la graine est également important est le *générateur congruentiel linéaire*, introduit par Lehmer publié en 1951 [Leh51]. Soit quatre entiers naturels :

- m le module,
- a le multiplicateur,
- c l'incrément,
- x_0 la graine.

La suite $(x_n)_{n\geq 0}$, à valeurs dans $\{0, 1, \ldots, m-1\}$, engendrée par le générateur est définie par la formule de récurrence suivante :

$$x_{n+1} = a \cdot x_n + c \pmod{m}, \quad n \geq 0.$$

Comme pour la méthode carré médian, tous les choix de paramètres (m, a, c, x_0) ne produisent pas des suites pseudo-aléatoires de même qualité. Par exemple, si $m = 2^8 = 256$, $a = 25$, $c = 16$ et $x_0 = 50$, la suite engendrée est

$$\underline{50, 242, 178, 114}, \underline{50, 242, 178, 114}, \ldots$$

Cette suite n'est pas une suite pseudo-aléatoire car sa période est très courte. De bons choix de paramètres pour le générateur congruentiel linéaire sont désormais connus, voir [Knu97, Chapter 3]. Notons que pour $m = 2$, le générateur congruentiel linéaire de paramètres $(2, a, c, x_0)$ ne peut engendrer que quatre suites différentes : la suite constante égale à `0`, la suite constante égale à `1`, la suite $\texttt{0}, \texttt{1}, \texttt{0}, \texttt{1}, \texttt{0}, \ldots$ ou la suite $\texttt{1}, \texttt{0}, \texttt{1}, \texttt{0}, \texttt{1}, \ldots$. Ce procédé s'étend naturellement à d'autres relations de récurrences plus "complexes". Cependant, pour une relation de récurrence fixée, la pertinence du générateur n'est pas garantie : ajouter de la complexité dans la construction de la suite n'implique pas nécessairement une suite plus proche d'une suite aléatoire.

En pratique, les implémentations d'un générateur de nombres pseudo-aléatoires utilisent par exemple la date et l'heure de l'ordinateur si aucune graine n'est spécifiée. De cette manière, deux appels successifs au générateur produisent deux résultats différents. De nos jours, il existe des générateurs de nombres pseudo-aléatoires très sophistiqués et qui possèdent de meilleures propriétés. Par exemple, le module `random` de Python utilise le *Mersenne twister*.

Registre à décalage à rétroaction.

Les *registres à décalage à rétroaction linéaire*, ou *linear feedback shift registers* (LFSR) en anglais, sont des dispositifs électroniques ou logiciels qui produisent une suite récurrente linéaire sur le corps $\mathbb{F}_2$. Un tel registre est un type particulier de registre à décalage où le bit entrant est donné par une fonction linéaire des bits du registre et justifie donc le terme de *rétroaction linéaire*. Par conséquent, le bit entrant est obtenu comme une combinaison de OU exclusif (ou XOR) entre les bits du registre. Ainsi, les LFSR sont simples à implémenter et efficaces en allocation de mémoire. La notion de LFSR a été généralisée à tout corps fini $\mathbb{F}_{p^n}$. De manière plus formelle, la définition d'un LFSR est la suivante.

Définition (LFSR). *Soient p un nombre premier et $n \geq 1$. Un LFSR sur le corps fini $\mathbb{F}_{p^n}$ est la donnée de :*

- *un entier r, appelé la* taille,
- *une fonction linéaire $f : \mathbb{F}_{p^n}^r \to \mathbb{F}_{p^n}$, appelée* fonction de retour.

À partir d'un état initial $(a_0, \ldots, a_{r-1}) \neq (0, \ldots, 0)$, où $a_i \in \mathbb{F}_{p^n}$ pour tout i, le LFSR engendre une suite $(a_i)_{i\geq 0}$ de la manière suivante :

- $a_r = f(a_0, \ldots, a_{r-1})$,
- a_0 sort du registre,
- les a_i restants sont décalés vers la droite dans le registre,
- a_r rentre dans le registre.

La suite $(a_i)_{i\geq 0}$ est appelée *suite de sortie* du LFSR. Avec le LFSR de la Figure 0.1, si l'état initial est $(1, 0, 0, 1)$, la première sortie est 1 et l'état suivant est $(1, 1, 0, 0)$. L'ensemble des valeurs successives dans le registre et de la sortie sont dans la table suivante.

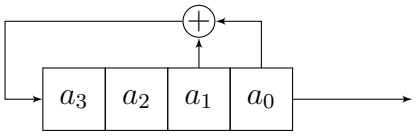


Figure 0.1. Un LFSR de taille 4.

a_3	a_2	a_1	a_0	Sortie
1	0	0	1	1
1	1	0	0	0
0	1	1	0	0
1	0	1	1	1
0	1	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1
0	1	1	1	1
0	0	1	1	1
0	0	0	1	1
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
1	0	0	1	1
⋮	⋮	⋮	⋮	⋮

La suite engendrée $(a_i)_{i\geq 0}$ prend donc les valeurs successives $\mathtt{100110101111000}\ \mathtt{100}\cdots$, et est périodique de période de longueur 15. La suite engendrée $(a_i)_{i\geq 0}$ est définie par la formule de récurrence suivante sur $\mathbb{F}_2$:

$$\begin{cases} a_0 = \mathtt{1}, \quad a_1 = \mathtt{0}, \quad a_2 = \mathtt{0}, \quad a_3 = \mathtt{1}, \\ a_i = a_{i-3} + a_{i-4}, \quad i \geq 4. \end{cases}$$

De manière générale, un LFSR sur $\mathbb{F}_{p^n}$ de taille r engendre une suite $(a_i)_{i\geq 0}$ définie par une formule de récurrence linéaire

$$a_{i+r} = c_1 a_{i+r-1} + \cdots + c_r a_i,$$

avec $c_i \in \mathbb{F}_{p^n}$ uniquement déterminés à l'aide de la fonction de retour f.

Un LFSR sur $\mathbb{F}_2$ de taille r engendre une suite de période au maximum $2^r - 1$. Un LFSR de *longueur maximale* est un LFSR sur $\mathbb{F}_2$ de taille r qui engendre une suite de période exactement $2^r - 1$. Le LFSR de la Figure 0.1 est donc de taille maximale, car la suite engendrée est de période $15 = 2^4 - 1$. Cependant, tous les LFSR ne sont pas de taille maximale. Par exemple, le LFSR de la Figure 0.2 engendre une suite de période strictement plus petite que 15, indépendamment de l'état initial. Pour l'état initial $(\mathtt{1}, \mathtt{0}, \mathtt{0}, \mathtt{1})$, l'ensemble des valeurs successives dans le registre et de la sortie sont dans la table suivante.

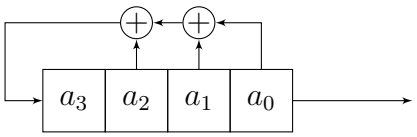


FIGURE 0.2. Un autre LFSR de taille 4.

a_3	a_2	a_1	a_0	Sortie
1	0	0	1	1
1	1	0	0	0
1	1	1	0	0
0	1	1	1	1
1	0	1	1	1
0	1	0	1	1
0	0	1	0	0
1	0	0	1	1
⋮	⋮	⋮	⋮	⋮

La suite engendrée est donc la suite $\mathtt{1001110}\ \mathtt{100}\cdots$, et est périodique de période 7. Pour la valeur initiale $(\mathtt{1}, \mathtt{1}, \mathtt{1}, \mathtt{1})$, la suite engendrée est la suite constante $\mathtt{1}, \mathtt{1}, \mathtt{1}, \mathtt{1} \ldots$.

Les LFSR sont encore utilisés de nos jours dans des nombreuses applications. Parmi les applications les plus connues, les LFSR de longueur maximale servent de générateur de nombre pseudo-aléatoires dans les *chiffrement par flot*. A contrario, un LFSR de longueur non maximale n'est pas recommandé pour une telle application.

La *complexité linéaire au rang* N d'une suite $\mathcal{S} = (s_n)_{n\geq 0}$ sur $\mathbb{F}_{p^n}$, notée $L(\mathcal{S}, N)$, est la taille du plus petit LFSR qui engendre les N premiers éléments de $\mathcal{S}$. La complexité linéaire est utilisée à la fois d'indicateur d'imprévisibilité d'une suite par un LFSR et de mesure de complexité pour les suites pseudo-aléatoires. En particulier, pour une suite

$\mathcal{S}$ sur $\mathbb{F}_2$ dont les N premières valeurs sont indépendantes et des variables aléatoires uniformément distribuées, la complexité linéaire au rang N attendue est

$$\frac{N}{2} + \frac{4 + \overline{N}}{18} + 2^{-N}\left(\frac{N}{3} + \frac{2}{9}\right),$$

où $\overline{N} = N \bmod 2$, voir [Rue86]. L'algorithme de Berlekamp–Massey calcule $L(\mathcal{S}, N)$ d'une suite $\mathcal{S}$ en $O(N^2)$ opérations. Par conséquent, cet algorithme rend l'utilisation des LFSR, dans cette configuration, cryptographiquement non sûre. Pour résoudre ce problème, plusieurs options sont envisageables. Dans cette thèse, nous nous intéresserons en particulier à une plus grande classe de registre à décalage où la fonction de retour n'est pas nécessairement linéaire. Ces dispositifs sont appelés *registres à décalage à rétroaction* ou *feedback shift registers* (FSR). Le livre référence de S. W. Golomb [Gol67] fournit une description détaillée des LFSR et des FSR en général, de leurs propriétés à leurs applications.

Définition (FSR). *Soient p un nombre premier et $n \geq 1$. Un FSR sur le corps fini $\mathbb{F}_{p^n}$ est la donnée de :*

- *un entier r, appelé la* taille.
- *une fonction $f : \mathbb{F}_{p^n}^r \to \mathbb{F}_{p^n}$, appelée* fonction de retour.

Si f n'est pas linéaire, le FSR est appelé *Non Linear Feedback Shift Register* (NLFSR). Le tableau suivant donne la suite engendrée par le NLFSR sur $\mathbb{F}_2$ de longueur 4, de fonction de retour et de valeur initiale $(0, 0, 1, 1)$.

a_3	a_2	a_1	a_0	Sortie
0	0	1	1	1
1	0	0	1	1
1	1	0	0	0
0	1	1	0	0
1	0	1	1	1
0	1	0	1	1
0	0	1	0	0
1	0	0	1	1
⋮	⋮	⋮	⋮	⋮

Un NLFSR de taille r engendre une suite de période au maximum $2^r - 1$, au même titre qu'un LFSR de taille r. L'avantage des NLFSR par rapport aux LFSR réside dans le fait qu'ils engendrent des suites pouvant atteindre des complexités linéaires plus grandes que celles engendrées par des LFSR. Par exemple, pour la suite $\mathcal{S} = 11001101\ 110\cdots$, obtenue à l'aide du NLFSR précédent, la complexité linéaire de $\mathcal{S}$ satisfait $L(\mathcal{S}, N) = 6$ pour tout $N \geq 11$.

La *complexité d'ordre maximal au rang N* d'une suite $\mathcal{S}$ sur $\mathbb{F}_{p^n}$, notée $M(\mathcal{S}, N)$, est la taille du plus petit FSR sur $\mathbb{F}_{p^n}$ qui engendre les N premiers termes de $\mathcal{S}$. À l'instar de la complexité linéaire, la complexité d'ordre maximal sert d'indicateur d'imprévisibilité

d'une suite par un FSR et de mesure complexité pour les suites pseudo-aléatoires. C. J. A. Jansen [Jan89] s'intéresse dans sa thèse aux propriétés de la complexité d'ordre maximal et à l'algorithme qui la calcule. Cette mesure de complexité est l'objet principal d'étude des premiers travaux de cette thèse.

Combinatoire des mots.

La combinatoire des mots est un domaine à la croisée des chemins entre l'informatique théorique et les mathématiques. Elle étudie les propriétés des mots finis ou infinis avec des outils empruntés à la théorie des nombres, à la théorie des groupes et à la combinatoire. La combinatoire des mots apparaît également dans la théorie des automates et des langages formels. Les premiers travaux remontent au début du vingtième siècle avec les travaux de A. Thue. Au milieu du vingtième siècle, les travaux de M-P. Schützenberger et de R. C. Lyndon ont contribué de manière significative au développement de la combinatoire des mots. Les livres références de Lothaire [Lot97, Lot02] et Pytheas Fogg [Fog02] donnent une large description de la littérature qui existe en combinatoire des mots.

Étudier si un mot évite un motif fixé est un sujet central en combinatoire des mots. Par exemple, un mot ω évite les *carrés* s'il ne contient aucun facteur de la forme xx. Il est évident de démontrer qu'un mot ω d'au moins quatre lettres sur un alphabet à deux lettres $\{\mathtt{0},\mathtt{1}\}$ n'évite pas les carrés :

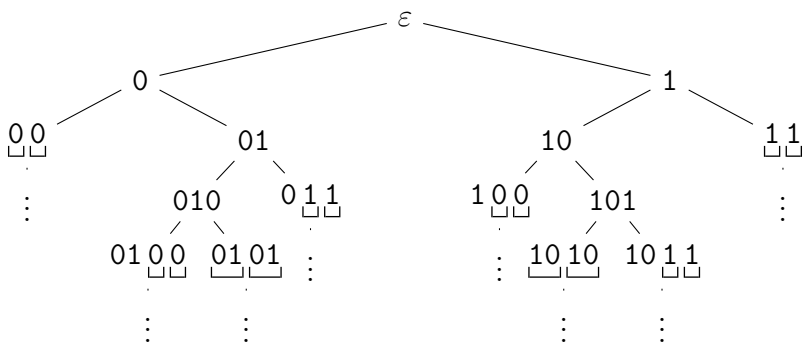


FIGURE 0.3. Les mots de plus de quatre lettres sur $\{\mathtt{0},\mathtt{1}\}$ n'évitent pas les carrés.

Suite à cette observation, il est naturel de se poser la question : existe-t-il un mot infini qui évite les cubes sur l'alphabet $\{\mathtt{0},\mathtt{1}\}$? Le mot de Prouhet–Thue–Morse, ou mot de Thue–Morse, donne une réponse positive à cette question. Soit f le morphisme de mots défini par :

$$f : \begin{cases} \mathtt{0} \mapsto \mathtt{01} \\ \mathtt{1} \mapsto \mathtt{10}. \end{cases}$$

Le point fixe de f débutant par $\mathtt{0}$, noté $f^\omega(0)$, est le mot de Thue–Morse, à savoir le mot

$$\mathfrak{t} = \mathtt{01101001100101100110}\cdots$$

La suite de Thue–Morse est la suite $\mathcal{T} = (t_n)_{n\geq 0}$ où t_n correspond à la n-ème lettre du mot infini $\mathfrak{t}$.

La suite $\mathcal{T}$ apparaît de manière implicite pour la première fois dans les travaux de E. Prouhet qui s'intéresse à une question posée par G. Tarry et E. B. Escott : existe-t-il deux ensembles disjoints I, J qui partitionnent l'ensemble $\{0, 1, \dots, 2^N - 1\}$ tels que

$$\sum_{i \in I} i^k = \sum_{j \in J} j^k, \quad \text{pour } k = 0, 1, \dots, t \text{ ?}$$

E. Prouhet [Pro51] montre qu'il est possible de construire une telle partition pour $N = t + 1$, en utilisant la suite $\mathcal{T}$.

A. Thue [Thu06, Thu12] redécouvre cette suite au début du vingtième siècle. Il démontre que le mot $\mathfrak{t}$ est sans *chevauchements*, c'est-à-dire qu'il n'existe pas de facteur de la forme $axaxa$, avec a une lettre et x un facteur, dans le mot $\mathfrak{t}$. Ces deux articles ouvrent la voie à d'autres questions reliées à l'évitabilité des motifs et sont considérés comme l'origine de la combinatoire des mots.

M. Morse [Mor21] découvre également la suite $\mathcal{T}$ en codant les géodésiques des surfaces à courbure constante négative par des mots infinis sur l'alphabet $\{a, b\}$.

Par son histoire et ses applications, la suite de Thue–Morse apparaît dans de nombreux domaines, voir [AS99, Mau01]. Parmi les apparitions les plus insolites, le grand maître M. Euwe a créé une partie d'échecs infinie à partir de la suite de Thue–Morse. L'économiste I. Palacios-Huerta [PH12], quant à lui, a proposé d'utiliser la suite de Thue–Morse comme ordre de passage aux tirs aux buts afin de réduire l'avantage de tirer en premier.

Une suite *automatique* est une suite engendrée par un automate fini déterministe. Un morphisme est *k-uniforme* si toutes ses images sont de même longueur k. Le Théorème de Cobham [Cob72] crée un pont entre la théorie des langages et la combinatoire des mots, il établit qu'une suite est automatique si et seulement si elle est obtenue comme point fixe d'un morphisme k-uniforme. Dans ce cas, la suite est dite *k-automatique*. La suite de Thue–Morse est par conséquent une suite 2-automatique.

La suite de Golay–Rudin–Shapiro, ou suite de Rudin–Shapiro, est un autre exemple de suite automatique. Soient f et π le morphisme et le codage définis par :

$$f : \begin{cases} a \mapsto ab, & b \mapsto ac, \\ c \mapsto db, & d \mapsto dc, \end{cases} \qquad \pi : \begin{cases} a, b \mapsto 0, \\ c, d \mapsto 1. \end{cases}$$

Le mot de Rudin–Shapiro est le mot infini $\mathfrak{r} = \pi \circ f^\omega(a)$, à savoir

$$\mathfrak{r} = \mathtt{0001001000011101} \cdots$$

La suite de Rudin–Shapiro est la suite $\mathcal{R} = (r_n)_{n \geq 0}$, où r_n désigne la n-ème lettre du mot $\mathfrak{r}$. Cette suite est historiquement définie sur $\{-1, 1\}$, notée $\mathcal{R}' = (r'_n)_{n \geq 0}$, où $r'_n = (-1)^{r_n}$. Cette suite a été étudiée par W. Rudin [Rud59] à partir des travaux de H. S. Shapiro [Sha53]. Cette suite a aussi été découverte indépendamment par E. Golay [Gol51] sous une autre forme, voir l'article de J-P. Allouche [All16] pour plus de détails historiques sur la découverte de la suite $\mathcal{R}$. La suite de Rudin–Shapiro a été initialement étudiée pour ses propriétés vis-à-vis de la norme L^2. La norme L^2 d'une

fonction f est

$$\|f\|_2 = \left(\frac{1}{2\pi}\int_0^{2\pi} |f(t)|^2 dt\right)^{1/2}.$$

Toute suite $(a_n)_{n\geq 0}$ sur $\{-1,1\}$ satisfait la minoration suivante

$$\sup_{\theta\in\mathbb{R}}\left|\sum_{0\leq n<N} a_n e^{in\theta}\right| \geq \left\|\sum_{0\leq n<N} a_n e^{in\theta}\right\|_2 = \sqrt{N}.$$

De plus, presque toute suite $(a_n)_{n\geq 0}$ sur $\{-1,1\}$ satisfait la majoration suivante [SZ54],

$$\sup_{\theta\in\mathbb{R}}\left|\sum_{0\leq n<N} a_n e^{in\theta}\right| = O(\sqrt{N\log N}).$$

La suite de Rudin–Shapiro sur $\{-1,1\}$ est un exemple de suite où il y a plus d'annulations dans la somme que "prévu". En effet, la suite $\mathcal{R}$ satisfait la propriété de la *racine carrée* : il existe une constante $C \geq 0$ telle que pour tout $N \geq 0$, on a

$$\sup_{\theta\in\mathbb{R}}\left|\sum_{0\leq n<N} r'_n e^{in\theta}\right| \leq C\sqrt{N}.$$

De plus la puissance de N est optimale et la meilleure constante connue à ce jour est $C = (2+\sqrt{2})\sqrt{3/5}$, voir [Saf86]. Il est conjecturé que la constante optimale est $C = \sqrt{6}$, voir [AS03, Page 122]. La suite de Thue–Morse sur $\{-1,1\}$, notée $t'_n = (-1)^{t_n}$, ne satisfait pas la propriété de la "racine carrée". En effet, A. Gelfond [Gel68] montre que

$$\sup_{\theta\in\mathbb{R}}\left|\sum_{0\leq n<N} t'_n e^{in\theta}\right| \leq CN^{\log 3/(2\log 2)},$$

où $\log 3/(2\log 2) \doteq 0.792\ldots$ et la puissance est optimale.

Les suites *morphiques* sont obtenues comme point fixe d'un morphisme de mots, pas nécessairement uniforme, à un codage près. Les suites morphiques partagent des propriétés communes avec les suites automatiques, mais sont sur certains aspects très différentes. En effet, le bien connu théorème de Christol [Chr79] affirme que, pour p premier, une suite est p-automatique si et seulement si sa série génératrice est algébrique. Les suites automatiques sont des suites morphiques. En revanche, il existe des exemples de suites morphiques non automatiques, comme la suite caractéristique des carrés, voir [Rit63]. Une myriade d'exemples de suites morphiques est donnée dans l'article [ACSZ17].

Nous présentons un lien qui relie la théorie des nombres et la combinatoire des mots, connu sous le nom de Conjecture de Sarnak. La fonction de Möbius μ est définie par

$$\mu(n) = \begin{cases} (-1)^k, \text{si } n = p_1\cdots p_k \text{ pour des premiers distincts } p_i, \\ 0, \text{ sinon.} \end{cases}$$

La fonction de Möbius joue un rôle important en théorie des nombres. Par exemple, l'estimation suivante

$$\left|\sum_{n\leq N}\mu(n)\right| = O\left(N^{\frac{1}{2}+\varepsilon}\right)$$

est équivalente à l'hypothèse de Riemann, voir [Tit86], et

$$\left|\sum_{n<N}\mu(n)\right| = o(N)$$

est équivalente au Théorème des Nombres Premiers [IK04].

Une suite $(a_n)_{n\geq 0}$ est dite *orthogonale* à $\mu(n)$ si

$$\left|\sum_{n<N} a_n\mu(n)\right| = o\left(\sum_{n<N}|a_n|\right).$$

Le principe d'aléa de Möbius postule qu'une suite complexe bornée "raisonnable" est orthogonale à la fonction de Möbius. La motivation de ce principe provient du fait que les changements de signe de la fonction de Möbius sont suffisamment aléatoires pour créer un grand nombre d'annulations dans la somme $\sum_{n<N} a_n\mu(n)$. Si une suite $(a_n)_{n\geq 0}$ satisfait le principe d'aléa de Möbius, on peut espérer démontrer un Théorème des Nombres Premiers pour la suite $(a_n)_{n\geq 0}$, c'est-à-dire une formule asymptotique pour $\sum_{n\leq x}\Lambda(n)a_n$, où Λ est la fonction de von Mangoldt.

P. Sarnak [Sar11] précise le terme "raisonnable" en prenant une suite issue d'un système dynamique déterministe. Un système dynamique est une paire (X, S), où X est un espace métrique compact et $S : X \to X$ est une fonction continue. Un système dynamique est dit *déterministe* si son entropie topologique est nulle. Une suite $(a_n)_{n\geq 0}$ est *réalisée* par un système dynamique (X, S) s'il existe $x_0 \in X$ et $f : X \to \mathbb{C}$ tel que $a_n = f(S^n(x_0))$ pour tout $n \geq 0$. En particulier, les suites automatiques et morphiques sont des suites réalisées par un système dynamique déterministe. La Conjecture de Sarnak postule que pour toute suite (a_n) réalisée par un système dynamique déterministe, on a

$$\sum_{n<N} a_n\mu(n) = o(N).$$

Cette conjecture a fait l'œuvre de nombreux travaux ces dernières années. Elle a été prouvée dans un premier temps pour certaines suites automatiques célèbres, comme la suite de Thue–Morse et la suite de Rudin–Shapiro, voir respectivement [IK01] et [MR15]. Finalement, C. Müllner [Mül17] établit que toutes les suites automatiques satisfont la Conjecture de Sarnak. La conjecture est encore ouverte pour les autres suites déterministes non-automatiques en général.

Une suite déterministe, comme une suite automatique, est définie à l'aide d'une simple règle locale répétée à l'infini. Par conséquent, ces suites ne sont pas considérées comme des suites pseudo-aléatoires. Cependant, des nombreux travaux récents portent un intérêt à créer des suites pseudo-aléatoires provenant des suites automatiques. L'idée générale est de raréfier une suite automatique $\mathcal{S} = (s_n)_{n\geq 0}$ le long d'une sous-suite,

c'est-à-dire étudier la suite $\mathcal{S}_f = (s_{f(n)})_{n\geq 0}$, où $f : \mathbb{N} \to \mathbb{N}$ est une fonction croissante, et d'étudier son potentiel caractère pseudo-aléatoire. Cette idée est au cœur de cette thèse. En effet, nous étudions plusieurs exemples de suites automatiques et morphiques, le long d'une sous-suite polynômiale ou le long des nombres premiers, et démontrons que ces suites se rapprochent bien plus d'un comportement aléatoire que la suite d'origine.

Fonctions digitales.

Tout le long de cette thèse, nous étudions des fonctions liées à des questions dites *digitales*. Un système de numération est une manière d'exprimer tout entier n comme combinaison linéaire finie $n = \sum_{0\leq i\leq \ell} a_i u_i$, où les u_i sont les éléments de base. Les entiers a_i sont les *chiffres* de n dans ce système de numération. Le système de numération le plus connu est le système décimal, et les systèmes de numération binaire et hexadécimal sont très utilisés en informatique. De manière générale, pour un entier $q \geq 2$, tout entier $n \geq 0$ s'écrit de manière unique $n = \sum_{0\leq i\leq \ell} \varepsilon_i(n) q^i$, où $0 \leq \varepsilon_i(n) < q$ et la suite $(\varepsilon_i(n))_{i\geq 0}$ est nulle à partir du rang ℓ. Le *système de numération en base q* est défini comme le système de numération d'éléments de base $1, q, q^2, \ldots$. L'indice ℓ est la *longueur* de n. La *représentation de n en base q* est le mot $(n)_q = \varepsilon_\ell(n) \cdots \varepsilon_0(n)$.

On note $|n|_k$ le nombre d'occurrences du chiffre k dans la représentation de n en base q. Soit $q \geq 2$ un entier. Une fonction $f : \mathbb{N} \to \mathbb{R}$ est *digitale* si pour tout entier n on a

$$f(n) = \sum_{0\leq k<q} \alpha_k |n|_k.$$

La fonction somme des chiffres en base q, notée $s_q(n)$, est un exemple de fonction digitale, car $s_q(n) = \sum_{0\leq k<q} k|n|_k$. Les fonctions s_q sont très étudiées en théorie des nombres, notamment dans le cadre du problème de Gelfond. En effet, A. Gelfond [Gel68] montre que la fonction $s_q(n)$ est équirépartie le long des progressions arithmétiques. Plus précisément, il montre que pour deux entiers $q, m \geq 2$ tels que $(q-1, m) = 1$, alors pour tout $0 \leq \ell < m$,

$$|\{n < N : s_q(an + b) \equiv \ell \pmod m\}| = \frac{N}{m} + O(N^{1-\eta}),$$

pour un certain $\eta = \eta(q, m) > 0$.

La suite de Thue–Morse $\mathcal{T} = (t_n)_{n\geq 0}$ satisfait l'identité $t_n \equiv s_2(n) \pmod 2$ et cette relation crée le lien du manuscrit entre la théorie des nombres et les suites automatiques. Dans cette thèse, nous utilisons les propriétés des fonctions digitales pour montrer de nouveaux résultats sur la suite de Thue–Morse.

Les fonctions digitales sont des fonctions q-additives. Une fonction $f : \mathbb{N} \to \mathbb{R}$ est *q-additive* si pour tout $(a, b, j) \in \mathbb{N}^3$ tels que $0 \leq b < q^j$,

$$f(aq^j + b) = f(aq^j) + f(b).$$

Pour l'étude des fonctions digitales, il est essentiel de s'intéresser au phénomène des propagations de retenues. Ce phénomène est bien connu en base décimale lorsqu'on additionne deux nombres : si dans une même colonne, la somme des deux chiffres

dépasse 9, alors une retenue est créée et se propage vers la gauche dans les additions des colonnes suivantes.

$$\begin{array}{r} {\scriptstyle 1\ 1\ \ } \\ 1\,2\,5 \\ +\quad 7\,8 \\ \hline 2\,0\,3 \end{array}$$

Ce phénomène est similaire pour tous les systèmes de numération en base q. Dans cette thèse, nous réalisons des études précises sur la propagation de retenues pour prouver de nouveaux résultats sur les fonctions digitales.

Nous nous intéressons également aux fonctions *bloc-digitales*, qui sont une généralisation des fonctions digitales. Pour ω un mot fini sur $\{0, 1, \dots, q-1\}$, on note $|n|_\omega$ le nombre d'occurrences de ω dans la décomposition de n en base q. Même si les fonctions bloc-digitales ne sont pas q-additives, nous avons une propriété similaire à la propagation de retenues et qui reste un outil central dans l'étude de ces fonctions. La suite de Thue–Morse et la suite de Rudin–Shapiro sont reliées à des fonctions digitales. La suite Rudin–Shapiro $\mathcal{R} = (r_n)_n$ satisfait $r_n = |n|_{\mathtt{11}} \bmod 2$, c'est-à-dire le nombre d'occurrences de `11` dans $(n)_2 \bmod 2$, tandis que la suite de Thue–Morse satisfait $t_n = |n|_1 \bmod 2$.

En plus des systèmes de numération en base q, il existe de nombreux autres systèmes de numérations. Par exemple, le système de numération de Zeckendorf, introduit par C. G. Lekkerkerker [Lek51] et E. Zeckendorf [Zec72], est un système de numération basé sur la suite de Fibonacci. La fonction somme des chiffres dans ce système de numération est une suite morphique, voir [Dek21], et constitue un autre exemple de suite déterministe reliée à un système de numération. Nous voulions savoir si les méthodes précédentes sur le système de numération en base q s'étendent à ce système de numération.

Plan de la thèse

Cette thèse est divisée en trois parties relativement indépendantes et une annexe. Dans la première partie (Chapitre I à Chapitre V) nous introduisons les notions utilisées dans l'ensemble de ce manuscrit et présentons nos résultats originaux sur les sous-suites polynomiales de certaines suites déterministes. Dans la deuxième partie (le Chapitre VI) nous nous intéressons à la somme des chiffres en base 2 des carrés parfaits. Enfin, la dernière partie (le Chapitre VII) est consacrée à l'étude des corrélations de la suite de Rudin–Shapiro le long des nombres premiers.

Chapitre I : Nous introduisons les mesures de complexité des suites pseudo-aléatoires étudiées dans cette thèse. À l'aide de ces mesures, nous rappelons que les suites automatiques ne sont pas aléatoires.

Chapitre II : Nous nous intéressons aux mesures de complexité de la suite de Thue–Morse et de ses sous-suites. Nous reproduisons les résultats de notre article [Pop20] sur la complexité d'ordre maximal de la suite de Thue–Morse et de ses généralisations le long des sous-suites polynomiales. Cet article fournit une réponse à une question posée par Z. Sun et A. Winterhof [SW19a].

Chapitre III : Nous introduisons les suites morphiques et les systèmes de numérations. Comme pour les suites automatiques, nous rappelons que les suites morphiques ne sont pas aléatoires.

Chapitre IV : Nous nous intéressons au système de numération de Zeckendorf. En particulier, nous présentons nos résultats de l'article [JPS21] sur la complexité d'ordre maximal de la suite de Fibonacci–Thue–Morse. Ces travaux ont été effectués en collaboration avec D. Jamet et T. Stoll.

Chapitre V : Nous introduisons le Graphe Acyclique Orienté du Mot, ou Directed Acyclic Word Graph (DAWG) en anglais. C. J. A. Jansen [Jan89] établit un lien entre ce graphe et la complexité d'ordre maximal. Nous nous intéressons aux conjectures énoncées dans les articles [JPS21], motivées par des simulations numériques.

Chapitre VI : Nous nous intéressons aux entiers impairs n tels que $s_2(n) = s_2(n^2) = k$ pour $k \geq 1$. La fonction somme des chiffres est un objet central dans cette thèse et K. G. Hare, S. Laishram et T. Stoll [HLS11a] ont partiellement résolu le problème. Nous traduisons en français l'article [AJK$^+$22], écrit en collaboration avec K. Aloui, D. Jamet, H. Kaneko, S. Kopecki et T. Stoll. Nous résolvons le problème pour la majorité des cas restants et introduisons de nouveaux outils potentiellement utiles à la résolution complète du problème. Ce travail a été accepté pour publication dans le journal *Theoretical Computer Science* (TCS).

Chapitre VII : Les corrélations de la suite de Thue–Morse le long des nombres premiers sont connues depuis les travaux de K. Aloui, C. Mauduit et M. Mkaouar [AMM21]. Dans ce chapitre, nous nous intéressons aux corrélations de la suite de Rudin–Shapiro le long des nombres premiers. En particulier, nous montrons que la raréfaction le long des nombres premiers change radicalement le comportement des corrélations de la suite de Rudin–Shapiro. Ce travail n'a pas encore été soumis ni publié.

> Random numbers should not be generated with a method chosen at random.
>
> Donald E. Knuth (The Art of Computer Programming, vol.2)

Introduction in English

Randomness in nature.

Randomness is by definition the lack of pattern or structure. We can find various phenomena in nature with no a clearly structure determined. A random phenomenon is intrinsically unpredictable. For instance, radioactive decay is a physical unpredictable phenomenon. Indeed, the rate of decay is calculable but it is impossible to predict which specific atom will decay in a fixed time interval. From a biological point of view, DNA inheritance is random. We get 50% of our DNA from each of our parents and it is unpredictable to know which part is inherited. In mathematics, the digits of the number π do not seem to follow a particular scheme. The number π is conjectured to be a disjunctive number, i.e. any finite sequence of numbers appears in the digits of π.

Measures of complexity.

The work of this thesis is in line with the desire to create random sequences. To assert that a sequence is *random*, there exist several theoretical notions. A possible interpretation is that a random is a sequence "without particular characteristics". Thus, if a notion is satisfied for almost all the sequences, a random sequence has to satisfy this notion. The notions that we investigate in this thesis are so-called *measures of complexity*. If a measure for a given sequence is too far from what is expected from the measure of a random sequence, we will not call the sequence random.

Random number generation.

A *Random Number Generator* (RNG) is a process by which a sequence of numbers is produced such that it is not possible to compute a term from the precedent terms. Historically introduced in games of chance, random number generators are nowadays used in many scientific fields. For instance, the foundation of cryptography is randomness, which is used to generate session keys. The cryptographic system is more secure if the key is more random. In numerical integration, random sequences are used in Monte-Carlo methods. In game theory, randomness serves to describe a non-rational behavior of a player. Some of these fields do not need perfect randomness and a sequence that is close to a random sequence is sufficient.

A simple dice with six faces is a random number generator that produces numbers uniformly between 1 and 6. Indeed, it is not possible to determine the result of a throw from the one that precedes it. However, with sufficiently many throws, the proportion

of each number is close to 1/6. We can "deform" this distribution over $\{1, \ldots, 6\}$ to obtain other uniform distributions over other intervals. For some applications, such as calibrating instruments, it is more convenient to look for random number generators that produce a normal distribution, such that the Galton Board.

On the other hand, some mathematical or natural structures, although appearing complex, even "random", are completely predictable. Famous examples include the fractal behavior of the Romanesco broccoli or the winding spirals of shells. These arrangements are entirely deterministic, as opposed to a random organisation. It is therefore important not to confuse randomness and complex structure.

The best random number generators are based on a physical phenomenon, also called *true random number generator* (TRNG). For example, generators based on nuclear radioactivity or thermal noise are almost unpredictable. It is possible to use these generators in cryptography to generate random cryptographic keys for data encryption for example. However, these generators produce a small flow of random numbers, generated sequences are not reproducible, and are difficult to set up for everyday use. Therefore, to correct theses flaws, a natural question, that seems paradoxical at first sight, arises: Is it possible to create randomness from a deterministic algorithm? A *pseudorandom* sequence is a sequence generated by a deterministic algorithm such that its behaviour is "close" to the one of a random sequence. A *Pseudo Random Number Generator* (PRNG) is an algorithm that generates pseudorandom sequences. The term *pseudo* refers to the fact that we use a deterministic algorithm to simulate randomness.

One of the first examples of PRNG, based on the *middle-square method*, that only use elementary arithmetical operations, was published by J. v. Neumann in 1951 [vN51]: Let m be a number, called *seed*, of 6 digits in base 10. Take the square of this number m^2 and let m' be the number of the 6 digits of the middle part of m^2. Then, iterate this process with m'. For example,

- $m = 923456$,
- $m^2 = 852\,\underline{770983}\,936$,
- $m' = 770983$.

The obtained sequence is $923456, 770983, 414786, 047425, \ldots$.This iterating process is easy to implement and computes a pseudorandom sequence quickly. Indeed, it seems relatively difficult to predict an element from its predecessors without knowing the method. In practice, the middle-square method is not used for the random number generation, since the obtained sequence possibly contains many repetitions and its period is generally short.

The choice of the seed determines the quality of the generated pseudorandom sequence, i.e. its period, its statistical properties, etc. Another example where the choice of the seed is also important is the *linear congruential generator*, published by D. H. Lehmer in 1951 [Leh51]. Let the following four numbers be given:

- m, the modulus,
- a, the multiplier,
- c, the increment,

- x_0, the seed.

The sequence $(x_n)_{n\geq 0}$, with values in the set $\{0, 1, \ldots, m-1\}$ and generated by this generator is defined by the following recurrence formula:

$$x_{n+1} = a \cdot x_n + c \pmod{m}, \quad n \geq 0.$$

Such as for the middle-square method, all choices of parameters (m, a, c, x_0) do not produce pseudorandom sequences with the same quality of randomness. For instance, let $m = 2^8 = 256$, $a = 25$, $c = 16$ and $x_0 = 50$, then the generated sequence is

$$\underbrace{50, 242, 178, 114}, \underbrace{50, 242, 178, 114}, \ldots$$

This sequence is obviously not a pseudorandom sequence since its period is extremely short. By now, many good choices of parameters for the linear congruential generator are known, see [Knu97, Chapter 3]. Notice that for $m = 2$, the linear congruential generator of parameters $(2, a, c, x_0)$ generates only four different binary sequences: the sequence constant equal to `0`, the sequence constant equal to `1`, the sequence $\mathtt{0}, \mathtt{1}, \mathtt{0}, \mathtt{1}, \ldots$, and the sequence $\mathtt{1}, \mathtt{0}, \mathtt{1}, \mathtt{0} \ldots$,, and none of these sequences is pseudorandom. This process naturally extends to a more "complicated" recurrence formula. However, for a fixed recurrence formula, the quality of the generator is not ensured: adding some complexity in the construction of the sequence does not imply necessarily that the sequence is more "random".

In practice, if no seed is specified, a PRNG uses the time and the date of the computer for the seed. In this manner, two successive calls to the generator will produce two different results. Nowadays, there exist more sophisticated PRNG with better properties. For instance, Python uses the *Mersenne twister* in the function `random` to generate a random sequence.

Feedback shift register.

Linear feedback shift register (LFSR) are electronic devices (or software) that produce a linear recurrence sequence on the field $\mathbb{F}_2$. It is a particular case of a shift register whose input bit is a linear function of its previous state. This justifies the term *linear feedback*. Therefore, the input bit is computed with a combination of XOR between the bits of the register. Consequently, a LFSR is easy to implement and efficient in memory allocation. The notion of LFSR was generalized to any finite field $\mathbb{F}_{p^n}$. More formally, the definition of a LFSR is as follows.

Definition (LFSR). *Let p be a prime number and $n \geq 1$. A LFSR on the finite field $\mathbb{F}_{p^n}$ is given by:*

- *an integer r, called the* length.
- *a linear function $f : \mathbb{F}_{p^n}^r \to \mathbb{F}_{p^n}$, called the* feedback function.

From a initial state $(a_0, \ldots, a_{r-1}) \neq (0, \ldots, 0)$, with $a_i \in \mathbb{F}_{p^n}$ for all $0 \leq i < r$, the LFSR generates a sequence $(a_i)_{i\geq 0}$ in the following way:

- $a_r = f(a_0, \ldots, a_{r-1})$,
- the output is a_0,
- the remaining a_i are shifted to the right in the register,

- a_r gets into the register.

The sequence $(a_i)_{i\geq 0}$ is called the *output sequence* of the LFSR. With the LFSR on $\mathbb{F}_2$ of Figure 0.4, if the initial state is $(1, 0, 0, 1)$, the first output is 1 and the next state is $(1, 1, 0, 0)$. The successive values of the output and the states are the following.

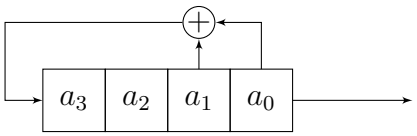


FIGURE 0.4. A LFSR of length 4.

a_3	a_2	a_1	a_0	Output
1	0	0	1	1
1	1	0	0	0
0	1	1	0	0
1	0	1	1	1
0	1	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1
0	1	1	1	1
0	0	1	1	1
0	0	0	1	1
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
1	0	0	1	1
⋮	⋮	⋮	⋮	⋮

The generated sequence has the successive values $101101011111000\ 100\cdots$, it is a periodic sequence of period 15. The generated sequence $(a_i)_{i\geq 0}$ is also defined by the following recurrence formula

$$\begin{cases} a_0 = 1, \quad a_1 = 0, \quad a_2 = 0, \quad a_3 = 1, \\ a_i = a_{i-3} + a_{i-4}, \quad i \geq 4. \end{cases}$$

Generally, a LFSR on $\mathbb{F}_{p^n}$ of length r generates a sequence $(a_i)_{i\geq 0}$ defined by a linear recurrence formula

$$a_{i+r} = c_1 a_{i+r-1} + \cdots + c_r a_i,$$

with $c_i \in \mathbb{F}_{p^n}$ uniquely determined by the feedback function f.

A LFSR of length r on $\mathbb{F}_2$ generates a sequence of period at most $2^r - 1$. A LFSR of *maximal length* is a LFSR of length r on $\mathbb{F}_2$ that generates a sequence of period of exactly $2^r - 1$. The LFSR of Figure 0.4 is of maximal length, since the generated sequence has period $15 = 2^4 - 1$. However, not all LFSR are of maximal length. For example, the LFSR of the Figure 0.5 generates a sequence with a smaller period than 15, independently of the initial value. For the initial value $(1, 0, 0, 1)$, the successive values of the output and the states are the following.

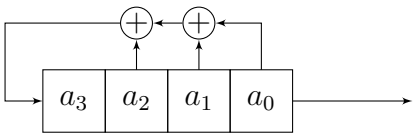


FIGURE 0.5. Another LFSR of length 4.

a_3	a_2	a_1	a_0	Output
1	0	0	1	1
1	1	0	0	0
1	1	1	0	0
0	1	1	1	1
1	0	0	1	1
0	1	0	1	1
0	0	1	0	0
1	0	0	1	1
⋮	⋮	⋮	⋮	⋮

The generated sequence is the sequence $\texttt{1001110 100}\cdots$, a periodic sequence of period 7. For the initial value $(\texttt{1},\texttt{1},\texttt{1},\texttt{1})$, the generated sequence is the sequence $\texttt{1},\texttt{1},\texttt{1},\ldots$.

LFSRs are still used nowadays in many applications. Among the most famous ones, a LFSR of maximal length serves as a pseudorandom number generator in a stream cipher. On the opposite, a LFSR of non maximal length is not suitable for such an application.

The *linear complexity at rank* N of a sequence $\mathcal{S}$ on $\mathbb{F}_{p^n}$, denoted $L(\mathcal{S}, N)$, is the length of the smallest LFSR on $\mathbb{F}_{p^n}$ that generates the first N elements of $\mathcal{S}$. The linear complexity serves both as a measure of the unpredictability of a sequence by a LFSR and as a measure of complexity for pseudorandom sequences. In particular, for a sequence on $\mathbb{F}_2$, where its first N values are independent and uniformly distributed random variables, the expected linear complexity at rank N is

$$\frac{N}{2} + \frac{4 + \overline{N}}{18} + 2^{-N}\left(\frac{N}{3} + \frac{2}{9}\right),$$

where $\overline{N} = N \bmod 2$, see [Rue86]. The Berlekamp–Massey algorithm computes $L(\mathcal{S}, N)$ of a sequence $\mathcal{S}$ in $O(N^2)$ operations. Thus, this algorithm makes the use of LFSRs, in this setup, cryptographically weak. To overcome this obstacle, several options are possible. In this thesis, we are interested in a larger class of shift registers where the feedback function is not necessarily linear. These devices are called *feedback shift registers* (FSR). The reference book of S. W. Golomb [Gol67] provides a detailed description of LFSR and FSR in general, from their properties to their applications.

DEFINITION (FSR). *Let p be a prime number and $n \geq 1$. A FSR on the field $\mathbb{F}_{p^n}$ is given by:*

- *an integer r, called the* length.
- *a function $f : \mathbb{F}_{p^n}^r \to \mathbb{F}_{p^n}$, called the* feedback function.

If f is not linear, the FSR is called *Non Linear Feedback Shift Register* (NLFSR). The following table gives the generated sequence of the NLFSR on $\mathbb{F}_2$ of length 4, of feedback function $f(x, y, z, t) = x \cdot y + x + z$ and with initial value $(\texttt{0},\texttt{0},\texttt{1},\texttt{1})$.

a_3	a_2	a_1	a_0	Output
0	0	1	1	1
1	0	0	1	1
1	1	0	0	0
0	1	1	0	0
1	0	1	1	1
0	1	0	1	1
0	0	1	0	0
1	0	0	1	1
⋮	⋮	⋮	⋮	⋮

A NLFSR on $\mathbb{F}_2$ of length r generates a sequence of period at most $2^r - 1$, such as a LFSR on $\mathbb{F}_2$ of length r. The main benefit of NLFSRs over LFSRs lies in the fact that they generate sequences with larger linear complexity than the ones generated by LFSRs. For instance, the sequence $\mathcal{S} = \texttt{11001101 110}\cdots$, generated by the precedent NLFSR, the linear complexity of $\mathcal{S}$ satisfies $L(\mathcal{S}, N) = 6$ for all $N \geq 11$.

The *maximum order complexity at rank* N of the sequence $\mathcal{S}$ on $\mathbb{F}_{p^n}$, denoted $M(\mathcal{S}, N)$, is the length of the smallest FSR on $\mathbb{F}_{p^n}$ that generates the first N values of $\mathcal{S}$. Such as the linear complexity, the maximum order complexity serves both as a measure of the unpredictability of a sequence by a FSR and as a measure of complexity for pseudorandom sequences. C. J. A. Jansen [Jan89] studied in his thesis the properties of the maximum order complexity and the algorithm that computes it. This measure of complexity is the main object of the first part of this thesis.

Combinatorics on words.

Combinatorics on words is a field at the intersection of theoretical computer science and mathematics. Researchers in this area study the properties of finite and infinite words with the help of tools from number theory, group theory and combinatorics. The first scientific work that can be related to combinatorics on words goes back to the early twentieth century with the work of A. Thue. In the middle of the twentieth century, the work of M-P. Schützenberger and R. C. Lyndon contributed in an essential way to the development of combinatorics on words. The reference books of Lothaire [Lot97, Lot02] and Pytheas Fogg [Fog02] offer a good overview of the literature that exists in combinatorics on words.

One of the principal topics consists in studying whether words can avoid a fixed pattern. For instance, a word ω avoids *squares* if it does not contain any factor of the form xx, where x is again a factor. It is trivial to see that any word ω on a binary alphabet $\{\texttt{0}, \texttt{1}\}$ with at least four letters can not avoid squares.

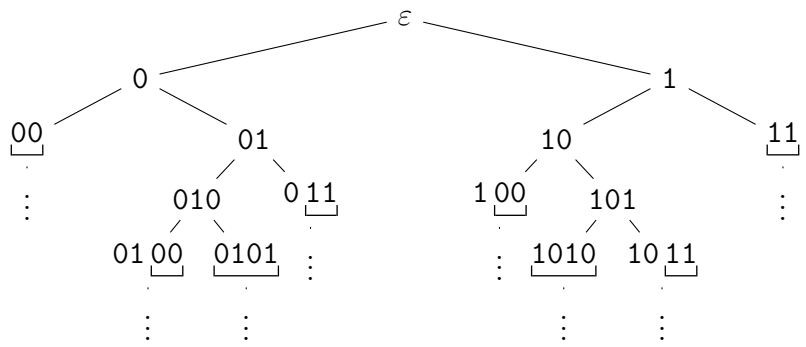


FIGURE 0.6. Words with four or more letters on $\{0, 1\}$ do not avoid squares.

Following this observation, it is natural to ask the question: Does there exist an infinite word that avoids cubes on the alphabet $\{0, 1\}$? The Thue–Morse word (or, Prouhet–Thue–Morse word) gives a positive answer to this question. Let f be the morphism on words defined by:

$$f : \begin{cases} 0 \mapsto 01, \\ 1 \mapsto 10. \end{cases}$$

The fixed point of f starting by 0, denoted by $\mathfrak{t} = f^\omega(0)$, is the Thue–Morse word, i.e. the infinite word

$$\mathfrak{t} = 0110100110010110011 0 \ldots$$

The Thue–Morse sequence is the sequence $\mathcal{T} = (t_n)_{n\geq 0}$ such that t_n is the nth letter of the infinite word $\mathfrak{t}$.

E. Prouhet was the first to find the sequence $\mathcal{T}$ in an implicit way by answering a question of G. Terry and E. B. Escott: Is it possible to find two distincts set I, J such as I and J are a partition of $\{0, 1, \ldots, 2^N - 1\}$ and

$$\sum_{i\in I} i^k = \sum_{j\in J} j^k$$

for $k = 0, 1, \ldots, t$? Prouhet [Pro51] showed that it is indeed possible to find such a partition for $N = t + 1$ with the help of the sequence $\mathcal{T}$.

The sequence $\mathcal{T}$ was discovered by A. Thue [Thu06, Thu12]. He shows that the infinite word $\mathfrak{t}$ is *overlap-free*, it does not contain any subword of the form $axaxa$ (a is a letter, x is a factor). These two articles have led to many other questions related to pattern avoidance and are considered to be the starting point of the field of combinatorics on words.

H. M. Morse [Mor21] also found this sequence in the context of differential geometry. More specifically, he studied geodesics over simply connected surfaces with constant negative curvature and the coding of these geodesics by infinite words on the alphabet $\{a, b\}$. He again discovered the sequence $\mathcal{T}$.

From its history and its applications, the Thue–Morse sequence appears in numerous fields, see [AS99, Mau01] for example for more appearances of this sequence in the

literature. We mention for instance that this sequence was used by grandmaster M. Euwe in chess to create an infinite game of chess. The economist I. Palacios-Huerta [PH12] proposed to use the Thue–Morse sequence for the ordering in the shoot-out at football to reduce the advantage to shoot first.

An *automatic* sequence is a sequence generated by a deterministic finite automaton. A morphism is *k-uniform* if all its images have the same length k. Cobham's theorem [Cob72] creates a bridge between language theory and combinatorics on words since it establishes that a sequence is automatic if and only if the sequence is the fixed point of k-uniform morphism. In this case, the sequence is said to be *k-automatic*. The Thue–Morse sequence is consequently a 2-automatic sequence.

The Golay–Rudin–Shapiro sequence, or Rudin–Shapiro sequence, is another example of an automatic sequence. Let the following morphism and coding be:

$$f : \begin{cases} a \mapsto ab, & b \mapsto ac, \\ c \mapsto db, & d \mapsto dc, \end{cases} \qquad \pi : \begin{cases} a, b \mapsto 0, \\ c, d \mapsto 1. \end{cases}$$

The Rudin–Shapiro word is the infinite word $\mathfrak{r} = \pi \circ f^\omega(a)$, i.e.

$$\mathfrak{r} = \texttt{0001001000011101}\ldots$$

The Rudin–Shapiro sequence is then $\mathcal{R} = (r_n)_{n\geq 0}$ where r_n is the nth letter of $\mathfrak{r}$. It is historically defined over $\{-1, 1\}$, denoted $\mathcal{R}' = (r'_n)_{n\geq 0}$, where $r'_n = (-1)^{r_n}$, and was studied by W. Rudin [Rud59] building upon work by H. S. Shapiro [Sha53]. This sequence was first discovered by E. Golay [Gol51] in another context, see [All16] for historical details on this sequence. The Rudin–Shapiro sequence was originally studied for its properties with respect to the L^2 norm. We define the L^2 norm of a function f by

$$\|f\|_2 = \left(\frac{1}{2\pi}\int_0^{2\pi} |f(t)|^2\right)^{1/2}.$$

For every sequence $(a_n)_{n\geq 0}$ over $\{-1, 1\}$, we have

$$\sup_{\theta\in\mathbb{R}} \left|\sum_{0\leq n<N} a_n e^{in\theta}\right| \geq \left\|\sum_{0\leq n<N} a_n e^{in\theta}\right\|_2 = \sqrt{N}.$$

Furthermore, almost every sequence $(a_n)_{n\geq 0}$ over $\{-1, 1\}$ satisfies [SZ54],

$$\sup_{\theta\in\mathbb{R}} \left|\sum_{0\leq n<N} a_n e^{in\theta}\right| = O(\sqrt{N \log N}).$$

The Rudin–Shapiro is an example of a sequence with more cancelations than "expected". Indeed, the sequence $\mathcal{R}$ satisfies the *square-root property*: There exists a constant $C \geq 0$ such that for all $N \geq 0$ we have

$$\sup_{\theta\in\mathbb{R}} \left|\sum_{0\leq n<N} r'_n e^{in\theta}\right| \leq C\sqrt{N}.$$

Moreover, the power of N is optimal and the best known constant is $C = (2+\sqrt{2})\sqrt{3/5}$. It is conjectured that the optimal constant is $C = \sqrt{6}$, see [AS03, Page 122]. The Thue–Morse sequence over $\{-1, 1\}$, denoted by $t'_n = (-1)^{t_n}$, does not satisfy the square-root property since A. Gelfond [Gel68] showed that

$$\sup_{\theta\in\mathbb{R}} \left| \sum_{0\leq n<N} t'_n e^{in\theta} \right| \leq C N^{\log 3/(2\log 2)}$$

and the power of N is optimal.

A *morphic* sequence is a fixed point of a morphism, that is not necessarily uniform, up to a coding. Morphic sequences share several properties with automatic sequences but are also different in various aspects. For instance, the well-known Christol's theorem asserts that for a prime p, a sequence is p-automatic if and only if its generating series is algebraic over $\mathbb{F}_p$. Automatic sequences are trivially morphic sequences. On the other hand, there exist examples of non-automatic morphic sequences, such as the characteristic sequence of squares, see [Rit63]. A rich variety of examples of morphic sequences is given in the article [ACSZ17].

We introduce another link between number theory and combinatorics on words, more precisely dynamical systems, known under the name of the Sarnak Conjecture. The Möbius function μ is

$$\mu(n) = \begin{cases} (-1)^k, \text{if } n = p_1 \cdots p_k \text{ for distinct primes } p_i, \\ 0, \text{ otherwise.} \end{cases}$$

This function plays an important role in number theory. For instance, the bound

$$\left| \sum_{n\leq N} \mu(n) \right| = O\left(N^{\frac{1}{2}+\varepsilon}\right),$$

is equivalent to the Riemann hypothesis, see [Tit86], and

$$\left| \sum_{n<N} \mu(n) \right| = o(N)$$

is equivalent to the Prime Number Theorem (PNT), see [IK04].

A sequence is said to be *orthogonal* to the Möbius function if

$$\left| \sum_{n<N} a_n \mu(n) \right| = o\left(\sum_{n<N} |a_n| \right).$$

The Möbius randomness principle states that every "reasonable" bounded complex sequence is orthogonal to the Möbius function. The main reasoning behind this principle is that changes of sign of the Möbius function are sufficiently random to create a large amount of cancelation in the sum. If a sequence $(a_n)_n$ satisfies the Möbius randomness principle, we can hope to find a PNT for this sequence, i.e. an asymptotic formula for $\sum_{n\leq x} \Lambda(n) a_n$, where Λ is the von Mangoldt function.

P. Sarnak [Sar11] made the term "reasonable" precise invoking a sequence from a deterministic dynamical systems. A dynamical system is a pair (X, S) with X a compact metric space and $S : X \to X$ a continuous function. A dynamical system is said to *deterministic* if its topological entropy is 0. A sequence $(a_n)_n$ is *realized* by the dynamical system (X, S) if there exists $x_0 \in X$ and $f : X \to \mathbb{C}$ such that $a_n = f(S^n(x_0))$ for every $n \geq 0$.

Sarnak's conjecture states that for any sequence $(a_n)_n$ realized by a deterministic dynamical system, we have that

$$\sum_{n<N} a_n \mu(n) = o(N).$$

This conjecture has received a lot of attention in the past few years. For instance, this conjecture was proved for the Thue–Morse sequence and the Rudin–Shapiro sequence, see [IK01] and [MR15] respectively. Finally, a complete answer for all automatic sequences is given by C. Müllner [Mül17]. This conjecture is still open for non-automatic deterministic sequences.

A deterministic sequence, such as an automatic sequence, is defined with a simple local rule repeated infinitely often. Consequently, these sequences are not considered to be random at all. However, recent work adressed subsequences of automatic sequences as potential pseudorandom sequences. Since the original sequence is not random, the main idea is to rarefy the sequence along sparse subsequences, such as polynomial subsequences or along primes. If we can prove that this process creates good pseudorandom sequences, we have a very fast generation since these sequences are easily generated by a computer. This thesis falls within the scope of this study.

Digital functions.

Throughout this thesis, we study functions that are related to questions on digits. A numeration system is a manner to represent every integer n as a linear combination $n = \sum_{0\leq i\leq \ell} a_i u_i$, where the u_i are the base elements. The integers a_i are the *digits* of n in this numeration system. The most famous numeration system is the decimal one but the binary and the hexadecimal numeration systems also have an important place due to their applications in computer science. More generally, for an integer $q \geq 2$, we define the *numeration system in base* q: Every integer $n \geq 0$ has exactly one representation of the form

$$n = \sum_{i\geq 0} \varepsilon_i(n) q^i,$$

with $0 \leq \varepsilon_i(n) < q$ and the sequence $(\varepsilon_i(n))_{i\geq 0}$ is 0 starting from some rank. The smallest index ℓ such that $\varepsilon_i(n) = 0$ for all $i > \ell$ is the *length* of n. The word $(n)_q = \varepsilon_\ell(n) \dots \varepsilon_0(n)$ is called the *representation of* n *in base* q.

We denote by $|n|_k$ the number of occurrences of the digit k in the representation of n in base q. A function $f : \mathbb{N} \to \mathbb{R}$ is *digital* if for every integer $n \geq 0$ we have

$$f(n) = \sum_{0\leq k<q} \alpha_k |n|_k.$$

The sum of digits function in base q, denoted by $s_q(n)$, is an example of a digital function since

$$s_q(n) = \sum_{0 \le k < q} k|n|_k.$$

These functions are widely studied in number theory, in particular in the context of Gelfond's problems [Gel68]. A. Gelfond showed that the function $s_q(n)$ is equidistributed along arithmetic progressions. More precisely, let $q, m \ge 2$ be integers such that $(q-1, m) = 1$. Then, for every $0 \le \ell < m$, we have

$$|\{n < N : s_q(an+b) \equiv \ell \pmod m\}| = \frac{N}{m} + O(N^{1-\eta}),$$

for some $\eta = \eta(q, m) > 0$.
The Thue–Morse sequence $\mathcal{T}$ satisfies $t(n) \equiv s_2(n) \pmod 2$, and this establishes a further link between number theory and automatic sequences. In this thesis, we use properties of digital functions to prove some new results on the Thue–Morse sequence.

One fundamental property of digital functions is that they are q-additive. A function $f : \mathbb{N} \to \mathbb{R}$ is *q-additive* if for every $(a, b, j) \in \mathbb{N}^3$ such that $0 \le b < q^j$ we have

$$f(aq^j + b) = f(aq^j) + f(b).$$

The phenomenon of carry propagation plays an important role in the study of digital functions. This phenomenon is well-known for the decimal numeration system: In the sum of two numbers, if the sum of two digits in the same column is greater than 10, a carry is created and it propagates to the next column.

$$\begin{array}{cccc} & {\scriptstyle 1} & {\scriptstyle 1} & \\ & 1 & 2 & 5 \\ + & & 7 & 8 \\ \hline & 2 & 0 & 3 \end{array}$$

In fact, carry propagation exists for each q-base numeration system. In this thesis, we need precise control of the various arising carry propagations, to prove our results on digital functions.

A possible generalization of the digital functions are the *block-digital functions*. For ω a word over $\{0, 1, \dots, q-1\}$, we denote $|\cdot|_\omega$ the function that counts the number of occurrences of ω in the q-base expansion of integers. Even though these functions are no longer q-additive, we have a property similar to the carry propagation. This property will again be central in the study of block-digital functions. Both the Thue–Morse sequence and the Rudin–Shapiro are related to block-digital functions. The Rudin–Shapiro sequence satisfies $r_n = |n|_{11} \bmod 2$, i.e. the number of occurrences of $\mathtt{11}$ in $(n)_2$ mod 2 whereas the Thue–Morse sequence satisfies $t_n = |n|_1 \bmod 2$.

Besides the numeration systems in base q, there exist a number of other numeration systems. For instance, the Zeckendorf numeration system, introduced by C. G. Lekkerkerker [Lek51] and E. Zeckendorf [Zec72], is a numeration system based on the Fibonacci sequence. The sum of digits function in this numeration system is a morphic sequence, see [Dek21]. This is another example of a deterministic sequence that is related to a

numeration system. The aim of the present thesis is to know if the former methods on the numeration systems in base q can also be applied to this numeration system.

Outline of the thesis

This thesis contains three relatively independent parts and an annex. The first part (Chapter I to Chapter V) introduces most of the notions used in the thesis and presents our original results on polynomial subsequences of some deterministic sequences. In the second part (Chapter VI) we study the sum of digits in base 2 of perfect squares. Finally, in the last part (Chapter VII) we focus on the correlations of the Rudin–Shapiro sequence along primes.

Chapter I: We introduce the studied measures of complexity for pseudorandom sequences in this thesis. Using these measures, we recall that automatic sequences are not random.

Chapter II: We discuss the measures of complexity of the Thue–Morse sequence and its subsequences. We reproduce results from our article [Pop20] on the maximum order complexity of the Thue–Morse sequence and its generalizations along polynomial subsequences. This article gives a general answer to a question posed by Z. Sun et A. Winterhof [SW19a].

Chapter III: We introduce morphic sequences, numeration systems and show various aspects of morphic sequences that point towards the fact that they are not random.

Chapter IV: We discuss the Zeckendorf numeration system. In particular, we reproduce results from our article [JPS21] on the maximum order complexity of the Fibonacci–Thue–Morse sequence. This article is jointly written with D. Jamet and T. Stoll.

Chapter V: We introduce the Directed Acyclic Weighted Graph of a word. C. J. A. Jansen [Jan89] enlights the link between this graph and the maximum order complexity. We discuss the conjectures stated in the article [JPS21] and support them by numerical simulations.

Chapter VI: We investigate the problem to find odd integers such that $s_2(n^2) = s_2(n) = k$ for fixed $k \geq 1$, where s_2 is the binary sum of digits function. This function plays a central role in this thesis. K. G. Hare, S. Laishram and T. Stoll [HLS11b] resolve partially the problem. We reproduce our article [AJK$^+$22], written in collaboration with K. Aloui, D. Jamet, H. Kaneko, S. Kopecki, and T. Stoll. We solve the problem for three cases and introduce some new tools that might help to solve the problem completely. This work is accepted for publication in the journal *Theoretical Computer Science* (TCS).

Chapter VII: We study correlations of the Rudin–Shapiro sequence along prime numbers. In this context, an analogous result is already known for the Thue–Morse sequence by K. Aloui, M. Mkaouar and C. Mauduit [AMM21]. In particular, we show that the behaviour of these correlations is different from the Rudin–Shapiro sequence itself. This is work in progress and has not yet been submitted or published.

Notations

Dans toute cette thèse, nous utilisons les notations suivantes :

- On note $\mathbb{N} = \{0, 1, \ldots\}$ l'ensemble des entiers naturels.
- On note $|A|$ ou $\mathrm{Card}(A)$, le cardinal d'un ensemble fini A.
- Nous utilisons indifféremment la notation de Landau $f = O(g)$ et la notation de Vinogradov $f \ll g$, à savoir : il existe une constante $C > 0$ telle que $|f| \leq C|g|$. On note $f \asymp g$ si $f = O(g)$ et $g = O(f)$.
- On note $f = o(g)$ si $\frac{f(x)}{g(x)} \to 0$ pour $x \to +\infty$, et $f \sim g$ si $\frac{f(x)}{g(x)} \to 1$ pour $x \to +\infty$.
- On note $e(x) = \exp(2i\pi x)$ pour un réel x.
- On note $\log_b(x)$ le logarithme en base b d'un réel $x > 0$.
- On note $\mathbb{F}_q$ le corps fini à q éléments, pour un nombre premier p et $\alpha \geq 1$ tels que $q = p^\alpha$.
- Les éléments de $\mathbb{F}_2$ sont notés `0` et `1`.
- La fonction φ est la fonction indicatrice d'Euler.
- Soient $x > 0$ et $A \geq 1$, $\log^A x$ désigne la puissance A-ème de $\log x$.

Nous listons les pages où certaines notations importantes de la thèse sont introduites :

CHAPITRE I

Mesures de complexité et suites automatiques

Dans ce chapitre, nous dressons l'état de l'art des mesures de complexité et des suites automatiques. Nous établissons également que ces mesures de complexité ne sont pas totalement indépendantes.

I.1. Définitions

Dans ce paragraphe, nous introduisons les notions élémentaires de la combinatoire des mots.

Un *alphabet* Σ est un ensemble fini de *lettres*. Un *mot fini* w *de longueur* $n \geq 0$ sur l'alphabet Σ est une application de $\{0, 1, \ldots, n-1\}$ vers Σ. Un mot fini w est noté $w_0 w_1 \cdots w_{n-1}$, où $w_i = w(i)$. Le mot fini de longueur 0 est appelé le *mot vide* et est noté ε. La longueur du mot fini w est notée $|w|$. Soit $a \in \Sigma$, on note $|w|_a$ le nombre d'occurrences de a dans $w_0 \cdots w_{n-1}$. L'ensemble des mots finis sur Σ est noté Σ^*.

La *concaténation* de deux mots finis $u, v \in \Sigma^*$, de longueurs respectives n et m, est le mot fini $w = u_0 \cdots u_{n-1} v_0 \cdots v_{m-1}$. Par exemple si $u = \texttt{comp}$ et $v = \texttt{lexe}$, alors $uv = \texttt{complexe}$. L'ensemble Σ^* muni de l'opération de concaténation est un monoïde d'élément neutre ε.

Soit $w \in \Sigma^*$. Un mot $u \in \Sigma^*$ est un *facteur*, ou *sous-mot*, de w s'il existe $x, y \in \Sigma^*$ tels que $w = xuy$. Un mot $u \in \Sigma^*$ est un *préfixe* (resp. *suffixe*) de w s'il existe un mot $x \in \Sigma^*$ tel que $w = ux$ (resp. $w = xu$). Un facteur (resp. préfixe, resp. suffixe) u de w est dit *propre* si $u \neq w$.

Soit $<$ un ordre total sur Σ. L'*ordre lexicographique*, également noté $<$, sur Σ^* est défini comme suit : soient $x, y \in \Sigma^*$, on a $x < y$ si x est un préfixe propre de y ou s'il existe $a, b \in \Sigma$ et $u, x', y' \in \Sigma^*$ tels que $a < b$, $x = uax'$ et $y = uby'$.

Un *mot infini* sur l'alphabet Σ est une application de $\mathbb{N}$ vers Σ et est noté $w_0 w_1 \cdots$, où $w_i = w(i)$. L'ensemble des mots infinis sur Σ est noté Σ^ω. Soient $w \in \Sigma^\omega$ et $u \in \Sigma^*$ de longueur n, on note uw le mot infini $u_0 \cdots u_{n-1} w_0 w_1 \cdots$. Le mot fini u est un *facteur* (resp. *préfixe*) de w s'il existe $x \in \Sigma^*$ et $y \in \Sigma^\omega$ tels que $w = xuy$ (resp. $w = uy$).

Plus généralement, un *mot* sur Σ est soit un mot fini sur Σ soit un mot infini sur Σ, c'est-à-dire un élément de l'ensemble $\Sigma^* \cup \Sigma^\omega$.

Pour plus de détails sur la combinatoire des mots, voir [AS03, Lot97, Lot02].

Dans cette thèse, nous considérons à la fois des suites et des mots infinis à droite. À une suite $\mathcal{S} = (s_n)_{n \geq 0}$, nous associons le mot infini $s_0 s_1 \cdots$, et réciproquement. C'est pourquoi, toutes les définitions présentées pour des suites sont applicables dans le contexte des mots infinis, et réciproquement. Par exemple, une suite est *ultimement périodique* si le mot infini associé est ultimement périodique.

I.2. Mesures de complexité

Dans ce paragraphe, nous introduisons plusieurs mesures de complexité. Notons qu'il existe d'autres mesures de complexité et que nous donnons celles que nous étudions ou qui sont pertinentes dans notre étude. Pour plus de détails sur d'autres mesures de complexité et sur les suites pseudo-aléatoires en général, voir [Gya13, Shp03] par exemple.

Soit Σ un alphabet de taille q. Une suite aléatoire sur Σ est une suite où chaque valeur est choisie avec probabilité $1/q$. Autrement dit, soit μ_q la mesure de probabilité uniforme sur Σ et μ_q^∞ la mesure de probabilité produit sur Σ^ω induite par μ_q. Dans cette thèse, nous formulons plusieurs résultats en fonction de la mesure μ_q^∞ pour décrire le comportement d'une suite aléatoire. Dans toute la suite de cette thèse, nous utilisons uniquement cette mesure et la mention presque partout désigne μ_q^∞ presque partout.

I.2.1. Complexité en sous-mots. Dans tout ce qui suit, sauf mention contraire, Σ désigne un alphabet de taille $q \geq 1$.

Soient w un mot sur Σ et $n \geq 0$ un entier. On note $\mathrm{Sub}_n(w)$ (resp. $\mathrm{Sub}(w)$) l'ensemble des facteurs de longueur n de w (resp. l'ensemble des facteurs de w). Nous pouvons désormais introduire la complexité en facteurs, ou complexité en sous-mots, d'un mot w.

Définition I.2.1. Soient w un mot sur Σ et $n \geq 0$ un entier. La *complexité en facteurs* de w est la fonction $p_w : \mathbb{N} \to \mathbb{N}$ définie par $p_w(n) = \mathrm{Card}(\mathrm{Sub}_n(w))$.

Exemple I.2.1. La complexité en facteurs de $010101\cdots$ est $p_w(n) = 2$ pour tout $n \geq 1$.

Lemme I.2.1. *Soit w un mot sur Σ. Pour tout $n \geq 0$, on a $1 \leq p_w(n) \leq q^n$.*

Un mot infini w est *purement périodique* s'il existe $x \in \Sigma^* \setminus \{\varepsilon\}$ tel que $w = x^\omega := xxx\cdots$, et *ultimement périodique* s'il existe $x, y \in \Sigma^* \setminus \{\varepsilon\}$ tel que $w = yx^\omega$. De manière générale, un mot est dit *périodique* s'il est purement périodique ou ultimement périodique.

Théorème I.2.1 (Théorème de Morse–Hedlund, [MH38]**).** *Soit $w \in \Sigma^\omega$. Le mot w est périodique si et seulement s'il existe un entier $n \geq 0$ tel que $p_w(n) \leq n$.*

Par conséquent, la complexité en facteurs d'un mot infini non-périodique w satisfait $p_w(n) \geq n+1$ pour tout $n \geq 0$. Un mot *sturmien* w est un mot infini tel que $p_w(n) = n+1$ pour tout $n \geq 1$. Pour d'autres caractérisations des mots sturmiens voir [Fog02, Section 6].

Nous introduisons la notion de facteur spécial, car elle joue un rôle central dans la première partie de cette thèse.

Définition I.2.2. Soient w un mot sur Σ et $u \in \mathrm{Sub}(w)$. Le mot u est *facteur spécial à droite* de w s'il existe $a, b \in \Sigma$ distincts tels que $ua, ub \in \mathrm{Sub}(w)$. De même, u est *facteur spécial à gauche* de w est un mot fini sur Σ s'il existe $a, b \in \Sigma$ distincts tels que $u, au, bu \in \mathrm{Sub}(w)$. Un facteur est dit *bi-spécial* s'il est spécial à gauche et à droite.

Exemple I.2.2. Soit $w = \texttt{1011010}$, $\texttt{1}$ est un facteur bi-spécial de w et $\texttt{01}$ est un facteur spécial à droite mais pas à gauche.

Sur un alphabet binaire, le nombre de facteurs spéciaux à droite de longueur n d'un mot infini w est égal à $p_w(n+1)-p_w(n)$. Les mots sturmiens sont donc les mots binaires possédant un seul facteur spécial à droite de longueur n pour tout $n \geq 1$.

Nous introduisons maintenant les graphes de de Bruijn et de Rauzy. Par leurs constructions, ces derniers sont liés à la complexité en facteurs et aux facteurs spéciaux.

Définition I.2.3. Soit $n \geq 1$ un entier. Le *graphe de de Bruijn* d'ordre n sur Σ est un graphe orienté (V, E) où $V = \Sigma^n$ et $(u, v) \in E$ si et seulement s'il existe $a, b \in \Sigma$ tels que $ua = bv$. Dans ce cas, l'arête (u, v) est étiquetée avec la lettre b.

Le degré entrant (le nombre d'arêtes entrantes) et le degré sortant (le nombre d'arêtes sortantes) d'un sommet du graphe de de Bruijn est le même pour tous les sommets, et est égal à $\mathrm{Card}(\Sigma)$.

Définition I.2.4. Soient w un mot sur Σ et $n \geq 1$ un entier. Le *graphe de Rauzy* d'ordre n de w est le graphe (V, E) avec $V = \mathrm{Sub}_n(w)$ et $(u, v) \in E$ s'il existe deux lettres $a, b \in \Sigma$ et un mot fini $x \in \Sigma^*$ tels que $u = ax, v = xb$ et $axb \in \mathrm{Sub}(w)$. Dans ce cas, l'arête (u, v) est étiquetée avec la lettre a. Ce graphe est noté $G_{n,w}$.

Par conséquent, le graphe de Rauzy d'ordre n de w est un sous-graphe du graphe de de Bruijn d'ordre n sur Σ.

Exemple I.2.3. Illustrons ces objets pour le mot $w = \mathtt{0110010}$. L'ensemble de ses facteurs, classés par longueur et par ordre d'apparition est

$$\mathrm{Sub}(w) = \{\mathtt{0}, \mathtt{1},\ \mathtt{01}, \mathtt{11}, \mathtt{10}, \mathtt{00},\ \mathtt{011}, \mathtt{110}, \mathtt{100}, \mathtt{001}, \mathtt{010},\ \mathtt{0110}, \mathtt{1100}, \mathtt{1001}, \mathtt{0010},$$
$$\mathtt{01100}, \mathtt{11001}, \mathtt{10010},\ \mathtt{011001}, \mathtt{110010},\ \mathtt{0110010}\}.$$

Les graphes de Rauzy de w d'ordre 2 et 3 sont donnés en **gras** dans les Figures I.2.3 et I.2.3, avec le graphe de de Bruijn d'ordre 2 et 3 en gris.

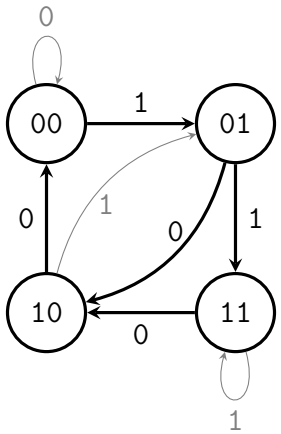


FIGURE I.1. Graphe de Rauzy d'ordre 2 pour $w = \mathtt{0110010}$.

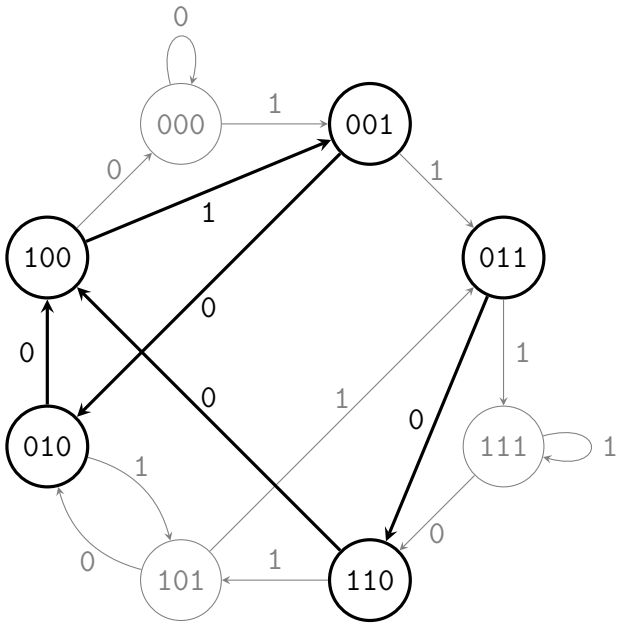


FIGURE I.2. Graphe de Rauzy d'ordre 3 pour $w = \texttt{0110010}$.

Listons quelques propriétés élémentaires des graphes de Rauzy, voir [Rig14] pour plus de détails.

Proposition I.2.1. Soit w un mot sur Σ et $n \geq 1$ un entier. Soit $G_{n,w} = (V, E)$ le graphe de Rauzy d'ordre n de w.

(1) $\mathrm{Card}(V) = p_w(n)$.
(2) $\mathrm{Card}(E) = p_w(n+1)$.
(3) Un facteur spécial à droite de w est un sommet de $G_{n,w}$ de degré sortant supérieur ou égal à 2.

Par conséquent, d'après ses graphes de Rauzy d'ordre 2 et 3, le mot $w = \texttt{0110010}$ possède un facteur spécial à droite de longueur 2, à savoir $\texttt{01}$, mais pas de facteur spécial de longueur 3 car tous les degrés sortants des sommets de $G_{3,w}$ sont égaux à 1.

Remarque I.2.1. Les graphes de Rauzy ne sont pas utilisés explicitement dans cette thèse mais sont fréquemment utilisés en combinatoire des mots. Nous utilisons un autre type de graphe dans cette thèse, voir Remarque V.2.1, page 108, pour plus détails.

Le théorème suivant établit le comportement de la complexité en facteurs pour presque tout mot sur un alphabet fini.

Théorème I.2.2. *Presque tout mot w sur Σ satisfait $p_w(n) = \mathrm{Card}(\Sigma)^n$ pour tout $n \geq 0$.*

PREUVE. Voir par exemple [AS03, Theorem 10.1.6] pour une preuve de ce résultat. □

D'après le Théorème I.2.2, un mot w ne peut pas être qualifié d'aléatoire si sa complexité en facteurs n'est pas maximale, c'est-à-dire il existe $n \geq 1$ tel que $p_w(n) = q^n$

pour tout $n \geq 1$. La complexité en facteurs sert donc de mesure de complexité pour les suites pseudo-aléatoires. Par exemple, la suite de Thue–Morse $\mathcal{T}$ n'est pas aléatoire, car $\mathcal{T}$ ne contient aucun cube, comme mentionné dans l'introduction.

I.2.2. Normalité. Soient $w = w_0 w_1 \cdots \in \Sigma^\omega$ et $a \in \Sigma$. Si la suite $\left(\frac{|w_0 w_1 \cdots w_{n-1}|_a}{n}\right)_{n\geq 1}$ converge, alors la *fréquence* de a dans w est définie par :

$$\mathrm{Freq}_a(w) := \lim_{n\to+\infty} \frac{|w_0 w_1 \cdots w_{n-1}|_a}{n}.$$

Soient w un mot infini et u un mot fini, on note $|w|_{u,n}$ le nombre d'occurrences de u dans le mot fini $w_0 \cdots w_{n-1}$, le préfixe de longueur n de w. Nous introduisons maintenant la notion de nombre normal.

Définition I.2.5. Soit $w \in \Sigma^\omega$. Le mot infini w est un mot *normal* sur Σ si pour tout mot fini $u \in \Sigma^*$

$$\lim_{n\to+\infty} \frac{|w|_{u,n}}{n} = \frac{1}{q^{|u|}}.$$

Le mot infini w est *simplement normal* si pour tout $a \in \Sigma$,

$$\lim_{n\to+\infty} \frac{|w|_{a,n}}{n} = \frac{1}{q}.$$

Par conséquent, un mot normal w possède une complexité en facteurs maximale. Le théorème suivant est un analogue au Théorème I.2.2, démontré par Borel [Bor09].

Théorème I.2.3 ([Bor09]**).** *Presque tout mot infini est normal sur* Σ*.*

Si un mot infini n'est pas normal, il n'est pas considéré comme aléatoire. Nous obtenons ainsi une nouvelle mesure de complexité pour les mots infinis.

Remarque I.2.2. Même si presque tout mot infini est normal, il est assez difficile de prouver qu'un mot est normal ou d'en trouver des constructions explicites. La construction la plus élémentaire est due à Champernowne [Cha33] : soient $\Sigma = \{0, 1, \ldots, 9\}$ et le mot infini

$$w = 01234567891011\cdots = \prod_{n\geq 0} (n)_{10},$$

obtenu par concaténation de toutes les écritures décimales des entiers positifs dans l'ordre croissant. Le mot w est un mot normal sur Σ, voir [Cha33]. Cette construction est valable pour tout alphabet $\{0, 1, \ldots, q-1\}$. Notons que cette construction a été étendue de plusieurs manières, par exemple en ne prenant que les nombres premiers par Copeland et Erdős [CE46].

Pour plus de détails sur les nombres normaux, voir [KN74, DT97].

I.2.3. Complexité linéaire. Pour $k \geq 2$, on identifie l'alphabet $\Sigma_k = \{0, \ldots, k-1\}$ avec l'ensemble d'entiers $\{0, \ldots, k-1\}$ lorsque ce sera nécessaire.

Soit $\mathbb{F}_q$ le corps à q éléments avec $q = p^\alpha$, p un nombre premier et $\alpha \geq 1$. Une suite $\mathcal{S} = (s_n)_{n\geq 0}$ sur $\mathbb{F}_q$ est dite *récurrente linéaire d'ordre k* s'il existe $a_0, \ldots, a_{k-1} \in \mathbb{F}_q$ tels que pour tout $n \geq 0$,

$$s_{n+k} = a_0 s_n + a_1 s_{n+1} + \cdots + a_{k-1} s_{n+k-1}.$$

Définition I.2.6. Soit $\mathcal{S} = (s_n)_{n\geq 0}$ une suite sur $\mathbb{F}_q$. Le *profil de complexité linéaire* de $\mathcal{S}$ est la suite $(L(\mathcal{S}, N))_{N\geq 1}$, où

$$L(\mathcal{S}, N) = \min\{L \in \mathbb{N} \ : \exists a_0, \ldots, a_{L-1} \in \mathbb{F}_q, \ \forall 0 \leq n \leq N-L-1, \\ s_{n+L} = a_0 s_n + a_1 s_{n+1} + \cdots + a_L s_{n+L-1}, \}.$$

Par convention $L(\mathcal{S}, N) = 0$ si les N premiers termes de $\mathcal{S}$ sont nuls et $L(\mathcal{S}, N) = N$ si les $N-1$ premiers termes sont nuls et le N-ème est non nul. La *complexité linéaire au rang N* est $L(\mathcal{S}, N)$. De plus, $L(\mathcal{S}) := \sup_{N\geq 1} L(\mathcal{S}, N)$ est appelée *complexité linéaire* de $\mathcal{S}$.

Par conséquent, $L(\mathcal{S}, N)$ est la taille de la plus petite relation de récurrence linéaire qui engendre les N premiers termes de $\mathcal{S}$. Comme mentionné dans l'introduction, la complexité linéaire au rang N d'une suite correspond à la taille du plus petit LFSR qui engendre les N premiers termes de la suite. Il est alors souhaitable pour des applications en cryptographie qu'une suite possède une complexité linéaire élevée afin d'assurer qu'elle soit non prévisible trop aisément par des LFSR. La complexité linéaire peut servir de test pour déterminer si une suite possède des bonnes propriétés aléatoires, comme implémentée parmi d'autres tests dans NIST [BRS+10] et TestU01 [LS07].

Exemple I.2.4. Soit $\mathcal{S} = \texttt{0110010}\cdots \in \{0,1\}^\omega$. Le profil de complexité linéaire de $\mathcal{S}$ est

N	2	3	4	5	6	7	$\cdots$
$L(\mathcal{S}, N)$	2	2	2	3	3	4	$\cdots$

Proposition I.2.2. Soit $\mathcal{S}$ une suite sur $\mathbb{F}_q$. Alors la suite $(L(\mathcal{S}, N))_{N\geq 1}$ est une suite croissante et $L(\mathcal{S}, N) \leq N$ pour tout $N \geq 1$.

Preuve. La majoration $L(\mathcal{S}, N) \leq N$ est évidente. Si une relation de récurrence d'ordre L engendre les $N+1$ premiers termes de $\mathcal{S}$, elle engendre en particulier les N premiers termes de $\mathcal{S}$. Donc la suite $(L(\mathcal{S}, N)_{N\geq 1}$ est une suite croissante. □

Par conséquent, le profil de complexité linéaire est une suite croissante d'entiers. Nous nous restreignons à l'étude de ses sauts pour le décrire complètement.

Le théorème suivant fournit un lien entre la complexité linéaire d'une suite ultimement périodique et sa série génératrice.

Théorème I.2.4 ([TW07]**).** *Soit $\mathcal{S}$ une suite ultimement périodique de période T sur $\mathbb{F}_q$. Alors $L(\mathcal{S}) < +\infty$. De plus, soit $S(X) = s_0 + s_1 X + \cdots + s_{T-1} X^{T-1}$, on a*

$$L(\mathcal{S}) = T - \deg(\mathrm{pgcd}(X^T - 1, S(X))).$$

Le profil de complexité linéaire d'une suite est calculable à l'aide de l'algorithme de Berlekamp–Massey, voir [Jun93, Section 6.7]. Les suites avec un profil de complexité linéaire trop faible ne sont pas utilisables en cryptographie car très prévisibles, voir par exemple [Nie92] pour plus de détails. Nous donnons en annexe une implémentation de l'algorithme de Berlekamp–Massey en Python, largement reprise de [TWN20].

Exemple I.2.5. Prenons un exemple qui provient de la théorie des nombres. Un entier $n \geq 1$ est un *résidu quadratique* modulo p s'il existe $a \neq 0$ tel que $n \equiv a^2 \bmod p$. Soit $p > 2$ un nombre premier et notons $\left(\frac{\cdot}{p}\right)$ le symbole de Legendre modulo p :

$$\left(\frac{n}{p}\right) = \begin{cases} 1, & \text{si } n \text{ est un résidu quadratique modulo } p, \\ -1, & \text{si } n \text{ n'est pas un résidu quadratique modulo } p \text{ et pas un multiple de } p, \\ 0, & \text{si } n \text{ est un multiple de } p. \end{cases}$$

La *suite de Legendre* $\mathcal{L} = (\ell_n)_n$ est définie par $\ell_n = 1$ si $\left(\frac{n}{p}\right) = -1$ et $\ell_n = 0$ sinon. Cette suite est périodique de période p et donc $L(\mathcal{L}, N) \leq p$ pour tout $N \geq 1$. D'après [TW07],on a

$$L(\mathcal{L}, N) > \frac{\min(N, p)}{1 + p^{1/2}(1 + \log p)}.$$

Il n'est pas toujours aisé de déterminer de manière exacte la valeur de la complexité linéaire au rang N d'une suite. Parfois une borne inférieure permet de s'assurer de sa pertinence en cryptographie.

Théorème I.2.5 ([Sme88, Nie90])**.** *Pour presque toute suite $\mathcal{S}$ sur $\mathbb{F}_q$, on a*

$$\lim_{N \to +\infty} \frac{L(\mathcal{S}, N)}{N} = \frac{1}{2}.$$

Les auteurs de [Sme88] donnent une preuve du précédent lemme en analysant l'algorithme de Berkelamp–Massey. Dans [Nie90], l'auteur en donne une preuve combinatoire. Par conséquent, on dit qu'une suite $\mathcal{S}$ possède un profil de complexité linéaire *parfait* si pour tout $N \geq 1$,

$$|2L(\mathcal{S}, N) - N| \leq 1.$$

Les suites avec un profil de complexité linéaire parfait sont les suites *apweniennes*, à un décalage près, voir [AHN20].

Le théorème suivant donne plus de précisions que le Théorème I.2.5 sur le comportement du profil de complexité linéaire d'une suite aléatoire.

Théorème I.2.6 ([Nie88, Theorem 10])**.** *Pour presque toute suite $\mathcal{S}$ sur $\mathbb{F}_q$, on a*

$$\limsup_{N \to +\infty} \frac{L(\mathcal{S}, N) - N/2}{\log N} = \frac{1}{2 \log q},$$

et

$$\liminf_{N \to +\infty} \frac{L(\mathcal{S}, N) - N/2}{\log N} = -\frac{1}{2 \log q}.$$

Il existe plusieurs raffinements à la complexité linéaire. Un premier raffinement consiste à conserver le caractère linéaire mais nous nous autorisons à changer quelques termes dans la suite. Nous aboutissons à la notion de *complexité linéaire à k erreurs*, noté $L_k(\mathcal{S})$: pour $k \geq 0$ fixé, la complexité linéaire à k erreurs est la plus petite complexité linéaire que l'on peut obtenir en changeant au plus k termes dans $\mathcal{S}$, voir [SM93] par exemple. Il est raisonnable de penser que pour une suite aléatoire, le fait de changer quelques termes ne devrait pas impacter cette complexité trop fortement. Un deuxième raffinement de la complexité linéaire est présenté dans le paragraphe suivant.

I.2.4. Complexité d'ordre maximal. Dans ce paragraphe, nous introduisons une des notions principales de cette thèse.

Définition I.2.7. Soit $\mathcal{S} = (s_n)_{n\geq 0}$ une suite sur $\mathbb{F}_q$. Le *profil de complexité d'ordre maximal* de $\mathcal{S}$ est la suite $(M(\mathcal{S}, N))_{N\geq 1}$, où

$$M(\mathcal{S}, N) = \min\{M \in \mathbb{N} : \exists f : \mathbb{F}_q^M \to \mathbb{F}_q,\ \forall 0 \leq n \leq N - M - 1,$$
$$s_{n+M} = f(s_n, \ldots, s_{n+M-1})\}.$$

Par convention $M(\mathcal{S}, N) = 0$ si les N premiers termes de $\mathcal{S}$ sont nuls et $M(\mathcal{S}, N) = N$ si les $N - 1$ premières termes sont nuls et le N-ème est non nul. La *complexité d'ordre maximal au rang* N est $M(\mathcal{S}, N)$. De plus, $M(\mathcal{S}) := \sup_{N\geq 1} M(\mathcal{S}, N)$ est appelée *complexité d'ordre maximal* de $\mathcal{S}$.

La complexité d'ordre maximal au rang N d'une suite est donc la longueur du plus petit FSR qui engendre les N premiers termes de la suite. L'inégalité suivante est immédiate :

$$M(\mathcal{S}, N) \leq L(\mathcal{S}, N) \leq N, \quad N \geq 1.$$

Remarque I.2.3. Notons que la borne $M(\mathcal{S}, N) \leq L(\mathcal{S}, N)$ ne peut pas être améliorée en général, même si on s'attend à ce que la complexité d'ordre maximal soit bien plus faible que la complexité linéaire. Par exemple, la suite $\mathcal{S} = (s_n)_{n\geq 0} = \mathtt{011011}\cdots = (\mathtt{011})^\omega$ satisfait $M(\mathcal{S}, N) = L(\mathcal{S}, N) = 2$ pour tout $N \geq 4$. En effet $\mathtt{1}$ est le plus grand facteur spécial à droite du mot $s_0 \cdots s_{N-1}$ et $s_{n+2} = s_{n+1} + s_n \bmod 2$ pour tout $n \geq 2$.

Une suite possédant une complexité d'ordre maximal faible ne peut pas être utilisée en cryptographie et la complexité d'ordre maximal est donc un outil plus fin que la complexité linéaire. Jansen [Jan89] montre que la valeur attendue de la complexité d'ordre maximal pour une suite aléatoire est de l'ordre de grandeur de $\log N$, voir aussi [NX14].

Nous introduisons également la notion de complexité k-non-linéaire où la relation de récurrence est donnée par un polynôme de degré au plus k en chaque indéterminée. On note $\mathbb{F}_q^{(k)}[X_1, \ldots, X_m]$ l'ensemble des polynômes à m indéterminées de degré au plus k en chaque indéterminée.

Définition I.2.8. Soient $\mathcal{S} = (s_n)_{n\geq 0}$ une suite sur $\mathbb{F}_q$ et $k \geq 1$. Le *profil de complexité k-non-linéaire* de $\mathcal{S}$ est la suite $(NL_k(\mathcal{S}, N))_{N\geq 1}$ où

$$NL_k(\mathcal{S}, N) = \min\{M \in \mathbb{N} : \exists f \in \mathbb{F}_q^{(k)}[X_1, \ldots, X_m],\ \forall 0 \leq n \leq N - M - 1,$$
$$s_{n+M} = f(s_n, \ldots, s_{n+M-1})\}.$$

Par convention $NL_k(\mathcal{S}, N) = 0$ si $s_n = 0$ pour tout $0 \leq n \leq N$.

Remarque I.2.4. Dans la Définition I.2.8, nous pouvons supposer sans perte de généralité que $1 \leq k \leq q-1$. En effet, tout polynôme $f : \mathbb{F}_q^m \to \mathbb{F}_q$ s'exprime sur $\mathbb{F}_q$ en m indéterminées de degré au plus $q-1$ en chaque indéterminée, voir [LN96]. Par conséquent, pour $k \geq q-1$, tous les profils de complexité k-non-linéaire au rang N sont égaux au profil de complexité d'ordre maximal au rang N.

Lemme I.2.2. *Soit $\mathcal{S}$ une suite sur $\mathbb{F}_2$. Alors $M(\mathcal{S}, N) = NL_k(\mathcal{S}, N)$, pour tout $N \geq 1$.*

Preuve. Ce lemme se déduit de la Remarque I.2.4. □

Notons que dans ce manuscrit nous considérons essentiellement des suites sur $\mathbb{F}_2$. Par conséquent, nous utilisons les notions de complexité d'ordre maximal et de complexité k-non linéaire indifféremment l'une de l'autre.

Exemple I.2.6. Déterminons le profil de complexité d'ordre maximal de la suite $\mathcal{S} = (s_n)_{n\geq 0} = \mathtt{0110010}\cdots$ sur $\mathbb{F}_2$. Par définition on a $M(\mathcal{S}, 2) = 1$ car $s_0 \neq s_1$.
Pour que $M(\mathcal{S}, 3) = 1$, il faut qu'il existe une fonction $f(x)$ telle que

$$\begin{cases} s_2 = f(s_1), \\ s_1 = f(s_0). \end{cases}$$

Autrement dit,

$$\begin{cases} 1 = f(1), \\ 1 = f(0). \end{cases}$$

Le polynôme $f(x) = 1$ convient, donc $M(\mathcal{S}, 3) = 1$.
Pour que $M(\mathcal{S}, 4) = 1$, il faut qu'il existe une fonction $f(x)$ telle que

$$\begin{cases} s_3 = f(s_2), \\ s_2 = f(s_1), \\ s_1 = f(s_0). \end{cases}$$

Autrement dit

$$\begin{cases} 0 = f(1), \\ 1 = f(1), \\ 1 = f(0), \end{cases}$$

ce qui n'est pas possible car $f(1)$ a deux valeurs différentes. Donc $M(\mathcal{S}, 4) \geq 2$.
Pour que $M(\mathcal{S}, 4) = 2$, il faut qu'il existe une fonction $f(x, y)$ telle que

$$\begin{cases} s_3 = f(s_2, s_1), \\ s_2 = f(s_1, s_0). \end{cases}$$

Autrement dit

$$\begin{cases} 0 = f(1, 1), \\ 1 = f(1, 0). \end{cases}$$

Le polynôme $f(x, y) = x + y$ convient. Donc $M(\mathcal{S}, 4) = 2$.

On vérifie que $M(\mathcal{S},5)=M(\mathcal{S},6)=2$. Pour que $M(\mathcal{S},7)=2$, il faut qu'il existe une fonction $f(x,y)$ telle que

$$\begin{cases} s_6=f(s_5,s_4),\\ s_5=f(s_4,s_3),\\ s_4=f(s_3,s_2),\\ s_3=f(s_2,s_1),\\ s_2=f(s_1,s_0); \end{cases}$$

Autrement dit

$$\begin{cases} 0=f(1,0),\\ 1=f(0,0),\\ 0=f(0,1),\\ 0=f(1,1),\\ 1=f(1,0), \end{cases}$$

ce qui n'est pas possible car $f(1,0)$ a deux valeurs distinctes.
Pour que $M(\mathcal{S},7)=3$, il faut qu'il existe un polynôme $f(x,y,z)$ telle que

$$\begin{cases} s_6=f(s_5,s_4,s_3),\\ s_5=f(s_4,s_3,s_2),\\ s_4=f(s_3,s_2,s_1),\\ s_3=f(s_2,s_1,s_0). \end{cases}$$

Autrement dit

$$\begin{cases} 0=f(1,0,0),\\ 1=f(0,0,1),\\ 0=f(0,1,1),\\ 0=f(1,1,0). \end{cases}$$

Le polynôme $f(x,y,z)=y+z+xy$ convient. Notons qu'à cette étape, nous utilisons un polynôme de degré ≥ 2 et qu'il n'est pas possible de trouver un polynôme de degré 1. En effet, pour $f(x,y,z)=a_0x+a_1y+a_2z$, avec $a_0,a_1,a_2\in\mathbb{F}_2$, on a

$$\begin{cases} 0=f(1,0,0),\\ 1=f(0,0,1),\\ 0=f(0,1,1),\\ 0=f(1,1,0). \end{cases} \Leftrightarrow \begin{cases} 0=a_0,\\ 1=a_2,\\ 0=a_1+a_2,\\ 0=a_0+a_1. \end{cases}$$

Ce système n'a pas de solutions. Par conséquent, la complexité linéaire au rang 7 sera nécessairement plus grande que la complexité d'ordre maximal au rang 7.
Le profil de complexité d'ordre maximal et le profil de complexité linéaire de $\mathcal{S}$ sont

N	2	3	4	5	6	7	$\cdots$
$M(\mathcal{S},N)$	1	1	2	2	2	3	$\cdots$
$L(\mathcal{S},N)$	2	2	2	3	3	4	$\cdots$

Exemple I.2.7. Il est possible que $M(\mathcal{S}, N+1) - M(\mathcal{S}, N) > 1$, même si on rajoute les conditions une par une au système d'équations. Considérons par exemple la suite commençant par $\mathcal{S} = \texttt{01011}\cdots$. On peut vérifier que $M(\mathcal{S}, 4) = 1$. On a $M(\mathcal{S}, 5) \geq 3$, car le système

$$\begin{cases} s_4 = f(s_3), \\ s_3 = f(s_2), \\ s_2 = f(s_1), \\ s_1 = f(s_0). \end{cases} \Leftrightarrow \begin{cases} \mathbf{1 = f(1)}, \\ 1 = f(0), \\ \mathbf{0 = f(1)}, \\ 1 = f(0), \end{cases}$$

et le système

$$\begin{cases} s_4 = f(s_3, s_2), \\ s_3 = f(s_2, s_1), \\ s_2 = f(s_1, s_0). \end{cases} \Leftrightarrow \begin{cases} \mathbf{1 = f(1,0)}, \\ 1 = f(0,1), \\ \mathbf{0 = f(1,0)}. \end{cases}$$

n'ont pas de solutions. L'ajout de la condition sur s_4 augmente d'au moins deux niveaux la complexité d'ordre maximal de $\mathcal{S}$.

Les Exemples I.2.6 et I.2.7 montrent comment calculer la complexité d'ordre maximal en général. Lorsque nous effectuons le calcul étape par étape de la complexité d'ordre maximal, nous remarquons que la complexité augmente lorsqu'une condition non satisfaisable est ajoutée au système. Jansen [Jan89] étudie en détail ce phénomène dans sa thèse. Nous donnons sa formulation de ce lemme important et reproduisons sa preuve.

Lemme I.2.3 ([Jan89, Proposition 3.1]**).** *Soit $\mathcal{S}$ une suite sur $\mathbb{F}_2$. Soit k la longueur de la plus longue sous-suite d'éléments consécutifs de $\mathcal{S}$ qui apparaît au moins deux fois avec deux successeurs différents dans les n premières valeurs. Alors la complexité d'ordre maximal au rang n de $\mathcal{S}$ est égale $k+1$.*

Preuve. Comme évoqué précédemment, la complexité d'ordre maximal au rang N est la taille du plus petit FSR qui engendre les N premières valeurs de la suite. À toute fonction de retour F, on associe une table de vérité : une liste d'arguments avec leurs valeurs correspondantes. Une fonction de retour F a une table de vérité *propre* si chaque argument correspond à une valeur unique. Par conséquent, pour obtenir une table de vérité propre pour la fonction de retour, la complexité d'ordre maximal au rang N de $\mathcal{S}$ doit être égal à 1 de plus que la longueur que la plus longue sous-suite qui apparaît au moins deux fois avec deux successeurs différents. □

Nous proposons une nouvelle formulation du lemme précédent, adaptée aux notions de la combinatoire des mots.

Lemme I.2.4. *Soit $\mathcal{S} = (s_n)_{n\geq 0}$ une suite sur $\mathbb{F}_2$. On note $w = s_0 s_1 \cdots$ le mot infini obtenu à l'aide des valeurs de $\mathcal{S}$. Soit k la longueur du plus long facteur spécial à droite de $w[0\cdots(N-1)] = s_0 \cdots s_{N-1}$. Alors on a $M(\mathcal{S}, N) = k+1$.*

Preuve. La preuve est immédiate par définition d'un facteur spécial à droite. □

Exemple I.2.8. Reprenons l'exemple du mot $w = \texttt{0110010}\cdots$. Nous listons à chaque étape un facteur spécial à droite de longueur maximale :

N	$w[0\cdots(N-1)]$	Plus long facteur spécial à droite	$M(\mathcal{S},N)$
2	01	ε	$\lvert\varepsilon\rvert+1=1$
3	011	ε	$\lvert\varepsilon\rvert+1=1$
4	0110	1	$\lvert 1\rvert+1=2$
5	01100	1	$\lvert 1\rvert+1=2$
6	011001	1	$\lvert 1\rvert+1=2$
7	0110010	01	$\lvert 01\rvert+1=3$

Pour une suite $\mathcal{S}=(s_n)_{n\geq 0}$ sur $\mathbb{F}_2$, l'opposée de $\mathcal{S}$ est la suite $\overline{\mathcal{S}}=(\overline{s_n})_{n\geq 0}$ où $\overline{s_n}=(s_n+1) \bmod 2$.

Proposition I.2.3 ([Jan89, Corollary 3.4]**).** Soit $\mathcal{S}$ une suite sur $\mathbb{F}_2$. Pour tout $N\geq 1$, $M(\mathcal{S},N)=M(\overline{\mathcal{S}},N)$.

Preuve. Soit $N\geq 1$. Un facteur spécial à droite du mot $s_0\cdots s_{N-1}$ est également un facteur spécial à droite du mot $\overline{s_0}\cdots\overline{s_{N-1}}$ et réciproquement. □

Remarque I.2.5. Contrairement au profil de complexité d'ordre maximal, le profil de complexité linéaire d'une suite n'est pas égal à celui de son opposée. En effet, pour la suite binaire $\mathcal{S}=0110010\cdots$ et son opposée $\overline{\mathcal{S}}=1001101\cdots$, leurs profils de complexité linéaire sont :

N	2	3	4	5	6	7	$\cdots$
$L(\mathcal{S},N)$	2	2	2	3	3	4	$\cdots$
$L(\overline{\mathcal{S}},N)$	1	1	3	3	4	4	$\cdots$

Pour des résultats plus généraux sur la complexité d'ordre maximal, voir [Jan89, Jan91].

Le profil de complexité d'ordre maximal d'une suite $\mathcal{S}$ est calculable en construisant le graphe acyclique orienté du mot associé à $\mathcal{S}$, ou DAWG (Directed Acyclic Word Graph) en anglais, du mot associé. Pour plus de détails, voir le Chapitre V.

La complexité d'ordre maximal a été étudiée par de nombreux auteurs. Les articles [IW17, NX14, TW07] exhibent des liens entre la complexité d'ordre maximal et d'autres mesures de complexité introduites dans la suite de cette thèse. Les articles [SW19b, SW19a] portent sur la complexité d'ordre maximal de la suite de Thue–Morse et l'article de Sun, Zeng et Lin sur la complexité d'ordre maximal d'un analogue à la suite de Rudin–Shapiro [SZL20].

I.2.5. Mesure de corrélation d'ordre k**.** Dans ce paragraphe, nous nous intéressons uniquement à des suites binaires sur l'alphabet $\{-1,1\}$ pour des raisons de

simplicité d'écriture. La série d'articles "On finite pseudorandom binary sequences" I–VII, initiée par Mauduit et Sárközy, étudie plusieurs mesures de complexité pour les suites pseudo–aléatoires binaires.

Soit $\mathcal{S}$ une suite sur $\{-1,1\}$ et notons $\mathcal{S}_N$ la suite finie de ses N premiers termes. Soient k, M des entiers, $D = (d_1, \dots, d_k) \in \mathbb{N}^k$ tel que $d_1 < d_2 < \cdots < d_k$. Notons

$$V(\mathcal{S}_N, M, D) = \sum_{0 \le n \le M-1} s_{n+d_1} s_{n+d_2} \cdots s_{n+d_k}.$$

Nous introduisons la mesure de corrélation d'ordre k, voir [MS97].

Définition I.2.9. Soit $k \ge 2$ un entier. La *mesure de corrélation d'ordre k* de $\mathcal{S}$, notée $C_k(\mathcal{S}, N)$, est définie par :

$$C_k(\mathcal{S}, N) = \max_{M,D} |V(\mathcal{S}_N, M, D)| = \max_{M,D} \left| \sum_{0 \le n \le M-1} s_{n+d_1} s_{n+d_2} \cdots s_{n+d_k} \right|,$$

où le maximum est pris sur tous les $D = (d_1, \dots, d_k) \in \mathbb{N}^k$ et M tels que $M + d_k \le N$.

La mesure de corrélation d'ordre k mesure si une suite et ses décalages ont peu de valeurs différentes. Si tel est le cas, le nombre d'annulations dans la somme $V(\mathcal{S}_N, M, D)$ est faible et sa mesure de corrélation d'ordre k sera grande. Ainsi, pour une suite binaire aléatoire, il est "raisonnable" de penser que ses mesures de corrélations d'ordre k seront faibles. Cassaigne, Mauduit et Sárközy [CMS02] ont précisé ce postulat dans le théorème suivant.

Théorème I.2.7 ([CMS02]). *Pour tout $k \ge 2$ et pour tout réel $\varepsilon > 0$, il existe $N_0 = N_0(\varepsilon, k)$ et $\delta = \delta(\varepsilon, k) > 0$ tels que pour tout $N \ge N_0$ on a avec probabilité $> 1 - 2\varepsilon$,*

$$\delta\sqrt{N} < C_k(\mathcal{S}, N) < 5\sqrt{kN \log N}.$$

Ce résultat est amélioré par Alon et al. [AKM+07].

Théorème I.2.8 ([AKM+07]). *Pour tout $\varepsilon > 0$, il existe un $N_0 = N_0(\varepsilon)$ tel que pour tout $N \ge N_0$, la probabilité que*

$$\frac{2}{5}\sqrt{N \log \binom{N}{k}} < C_k(\mathcal{S}, N) < \frac{7}{4}\sqrt{N \log \binom{N}{k}}$$

pour tout entier k tel que $2 \le k \le N/4$, est supérieure ou égale $1 - \varepsilon$.

Dans le Chapitre VII, nous donnons un nouveau résultat sur la corrélation d'ordre k de la suite de Rudin–Shapiro le long des nombres premiers.

I.2.6. Complexité d'expansion. Pour un polynôme à deux indéterminées ou plus, le *degré* d'un terme est la somme des exposants des indéterminées dans ce monôme. Le *degré total*, noté degtot, d'un polynôme à deux indéterminées ou plus est le maximum des degrés de ces monômes. Par exemple le polynôme $f(x, y, z) = xy^2z + x^2z^3$ est de degré total 5.

Pour une suite $\mathcal{S} = (s_n)_{n\geq 0}$ sur $\mathbb{F}_q$, on note $G_{\mathcal{S}}(x)$ sa série génératrice définie par

$$G_{\mathcal{S}}(x) = \sum_{i\geq 0} s_i x^i.$$

La complexité d'expansion a été introduite par Diem [Die12].

Définition I.2.10 ([Die12]**).** Soit $\mathcal{S} = (s_n)_{n\geq 0}$ une suite sur $\mathbb{F}_q$. Le *profil de complexité d'expansion* de $\mathcal{S}$ est la suite $(E(\mathcal{S}, N))_{N\geq 1}$, où

$$E(\mathcal{S}, N) = \min\{d \in \mathbb{N} : \exists h \in \mathbb{F}_2[x, y] \setminus \{0\},\ \mathrm{degtot}(h) = d,$$
$$h(x, G_{\mathcal{S}}(x)) \equiv 0 \pmod{x^N}\},$$

si $s_0, \ldots, s_{N-1}$ sont non tous nuls, $E(\mathcal{S}, N) = 0$ sinon. La *complexité d'expansion au rang* N est le N-ème terme du profil de complexité d'expansion, le terme $E(\mathcal{S}, N)$. De plus, $E(\mathcal{S}) := \sup_{N\geq 1} E(\mathcal{S}, N)$ est appelée *complexité d'expansion* de $\mathcal{S}$.

Diem [Die12] démontre que les suites avec une complexité d'expansion faible sont prévisibles avec relativement peu de termes.

Théorème I.2.9 ([Die12]**).** *Soit $\mathcal{S}$ une suite sur $\mathbb{F}_q$ telle que $E(\mathcal{S}) = d$. À partir des d^2 premiers éléments, on peut déterminer un polynôme irréductible $h(x, y) \in \mathbb{F}_q[x, y]$ de degré $\deg h \leq d$ tel que $h(x, G_{\mathcal{S}}) = 0$ en un temps polynomial en $d \cdot \log q$.*

Pour prouver ce théorème, Diem utilise des méthodes algébriques, notamment issues de la théorie algébrique des nombres.

Théorème I.2.10 ([GPMN18]**).** *Soit $\mathcal{S}$ une suite sur $\mathbb{F}_q$. La complexité d'expansion au rang N de $\mathcal{S}$ satisfait la majoration suivante :*

$$\binom{E(\mathcal{S}, N) + 1}{2} \leq N$$

Le comportement de la complexité d'expansion d'une suite aléatoire est donnée par le théorème suivant.

Théorème I.2.11 ([GPM20, MNW17]**).** *Pour presque toute suite $\mathcal{S}$ sur $\mathbb{F}_q$,*

$$\liminf_{N\to+\infty} \frac{E(\mathcal{S}, N)}{\sqrt{N}} \geq \frac{\sqrt{2}}{2}.$$

Il est relativement difficile d'obtenir des résultats généraux ou spécifiques sur la complexité d'expansion et très peu de résultats sont connus, voir la bibliographie relativement restreinte des articles cités précédemment.

Diem [Die12] montre également que les suites possédant un profil de complexité linéaire parfait ne sont pas utilisables en cryptographie car sont vulnérables à une attaque basée sur leur série génératrice. Par conséquent, il est important de connaître la complexité d'expansion d'une suite pour en déduire de sa pertinence en cryptographie.

I.2.7. Ordres de grandeur attendus pour une suite aléatoire. Dans ce paragraphe, nous rappelons tous les ordres de grandeur attendus des mesures de complexité introduites dans les paragraphes précédents pour une suite aléatoire sur $\Sigma_q = \{0, 1, \ldots, q-1\}$.

Mesure de complexité	Ordre de grandeur attendu pour $\mathcal{S}$
Complexité en facteurs $p_{\mathcal{S}}(N)$	q^N
Normalité	Oui
Profil de complexité linéaire $L(\mathcal{S}, N)$	$N/2$
Profil de complexité d'ordre maximal $M(\mathcal{S}, N)$	$\log(N)$
Complexité d'expansion $E(\mathcal{S}, N)$	$2\sqrt{N}$
Corrélation d'ordre k, $C_k(\mathcal{S}, N)$	$\sqrt{N \log \binom{N}{k}}$

TABLEAU I.1. Ordres de grandeur pour les mesures de complexité introduites.

I.2.8. Liens entre les mesures de complexité. Dans ce paragraphe, nous présentons différentes relations d'interdépendance entre les mesures de complexité. Soit $\mathcal{S} = (s_n)_{n\geq 0}$ une suite sur $\mathbb{F}_2$.

Complexité d'ordre maximal et Complexité linéaire

Nous avons déjà vu que pour tout $N \geq 1$, on a

$$M(\mathcal{S}, N) \leq L(\mathcal{S}, N).$$

Par conséquent, une complexité d'ordre maximal grande implique une complexité linéaire grande.

Complexité linéaire et Complexité d'expansion

Mérai, Niederreiter et Winterhof [MNW17] montrent qu'une complexité linéaire grande, c'est-à-dire proche de N, implique une complexité d'expansion faible.
Plus précisément, soit $N \geq 2$, $L_N = L(\mathcal{S}, N)$ sa complexité linéaire au rang N. Soient $(c_\ell)_{\ell_N \leq \ell \leq L_N}$ tels que $\sum_{\ell=\ell_N}^{L_N} c_\ell s_{i+\ell} = 0, \quad 0 \leq i \leq N - L_N - 1$, avec $c_{L_N} = 1$ et $c_{\ell_N} \neq 0$. Alors, d'après [MNW17], on a

$$E(\mathcal{S}, N) \geq \begin{cases} L_N - \ell_N + 1, \text{ pour } N > (L_N - \ell_N)(L_N - \min\{1, \ell_N - 1\}), \\ \left\lceil \frac{N}{L_N - \min\{1, \ell_N - 1\}} \right\rceil, \text{ sinon,} \end{cases}$$

et

$$E(\mathcal{S}, N) \leq \min\{L_N + \max\{-1, 1 - \ell_N\}, N - L_N + 2\}.$$

Complexité linéaire et Corrélation d'ordre k

Brandstätter et Winterhof [BW06] montrent qu'une suite avec des mesures de corrélation d'ordre k faibles, pour un k suffisamment grand, possède également une grande complexité linéaire. Plus précisément, pour tout $N \geq 1$, on a

$$L(\mathcal{S}, N) \geq N - \max_{1 \leq k \leq L(S,N)+1} C_k(\mathcal{S}, N).$$

Complexité d'ordre maximal et Corrélation d'ordre k

Işik et Winterhof [IW17] montrent que pour tout $N \geq 1$, on a

$$M(\mathcal{S}, N) \geq N - 2^{M(\mathcal{S},N)+1} \max_{1 \leq k \leq M(\mathcal{S},N)+1} C_k(\mathcal{S}, N).$$

De plus

$$C_2(\mathcal{S}, N) \geq M(\mathcal{S}, N) - 1. \tag{I.2.1}$$

En effet, d'après le Lemme I.2.3, il existe $0 \leq n \leq m \leq N - M(\mathcal{S}, N) - 1$ tels que

$$s_{n+i} = s_{m+i}, \qquad 0 \leq i \leq M(\mathcal{S}, N) - 2,$$
$$s_{n+M(\mathcal{S},N)-1} \neq s_{m+M(\mathcal{S},N)-1}.$$

Donc $M(\mathcal{S}, N) - 1 = \sum_{i=0}^{M(\mathcal{S},N)-2} (-1)^{s_{n+i}+s_{m+i}} \leq C_2(\mathcal{S}, N)$.

Complexité en facteurs et Corrélation d'ordre k

Cassaigne et al. [CCSS14] montrent que si pour des entiers k et N, la corrélation d'ordre ℓ pour tout $1 \leq \ell \leq k$ satisfait

$$C_\ell(\mathcal{S}, N) \leq \frac{N}{2^{2k+1}},$$

alors $p_{\mathcal{S}}(k) = 2^k$.

Complexité d'ordre maximal et Complexité d'expansion

Aucune formule reliant $E(\mathcal{S}, N)$ et $M(\mathcal{S}, N)$ qui soit valable pour toute suite $\mathcal{S}$ n'est connue. Pour une suite aléatoire, l'ordre de grandeur attendu de $E(\mathcal{S}, N)$ est plus grand que celui de $M(\mathcal{S}, N)$. Cependant, nous constaterons que pour les suites automatiques, la complexité d'expansion est faible alors que la complexité d'ordre maximal est grande.

Par conséquent, une suite ne peut pas avoir toutes ses mesures de complexité "meilleures" qu'une suite aléatoire. Une suite avec une "bonne" mesure pour une mesure de complexité et une "mauvaise" mesure pour une autre mesure de complexité est considérée comme cryptographiquement faible, car sensible à certains types d'attaques ciblés sur leur "mauvaise" mesure de complexité.

Complexité de Kolmogorov et Complexité linéaire

Une suite pseudo-aléatoire n'est pas aléatoire par essence puisque elle est engendrée par un algorithme déterministe. La complexité théorique qui décrit le plus petit algorithme qui engendre une suite est la *complexité de Kolmogorov*. Dans cette thèse, nous ne poursuivons pas l'étude de cette mesure de complexité. Notons, au passage, que la complexité de Kolmogorov n'est pas calculable, c'est-à-dire qu'il n'existe pas d'algorithme qui calcule la complexité de Kolmogorov pour toute suite; mais il existe un lien entre cette mesure et la complexité linéaire, voir [Wan99].

I.2.9. Équirépartition et méthodes de Monte–Carlo.

Définition I.2.11. Une suite $\mathcal{S} = (s_n)_{n\geq 0}$ de nombres réels est dite *équirépartie* sur un intervalle $[a, b]$ si pour tout sous-intervalle $[c, d]$ de $[a, b]$, on a

$$\lim_{n\to+\infty} \frac{\text{Card}\left(\{s_0, \ldots, s_{n-1}\} \cap [c, d]\right)}{n} = \frac{d-c}{b-a}.$$

Une suite $\mathcal{S} = (s_n)_{n\geq 0}$ de nombres réels est dite *équirépartie modulo 1* si la suite de ses parties fractionnaires $(\{s_n\})_{n\geq 0}$ est équirépartie sur l'intervalle $[0, 1]$.

On note $e(x) = \exp(2i\pi x)$ pour un réel x. Le critère de Weyl est un résultat fondamental de cette théorie.

Proposition I.2.4 (Critère de Weyl). Soit $\mathcal{S} = (s_n)_{n\geq 0}$ une suite de nombres réels. Les assertions suivantes sont équivalentes.

(i) $\mathcal{S}$ est équirépartie modulo 1.

(ii) Pour tout entier ℓ non nul, on a

$$\lim_{N\to+\infty} \frac{1}{N} \sum_{n=0}^{N-1} e(\ell s_n) = 0.$$

(iii) Pour toute fonction Riemann-intégrable $f : [0, 1] \to \mathbb{C}$, on a

$$\lim_{N\to+\infty} \frac{1}{N} \sum_{n=1}^{N} f(\{s_n\}) = \int_0^1 f(x)dx.$$

L'assertion (iii) de la Proposition I.2.4 est le point de départ de la méthode d'intégration de Monte-Carlo, appelée *estimation de Monte-Carlo*. Si la suite $\mathcal{S}$ est aléatoire, le terme d'erreur de l'estimation de Monte-Carlo est de l'ordre de $N^{-1/2}$. Le principe des méthodes de quasi-Monte-Carlo est de choisir de manière déterministe la suite $\mathcal{S}$ pour améliorer ce terme d'erreur. Les méthodes de quasi-Monte-Carlo sont une application importante de la génération de nombres pseudo-aléatoires.

Remarque I.2.6. Les suites avec un profil de complexité linéaire trop faible ne sont pas utilisables pour une méthode de quasi-Monte-Carlo, voir [Nie92] pour plus de détails.

I.3. Langages et suites automatiques

Dans ce paragraphe, nous suivons les définitions présentes dans [AS03, Rig14]. Soit Σ un alphabet fini de taille q.

I.3.1. Langages. Un *langage* sur Σ est un sous-ensemble de Σ^*. Pour deux langages L_1, L_2 sur Σ, nous définissons les deux opérations suivantes :

- Produit : $L_1L_2 = \{wx : w \in L_1, x \in L_2\}$.
- Clôture de Kleene : $L^* = \cup_{i\geq 0} L^i$ avec $L^0 = \{\varepsilon\}$ et $L^i = LL^{i-1}$.

Pour un mot w, l'ensemble de ses facteurs $\mathrm{Sub}(w)$ est appelé *langage de* w, et est également noté $L(w)$.

Définition I.3.1. Un langage L est *régulier* sur un alphabet Σ s'il satisfait une des conditions suivantes :

- L est vide.
- $L = \{a\}$, pour une lettre $a \in \Sigma$.
- $L = A^*$, pour A un langage régulier.
- $L = A \cup B$ ou $L = AB$, pour A, B deux langages réguliers.

Exemple I.3.1. Le langage $1^*0^* := \{1^i0^j : i, j \geq 0\}$ est un langage régulier car c'est le produit des deux langages 1^* et 0^*, qui sont eux même réguliers.

Lemme I.3.1 (Lemme de l'étoile). *Soit L un langage régulier. Il existe une constante $n \geq 1$ telle que pour tout $z \in L$ avec $|z| \geq n$, il existe une décomposition $z = uvw$ avec $|uv| \leq n$, $|v| \geq 1$ tel que $uv^iw \in L$ pour tout $i \geq 0$.*

Preuve. Pour une preuve de ce lemme, voir [AS03, Lemma 4.2.1]. □

Le lemme de l'étoile est généralement utiliser pour prouver la non-régularité d'un langage.

Exemple I.3.2.

- Le langage $L = \{1^j0^j : j \geq 0\}$ n'est pas un langage régulier, voir [AS03, Example 4.2.2].
- Le langage d'un mot sturmien n'est pas un langage régulier, voir [Fog02, Corollary 6.1.11].

I.3.2. Suites automatiques. Les suites automatiques peuvent être définies de plusieurs manières : par des automates, par des morphismes, par leurs séries génératrices et par leurs noyaux. Nous redonnons ici les différentes définitions ainsi que quelques propriétés, voir [AS03] pour plus de détails.

Définition par les automates

Définition I.3.2. Un *Automate Fini Déterministe* M, ou en anglais "Deterministic Finite Automaton" (DFA), est un quintuplet $M = (Q, \Sigma, \delta, q_0, F)$ où

- Q est un ensemble fini, dont les éléments sont appelés *états*,
- Σ est un alphabet fini,
- $\delta : Q \times \Sigma \to Q$ est une fonction, appelée *fonction de transition*,

- $q_0 \in Q$ est un état, appelé *état initial*,
- $F \subset Q$ est un ensemble, appelé *ensemble des états acceptants*.

Soit $M = (Q, \Sigma, \delta, q_0, F)$ un automate fini déterministe. On étend la fonction δ à $Q \times \Sigma^*$ de la manière suivante :

- $\delta(q, \varepsilon) = q$ pour tout $q \in Q$,
- $\delta(q, xa) = \delta(\delta(q, x), a)$ pour tout $q \in Q, x \in \Sigma^*$ et $a \in \Sigma$.

Le langage $L(M)$ défini par $L(M) = \{w \in \Sigma^* : \delta(q_0, w) \in F\}$ est appelé le *langage reconnu*, ou le *langage accepté*, par M.

Exemple I.3.3. Soit M l'Automate Fini Déterministe défini par :

- $Q = \{q_0, q_1, q_2\}$, q_0 est l'état initial,
- $\Sigma = \{0, 1\}$,
- $\delta : Q \times \Sigma \to Q$ défini par :
  - $\delta(q_0, 0) = q_1, \qquad \delta(q_0, 1) = q_0,$
  - $\delta(q_1, 0) = q_1, \qquad \delta(q_1, 1) = q_2,$
  - $\delta(q_2, 0) = \delta(q_2, 1) = q_2.$
- $F = \{q_0, q_1\}$.

Le parcours d'un mot fini w termine dans l'état q_2 si et seulement si le mot 01 est un facteur de w. Par conséquent, le langage $L(M)$ est l'ensemble des mots finis sur $\{0, 1\}$ de la forme $1^i0^j, i, j \geq 0$. D'après l' Exemple I.3.1, $L(M)$ est un langage régulier. L'automate M est représenté dans la Figure I.3. Les états acceptants sont tracés par un double cercle et l'état initial possède une flèche entrante non étiquetée.

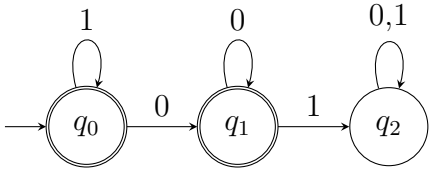


FIGURE I.3. DFA acceptant les mots binaires de la forme 1^i0^j, pour des entiers $i, j \geq 0$.

Théorème I.3.1 (Théorème de Kleene, [AS03, Theorem 4.1.5]**).** *Un langage L est reconnu par un automate fini déterministe si et seulement s'il est régulier.*

Le théorème de Kleene est un théorème fondamental qui relie la théorie des automates et la théorie des langages.

Définition I.3.3. Un *Automate Fini Déterministe avec Sorties M*, ou Deterministic Finite Automaton with Output (DFAO) en anglais, est un sextuplet $M = (Q, \Sigma, \delta, q_0, \Sigma', \tau)$ tel que

- $(Q, \Sigma, \delta, q_0, Q)$ est un DFA.
- Σ' est un alphabet fini.
- $\tau : Q \to \Sigma'$ est une fonction, appelée *fonction de sortie*.

Exemple I.3.4.

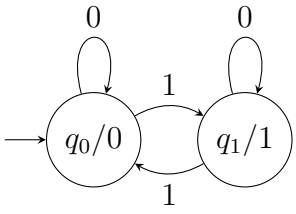


FIGURE I.4. DFAO calculant la somme des chiffres en base 2 modulo 2.

Le parcours d'un mot fini w termine à l'état q_0 (resp. q_1) si et seulement si le nombre d'occurrences de 1 dans w est pair (resp. impair). Par conséquent, pour un entier n et sa représentation en base 2, notée $(n)_2$, l'automate calcule bien la somme des chiffres en base 2 modulo 2.

Théorème I.3.2 ([AS03, Theorem 4.3.1]**).** *Soit $M = (Q, \Sigma, \delta, q_0, \Sigma', \tau)$ un Automate Fini Déterministe avec Sorties. Pour tout $d \in \Sigma'$, l'ensemble*

$$L_d(M) = \{w \in \Sigma^* : \tau(\delta(q_0, w)) = d\}$$

est un langage régulier.

Nous pouvons désormais introduire une première définition des suites automatiques. Rappelons que pour $k \geq 2$, la représentation d'un entier naturel n en base k est le mot fini $(n)_k$. Pour l'alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$, $k \geq 2$, nous utilisons le terme de k-DFAO pour désigner un DFAO sur Σ_k.

Définition I.3.4. Une suite $(a_n)_{n\geq 0}$ est *k-automatique* s'il existe un k-DFAO $M = (Q, \Sigma_k, \delta, q_0, \Sigma', \tau)$ tel que pour tout $n \geq 0$, on a

$$a_n = \tau(\delta(q_0, w)).$$

Dans ce cas, on dit que l'automate M *engendre* la suite $(a_n)_{n\geq 0}$. De manière générale, une suite est dite *automatique* si elle est k-automatique pour un certain $k \geq 2$.

Définition I.3.5. La suite de Thue–Morse $\mathcal{T} = (t_n)_{n\geq 0}$ est la suite engendrée par l'automate fini déterministe avec sortie de la Figure I.4.

Définition I.3.6. La suite de Rudin–Shapiro $\mathcal{R} = (r_n)_{n\geq 0}$ est la suite engendrée par l'automate suivant :

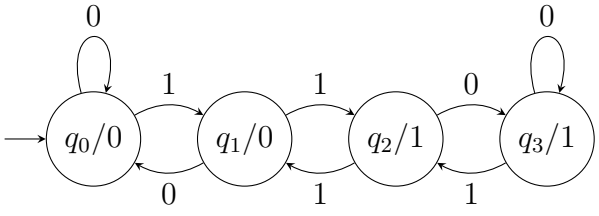


FIGURE I.5. DFAO générant la suite de Rudin–Shapiro $\mathcal{R}$.

Définition par les morphismes uniformes

Soient Σ, Σ' deux alphabets finis. Une application $f : \Sigma^* \to \Sigma'^*$ est un *morphisme* si $f(uv) = f(u)f(v)$ pour tout $u, v \in \Sigma^*$. Un morphisme est dit *k-uniforme* si pour tout $a \in \Sigma$, alors $|f(a)| = k$. Un *codage* est un morphisme 1-uniforme.

Soient $k \geq 2$ et f un morphisme k-uniforme. Le morphisme f est *prolongeable en a* s'il existe un $x \in \Sigma^* \setminus \{\varepsilon\}$, tel que $f(a) = ax$. Dans ce cas, le mot infini

$$u = f^\omega(a) := axf(x)f^2(x)\cdots$$

est l'unique point fixe de f commençant par a.

Le théorème suivant caractérise les suites automatiques par les morphismes uniformes.

Théorème I.3.3 ([Cob72])**.** *Soit $k \geq 2$. Une suite $\mathcal{S} = (s_n)_{n\geq 0}$ est k-automatique si et seulement si le mot infini $w = s_0 s_1 \cdots$ est l'image d'un point fixe d'un morphisme k-uniforme, à un codage près.*

La suite de Thue–Morse $\mathcal{T}$ est le point fixe commençant par 0 du morphisme suivant :

$$f : \begin{cases} 0 \mapsto 01 \\ 1 \mapsto 10. \end{cases} \tag{I.3.1}$$

La suite de Rudin–Shapiro est égale à $\pi \circ f^\omega(a)$, où

$$f : \begin{cases} a \mapsto ab, & b \mapsto ac, \\ c \mapsto db, & d \mapsto dc, \end{cases} \qquad \pi : \begin{cases} a, b \mapsto 0, \\ c, d \mapsto 1. \end{cases}$$

Une suite $\mathcal{S}$ périodique est k-automatique pour tout $k \geq 2$, voir [AS03, Theorem 5.4.2]. Le théorème suivant est, en quelque sorte, la réciproque de ce résultat. Rappelons que deux entiers $k, \ell \geq 2$ sont dits *multiplicativement indépendants* si pour tout $a, b \geq 1$, on a $k^a \neq \ell^b$.

Théorème I.3.4 ([Cob69])**.** *Soient k, ℓ deux entiers multiplicativement indépendants. Si une suite $\mathcal{S}$ est k-automatique et ℓ-automatique, alors $\mathcal{S}$ est ultimement périodique.*

Définition I.3.7. Soient deux entiers $n \geq 0$ et $q \geq 2$. On note $s_q(n)$ la somme des chiffres en base q, c'est-à-dire

$$s_q(n) = \sum_{i\geq 0} \varepsilon_i,$$

où $n = \sum_{i\geq 0} \varepsilon_i q^i$ et $0 \leq \varepsilon_i < q$ pour tout $i \geq 0$.

Proposition I.3.1. Soit $q \geq 2$. La suite $(s_q(n) \bmod q)_{n\geq 0}$ est une suite automatique.

Preuve. La suite $(s_q(n) \bmod q)_{n\geq 0}$ est égale à $f_q^\omega(0)$, où

$$f_q : \begin{cases} 0 & \mapsto 01\cdots(q-2)(q-1) \\ 1 & \mapsto 12\cdots(q-1)0 \\ \vdots & \\ q-1 & \mapsto (q-1)0\cdots(q-3)(q-2). \end{cases}$$

En effet, ce morphisme est une généralisation du morphisme qui définit la suite de Thue–Morse. □

Nous introduisons une famille de suites automatiques, étudiée dans le Chapitre II.

Proposition I.3.2. Soient $k \geq 1$ et $\omega_k = \mathtt{1}^k \in \Sigma_2$. On note $s_{2,k}(n)$ le nombre d'occurrences de ω_k dans le mot $(n)_2$. La suite $\mathcal{P}_k = (p_k(n))_{n\geq 0}$ est une suite 2-automatique, où $p_k = s_{2,k} \bmod 2$.

Preuve. La suite $\mathcal{P}_k$ est égale à $\pi \circ f^\omega(a_0)$, où

$$f : \begin{cases} a_0 & \mapsto a_0a_1 \\ a_1 & \mapsto a_0a_2 \\ \vdots & \\ a_{k-1} & \mapsto a_0a_k \\ a_k & \mapsto a_{2k-1}a_{k-1} \\ a_{k+1} & \mapsto a_{2k-1}a_k \\ \vdots & \\ a_{2k-2} & \mapsto a_{2k-1}a_{2k-3} \\ a_{2k-1} & \mapsto a_{2k-1}a_{2k-2} \end{cases} \quad \text{et} \quad \pi : \begin{cases} a_0, \ldots, a_{k-1} & \mapsto 0 \\ a_k, \ldots, a_{2k-1} & \mapsto 1 \end{cases}$$

En effet, ce morphisme est une généralisation du morphisme qui définit la suite de Rudin–Shapiro. □

Définition I.3.8. La suite $\mathcal{P}_k$ est appelée *suite de motifs d'ordre k.*

Par conséquent, la suite de Thue–Morse est la suite de motifs d'ordre 1, c'est-à-dire $\mathcal{T} = P_1$. La suite de Rudin–Shapiro est quant à elle la suite de motifs d'ordre 2, c'est-à-dire $\mathcal{R} = P_2$. Le terme de suite de motifs a été utilisé dans les articles [SW19a, SW19b].

Définition par les noyaux

Définition I.3.9. Le *k-noyau* d'une $\mathcal{S} = (s_n)_{n\geq 0}$ est l'ensemble des sous-suites de $\mathcal{S}$ défini par :

$$K_k(\mathcal{S}) = \{(s_{k^i n+j}) : i \geq 0,\ 0 \leq j < k^i\}.$$

Le théorème suivant fournit une condition nécessaire et suffisante pour qu'une suite soit k-automatique en fonction du cardinal de $K_k(\mathcal{S})$.

Théorème I.3.5 ([Chr79]**).** *Soit $k \geq 2$. Une suite $\mathcal{S}$ est k-automatique si et seulement si $K_k(\mathcal{S})$ est fini.*

Considérons à nouveau l'exemple de la suite de Thue–Morse $\mathcal{T} = (t_n)_{n\geq 0}$. Comme $t_{2n} = t_n$ et $t_{2n+1} \equiv t_n + 1 \bmod 2$, on a que $K_2(\mathcal{T})$ contient exactement 2 suites distinctes, la suite $\mathcal{T}$ et son opposée :

$$\mathtt{01101001100101101}\cdots,$$
$$\mathtt{10010110011010010}\cdots.$$

Notons que la deuxième suite est le point fixe commençant par $\mathtt{1}$ du morphisme défini dans (I.3.1).

Le théorème suivant est le premier résultat que nous introduisons traitant des sous-suites d'une suite automatique. Il postule que toutes les sous-suites d'une suite automatique indexées par un polynôme de degré 1 sont également des suites automatiques.

Théorème I.3.6 ([Cob72])**.** *Soit $\mathcal{S} = (s_n)_{n\geq 0}$ une suite k-automatique. Pour tous entiers $a, b \geq 0$, la suite $\mathcal{S}_{a,b} = (s_{an+b})_{n\geq 0}$ est k-automatique.*

Le théorème suivant fournit sous quelles conditions les sous-suites polynomiales de $(s_q(n) \bmod m)_{n\geq 0}$ sont automatiques.

Théorème I.3.7 ([All82])**.** *Soient q, m deux entiers tels que $q \geq 2$ et $m \geq 1$. Soit $P(X) \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$. La suite $(s_q(P(n)) \mod m)_{n\geq 0}$ est q-automatique si et seulement si $\deg P \leq 1$ ou $m|(q-1)$.*

Ce théorème est l'un des points de départ de cette thèse. Il montre que l'automaticité d'une suite peut ne pas être conservée le long de ses sous-suites polynomiales. Alors que les suites automatiques sont "facilement" générées, très prévisibles et non aléatoires, leurs sous-suites polynomiales ne sont pas nécessairement automatiques et deviennent donc de potentiels candidats pour des suites pseudo-aléatoires.

Remarque I.3.1. Il n'existe pas de résultat général sur l'automaticité des sous-suites polynomiales d'une suite automatique. Nous donnons un exemple où la sous-suite des carrés d'une suite automatique est également une suite automatique. Soit $C_n = \frac{1}{n+1}\binom{2n}{n}$ le nombre de Catalan d'indice n. Pour tout entier $n \geq 1$, C_n est impair si et seulement si $n = 2^k - 1$ pour un entier $k \geq 0$. Par conséquent, la suite $\mathcal{C} = (C_n \mod 2)_{n\geq 0}$ est automatique, engendrée par l'automate de la Remarque I.3.1 :

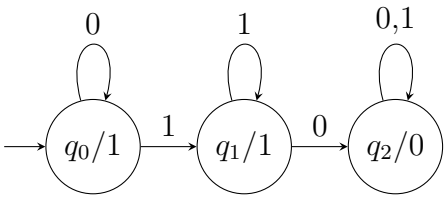


FIGURE I.6. DFAO générant la suite $(C_n \mod 2)_{n\geq 0}$.

La suite $\mathcal{C}$ le long des carrés est la suite 10^ω. En effet $n^2 = 2^k - 1$ si et seulement si $n = k = 1$. La suite 10^ω est automatique car ultimement périodique. Cet exemple montre que pour obtenir de potentiels candidats de suites pseudo-aléatoires à partir d'une suite automatique, cette dernière doit être choisie judicieusement.

Définition par la suite génératrice

Soit K un corps. L'anneau $K[[X]]$ des *séries formelles* sur K est

$$K[[X]] = \left\{ \sum_{n\geq 0} a_n X^n : a_n \in K \right\}.$$

Une série formelle $F \in K[[X]]$ est *algébrique* sur le corps des fractions rationnelles $K(X)$ s'il existe un entier $d \geq 1$ et des polynômes $A_0, \ldots, A_d \in K[X]$ non tous nuls tels que

$$A_0 + A_1 F + \cdots + A_d F^d = 0.$$

La *série génératrice* d'une suite $\mathcal{S} = (s_n)_{n\geq 0}$ sur K est la série formelle

$$G_{\mathcal{S}}(X) := \sum_{n\geq 0} s_n X^n.$$

Le théorème suivant caractérise les suites p-automatiques, pour p premier, selon l'algébricité de leur série génératrice.

Théorème I.3.8 (Théorème de Christol, [Chr79]**).** *Soit p un nombre premier. Soit $\mathcal{S}$ une suite sur Σ_p. La suite $\mathcal{S}$ est p-automatique si et seulement si la série génératrice de $\mathcal{S}$ est algébrique sur $\mathbb{F}_p$.*

Considérons à nouveau l'exemple de la suite de Thue–Morse $\mathcal{T} = (t_n)_{n\geq 0}$. Montrons que $\mathcal{T}$ est 2-automatique avec le théorème de Christol. La série génératrice de $\mathcal{T}$ est la série $G_{\mathcal{T}}(X) = \sum_{n\geq 0} t_n X^n = X + X^2 + X^4 + X^7 + \cdots$. Donc on a

$$\begin{aligned} G_{\mathcal{T}}(X) &= \sum_{n\geq 0} t_n X^n = \sum_{n\geq 0} t_{2n} X^{2n} + \sum_{n\geq 0} t_{2n+1} X^{2n+1}, \\ &= \sum_{n\geq 0} t_n X^{2n} + X \sum_{n\geq 0} (t_n + 1) X^{2n}, \\ &= G_{\mathcal{T}}(X^2) + X G_{\mathcal{T}}(X^2) + X \frac{1}{1 - X^2}. \end{aligned}$$

Par conséquent, on a

$$(1 - X^2) G_{\mathcal{T}}(X) - (1 + X)(1 - X^2) G_{\mathcal{T}}(X^2) - X = 0$$

et comme le corps sous-jacent est de caractéristique 2,

$$(1 + X)^3 G_{\mathcal{T}}(X)^2 + (1 + X)^2 G_{\mathcal{T}}(X) + X = 0. \tag{I.3.2}$$

I.3.3. Mesures de complexité pour les suites automatiques. La construction des suites automatiques ne semble pas être un bon procédé pour créer des suites pseudo-aléatoires. En effet, elles sont déterminées par une simple règle locale, comme un morphisme, et la seule connaissance de cette règle, sans clé ou "graine", suffit pour reconstruire entièrement la suite. Dans ce paragraphe, nous donnons des résultats généraux sur les mesures de complexité, introduites dans le Paragraphe I.2, des suites automatiques.

Complexité en facteurs

La complexité en facteurs des suites automatiques est faible. Elle est bornée par une fonction linéaire.

Théorème I.3.9 ([AS03, Corollary 10.3.2]**).** *Soit $\mathcal{S}$ une suite automatique. Alors*

$$p_{\mathcal{S}}(N) \ll N.$$

En particulier, une suite automatique n'est pas une suite normale. En raison de sa faible complexité en facteurs, une suite automatique est qualifiée de *déterministe* dans le contexte des systèmes dynamiques, voir le Paragraphe III.1.2.

Complexité linéaire

Pour la complexité linéaire des suites automatiques non périodiques, Mérai et Winterhof démontrent le résultat suivant [MW18a].

Théorème I.3.10 ([MW18a]**).** *Soit $\mathcal{S}$ une suite q-automatique non périodique sur $\mathbb{F}_q$, avec $q = p^\alpha$, p premier et $\alpha \geq 1$. Soit $h(x,y) = h_0(x) + h_1(x)y + \cdots + h_d(x)y^d \in \mathbb{F}_q[x,y]$ un polynôme non nul tel que $h(G_{\mathcal{S}}(x), x) = 0$ et h n'a pas de zéros rationnels. Soit $M = \max_{0\leq i\leq d}\{\deg h_i - i\}$. On a*

$$\frac{N-M}{d} \leq L(\mathcal{S}, N) \leq \frac{(d-1)N + M + 1}{d}.$$

Par conséquent, le profil de complexité linéaire d'une suite automatique est élevé car encadré par deux fonctions linéaires : pour une suite automatique $\mathcal{S}$, on a $L(\mathcal{S}, N) \asymp N$.

Complexité d'ordre maximal

Il n'existe pas de résultat général sur la complexité d'ordre maximal d'une suite automatique et une étude au cas par cas semble nécessaire. Dans le Chapitre II, nous présentons des nouveaux résultats sur la complexité d'ordre maximal de certaines suites automatiques et leurs sous-suites polynomiales.

Complexité d'expansion

D'après la caractérisation des suites automatiques en fonction de leur série génératrice et du Théorème I.3.8, la complexité d'expansion $E(\mathcal{S}, N)$ d'une suite automatique $\mathcal{S}$ est bornée.

Corrélation d'ordre 2

La corrélation d'ordre 2 des suites automatiques a été étudiée par Mérai et Winterhof [MW18b].

Théorème I.3.11 ([MW18b]**).** *Soit $\mathcal{S}$ une suite q-automatique. On a*

$$C_2(\mathcal{S}, N) \gg N,$$

où la constante impliquée dépend de q et du nombre d'états de l'automate.

Le tableau suivant rassemble les mesures de complexité connues pour les suites automatiques.

Mesures de complexité	Suite aléatoire	Suite automatique
Complexité en facteurs $p_{\mathcal{S}}(N)$	q^N	$\ll N$
Normalité	Oui	Non
Profil de complexité linéaire $L(\mathcal{S}, N)$	$N/2$	$\asymp N$
Profil de complexité d'ordre maximal $M(\mathcal{S}, N)$	$\log(N)$	-
Complexité d'expansion $E(\mathcal{S}, N)$	$2\sqrt{N}$	$\leq C(\mathcal{S})$, constant
Corrélation d'ordre 2, $C_2(\mathcal{S}, N)$	$\sqrt{N \log N}$	$\gg N$

TABLEAU I.2. Mesures de complexité pour une suite automatique sur Σ_q.

En résumé, la plupart des mesures de complexité font défaut pour une suite automatique. Par conséquent une suite automatique ne peut pas être qualifiée de suite pseudo-aléatoire. Cela corrobore le fait que les règles locales qui engendrent une suite automatique ne peuvent pas créer un aléa de bonne qualité.

CHAPITRE II

Sous-suites polynomiales de la suite de Thue–Morse

Pour $q \geq 2$, on note $(n)_q$ sa représentation en base q d'un entier $n \geq 0$ et $s_q(n)$ la somme des chiffres en base q de n. Nous notons $s_{2,k}(n)$ le nombre d'occurrences du mot 1^k dans la représentation binaire de n, et $p_k(n) = s_{2,k}(n) \bmod 2$. Rappelons également que la suite de Thue–Morse $\mathcal{T} = (t_n)_{n\geq 0}$ satisfait $t_n = s_2(n) \bmod 2$ pour tout $n \geq 0$.

Ce chapitre est structuré comme suit : après avoir dressé un état de l'art sur les mesures de complexité de la suite de Thue–Morse, nous nous intéressons aux sous-suites polynomiales de $\mathcal{T}$ et reproduisons les résultats [Pop20].

II.1. Mesures de complexité pour la suite de Thue–Morse

Complexité en facteurs

D'après le Théorème I.3.9, la complexité en facteurs de la suite de Thue–Morse est au plus linéaire. Plus précisément, une formule explicite de la complexité en facteurs de la suite de Thue–Morse est connue depuis 1989, trouvée indépendamment par Brlek [Brl89] et de Luca et Varricchio [dLV89].

Théorème II.1.1 ([dLV89])**.** *Pour $N \geq 3$, la complexité en facteurs $p_{\mathcal{T}}(N)$ de la suite de Thue–Morse est :*

$$p_{\mathcal{T}}(N) = \begin{cases} 6 \cdot 2^{r-1} + 4p, & si\ 0 < p \leq 2^{r-1}, \\ 8 \cdot 2^{r-1} + 2p, & si\ 2^{r-1} < p \leq 2^r, \end{cases}$$

où r, p sont définis de manière unique tels que $N = 2^r + p + 1$, $0 < p \leq 2^r$.

Les auteurs de [dLV89] en déduisent que pour tout $N \geq 3$,

$$3N \leq p_{\mathcal{T}}(N+1) \leq \frac{10}{3}N.$$

Notons que ces bornes sont optimales car elles sont atteintes respectivement pour les valeurs de $N = 2^r + 1$ et $N = 3 \cdot 2^{r-1} + 1$.

Complexité linéaire

Le profil de complexité linéaire de la suite de Thue–Morse a été entièrement déterminé par Mérai et Winterhof [MW18a].

Théorème II.1.2 ([MW18a])**.** *Le profil de complexité linéaire $(L(\mathcal{T}, N))_{N\geq 1}$ de la suite de Thue–Morse est :*

$$L(\mathcal{T}, N) = 2\left\lfloor \frac{N+2}{4} \right\rfloor, \quad N \geq 1.$$

Du Théorème II.1.2, nous déduisons que $L(\mathcal{T}, N) = N/2 + O(1)$. Aussi, le profil de complexité linéaire ne satisfait pas les écarts de taille logarithmique par rapport à $N/2$ du Théorème I.2.6. Autrement dit, même si la complexité linéaire est relativement grande, ce critère disqualifie Thue–Morse comme candidat d'une suite pseudo-aléatoire.

Complexité d'ordre maximal

Le profil de complexité d'ordre maximal de la suite de Thue–Morse a également été totalement déterminé par Sun et Winterhof [SW19b].

Théorème II.1.3 ([SW19b]**).** *Pour $N \geq 4$, le profil de complexité d'ordre maximal $(M(\mathcal{T}, N))_{N\geq 1}$ de la suite de Thue–Morse satisfait :*

$$M(\mathcal{T}, N) = 2^{\ell} + 1,$$

où $\ell = \left\lceil \frac{\log(N/5)}{\log 2} \right\rceil$.

Les auteurs de [SW19b] en déduisent que pour tout $N \geq 4$,

$$\frac{N}{5} + 1 \leq M(\mathcal{T}, N) \leq 2\,\frac{N-1}{5} + 1.$$

Notons que la borne inférieure est optimale car elle est atteinte pour les valeurs $N = 5\cdot 2^r$, pour tout $r \geq 2$. Par conséquent, la complexité d'ordre maximal de la suite de Thue–Morse est grande, bien plus que celle attendue pour une suite prise au hasard qui est de l'ordre de $\log N$. D'après le Paragraphe I.2.8, ce n'est pas une caractéristique souhaitable pour une suite pseudo-aléatoire.

Complexité d'expansion

L'égalité (I.3.2), page 62, permet de déterminer une borne de $E(\mathcal{T}, N)$. En effet, pour tout $N \geq 1$,

$$E(\mathcal{T}, N) \leq 5.$$

La complexité d'expansion de la suite de Thue–Morse est donc très faible et trop éloignée de ce qui est attendu pour une suite aléatoire.

Corrélation d'ordre 2

La corrélation d'ordre k de la suite de Thue–Morse est étudiée depuis le début du vingtième siècle. Dans un premier temps, Mahler [Mah27] montre que pour tout entier k, la suite $(c_N(k))_{N\geq 0}$ définie par

$$c_N(k) = \frac{1}{N}\sum_{n\leq N} (-1)^{s_2(n)+s_2(n+k)}$$

converge vers un nombre réel $c(k)$. Plus précisément, Mauduit [Mau01] montre que

$$\begin{cases} c(0) = 1, \\ c(2k) = c(k), \\ c(2k+1) = -\frac{1}{2}c(k) - \frac{1}{2}c(k+1). \end{cases}$$

Le tableau suivant liste les premières valeurs de $c(k)$.

k	0	1	2	3	4	5	6	7	8	9	10	$\cdots$
$c(k)$	1	$-\frac{1}{3}$	$-\frac{1}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$	0	$\frac{1}{3}$	0	$-\frac{1}{3}$	$\frac{1}{6}$	0	$\cdots$

Nous appelons la suite $\mathcal{T}' = (t'_n)_{n\geq 0}$, *la suite de Thue–Morse sur l'alphabet* $\{-1, 1\}$, où t'_n est obtenu à l'aide de la transformation suivante $t'_n = (-1)^{t_n} = (-1)^{s_2(n)}$. Mauduit et Sárközy [MS98] démontrent le théorème suivant.

Théorème II.1.4 ([MS98]**).** *La corrélation d'ordre* 2 *de la suite de Thue–Morse sur l'alphabet* $\{-1, 1\}$ *satisfait*

$$C_2(\mathcal{T}', N) \geq \frac{1}{12}N, \quad N \geq 5.$$

Notre théorème étudie les corrélations d'ordre 2^ℓ pour $\ell > 1$, en se basant sur les éléments de la preuve du Théorème II.1.4.

Théorème II.1.5. *Soit* $\ell > 1$. *La corrélation d'ordre* 2^ℓ *de la suite de Thue–Morse satisfait*

$$C_{2^\ell}(\mathcal{T}', N) \gg N, \quad N \gg 1.$$

Preuve. Nous nous inspirons de la méthode de preuve de Mauduit et Sárközy de [MS98]. Soit M tel que $2^M < N \leq 2^{M+1}$ et posons $D = (0, 1, \ldots, 2^\ell - 1)$. Alors on a

$$\begin{aligned} V(\mathcal{T}'_N, 2^M, D) &= \sum_{n<2^M} t'_n \cdots t'_{n+2^\ell-1} \\ &= \sum_{n<2^{M-1}} t'_{2n} \cdots t'_{2n+2^\ell-1} + \sum_{n<2^{M-1}} t'_{2n+1} \cdots t'_{2n+2^\ell} \end{aligned}$$

Comme $t'_{2n}t'_{2n+1} = -(t'_n)^2 = -1$, en regroupant les termes 2 par 2 dans les deux sommes précédentes, on a

$$V(\mathcal{T}'_N, 2^M, D) = (-1)^{2^{\ell-1}}2^{M-1} + (-1)^{2^{\ell-1}-1} \sum_{n<2^{M-1}} t'_n t'_{n+2^{\ell-1}},$$

Alors

$$V(\mathcal{T}'_N, 2^M, D) = 2^{M-1} - \gamma_{M-1}(2^{\ell-1}),$$

où $\gamma_M(d) = \sum_{n<2^M} t'_n t'_{n+d}$ pour tout $d \geq 1$. De plus, $\gamma_M(1) = -\frac{1}{3}2^M - \frac{2}{3}(-1)^M$, voir [MS98, Theorem 2], où les notations sont différentes. On obtient la formule de récurrence suivante

$$\begin{cases} \gamma_M(2d) = 2\gamma_{M-1}(d), \\ \gamma_M(2d+1) = -\gamma_{M-1}(d) - \gamma_{M-1}(d+1), \end{cases}$$

car

$$\begin{aligned}\gamma_M(2d) &= \sum_{n<2^M} t'_n t'_{n+2d} \\ &= \sum_{n<2^{M-1}} t'_{2n} t'_{2n+2d} + \sum_{n<2^{M-1}} t'_{2n+1} t'_{2n+2d+1} \\ &= \sum_{n<2^{M-1}} t'_n t'_{n+d} + \sum_{n<2^{M-1}} (-t'_n)(-t'_{n+d}) \\ &= \gamma_{M-1}(d) + \gamma_{M-1}(d) = 2\gamma_{M-1}(d),\end{aligned}$$

et

$$\begin{aligned}\gamma_M(2d+1) &= \sum_{n<2^M} t'_n t'_{n+2d+1} \\ &= \sum_{n<2^{M-1}} t'_{2n} t'_{2n+2d+1} + \sum_{n<2^{M-1}} t'_{2n+1} t'_{2n+2d+2} \\ &= \sum_{n<2^{M-1}} t'_n (-t'_{n+d}) + \sum_{n<2^{M-1}} (-t'_n) t'_{n+d+1} \\ &= -\gamma_{M-1}(d) - \gamma_{M-1}(d+1).\end{aligned}$$

Si $M \geq \ell$, on en déduit

$$\begin{aligned}\gamma_M(2^\ell) &= 2\gamma_{M-1}(2^{\ell-1}) = \cdots = 2^\ell \gamma_{M-\ell}(1) \\ &= 2^\ell \left(-\frac{1}{3} 2^{M-\ell} - \frac{2}{3}(-1)^{M-\ell} \right) \\ &= -\frac{1}{3} 2^M - \frac{2}{3}(-1)^M (-2)^\ell.\end{aligned}$$

Donc

$$\begin{aligned}V(\mathcal{T}'_N, 2^M, D) &= 2^{M-1} + \frac{1}{3} 2^{M-1} + \frac{2}{3}(-1)^{M-1}(-2)^{\ell-1} \\ &= \frac{4}{3} 2^M - \frac{2}{3}(-1)^M (-2)^{\ell-1}.\end{aligned}$$

Alors

$$C_2(\mathcal{T}', N) \geq \left| V(\mathcal{T}'_N, 2^M, D) \right| \gg 2^M \gg N,$$

pour N assez grand. □

L'ensemble de ces résultats est résumé dans le Tableau II.1, page 70.

II.2. Sous-suites polynomiales

Pour les sous-suites polynomiales de la suite de Thue–Morse, cette étude rentre dans le cadre du problème de Gelfond [Gel68]. Dans tout ce qui suit, pour un polynôme $P \in \mathbb{Z}[X]$ de degré $d \geq 2$ tel que $P(\mathbb{N}) \subset \mathbb{N}$, on note la sous-suite de Thue–Morse le long de ce polynôme $\mathcal{T}_P = (t(P(n)))_{n\geq 0}$. Pour le cas particulier $P(X) = X^2$, on note $\mathcal{T}_2 = \mathcal{T}_P$.

Problème II.2.1 (Problème de Gelfond). Soient $P \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$, $a \in \mathbb{Z}$, $m \geq 1$ et $x \geq 1$. Déterminer le nombre d'entiers $n \leq x$ tel que

$$s_q(P(n)) \equiv a \pmod{m}.$$

Dartyge et Tenenbaum [DT05, DT06] montrent qu'il existe une proportion strictement positive de tels entiers.

Théorème II.2.1 ([DT05, DT06])**.** *Soient q, m deux entiers tels que $q \geq 2$ et $\mathrm{pgcd}(m, q-1) = 1$. Soit $P \in \mathbb{Z}[X]$ de degré d tel que $P(\mathbb{N}) \subset \mathbb{N}$. Il existe deux constantes $C = C(P, q, m)$ et $N_0 = N_0(P, q, m)$ telles que pour tout $a \in \{0, 1, \ldots, m-1\}$ et pour tout entier $N \geq N_0$ on a*

$$\mathrm{Card}\{n < N : s_q(P(n)) \equiv a \pmod{m}\} \geq CN^{\alpha},$$

avec $\alpha = \min(1, 2/d!)$.

Stoll [Sto12] améliore la borne du Théorème II.2.1 avec $\alpha = \frac{4}{3d+1}$ pour $d \geq 3$. Mauduit et Rivat [MR09] montrent une équirépartition dans le problème de Gelfond pour $P(X) = X^2$. En particulier, la suite $\mathcal{T}_2$ est simplement normale. Notons que Drmota Mauduit et Rivat étendent le résultat de [MR09] à des polynômes d'un degré quelconque pour q assez grand.

Complexité en facteurs

Moshe [Mos07] fournit une borne inférieure de la complexité en facteurs des sous-suites polynomiales de la suite de Thue–Morse, répondant à une question posée par Allouche et Shallit [AS03, Open problem 10.7]. Plus précisément, ce résultat donne une complexité en facteurs maximale dans le cas d'un polynôme quadratique, et une complexité en facteurs exponentielle pour les polynômes de degré ≥ 3. Récemment, Shen [She22] montre que la complexité en facteurs des sous-suites polynomiales de la suite de Thue–Morse est maximale pour tout polynôme de degré ≥ 2.

Après leur complexité en facteurs, il est naturel de s'intéresser à la normalité des sous-suites polynomiales de la suite de Thue–Morse. Drmota, Mauduit et Rivat [DMR19] montrent que la suite $\mathcal{T}_2$ est une suite normale sur $\{0, 1\}$. Leur démonstration utilise les sommes d'exponentielles et nous les utiliserons plus tard également, voir le Chapitre VII.

Tous ces résultats indiquent un changement radical de comportement de la complexité en facteurs de la suite de Thue–Morse le long de sous-suites polynomiales.

Complexité d'ordre maximal

Le théorème suivant donne une borne inférieure pour la complexité d'ordre maximal pour la suite $\mathcal{T}_2$.

Théorème II.2.2 ([SW19a, Theorem 1])**.** *Soit $\mathcal{T}_2 = (t(n^2))_n$ la sous-suite de Thue–Morse le long des carrés. La complexité d'ordre maximal au rang N de $\mathcal{T}_2$ satisfait*

$$M(\mathcal{T}_2, N) \geq \sqrt{\frac{2N}{5}}, \qquad N \geq 21.$$

La complexité d'ordre maximal de la suite de Thue–Morse le long des carrés reste relativement grande en comparaison à l'ordre de grandeur attendu pour une suite aléatoire.

Complexité d'expansion

La suite $\mathcal{T}_P$ n'est en général pas une suite automatique, voir le Théorème I.3.7. Cela entraîne donc que la complexité d'expansion de $\mathcal{T}_P$ est non bornée (voir Théorème I.3.8). Autrement dit,

$$E(\mathcal{T}_P, N) \to +\infty, \qquad N \to +\infty.$$

Notons qu'à ce jour aucun résultat n'est connu sur la vitesse de croissance de $E(\mathcal{T}_2, N)$ et les estimations semblent indiquer que la vitesse satisfait l'ordre de grandeur attendu, voir [MW22], à savoir $\sqrt{N}$.

Corrélation d'ordre k

Drmota, Mauduit et Rivat [DMR19] démontrent le résultat suivant.

Théorème II.2.3 ([DMR19]**).** *Pour tout entier $k \geq 1$ et $(\alpha_0, \ldots, \alpha_{k-1}) \in \{0,1\}^k$ tels que $(\alpha_0, \ldots, \alpha_{k-1}) \neq (0, \ldots, 0)$, il existe $\eta > 0$ tel que*

$$\sum_{n<N} (-1)^{\alpha_0 t(n^2) + \alpha_1 t((n+1)^2) + \cdots + \alpha_{k-1} t((n+k-1)^2)} \ll N^{1-\eta}.$$

Autrement dit, pour $D = (0, 1, \ldots, k-1)$, on a $\frac{1}{N} V((\mathcal{T}_2)_N, M, D) \to 0$ pour $N \to +\infty$. Cependant, le théorème précédent ne donne pas de majorations pour la corrélation d'ordre k de la suite $\mathcal{T}_2$, car il décrit seulement les corrélations pour des valeurs consécutives. Les estimations coïncident avec l'ordre moyen attendu, voir [MW22].

Nous résumons les différences des mesures de complexité entre la suite de Thue–Morse $\mathcal{T}$ et sa sous-suite le long des carrés $\mathcal{T}_P$ dans le Tableau II.1 :

Mesures de complexité	$\mathcal{S} = \mathcal{T}$	$\mathcal{S} = \mathcal{T}_2$
Complexité en facteurs $p_{\mathcal{S}}(N)$	Linéaire	2^N
Normalité	Non	Oui
Profil de complexité linéaire $L(\mathcal{S}, N)$	$\sim N/2$	$\gg N^{1/2}$
Profil de complexité d'ordre maximal $M(\mathcal{S}, N)$	$\asymp N$	$\gg N^{1/2}$
Complexité d'expansion $E(\mathcal{S}, N)$	≤ 5	Non bornée
Corrélation d'ordre k, $C_k(\mathcal{S}, N)$	$\gg N$ si $k = 2^\ell$	$o(N)$ si $d_i = i$

Tableau II.1. Mesures de complexité de $\mathcal{T}$ et $\mathcal{T}_2$.

Certains des résultats précédents ont été étendus à la suite de Rudin–Shapiro ou aux suites de motifs d'ordre k. En particulier, la suite de Rudin–Shapiro est normale le long des carrés, [MR18, Mül18]. Sun et Winterhof [SW19a] ont également étudié les suites de motifs d'ordre k le long des carrés.

Théorème II.2.4 ([SW19a, Theorem 2])**.** *Soient $k \geq 2$ et $\mathcal{P}_{k,2} = (p_k(n^2))_n$ la sous-suite de $\mathcal{P}_k$ le long des carrés. La complexité d'ordre maximal au rang N de $\mathcal{P}'_k$ satisfait*

$$M(\mathcal{P}_{k,2}, N) \geq \sqrt{\frac{N}{8}}, \qquad N \geq 2^{2k+2}.$$

Sun et Winterhof [SW19a, Problem 4] ont posé le problème suivant : étendre les Théorèmes II.2.2 et II.2.4 à toute sous-suite polynomiale. L'article [Pop20] résout ce problème pour un polynôme unitaire.

II.3. Résultats

Dans les paragraphes suivants, nous reproduisons essentiellement l'article [Pop20], publié dans le journal *Uniform Distribution Theory (UDT)*.

Dans tout ce qui suit, pour un polynôme $P \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$, on note $\mathcal{T}_P$ la sous-suite de Thue–Morse le long de ce polynôme $\mathcal{T}_P = (t(P(n)))_{n\geq 0}$ et $\mathcal{P}_{k,P} = (p_k(P(n)))_{n\geq 0}$ la sous-suite de la suite de motifs d'ordre k. Les résultats principaux de l'article [Pop20] sont les deux théorèmes suivants.

Théorème II.3.1 ([Pop20])**.** *Soit $d \geq 2$ et $P(X) \in \mathbb{Z}[X]$ un polynôme unitaire de degré d tel que $P(\mathbb{N}) \subset \mathbb{N}$. La complexité d'ordre maximal au rang N de $\mathcal{T}_P$ satisfait*

$$M(\mathcal{T}_P, N) \gg N^{1/d},$$

où la constante implicite dépend uniquement du polynôme P.

Théorème II.3.2 ([Pop20])**.** *Soit $d \geq 2$ et $P(X) \in \mathbb{Z}[X]$ un polynôme unitaire de degré d tel que $P(\mathbb{N}) \subset \mathbb{N}$. La complexité d'ordre maximal au rang N de $\mathcal{P}_{k,P}$ satisfait*

$$M(\mathcal{P}_{k,P}, N) \gg N^{1/d},$$

où la constante implicite dépend uniquement du polynôme P et de k.

II.4. Preuve du Théorème II.3.1

La fonction somme des chiffres est centrale pour la démonstration du Théorème II.3.1. Une plus ample introduction de la fonction somme des chiffres et ses propriétés sera donnée dans le Chapitre VI. L'identité suivante sur la fonction somme des chiffres est à la fois évidente et cruciale pour cette thèse. Pour $a, b, r \geq 0$ des entiers tels que $0 \leq b < 2^r$, on a

$$s_2(a2^r + b) = s_2(a) + s_2(b). \tag{II.4.1}$$

En effet,

$$\begin{array}{cccc} & (a)_2 & \mathtt{0}^\lambda & \mathtt{0}\cdots\mathtt{0} & = a2^r \\ + & & \mathtt{0}^\lambda & (b)_2 & = b \\ \hline & (a)_2 & \mathtt{0}^\lambda & (b)_2 & = a2^r + b. \end{array}$$

pour un certain $\lambda \geq 0$. Pour de tels a, b, r, on dit que la somme $a + 2^r b$ est *non-interférente* : les chiffres de b ne peuvent pas interférer avec ceux de a car les chiffres de a sont décalés vers la gauche d'un écart suffisamment grand dans le schéma d'addition. En effet, en base q, multiplier un entier n par q^r revient à décaler les chiffres de n vers la gauche de r pas et de compléter par des 0. La preuve du Théorème II.3.1 est basée à la fois sur les sommes non-interférentes et sur une étude précise des propagations de retenues.

Le lemme suivant étudie les facteurs spéciaux à droite de la suite $\mathcal{T}_P$.

Lemme II.4.1. *Soient $d \geq 2$ et $P(X) \in \mathbb{Z}[X]$ avec $P(X) = X^d + \alpha_{d-1}X^{d-1} + \cdots + \alpha_1 X + \alpha_0$ tel que $P(\mathbb{N}) \subset \mathbb{N}$, $\alpha_i \geq 0$ pour $0 \leq i \leq d-1$ et $\alpha_{\max} = \max(\alpha_i)$.*

Alors il existe un entier positif $\ell_0(P)$ tel que pour tout $\ell > \ell_0(P)$:

(1) pour tout $1 \leq n < \frac{1}{2(2\alpha_{\max})^{1/d}}\, 2^\ell$ et pour tout $r \geq 1$, on a

$$t(P(n + 2^{d\ell})) = t(P(n + 2^{d\ell+r})).$$

(2) il existe deux entiers positifs y et r, ne dépendant que de P, tels que

$$t(P(1 + y2^\ell + 2^{d\ell})) \neq t(P(1 + y2^\ell + 2^{d\ell+r})).$$

Preuve. Posons $\alpha_d = 1$. On a

$$\begin{aligned} P(n + 2^{d\ell}) &= \sum_{0\leq j\leq d} \alpha_j (n + 2^{d\ell})^j, \\ &= \sum_{0\leq i\leq d} \left(\sum_{i\leq j\leq d} \binom{j}{i} \alpha_j n^{j-i} \right) 2^{id\ell}, \\ &= \sum_{0\leq i\leq d} \beta_i 2^{id\ell}, \end{aligned} \tag{II.4.2}$$

où $\beta_i = \sum_{i\leq j\leq d} \binom{j}{i} \alpha_j n^{j-i}$. Pour tout $0 \leq i \leq d$, on a

$$\beta_i \leq \alpha_{\max}\, n^{d-i} \sum_{i\leq j\leq d} \binom{j}{i} \leq \alpha_{\max}\, n^d \binom{d+1}{i+1} \leq \alpha_{\max}\, n^d 2^{d+1}.$$

Soit $\ell_0(P) \in \mathbb{N}$ tel que pour tout $\ell > \ell_0(P)$ et pour tout $n \geq 0$ vérifiant

$$1 \leq n < \frac{1}{2(2\alpha_{\max})^{1/d}}\, 2^\ell, \quad 0 \leq i \leq d, \tag{II.4.3}$$

on a $\beta_i < 2^{d\ell}$, $0 \leq i \leq d$. Ainsi, pour $\ell > \ell_0(P)$ et pour tout $n \geq 0$ tel que (II.4.3) est vérifiée, la somme dans (II.4.2) est non-interférente. Alors on a pour tout $r \geq 1$,

$$t(P(n + 2^{d\ell})) = \sum_{0\leq i\leq d} t(\beta_i) = t(P(n + 2^{d\ell+r})),$$

ce qui démontre la première partie.

Pour la seconde partie, on décompose la somme comme précédemment,

$$P(1+y2^{\ell}+2^{d\ell}) = \sum_{0\le i\le d}\left(\sum_{i\le j\le d}\binom{j}{i}\alpha_j(1+y2^{\ell})^{j-i}\right)2^{id\ell}. \tag{II.4.4}$$

On factorise les termes par puissance de 2 et on regarde les éventuelles interférences. Le terme général est de la forme $2^{id\ell+l(j-i)}$, pour des entiers i,j tels que $0\le i\le j\le d$, où les coefficients dépendent uniquement du polynôme P et de y. On représente les termes généraux dans le tableau suivant :

i \ j	0	1	2	$\cdots$	d
0	2^0	2^{ℓ}	$2^{2\ell}$		$2^{d\ell}$
1		$2^{d\ell}$	$2^{d\ell+l}$		$2^{d\ell+(d-1)\ell}$
2			$2^{2d\ell}$		$2^{2d\ell+(d-2)\ell}$
$\vdots$				$\ddots$	
d					$2^{d\cdot d\ell}$

La seule interférence possible, pour $\ell > \ell_1(P)$, est entre le couple $(i,j)=(0,d)$ et le couple $(i,j)=(1,1)$, qui correspondent tous les deux au terme général $2^{d\ell}$. Aucun coefficient devant $2^{id\ell+(j-i)\ell}$, pour deux entiers i,j, dépend de ℓ car y a vocation d'être choisi plus tard pour ne dépendre que de P. De plus, l'écart entre deux termes généraux distincts dans (II.4.4) est au moins de 2^{ℓ}. Le terme d'interférence est le coefficient de $2^{d\ell}$ dans l'écriture suivante,

$$\begin{aligned}&(\alpha_0+\alpha_1(1+y2^{\ell})+\cdots+(1+y2^{\ell})^d)\,2^{0\cdot d\ell}\\&+(\alpha_1+2\alpha_2(1+y2^{\ell})+\cdots+d(1+y2^{\ell})^{d-1})\,2^{1\cdot d\ell},\end{aligned}$$

car tous les autres termes généraux sont au moins de taille $2^{2d\ell}$. Ainsi, le terme d'interférence est égal à $y^d+\sum_{1\le i\le d} i\alpha_i$.
D'autre part, par un calcul similaire, on a

$$P(1+y2^{\ell}+2^{d\ell+r}) = \sum_{0\le i\le d}\left(\sum_{i\le j\le d}\binom{j}{i}\alpha_j(1+y2^{\ell})^{j-i}\right)2^{i(d\ell+r)}.$$

On factorise les termes par puissance de 2 comme précédemment. Le terme général est cette fois-ci de la forme $2^{i(d\ell+r)+(j-i)\ell}$, pour des entiers i,j, et on a le tableau suivant :

i \ j	0	1	2	$\cdots$	d
0	2^0	2^ℓ	$2^{2\ell}$		$2^{d\ell}$
1		$2^{d\ell+r}$	$2^{d\ell+l}2^r$		$2^{d\ell+(d-1)\ell}2^r$
2			$2^{2d\ell}2^{2r}$		$2^{2d\ell+(d-2)\ell}2^{2r}$
$\vdots$				$\ddots$	
d					$2^{d\cdot d\ell}2^{dr}$

De même, la seule interférence possible est pour le couple $(i,j)=(0,d)$ et le couple $(i,j)=(1,1)$ pour $\ell>\ell_2(P)$. Ici, le terme d'interférence est $y^d+2^r\sum_{1\le i\le d} i\alpha_i$. Tous les coefficients dans le second tableau sont identiques à ceux du premier tableau à un facteur multiplicatif près (une puissance de 2 même). Comme $t(2^\mu n)=t(n)$ pour tout $\mu\ge 0$, les contributions qui proviennent de ces termes non-interférents sont les mêmes que dans le cas précédent.

Posons $z=\sum_{1\le i\le d} i\alpha_i$, et remarquons que cet entier z vérifie $z\ge 2$ et ne dépend que de P. Pour résumer, pour $\ell>\ell_3(P)$, on a

$$\begin{aligned} t(P(1+y2^\ell+2^{d\ell}))+t(P(1+y2^\ell+2^{d\ell+r})) \\ \equiv t(y^d+z)+t(y^d+2^r z) \pmod 2. \end{aligned} \tag{II.4.5}$$

Notre objectif final est de garantir l'existence de deux entiers r et y, qui ne dépendent que de P, tel que le côté droit de (II.4.5) est 1 (mod 2).

Soit $\lambda\ge 1$ l'unique entier tel que $2^\lambda\le z<2^{\lambda+1}$, c'est-à-dire le bit de poids fort de z est en position λ. On pose $y=2^\lambda$, et on a $z<y^d=2^{\lambda d}$. Donc on a

$$t(y^d+z)=t(2^{\lambda d}+z)\equiv 1+t(z) \pmod 2.$$

On réalise l'addition suivante, où le $\mathtt{1}$ est en position λd,

$$\begin{array}{rcccl} & \mathtt{1} & \mathtt{0}\cdots\mathtt{0} & \mathtt{0}\cdots\mathtt{0} & =y^d \\ + & & & (z)_2 & =z \\ \hline & \mathtt{1} & \mathtt{0}\cdots\mathtt{0} & (z)_2 & =y^d+z. \end{array}$$

Soit $r=\lambda d-\lambda\ge 1$, on a alors

$$t(y^d+2^r z)=t(2^{\lambda d}+2^{\lambda d-\lambda}z)=t(2^\lambda+z)=t(z)$$

car le bit de poids fort de z est en position λ. On réalise une somme similaire à la somme y^d+z, mais on décale z suffisamment vers la gauche pour qu'uniquement son bit de poids fort soit impacté par la somme avec $2^{\lambda d}$. En écrivant $(z)_2=\mathtt{1}(z')_2$ pour un entier z', on réalise la somme suivante

$$\begin{array}{rccl} & \mathtt{1} & \mathtt{0}\cdots\mathtt{0} & =y^d \\ + & \mathtt{1} & (z')_2 & =2^r z \\ \hline \mathtt{1} & \mathtt{0} & (z')_2 & =y^d+2^r z. \end{array}$$

On obtient donc

$$t(P(1+y2^{\ell}+2^{d\ell})) + t(P(1+y2^{\ell}+2^{d\ell+r})) \equiv 1 + 2\,t(z) \equiv 1 \pmod 2$$

et la seconde partie du lemme est prouvée. □

Remarque II.4.1. Pour un coefficient dominant α_d quelconque, en répétant le même schéma de preuve que le lemme précédent, il faut trouver un couple (y, r) tel que

$$t(\alpha_d y^d + z) + t(\alpha_d y^d + 2^r z) \equiv 1 \pmod 2.$$

Pour le même couple (y, r), le calcul n'aboutit pas à la même conclusion du lemme pour $\alpha_d = 2^k - 1$ avec k pair. En effet dans la propagation de retenues, un nombre pair de `1` est supprimé et la parité du nombre de `1` dans la somme est donc inchangée. C'est pourquoi, des nouveaux arguments semblent nécessaires pour ce cas, en fonction de la structure binaire de z et α_d.

Nous pouvons désormais démontrer le Théorème II.3.1.

Preuve du Théorème II.3.1. Sans pertes de généralité, on peut supposer que les coefficients de P sont tous des entiers positifs. En effet, posons $\alpha_d = 1$ et supposons que $\alpha_i \geq 0$ pour tout $0 \leq i < d-1$. Ainsi, pour des entiers positifs n, a, on a $P(n+a) = \sum_{0\leq i\leq d} \beta_i n^i$ avec $\beta_i = \sum_{i\leq j\leq d} \binom{j}{i} \alpha_j a^{j-i}$ et $\alpha_d = 1$. Si a est suffisamment grand, alors on a

$$\beta_i = a^{d-i}\left(\binom{d}{i}\alpha_d + \sum_{i\leq j<d}\binom{j}{i}\alpha_j a^{j-d}\right) \gg_P a^{d-i},$$

et ainsi on a $\beta_i \geq 0$ pour tout $i \geq 0$. Cette translation par un entier positif a qui ne dépend que de P ne modifie pas la complexité d'ordre maximal au rang N pour N suffisamment grand.

Soient $\alpha_{\max}, z, \lambda, y, r$ tels que

$$\alpha_{\max} = \max(\alpha_i), \quad z = \sum_{1\leq i\leq d} i\alpha_i, \quad 2^{\lambda} \leq z < 2^{\lambda+1}, \quad r = \lambda(d-1),$$

les quantités définies dans la preuve du Lemme II.4.1. Notons que toutes ces quantités dépendent uniquement de P et pas de ℓ. Soit $N > N_0(P)$ et $M(\mathcal{T}_P, N) = M$. Soit $\ell \geq 2$ l'entier défini par

$$1 + y2^{\ell} + 2^{d\ell+r} < N \leq 1 + y2^{l+1} + 2^{d(l+1)+r}.$$

Supposons que

$$M < \frac{1}{2(2\alpha_{\max})^{1/d}}\,2^{\ell}, \tag{II.4.6}$$

et soit $f(x_1, \ldots, x_M)$ un polynôme en M indéterminées tel que

$$t(P(j+M)) = f(t(P(j)), \ldots, t(P(j+M-1))), \quad j = 0, 1 \ldots, N-M-1. \tag{II.4.7}$$

L'existence de f est assurée par définition de la complexité d'ordre maximal. Notons que pour $0 \leq k \leq N-M-1$, les valeurs de $t(P(k+M)), \ldots, t(P(N-1))$ sont

uniquement déterminées par les valeurs $t(P(k)),\dots,t(P(k+M-1))$ en appliquant (II.4.7) successivement pour $j=k,\dots,N-M-1$. En particulier, si

$$(t(P(k_1)),\dots,t(P(k_1+M-1))) = (t(P(k_2)),\dots,t(P(k_2+M-1))) \tag{II.4.8}$$

pour k_1 et k_2 avec $0\le k_1<k_2\le N-M-1$, alors

$$(t(P(k_1+M)),\dots,t(P(k_1+N-k_2-1))) = (t(P(k_2+M)),\dots,t(P(N-1))).$$

Prenons $k_1=2^{d\ell}$ et $k_2=2^{d\ell+r}$. Alors d'après le Lemme II.4.1, (k_1,k_2) satisfait (II.4.8):

$$(t(P(2^{d\ell}+M)),\dots,t(P(N+2^{d\ell}(1-2^r)-1))) = (t(P(2^{d\ell+r}+M)),\dots,t(P(N-1))).$$

Comme $N-1\ge 1+y2^\ell+2^{d\ell+r}$ et $M\le 1+y2^\ell$, alors

$$t(P(1+y2^\ell+2^{d\ell})) = t(P(1+y2^\ell+2^{d\ell+r})),$$

ce qui contredit la seconde partie du Lemme II.4.1 et on a

$$M \ge \frac{1}{2(2\alpha_{\max})^{1/d}}\,2^\ell \gg_P N^{1/d}. \tag{II.4.9}$$

Ainsi le Théorème II.3.1 est prouvé. □

II.5. Preuve du Théorème II.3.2

L'identité (II.4.1), page 71, n'est pas vérifiée en règle générale pour $s_{2,k}$ en lieu et place de s_2. Par exemple, $s_{2,2}(4+2)=s_{2,2}(\mathtt{110})=1$ alors que $s_{2,2}(4)+s_{2,2}(2)=s_{2,2}(\mathtt{100})+s_{2,2}(\mathtt{10})=0$. Cependant, une identité similaire peut être déduite en remplaçant r par $r+1$. Soient a,b des entiers positifs et $0\le b<2^r$, alors on a pour tout $k\ge 2$,

$$s_{2,k}(a2^{r+1}+b) = s_{2,k}(a)+s_{2,k}(b). \tag{II.5.1}$$

Le lemme suivant est l'analogue du Lemme II.4.1. La méthode utilisée précédemment ne convient pas pour le cas des suites de motifs. Nous adaptons la méthode utilisée dans les articles suivants [HLS11a, MS14, Sto12, Sto16].

Lemme II.5.1. *Soient $d\ge 2$, $P(X)\in\mathbb{N}[X]$ avec $P(X)=X^d+\alpha_{d-1}X^{d-1}+\cdots+\alpha_1X+\alpha_0$ tel que $P(\mathbb{N})\subset\mathbb{N}$ et $\alpha_{\max}=\max(\alpha_i)$. Il existe un entier positif $\ell_0(P,k)$ tel que pour tout $\ell>\ell_0(P,k)$ les propriétés suivantes sont vérifiées :*
(1) Pour tout $1\le n<\frac{1}{4(2\alpha_{\max})^{1/d}}\,2^\ell$ et pour tout $s\ge 1$,

$$p_k(P(n+2^{d\ell})) = p_k(P(n+2^{d\ell+s})).$$

(2) Il existe des entiers positifs $y=y(P,k)$ et $s=s(P,k)$ tels que

$$p_k(P(1+y2^\ell+2^{d\ell})) \ne p_k(P(1+y2^\ell+2^{d\ell+s})).$$

Preuve. On procède de la même manière que dans la preuve du Lemme II.4.1 avec (II.4.1) remplacée par (II.5.1). Notons que le changement de r en $r+1$ entre ces deux identités se traduit par une division par 2 dans la constante. On reprend les mêmes notations du Lemme II.4.1. Pour la seconde partie on a $\ell>\ell_0(P)$,

$$\begin{aligned} &p_k(P(1+y2^\ell+2^{d\ell})) + p_k(P(1+y2^\ell+2^{d\ell+s})) \\ &\equiv p_k(y^d+z)+p_k(y^d+2^sz) \pmod 2. \end{aligned}$$

Soit $f_a(x) = ax^3 + ax^2 - x + a$ un polynôme où a est un entier et on a :

$$f_a(x)^d = \sum_{0 \le i \le 3d} \mu_i x^i$$

avec $\mu_i \in \mathbb{Z}$. Si $a > a_0(d)$ est suffisamment grand, on a $\mu_i > 0$ pour $i \neq 1$ et $\mu_1 = -da^{d-1} < 0$, voir [HLS11a]. Il suit que pour $u > u_0(d)$ suffisamment grand, on a

$$(f_a(2^u)^d)_2 = \eta_1 \mathtt{0} \cdots \mathtt{0} \eta_2 \mathtt{0} \cdots \mathtt{0} \eta_{t-1} \mathtt{1} \cdots \mathtt{1} \eta_t \mathtt{0} \cdots \mathtt{0} \eta_{t+1} \tag{II.5.2}$$

avec $t = 3d$ et des certains $\eta_i = \eta_i(a,d)$. Pour $i \in \{1, \ldots, t-2, t+1\}$, on peut choisir les η_i comme étant les écritures en base 2 des μ_i, η_{t-1} tel que son chiffre le plus à gauche est $\mathtt{1}$ et son chiffre le plus à droite est $\mathtt{0}$ et η_t tel que son chiffre le plus à gauche est $\mathtt{0}$ et son chiffre le plus à droite est $\mathtt{1}$. Remarquons que la décomposition de (II.5.2) n'est pas unique en général. Pour $a = 2^\lambda$ avec $\lambda > \lambda_0(z)$ on a

$$p_k(f_a(2^u)^d + z) = p_k(f_a(2^u)^d) + p_k(z). \tag{II.5.3}$$

La longueur du bloc de chiffres $\mathtt{1}$ ne dépend que de u et le passage de u à $u+1$ ajoute exactement un chiffre $\mathtt{1}$ dans ce bloc.

Soit $y = f_a(2^u)$ pour $a = 2^\lambda > \max(a_0, 2^{\lambda_0})$ et $u > u_0(a,d)$. On écrit dans la suite

$$y^d = \omega_1 \mathtt{0} \cdots \mathtt{0} \omega_2 \mathtt{0} \mathtt{1}^\alpha \mathtt{0} \omega_3$$

où $\mathtt{1}^\alpha$ est le bloc de chiffres de $\mathtt{1}$ dans (II.5.2) entre η_{t-1} et η_t. Pour $u > u_1(a,d,z,k)$ suffisamment grand, $\alpha > \max(\lceil \log_2(z) \rceil, k)$. En écrivant $z = \mathtt{1}^i \mathtt{0} \omega'$ ou $z = \mathtt{1}^i$ pour un certain $i \ge 1$ et l'un bloc de chiffres ω' potentiellement vide. On choisit s de telle manière que les i premiers chiffres $\mathtt{1}$ interfèrent avec les i derniers chiffres $\mathtt{1}$ du bloc intérieur composé que de $\mathtt{1}$ dans y^d. Cela est toujours possible car α est plus grand que la longueur de z. On distingue alors les deux cas suivants :

- Si z s'écrit sous la forme $z = \mathtt{1}^{(i)} \mathtt{0} \omega'$, alors on a

$$\begin{array}{cccccccc}
 & \omega_1 \mathtt{0} \cdots \mathtt{0} & w_2 & \overbrace{\mathtt{1} \cdots \mathtt{1}}^{\alpha - i} & \overbrace{\mathtt{1} \cdots \mathtt{1} \mathtt{1}}^{i} & \mathtt{0} \omega_3 & = y^d \\
+ & & & & \mathtt{1} \cdots \mathtt{1} \mathtt{1} & \mathtt{0} \omega' & = 2^s z \\
\hline
 & \omega_1 \mathtt{0} \cdots \mathtt{0} & (w_2 + 1) & \mathtt{0} \cdots \mathtt{0} & \mathtt{1} \cdots \mathtt{1} \mathtt{0} & \mathtt{0}(\omega_3 + \omega') & = y^d + 2^s z
\end{array}$$

- Si z s'écrit sous la forme $z = \mathtt{1}^{(i)}$, alors on a

$$\begin{array}{cccccccc}
 & \omega_1 \mathtt{0} \cdots \mathtt{0} & w_2 & \overbrace{\mathtt{1} \cdots \mathtt{1}}^{\alpha - i} & \overbrace{\mathtt{1} \cdots \mathtt{1} \mathtt{1}}^{i} & \mathtt{0} \omega_3 & = y^d \\
+ & & & & \mathtt{1} \cdots \mathtt{1} \mathtt{1} & \mathtt{0} \cdots \mathtt{0} & = 2^s z \\
\hline
 & \omega_1 \mathtt{0} \cdots \mathtt{0} & (w_2 + 1) & \mathtt{0} \cdots \mathtt{0} & \mathtt{1} \cdots \mathtt{1} \mathtt{0} & \mathtt{0} \omega_3 & = y^d + 2^s z
\end{array}$$

Dans les deux cas, pour $u > u_2(a,d,z,k)$ suffisamment grand, $p_k(y^d + 2^s z)$ ne dépend pas de u car aucun motif de taille k n'est créé ou détruit. En réalité, seulement les longueurs des blocs de $\mathtt{0}$ et le bloc de $\mathtt{1}$ changent avec u dans (II.5.2) mais la somme $y^d + 2^s z$ réduit le bloc de $\mathtt{1}$ de longueur α à un bloc de $\mathtt{1}$ de longueur $(i-1)$, qui ne dépend désormais plus de u. De plus, le nombre de motifs de taille k dans $\omega_1, \omega_2, \omega_3$

et ω' est indépendant du choix de u. Ceci implique que $p_k(y^d + 2^s z)$ est constant pour tout $u > u_2(a, d, z, k)$.

Ce qui importe désormais est le nombre de motifs de taille k que l'on détruit dans y^d en lui ajoutant $2^s z$. Comme $\alpha > k$, le passage de u à $(u+1)$ implique que si l'on ajoute $2^s z$ à y^d, on détruit un motif de taille k en plus dans y^d. Alors on choisit $u > u_2(a, d, z, k)$ de telle manière que la parité du nombre de motifs de taille k qui sont détruits dans y^d en lui ajoutant $2^s z$ change. Cela mène alors à une solution de

$$p_k(y^d + 2^s z) \equiv p_k(y^d) + p_k(z) + 1 \pmod 2. \tag{II.5.4}$$

Ainsi le lemme est démontré en combinant les égalités (II.5.3) et (II.5.4).

□

Nous pouvons désormais prouver le Théorème II.3.2.

Preuve du Théorème II.3.2. La preuve est similaire à la preuve réalisée pour le Théorème II.3.1. En effet, il suffit de remplacer le Lemme II.4.1 par le Lemme II.5.1. On remarque que la taille de y n'apparait que dans l'étape finale de la preuve, c'est-à-dire à l'inégalité (II.4.9). Alors on a

$$M \geq \frac{1}{4(2\alpha_{\max})^{1/d}}\, 2^{\ell} \gg_{P,k} N^{1/d},$$

ce qui prouve le Théorème II.3.2. □

II.6. Remarques

Remarque II.6.1. Dans le Chapitre V, nous discuterons de l'optimalité des bornes obtenues dans les Théorèmes II.3.1 et II.3.2.

Remarque II.6.2. Il est possible de généraliser les deux Théorèmes II.3.1 et II.3.2 aux polynômes $P \in \mathbb{Q}[X]$ unitaire de degré d tel que $P(\mathbb{N}) \subset \mathbb{N}$ sans de nouveaux éléments dans la preuve. Cependant, les démonstrations des Théorèmes II.3.1 et II.3.2 dépendent fortement du caractère unitaire du polynôme considéré. D'après nos simulations numériques, cette hypothèse n'est pas nécessaire pour obtenir les mêmes bornes. Une piste envisagée pour résoudre ce problème est d'utiliser le lemme de Hensel mais son application se restreint au cas où le coefficient dominant est impair.

Comme $M(S, N) \leq L(S, N)$, pour tout polynôme $P \in \mathbb{Z}[X]$, tel que $P(\mathbb{N}) \subset \mathbb{N}$ unitaire de degré d, on a

$$L(\mathcal{T}_P, N) \gg N^{1/d}. \tag{II.6.1}$$

La majoration précédente est la meilleure borne connue à ce jour pour la complexité linéaire de la suite $\mathcal{T}_P$. Cependant pour la suite de Thue–Morse le long des carrés, les estimations donnent que $L(\mathcal{T}_2, N) = \frac{N}{2} + o(N)$, voir [MW22] et la figure suivante.

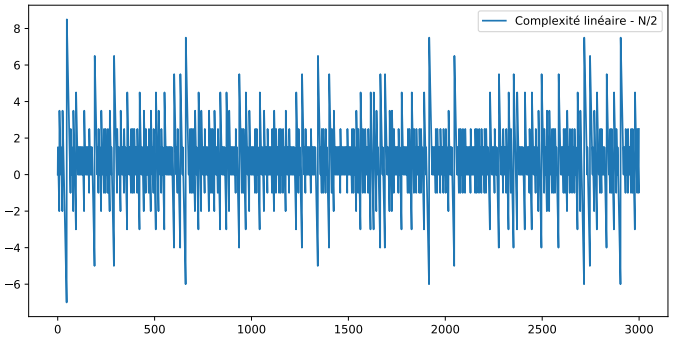


FIGURE II.1. Complexité linéaire de la suite de Thue–Morse le long des carrés.

Dans cette thèse, tous les calculs portant sur la complexité linéaire ont été réalisées à l'aide de l'implémentation de l'algorithme de Berlekamp–Massey donnée en annexe.

II.7. Ouvertures

II.7.1. Fonctions digitales. Cette thèse se focalise sur les applications des fonctions digitales pour créer des suites pseudo-aléatoires. Nous rappelons les définitions et propriétés des fonctions digitales. Dans un premier temps, nous introduisons les fonctions q-additives. Une fonction $f : \mathbb{N} \to \mathbb{R}$ est *q-additive* si pour tout $(a, b, j) \in \mathbb{N}^3$ tels que $0 \leq b < q^j$, on a

$$f(aq^j + b) = f(aq^j) + f(b).$$

De plus, si $f(aq^j) = f(a)$ pour tout $j \geq 1$, la fonction f est dite *fortement q-additive*. La fonction somme des chiffres s_q est une fonction fortement q-additive. Nous avons déjà eu l'occasion plusieurs fois cette propriété dans cette thèse.

Rappelons que $|n|_k$ est le nombre d'occurrences de k dans la représentation en base q de n. Si f est fortement q-additive, alors pour tout entier $n = \sum_{0\leq i\leq \ell} \varepsilon_i q^i$, on a

$$f\left(\sum_{0\leq i\leq \ell} \varepsilon_i q^i\right) = \sum_{0\leq i\leq \ell} f(\varepsilon_i) = \sum_{1\leq k<q} f(k)|n|_k.$$

Donc f est entièrement déterminée par les valeurs de $f(k)$ pour $1 \leq k < q$.

Une fonction $f : \mathbb{N} \to \mathbb{R}$ est *digitale* si f est de la forme

$$f(n) = \sum_{0\leq k<q} \alpha_k |n|_k$$

avec α_i réels. La fonction somme des chiffres s_q est un exemple de fonction digitale avec $\alpha_i = i$ pour tout $0 \leq i < q$. Notons que la fonction qui compte le nombre d'occurrences du bloc `11` n'est pas une fonction digitale mais une fonction *bloc-digitale*, où la définition de ces fonctions étend celle des fonctions digitales.

Les Théorèmes II.3.1 et II.3.2 semblent pouvoir être généralisés à d'autres fonctions digitales ou bloc-digitales. En effet, la preuve repose essentiellement sur la forte q-additivité de la fonction somme des chiffres pour la preuve du Théorème II.3.1. Les arguments de la preuve semblent aussi pouvoir se généraliser aux fonctions quasi-fortement q-additives, voir [AS93].

II.7.2. Autres sous-suites. Nous pouvons également nous intéresser à d'autres sous-suites que les sous-suites polynomiales. Nous donnons deux exemples :

- Pour la sous-suite des nombres premiers, Mauduit et Rivat [MR10] montrent une équirépartition de la suite $(s_q(p) \bmod m)_p$ sur $\{0, \dots, m-1\}$: pour tout $0 \leq a < m$, on note $d = (m, q-1)$ et on a
$$\frac{1}{\pi(N)} \operatorname{Card}\{p < N : s_q(p) \equiv a \pmod m\} \to \frac{d}{\varphi(d)m}.$$
Le même résultat est démontré par les mêmes auteurs pour la suite $(r_p \bmod m)_p$, voir [MR15].
- La suite de Piatetski-Shapiro est la suite $(\lfloor n^c \rfloor)_{n\geq 0}$ pour $c > 1$, appelée sous son nom après qu'il ait montré un théorème des nombres premiers pour $1 < c < \frac{12}{11}$, c'est-à-dire
$$\operatorname{Card}\{n < x : \lfloor n^c \rfloor \text{ est premier}\} \sim \frac{x}{c \log x}.$$
Müllner et Spiegelhofer [MS17] montrent que la suite de Thue–Morse le long de $\mathcal{T}_{n^c} := (t(\lfloor n^c \rfloor))_{n\geq 0}$ est normale pour $1 < c < 3/2$, et Spiegelhofer [Spi20] montre que $\mathcal{T}_{n^c}$ est simplement normale pour $1 < c < 2$.

La complexité d'ordre maximal de la suite de Thue–Morse le long de ces sous-suites est encore inconnue à ce jour.

CHAPITRE III

Suites morphiques et systèmes de numération

Dans ce chapitre, nous nous intéressons aux points fixes (à codage près) de morphismes et aux systèmes de numération. On note la suite de Fibonacci $\mathfrak{F} = (F_n)_{n\geq 0}$ définie par $F_0 = 0, F_1 = 1$ et $F_{n+2} = F_{n+1} + F_n$ pour $n \geq 0$.

III.1. Suites morphiques

III.1.1. Définition, exemples et propriétés. Soit h un morphisme, une lettre a est dite *mortelle* s'il existe $j \geq 1$ tel que $h^j(a) = \varepsilon$. Un morphisme h est dit prolongeable en a s'il existe une lettre a tel que $h(a) = ax$. Dans ce cas, la suite des mots $a, h(a), h^2(a), \ldots$ converge et on note le point fixe de h commençant par a le mot infini

$$h^\omega(a) = axh(x)h^2(x)\cdots$$

Définition III.1.1. Une suite $\mathcal{S}$ est une suite *morphique* s'il existe un morphisme $h : \Sigma \to \Sigma$ prolongeable en a et un codage $\pi : \Sigma \to \Delta$ tels que $\mathcal{S} = \pi \circ h^\omega(a)$.

Exemple III.1.1. Soit f le morphisme sur $\{0, 1\}$ défini par :

$$f : \begin{cases} 0 \mapsto 01 \\ 1 \mapsto 0. \end{cases}$$

Le mot infini, point fixe de f commençant par 0, est le mot $f^\omega(0) = 010010100100101\cdots$, appelé *mot infini de Fibonacci*. Ce mot est un mot sturmien.

Exemple III.1.2. Soit $\mathcal{S} = (s_n)_{n\geq 0}$ la suite caractéristique des carrés : $s_n = 1$ si n est un carré parfait et $s_n = 0$ sinon. La suite $\mathcal{S}$ est une suite morphique :

$$f : \begin{cases} a \mapsto abcc, \\ b \mapsto bcc, \\ c \mapsto c. \end{cases} \quad \text{et} \quad \pi : \begin{cases} a, b \mapsto 1, \\ c \mapsto 0. \end{cases}$$

La suite $\mathcal{S}$ est donnée par le mot infini $\pi \circ f^\omega(a) = 1100100001000000100\cdots$.

Le théorème suivant est une propriété sur la fréquence des lettres, si elle existe, d'une suite automatique ou morphique.

Théorème III.1.1 ([AS03, Theorem 8.4.5]**).** *Soit $\mathcal{S}$ une suite.*
(1) Si $\mathcal{S}$ est automatique et que la fréquence d'une lettre existe, elle est rationnelle.
(2) Si $\mathcal{S}$ est morphique et que la fréquence d'une lettre existe, elle est algébrique.

Notons que la fréquence logarithmique d'une lettre d'une suite automatique existe toujours et que ce résultat a été généralisé à certaines sous-suites des suites automatiques [ADM22].

Remarque III.1.1. Le mot infini de Fibonacci, voir l'Exemple III.1.1, n'est pas automatique car les fréquences de ses lettres sont irrationnelles. La suite caractéristique des carrés, voir l'Exemple III.1.2, n'est pas automatique, voir [Rit63], et la preuve utilise un argument de type *lemme de l'étoile*.

Remarque III.1.2. Il n'est a priori pas évident de déterminer si une suite morphique est automatique. Nous avons vu des exemples de suites morphiques non automatiques, voir les Exemples III.1.1 et III.1.2. L'exemple suivant est un exemple de suite automatique "cachée". Pour plus de détails sur ces suites automatiques cachées ou pour prouver qu'une suite n'est pas automatique voir [ADQ21, ASY22].

Exemple III.1.3. Soit f le morphisme sur $\{0,1,2\}$ suivant :

$$f : \begin{cases} 0 \mapsto 12 \\ 1 \mapsto 102 \\ 2 \mapsto 0. \end{cases}$$

Le mot infini est $f^\omega(1) = 102120102012\cdots$. Cette suite est automatique car $f^\omega(1) = \pi \circ g^\omega(1)$, voir [Ber79], où

$$g : \begin{cases} 0 \mapsto 12, & 1 \mapsto 13, \\ 2 \mapsto 20, & 3 \mapsto 21, \end{cases} \qquad \text{et } \pi : x \mapsto x \bmod 3.$$

Pour plus de détails sur les suites morphiques, voir [AS03] par exemple.

III.1.2. Systèmes dynamiques. L'ensemble des suites $\Sigma^{\mathbb{N}}$ sur un alphabet Σ est muni de la topologie produit discret, définie par la distance ultramétrique

$$d(x,y) = \begin{cases} \exp(-\inf\{k \geq 0, x_k \neq y_k\}), \text{ si } x \neq y, \\ 0, \text{ si } x = y. \end{cases}$$

Ainsi $\Sigma^{\mathbb{N}}$ est un espace métrique compact comme produit d'espaces compacts. Nous définissons $T : \Sigma^{\mathbb{N}} \to \Sigma^{\mathbb{N}}$ l'opération de décalage à une direction comme suit : pour $x = (x_k)_{k\geq 0}$, on pose $T(x) = (x_{k+1})_{k\geq 0}$. On note Tx l'image $T(x)$. L'application T est continue et surjective. L'orbite de x sous l'action de T est $\mathcal{O}(x) = \{T^n x, n \geq 0\}$.

Soit $f : \Sigma^* \to \Sigma^*$ un morphisme prolongeable en $a \in \Sigma$ tel que pour tout $b \in \Sigma$, $|f^n(b)| \to +\infty$ pour $n \to +\infty$. Dans la suite on notera $\mathbf{x}$ le point fixe de f commençant par a. Soit $X = \overline{\mathcal{O}(\mathbf{x})}$, l'adhérence dans $\Sigma^{\mathbb{N}}$ de l'orbite du point fixe de f sous l'action de T. La paire (X,T) est appelée *système dynamique topologique associé à f*. Une suite $\mathcal{S} \in X$ est dite *réalisée* par le système dynamique (X,T). L'entropie topologique du système dynamique (X,T) est égale à $\lim_{k\to+\infty} \frac{\log p_{\mathbf{x}}(k)}{k}$, voir [Kur03]. Une suite est dite *déterministe* si elle est réalisée par un système dynamique d'entropie topologique nulle. Cela corrobore bien le fait que presque toute suite satisfait une complexité en facteurs maximale, voir le Théorème I.2.2.

Pour plus de détails sur les systèmes dynamiques, voir [Que87].

III.1.3. Déterminisme des suites morphiques. Les suites morphiques qui sont aussi non automatiques sont de meilleures candidates pour des applications en cryptographie au regard de la complexité d'expansion. Cependant, cela n'affirme pas pour autant que ces suites sont pseudo-aléatoires. En effet, la complexité en facteurs des suites morphiques est faible, comme pour les suites automatiques.

Théorème III.1.2 ([AS03, Theorem 4.1.8])**.** *Soit $\mathcal{S}$ une suite morphique. Alors $p_{\mathcal{S}}(n) = O(n^2)$.*

D'après le théorème précédent, l'entropie topologique du système dynamique associé à une suite morphique est nulle. Les suites morphiques sont donc déterministes et ne peuvent pas être qualifiées de suites pseudo-aléatoires.

Exemple III.1.4. Reprenons l'exemple du mot infini de Fibonacci, voir Exemple III.1.1. Les figures suivantes illustrent le déterminisme de ce mot. En effet, une suite pseudo-aléatoire ne doit pas faire apparaître une telle structure régulière (qui se répète à l'infini).

Dans toute cette thèse, ce type de figure est obtenu selon la règle suivante : pour une suite $(s_n)_{n\geq 0}$, si C est le nombre de colonne, la ligne i correspond aux valeurs de $(s_{(i-1)C+n})_{0\leq n<C}$ et la lecture se fait de gauche à droite et haut en bas. Les carrés noirs représentent la valeur 1 et les carrés blancs la valeur 0.

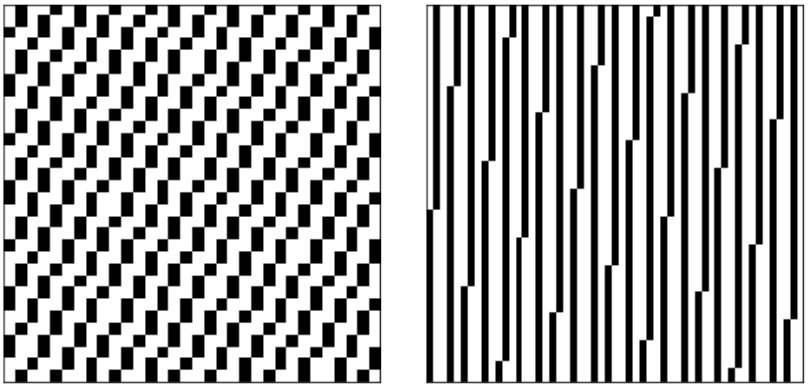


FIGURE III.1. Les 32×32 (à gauche) et les 55×55 (à droite) premiers termes du mot infini de Fibonacci.

Exemple III.1.5. Soit φ le morphisme sur l'alphabet $\{0, 1, 3, 4\}$ défini comme suit :

$$\varphi = \begin{cases} 0 \mapsto 03 \\ 1 \mapsto 43 \\ 3 \mapsto 1 \\ 4 \mapsto 01. \end{cases}$$

Le mot infini est $\varphi^\omega(0) = 03143011034343031011\cdots$. En représentant les premiers termes de $\varphi^\omega(0)$, nous observons que le mot $\varphi^\omega(0)$ ne peut pas être qualifié d'aléatoire. En effet, nous observons des "nuages" de certaines couleurs où certaines couleurs ne sont que très peu représentées.

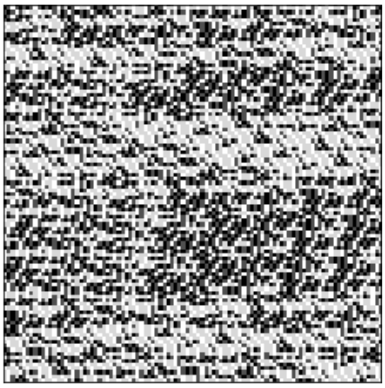


FIGURE III.2. Les 100×100 premiers termes de $\varphi^\omega(0)$.

Notons que le mot infini φ^ω évite les *cubes additifs* : il n'existe pas x, y, z des mots finis de même taille et de somme tels que le mot xyz est un facteur de $\varphi^\omega(0)$, voir [CCSS14].

III.2. Systèmes de numération

Un *système de numération* est un ensemble de règles permettant d'exprimer un entier $n \geq 0$ comme une combinaison linéaire $n = \sum_{0 \leq i \leq r} a_i(n) u_i$ en fonction d'éléments de base u_i. Les $a_i(n)$ sont appelés les *chiffres* de n. Le système de numération le plus connu est le système en base 10, où les éléments de base sont les 10^i, $i \geq 0$. La construction se généralise à tout entier $q \geq 2$ à la place de 10. En effet, tout entier $n \geq 0$ peut s'écrire $n = \sum_{0 \leq i \leq r} a_i q^i$, avec $0 \leq a_i < q$ et $a_r \neq 0$ et cela de manière unique. Dans ce cas, on parle de *système de numération en base q*.

Si la suite $(u_i)_{i \geq 0}$ est une suite strictement croissante tel que $u_0 = 1$, il existe une unique décomposition d'un entier n comme $n = \sum_{0 \leq i \leq r} a_i(n) u_i$ telle que $a_i \leq 0$ et $a_0 u_0 + \cdots + a_i u_i < u_{i+1}$ pour tout $i \geq 0$, voir [AS03, Theorem 3.1.1]. Pour trouver les chiffres a_i d'un entier n dans un tel système de numération, nous disposons de l'algorithme `glouton`.

Algorithme 1 : `Glouton`

```
1 Fonction Glouton(n):
2 ℓ = 0 ;
3 while u_{t+1} ≤ n do
4 |  ℓ ← ℓ + 1 ;
5 for i = t downto 0 do
6 |  a_i = ⌊N / u_i⌋ ;
7 |  n ← n − a_i u_i ;
8 |  output(a_i) ;
```

Nous introduisons un type particulier de système de numération, connu sous le nom de système de numération d'Ostrowski. Dans un premier temps, nous rappelons la définition de fraction continue d'un réel.

Proposition III.2.1 ([HW08])**.** Tout nombre réel $\alpha \geq 1$ peut s'exprimer de la manière unique suivante

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots}},$$

avec a_i entiers strictement positifs pour tout i. Dans ce cas, on note $\alpha = [a_0, a_1, a_2, \ldots]$.

Définition III.2.1. Soit $\alpha \geq 1$ un nombre réel et $\alpha = [a_0, a_1, a_2, \ldots]$. La suite $(a_i)_{i\geq 0}$ est appelée *développement en fraction continue* de α. On définit les suites $(p_i)_{i\geq -2}, (q_i)_{i\geq -2}$ comme suit

$$\begin{cases} (p_{-2}, q_{-2}) = (0, 1) \\ (p_{-1}, q_{-1}) = (1, 0) \end{cases}, \quad \begin{cases} p_i = a_i p_{i-1} + p_{i-2} \\ q_i = a_i q_{i-1} + q_{i-2}. \end{cases}$$

La suite des $\left(\frac{p_i}{q_i}\right)_{i\geq 0}$ est appelée suite des *réduites de* α et vérifie $\frac{p_i}{q_i} = [a_0, \ldots, a_i], i \geq 0$.

Il est clair que le développement en fraction continue d'un rationnel est fini. Le développement d'un irrationnel quadratique est quant à lui périodique, voir [Dav08, page 84].

Théorème III.2.1 ([AS03, Theorem 3.1.1])**.** *Soit $\alpha > 1$ un réel irrationnel et soit $(q_i)_{i\geq 0}$ la suite des dénominateurs de ses réduites. Chaque entier n peut être représenté de manière unique $n = \sum_{0\leq i\leq r} b_i q_i$ avec des entiers b_i tels que*

- $0 \leq b_0 < a_1$,
- $0 \leq b_i \leq a_{i+1}$,
- *Pour $i \geq 1$, si $b_i = a_{i+1}$ alors $b_{i-1} = 0$.*

Définition III.2.2. Soit $\alpha > 1$ un réel irrationnel. La suite $(q_i)_{i\geq 0}$ des dénominateurs des réduites de α forme un système de numération, appelé *système de numération α-Ostrowski.*

Exemple III.2.1. Soit $\alpha = 1 + \sqrt{2}$, le nombre d'argent. Son développement en fraction continue est $\alpha = [2, 2, 2, \ldots]$ et la suite des dénominateurs $(q_i)_{i\geq 0}$ est $1, 2, 5, 12, 29, \ldots$. Par exemple, l'entier 26 se décompose selon cette base de la manière suivante :

$$24 = 2 \cdot 12 + 0 \cdot 5 + 1 \cdot 2 + 0 \cdot 1$$

et on lui associe le mot $(26)_\alpha = 2010$.

Pour plus de détails sur les systèmes de numérations, voir [AS03, Lot02, Rig14].

CHAPITRE IV

Somme des chiffres en base de Zeckendorf

Le système de numération de Zeckendorf est un système de numération se basant sur la suite de Fibonacci. Dans cette thèse, l'étude de la fonction somme des chiffres s_F, associée à ce système de numération est survenue naturellement comme possible extension des travaux réalisés dans le Chapitre II.

Ce chapitre est structuré comme suit : nous dressons dans un premier temps l'état de l'art sur la base de Zeckendorf et la fonction somme des chiffres associée s_F. Dans un second temps, nous reproduisons les résultats [JPS21], portant sur la complexité d'ordre maximal de la suite $\mathcal{S}_Z = (s_F(n) \bmod 2)_{n\geq 0}$ et de ses sous-suites polynomiales. Enfin, nous donnons des possibles ouvertures aux résultats principaux de ce chapitre.

IV.1. Définition et propriétés

Nous reprenons les notations des chapitres précédents. Soit $\mathfrak{F} = (F_n)_{n\geq 0}$, avec $F_0 = 0$, $F_1 = 1$ et $F_{n+2} = F_{n+1} + F_n$ pour tout $n \geq 0$. On note également $\varphi = (1+\sqrt{5})/2$ le nombre d'or. Rappelons la formule de Binet :

$$F_n = \frac{1}{\sqrt{5}}\varphi^n - \frac{1}{\sqrt{5}}\varphi^{-n}, \qquad n \geq 0. \tag{IV.1.1}$$

Théorème IV.1.1 ([Zec72])**.** *Tout entier $n \geq 0$ est représentable de manière unique*

$$n = \sum_{0\leq i\leq \ell} \varepsilon_i F_{i+2},$$

avec $\varepsilon_i(n) \in \{0,1\}$, $\varepsilon_\ell = 1$ et $\varepsilon_i\varepsilon_{i+1} = 0$ pour tout $0 \leq 0 < \ell$.

Autrement dit, tout entier naturel est la somme unique d'un nombre fini de nombres de Fibonacci, non consécutifs deux à deux. Lekkerkerker attribua le nom du théorème à Zeckendorf, voir [Lek51], même si la preuve de Zeckendorf sera publiée plus tard, voir [Zec72].

Si pour tout $i \geq 0$ les entiers ε_i vérifient $\varepsilon_i\varepsilon_{i+1} = 0$, ils sont dits *non-adjacents*. L'hypothèse de non-adjacence des ε_i est nécessaire pour garantir l'unicité. Par exemple 9 admet au moins deux représentations différentes sans cette hypothèse : $9 = 8+1 = F_6 + F_2 = 5+3+1 = F_5 + F_4 + F_2$. La représentation en base de Zeckendorf d'un entier n est notée $(n)_F := \varepsilon_\ell \cdots \varepsilon_0$ tel que $n = \sum_{0\leq i\leq \ell} \varepsilon_i F_{i+2}$ avec $\varepsilon_i \in \{0,1\}$ et $\varepsilon_i\varepsilon_{i+1} = 0$ pour tout $0 \leq i < \ell$.

Le système de numération induit par le Théorème IV.1.1 est appelé *système de numération de Zeckendorf*. C'est également un système de numération d'Ostrowski pour l'irrationnel φ. En effet le développement en fraction continue de φ est $\varphi = [1,1,\ldots]$. Par

conséquent, la suite des dénominateurs des réduites $(q_i)_{i\geq 0}$ de φ est la suite $1, 1, 2, 3, 5, \ldots$, qui est la suite $(F_{i+1})_{i\geq 0}$.

Pour tout $\ell \geq 0$, on a $F_\ell = \left\lfloor \frac{\varphi^\ell}{\sqrt{5}} + \frac{1}{2} \right\rfloor$. Le nombre de chiffres nécessaires pour écrire n en base de Zeckendorf, appelé la *longueur* de n en base de Zeckendorf, est égal à $\left\lfloor \frac{\log(n\sqrt{5}+1/2)}{\log(\varphi)} \right\rfloor - 1$.

Définition IV.1.1. Soit $s_F : \mathbb{N} \to \mathbb{N}$ la fonction définie par $s_F(n) = \sum_{0\leq i\leq \ell} \varepsilon_i$, où $(n)_F = \varepsilon_\ell \cdots \varepsilon_0$.

La fonction s_F est appelée la *fonction somme des chiffres en base de Zeckendorf*. La suite $(s_F(n))_{n\geq 0}$ a été étudiée par plusieurs auteurs, voir [OEI22, A007895] pour plus de détails. Stoll [Sto13] étudie cette suite dans le cadre problème de Gelfond, voir le Problème II.2.1, où s_q est remplacée par s_F.

Théorème IV.1.2 ([Sto13]**).** *Soient $m \geq 2$ et $P \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$ de degré $d \geq 2$. Pour tout $a \in \mathbb{Z}$ tel que $0 \leq a < m$, on a*

$$\mathrm{Card}\{n < N : s_F(P(n)) \equiv a \pmod m\} \gg_{P,m} N^{4/(6d+1)}.$$

De plus, Stoll [Sto13] montre également un analogue au résultat de Stolarsky sur la somme des chiffres en base 2, voir le Chapitre VI pour plus de détails.

Théorème IV.1.3 ([Sto13]**).** *Soient $m \geq 2$ et $P \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$ de degré $d \geq 2$. On a*

$$\liminf_{n\to+\infty} \frac{s_F(P(n))}{s_F(n)} = 0, \qquad \limsup_{n\to+\infty} \frac{s_F(P(n))}{s_F(n)} = +\infty.$$

Remarque IV.1.1. Le langage des mots ne contenant pas le facteur `11` est reconnu par l'automate suivant :

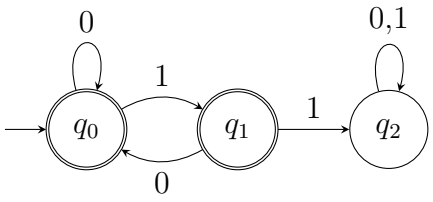


FIGURE IV.1. DFA acceptant les mots binaires sans deux 1 consécutifs.

À l'instar de la fonction s_q somme des chiffres en base q, la fonction somme des chiffres en base de Zeckendorf s_F est sous-additive : pour tous entiers $n, m \geq 0$, on a

$$s_q(m+n) \leq s_q(m) + s_q(n),$$
$$s_F(m+n) \leq s_F(m) + s_F(n),$$

voir [HLS11a, Proposition 2.2] pour s_q et [Sto13, Proposition 1] pour s_F. Cependant il existe des différences notables entre les fonctions s_q et s_F. Par exemple s_q est une

fonction sous-multiplicative, c'est-à-dire pour tous entiers $n, m \geq 0$, $s_q(a \cdot b) \leq s_q(a) \cdot s_q(b)$, voir [HLS11a, Proposition 2.2], alors que s_F n'est pas sous-multiplicative. En effet,

$$s_F(11 \cdot 18) = s_F(198) = s_F(144 + 34 + 13 + 5 + 2) = s_F(F_{12} + F_9 + F_7 + F_5 + F_3) = 5,$$
$$s_F(11) \cdot s_F(18) = s_F(8 + 3) \cdot s_F(13 + 5) = s_F(F_6 + F_4) \cdot s_F(F_7 + F_5) = 4 < 5.$$

Remarque IV.1.2. Intuitivement, la fonction s_F n'est vraisemblablement pas sous-multiplicative dans la mesure où la multiplication d'un entier par un nombre de Fibonacci modifie considérablement la décomposition en base de Zeckendorf de cet entier. Par exemple, les seuls nombres de Fibonacci qui sont des puissances pures d'un nombre de Fibonacci sont les nombres $1, 8$ et 144, voir [BMS06]. La preuve de la sous-multiplicativité de la fonction s_q repose, entre autres, sur l'identité $s_q(q^j a) = s_q(a)$ pour tous entiers $a, j \geq 0$.

Soient les représentations de Zeckendorf de deux entiers $m, n \geq 0$,

$$m = \sum_{j=k_1}^{K_1} \varepsilon_j^{(1)} F_{j+2}, \qquad n = \sum_{j=k_2}^{K_2} \varepsilon_j^{(2)} F_{j+2},$$

avec $\varepsilon_{k_i}^{(i)} = \varepsilon_{K_i}^{(i)} = 1$ pour $i = 1, 2$ et supposons que $k_1 \leq k_2$. On dit que m et n *n'interfèrent pas* si $k_2 - K_1 \geq 2$, comme précédemment pour le cas de la q-base classique. Ceci implique que les blocs de chiffres n'interférèrent pas et qu'aucune normalisation n'est nécessaire après addition chiffre par chiffre de m et n. Par conséquent, on a

$$s_F(m + n) = s_F(m) + s_F(n). \tag{IV.1.2}$$

Intéressons-nous, à partir d'exemples, à la propagation de retenues en base de Zeckendorf.

Exemple IV.1.1. Soient $a = 3 = F_4 = (\mathtt{100})_F$ et $b = 6 = F_5 + F_2 = (\mathtt{1001})_F$. Alors l'addition de a à b est

		1	0	0	1	$= 6$
$+$			1	0	0	$= 3$
$=$		①	①	0	1	
$=$	①	0	0	0	1	$= 9.$

La propagation de retenues en base de Zeckendorf est donc "transversale", car $F_j = F_{j-1} + F_{j-2}$, pour tout $j \geq 2$. Ce cas de figure n'apparaît pas dans une base q classique. Un deuxième cas de figure intéressant est celui où a et b partagent un même nombre de Fibonacci dans leurs décomposition. Soient $a = b = 5 = F_5 = (\mathtt{1000})_F$. Alors l'addition de a à b est

		1	0	0	0	$= 5$
$+$		1	0	0	0	$= 5$
$=$		②	0	0	0	
$=$	①	0	0	①	0	$= 10.$

La propagation de retenues se fait dans les deux sens, car $2F_j = F_{j+1} + F_{j-2}$ pour tout $j \geq 2$ ce qui n'est pas le cas dans une base q classique.

Certaines sommes peuvent cumuler ces deux nouveaux phénomènes. Prenons par exemple $5 = F_5$, $6 = F_5 + F_2$. Donc $5 + 6 = 11 = F_6 + F_4$, car on a :

$$\begin{array}{cccccc} & & 1 & 0 & 0 & 0 \\ + & & 1 & 0 & 0 & 1 \\ \hline = & & 2 & 0 & 0 & 1 \\ \hline = & 1 & 0 & 0 & 1 & 1 \\ \hline = & 1 & 0 & 1 & 0 & 0 \end{array}$$

De manière générale, l'addition de deux entiers en base de Zeckendorf est réalisable en un temps linéaire, voir [AUFP13].

Remarque IV.1.3. Rappelons que pour la fonction s_q somme des chiffres en base q, nous avons la propriété suivante : pour a, b des entiers tels que $a < q^\ell$ pour un certain $\ell \geq 0$, on a

$$s_q(a + q^\ell b) = s_q(a) + s_q(b), \tag{IV.1.3}$$

où s_q est la fonction somme des chiffres en base q. Cependant, l'identité (IV.1.3) n'est pas vraie en général si nous remplaçons q^ℓ par F_ℓ, s_q par s_F et la condition $a < q^\ell$ par $a < F_\ell$. Par exemple, $s_F(1 + F_4 \cdot 2) = s_F(1 + 4) = s_F(5) = 1$ alors que $s_F(1) + s_F(2) = 2$. Nous utiliserons dans la suite les nombres de Lucas pour remplacer le décalage q^ℓ.

Nous introduisons l'objet d'étude central de ce chapitre.

Définition IV.1.2. Soit $\mathcal{S}_Z$ la suite définie par

$$\mathcal{S}_Z = (s_F(n) \bmod 2)_{n \geq 0}\,.$$

La suite $\mathcal{S}_Z$ est appelée la suite de Fibonacci–Thue–Morse, par analogie à la définition de la suite de Thue–Morse. La suite $\mathcal{S}_Z$ est une suite morphique, voir [Bru95, p.14], engendrée par le morphisme et le codage :

$$f : \begin{cases} a \mapsto ab \\ b \mapsto c \\ c \mapsto cd \\ d \mapsto a \end{cases} \quad \text{et} \quad g : \begin{cases} a \mapsto \mathtt{0} \\ b \mapsto \mathtt{1} \\ c \mapsto \mathtt{1} \\ d \mapsto \mathtt{0}. \end{cases}$$

Par conséquent, nous obtenons que $\mathcal{S}_Z = \mathtt{01110100100011000101 1}\cdots$

La suite $\mathcal{S}_Z$ est une suite *Fibonacci-automatique*, c'est-à-dire que c'est une suite engendrée par un automate où les chiffres lus par l'automate sont ceux de la décomposition d'un nombre en base de Zeckendorf. Dans toute cette thèse, nous réservons le terme de suite *automatique* pour désigner une suite k-automatique, voir la Paragraphe I.3.2. L'automate qui génère la suite $\mathcal{S}_Z$ est le suivant :

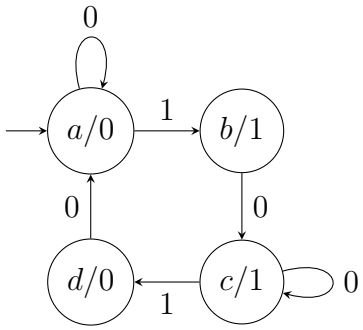


FIGURE IV.2. Automate pour la somme des chiffres modulo 2 en base de Zeckendorf.

Certaines mesures de complexité sont déterminées pour la suite $\mathcal{S}_Z$:

- La complexité en facteurs $p_{\mathcal{S}_Z}(n)$ a été totalement déterminée par Shallit [Sha21].
- La complexité d'expansion $(E(\mathcal{S}_Z, N))_{N\geq 1}$ est non bornée. En effet, la suite $\mathcal{S}_Z$ n'est pas une suite automatique, voir [DMS18, Remark 1].

En tant que suite morphique, on ne s'attend pas à ce que $\mathcal{S}_Z$ possède des bonnes propriétés aléatoires. Voici une représentation graphique de ces premiers termes.

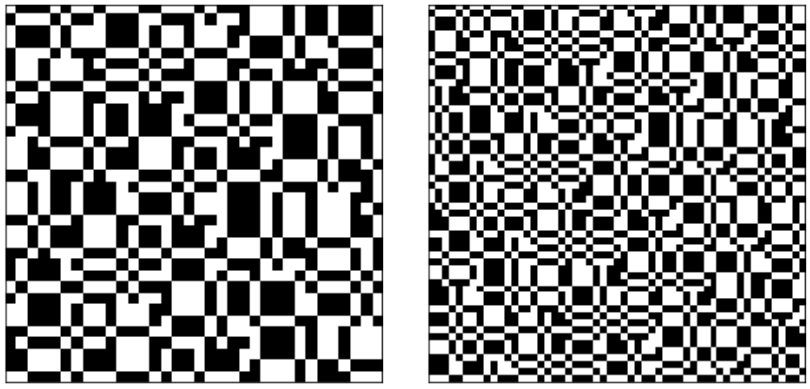


FIGURE IV.3. Les 32×32 (à gauche) et les 55×55 (à droite) premiers termes de $\mathcal{S}_Z$.

Remarque IV.1.4. L'objectif de ce chapitre est d'étendre l'étude réalisée dans les Théorèmes II.3.1 et II.3.2 à la base de Zeckendorf. Pour la preuve de ces résultats, une compréhension de la propagation de retenues était nécessaire. Les nouveaux phénomènes de cette propagation de retenues seront visibles dans les résultats, voir Théorème IV.2.2.

IV.2. Résultats

Théorème IV.2.1 ([JPS21])**.** *Il existe $N_0 > 0$ tel que pour tout $N > N_0$ on a*

$$M(\mathcal{S}_Z, N) \geq \frac{1}{\varphi + \varphi^3} N + 1.$$

Nous pouvons en déduire, d'après l'inégalité (I.2.1), que la corrélation d'ordre 2 de la suite $\mathcal{S}_Z$ est grande. Cela confirme, une fois de plus, que la suite $\mathcal{S}_Z$ ne peut être qualifiée d'aléatoire. Shallit [Sha22b] exhibe une formule exacte de la complexité d'ordre maximal de $\mathcal{S}_Z$, voir Paragraphe V.3 pour plus de détails.

Les représentations graphiques suivantes suggèrent que les sous-suites polynomiales de $\mathcal{S}_Z$ possèdent un comportement plus proche d'une suite aléatoire, comme pour le cas de la suite de Thue–Morse $\mathcal{T}$.

FIGURE IV.4. Les 55×55 premiers termes de $\mathcal{S}_Z$ le long des carrés (à gauche) et le long des cubes (à droite).

Pour $P \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$, on note $\mathcal{S}_{Z,P} = (s_F(P(n)) \bmod 2)_{n\geq 0}$ la suite de $\mathcal{S}_Z$ le long du polynôme P.

Théorème IV.2.2 ([JPS21])**.** *Soit $P(X) \in \mathbb{Z}[X]$ un polynôme unitaire de degré $d \geq 2$ avec $P(\mathbb{N}) \subset \mathbb{N}$. La complexité d'ordre maximal de la suite $\mathcal{S}_{Z,P}$ satisfait*

$$M(\mathcal{S}_{Z,P}, N) \gg N^{1/(2d)}, \qquad N \to \infty,$$

où la constante ne dépend que de P.

IV.3. Preuve du Théorème IV.2.1

Pour prouver une borne inférieure de la complexité d'ordre maximal au rang N de la suite $\mathcal{S}_Z$ ou ses sous-suites, nous utilisons le Lemme I.2.3, page 49. Nous avons le lemme suivant.

Lemme IV.3.1. *Soit $\ell \geq 2$, pour tout $0 \leq n < F_\ell$ on a*

$$s_F(n + F_{\ell+1}) = s_F(n + F_{\ell+2}),$$
$$s_F(F_\ell + F_{\ell+1}) \bmod 2 \neq s_F(F_\ell + F_{\ell+2}) \bmod 2.$$

PREUVE. Si $n < F_\ell$, les termes n et $F_{\ell+1}$, respectivement, n et $F_{\ell+2}$ sont non-interférents, voir l'identité (IV.1.2). La deuxième inégalité provient de $s_F(F_\ell + F_{\ell+1}) = 1$ et $s_F(F_\ell + F_{\ell+2}) = 2$. □

Nous pouvons désormais montrer le Théorème IV.2.1.

Preuve du Théorème IV.2.1. On choisit $\ell \geq 2$ tel que $F_\ell + F_{\ell+2} \leq N < F_{\ell+1} + F_{\ell+3}$. Ceci implique trivialement que $N \geq F_2 + F_4 = 4$. D'après le Lemme IV.3.1, les deux sous-suites $(s_F(n + F_{\ell+1}) \bmod 2)_{0\leq n\leq F_\ell}$ et $(s_F(n + F_{\ell+2}) \bmod 2)_{0\leq n\leq F_\ell}$ sont de mêmes longueurs, identiques pour $0 \leq n < F_\ell$ et de valeur différente pour $n = F_\ell$. Par conséquent, d'après le Lemme I.2.4, on a

$$M(\mathcal{S}_Z, N) \geq F_\ell + 1.$$

De plus, comme $\lim_{n\to\infty} \frac{F_{n+k}}{F_n} = \varphi^k$, pour $k \geq 1$, on a $N < (\varphi+\varphi^3)F_\ell$ pour ℓ suffisamment grand. Le théorème est bien prouvé. □

Remarque IV.3.1. La méthode ne semble pas s'adapter pour les sous-suites de la forme $(an+b)_{n\geq 0}$ avec $a \neq 1$. À l'aide des arguments utilisés dans le paragraphe suivant, nous pouvons établir une borne de l'ordre de $N^{1/2}$ mais cela ne correspond pas aux simulations.

IV.4. Preuve du Théorème IV.2.2

IV.4.1. Nombres de Lucas.

Définition IV.4.1. Soit $\mathcal{L} = (L_n)_{n\geq 0}$ la suite définie par $L_0 = 2$, $L_1 = 1$ et $L_{n+2} = L_{n+1} + L_n$ pour tout $n \geq 0$.

Pour tout $n \geq 1$, on a $L_n = F_{n+1} + F_{n-1}$, et $s_F(L_n) = 2$. Nous introduisons une série de lemmes portant sur des propriétés élémentaires des nombres de Lucas et leurs liens avec la décomposition en base de Zeckendorf. Le premier lemme décrit la somme des chiffres de deux nombres de Lucas.

Lemme IV.4.1. *Soient k, ℓ deux entiers tels que $k \geq \ell \geq 5$. On a*

$$s_F(L_k + L_\ell) \equiv 1 \pmod 2 \quad \Longleftrightarrow \quad k = \ell + 2.$$

Preuve. On a $L_k = F_{k+1} + F_{k-1}$ et $L_\ell = F_{\ell+1} + F_{\ell-1}$. On décale deux blocs de `101` l'un contre l'autre, on les additionne et on étudie la propagation de retenues. De manière évidente, si on a $k \geq \ell + 4$, alors il n'y a pas de propagation de retenues et $s_F(L_k + L_\ell) = 4 \equiv 0 \pmod 2$. Par symétrie, il ne reste que quatre cas à considérer. Supposons $\ell \geq 5$:

- Cas $k = \ell$:

			1	0	1		
+			1	0	1		
=			2	0	2		
=		1	0	1	1	0	1
=	1	0	0	0	0	0	1

  Alors on a $s_F(L_k + L_\ell) \equiv 0 \pmod 2$.
- Cas $k = \ell + 1$:

		1	0	1		
+			1	0	1	
=	1	0	1	0	0	

  Alors on a $s_F(L_k + L_\ell) \equiv 0 \pmod 2$.

- Cas $k = \ell + 2$:

$$\begin{array}{cccccccc} & & 1 & 0 & 1 & & & \\ + & & & & 1 & 0 & 1 & \\ \hline = & & 1 & 1 & 0 & 0 & 2 & \\ \hline = & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

  Alors on a $s_F(L_k + L_\ell) \equiv 1 \pmod 2$.
- Cas $k = \ell + 3$:

$$\begin{array}{ccccccc} & & 1 & 0 & 1 & & \\ + & & & & & 1 & 0 & 1 \\ \hline = & & 1 & 1 & 0 & 0 & 0 & 1 \\ \hline = & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

  Alors on a $s_F(L_k + L_\ell) \equiv 0 \pmod 2$.

Donc le lemme est démontré. □

Nous avons déjà évoqué dans le Chapitre II le fait que l'écriture en base q de xq^j est un décalage vers la gauche de longueur j de l'écriture de x en base q. Dans la base de Zeckendorf, les nombres de Fibonacci n'ont pas cette propriété, voir Remarque IV.1.2. Cependant, les nombres de Lucas forment un analogue intéressant des puissances de q dans une q-base. En effet, nous avons les formules suivantes pour les puissances des nombres de Lucas, voir par exemple [Sto13].

Lemme IV.4.2 ([Sto13])**.** *Pour tous $k \geq \ell \geq 0$ et $h \geq 0$, on a*
(1) $L_k L_\ell = L_{k+\ell} + (-1)^\ell L_{k-\ell}$,
(2) $L_k F_\ell = F_{k+\ell} - (-1)^\ell F_{k-\ell}$,
(3) $L_k^h = \sum_{i=0}^{\frac{h-1}{2}} \binom{h}{i} (-1)^{ik} L_{(h-2i)k}$, *$h$ impair,*
(4) $L_k^h = \sum_{i=0}^{\frac{h}{2}-1} \binom{h}{i} (-1)^{ik} L_{(h-2i)k} + \binom{h}{h/2} (-1)^{hk/2}$, *$h$ pair.*

Le lemme suivant a un rôle important dans la construction des sommes non-interférentes.

Lemme IV.4.3. *Soit $m > 0$ un entier. Il existe des entiers $u(m) = u$ et $v = v(m)$ des entiers et des coefficients non-adjacents*

$$\varepsilon_{-(2u+1)}(m), \ldots, \varepsilon_{2u+v}(m) \in \{0, 1\},$$

avec $\varepsilon_{-2u-1}(m) = \varepsilon_{2u+v}(m) = 1$ tels que pour tout $k \geq 2u + 3$,

$$mL_k = \sum_{j=-(2u+1)}^{2u+v} \varepsilon_j(m) F_{k+j}$$

et $[-(2u+1), 2u+v] \subseteq [k-\ell-1, k+\ell+1]$ où ℓ vérifie $F_\ell \leq m < F_{\ell+1}$.

Par conséquent, pour des entiers m et k assez grand, l'écriture de mL_k en base de Zeckendorf est centrée autour de F_k.

Nous commençons la démonstration du Théorème IV.2.2 par le cas particulier $P(X) = X^d$ pour $d \geq 2$. En effet, ce cas est plus simple et il est instructif pour le cas général.

IV.4.2. Cas des sous-suites monomiales. Soit $P(X) = X^d$ pour $d \geq 2$. On note $\overline{\lambda} = \lambda \bmod 2$ avec $\overline{\lambda} \in \{0, 1\}$. Dans un premier temps, nous introduisons une série de lemmes. Nous commençons par étudier l'expression $(n + L_\ell)^d$ en fonction des nombres de Lucas $L_{\lambda\ell}$ pour $1 \leq \lambda \leq d$.

Lemme IV.4.4. *Soit $n \geq 0$ un entier et $\ell \geq 0$ un entier pair. On a*

$$(n + L_\ell)^d = \sum_{i=0}^{\frac{d-\overline{d}}{2}} \eta_{i,0} n^{d-2i} + \sum_{\lambda=1}^{d} \left(\sum_{i=\frac{\lambda-\overline{\lambda}}{2}}^{\frac{d-\overline{d}}{2}-\overline{\lambda}} \eta_{i,\lambda} n^{d-2i-\overline{\lambda}} \right) L_{\lambda\ell},$$

avec $\eta_{i,\lambda} = \binom{d}{2i+\overline{\lambda}} \binom{2i+\overline{\lambda}}{i-\frac{\lambda-\overline{\lambda}}{2}}$.

Preuve. Supposons que d est pair. Alors par Lemme IV.4.2 on a

$$\begin{aligned}
(n + L_\ell)^d &= \sum_{i=0}^{d/2} \binom{d}{2i} n^{d-2i} L_\ell^{2i} + \sum_{i=0}^{d/2-1} \binom{d}{2i+1} n^{d-2i-1} L_\ell^{2i+1}, \\
&= n^d + \sum_{i=1}^{d/2} \binom{d}{2i} n^{d-2i} \left(\sum_{j=0}^{i-1} \binom{2i}{j} L_{(2i-2j)\ell} + \binom{2i}{i} \right) \\
&\quad + \sum_{i=0}^{d/2-1} \binom{d}{2i+1} n^{d-2i-1} \left(\sum_{j=0}^{i} \binom{2i+1}{j} L_{(2i-2j+1)\ell} \right), \\
&= \sum_{i=0}^{\frac{d}{2}} \binom{d}{2i} \binom{2i}{i} n^{d-2i} \\
&\quad + \sum_{\lambda=1}^{d} \left(\sum_{i=\frac{\lambda-\overline{\lambda}}{2}}^{\frac{d}{2}-\overline{\lambda}} \binom{d}{2i+\overline{\lambda}} \binom{2i+\overline{\lambda}}{i-\frac{\lambda-\overline{\lambda}}{2}} n^{d-2i-\overline{\lambda}} \right) L_{\lambda\ell}.
\end{aligned}$$

Pour d impair, la preuve est similaire et on obtient

$$\begin{aligned}
(n + L_\ell)^d &= \sum_{i=0}^{\frac{d-1}{2}} \binom{d}{2i} \binom{2i}{i} n^{d-2i} \\
&\quad + \sum_{\lambda=1}^{d} \left(\sum_{i=\frac{\lambda-\overline{\lambda}}{2}}^{\frac{d-1}{2}-\overline{\lambda}} \binom{d}{2i+\overline{\lambda}} \binom{2i+\overline{\lambda}}{i-\frac{\lambda-\overline{\lambda}}{2}} n^{d-2i-\overline{\lambda}} \right) L_{\lambda\ell}.
\end{aligned}$$

□

Lemme IV.4.5. *Soit $n \geq d$ un entier $d \geq 3$ un entier impair et $1 \leq \lambda \leq d$. Alors on a*

$$\sum_{i=\frac{\lambda-\overline{\lambda}}{2}}^{\frac{d-1}{2}-\overline{\lambda}} \binom{d}{2i+\overline{\lambda}} \binom{2i+\overline{\lambda}}{i-\frac{\lambda-\overline{\lambda}}{2}} n^{d-2i-\overline{\lambda}} < \sum_{i=0}^{\frac{d-1}{2}} \binom{d}{2i} \binom{2i}{i} n^{d-2i}.$$

Preuve. On distingue les deux cas suivants :

(1) Si $\overline{\lambda} = 0$, on a

$$\sum_{i=0}^{\frac{d-1}{2}} \binom{d}{2i} \binom{2i}{i} n^{d-2i} = \sum_{i=0}^{\frac{\lambda}{2}-1} \binom{d}{2i} \binom{2i}{i} n^{d-2i} + \sum_{i=\frac{\lambda}{2}}^{\frac{d-1}{2}} \binom{d}{2i} \binom{2i}{i} n^{d-2i}.$$

Or $\binom{2i}{i} \geq \binom{2i}{i-\frac{\lambda}{2}}$ pour tout $i \geq \frac{\lambda}{2}$, on a bien le résultat voulu.

(2) Si $\overline{\lambda} = 1$, on a

$$\sum_{i=0}^{\frac{d-1}{2}} \binom{d}{2i} \binom{2i}{i} n^{d-2i} = \sum_{i=0}^{\frac{\lambda-1}{2}-1} \binom{d}{2i} \binom{2i}{i} n^{d-2i} + \sum_{i=\frac{\lambda-1}{2}}^{\frac{d-1}{2}-1} \binom{d}{2i} \binom{2i}{i} n^{d-2i} + d\binom{d-1}{\frac{d-1}{2}} n,$$

et

$$\sum_{i=\frac{\lambda-1}{2}}^{\frac{d-1}{2}-1} \binom{d}{2i+1} \binom{2i+1}{i-\frac{\lambda-1}{2}} n^{d-2i-1} \leq \frac{d-(\lambda-1)}{n} \sum_{i=\frac{\lambda-1}{2}}^{\frac{d-1}{2}-1} \binom{d}{2i} \binom{2i}{i-\frac{\lambda-1}{2}} n^{d-2i}.$$

On a bien le résultat pour $n \geq d$.

□

Pour $k \geq 1$, nous considérons

$$t(k) = m_3 L_{6k} - m_2 L_{4k} + m_1 L_{2k} + m_0 L_0,$$

avec des paramètres réels m_i. Pour $d \geq 1$, on pose

$$T_d(k) = t(k)^d = \sum_{0 \leq i \leq 3d} c_i L_{2ik}.$$

Notons que pour k suffisamment grand, les coefficients c_i ne dépendent pas de k.

Lemme IV.4.6 ([Sto13, Lemma 4])**.** *Soit $M \geq 1$, et $m_0, m_1, m_2, m_3 \in \mathbb{R}$ avec*

$$1 \leq m_0, m_1, m_3 < M, \qquad 0 < m_2 < \frac{1}{d^3(32M)^d}.$$

Alors on a $c_{3d} > 0$, $c_{3d-1} < 0$ et $c_i > 0$ pour $i = 0, 1, \ldots, 3\ell - 2$.

Ce lemme nous permettra de construire deux éléments de différentes sommes de chiffres en base de Zeckendorf lorsqu'ils seront élevés à la puissance d. En effet, seul le coefficient sous-dominant est négatif et nous sommes capables de créer un bloc de chiffres de la forme $\mathtt{1010\cdots10}$ de longueur k pour tout k suffisamment grand. Comme nous allons le préciser plus tard, le changement de k à $k+1$ ajoute exactement un bloc de $\mathtt{10}$. Nous avons déjà utilisé une méthode très similaire pour le cas précédent de la suite de Thue–Morse, voir le Chapitre II.

Soient $\alpha \geq 1$ un entier tel que

$$\varphi^\alpha > d^3\varphi(32\varphi)^d,$$

et $m_2 = 1$, m_0, m_1, m_3 des entiers tels que

$$\varphi^{\alpha-1} \leq m_0, m_1, m_3 < \varphi^\alpha. \tag{IV.4.1}$$

Sous ces hypothèses, $T_d(k)$ et $T_d(k+1)$ n'ont que des coefficients entiers positifs, à l'exception du coefficient de $L_{2k(3d-1)}$ et $L_{2(k+1)(3d-1)}$ respectivement, voir [Sto13]. Notons que ces coefficients sont les mêmes pour ces deux polynômes si on suppose k suffisamment grand.

Lemme IV.4.7. *Soit $d \geq 3$ un entier impair et $k \geq 0$ un entier suffisamment grand. Soit n un entier tel que $n \geq d$ et*

$$\sum_{i=0}^{(d-1)/2} \binom{d}{2i}\binom{2i}{i} n^{d-2i} < F_{3k}. \tag{IV.4.2}$$

Alors on a

$$s_F((n+L_{6k+2})^d) = s_F((n+L_{6k+4})^d), \tag{IV.4.3}$$

$$s_F(T_d(k)) \bmod 2 \neq s_F(T_d(k+1)) \bmod 2. \tag{IV.4.4}$$

Preuve. Soit $\ell \geq 0$ un entier pair. D'après le Lemme IV.4.4, on a

$$(n+L_\ell)^d = \sum_{i=0}^{\frac{d-1}{2}} \eta_{i,0} n^{d-2i} + \sum_{\lambda=1}^{d} \left(\sum_{i=\frac{\lambda-\overline{\lambda}}{2}}^{\frac{d-1}{2}-\overline{\lambda}} \eta_{i,\lambda} n^{d-2i-\overline{\lambda}} \right) L_{\lambda\ell}. \tag{IV.4.5}$$

La condition (IV.4.2) et le Lemme IV.4.5 impliquent que le coefficient devant chaque $L_{\lambda\ell}$, pour $1 \leq \lambda \leq d$, est strictement inférieur à F_{3k}. Pour ℓ suffisamment grand, le rang des chiffres de

$$\left(\sum_{i=\frac{\lambda-\overline{\lambda}}{2}}^{d/2-\overline{\lambda}} \binom{d}{2i+\overline{\lambda}} \binom{2i+\overline{\lambda}}{i-\frac{\lambda-\overline{\lambda}}{2}} n^{d-2i-\overline{\lambda}} \right) L_{\lambda\ell}$$

est inclus dans l'intervalle $[\lambda\ell - 3k, \lambda\ell + 3k]$ pour $\lambda > 0$ et $[0, 3k-1]$ pour le cas particulier $\lambda = 0$, d'après Lemme IV.4.2.

On suppose que

$$\begin{aligned}[0,3k-1]\cap[\ell-3k-1,\ell+3k+1]&=\emptyset,\\ [\ell-3k-1,\ell+3k+1]\cap[2\ell-3k-1,2\ell+3k+1]&=\emptyset,\\ &\dots,\\ [(d-1)\ell-3k-1,(d-1)\ell+3k+1]\cap[d\ell-3k-1,d\ell+3k+1]&=\emptyset.\end{aligned}$$

Pour que ces conditions soient atteintes, il est suffisant de supposer que $\ell > 6k$ et on a imposé ℓ pair. En choisissant $\ell = 6k+2$, on obtient une somme non-interférente pour la somme du membre de droite de (IV.4.5). Une preuve identique est réalisable pour $\ell = 6k+4$ et (IV.4.3) est démontrée, car les coefficients sont identiques dans les deux cas.

Pour la seconde partie, pour $m_1, m_2 \geq 1$ et $k_1 > k_2$ suffisamment grands, on a

$$s_F(m_1 L_{2k_1} - m_2 L_{2k_2}) = k_1 - k_2 + C(m_1, m_2),$$

où $C(m_1, m_2)$ ne dépend que de m_1 et m_2. D'après [Sto13, Lemma 3], pour k suffisamment grand, on a

$$\begin{aligned}s_F(t(k)^d) &= s_F(c_{3d}L_{6kd} - (-c_{3d-1})L_{2k(3d-1)} + \dots + c_0 L_0)\\ &= (3dk - k(3d-1)) + C(c_{3d}, c_{3d-1}) + \kappa\\ &= k + C(c_{3d}, c_{3d-1}) + \kappa,\end{aligned}$$

pour un certain κ indépendant de k et

$$\begin{aligned}s_F(t(k+1)^d) &= s_F(c_{3d}L_{6(k+1)d} - (-c_{3d-1})L_{2(k+1)(3d-1)} + \dots + c_0 L_0)\\ &= (3d(k+1) - (k+1)(3d-1)) + C(c_{3d}, c_{3d-1}) + \kappa\\ &= k + 1 + C(c_{3d}, c_{3d-1}) + \kappa.\end{aligned}$$

L'inégalité (IV.4.4) est donc démontrée. □

Nous pouvons désormais démontrer le Théorème IV.2.2 dans le cas $P(X) = X^d$.

Preuve de Théorème IV.2.2. Soit d impair. On suppose les mêmes hypothèses que celles dans Lemme IV.4.7 et on choisit $k \geq 0$ tel que

$$t(k+1) < N \leq t(k+2). \tag{IV.4.6}$$

La condition (IV.4.2) donne $n^d \ll F_{3k}$ et $n \ll F_k^{3/d}$. D'après le Lemme I.2.4, on a

$$M(\mathcal{S}_{Z,P}, N) \gg F_k^{3/d}.$$

Il reste à s'assurer que

$$t(k) \gg F_k^{3/d} + L_{6k+2}, \tag{IV.4.7}$$

car nous avons besoin que $t(k)^d$ soit un successeur d'un bloc non-interférent. Or, on a $t(k) \sim m_3 L_{6k} \sim m_3 \varphi^{6k}$ et $F_k^{3/d} + L_{6k+2} \sim L_{6k+2} \sim \varphi^{6k+2}$ pour $k \to +\infty$. De plus, par (IV.4.1), $m_3 \geq d^3(32\varphi)^d$, on a $t(k) \gg 32^d d^3 \varphi^{6k+d}$ pour $k \to +\infty$. Par

conséquent, pour k suffisamment grand, (IV.4.7) est vérifiée. Pour les mêmes raisons que précédemment, il reste à s'assurer que

$$t(k+1) \gg F_k^{3/d} + L_{6k+4}. \tag{IV.4.8}$$

Une preuve similaire de (IV.4.7) donne (IV.4.8) pour k suffisamment grand. De plus, (IV.4.6) donne

$$N \leq t(k+2) \ll L_{6(k+2)} \ll F_k^6.$$

On obtient finalement

$$M(\mathcal{S}_{Z,P}, N) \gg F_k^{3/d} \gg N^{1/(2d)},$$

et le théorème est prouvé si d est impair.

Pour d pair, nous montrons un résultat similaire à Lemme IV.4.7 et une preuve similaire est possible avec les mêmes arguments, nous omettons les détails.

La preuve du Théorème IV.2.2 est donc complète pour le cas $P(X) = X^d$. □

IV.4.3. Cas général. Soit $P(X) = \alpha_d X^d + \cdots + \alpha_0$ un polynôme tel que $P(\mathbb{N}) \subset \mathbb{N}$ et $\alpha_d = 1$. Comme précédemment, nous étudions exactement $P(n + L_k)$ pour des entiers $n, k \geq 0$.

Lemme IV.4.8. *On a pour des entiers $n, k \geq 0$,*

$$P(n + L_k) = \sum_{0 \leq \lambda \leq d} \beta_\lambda(n) L_{\lambda k}$$

avec $\beta_\lambda(n) = \sum_{\lambda \leq i \leq d} \alpha_i \sum_{j=\frac{\lambda-\overline{\lambda}}{2}}^{\lfloor \frac{i-\overline{\lambda}}{2} \rfloor} \gamma_{i,j,\lambda} n^{i-2j-\overline{\lambda}}$ *et* $\gamma_{i,j,\lambda} = \binom{i}{2j+\overline{\lambda}} \binom{2j+\overline{\lambda}}{j-\frac{\lambda-\overline{\lambda}}{2}}$.

Preuve. La preuve est similaire à celle du Lemme IV.4.4. □

Lemme IV.4.9. *Soient $n \geq C_d$ et $\lambda \geq 1$ des entiers avec C_d une constante absolue qui ne dépend que de d. Alors on a $\beta_\lambda(n) < \beta_0(n)$.*

Preuve. La preuve est similaire à celle du Lemme IV.4.5. □

Soit $\mu \geq 0$ un entier. D'après le Lemme IV.4.8, on a

$$P(n + L_{2d\mu k+2}) = \beta_0(n) + \beta_1(n) L_{2d\mu k+2} + \cdots + \beta_d(n) L_{d(2d\mu k+2)}. \tag{IV.4.9}$$

Le lemme suivant est un analogue à la première partie du Lemme IV.4.7.

Lemme IV.4.10. *Soit $k \geq 0$ un entier suffisamment grand et $r > 1$. Pour tout entier n tel que $C_d < n < F_{\mu k}$ on a*

$$s_F(P(n + L_{2d\mu k+2})) = s_F(P(n + L_{2d\mu k+2r})). \tag{IV.4.10}$$

Preuve. Pour $n < F_{\mu k}$ on a $\beta_0(n) \ll L_{d\mu k}$. Par conséquent, les termes dans le membre de droite de l'égalité (IV.4.9) sont non-interférents pour k suffisamment grand comme dans la preuve de Lemme IV.4.7, où Lemme IV.4.5 est remplacée par Lemme IV.4.9. Donc, on a

$$s_F(P(n + L_{2d\mu k})) = s_F(\beta_0(n)) + s_F(\beta_1(n)) + \cdots + s_F(\beta_d(n)).$$

Sous les mêmes hypothèses, on a pour tout $r > 1$,

$$s_F((P(n + L_{2d\mu k+2r})) = s_F(\beta_0(n)) + s_F(\beta_1(n)) + \cdots + s_F(\beta_d(n)).$$

Par conséquent, pour tout $n < F_{\mu k}$, k suffisamment grand et $r > 1$, on a

$$s_F(P(n + L_{2d\mu k})) = s_F((P(n + L_{2d\mu k+2r})).$$

Le lemme est démontré. □

Pour obtenir un analogue à la deuxième partie du Lemme IV.4.7, nous cherchons désormais un entier n de la forme L_λ pour un certain λ et $r > 1$ tels que

$$s_F(P(L_\lambda + L_{2d\mu k+2})) \neq s_F(P(L_\lambda + L_{2d\mu k+2r})). \tag{IV.4.11}$$

D'après Lemme IV.4.10, il faut que $\lambda \geq \mu k$. Notons que nous voulons prouver que $M(\mathcal{S}_{Z,P}, N) \gg N^{1/(2d)}$ pour obtenir la même borne que dans le cas monomial. Le lemme suivant donne des conditions suffisantes sur la taille de λ et r pour obtenir une telle borne.

Lemme IV.4.11. *Si $\lambda \leq 2d\mu k$ et r est constant, alors on a $M(\mathcal{S}_{Z,P}, N) \gg N^{1/(2d)}$.*

Preuve. Si on a λ tel que (IV.4.11) est vérifiée, on montre de la même manière que précédemment que

$$M(\mathcal{S}_{Z,P}, N) \geq F_{\mu k},$$

où k est choisi tel que

$$L_\lambda + L_{2d\mu k+2r} < N \leq L_\lambda + L_{2d\mu(k+1)+2r}.$$

Ceci implique donc

$$\begin{aligned} N &\ll L_\lambda + L_{2d\mu k+2r} \\ &\ll F_{\mu k}^{\lambda/\mu k} + F_{\mu k}^{2d+2r/\mu k}. \end{aligned}$$

Si $\lambda \leq 2d\mu k$ et r est constant, on a $N \ll F_{\mu k}^{2d}$ et $M(\mathcal{S}_{Z,P}, N) \gg N^{1/(2d)}$. Notons que μ ne dépend pas de k, ce qui était requis pour obtenir un tel résultat. □

Nous cherchons désormais un couple (λ, r) tel que $\mu k \leq \lambda \leq 2d\mu k$, r constant et (IV.4.11) est vérifiée. Nous pouvons désormais prouver le Théorème IV.2.2.

Preuve du Théorème IV.2.2. On étudie désormais les interférences dans

$$P(L_\lambda + L_{2d\mu k+2}) = \beta_0(\lambda) + \beta_1(\lambda)L_{2d\mu k+2} + \cdots + \beta_d(\lambda)L_{d(2d\mu k+2)}.$$

Pour chaque $\beta_i(\lambda)L_{i(2d\mu k+2)}$ on localise le chiffre de poids faible et le chiffre de poids fort. Les interférences suivantes sont possibles :

$$\begin{aligned} L_{\lambda d} + (-1)^{(d-1)\lambda} L_{2d\mu k+2-(d-1)\lambda} \\ L_{2d\mu k+2+(d-1)\lambda} + (-1)^{(d-2)\lambda} L_{2(2d\mu k+2)-(d-2)\lambda} \\ \cdots \\ L_{(d-1)(2d\mu k+2)+\lambda} + L_{d(2d\mu k+2)} \end{aligned}$$

car $\alpha_d = 1$. On choisit λ pair comme précédemment. Par conséquent, on a pour la première interférence $L_{\lambda d} + L_{2d\mu k-(d-1)\lambda+2}$. Par Lemme IV.4.1 on a pour k suffisamment grand

$$\begin{aligned} s_F(L_{\lambda d} + L_{2d\mu k-(d-1)\lambda+2}) = 1 &\Leftrightarrow 2d\mu k - (d-1)\lambda + 2 = \lambda d + 2 \\ &\Leftrightarrow \lambda = \frac{2d\mu k}{2d-1}. \end{aligned} \tag{IV.4.12}$$

Ainsi il suffit de prendre $\mu = 2d-1$ et $\lambda = 2dk$. Par conséquent, toutes les autres interférences possibles n'existent pas par choix de λ et μ. Nous devons finalement vérifier si λ satisfait toutes les conditions : λ est pair et $(2d-1)k \leq \lambda \leq 2d(2d-1)k$. De plus on peut prendre $r = 2$ par exemple et pour tous les blocs de chiffres qui apparaissent n'interfèrent pas. Pour résumer, nous avons prouvé que $s_F\left(P\left(L_\lambda + L_{2d(2d-1)k+2}\right)\right) \neq s_F\left(P\left(L_\lambda + L_{2d(2d-1)k+4}\right)\right)$. Ainsi, nous avons prouvé le cas général du Théorème IV.2.2. □

Remarque IV.4.1. Dans le cas général, l'approche est assez différente du cas monomial et du cas des sous-suites polynomiales de Thue–Morse, voir le Chapitre II. En effet, nous avons amélioré la méthode pour que λ et r puissent dépendre de ℓ tout s'assurant que cela n'impacte pas l'estimation finale.

Remarque IV.4.2. Comme pour les résultats obtenus dans le cadre de la suite de Thue–Morse, nous discuterons de l'optimalité des bornes obtenues dans les Théorèmes IV.2.1 et IV.2.2 dans le Chapitre V.

IV.5. Ouvertures

IV.5.1. Sommes des chiffres modulo k. La suite $\mathcal{S}_{Z,k} = (s_F(n) \bmod k)_{n\geq 0}$ est aussi une suite morphique sur l'alphabet Σ_k. En effet, nous définissons les morphismes suivants

$$f : \begin{cases} a_1 \mapsto a_1 a_2 \\ a_2 \mapsto a_3 \\ a_3 \mapsto a_3 a_4 \\ a_4 \mapsto a_5 \\ \vdots \\ a_{2k-1} \mapsto a_{2k-1} a_{2k} \\ a_{2k} \mapsto a_1 \end{cases} \quad \text{et} \quad g : \begin{cases} a_1, a_{2k} \mapsto 0 \\ a_2, a_3 \mapsto 1 \\ \vdots \\ a_{2k-2}, a_{2k-1} \mapsto k-1 \end{cases}$$

Et nous avons $\mathcal{S}_{Z,k} = g \circ f^\omega(a_1)$. Ce morphisme est obtenu par extension du morphisme de la suite de Fibonacci–Thue–Morse. Nous pensons que les Théorèmes IV.2.1 et IV.2.2 peuvent se généraliser pour cette suite en utilisant les mêmes méthodes.

IV.5.2. Système de numération d'Ostrowski. Nous pensons également que les Théorèmes IV.2.1 et IV.2.2 peuvent se généraliser à d'autres systèmes de numérations. En effet, le système de numération de Zeckendorf est un système de numération d'Ostrowski pour le nombre d'or φ, voir le Paragraphe III.2.

Soit un irrationnel quadratique $\alpha > 1$ tel que sa fraction continue soit de la forme $\alpha = [1, a, a, \ldots]$. Dans ce cas particulier, la suite des dénominateurs vérifie la relation de récurrence $q_{n+2} = \frac{1}{a}\, q_{n+1} + q_n$, une relation de récurrence similaire à celle de la suite de Fibonacci. Notons γ le zéro du polynôme $x^2 - \frac{1}{a}x - 1$ tel que son conjugué de Galois $\overline{\gamma}$ vérifie $|\overline{\gamma}| < |\gamma|$. Par conséquent, nous avons $q_n = \frac{1}{\sqrt{a^{-2}+4}}(\gamma^n - \overline{\gamma}^n)$, un analogue à la formule de Binet pour les nombres de Fibonacci. Nous pouvons définir un analogue pour les nombres de Lucas par $L'_n = \gamma^n + \overline{\gamma}^n$ pour tout $n \geq 0$. Les nombres de Lucas L'_n vérifient la propriété $L'_n = q_{n+1} + q_{n-1}$ et un analogue à la formule du Lemme IV.4.2 est vraie. Ainsi, la fonction somme des chiffres dans ce système de numération vérifie des propriétés similaires à la fonction s_F.

Pour le cas général d'un système d'Ostrowski, il semble que d'autres arguments soient nécessaires. Le cas d'un nombre de Pisot semble une généralisation naturelle.

IV.5.3. Suites de motifs en base de Zeckendorf. Nous proposons également de généraliser la construction de la suite de Fibonacci–Thue–Morse, en s'inspirant des suites de motifs d'ordre k, voir Définition I.3.8. Rappelons que la suite de Thue–Morse est la suite de motifs d'ordre 1.

Définition IV.5.1. Soit $k \geq 1$, on note $s_{F,k} : \mathbb{N} \to \mathbb{N}$ la fonction définie par

$$s_{F,k}(n) = \sum_{0 \leq i \leq \ell} \varepsilon_{i+2(k-1)} \cdots \varepsilon_{i+2}\varepsilon_i,$$

où $(n)_F = \varepsilon_\ell \cdots \varepsilon_0$.

Par conséquent, $s_{F,k}(n)$ est le nombre d'occurrences du mot $(\mathtt{10})^k$. Notons qu'il n'est pas possible de considérer le mot $\mathtt{1}^k$ en base de Zeckendorf et que l'équivalent de ce mot est $(\mathtt{10})^k$.

Définition IV.5.2. La suite $P_{F,k} = (p_{F,k}(n))_{n \geq 0}$ est appelée *suite de motifs de Fibonacci d'ordre k*, où $p_{F,k}(n) = s_{F,k}(n) \bmod 2$.

La suite de Fibonacci–Thue–Morse est la suite de motifs de Fibonacci d'ordre 1, et nous définissons la suite Fibonacci–Rudin–Shapiro $\mathcal{R}_F$ est définie comme la suite de motifs de Fibonacci d'ordre 2.

La suite $\mathcal{R}_F$ est une suite morphique. En effet nous définissons le morphisme et le codage suivants :

$$f : \begin{cases} a \mapsto ab \\ b \mapsto c \\ c \mapsto ad \\ d \mapsto e \\ e \mapsto fb \\ f \mapsto fd \end{cases} \quad \text{et} \quad g : \begin{cases} a, b, c \mapsto 0 \\ d, e, f \mapsto 1. \end{cases}$$

Et nous avons $\mathcal{R}_F = 000010010000100000111000\cdots$. La suite $\mathcal{R}_F$ est également Fibonacci-automatique. L'automate est le suivant :

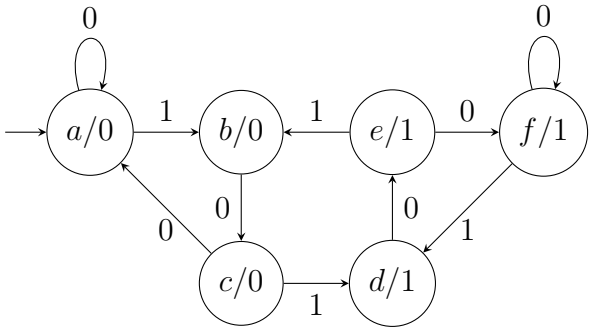


FIGURE IV.5. Automate pour la suite de Fibonacci–Rudin–Shapiro.

En effet, lors du parcours d'un mot w, si le mot 101 est lu, alors il correspond à l'un des chemins suivants :

- $a \to b \to c \to d$.
- $f \to d \to e \to b$.
- $c \to d \to e \to b$.
- $e \to b \to c \to d$.

Dans le premier chemin, l'apparition de 101 est précédée de 00 dans w, et le nombre de 101 est augmenté de 1. Comme $a \to 0$ et $d \to 1$, nous retrouvons bien la bonne augmentation d'apparition de 101 dans w. Le deuxième chemin est similaire. Dans le troisième chemin, l'apparition de 101 est précédée de 10, et le nombre de 101 est donc augmenté de 2. Comme $c \to 0$ et $b \to 0$, nous retrouvons bien la bonne augmentation d'apparition de 101 dans w. Le quatrième chemin est similaire. Alors l'automate engendre bien la suite $\mathcal{R}_F$.

Nous pensons qu'il est possible de généraliser les Théorèmes IV.2.1 et IV.2.2 pour la suite $P_{F,k}$, en combinant à la fois les méthodes de ce chapitre et du Chapitre II.

CHAPITRE V

Graphe Acyclique Orienté de Mot (DAWG)

Dans ce chapitre, nous nous intéressons au calcul de la complexité d'ordre maximal, $M(\mathcal{S}, N)$. Pour cela nous introduisons le DAWG, défini par Blumer et al. [BBH$^+$85], et détaillons les moyens mis en œuvre pour implanter ce calcul en machine. Pour terminer, nous rappelons les conjectures énoncées dans [JPS21].

V.1. Définition et propriétés

Soit Σ un alphabet fini. Dans tout ce paragraphe, nous reprenons les définitions, notations et propriétés de Blumer et al. [BBH$^+$85].

Définition V.1.1. Soit $w = a_1 \cdots a_n \in \Sigma^*$. Soit $y \in \Sigma^* \setminus \{\varepsilon\}$. L'ensemble $E_w(y)$ des *positions finales* de y dans w est l'ensemble $\{i : y = a_{i-|y|+1} \cdots a_i\}$. Deux mots $x, y \in \Sigma^*$ sont dits *suffixe-équivalents* dans w si $E_w(x) = E_w(y)$.

Étant donné $w \in \Sigma^*$ et $y \in \Sigma^*$, l'ensemble $E_w(y)$ est non vide si et seulement si y est un facteur de w. Pour deux mots x, y suffixe-équivalents, on note $x \equiv_w y$. Nous donnons une liste de propriétés élémentaires de la relation $\equiv_w$.

Proposition V.1.1. Soit $w = a_1 \cdots a_n \in \Sigma^*$.

(1) La relation $\equiv_w$ est une relation d'équivalence. Pour $x \in \Sigma^*$, on note $[x]_w$ sa classe d'équivalence.
(2) Pour tous $x, y, z \in \Sigma^*$, $x \equiv_w y \implies xz \equiv_w yz$. La relation $\equiv_w$ est *invariante à droite*.
(3) Si deux facteurs x, y de w sont suffixe-équivalents, alors x est suffixe de y ou y est suffixe de x.
(4) Deux facteurs xy et y de w sont suffixe-équivalents si et seulement si toute occurrence de y est immédiatement précédée d'une occurrence de x.
(5) Un facteur propre x de w est le plus long élément de sa classe $[x]_w$ si et seulement si x est un préfixe de w ou x est un facteur spécial à gauche de w.

Preuve. (1) La réflexivité, la symétrie et la transitivité sont évidentes.
(2) Soient x, y deux mots suffixe-équivalents dans w, soit $z \in \Sigma^*$ de longueur $\ell \leq 0$. On a $E_w(xz) = E_w(x) + \{\ell\} = \{i + \ell : i \in E_w(x)\} = \{i + \ell : i \in E_w(y)\} = E_w(y) + \{\ell\} = E_w(yz)$.
(3) On peut supposer $|x| > |y|$ sans perte de généralité, car si $|x| = |y|$, on a $x = y$ et donc le résultat. Alors on note $x = b_1 \cdots b_r$, $y = c_1 \cdots c_s$, $r > s$ et montrons que y est un suffixe de x, c'est-à-dire qu'il existe un mot z tel que $x = zy$. Soit $0 \leq i \leq n$ un élément de $E_w(x)$, on a alors $a_{i-r+1} \cdots a_i = b_1 \cdots b_r$ et $a_{i-s+1} \cdots a_i = c_1 \cdots c_s$.

Alors on a

$$b_1 \cdots b_r = a_{i-r+1} \cdots a_{i-s} c_1 \cdots c_s,$$

et donc le résultat avec $z = a_{i-r+1} \cdots a_{i-s}$.

(4) Chaque élément de $E_w(y)$ correspond à une occurrence de y dans w et chaque élément de $E_w(xy)$ correspond à une occurrence de xy dans w.
Supposons que $E_w(y) = E_w(xy)$ et soit i la position finale d'une occurrence de y dans w. Par hypothèse, $i \in E_w(xy)$ et donc cette occurrence de y est précédée directement de x.
Supposons que chaque occurrence de y est précédée de x. On a trivialement $E_w(xy) \in E_w(y)$. Soit $i \in E_w(y)$, $y = b_1 \cdots b_r$ et on a

$$a_{i-r+1} \cdots a_i = b_1 \cdots b_r.$$

On écrit $x = c_1 \cdots c_s$ et on a par hypothèse $a_{i-r-s+1} \cdots a_{i-r} a_{i-r+1} \cdots a_i = c_1 \cdots c_s b_1 \cdots b_r$. On en déduit que $i \in E_w(xy)$.

(5) Soit $y \in [x]_w$ tel que $|y| > |x|$. D'après le point (3) on a $y = zx$ pour un certain z. Alors x et zx sont suffixe-équivalents et d'après le point (4), x n'est pas un facteur spécial à gauche ni un préfixe car chacune de ces occurrences est précédée de z.
On suppose que x est l'élément le plus long de sa classe d'équivalence. Supposons que x n'est pas un préfixe. Si x n'est pas un facteur spécial à gauche, alors il existe une unique lettre a telle que ax soit suffixe-équivalent à x et cela contredit donc l'hypothèse.

□

Définition V.1.2. Soit $w \in \Sigma^*$. Le *Graphe Orienté Acyclique de Mot*, ou *Directed Acyclic Word Graph* (DAWG) en anglais, de w est un graphe $D_w = (V, E)$ où

- V est l'ensemble des classes d'équivalence pour la relation $\equiv_w$ des facteurs de w.
- $E = \{[x]_w \xrightarrow{a} [xa]_w : x, xa \in \mathrm{Sub}(w)\}$.

Proposition V.1.2 ([BBH+85, Theorem 1.7]). Soit $w \in \Sigma^*$ tel que $|w| > 2$. Alors :

- D_w possède au plus $2|w| - 1$ sommets.
- D_w possède exactement $2|w| - 1$ sommets si et seulement si $w = ab^n$ pour $a, b \in \Sigma^*$ et $a \neq b$.
- D_w possède au plus $3|w| - 4$ arêtes.

Étant donné un mot $w \in \Sigma^*$, le graphe D_w reconnait le langage de w, voir [BBH+85, Lemma 1.2]. Même si le graphe D_w n'est pas le plus petit graphe possédant cette propriété, le nombre de sommets et d'arêtes du plus petit graphe qui reconnait le langage de w est proche de celui de D_w, voir [BBH+85, Theorem 3.1]. L'avantage du DAWG réside dans le fait que Blumer et al. [BBH+85] fournissent un algorithme linéaire en temps et en espace pour la construction de ce graphe.

Exemple V.1.1. Reprenons l'exemple $w = \mathtt{0110010}$, étudié dans le Paragraphe I.2.4. Rappelons que l'ensemble des facteurs de w est l'ensemble

$$\mathrm{Sub}(w) = \{\mathtt{0}, \mathtt{1};\ \mathtt{01}, \mathtt{11}, \mathtt{10}, \mathtt{00};\ \mathtt{011}, \mathtt{110}, \mathtt{100}, \mathtt{001}, \mathtt{010};\ \mathtt{0110}, \mathtt{1100}, \mathtt{1001}, \mathtt{0010};$$
$$\mathtt{01100}, \mathtt{11001}, \mathtt{10010};\ \mathtt{011001}, \mathtt{110010};\ \mathtt{0110010}\}.$$

Dans chaque sommet, nous notons uniquement l'élément le plus long de la classe d'équivalence. Le DAWG de w est le graphe suivant :

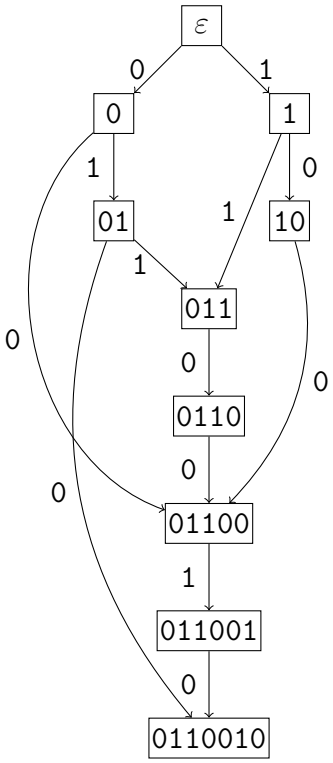


FIGURE V.1. DAWG pour $w = \texttt{0110010}$.

Les arêtes d'un DAWG sont divisées en deux catégories distinctes. Une arête est *primaire* si elle appartient à un chemin primaire, qui est le chemin le plus long entre la source (le sommet de la classe de ε) et le sommet, sinon elle est *secondaire*. On définit la profondeur d'un sommet comme la longueur d'un chemin composé d'arêtes primaires entre la source et ce sommet. Dans la suite on notera $d(\cdot)$ la profondeur d'un sommet.

Exemple V.1.2. En reprenant l'exemple précédent, on a les profondeurs suivantes

$$\begin{aligned} &d(\varepsilon) = 0, \quad d(\texttt{0}) = 1, \quad d(\texttt{1}) = 1, \\ &d(\texttt{01}) = 2, \quad d(\texttt{10}) = 2, \quad d(\texttt{011}) = 3, \\ &d(\texttt{0110}) = 4, \quad d(\texttt{01100}) = 5, \\ &d(\texttt{011001}) = 6, \quad d(\texttt{0110010}) = 7. \end{aligned}$$

Soit $B(w)$ l'ensemble des sommets de degré sortant supérieur ou égal à 2, appelé ensemble des *sommets à branches* de w. On définit également la profondeur maximale sur tous les sommets à branches $\hat{d}(w)$ comme

$$\hat{d}(w) = \max_{a \in B(w)} d(a).$$

Exemple V.1.3. Dans l'exemple précédent, on a $\hat{d}(w) = 2$ car le sommet `01` est le sommet à branche le plus profond de profondeur 2.

V.2. Application à la complexité d'ordre maximal

D'après le Lemme I.2.4, le profil de complexité d'ordre maximal d'une suite se calcule avec les facteurs spéciaux à droite du mot associé. Un facteur spécial à droite de w est un sommet du DAWG avec au moins deux flèches sortantes. C'est pourquoi nous pouvons déterminer le profil de complexité d'ordre maximal grâce au DAWG. Cette correspondance entre complexité d'ordre maximal et DAWG a été faite par Jansen [Jan89]. Nous la reformulons de la manière suivante.

Théorème V.2.1. *Soit $\mathcal{S} = (s_n)_{n\geq 0}$ une suite sur Σ. Pour $N \geq 1$, $w_N = s_0 s_1 \cdots s_{N-1}$ le préfixe de longueur N du mot infini associé à $\mathcal{S}$. La complexité d'ordre maximal de $\mathcal{S}$ satisfait*

$$M(\mathcal{S}, N) = \begin{cases} 0, & \text{si } B(w_N) = \emptyset, \\ \hat{d}(w_N) + 1, & \text{sinon.} \end{cases}$$

Blumer et al. [BBE+83] propose un algorithme qui crée le DAWG d'un mot.

Remarque V.2.1. Pour trouver un facteur spécial à droite de longueur k, nous pouvons utiliser le graphe de Rauzy d'ordre k, voir Paragraphe I.2.1. Cependant, pour la complexité d'ordre maximal, nous sommes plutôt intéressés par chercher un facteur spécial à droite de grande longueur. C'est pourquoi, le DAWG est plus adapté pour ce problème, car la construction d'un seul graphe sera suffisante.

V.3. Estimations et conjectures

Pour une implémentation en `C++` de l'algorithme de Blumer et al., voir :

`https://gitlab.inria.fr/ppopoli/max-order-comp`.

Dans cette implémentation, une arête nécessite 6 octets et un sommet nécessite 12 octets. Soit w un mot de longueur L. La totalité du DAWG de w est stocké en mémoire vive et aucune arête ni aucun sommet ne peuvent être supprimés du graphe avant la fin du parcours complet de w. Par conséquent, d'après la Proposition V.1.2, le stockage en mémoire du DAWG de w nécessite

$$6(2L - 1) + 12(3L - 4) = 48L - 54 \text{ octets.}$$

Pour $L = 100\ 000\ 000$, la mémoire vive théorique nécessaire dans le pire des cas du DAWG de w est donc de 480 Go (447 GiB). C'est pourquoi, nous avons utilisé un nœud du cluster *yeti* de Grid'5000, dont les caractéristiques sont les suivantes : 4x Intel Xeon Gold 6130 CPU et 16 cœurs / CPU, avec 768 GiB de mémoire vive, voir la page web `https://www.grid5000.fr/w/Grenoble:Hardware#yeti`. Dans la suite de ce paragraphe, nous donnons les simulations réalisées de ce cluster.

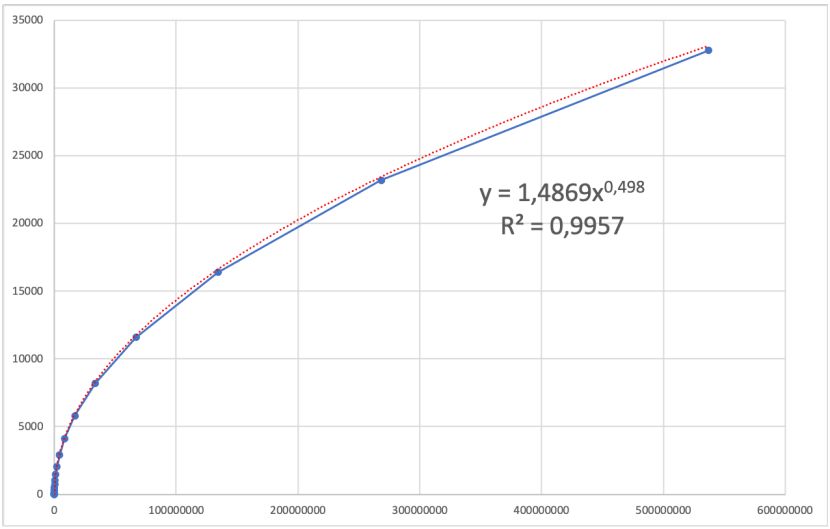


FIGURE V.2. Complexité d'ordre maximal de la suite $\mathcal{T}_2$.

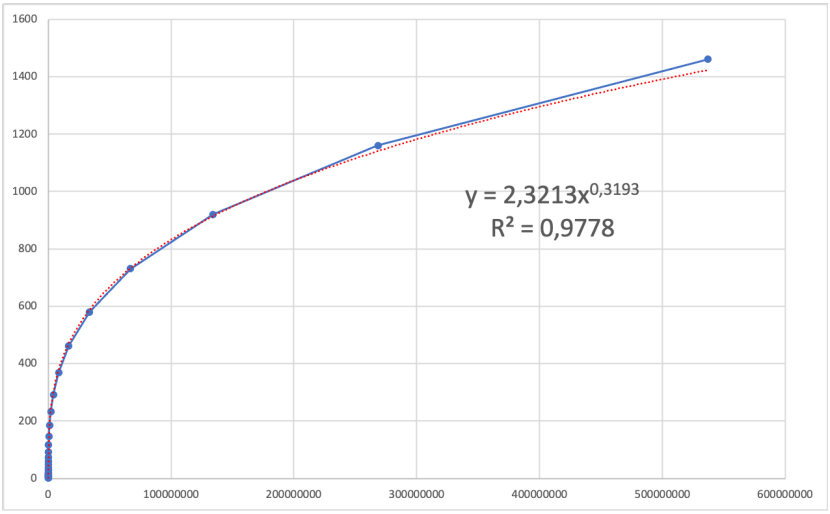


FIGURE V.3. Complexité d'ordre maximal de la suite $\mathcal{T}_3$.

Les Figures V.2 et V.3 montrent que la croissance du profil de complexité d'ordre maximal des suites $\mathcal{T}_2 = (t(n^2))_{n\geq 0}$ et $\mathcal{T}_3 = (t(n^3))_{n\geq 0}$ sont proches de $N^{1/2}$ et $N^{1/3}$, respectivement. De plus, nous avons également calculé la complexité d'ordre maximal sur d'autres sous-suites polynomiales et les allures des courbes sont similaires, elles ne dépendent que du degré du polynôme. C'est pourquoi nous formulons la conjecture suivante, présente dans [JPS21]. Rappelons que pour un polynôme $P \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$, on note $\mathcal{T}_P = (t(P(n)))_{n\geq 0}$.

Conjecture V.3.1 ([JPS21])**.** *Soit $P \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$ de degré d. La complexité d'ordre maximal au rang N de la suite $\mathcal{T}_P$ satisfait $M(\mathcal{T}_P, N) \asymp N^{1/d}$.*

Une preuve de cette conjecture impliquerait que la borne inférieure du Théorème II.3.1 est optimale.

Remarque V.3.1. Prouver une borne supérieure non triviale pour la complexité d'ordre maximal est plus difficile à obtenir qu'une borne inférieure. Dans le contexte de la conjecture précédente, nous devrions montrer qu'avec N termes, tout facteur spécial à droite dans ces N termes est de longueur $\ll N^{1/d}$.

Le profil de complexité d'ordre maximal est une fonction en escaliers, car croissante et à valeurs entières. À chaque fois que $M(\mathcal{S}, N)$ a une valeur différente de $M(\mathcal{S}, N-1)$, on dit que N est un *palier*. C'est pourquoi le quotient entre la complexité d'ordre maximal et sa vitesse conjecturée évolue en "dents de scie". Nous avons alors les figures suivantes, où $\mathcal{T}_2 = (t(n^2))_{n\geq 0}$ et $\mathcal{T}_3 = (t(n^3))_{n\geq 0}$.

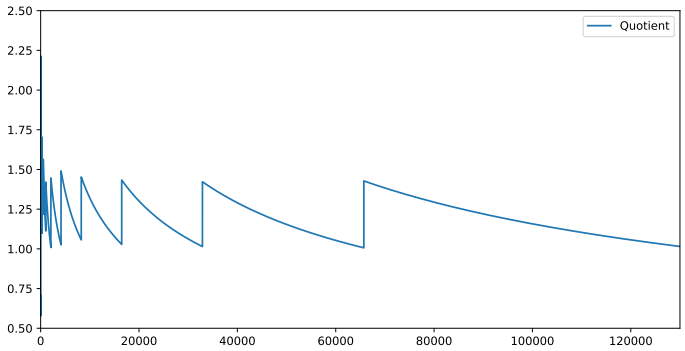


FIGURE V.4. Le quotient $M(\mathcal{T}_2, N)/N^{1/2}$ pour $2 \leq N \leq 130000$.

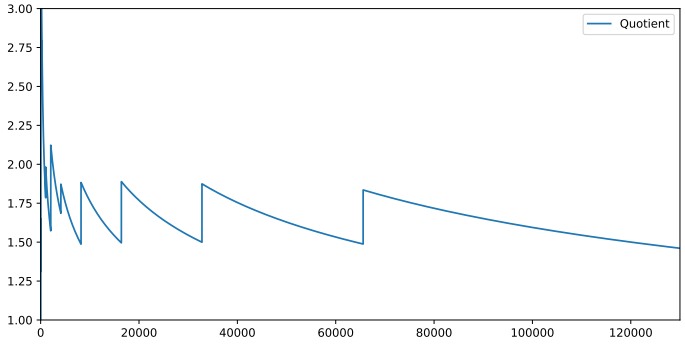


FIGURE V.5. Le quotient $M(\mathcal{T}_3, N)/N^{1/3}$ pour $2 \leq N \leq 130000$.

Les figures précédentes permettent de donner une valeur approximative des constantes de la Conjecture V.3.1 dans le cas $P(X) = X^2$ et le cas $P(X) = X^3$.

Nous avons ensuite calculé le profil de complexité d'ordre maximal pour la suite $\mathcal{S}_Z$ et de $\mathcal{S}_Z$ le long des sous-suites polynomiales. Notons que cette complexité est algorithmiquement plus complexe à réaliser que celle pour la suite de Thue–Morse.

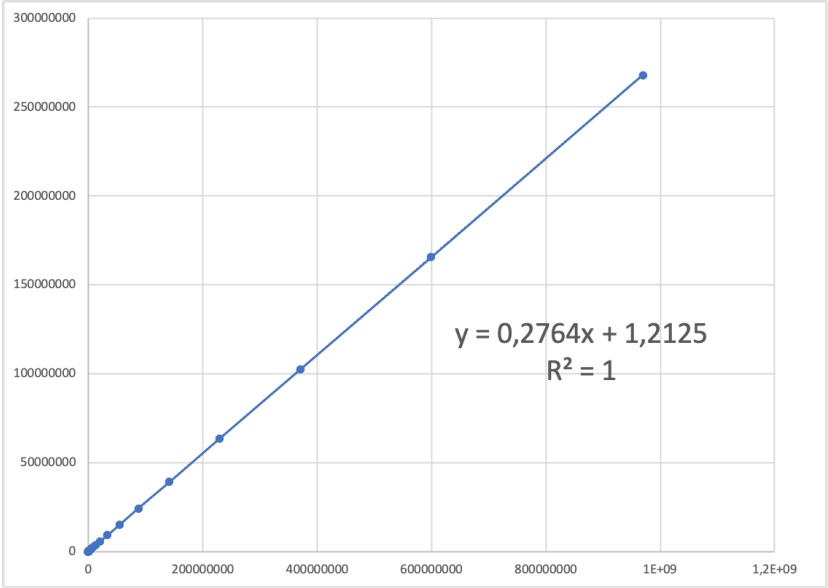


FIGURE V.6. Complexité d'ordre maximal de la suite $\mathcal{S}_Z$.

À partir de la figure précédente et de l'identité $\frac{1}{1+\varphi^2} = 0.27639\cdots$, nous formulons la conjecture suivante, présente dans [JPS21].

Conjecture V.3.2 ([JPS21]). *La complexité d'ordre maximal au rang N de $\mathcal{S}_Z$ satisfait*

$$M(\mathcal{S}_Z, N) \sim \frac{1}{1+\varphi^2} N.$$

Shallit [Sha22b] a réfuté cette conjecture et démontre le résultat suivant, à l'aide de `Walnut`.

Théorème V.3.1 ([Sha22b]). *La complexité d'ordre maximal de $\mathcal{S}_Z$ satisfait*

$$\liminf_{N\to+\infty} \frac{M(S_Z, N)}{N} = \frac{1}{\varphi+\varphi^3}, \qquad \limsup_{N\to+\infty} \frac{M(S_Z, N)}{N} = \frac{1}{1+\varphi^2}.$$

Une preuve par `Walnut` de ce théorème est possible car la propriété de facteur spécial à droite est traduisible en logique du premier ordre et la suite de Fibonacci–Thue–Morse est Fibonacci-automatique. Pour plus de détails sur le logiciel de preuve automatique `Walnut`, voir [Mou16] et [Sha22a]. Notons que dans le Théorème IV.2.1, nous avons trouvé la limite inférieure de $M(\mathcal{S}_Z, N)$, ce qui justifie que notre méthode est pertinente pour la recherche de bornes inférieures.

On note $c_1 = \frac{1}{1+\varphi^2} = 0.27639\cdots$ et $c_2 = \frac{1}{\varphi+\varphi^3} = 0.17082\cdots$. Ces deux valeurs apparaissent distinctement dans la figure suivante.

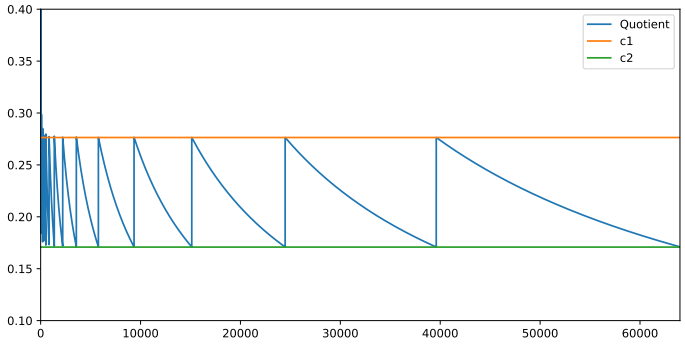


FIGURE V.7. Le quotient $M(\mathcal{S}_Z, N)/N$ pour $2 \leq N \leq 64000$.

Nous avons également calculé la complexité d'ordre maximal de la fonction somme des chiffres modulo 2 en base Zeckendorf le long des carrés et des cubes. Nous avons les résultats expérimentaux suivants, pour $N \leq 800\ 000\ 000$.

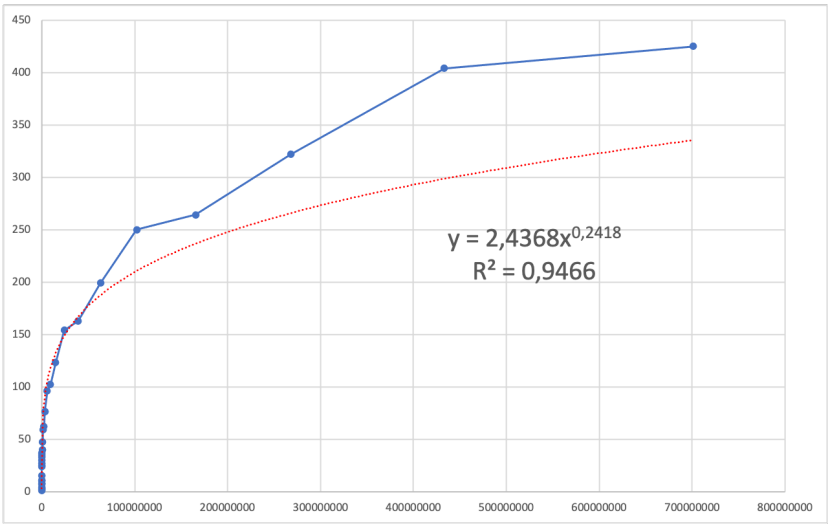


FIGURE V.8. Complexité d'ordre maximal de $\mathcal{S}_Z$ le long des carrés.

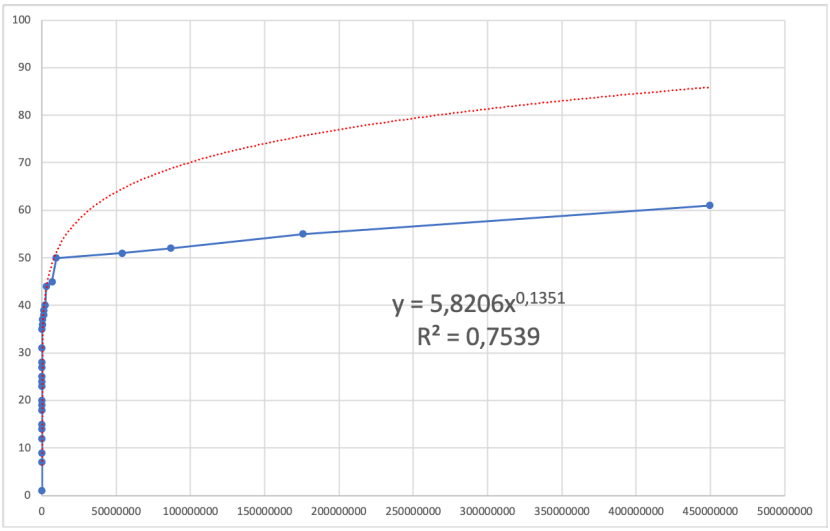


FIGURE V.9. Complexité d'ordre maximal de $\mathcal{S}_Z$ le long des cubes.

Comme pour les sous-suites polynomiales de la suite de Thue–Morse, nous formulons la conjecture analogue suivante, présente dans [JPS21]. Rappelons que pour $P \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$, on note $\mathcal{S}_{Z,P} = (s_F(P(n)) \bmod 2)_{n\geq 0}$.

Conjecture V.3.3 ([JPS21]). *Soit $P \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$ de degré $d \geq 2$.La complexité d'ordre maximal au rang N de la suite $\mathcal{S}_{Z,P}$ satisfait $M(\mathcal{S}_{Z,P}, N) \asymp N^{1/(2d)}$.*

Cette conjecture est motivée par les résultats expérimentaux, comme la Figure V.8. Comme précédemment, une preuve de cette conjecture implique que la borne inférieure du Théorème IV.2.2 est optimale. Remarquons qu'il y a ici une disparité plus importante entre le cas linéaire et quadratique pour la complexité d'ordre maximal de $\mathcal{S}_Z$.

Remarque V.3.2. La Figure V.9 ne met pas en évidence les Conjectures V.3.3 et V.3.4 (voir page 116). Avec notre programme et notre ordinateur, il n'était pas possible de calculer la complexité d'ordre maximal au-delà des 10^9 premiers termes. Néanmoins, nous pensons que s'il est possible de calculer des termes supplémentaires, les deux conjectures devraient apparaître plus distinctement.

Soit $\mathcal{S}_{Z,2} = (s_F(n^2) \bmod 2)_{n\geq 0}$. Le quotient de $M(\mathcal{S}_{Z,2}, N)$ et de sa vitesse conjecturée est donné dans la figure suivante.

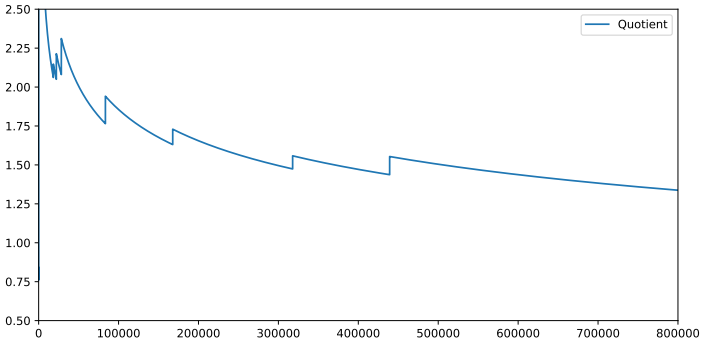


FIGURE V.10. Le quotient $M(\mathcal{S}_{Z,2}, N)/N^{1/4}$ pour $2 \leq N \leq 800000$.

Remarque V.3.3. La complexité d'ordre maximal au rang N de la suite $\mathcal{S}_{Z,P}$ ne peut pas être déterminée à l'aide de `Walnut` puisque ces suites ne sont plus automatiques ou Ostrowski-automatiques (engendrée par un automate défini sur un système de numération d'Ostrowski).

Nous avons également découvert un phénomène inattendu sur les *paliers* de variations du profil de complexité d'ordre maximal pour ces suites. Le ratio des paliers successifs est le ratio N_1/N_2 pour deux paliers N_1, N_2 tels qu'il n'y ait pas de $N \geq 1$ avec $M(\mathcal{S}, N_1) < M(\mathcal{S}, N) < M(\mathcal{S}, N_2)$. Nous avons observé que le ratio des paliers semble converger vers une valeur qui dépend de la base de la suite choisie.

N	Ratio des paliers de la Figure V.2.
2	-
4	2
10	2,5
23	2,3
37	1,608695652
52	1,405405405
73	1,403846154
138	1,890410959
270	1,956521739
530	1,962962963
1048	1,977358491
2082	1,986641221
4146	1,991354467
8258	1,991799325
16478	1,995398402
32898	1,996480155
65720	1,997689829
131330	1,998326233
262510	1,998857839
524802	1,999169555
1049302	1,999424545

N	Ratio des paliers de la Figure V.6.
2	-
5	2,5
12	2,4
19	1,583333333
30	1,578947368
48	1,6
77	1,604166667
124	1,61038961
200	1,612903226
323	1,615
522	1,616099071
844	1,616858238
1365	1,617298578
2208	1,617582418
3572	1,617753623
5779	1,617861142
9350	1,617926977
15128	1,617967914
24477	1,617993125
39604	1,618008743
64080	1,618018382

N	Ratio des paliers de la Figure V.8.
2	-
4	2
6	1,5
8	1,33333333
18	2,25
24	1,33333333
42	1,75
59	1,40476191
171	2,89830509
448	2,61988304
18367	40,9977679
22118	1,20422497
28371	1,28271091
83517	2,94374538
167773	2,00884850
317825	1,89437514
439221	1,38195863
832054	1,89438574
1346308	1,61805364
2178338	1,61800866
3524634	1,61803816

N	Ratio des paliers de la Figure V.9.
2	-
9	4,5
37	4,11111111
59	1,59459460
67	1,13559322
167	2,49253731
273	1,63473054
1677	6,14285714
1960	1,16875373
2178	1,11122449
6378	2,92837466
11095	1,73957353
17046	1,53636773
20398	1,19664437
40879	2,00406903
57958	1,41779398
510998	8,81669485
567415	1,11040552
1356065	2,38989981
1390403	1,02532180
2264908	1,62895794

La conjecture suivante porte sur la convergence du ratio des paliers, motivée par les tables précédentes, présente dans [JPS21].

Conjecture V.3.4 ([JPS21]**).** *Le ratio des paliers successifs pour la complexité d'ordre maximal relié à la suite de Thue–Morse et ses sous-suites polynomiales (resp. la suite $\mathcal{S}_Z$ et ses sous-suites polynomiales) converge vers* 2 *(resp. vers φ).*

V.4. Problèmes ouverts

Tout au long de ces premiers chapitres, nous avons fréquemment parlé de sous-suites polynomiales. Il est alors intéressant de se demander si nos méthodes peuvent s'utiliser pour d'autres sous-suites.

V.4.1. Raréfaction le long des nombres premiers. En vertu de l'article de Gelfond [Gel68], voir le Paragraphe II.2, il est naturel de s'intéresser à la suite de Thue–Morse le long des nombres premiers, la suite $\mathcal{T}_{\mathbb{P}} = (t(p))_{p\in\mathbb{P}}$ où p parcourt l'ensemble des nombres premiers, noté $\mathbb{P}$. À ce jour, aucun résultat n'est connu sur la complexité d'ordre maximal de $\mathcal{T}_{\mathbb{P}}$, ni même sur sa complexité linéaire. Nous pouvons nous attendre à un comportement assez aléatoire pour $\mathcal{T}_{\mathbb{P}}$, voir la figure suivante.

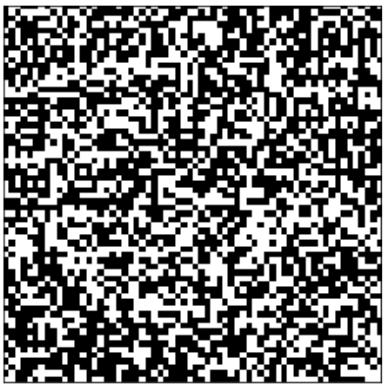


FIGURE V.11. Les 64×64 premiers termes de $\mathcal{T}_{\mathbb{P}}$.

Nous avons obtenu les estimations suivantes pour la complexité linéaire de $\mathcal{T}_{\mathbb{P}}$. Plus précisément, nous donnons le graphique de $L(\mathcal{T}_{\mathbb{P}}, N) - \frac{N}{2}$, voir le Paragraphe I.2.3 pour plus de détails sur ce choix.

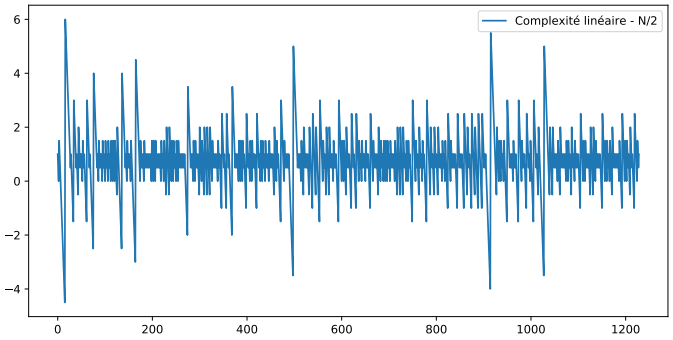


FIGURE V.12. Complexité linéaire de $\mathcal{T}_{\mathbb{P}}$.

Nous avons également calculé la complexité d'ordre maximal de $\mathcal{T}_{\mathbb{P}}$ et le quotient entre $M(\mathcal{T}_{\mathbb{P}}, N)$ et sa vitesse supposée $\log_2(x)$.

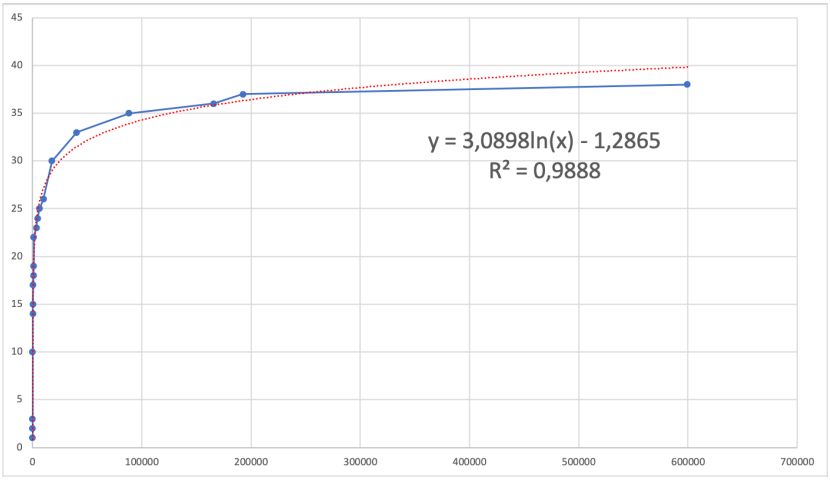


FIGURE V.13. Complexité d'ordre maximal de $\mathcal{T}_{\mathbb{P}}$.

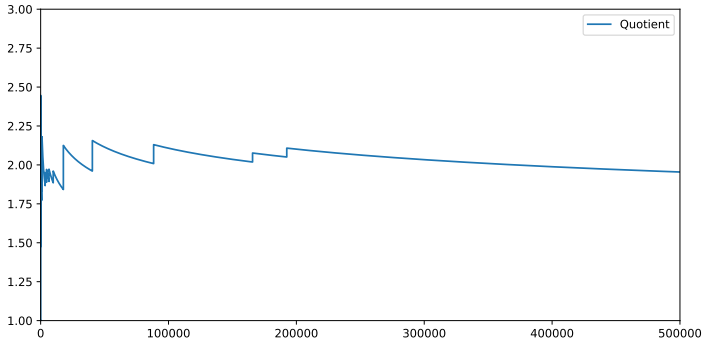


FIGURE V.14. Le quotient $M(\mathcal{T}_{\mathbb{P}}, N)/\log_2(N)$ pour $2 \leq N \leq 500000$.

Récemment, Drmota, Müllner et Spiegelhofer ont montré un résultat d'équirépartition de la fonction somme des chiffres des nombres premiers en base de Zeckendorf, voir [DMS21]. Il serait intéressant de déterminer si les méthodes de cet article peuvent s'adapter dans le contexte de l'étude de la complexité d'ordre maximal de la suite $\mathcal{S}_Z$ le long des nombres premiers.

V.4.2. Raréfaction le long d'une suite récurrente. Nous pouvons également étudier la suite de Thue–Morse le long d'une suite récurrente. Par exemple, la suite $\mathcal{T}_F = (t(F_n))_{n\geq 0}$, la suite de Thue–Morse le long de la suite de Fibonacci. Le principe général pour créer une suite avec de bonnes propriétés aléatoires est de prendre une suite de la forme $\mathcal{S}_f = (s_{f(n)})_{n\geq 0}$ avec $\mathcal{S} = (s_n)_{n\geq 0}$ qui n'est pas "liée" à la fonction extractrice f. C'est un principe inspiré en quelque sorte par le théorème de Cobham.

FIGURE V.15. Les 64×64 premiers termes de $\mathcal{T}_F$.

CHAPITRE VI

Sur les chiffres en base 2 de n et de n^2

Ce chapitre est une reproduction traduite de l'article écrit en collaboration avec K. Aloui, D. Jamet, H. Kaneko, S. Kopecki et T. Stoll [AJK+22], accepté pour publication le 12/10/2022 dans le journal *Theoretical Computer Science*. Notons que dans ce chapitre, nous étudions uniquement la fonction somme des chiffres en base 2. Par conséquent, seulement pour ce chapitre, $s(n)$ désigne la fonction somme des chiffres en base 2, aussi appelée poids de Hamming, d'un entier naturel $n \in \mathbb{N}$. De même, dans toute la suite, l'appellation chiffre désigne chiffre binaire sauf mention contraire.

VI.1. Introduction

Dans ce chapitre, nous nous intéressons à déterminer s'il existe une infinité de solutions impaires à l'équation

$$s(n) = s(n^2) = k \tag{VI.1.1}$$

pour un $k \in \mathbb{N}$ fixé. La majeure partie des cas a été résolu par Hare, Laishram et Stoll [HLS11b], à l'exception des cas où $k \in \{9, 10, 11, 14, 15\}$. Notre contribution est de montrer, via des outils de combinatoire et une approche algorithmique, qu'il n'y a qu'un nombre fini de solutions pour $k \in \{9, 10, 11\}$. Nous étudions également les deux cas restants $k = 14, 15$, nous formulons des conjectures sur ces cas précis et expliquons leurs motivations. Nous verrons que ces conjectures ne peuvent pas seulement se fonder sur des heuristiques et nous donnons des arguments supplémentaires.

La motivation principale d'étudier (VI.1.1) vient d'un travail de Madritsch et Stoll [MS14] qui ont montré que $(s(n^2)/s(n))_{n\geq 1}$ est dense dans $\mathbb{R}^+$. Ceci généralise directement un ancien résultat de Stolarsky [Sto78] qui établit que $\liminf_{n\to\infty} s(n^2)/s(n) = 0$, voir aussi les travaux [HLS11a, Lin97, Mel05, Mei15, Sau15] pour de plus amples résultats.

Dans la preuve de Stolarsky, le nombre de chiffres pour démontrer que $\liminf_{n\to\infty} s(n^2)/s(n) = 0$ n'est pas constant et (VI.1.1) se place donc dans un contexte différent. Or, pour un entier $n \geq 1$, $s(n^2)$ est deux fois plus grand que $s(n)$ en moyenne, voir [BK95, Pet02]. L'équation (VI.1.1) concerne un ensemble exceptionnel d'entiers. En particulier, il est assez intriguant de voir que pour certaines valeurs de k, il y a une infinité de solutions impaires n et pour d'autres valeurs de k seulement un nombre fini. Un des résultats de Hare, Laishram et Stoll [HLS11b] établit qu'il existe des familles infinies paramétrées de solutions pour $k = 12, 13$ et $k \geq 16$. Plus précisément, ils montrent que

$$s(n) = s(n^2) = 12, \text{ pour tout } n = 111 \cdot 2^t + 111, \text{ avec } t \geq 15, \tag{VI.1.2}$$

$$s(n) = s(n^2) = 13, \text{ pour tout } n = 23 \cdot 2^t + 1471, \text{ avec } t \geq 21, \tag{VI.1.3}$$

$$s(n) = s(n^2) = 16, \text{ pour tout } n = 111 \cdot 2^t + 1919, \text{ avec } t \geq 21. \tag{VI.1.4}$$

De l'autre côté du spectre, il n'existe qu'un nombre fini de solutions pour $k \leq 8$. Par exemple l'équation

$$s(n) = s(n^2) = 8$$

ne possède que 64 solutions impaires et la plus grande solution est $n = 266335$, voir [HLS11b, Table 2]. Ces résultats de finitude de solutions sont basés sur un algorithme capable de traiter tous les possibles arrangements des exposants de n^2 lorsqu'on écrit n en base 2. Lorsque le poids de Hamming k est faible, il est possible de le faire de manière exhaustive et la méthode de [HLS11b] détermine explicitement toutes les solutions pour $k \leq 8$. Cependant, le temps de calcul et la longueur du programme explosent pour de plus grandes valeurs de k. Quelques arguments heuristiques sont donnés dans [HLS11b, Section 5] pour supporter la conjecture suivante : il y a qu'un nombre fini de solutions également pour $k \in \{9, 10\}$.

Les objectifs de cet article sont de combiner un nouveau lemme de factorisation, basé uniquement sur la combinatoire ainsi que deux algorithmes basés sur les résultats récents de Kaneko et Stoll [KS22] pour réduire la recherche à seulement un nombre fini de cas. De cette manière, nous prouvons que (VI.1.1) admet un nombre fini de solutions impaires pour $k \in \{9, 10, 11\}$.

VI.2. Résultats principaux

Dans ce paragraphe, nous exposons les résultats principaux démontrés dans l'article ainsi que le contexte dans lequel ils se placent.

VI.2.1. Sur les équations $s(n) = s(n^2)$. Notre premier théorème établit une résolution complète de (VI.1.1) pour $k = 9, 10, 11$.

Théorème VI.2.1. *Soit $k \in \{9, 10, 11\}$. Le nombre d'entiers impairs n tel que $s(n) = s(n^2) = k$ est fini.*

Contrairement à [HLS11b], notre méthode ne permet pas de donner toutes ces solutions, car nous verrons que la borne est bien trop grande pour pouvoir être atteinte par un ordinateur en un temps raisonnable.

Nous faisons également une recherche globale pour $s(n^2) = s(n) = k$, $11 \leq k \leq 15$, jusqu'à $n < 2^{80}$, voir Paragraphe VI.7. Après une inspection des résultats, aucune famille infinie n'apparaît clairement pour les cas $k = 14$ et $k = 15$ contrairement aux cas $k \in \{12, 13, 16\}$, voir (VI.1.2)–(VI.1.4). Nous formulons la conjecture suivante.

Conjecture VI.2.1. *Soit $k \in \{14, 15\}$. Le nombre d'entiers impairs n tel que $s(n) = s(n^2) = k$ est fini.*

Pour la démonstration du Théorème VI.2.1, il est crucial d'avoir des algorithmes efficaces à notre disposition capables de calculer des ensembles d'entiers particuliers. En effet, ces ensembles d'entiers apparaissent à des étapes intermédiaires de la preuve et nécessitent de calculer des entiers jusqu'à des bornes importantes. Pour $\ell_1, \ell_2, m \geq 1$ donnés, nous définissons ensembles suivants :

$$\Delta_{\ell_1,\ell_2,m} := \{n \in \mathbb{N} : \quad s(n) = \ell_1, \quad s(n^2) \leq \ell_2, \quad n < 2^m, \quad n \text{ impair}\}. \tag{VI.2.1}$$

Notre premier algorithme, appelé `next`, a pour objectif de calculer $\Delta_{\ell_1,\ell_2,m}$ pour des petites valeurs de ℓ_1, ℓ_2, m et en temps rapide.

VI.2.2. Sur les équations $s(n^2) \in \{4, 5\}$. Nous étudions également les équations $s(n^2)$ faible à poids fixé, qui se sont retrouvées indispensables pour la preuve du Théorème VI.2.1. Pour $k \in \mathbb{N}$ et $\lambda \geq 1$ donnés, nous définissons les ensembles suivants :

$$E_{k,\lambda} := \{n \in \mathbb{N} : \quad s(n^2) = k, \quad s(n) = \lambda, \quad n \text{ impair}\}, \tag{VI.2.2}$$

et

$$E_k \ := \bigcup_{\lambda \geq 1} E_{k,\lambda}. \tag{VI.2.3}$$

L'objectif du deuxième algorithme, appelé `max-integer`, est de calculer efficacement $E_{k,\lambda}$ pour des petites valeurs de k et λ. Plusieurs résultats sur les ensembles E_k sont déjà connus dans la littérature. Il est élémentaire de montrer que $E_2 = \{3\}$.

Pour les entiers impairs dont le carré est de poids 3, Szalay [Sza02] montre que

$$E_3 = \{2^t + 1 : \ t \geq 2\} \cup \{7, 23\}, \tag{VI.2.4}$$

et voir aussi [Luc04] pour une généralisation de ce résultat. La preuve de Szalay repose sur un résultat de Beukers sur l'équation de Ramanujan–Nagell. Plus précisément, on a le théorème suivant.

Théorème VI.2.2 ([Sza02]). *Soient a, b et n des entiers positifs tels que $a > b > 0$ et $2^a + 2^b + 1 = n^2$. Alors*

- $(a, b, n) \in \{(2t, t + 1, 2^t + 1)\}$ *pour* $t \geq 2$ *ou*
- $(a, b, n) \in \{(5, 4, 7), (9, 4, 23)\}$

De tels ensembles sont typiquement composés d'union de familles infinies paramétrées et d'un ensemble fini de solutions dites sporadiques. Dans le théorème précédent, les éléments de la famille infinie sont toujours de poids 2 et parmi les solutions sporadiques 7 est de poids 3 et 23 est de poids 4. Notons que les solutions sporadiques sont de poids plus grand que 3, la donnée du problème, alors que la famille infinie ne fournit que des éléments de poids plus petit que 3. Nous retrouverons ce schéma plus tard également. Puis, les résultats des travaux [BBM12, BB14, HP14, BHMP16, Ben17] ont généralisé le résultat de Szalay à d'autres bases ou d'autre puissances. Finalement, mentionnons le résultat récent de Szalay [Sza20] qui a considéré des algorithmes pour chercher les solutions à l'équation diophantienne $2^n + \alpha \cdot 2^m + \alpha^2 = x^2$, où α est un entier positif fixé.

En ce qui concerne E_4, il est connu que c'est un ensemble fini et donc aucune famille infinie n'apparaît, voir Bennett, Bugeaud et Mignotte [BBM12], ou Corvaja et

Zannier [CZ13] qui ont montré ce résultat indépendamment. De plus, Bennett, Bugeaud et Mignotte ont formulé la conjecture suivante.

Conjecture VI.2.2 ([BBM12]).

$$E_4 = \{13,\ 15,\ 47,\ 111\}. \tag{VI.2.5}$$

Dans ce chapitre, nous étudions une version plus fine de la Conjecture VI.2.2. Plus précisément, nous regardons les entiers n avec un nombre fixé de chiffres plutôt que tous les entiers en général. Cette étude sera particulièrement utile pour l'étude de (VI.1.1), où nous avons besoin de connaître explicitement $E_{4,\lambda}$ et $E_{5,\lambda}$ pour de petites valeurs de λ. En effet, une famille infinie qui satisfait (VI.1.3) est construit à partir des entiers 23 et 1471. Comme $s(23^2) = 3$, $s(1471^2) = 5$ et $s(23 \cdot 1471) = 5$, nous retrouvons le bon montant de bits dans le carré de $n = 23 \cdot 2^t + 1471$ pour t suffisamment grand. En effet, pour t suffisamment grand, les blocs de chiffres qui composent n^2 n'interfèrent pas entre eux et on a

$$\begin{aligned} &s(n) = s(23) + s(1471) = 4 + 9 = 13, \\ &n^2 = 2^{2t} \cdot 23^2 + 2^{t+1} \cdot 23 \cdot 1471 + 1471^2, \\ &s(n^2) = s(23^2) + s(23 \cdot 1471) + s(1471^2) = 3 + 5 + 5 = 13. \end{aligned}$$

Notons que nous retrouvons les méthodes similaires aux chapitres précédents sur les sommes non-interférentes.

En appliquant l'algorithme `max-integer` et le lemme de factorisation décrit dans la suite de ce chapitre, nous montrons le résultat suivant.

Théorème VI.2.3. *On a*

$$\bigcup_{\lambda \le 17} E_{4,\lambda} = \{13,\ 15,\ 47,\ 111\}, \tag{VI.2.6}$$

et

$$\bigcup_{4 \le \lambda \le 15} E_{5,\lambda} = \{29,\ 31,\ 51,\ 79,\ 91,\ 95,\ 157,\ 223,\ 279,\ 479,\ 727,\ 1471,\ 5793\}. \tag{VI.2.7}$$

De plus on a

$$E_5 = \{1 + 2 + 2^\ell :\ \ell \ge 3\} \cup \{1 + 2^\ell + 2^{\ell+1} :\ \ell \ge 3\} \cup \{1 + 2^\ell + 2^{2\ell-1} :\ \ell \ge 3\} \cup E', \tag{VI.2.8}$$

où E' est un ensemble fini.

Ce théorème donne plus de détails sur la Conjecture VI.2.2, car il indique que tous les $E_{4,\lambda}$ sont vides pour $7 \le \lambda \le 17$. Ce résultat peut aussi suggérer une preuve calculatoire potentielle de la conjecture s'il est possible d'établir une borne universelle sur λ.

Pour (VI.2.8), nous formulons la conjecture suivante.

Conjecture VI.2.3. *L'ensemble E' de* (VI.2.8) *est*

$$E' = \{29,\ 31,\ 51,\ 79,\ 91,\ 95,\ 157,\ 223,\ 279,\ 479,\ 727,\ 1471,\ 5793\}.$$

Cette conjecture est motivée par le fait que le plus grand poids qui apparaît pour un entier de E' est pour l'entier 1471 et $s(1471) = 9$. Par conséquent, les ensembles $E_{5,\lambda}$ sont tous vides pour $10 \leq \lambda \leq 15$.

Notons qu'un calcul élémentaire montre qu'il existe au moins trois familles infinies paramétrées telles que $s(n^2) = 5$:

- $n = 1 + 2 + 2^\ell$ pour $\ell \geq 3$.
- $n = 1 + 2^\ell + 2^{\ell+1}$ pour $\ell \geq 3$.
- $n = 1 + 2^\ell + 2^{2\ell-1}$ pour $\ell \geq 3$.

Cependant le poids des entiers de ces familles est faible, ici 3, et aucune autre famille infinie ne semble apparaître. Notons la forte ressemblance avec le cas $s(n^2) = 3$ puisque les familles infinies proviennent d'un poids plus faible que celui du carré.

Le reste de ce chapitre est structuré de la manière suivante. Dans le Paragraphe VI.3, nous établissons plusieurs résultats auxiliaires. En particulier, nous prouvons un nouveau lemme combinatoire de factorisation qui est au cœur de notre méthode. Le Paragraphe VI.4 est dédiée à la preuve du Théorème VI.2.1, où nous utilisons l'algorithme `next`. Dans le Paragraphe VI.5, nous donnons une preuve du Théorème VI.2.3 à l'aide de l'algorithme `max-integer`. Les descriptions de ces algorithmes seront faites plus tard, à savoir dans le Paragraphe VI.6 pour plus de lisibilité. Enfin nous terminons ce chapitre par le Paragraphe VI.7 avec quelques remarques sur les cas restants de (VI.1.1), c'est-à-dire les cas $k = 14$ et $k = 15$, qui sont encore ouverts.

VI.3. Préliminaires

Soit $n = \sum_{i=0}^{\ell-1} \varepsilon_i 2^i$ avec $\varepsilon_i \in \{0, 1\}$ et $\varepsilon_{\ell-1} = 1$. Dans toute la suite, nous notons $\ell = \ell(n)$ la longueur de l'écriture binaire d'un entier n et $(n)_2$ pour sa décomposition binaire. Rappelons que nous notons $s(n) = \sum_{i=0}^{\ell-1} \varepsilon_i$ la somme des chiffres de n. On utilise la lettre x, avec ou sans indices, pour des blocs de chiffres binaires qui terminent toujours par un `1`, ils correspondent donc à des écritures binaires d'entiers impairs. Par abus de langage, on note parfois x également l'entier associé au bloc x lorsque qu'il n'y a pas de confusion. Par exemple $x = 3 = \texttt{11}$ et on note 1 pour l'entier "un" et `1` le un-bit.

Pour un entier impair n tel que $s(n) = k$, on considère une décomposition de son écriture binaire de la forme

$$(n)_2 = x_m \texttt{0}^{\ell_m} x_{m-1} \cdots x_1 \texttt{0}^{\ell_1} x_0 \tag{VI.3.1}$$

en $m < k$ blocs de `0`-bits de longueurs ℓ_i, pour $1 \leq i \leq m$, qui séparent les x_i, pour $0 \leq i \leq m$. Rappelons que tous les x_i terminent par un `1`-bit. Notons que la décomposition (VI.3.1) n'est pas unique car on peut fusionner des blocs intérieurs et ainsi réduire le nombre de x_i et obtenir d'autres factorisations. On note $y_{i,j} = x_i x_j$ pour $0 \leq i, j \leq m$, les blocs qui composent n^2 et notons que $y_{i,j}$ termine aussi par un `1`-bit. Soient

$$\hat{\ell}_j = \sum_{i=1}^{j} (\ell_i + \ell(x_{i-1}))$$

pour $0 \leq j \leq m$, qui représentent la longueur du bloc $\mathtt{0}^{\ell_j} x_{j-1} \cdots x_1 \mathtt{0}^{\ell_1} x_0$. De plus, soient $\ell_{i,i} = 2\hat{\ell}_i$ pour $0 \leq i \leq m$ et $\ell_{i,j} = \hat{\ell}_i + \hat{\ell}_j + 1$ pour $0 \leq i, j \leq m$ et $i \neq j$. Par conséquent, le carré n^2 est la somme des entiers :

$$u_{i,i} = 2^{\ell i,i} y_{i,i} \quad \text{pour } 0 \leq i \leq m \text{ et}$$
$$u_{i,j} = 2^{\ell i,j} y_{i,j} \quad \text{pour } 0 \leq i < j \leq m.$$

On dit que le terme $y_{i,j}$ *interfère* avec le terme $y_{i',j'}$ si dans l'addition des deux, une propagation de retenues causée par $y_{i,j}$ atteint un chiffre de $y_{i',j'}$, ou vice-versa, voir leChapitre II. Nous discuterons souvent de la situation de combien de `1`-bits reste dans l'addition de termes interférants et nous rejetterons les possibilités où les additions causent de trop grandes contributions. Si les blocs sont *non-interférants*, alors le nombre de `1`-bits de leur somme est le somme des chiffres des termes, comme au Chapitre II. Ici on s'accorde également des décalages possibles, à gauche comme à droite, car les termes qui composent n^2 sont des décalages des $y_{i,j}$.

Nous donnons un exemple simple du fonctionnement de cette procédure. Si on ajoute `111` ou ses décalages à `11100001`, on observe directement qu'il impossible de trouver un décalage tel que la somme des termes n'a qu'un seul chiffre :

```
  11100001
+  <-- 111 -->
```

Dans ce cas, on peut dire que le bloc de `0`-bit intérieur est trop grand pour que l'interférence entre `11100001` et `111` puisse produire un seul `1`-bit. Cette procédure peut être aisément implémentée puisque qu'il suffit de tester via une boucle `for`. Si plus que deux termes doivent être ajoutés, il suffit d'effectuer plus de boucles `for`.

Le lemme suivant donne une condition suffisante pour que deux termes soient non-interférants.

Lemme VI.3.1. *Soient $y_{i,j}$, $y_{i',j'}$ deux termes définis comme précédemment. Si $\ell_{i,j} \geq \ell_{i',j'} + \ell(y_{i',j'}) + k^2$, alors $y_{i,j}$ n'interfère pas avec $y_{i',j'}$.*

Preuve. Si ces deux termes interfèrent, il y aurait au moins k^2 `1`-bits impliqués dans la propagation de retenues de $y_{i',j'}$ vers $y_{i,j}$. Or pour $s(n) = k$, n^2 contient au maximum $k + \frac{k(k-1)}{2}$ `1`-bits, où k est le nombre de termes provenant d'un carré et $k + \frac{k(k-1)}{2}$ le nombre de termes provenant d'un double produit. □

Soit $\ell_{\min} = \min_{1 \leq i \leq m}(\ell_i)$ et $\ell_{\max} = \max_{0 \leq i \leq m} \ell(x_i)$. Par le Lemme VI.3.1 on peut en déduire que si $\ell_{\min} > 2\ell_{\max} + k^2$, alors deux termes $y_{i,j}$, $y_{i',j'}$ n'interfèrent pas si $i > i'$ et $j \geq j'$.

Le lemme suivant est au coeur de notre méthode et nous l'appellerons par la suite lemme de factorisation.

Lemme VI.3.2 (Lemme de factorisation). *Pour $k \geq 1$, il existe une borne N_k tel que pour toute décomposition binaire d'un entier impair $n \geq N_k$ qui vérifie $s(n) = s(n^2) = k$ peut se factoriser comme*

$$(n)_2 = x_m \mathtt{0}^{\ell_m} x_{m-1} \cdots x_1 \mathtt{0}^{\ell_1} x_0 \tag{VI.3.2}$$

où $1 \leq m < k$, $x_0, \ldots, x_m$ *sont les décompositions binaires d'entiers impairs et* $\ell_1, \ldots, \ell_m \in \mathbb{N}$ *tels que* $\ell_{\min} > 2x_{\max} + k^2$ *où* $\ell_{\min} = \min_{1\leq i\leq m}(\ell_i)$ *et* $x_{\max} = \max_{0\leq i\leq m} |x_i|$.

PREUVE. Soit $f(i) = 4i + k^2$ et $N_k = 2^{f^k(1)}$ où f^k est la k-ème composition de f. Considérons un entier impair $n \geq N_k$ tel que $s(n) = k$. Si sa décomposition binaire contient $(k-1)$ blocs de 0-bits et si chacun de ces blocs de 0-bits est plus long k^2+2, alors chaque 1-bit dans la décomposition forme un des x_i avec $m = k-1$ et on a terminé. Sinon, on fusionne tous les blocs de 0-bits qui sont de longueur au plus k^2+2 avec leurs bords des 1-bits et ils forment un des facteurs x_i. Si tous les blocs de 0-bits restants sont plus longs que $2(k^2+4)+k^2$, on trouve une factorisation convenable pour $(n)_2$ comme avant mais avec $m = k-2$. Sinon, on répète ce processus et on obtient que pour $n > 2^{f^k(1)}$ a la bonne factorisation. □

Nous avons également besoin de résultats élémentaires sur les multiples de 3 avec peu de 1-bits. Ces résultats sont inscrits dans les trois prochains lemmes.

Lemme VI.3.3. *Soit* n *un entier impair avec* $s(3n) = 2$. *Alors* $(n)_2 \in \{(10)^\ell 11 : \ \ell \geq 0\} \cup \{1\}$.

PREUVE. Pour $n \geq 5$, on pose $n = 1\varepsilon_d\varepsilon_{d-1}\cdots\varepsilon_0 1$ ($\varepsilon_i \in \{0,1\}, d \geq 0$) et on observe que l'addition $2n+n$ se traduit par

$$\begin{array}{cccccccc} & 1 & \varepsilon_d & \varepsilon_{d-1} & \cdots & \varepsilon_1 & \varepsilon_0 & 1 \\ + & & 1 & \varepsilon_d & \varepsilon_{d-1} & \cdots & \varepsilon_1 & \varepsilon_0 \ 1. \end{array}$$

Comme le dernier 1-bit ne peut pas être modifié par cette addition, l'addition de l'avant-dernier 1-bit à ε_0 doit créer une propagation de retenues et se propager vers les chiffres les plus à gauche. La seule manière pour que cela arrive sans créer de 1-bit additionnel est d'alterner les 1-bits dans les ε_i, i.e. avoir $\varepsilon_0 = \varepsilon_2 = \varepsilon_4 = \cdots = 1$ et $\varepsilon_1 = \varepsilon_3 = \varepsilon_5 = \cdots = 0$. On obtient bien alors la forme voulue. □

Lemme VI.3.4. *Soit* n *un entier impair tel que* $s(3n) = 4$. *Alors* $(n)_2 = x_1 0^s x_0$ *pour un certain* $s \geq 2$ *et avec*

$$x_1, x_0 \in \{(10)^\ell 11 : \ \ell \geq 1\} \cup \{1\},$$

ou $n \leq 2^{2s(n)-1}$.

PREUVE. S'il existe un bloc de 0-bits de longueur ≥ 2 dans $(n)_2$, alors dans l'addition $2n+n = 3n$, il y a des termes non-interférents et ces additions doivent avoir deux 1-bits dans la somme des portions correspondantes. On peut alors utiliser le Lemme VI.3.3 et les seules possibilités sont les blocs $(10)^\ell 11$ pour un certain $\ell \geq 0$, et un bloc contenant le seul chiffre 1. On a alors la première partie du lemme. S'il n'existe pas deux 0 consécutifs, alors n est trivialement borné par $2^{2s(n)} - 1$. □

Lemme VI.3.5. *Soit* n *un entier impair tel que* $s(3n) = 3$. *Alors*

$$(n)_2 \in \{1(01)^{\ell_1}(10)^{\ell_2}11 : \ \ell_1, \ell_2 \geq 0\}$$

Preuve. Nous rappelons l'argument dans la preuve du Lemme VI.3.3. Une possible retenue doit se propager vers les chiffres les plus significatifs pour générer uniquement des 0-bits à l'exception du chiffre le moins significatif et le plus significatif. Dans la preuve précédente, cela impliquait une alternance de 0-bits et de 1-bits dans la partie du milieu. Dans l'énoncé du présent lemme, comme la somme finale contient trois 1-bits, cette alternance doit être brisée au moins une fois. On a alors la somme suivante :

$$\begin{array}{rcccccccccccc} & 1 & \varepsilon_d & \cdots & \varepsilon_k & 1 & \underline{1} & 0 & \cdots & 1 & 0 & 1 & 1 \\ + & & 1 & \varepsilon_d & \cdots & \varepsilon_k & \underline{1} & 1 & 0 & \cdots & 1 & 0 & 1 & 1. \end{array}$$

Le chiffre le moins significatif 1-bit reste dans la somme de ces deux nombres car il n'interagit pas avec les autre chiffres. Dans le chevauchement des 1-bits au point de ruptude (souligné dans la somme précédente), il va rester un 1-bit dans la somme. La somme de ces chiffres génère une propagation qui doit générer que des 0-bits jusqu'au chiffres les plus significatifs. La seule façon d'y parvenir est à nouveau d'alterner les bits 0 et les bits 1. Cela se traduit directement dans la forme donnée. □

Dans les trois précédents lemmes, on voit que le bloc $(10)^\ell 11$ apparaît fréquemment. Il est alors naturel de s'intéresser au carré de ce bloc et à son poids pour notre problème. Le lemme suivant donne une borne inférieure sur le poids de ce carré.

Lemme VI.3.6. *Si $(n)_2 \in \{(10)^\ell 11 : \ \ell \geq 2\}$ alors $s(n^2) \geq 7$.*

Preuve. Un calcul élémentaire montre que si $(n)_2 = (10)^\ell 11$, $\ell \geq 1$, alors

$$(n^2)_2 = \begin{cases} (111000)^{j-1}111001(000111)^j001 & \text{si } \ell = 3j \text{ pour un } j \geq 1, \\ (111000)^j1111(000111)^j001 & \text{si } \ell = 3j+1 \text{ pour un } j \geq 0, \\ (111000)^j11100111(000111)^j001 & \text{si } \ell = 3j+2 \text{ pour un } j \geq 0. \end{cases}$$

Donc, dans tous les cas $s(n^2) \geq 7$. □

Par la suite, nous fixons la valeur de $s(n)$, ce qui correspond à ℓ est petit dans Lemme VI.3.4. Le Lemme VI.3.6 garantit alors que le carré de tels entiers a trop de 1-bits et nous permet d'aboutir à des contradictions. Nous utilisons cette procédure plusieurs fois dans notre étude, en particulier pour vérifier qu'il n'y a pas de solutions impaires pour le système d'équations

$$s(3n) = 4, s(n) = 9 \text{ et } s(n^2) = 5.$$

Les solutions sporadiques bornées par le Lemme VI.3.4 peuvent être trouvées directement par une recherche exhaustive avec un ordinateur.

Nous rappelons les deux récents résultats de Kaneko et Stoll [KS22], portant sur le produit d'entiers avec un poids faible.

Lemme VI.3.7. *Soient $\ell, m \geq 2$, et soient a et b deux entiers impairs tels que $s(a) = \ell$ et $s(b) = m$. Si $s(ab) = 2$, alors on a*

$$ab < 2^{2\ell m - 4}.$$

Lemme VI.3.8. *Soient $\ell, m \geq 2$, et soient a et b deux entiers impairs tels que $s(a) = \ell, s(b) = m$. et $m\ell \geq 5$. Si $s(ab) = 3$, alors on a*

$$ab < 2^{4\ell m - 13}.$$

Nous utilisons ces résultats lors de la recherche de solutions de la forme $x_1\mathtt{0}\cdots\mathtt{0}x_0$ pour un grand bloc de 0-bits intérieur. Pour une telle structure, nous avons trois contributions distinctes à la décomposition binaire du carré : x_1^2, x_0^2 et le double produit x_1x_0 qui n'interfèrent pas car tous les blocs sont bien séparés.

Remarque VI.3.1. Un analogue au Lemme VI.3.7 ou au Lemme VI.3.8 pour $s(ab) = 4$ est faux, voir [KS22, Theorem 2.3].

VI.4. Preuve du Théorème VI.2.1

D'après le Lemme VI.3.2, s'il existe un nombre infini d'entiers impairs solutions à (VI.1.1) pour un k fixé, alors presque toutes les solutions (i.e. toutes à part un nombre fini d'exceptions) ont leurs décompositions binaires qui peuvent être factorisées avec ce lemme. Considérons une factorisation de $(n)_2$ comme dans Lemme VI.3.2 et notons qu'aucun des $2m+1$ termes $y_{m,m}$, $y_{m,m-1}$, …, $y_{m,0}$, $y_{m-1,0}$,…, $y_{0,0}$ interfèrent entre eux. Certains de ces termes peuvent interférer avec d'autres termes, mais même dans ces cas là chacun contribue au moins à hauteur d'un 1-bit dans l'écriture binaire de n^2. Par conséquent, si $m \geq k/2$, alors $s(n^2) \geq 2m + 1 > k$ et n ne peut pas être une solution de (VI.1.1). On peut alors supposer que $1 \leq m < k/2$.

Nous avons les graphes correspondants pour montrer les possibles interférences entre les termes $y_{i,j}$. Ici, les sommets sont les termes et les arrêtes correspondent à une possible interférence entre ces termes.

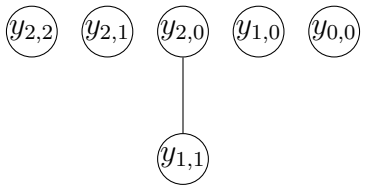


FIGURE VI.1. Graphe d'interférence pour $m = 2$.

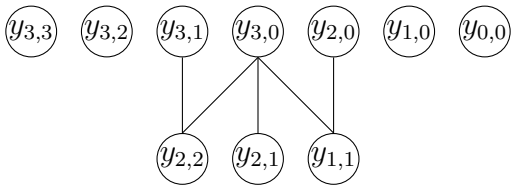


FIGURE VI.2. Graphe d'interférence pour $m = 3$.

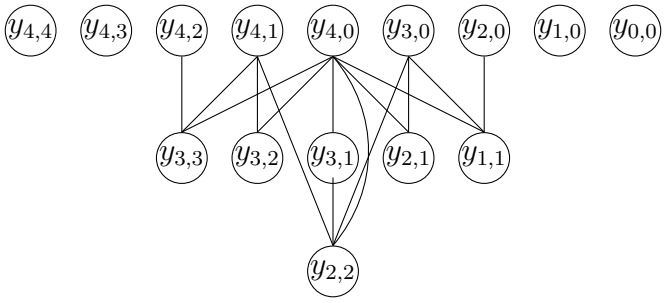


FIGURE VI.3. Graphe d'interférence pour $m = 4$.

Le lemme suivant implique le Théorème VI.2.1, à l'aide du Lemme VI.3.2,

Lemme VI.4.1. *Soient $k \in \{9, 10, 11\}$ et $n \geq N_k$ un entier impair suffisamment grand tel que*

$$s(n) = s(n^2) = k.$$

Alors il n'existe pas de factorisation de $(n)_2$ qui satisfasse le Lemme VI.3.2.

Dans toute la suite de ce chapitre, pour simplifier la lisibilité des distinctions de cas, nous encadrons la valeur de k. Pour $\boxed{k = 9}$ et $\boxed{k = 10}$, $1 \leq m \leq 4$, et pour $\boxed{k = 11}$, on a $1 \leq m \leq 5$. La preuve du Lemme VI.4.1 est basée sur des considérations élémentaires mais nous avons besoin de combiner plusieurs éléments pour conclure : le résultat de Szalay sur E_3, les résultats de Kaneko et Stoll sur le produit d'entiers de poids 2 et 3, la non-existence de certains carrés modulo une puissance de 2, une analyse des possibles interférences entre des blocs, les tableaux de Hare, Laishram et Stoll [HLS11b] etc. La recherche se résume à une analyse d'un nombre fini de cas où les détails dépendent des paramètres en jeu.

Pour la lisibilité du lecteur, nous arrangeons la preuve en fixant le paramètre m plutôt que le paramètre k, car les arguments sont similaires à chaque k. En effet, c'est en premier lieu la forme de la factorisation et le graphe d'interférence qui dirigent la distinction de cas. Nous utilisons librement les algorithmes `next` et `max-integer`, où leurs descriptions seront faites dans le Paragraphe VI.6.

VI.4.1. Le cas $m = 1$. On a $(n)_2 = x_1\mathtt{0}\cdots\mathtt{0}x_0$ avec un (long) bloc intérieur continu de `0`-bits. Tout d'abord, on observe que si $x_0 = 1$, alors n ne peut pas être solution de (VI.1.1) puisque cela impliquerait que $s(n^2) = s(x_1^2)+s(x_1)+1 = s(x_1^2)+(k-1)+1 > k$. Symétriquement, on a $x_1 \neq 1$ et on suppose alors sans perte de généralité que $s(x_1) \geq s(x_0)$. Donc, nous devons résoudre le système suivant :

$$\begin{cases} s(x_1) + s(x_0) = k, \\ s(x_1^2) + s(x_1x_0) + s(x_0^2) = k, \\ s(x_1),\ s(x_0),\ s(x_1^2),\ s(x_0^2),\ s(x_1x_0) \geq 2, \\ s(x_1) \geq s(x_0). \end{cases} \tag{VI.4.1}$$

On distingue les cas en fonction de la valeur de $s(x_1x_0) \in \{2, \ldots, k-4\}$.

(1) $s(x_1x_0) = 2$. Ici, le Lemme VI.3.7 donne une borne supérieure pour les entiers x_1 et x_0. Rappelons que la définition de l'ensemble $\Delta_{\ell_1,\ell_2,m}$ est donnée dans (VI.2.1). La table suivante liste tous les ensembles à calculer pour trouver un possible couple de solutions (x_1, x_0) au système d'équations (VI.4.1) :

Ensembles Δ pour $k = 9$	Ensembles Δ pour $k = 10$	Ensembles Δ pour $k = 11$
$\Delta_{5,5,36} \times \Delta_{4,5,36}$	$\Delta_{5,6,46} \times \Delta_{5,6,46}$	$\Delta_{6,7,56} \times \Delta_{5,7,56}$
$\Delta_{6,5,36} \times \Delta_{3,5,36}$	$\Delta_{6,6,46} \times \Delta_{4,6,46}$	$\Delta_{7,7,56} \times \Delta_{4,7,56}$
$\Delta_{7,5,36} \times \Delta_{2,5,36}$	$\Delta_{7,6,46} \times \Delta_{3,6,46}$	$\Delta_{8,7,56} \times \Delta_{3,7,56}$
-	$\Delta_{8,6,46} \times \Delta_{2,6,46}$	$\Delta_{9,7,56} \times \Delta_{2,7,56}$

TABLEAU VI.1. Ensembles Δ pour $m = 1$ et $s(x_1x_0) = 2$.

Nous construisons ces ensembles à l'aide de l'algorithme `next` d'une manière efficace puis on vérifie s'il y a une solution au système (VI.4.1); après vérification, il n'y a pas de solution.

Remarque VI.4.1. Dans ce cas précis, nous aurions aussi pu utiliser les tableaux de [HLS11b] pour les cas $\boxed{k = 9}$ et $\boxed{k = 10}$. Cependant, pour le cas $\boxed{k = 11}$, ces tableaux ne sont pas suffisants pour conclure et nous avions besoin d'une autre méthode pour construire ces ensembles.

(2) $s(x_1x_0) = 3$. Avec l'aide du Lemme VI.3.8, on peut de manière similaire au cas précédent se restreindre à un nombre fini d'ensembles. Les ensembles à déterminer sont les suivants :

Ensembles Δ pour $k = 9$	Ensembles Δ pour $k = 10$	Ensembles Δ pour $k = 11$
$\Delta_{5,4,67} \times \Delta_{4,4,67}$	$\Delta_{5,5,87} \times \Delta_{5,5,87}$	$\Delta_{6,6,107} \times \Delta_{5,6,107}$
$\Delta_{6,4,67} \times \Delta_{3,4,67}$	$\Delta_{6,5,87} \times \Delta_{4,5,87}$	$\Delta_{7,6,99} \times \Delta_{4,6,99}$
$\Delta_{7,4,67} \times \Delta_{2,4,67}$	$\Delta_{7,5,87} \times \Delta_{3,5,87}$	$\Delta_{8,6,83} \times \Delta_{3,6,83}$
-	$\Delta_{8,5,87} \times \Delta_{2,5,87}$	$\Delta_{9,6,59} \times \Delta_{2,6,59}$

TABLEAU VI.2. Ensembles Δ pour $m = 1$ et $s(x_1x_0) = 3$.

Après vérification comme précédemment, il n'y a pas de solutions.

Remarque VI.4.2. À noter que dans la dernière colonne, nous avons réduit le nombre de cas à considérer pour accélérer le temps de calcul, ce qui n'était pas réellement nécessaire dans le cas précédent.

(3) $s(x_1x_0) = 4$. À partir de cette valeur sur le poids du double produit, la méthode sera différente car il n'existe pas de bornes universelles sur le produit x_1x_0, voir Remarque VI.3.1. On distingue alors plusieurs cas en fonction de k et du poids du carré du terme x_1.

- Soit $\boxed{k = 9}$. Alors on a $\max(s(x_1^2), s(x_0^2)) = 3$ et $\min(s(x_1^2), s(x_0^2)) = 2$. Rappelons que les définitions de $E_{k,\lambda}$ et E_k sont données dans (VI.2.2) et (VI.2.3),

et que l'on dispose du Théorème VI.2.2. On a $x_1 \in E_3$ et $x_0 = 3$, ou $x_0 \in E_3$ et $x_1 = 3$. Alors $s(x_1) + s(x_0) \leq 6 < 9$, ce qui est une contradiction.

- Soit $\boxed{k = 10}$, on distingue plusieurs cas en fonction de $s(x_1^2)$.
  (a) Si $s(x_1^2) = 4$, alors on a $s(x_0^2) = 2$ et $x_0 = 3$. Ainsi $s(x_1) = 8$ et il n'existe pas un tel x_1, voir [HLS11b, Table 3].
  (b) Si $s(x_1^2) = 3$, alors $x_0, x_1 \in E_3$ et on a $s(x_1) + s(x_0) \leq 8 < 10$. C'est une contradiction.
  (c) Si $s(x_1^2) = 2$, alors $s(x_1) = 2$ et $s(x_0) = 8$. On conclut comme dans le cas (a).
- Soit $\boxed{k = 11}$, on distingue plusieurs cas en fonction de $s(x_1^2)$.
  (a) Si $s(x_1^2) = 5$ alors on a $s(x_0^2) = 2$ et $x_0 = 3$. Ainsi x_1 satisfait

$$s(3x_1) = 4, \qquad s(x_1) = 9. \tag{VI.4.2}$$

  Un calcul par ordinateur montre qu'il n'existe pas de solutions à (VI.4.2) avec $x_1 \leq 2^{17}$ qui satisfait également $s(x_1^2) = 5$. Le Lemme VI.3.4 implique alors que s'il existait une solution à (VI.4.2) avec $x_1 > 2^{17}$ alors

$$x_1 \in \{\mathtt{10}^\alpha(\mathtt{10})^6\mathtt{11} : \ \alpha \geq 1\} \ \cup \ \{(\mathtt{10})^6\mathtt{110}^\alpha\mathtt{1} : \ \alpha \geq 1\}$$
$$\cup \ \{(\mathtt{10})^{\ell_1}\mathtt{110}^\alpha(\mathtt{10})^{\ell_2}\mathtt{11} : \ \ell_1 + \ell_2 = 4, \ \ell_1, \ell_2 \geq 0, \ \alpha \geq 1\}.$$

  Par une recherche directe à l'ordinateur, on montre qu'aucune des formes précédentes ne satisfait $s(x_1^2) = 5$. Notons quand même que pour α suffisamment grand, la somme des chiffres des formes précédentes se stabilisent car les blocs deviennent non-interférents et c'est donc bien une vérification par ordinateur sur un nombre fini de cas. Alternativement, on aurait pu aussi vérifier les solutions à (VI.4.2) via les algorithmes de le Paragraphe VI.5.
  (b) Si $s(x_1^2) = 4$. On a $s(x_0^2) = 3$ et $x_0 \in E_3$. Ainsi $s(x_1) \in \{7, 8, 9\}$. L'algorithme `max-integer` montre que $E_{4,\lambda}$ est vide pour $\lambda \in \{7, 8, 9\}$ ce qui nous permet de conclure.

(4) $\underline{s(x_1x_0) = 5.}$

- Soit $\boxed{k = 9}$. Alors on a $s(x_1^2) = s(x_0^2) = 2$ et $x_1 = x_0 = 3$. Donc $s(x_1) + s(x_0) = 4 < 9$.
- Soit $\boxed{k = 10}$. Alors on a $\max(s(x_1^2), s(x_0^2)) = 3$ et $\min(s(x_1^2), s(x_0^2)) = 2$ et $s(x_1) + s(x_0) \leq 6 < 10$.
- Soit $\boxed{k = 11}$, on distingue plusieurs cas en fonction de $s(x_1^2)$.
  (a) Si $s(x_1^2) = 4$, alors on a $s(x_0^2) = 2$ et $x_0 = 3$. Donc $s(x_1) = 9$ et on conclut car $E_{4,9}$ est vide.
  (b) Si $s(x_1^2) = 3$, alors on a $x_0, x_1 \in E_3$ et on a $s(x_1) + s(x_0) \leq 8 < 11$.

(5) $\underline{s(x_1x_0) = 6.}$

- Soit $\boxed{k = 10}$. Alors on a $s(x_1^2) = s(x_0^2) = 2$ et $x_1 = x_0 = 3$. Alors $s(x_1) + s(x_0) = 4 < 10$.
- Soit $\boxed{k = 11}$. Alors on a $\max(s(x_1^2), s(x_0^2)) = 3$ et $\min(s(x_1^2), s(x_0^2)) = 2$ et encore $s(x_1) + s(x_0) \leq 6 < 11$.

(6) $\underline{s(x_1x_0) = 7.}$ Ici, on a nécessairement $\boxed{k = 11}$. De plus on a $s(x_1^2) = s(x_0^2) = 2$ et $x_1 = x_0 = 3$. Donc $s(x_1) + s(x_0) = 4 < 11$.

Ainsi la preuve du Lemme VI.4.1 est complète pour le cas $m = 1$.

VI.4.2. Le cas $m = 2$. Dans ce cas, nous changeons de stratégie et nous utilisons le graphe d'interférence donné dans Figure VI.1. Il y a cinq termes indépendants $y_{2,2}, y_{2,1}, y_{2,0}, y_{1,0}, y_{0,0}$ qui contribuent chacun à $(n)_2$ avec au moins un 1-bit et uniquement $y_{2,0}$ peut interférer avec $y_{1,1}$. Aucun de ces cinq termes ne peut contribuer à plus que $(k-4)$ 1-bits car $s(n^2) = k$. Notons que dans ce cas, le graphe d'interférence est relativement simple car il n'y a qu'une interférence possible et par conséquent peu de cas seront à considérer.

(1) $\underline{x_2 = 1.}$

- Soit $\boxed{k = 9}$. Alors $s(y_{2,1}) = s(x_1)$ et x_1 a au plus cinq 1-bits. Ainsi $x_0 \neq 1$ et $s(y_{1,0}) \geq 2$. Donc
$$1 + s(x_1) + s(x_0) = 9 \geq 1 + s(x_1) + 1 + 2 + s(x_0^2).$$
Par conséquent $s(x_0^2) \leq s(x_0) - 3$ et cette condition n'a aucune solution pour $2 \leq s(x_0) \leq 7$, voir [HLS11b].
- Soit $\boxed{k = 10}$. Alors $s(y_{2,1}) = s(x_1)$ et x_1 a au plus six 1-bits. Ainsi $x_0 \neq 1$ et $s(y_{1,0}) \geq 2$. Comme précédemment, on a $s(x_0^2) \leq s(x_0) - 3$. La seule solution est pour $x_0 =$ 111011111 et $x_1 = 1$, voir [HLS11b, Table 3]. Mais maintenant $y_{1,0} = x_0$ a $8 > 6$ 1-bits, ce qui est trop.
- Soit $\boxed{k = 11}$. Alors $s(y_{2,1}) = s(x_1)$ et x_1 a au plus sept 1-bits. Ainsi $x_0 \neq 1$ et $s(y_{1,0}) \geq 2$. Encore une fois, $2 \leq s(x_0^2) \leq s(x_0) - 3$ et la seule solution pour $s(x_0) \leq 8$ est $x_0 =$ 111011111. On obtient alors $x_1 =$ 10$^\alpha$1 pour un certain $\alpha \geq 0$. Mais cela implique $s(y_{1,0}) \geq 7$ et cette contribution est trop grande pour $s(n^2) = 11$ car $s(y_{0,0}) = 5$. En fait, $s(y_{1,0})$ est constant pour $\alpha \geq 9$ et on peut vérifier pour les valeurs $0 \leq \alpha \leq 9$ directement. S'il existe une solution pour $s(x_0) = 9$, alors $x_1 = 1$ et donc $s(y_{1,0}) = 9 > 11 - 4$ ce qui donne encore une trop grande contribution à n^2.

(2) $\underline{x_0 = 1}$. C'est symétrique au premier cas.

(3) $\underline{x_2 \neq 1 \text{ et } x_0 \neq 1.}$ C'est le cas le plus difficile pour $m = 2$. Les quatre termes $y_{2,2}, y_{2,1}, y_{1,0}$ et $y_{0,0}$ contiennent plus qu'un seul 1-bit et notons aussi que $x_1 \neq 1$. En fait, si $x_1 = 1$, alors $s(n) = s(x_2) + 1 + s(x_0) \leq s(y_{2,1}) + s(y_{2,0}) + s(y_{1,0}) < s(n^2)$.

- Soit $\boxed{k = 9}$. Alors tous ces termes contiennent exactement deux 1-bits et donc $x_2 = x_0 = 3$. Le fait que $s(x_1) = 5$ et $s(x_2x_1) = 2$, combiné avec le (VI.3.3), implique que $x_1 =$ 10101011. Alors le terme $y_{1,1} =$ 111001000111001 contribue avec plus qu'un seul 1-bit à $(n^2)_2$. En effet, si on décale $y_{1,1} =$ 111001000111001 contre $y_{2,0} =$ 1001 et en ajoutant les termes, le résultat a toujours plus qu'un seul 1-bit. On a la situation suivante

111001000111001

\+ $\longleftarrow$ 1001 $\longrightarrow$

Cela peut être vérifié par une simple boucle `for`, voir le Paragraphe VI.3 pour plus de détails.

- Soit $\boxed{k = 10}$. Alors au moins trois des quatre termes précédents ont exactement un seul 1-bits. Par symétrie, on peut supposer que $y_{2,2}$ et $y_{2,1}$ contiennent exactement un seul 1-bits. Cela implique que $x_2 = 3$ et $x_1 = (10)^\alpha 11$ pour un certain $\alpha \geq 0$, dépendant de x_0. On distingue plusieurs cas en fonction de $s(y_{0,0})$.
  (a) Si $s(y_{0,0}) = 2$ alors $x_0 = 3$ et $x_1 = 1010101011$. Alors $y_{1,1} = 1110001111000111001$ peut interférer avec $y_{2,0} = 1001$. On a la situation suivante

  $$\begin{array}{cc} & 1110001111000111001 \\ + & \longleftarrow 1001 \longrightarrow \end{array}$$

  Cette interférence contribue toujours plus que deux 1-bits.
  (b) Si $s(y_{0,0}) = 3$ alors les choix possibles pour x_0 sont 10111, 111 et $10^\beta 1$ pour un certain $\beta \geq 1$, voir le Théorème VI.2.2. Ainsi, $4 \leq s(x_1) \leq 6$ et on peut aisément vérifier que dans tous les cas $y_{1,0}$ a plus que deux 1-bits.
- Soit $\boxed{k = 11}$. Alors les quatre termes $y_{2,2}, y_{2,1}, y_{1,0}, y_{0,0}$ contiennent plus qu'un seul 1-bit, et donc au moins deux d'entre eux contiennent exactement deux 1-bits. On distingue maintenant plusieurs cas en fonction du nombre de termes avec exactement deux 1-bits.

  En premier, supposons qu'il y a au moins trois termes parmi $y_{2,2}, y_{2,1}, y_{1,0}, y_{0,0}$ qui contiennent exactement deux 1-bits. Par symétrie, on peut supposer que $y_{2,2}$ et $y_{2,1}$ contiennent exactement deux 1-bits. Cela implique que $(x_2)_2 = 11$ et $(x_1)_2 = (10)^\alpha 11$ pour un certain $\alpha \geq 0$. On distingue plusieurs cas en fonction de $s(y_{0,0})$.
  (a) Si $s(y_{0,0}) = 2$, alors $(x_0)_2 = 11$, $(x_1)_2 = (10)^5 11$ et

  $$(y_{1,1})_2 = 11100011100111000111001$$

  qui peut interférer avec $y_{2,0} = 1001$. Cette interférence a toujours plus que trois 1-bits.
  (b) Si $s(y_{0,0}) = 3$, alors les choix possibles pour $(x_0)_2$ sont 10111, 111, et $10^\beta 1$ pour un certain $\beta \geq 1$. Ainsi, $s(x_1) \geq 5$ et dans tous les cas $y_{1,0}$ a plus que deux 1-bits.
  (c) Si $s(y_{0,0}) = 4$ alors comme $s(x_0) \leq 7$, les choix possibles pour $(x_0)_2$ sont 101111, 1101111, 1111, et 1101 (voir Table 1 et Table 3 de [HLS11b]). Ainsi, $s(x_1) \geq 3$ et dans tous les cas $y_{1,0}$ a plus que deux 1-bits.

  Ensuite, supposons maintenant qu'il y a exactement deux termes parmi $y_{2,2}, y_{2,1}, y_{1,0}, y_{0,0}$ qui contiennent deux 1-bits. Il est alors suffisant de considérer les cas suivants :
  (a) Si $y_{2,2}$ et $y_{1,0}$ (ou de manière analogue $y_{2,1}$ et $y_{0,0}$) contiennent exactement deux 1-bits. Alors $(x_2)_2 = 11$ et $s(y_{0,0}) = 3$ et donc $x_0 \in E_3$. Ainsi, $5 \leq s(x_1) \leq 7$ et on a $y_{2,1} = 3x_1$ et $s(3x_1) = 3$. Par le VI.3.5, les

seules solutions pour x_1 sont $\mathtt{1(01)}^{\ell_1}\mathtt{(10)}^{\ell_2}\mathtt{11}$ pour $\ell_1, \ell_2 \geq 0$ tels que $2 \leq \ell_1 + \ell_2 \leq 4$. Les valeurs possibles pour $\mathtt{y}_{2,0}$ et $\mathtt{y}_{1,1}$ sont :

$y_{2,0}$	$y_{1,1}$
10101	1110110010001
1000101	10000001011001
$110^{t-2}11$, $t \geq 2$	10110010111001
	11100101110010001
	11101011001011001
	100000001010111001
	100000001010111001
	1110101011010101111001
	1110101011010101111001
	1110101011010101111001
	1000000010000000111001
	1000000010000000111001

Ainsi, dans tous les cas, l'interférence entre $y_{1,1}$ et $y_{2,0}$ contribue à plus qu'un seul 1-bit.

(b) Si $y_{2,2}$ et $y_{0,0}$ contiennent exactement deux 1-bits. Alors $(x_2)_2 = (x_0)_2 = 11$ et donc $s(x_1) = 7$. Alors la contribution de l'interférence entre $y_{2,0}$ et $y_{1,1}$ a un seul 1-bit si et seulement si $(x_1^2)_2$ est de la forme $1 \cdots 10111$. Cette forme implique en particulier que $x_1^2 \equiv 7 \pmod 8$, et ce n'est pas possible pour un entier impair x_1.

(c) Si $y_{2,1}$ et $y_{1,0}$ contiennent exactement deux 1-bits. Alors $y_{2,2}$ et $y_{0,0}$ contiennent exactement trois 1-bits et donc $x_0, x_2 \in E_3$. Le terme $y_{2,0}$ qui peut interférer avec $y_{1,1}$ doit avoir un seul 1-bit.

Les formes suivantes pour $y_{2,0}$ contredisent le fait que $y_{1,1} \equiv 1 \pmod 8$:

$y_{2,0}$	(x_2, x_0)	Forme requise pour $y_{1,1}$
110001	(7,7)	$(*)$11
1000010001	(23,23)	$(*)$11
10100001	(7,23)	$(*)$11
10001	$(7, 2^2+1)$	$(*)$11
1110011	$(23, 2^2+1)$	$(*)$101
-	$(2^{t_2}+1, 2^{t_0}+1), t_2, t_0 \geq 2$	$(*)$11

Les formes restantes sont plus difficiles car l'argument précédent ne marche pas.

$y_{2,0}$	(x_2, x_0)	Forme requise pour $y_{1,1}$
11001111	$(23, 2^3+1)$	$1 \cdots 100110001$
110000111	$(23, 2^4+1)$	$1 \cdots 1001111001$
$1110^{t-3}111$	$(7, 2^t+1), t \geq 3$	$1 \cdots 10001^{t-3}001$
$101110^{t-5}10111$	$(23, 2^t+1), t \geq 5$	$1 \cdots 1010001^{t-5}01001$

Le premier cas implique que $x_1^2 = 2^8(2^\lambda - 1) + 49$ pour un certain $\lambda \geq 1$, car $s(x_1) = 5$ donne que $x_1^2 = 49$ n'est pas possible. Cependant, λ est borné car $s(x_1^2) \leq s(x_1)(s(x_1)+1)/2$, et donc $\lambda \leq 12$. Alors il est suffisant de vérifier si les entiers $2^8(2^\lambda - 1) + 49$, pour $0 \leq \lambda \leq 12$ sont des carrés parfaits, et ce n'est pas le cas.
Le deuxième cas est similaire car on a $x_1^2 = 2^8(2^\lambda - 1) + 111$ pour un certain $\lambda \geq 1$ et $s(x_1) = 4$. On conclut de la même manière que dans le premier cas.
Et pour le troisième cas, on a $x_1^2 = 2^{t+4}(2^\lambda - 1) + 2^3(2^{t-3} - 1) + 1$ pour un certain $\lambda \geq 1$ et $s(x_1) = 6$. Encore une fois, t est borné et on trouve que l'unique solution est $x_1 = 31$ pour $t = 3$. Cependant, $s(y_{2,1}) = s(7 \times 31) = 5$ contredit notre première hypothèse dans ce cas.
Le dernier cas marche de la même manière que le troisième cas.

Ainsi la preuve du Lemme VI.4.1 est complète pour le cas $m = 2$.

VI.4.3. Le cas $m = 3$. Nous gardons ici dans les grandes lignes la même stratégie que pour $m = 2$ avec quelques différences notables. Dans ce cas, il y a sept termes indépendants $y_{3,3}, \ldots, y_{3,0}, \ldots, y_{0,0}$ chacun contribuant à $(n^2)_2$ avec au moins un 1-bit, voir Figure VI.2. Dans ce paragraphe, les raisonnements pour $k = 9, 10, 11$ vont être assez différents. C'est pourquoi nous séparons ces cas dès le début et plus k augmente, plus le nombre de cas à traiter est élevé car il y a plus de répartitions possibles sur ces sept termes indépendants.

Soit $\boxed{k = 9}$. Alors au moins cinq des sept termes indépendants doivent contenir seulement un 1-bit.

(1) $\underline{x_3 = x_0 = 1.}$ On a $y_{3,2} = x_2$ et $y_{1,0} = x_1$. Ainsi, $s(n^2) > s(y_{3,3}) + s(y_{3,2}) + s(y_{1,0}) + s(y_{0,0}) = s(x_3) + s(x_2) + s(x_1) + s(x_0) = 9$, c'est une contradiction.

(2) $\underline{x_3 \neq 1.}$ Les termes $y_{3,3}$ et $y_{3,2}$ contiennent deux 1-bits alors que les autres termes doivent contribuer uniquement un 1-bit. Alors on en déduit que $x_3 = 3$, $x_1 = x_0 = 1$, et $s(x_2) = 5$. Le (VI.3.3) montre que la seule solution à $s(y_{3,2}) = 2$ est $x_2 = 10101011$. Cependant, le terme $y_{2,2} = 111001000111001$ peut uniquement interférer avec soit $y_{3,1} = 11$ ou soit $y_{3,0} = 11$ et dans les deux cas la contribution à $(n^2)_2$ est plus grande qu'un 1-bit.

(3) $\underline{x_0 \neq 1.}$ Ce cas est symétrique au cas précédent.

Soit $\boxed{k = 10}$. Alors parmi les sept termes indépendants $y_{3,3}, \ldots, y_{3,0}, \ldots, y_{0,0}$, au moins quatre doivent contenir exactement un 1-bit et aucun de ces termes ne peut contribuer à plus que trois 1-bits. En effet, si un de ces termes contribue à plus que trois 1-bit, tous les autres termes indépendants doivent contribuer à hauteur d'un seul 1-bit, ce qui est impossible. On distingue les cas suivants.

(1) $\underline{x_3 = x_0 = 1.}$ On a $y_{3,2} = x_2$ et $y_{1,0} = x_1$, Ainsi, $s(n^2) > s(y_{3,3}) + s(y_{3,2}) + s(y_{1,0}) + s(y_{0,0}) = 10$, c'est une contradiction.

(2) $\underline{s(y_{3,3}) = 2.}$ Cela implique directement que $x_3 = 11$. Alors le terme $y_{3,2}$ contient au moins deux 1-bits, et par conséquent un des deux termes $y_{1,0}$ et $y_{0,0}$ est 1. Cela

implique donc que $x_0 = \mathtt{1}$. On distingue désormais les cas suivants en fonction de $s(y_{3,2})$ et $s(y_{1,0})$.

(a) $\underline{s(y_{3,2}) = 2 \text{ et } s(y_{1,0}) = 2.}$ Comme $y_{1,0} = x_1$ et $s(x_1) = 2$, on obtient $x_2 = (\mathtt{10})^3\mathtt{11}$ et $y_{2,2} = \mathtt{111001000111001}$. Ces termes peuvent interférer avec un des

$$y_{3,1} \in \{\mathtt{110}^\gamma\mathtt{11} : \gamma \geq 0\} \cup \{\mathtt{1001}\}$$

et $y_{3,0} = \mathtt{11}$. Dans les deux cas leurs contributions auront plus qu'un seul 1-bit. Notons que pour $y_{3,1}$ on peut utiliser une boucle `for` sur γ pour conclure.

(b) $\underline{s(y_{3,2}) = 2 \text{ et } s(y_{1,0}) = 1.}$ On a $x_1 = 1$, $x_2 = (\mathtt{10})^4\mathtt{11}$ et

$$y_{2,2} = \mathtt{1110001111000111001}.$$

Ces termes peuvent interférer avec un des $y_{3,1} = x_3 = \mathtt{11}$ et $y_{3,0} = x_3 = \mathtt{11}$, mais ces contributions auront toutes plus que deux 1-bits.

(c) $\underline{s(y_{3,2}) = 3.}$ Ici encore $x_1 = \mathtt{1}$ et le terme $y_{2,1} = x_2$ peut uniquement interférer avec $y_{3,0} = \mathtt{11}$ et ils doivent s'ajouter à une puissance de 2 près. D'après le Lemma VI.3.5, le système $s(3x_2) = 3$, $s(x_2) = 6$ implique que

$$x_2 \in \{\mathtt{101010111}, \mathtt{101101011}, \mathtt{101011011}, \mathtt{110101011}\}.$$

Dans tous les cas, l'interférence entre $y_{2,1}$ et $y_{3,0} = 3$ contribuera à trop de chiffres.

(3) $\underline{s(y_{3,3}) = 3.}$ On a $x_3 \in E_3$ et $s(y_{3,2}) \geq 2$. De plus, on a $x_1 = x_0 = \mathtt{1}$ car $y_{1,0} = \mathtt{1}$. Dans tous les cas, $4 \leq s(x_2) \leq 6$ et le terme $y_{2,0} = x_2$ doit interférer avec $y_{1,1} = \mathtt{1}$ et s'additionner pour donner une puissance de 2. Alors, $x_2 = \mathtt{1}^j$ pour $4 \leq j \leq 6$ et donc $y_{2,2} = \mathtt{1}^{j-1}\mathtt{0}^j\mathtt{1}$. Ce terme peut uniquement interférer avec un des $y_{3,1} = x_3$ et $y_{3,0} = x_3$ et on voit que ces contributions donnent encore une fois plus qu'un seul 1-bit dans tous les cas.

(4) $\underline{s(y_{0,0}) \geq 2}$. Ces cas sont symétriques aux cas précédents.

Soit $\boxed{k = 11}$. Alors au moins trois des sept termes indépendants doivent contenir un seul 1-bit. On distingue les cas suivants comme pour le cas précédent $k = 10$.

(1) $\underline{x_3 = x_0 = 1.}$ On a $y_{3,2} = x_2$ et $y_{1,0} = x_1$, Ainsi, $s(n^2) > s(y_{3,3}) + s(y_{3,2}) + s(y_{1,0}) + s(y_{0,0}) = 11$, c'est une contradiction.

(2) $\underline{s(y_{3,3}) = 4.}$ Comme $s(x_3) \leq 8$ les valeurs possibles pour $(x_3)_2$ sont $\mathtt{101111}$, $\mathtt{1101111}$, $\mathtt{1111}$ et $\mathtt{1101}$ (voir [HLS11b]). De plus, puisque dans ce cas on a $s(y_{3,2}) \geq 2$, cela implique que $x_1 = x_0 = 1$ et $3 \leq s(x_2) \leq 6$. Le terme $y_{2,0} = x_2$ doit interférer avec $y_{1,1} = \mathtt{1}$ et cette contribution doit être exactement un 1-bit. Ainsi, $x_2 = \mathtt{1}^i$ pour $3 \leq i \leq 6$ et $y_{2,2} = \mathtt{1}^{i-1}\mathtt{0}^i\mathtt{1}$. Ces termes peuvent seulement interférer avec un des $y_{3,1} = x_3$ et $y_{3,0} = x_3$ et on peut voir que ces contributions auront toutes plus qu'un seul 1-bit dans tous les cas.

(3) $\underline{s(y_{3,3}) = 3.}$ Les valeurs possibles de x_3 sont $\mathtt{10111}$, $\mathtt{111}$ et $\mathtt{10}^i\mathtt{1}$ pour un certain $i \geq 1$. De plus, $x_0 = 1$ car un des deux termes $y_{0,0}$ et $y_{1,0}$ est $\mathtt{1}$. On distingue les cas suivants en fonction du poids de x_1.

(a) Si $s(x_1) = 2$, alors $4 \leq s(x_2) \leq 6$. Le terme $y_{2,0} = x_2$ doit interférer avec $y_{1,1}$ et contribuer avec un seul 1-bit. Comme $(x_1)_2 = \mathtt{10}^j\mathtt{1}$ pour un certain $j \geq 0$, on

a $y_{1,1}$ égal à $\mathtt{1001}$ ou $\mathtt{10}^{j-1}\mathtt{10}^{j+1}\mathtt{1}$ pour un certain $j \geq 1$. Un calcul rapide de toutes les interférences possibles montre qu'on a les cas suivants

$s(x_2)$	x_2
4	$\{\mathtt{10111}, \mathtt{100111}\}$
5	$\{\mathtt{110111}, \mathtt{1100111}, \mathtt{101111}\}$
6	$\{\mathtt{1110111}, \mathtt{11100111}, \mathtt{10101111}\}$

En calculant chaque valeur de $y_{2,2}$, on peut vérifier toutes les interférences possibles entre $y_{2,2}$ et $y_{3,1}$ ou $y_{3,0} = x_3$. Dans tous les cas le résultat a plus qu'un seul 1-bit et on aboutit à une contradiction.

(b) Si $x_1 = 1$, alors $5 \leq s(x_2) \leq 7$. Supposons dans un premier temps que le terme $y_{2,0} = x_2$ interfère avec $y_{1,1} = 1$ et les deux termes s'ajoutent pour donner une puissance 2. Alors $(x_2)_2 = \mathtt{1}^j$ pour $5 \leq j \leq 7$ et $s(y_{3,2}) \geq 3$. Le terme $y_{2,2} = \mathtt{1}^{j-1}\mathtt{0}^j\mathtt{1}$ peut uniquement interférer avec un des $y_{3,1} = x_3$ et $y_{3,0} = x_3$ et cette contribution est plus grande qu'un seul 1-bit.
Dans un deuxième temps, si $y_{2,0} = x_2$ interfère avec $y_{1,1} = 1$ et donne une contribution de la forme $\mathtt{10}^j\mathtt{1}$, alors x_2 est de la forme $\mathtt{11110}^{j-4}\mathtt{1}$, $\mathtt{111110}^{j-5}\mathtt{1}$ ou $\mathtt{1111110}^{j-6}\mathtt{1}$. On calcule $y_{2,2}$ dans tous ces cas et on conclut que les termes contribuent à plus qu'un seul 1-bit quand ils interagissent avec x_3.

(4) $s(y_{3,3}) = 2$. Alors on a $x_3 = \mathtt{11}$.

Supposons dans un premier temps que $x_0 = \mathtt{1}$. On a alors les cas suivants :

$s(y_{3,2})$	$s(y_{1,0})$	x_2	$y_{2,2}$
2	3	$(\mathtt{10})^3\mathtt{11}$	$\mathtt{111001000111001}$
2	2	$(\mathtt{10})^4\mathtt{11}$	$\mathtt{1110001111000111001}$
2	1	$(\mathtt{10})^5\mathtt{11}$	$\mathtt{11100011100111000111001}$
3	2	$(\mathtt{10})^4\mathtt{11}$	$\mathtt{1110001111000111001}$
3	1	$(\mathtt{10})^5\mathtt{11}$	$\mathtt{11100011100111000111001}$
4	1	$(\mathtt{10})^5\mathtt{11}$	$\mathtt{11100011100111000111001}$

Dans chaque cas, $y_{2,2}$ peut interférer avec un des $y_{3,1} = x_3x_1$ et $y_{3,0} = \mathtt{11}$. Dans ces deux cas, cette interférence est toujours plus qu'un seul 1-bit à cause de la sous-multiplicativité de la fonction somme des chiffres, voir le premier chapitre pour plus de détails. En effet, la sous-multiplicativité implique que $s(x_3x_1) \leq s(x_3)s(x_1) \leq 6$ et ce n'est pas suffisant pour annuler tous les 1-bits intérieurs dans $y_{2,2}$.

Dans un second temps, on suppose que $x_0 \neq 1$. Alors on a $s(y_{0,0}) = s(y_{1,0}) = s(y_{3,2}) = 2$ et donc $x_0 = \mathtt{11}$, $x_1 = (\mathtt{10})^i\mathtt{11}$ et $x_2 = (\mathtt{10})^j\mathtt{11}$ pour des certains entiers positifs i, j et tels que $i + j = 3$. (Le cas $x_1 = \mathtt{1}$ et/ou $x_2 = \mathtt{1}$ est plus simple et peut être traité de la même manière.) Par symétrie on peut supposer que $i < j$.

(a) Si $j = 3$, alors $x_2 = \mathtt{10101011}$ et $x_1 = \mathtt{11}$. Alors on a $y_{2,2} = \mathtt{111001000111001}$. Ce terme peut interférer avec un des $y_{3,1} = \mathtt{1001}$ et $y_{3,0} = \mathtt{1001}$, mais cette contribution a toujours plus qu'un 1-bit.

(b) Si $j = 2$, alors $x_2 = \mathtt{101011}$ et $x_1 = \mathtt{1011}$. Alors on a $y_{2,2} = \mathtt{11100111001}$. Ce terme peut interférer avec un des $y_{3,1} = \mathtt{100001}$ et $y_{3,0} = \mathtt{1001}$, mais cette contribution a toujours plus qu'un 1-bit.

Ainsi la preuve du Lemme VI.4.1 est complète pour le cas $m = 3$.

VI.4.4. Le cas $m = 4$. Dans ce cas, il y a neuf termes indépendants $y_{4,4}, \ldots, y_{4,0}, \ldots, y_{0,0}$ chacun contribuant n^2 avec au moins un 1-bit et au plus $(k-8)$ 1-bits, voir la Figure VI.3. Ce cas a plus de restrictions sur les termes indépendants et est donc plus simple à étudier de ce point de vue. Mais d'un autre côté nous allons devoir étudier plus profondément les interactions dans le graphe d'interférence ce qui s'avère plus difficile que précédemment.

Soit $\boxed{k = 9}$. C'est le cas plus simple car chaque terme indépendant doit contribuer à n^2 avec exactement un seul 1-bit. On a alors $y_{4,4} = y_{4,3} = y_{1,0} = y_{0,0} = 1$ et cela implique directement $x_4 = x_3 = x_1 = x_0 = 1$ et $s(x_2) = 5$. Le terme $y_{4,2} = x_2$ peut uniquement interférer avec $y_{3,3} = 1$ et le résultat de l'addition de ces deux termes doit être une puissance de 2. C'est uniquement possible si $x_2 = 11111$. Maintenant, le terme $y_{2,2} = 111100001$ peut interférer avec deux des termes suivants $y_{3,1}, y_{4,1}, y_{3,0}$ et $y_{4,0}$. Comme ces quatre termes sont tous égaux à 1, la contribution à n^2 aura plus qu'un seul 1-bit et donc ce cas est terminé.

Soit $\boxed{k = 10}$. Parmi les neuf termes indépendants, seulement un seul peut contribuer à hauteur de deux 1-bits. On obtient immédiatement que $x_0 = x_4 = 1$ et un des x_1 et x_3 est 1. On distingue alors les cas suivants.

(1) $\underline{x_3 \neq 1}$ (ou symétriquement $x_1 \neq 1$). On voit que $s(x_3) = s(y_{4,3}) = 2$ et $s(x_2) = 5$. Le terme $y_{2,0} = x_2$, peut uniquement interférer avec $y_{1,1} = 1$ et doit contribuer avec uniquement un 1-bit. Ainsi, $x_2 = 11111$ et donc on a $y_{2,2} = 1111000001$. Ce terme peut interférer avec deux des termes $y_{3,1} = x_3, y_{4,1} = 1, y_{3,0} = x_3$, et $y_{4,0} = 1$. Cette contribution à n^2 aura plus qu'un seul 1-bit.

(2) $\underline{x_3 = x_1 = 1.}$ On a $s(x_2) = 6$ et le terme $y_{2,0} = x_2$ peut seulement interférer avec $y_{1,1} = 1$ et doit contribuer avec un 1-bit seulement. En effet, s'il contribue avec plus qu'un seul 1-bit, par symétrie $y_{4,2}$ va interférer avec $y_{3,3} = 1$ avec plus qu'un 1-bit et on obtient une contradiction. En fait, si $x_2 \neq 111111$, alors le terme $y_{2,0} = x_2$ (resp. $y_{4,2} = x_2$), qui peut seulement interférer avec $y_{1,1} = 1$ (resp. $y_{3,3} = 1$) contribue avec plus qu'un seul 1-bit. Ainsi, $x_2 = 111111$ et le terme $y_{2,2} = 111110000001$ peut interférer avec deux des termes $y_{3,1} = 1$, $y_{4,1} = 1$, $y_{3,0} = 1$, et $y_{4,0} = 1$ et sa contribution à n^2 doit être au plus de deux 1-bits. Il existe une solution pour ces conditions puisque le terme peut être décalé contre un autre. Cependant, comme nous allons le voir, la contradiction survient lorsque l'on essaie de trouver une solution pour $\hat{\ell}_1, \ldots, \hat{\ell}_4$, où

$$\hat{\ell}_j = \sum_{i=1}^{j} \ell_i + |x_{i-1}|,$$

voir le début de le Paragraphe VI.3 pour la notation. Les paires suivantes de termes doivent interférer l'un avec l'autre tel que leurs contributions ont seulement un 1-bit :

$$(y_{2,0}, y_{1,1}),\ (y_{3,0}, y_{2,1}),\ (y_{4,1}, y_{3,2}),\ (y_{4,2}, y_{3,3}).$$

Plus précisément, leurs 1-bits de poids faible doivent s'aligner et donne donc le système suivant :

$$\begin{cases} \hat{\ell_2} + 1 = 2\hat{\ell_1} \implies \hat{\ell_2} = 2\hat{\ell_1} - 1, \\ \hat{\ell_3} + 1 = \hat{\ell_2} + \hat{\ell_1} + 1 \implies \hat{\ell_3} = 3\hat{\ell_1} - 1, \\ \hat{\ell_4} + \hat{\ell_1} + 1 = \hat{\ell_3} + \hat{\ell_2} + 1 \implies \hat{\ell_4} = 4\hat{\ell_1} - 2, \\ \hat{\ell_4} + \hat{\ell_2} + 1 = 2\hat{\ell_3}. \end{cases}$$

Rappellons que $\hat{\ell_0} = 0$. Donc, les termes $y_{2,2}, y_{3,1}$, et $y_{4,0}$ s'alignent de la manière suivante

$$\begin{array}{r} \texttt{111110000001} \\ \texttt{1} \\ \texttt{1} \end{array}$$

et leur contributions à n^2 est `111110000111` qui contient trop de chiffres.

Soit $\boxed{k = 11}$. On distingue les cas suivants :

(1) $\underline{x_4 \neq 1.}$ Le terme $y_{4,4}$ et $y_{4,3}$ doivent contenir deux 1-bits et tous les autres termes contribuent seulement à hauteur d'un seul 1-bit. Alors $x_4 = \texttt{11}$ et $x_1 = x_0 = 1$. Comme $s(y_{4,3}) = 2$, on a $x_3 = (\texttt{10})^i\texttt{11}$ pour un certain $0 \leq i \leq 4$. De plus $y_{2,0} = x_2$ peut seulement interférer avec $y_{1,1}$ et ce terme doit contribuer avec un seul 1-bit et donc $x_2 = \texttt{1}^k$ pour un certain $1 \leq k \leq 5$. Alors le terme $y_{4,2}$ peut seulement interférer avec $y_{3,3}$ et $\sum_{0 \leq i \leq 4} s(x_i) = 11$ implique $i + k = 5$. Par conséquent les valeurs possibles pour le couple $(y_{4,2}, y_{3,3})$ sont données par le tableau suivant :

$y_{4,2}$	$y_{3,3}$
`1011101`	`1001`
`101101`	`1111001`
`10101`	`11100111001`
`1001`	`111001000111001`
`11`	`1110001111000111001`

Cela donne une contribution à n^2 avec plus qu'un seul 1-bit pour tous les cas.

(2) $\underline{x_0 \neq 1.}$ Ce cas est symétrique au cas précédent.

(3) $\underline{x_4 = x_0 = 1.}$ Par symétrie on peut supposer que $s(x_3) \geq s(x_1)$. Chacun des neuf termes indépendants contribuent au moins à hauteur d'un 1-bit pour n^2. Une inspection directe de $y_{4,3}$ donne $s(x_3) \leq 3$. En regardant $y_{1,0} = x_1$, on a de la même manière $s(x_1) \leq 3$. Si $s(x_3) = 3$, alors $s(x_1) = 1$. On doit alors distinguer les 4 cas suivants :

$s(x_3)$	$s(x_1)$	$s(x_2)$
3	1	5
2	2	5
2	1	6
1	1	7

Pour la première ligne de ce tableau, à part pour $y_{4,3} = x_3$, toutes les autres contributions doivent être exactement d'un seul 1-bit. Or $y_{2,0} = x_2$ peut seulement interférer avec $y_{1,1} = 1$, on a donc $x_2 = 11111$. De plus, $y_{4,2} = x_2$ peut seulement interférer avec $y_{3,3} = x_3^2$ et cette contribution doit être d'un seul 1-bit. Cela arrive si et seulement si $x_3^2 = 1 \cdots 100001$, i.e. x_3^2 correspond à l'entier $1 + 2^5(2^\lambda - 1)$ pour un certain $\lambda \geq 1$. On peut alors résoudre ceci comme dans le cas $m = 2, k = 11$ (c), ou directement avec le calcul suivant; on écrit $x_3 = 1 + 2^a + 2^b$ pour un certain $0 < a < b$ et on a

$$\begin{aligned} x_3^2 &= 1 + 2^{a+1} + 2^{b+1} + 2^{2a} + 2^{a+b+1} + 2^{2b}, \\ &= 1 + 2^{a+1}(1 + 2^{b-a} + 2^{a-1} + 2^b + 2^{2b-a-1}). \end{aligned}$$

Donc on a $a = 4$ et toutes les autres puissances de 2 doivent être successives. En particulier, cela implique $b - a = 2$ et $b = 3$. Ce n'est pas possible puisque $a < b$.

Pour la deuxième ligne, $x_1 = 10^i1$ pour un certain $i \geq 0$. Alors $y_{1,1} = 1001$ pour $i = 0$ ou $10^{i-1}10^{i+1}1$ pour $i \geq 1$. Encore une fois, le terme $y_{2,0} = x_2$ peut seulement interférer avec $y_{1,1}$ et cette contribution a plus qu'un seul 1-bit sauf pour les cas $(x_1, x_2) = (11, 110111)$ et $(101, 1100111)$. Pour $i \geq 2$, avoir une contribution d'un seul 1-bit implique $s(x_2) \geq 6$, et les blocs de 0-bits de $y_{1,1}$ sont trop grands pour être recouverts. Alors $y_{4,2} = x_2$ peut seulement interférer avec $y_{3,3}$ avec un seul 1-bit et on a le même résultat qu'avant. La valeur de x_2 fixe les valeurs de x_1 et x_3, et on doit étudier les deux cas suivants en fonction du poids de x_2.

(a) Si $x_2 = 110111$, alors on a $x_1 = x_3 = 11$. Cela implique $y_{3,2} = 10100101$ et $y_{3,2}$ peut interférer avec $y_{4,1} = x_1$ et $y_{4,0} = x_0$. Dans tous les cas, la contribution a plus qu'un seul 1-bit.

(b) Si $x_2 = 1100111$, alors on a $x_1 = x_3 = 101$. Ceci implique $y_{3,2} = 1000000011$ et $y_{3,2}$ peut interférer avec $y_{4,1} = x_1$ et $y_{4,0} = x_0$. Dans tous les cas, la contribution a plus qu'un seul 1-bit.

Pour la troisième ligne, on a $y_{2,0} = x_2$ qui peut seulement interférer avec $y_{1,1} = x_1^2 = 1$ avec au plus deux 1-bits. On écrit $x_3 = 10^i1$ pour un certain $i \geq 0$. On distingue deux cas en fonction de cette contribution.

(a) Si cette contribution a un seul 1-bit alors on a $x_2 = 111111$. Comme $y_{4,2} = x_2$ peut seulement interférer avec $y_{3,3} = x_3^2$ et cette contribution a au plus deux 1-bits. Ceci implique $x_3 \in \{11, 1001, 100001, 10000001\}$, i.e $i \in \{0, 2, 4, 6\}$. Pour voir ceci, si $i \geq 8$, les blocs de 0-bits sont trop grands dans x_3^2 et on peut vérifier que les autres valeurs possibles de i directement. Dans tous ces cas la contribution est exactement de deux 1-bits. Le terme $y_{3,2} = x_3x_2$ peut seulement interférer avec $y_{4,1} = 1$ et $y_{4,0} = 1$. Comme $y_{3,2} \in \{10111101, 1000110111, 100000011111, 1111110111111\}$, toutes ces contributions ont plus qu'un seul 1-bit.

(b) Supposons maintenant que la contribution entre $y_{2,0}$ et $y_{1,1}$ est exactement de deux 1-bits. On a alors $x_2 = 1011111$ ou $x_2 = 1111101$. De plus, $y_{4,2} = x_2$ peut seulement interférer avec $y_{3,3} = x_3^2$ et cette contribution doit être exactement un 1-bit. Cependant, dans les deux cas , cette contribution dépasse un 1-bit, par un argument similaire à avant pour les différentes valeurs de i.

Pour la dernière ligne, le terme $y_{2,0} = x_2$ peut seulement interférer avec $y_{1,1} = x_1$ avec au plus trois 1-bits. Comme dans le cas précédent, on a une contribution d'un seul 1-bit si et seulement si $x_2 = \mathtt{1111111}$, resp., une contribution de deux 1-bits si et seulement si

$$x_2 \in \{\mathtt{10}^a\mathtt{111111} : \ a \geq 1\} \ \cup \ \{\mathtt{1111110}^a\mathtt{1} : \ a \geq 1\} := B_2,$$

resp., une contribution de trois 1-bits si et seulement si

$$x_2 \in \{\mathtt{10}^{a'}\mathtt{10}^a\mathtt{11111} : \ a, a' \geq 1\} \ \cup \ \{\mathtt{10}^{a'}\mathtt{111110}^a\mathtt{1} : \ a, a' \geq 1\}$$
$$\cup \ \{\mathtt{111110}^{a'}\mathtt{10}^a\mathtt{1} \ : \ a, a' \geq 1\}.$$

On note la dernière union B_3. On distingue les cas en fonction de quel ensemble appartient x_2.

(a) Si $x_2 \in B_3$, alors $y_{4,2}$ peut seulement interférer avec $y_{3,3} = 1$ et en un seul 1-bit. Ce n'est pas possible car $y_{4,2} = x_2$.

(b) Si $x_2 \in B_2$, alors $y_{4,2} = x_2$ peut seulement interférer avec $y_{3,3} = 1$ avec au plus deux 1-bits. Si cette contribution ne dépasse pas deux bits, elle doit être exactement de deux 1-bits à cause de la forme de l'écriture binaire de x_2. Ceci implique que le terme $y_{3,2} = x_2$ peut interférer avec $y_{4,1} = 1$ et $y_{4,0} = 1$ avec au plus un 1-bit. Ce n'est pas possible.

(c) Si $x_2 = \mathtt{11111111}$ alors on a $y_{2,2} = \mathtt{11111100000001}$ qui peut interférer avec $y_{4,1}$, $y_{4,0}$, $y_{3,1}$ et $y_{3,0}$ et toutes ces contributions doivent valoir 1 à la fin. Cependant, toutes ces contributions ont plus qu'un seul 1-bit car le bloc de 0-bits est trop grand.

Ainsi la preuve du Lemme VI.4.1 est complète pour le cas $m = 4$.

VI.4.5. Le cas $m = 5$. Dans ce cas on a nécessairement $\boxed{k = 11}$. Il y a onze termes $y_{5,5}, \ldots, y_{5,0}, \ldots, y_{0,0}$ chacun contribuant à n^2 avec exactement un 1-bit. Nous n'avons pas tracé le graphe d'interférence car il est trop grand et n'est pas nécessaire pour la compréhension des arguments ci-dessous. Comme $y_{5,5} = y_{5,4} = y_{1,0} = y_{0,0} = 1$, alors $x_5 = x_4 = x_1 = x_0 = 1$, le terme $y_{5,3} = x_3$ peut seulement interférer avec $y_{4,4} = 1$ et le résultat de l'interférence de ces termes doit être une puissance de deux. Le même raisonnement est également vrai pour $y_{2,0}$ qui peut seulement interférer avec $y_{1,1} = 1$. Par conséquent, on a $x_3 = \mathtt{1}^{n_3}$ et $x_2 = \mathtt{1}^{n_2}$ pour un certain $n_3 \geq 1$ et $n_2 \geq 1$ tel que $n_3 + n_2 = 7$. On distingue les cas en fonction de la valeur de n_3.

(1) Si $n_3 = 6$, i.e. $x_3 = \mathtt{111111}$, alors $y_{3,3} = \mathtt{111110000001}$ peut interférer avec trois des termes $y_{5,2}, y_{5,1}, y_{5,0}, y_{4,2}, y_{4,1}$, et $y_{4,0}$. Comme chacun de ces six termes est 1, cette contribution à n^2 a plus qu'un seul 1-bit puisque le bloc de 0-bits de $y_{3,3}$ contient six 0-bits consécutifs.

(2) Si $n_3 = 5$, i.e. $x_3 = \mathtt{11111}$ et $x_2 = \mathtt{11}$. Alors $y_{3,3} = \mathtt{1111000001}$ peut interférer avec trois des termes $y_{5,2} = y_{4,2} = \mathtt{11}$, $y_{5,1} = y_{5,0} = y_{4,1} = y_{4,0} = 1$. On conclut de la même manière que précédemment.

(3) Si $n_3 = 4$, i.e $x_3 = \mathtt{1111}$ et $x_2 = \mathtt{111}$. Alors $y_{3,3} = \mathtt{11100001}$ peut interférer avec trois des termes $y_{5,2} = y_{4,2} = \mathtt{111}$, $y_{5,1} = y_{5,0} = y_{4,1} = y_{4,0} = 1$. Cette fois, il existe

une solution que l'on représente comme suit :

$$\begin{array}{r} \texttt{1111100001} \\ \texttt{111} \\ \texttt{1} \\ \texttt{1} \end{array}$$

Ici on doit étudier d'autres interférences pour obtenir une contradiction. Les paires suivantes de termes doivent interférer et avec un seul 1-bit : $(y_{2,0}, y_{1,1})$, $(y_{3,0}, y_{2,1})$, $(y_{5,2}, y_{4,3})$, $(y_{5,3}, y_{4,4})$. Plus précisément, leurs chiffres significatifs doivent s'aligner et donnent donc le système suivant :

$$\begin{cases} \hat{\ell_2} + 1 = 2\hat{\ell_1}, \\ \hat{\ell_3} + 1 = \hat{\ell_2} + \hat{\ell_1} + 1, \\ \hat{\ell_5} + \hat{\ell_2} + 1 = \hat{\ell_4} + \hat{\ell_3} + 1, \\ \hat{\ell_5} + \hat{\ell_3} + 1 = 2\hat{\ell_4}. \end{cases}$$

Ceci implique $\hat{\ell_2} = 2\hat{\ell_1} - 1$, $\hat{\ell_3} = 3\hat{\ell_1} - 1$, $\hat{\ell_4} = 4\hat{\ell_1}$ et $\hat{\ell_5} = 5\hat{\ell_1}$. Alors le terme $y_{3,3}$ ne s'aligne pas correctement avec les autres termes. En fait, comme on cherche des contributions d'un seul 1-bit , on a besoin d'aligner sur le premier bit. Ici ce n'est pas le cas car toutes les valeurs suivantes sont différentes

$$\begin{cases} 2\hat{\ell_3} = 6\hat{\ell_1} - 2, \\ \hat{\ell_5} + \hat{\ell_2} + 1 = 7\hat{\ell_1}, \\ \hat{\ell_4} + \hat{\ell_2} + 1 = 6\hat{\ell_1}, \\ \hat{\ell_5} + \hat{\ell_1} + 1 = 6\hat{\ell_1} + 1, \\ \hat{\ell_5} + 1 = 5\hat{\ell_1} + 1, \\ \hat{\ell_4} + \hat{\ell_1} + 1 = 5\hat{\ell_1} + 1, \\ \hat{\ell_4} + 1 = 4\hat{\ell_1} + 1. \end{cases}$$

Et cela nous fournit donc une contradiction.

(4) Si $1 \leq n_3 \leq 3$, alors on a $x_2 = \texttt{1}^{n_2}$ pour un certain $4 \leq n_2 \leq 6$, ce qui est symétrique à un cas précédent.

Ainsi la preuve du Lemme VI.4.1 est complète pour le cas $m = 5$.

Tous les cas $1 \leq m \leq 5$ sont donc démontrés. Par conséquent le Lemme VI.4.1 est prouvé, de même que le Théorème VI.2.1.

VI.5. Sur les équations $s(n^2) \in \{4, 5\}$

L'objectif de cette section est d'étudier les équations $s(n^2) \in \{4, 5\}$ pour des entiers impairs. Comparé aux sections précédentes, le point de vue est différent car il n'y a pas de condition sur le poids de n. Comme il n'y a *a priori* pas de conditions sur le poids de n, une simple recherche par ordinateur n'est pas suffisante pour déterminer s'il y a un nombre fini de solutions ou non. Notre but est alors de résoudre ces équations pour tout n composé du plus de 1-bits possible. Les heuristiques indiquent que plus le poids de n est grand, moins n a de chance d'être solution à $s(n^2) \leq 5$. En effet, si n est de

poids grand, il faut de plus en plus de propagations de retenues qui annulent ses 1-bits. Au vue de la Conjecture VI.2.2, sous cette heuristique, cela montre qu'il est de plus en plus improbable de trouver de nouvelles solutions données par cette conjecture.

VI.5.1. L'équation $s(n^2) = 4$. Soit n un entier impair avec $s(n) = k \geq 9$ tel que $s(n^2) = 4$. Tout d'abord on suppose que $k \geq 9$ puisque tous les autres cas sont déjà résolus dans [HLS11b], mais nos algorithmes donneront aussi les mêmes résultats pour les plus petites valeurs de k. On écrit $n = 1 + 2^\ell m$ avec $\ell \geq 1$ et m un entier impair qui satisfait $s(m) = k - 1$. Donc on a $n^2 = 1 + 2^{\ell+1}m + 2^{2\ell}m^2$ et cela implique $s(2^{\ell+1}m + 2^{2\ell}m^2) = 3$. Par conséquent, on a

$$s(m + 2^{\ell-1}m^2) = 3. \tag{VI.5.1}$$

Deux angles d'attaque sont possibles pour résoudre ce problème. Le premier est d'utiliser le Lemme VI.3.8, plus précisément, un cas spécifique dans la preuve de ce résultat donne une borne supérieure meilleure au cas général, voir [KS22]. Cela permet d'obtenir la borne suivante :

$$m(1 + 2^{\ell-1}m) \leq 2^{k(k-1)-13}, \tag{VI.5.2}$$

et on obtient $m < 2^{k(k-1)-(\ell-1)/2-2}$. Ceci implique à son tour une borne sur ℓ et il reste à utiliser l'algorithme `next` pour chaque ℓ possible pour trouver l'ensemble des solutions. Cependant, cette méthode n'est pas suffisante pour $s(n^2) = 5$. C'est la raison principale de la création de l'algorithme `max-integer` que nous décrivons brièvement dans la suite (nous donnons une description plus détaillée dans le Paragraphe VI.6).

À partir de (VI.5.1) nous devons allouer trois 1-bits dans la somme $S = m + 2^{\ell-1}m^2$. Pour se faire, on utilise le fait basique que si deux entiers a, b vérifient $a \equiv b \pmod{2^\lambda}$ alors $a^2 \equiv b^2 \pmod{2^{\lambda+1}}$ pour tout $\lambda \geq 2$ et si $a \equiv b \pmod 2$, alors $a^2 \equiv b^2 \pmod 8$. En fait, ce phénomène peut se traduire par le fait que l'élévation au carré améliore les congruences modulo les puissances de 2. De cette manière, si on écrit la décomposition binaire de m bit par bit à partir des chiffres les moins significatifs vers les plus significatifs, on peut en déduire par la même occasion la décomposition de $2^{\ell-1}m^2$ bit par bit également des chiffres les moins significatifs vers les plus significatifs. Alors l'algorithme teste si le prochain bit dans la décomposition binaire de m peut être un 1-bit ou un 0-bit pour vérifier l'équation (VI.5.1). Comme on va supposer une borne sur le poids de m, l'algorithme s'arrêtera lorsque le bon montant de 1-bits est atteint.

Illustrons un exemple lorsque l'on suppose $\ell = 1$ et les $\ell(m)$ chiffres les moins significatifs de S est 0, i.e. toute la partie droite de S ne contient que des 0. Le premier chiffre (le plus à droite) que l'on ajoute dans la structure binaire de m est un 1-bit dans le but de faire une propagation de retenues. On déduit alors le deuxième bit m^2, même si ici il était déjà déterminé en réalité, c'est un 0-bit. Comme on a supposé qu'il n'y a pas de 1-bits sur la partie basse de la somme S, le troisième bit m est nécessairement un 1-bit, et on en déduit donc que $m \equiv 7 \pmod 8$. Par conséquent, $m^2 \equiv 1 \pmod{16}$, et le quatrième bit de m^2 est un 0-bit. En itérant cet argument, on voit qu'on ne peut ajouter que 1-bits dans m et on obtient que la somme suivante avec un bloc de $(k-1)$ 1-bits dans m (comme avant, on écrit (*) pour une chaîne de caractère finie de bits).

$$\begin{array}{rccccccc} & & 1 & \cdots & 1 & 1 & 1 & = m \\ + & (*) & 0 & \cdots & 0 & 0 & 1 & = m^2 \\ \hline & (*) & 0 & \cdots & 0 & 0 & 0 & = S. \end{array}$$

Donc, la structure de m est totalement déterminée en fonction de ce qu'on a exigé comme structure sur $\ell(m)$ premiers chiffres les moins significatifs de S. L'algorithme considère également les cas où la partie droite de S contient un ou deux 1-bits, voir Paragraphe VI.6. Le cas précédent donne l'idée principale de l'algorithme, à l'aide de l'amélioration des congruences de l'élévation au carré, nous en déduisons bit par bit la structure de m^2 et de S en fonction de m.

Pour que l'algorithme soit efficace, il est nécessaire de trouver de bonnes bornes pour ℓ. Le lemme suivant donne une borne bien plus forte sur ℓ que celle donnée par (VI.5.2).

Lemme VI.5.1. *Soit n un entier impair tel que $s(n) = k \geq 4$, $s(n^2) = 4$ et $n = 1 + 2^\ell m$ avec $\ell \geq 2$ et m un entier impair. Alors on a $\ell \leq 2k$.*

Preuve. Supposons que $\ell > 2k$ et posons $S = m + 2^{\ell-1}m^2$. Alors $s(S) = 3$ et S est un entier impair. On considère alors l'addition suivante (ω, ω' sont des représentations binaires d'entiers et $\varepsilon_i \in \{0, 1\}$) :

$$\begin{array}{rccccccccc} & & & \omega & \varepsilon_{\ell-3} & \cdots & \varepsilon_0 & 1 & = m \\ + & (*) & \omega' & 1 & & & & & = 2^{\ell-1}m^2 \\ \hline & (*) & (*) & (*) & \varepsilon_{\ell-3} & \cdots & \varepsilon_0 & 1 & = S. \end{array}$$

Le bloc $\varepsilon_{\ell-3} \cdots \varepsilon_0$ est composé d'au plus un 1-bit car S contient trois 1-bits et les propagations de retenues ne vont que vers les chiffres les plus significatifs.

On distingue deux cas en fonction des $\ell(m)$ chiffres les moins significatifs de S. Notons que cette partie contient bien la contribution de ω à partir des premiers chiffres m et ses interférences avec les 1-bit les plus à droite de $2^{\ell-1}m^2$. Cette partie va être appelée *partie droite de S* dans la suite. Elle contient au moins un 1-bit (le bit de parité), et au plus deux 1-bits (qui inclut le bit de parité). Elle ne peut pas contenir trois 1-bits car le second terme a une écriture binaire strictement plus grande que le premier. On écrit w l'entier dont la décomposition binaire corresponds à ω. On distingue alors les deux cas suivants en fonction du nombre de bits dans la partie droite de S.

(1) La partie droite S contient seulement le bit de parité 1. Ceci implique que $\varepsilon_i = 0$ pour tout $0 \leq i \leq \ell - 3$. Donc $m = 1 + 2^{\ell-1}w$ et $m^2 = 1 + 2^\ell w + 2^{2\ell-2}w^2$. Alors les ℓ chiffres les moins significatifs de m^2 sont tous des 0-bits, à l'exception du bit de parité. On a alors la somme suivante

$$\begin{array}{rccccccccccc} & & & & & & \omega & 0 & \cdots & 0 & 1 & = 1 + 2^{\ell-1}w \\ + & & & \omega & 0 & \cdots & 0 & 1 & & & & = 2^{\ell-1}(1 + 2^\ell w) \\ + & \omega^2 & 0 & \cdots & 0 & \cdots & 0 & 0 & & & & = 2^{\ell-1} \times 2^{2\ell-2}w^2 \\ \hline & & (*) & (*) & & (*) & & 0 & \cdots & 0 & 1 & = S. \end{array}$$

Désormais, considérons les additions de ω et 1 dans la partie du milieu entre le premier et le deuxième terme. Par hypothèse on a $\ell > 2k > k$, alors le mot ω dans le deuxième terme ne peut pas interférer avec le ω du premier terme. Ceci implique que ω est un unique bloc de $(k-1)$ 1-bits consécutifs sinon la propagation de retenues

ne va pas assez loin. Ceci implique directement que $m = 1 + 2^{\ell-1}(2^{k-1} - 1)$. Alors $m^2 = 1 + 2^{\ell}(2^{k-1} - 1) + 2^{2\ell-2}(2^{2k-2} - 2^k + 1)$, et on a

$$\begin{aligned} m + 2^{\ell-1}m^2 &= 1 + 2^{\ell-1}(2^{k-1} - 1) + 2^{\ell-1} + 2^{2\ell-1}(2^{k-1} - 1) + 2^{3\ell-3}(2^{2k-2} - 2^k + 1) \\ &= 1 + 2^{\ell-2+k} + 2^{2\ell-1}(2^{k-1} - 1) + 2^{3\ell-3}(2^{2k-2} - 2^k + 1). \end{aligned}$$

Or $\ell > 2k$, les termes dans la somme ci-dessus sont non-interférents. Alors $m+2^{\ell-1}m^2$ contient trop de 1-bits.

(2) La partie droite de S contient deux 1-bits isolés. On distingue deux cas en fonction de la localisation du deuxième 1-bit puisque le bit de parité est déjà fixé.

Dans un premier temps, si ce 1-bit est localisé dans le bloc $\varepsilon_{\ell-3}\cdots\varepsilon_0$, alors il existe i_0 tel que $\varepsilon_{i_0} = 1$ et $\varepsilon_i = 0$ pour $i \neq i_0$. Alors $m = 1 + 2^{i_0} + 2^{\ell-1}w$, avec $1 \leq i_0 < \ell - 1$, et

$$2^{\ell-1}m^2 = 2^{\ell-1} + 2^{i_0+\ell} + 2^{2i_0+\ell-1} + 2^{2\ell-1}w + 2^{i_0+2\ell-1}w + 2^{3\ell-3}w^2.$$

On distingue désormais les trois cas suivants en fonction de la valeur de i_0 par rapport à ℓ.

(a) Soit $2i_0 \geq \ell$. Alors on a $2^{\ell-1}m^2 = 2^{\ell-1} + 2^{i_0+\ell} + 2^{2\ell-1}w'$ pour un certain entier w'. Avec un argument similaire au cas précédent on obtient

$$m = 1 + 2^{i_0} + 2^{\ell-1}((2^{i_0} - 1) + 2^{i_0+1}(2^{k'} - 1)),$$

pour un certain $k' \geq 0$ avec $k' + i_0 = k - 2$. Cela implique $i_0 \geq \ell/2 > k > k - 2$, qui donne une contradiction.

(b) Soit $\ell/4 < i_0 < \ell/2$. Alors on a $2^{2i_0+\ell-1} < 2^{2\ell}$ et ceci implique que m a la forme

$$m = 1 + 2^{i_0} + 2^{\ell-1}((2^{i_0} - 1) + 2^{i_0+1}(2^{i_0-1} - 1) + 2^{2i_0+1}(2^{k'} - 1)).$$

Cela implique $s(m) \geq 2i_0 > \ell/2 > k$, qui donne une contradiction.

(c) Soit $i_0 \leq \ell/4$. Alors on a $2^{\ell-1}m^2 = 2^{\ell-1} + 2^{i_0+\ell} + 2^{2i_0+\ell-1} + 2^{2\ell-1}w'$ pour un certain entier w' et on obtient

$$m = 1 + 2^{i_0} + 2^{\ell-1}((2^{i_0} - 1) + 2^{i_0+1}(2^{i_0-1} - 1) + 2^{2i_0+1}(2^{\ell-2i_0-1} - 1)).$$

Cela implique $s(m) \geq \ell - 2i_0 \geq \ell/2 > k$, qui donne une contradiction.

Si $\varepsilon_i = 0$ pour tout i, alors on a deux cas possibles. Si w est pair, alors on peut utiliser le même raisonnement que dans le cas (2)(a) puisque S a deux 1-bits isolés. Si w est impair alors on écrit $\omega = \omega' 01^{\lambda}$, avec un mot binaire ω' potentiellement vide et $1 \leq \lambda \leq k - 1$. On peut en réalité supposer que ω' est non vide car le cas $\omega = 1^{k-1}$ est déjà fait dans le cas (1). Par ailleurs, S a deux 1-bits isolés et ω' contient un bloc de 0-bits de longueur au moins ℓ. Ainsi, on peut conclure avec le même argument que précédemment.

□

Notre implémentation de l'algorithme `max-integer` montre que toutes les solutions impaires n tel que $s(n) \leq 17$ sont $(\ell, m) \in \{(3, 2), (1, 7), (1, 23), (1, 55)\}$ et se traduit directement par

$$\bigcup_{\lambda \leq 17} E_{4,\lambda} = \{13,\ 15,\ 47,\ 111\}.$$

Nous retrouvons précisément l'ensemble (VI.2.6) dans le Théorème VI.2.3.

Remarque VI.5.1. Cette méthode peut être étendue pour une plus grande borne sur $s(n)$ et donne une piste de preuve pour la conjecture de [BBM12] si on est capable de trouver une borne sur le poids de n tel que $s(n^2) = 4$.

VI.5.2. L'équation $s(n^2) = 5$. Le lemme suivant implique (VI.2.8) dans le Théorème VI.2.3.

Lemme VI.5.2. *Soit n un entier impair tel que $s(n^2) = 5$. Alors on a les deux assertions suivantes.*
(1) Il existe un nombre fini d'entiers impairs n tel que $s(n) \geq 4$.
(2) Si $s(n) = 3$, alors n est de la forme

$$1 + 2^\ell + 2^{\ell+1}, \quad 1 + 2 + 2^\ell, \quad \text{ou} \quad 1 + 2^\ell + 2^{2\ell-1},$$

pour un certain $\ell \geq 3$.

Preuve. On adapte le Lemme VI.3.2 lorsque le montant de 1-bits dans le carré est fixé à 5. La constante devient alors (i.e. la constante N_k apparaissant dans la preuve) va être différente mais on garde quand même le fait que s'il y a un nombre infini de solutions, alors presque toute les solutions (i.e. toutes sauf un nombre fini) de ces solutions doivent se factoriser de cette manière. On distingue les cas alors encore une fois selon le nombre de blocs m de cette factorisation.

- Soit $m = 1$. Alors on a $(n)_2 = x_1 0 \cdots 0 x_0$, avec un grand bloc de 0-bits intérieur. Par symétrie on peut supposer que $s(x_1) \geq s(x_0)$. Comme on a trois contributions indépendantes x_1^2, $x_1 x_0$ et x_0^2 pour n^2, on voit que exactement un de ces blocs doit contenir un seul 1-bit. Alors $x_0 = 1$ et $s(x_1^2) + s(x_1) = 4$, i.e $x_1 = 3$. Alors n est de la forme $1 + 2^\ell + 2^{\ell+1}$ pour un ℓ suffisamment grand. On peut aisément vérifier que cette forme est valide pour tout $\ell \geq 3$. Par symétrie on obtient une deuxième famille, à savoir $1 + 2 + 2^\ell$ pour $\ell \geq 3$.
- Soit $m = 2$. Alors on utilise le graphe d'interférence donnée dans la Figure VI.1 pour en déduire que $x_2 = x_1 = x_0 = 1$ et que la contribution de $x_1^2 + x_2 x_0$ est uniquement d'un seul 1-bit. Ceci implique que n est de la forme $1 + 2^\ell + 2^{2\ell-1}$ pour un ℓ suffisamment grand. Comme avant, on vérifie que cette forme est valide pour tout $\ell \geq 3$.

□

Nous montrons désormais comment nous obtenons (VI.2.7) via l'algorithme `max-integer`. La méthode employée ici est similaire au cas précédent $s(n^2) = 4$ mais a quand même quelques différences importantes. Supposons que $k \geq 4$ pour éviter les familles infinies précédentes du Lemme VI.5.2. Soit n un entier impair tel que $s(n) = k \geq 4$ et $s(n^2) = 5$. On écrit $n = 1 + 2^{\ell_1} + 2^{\ell_1+\ell_2} m$ avec m un entier impair avec $s(m) = k - 2$ et $\ell_1, \ell_2 \geq 1$. On a

$$n^2 = 1 + 2^{\ell_1+1} + 2^{2\ell_1} + 2^{\ell_1+\ell_2+1} m + 2^{2\ell_1+\ell_2+1} m + 2^{2\ell_1+2\ell_2} m^2. \tag{VI.5.3}$$

On évalue le nombre de bits isolés et on traite tous les différents cas en fonction des valeurs de ℓ_1 et ℓ_2.

(1) Soit $\ell_1 = 1$. Ici (VI.5.3) devient

$$n^2 = 1 + 2^3 + 2^{\ell_2+2}m + 2^{\ell_2+3}m + 2^{2\ell_2+2}m^2.$$

Deux nouveaux cas émergent :

(a) Soit $\ell_2 > 1$. On a ici deux bits isolés (associés aux puissances 1 et 2^3) et on doit résoudre

$$s(m(3 + 2^{\ell_2}m)) = 3.$$

Par une petite adaptation de l'algorithme `max-integer`, on trouve que les seules solutions sont $\ell_2 = 2, m = 11$ et $\ell_2 = 3, m = 3$. Alors $n = 51$ et $n = 91$ vérifient $s(n^2) = 5$.

(b) Soit $\ell_2 = 1$. On a

$$s(1 + 3m + 2m^2) = s((2m + 1)(m + 1)) = 4.$$

Encore une fois, on adapte l'algorithme `max-integer` et les solutions pour m est l'ensemble

$$\{7,\ 19,\ 23,\ 55,\ 69,\ 119,\ 181,\ 367\}.$$

Alors l'ensemble des solutions pour n est donc

$$\{31,\ 79,\ 95,\ 223,\ 279,\ 479,\ 727,\ 1471\}.$$

Mentionnons que c'est précisément ce cas qui nous a motivé à créer l'algorithme `max-integer` car les résultats de [KS22] ne sont pas suffisants pour conclure et une nouvelle méthode était nécessaire.

(2) Soit $\ell_1 > 1$. On a alors deux bits isolés (associés aux puissances 1 et 2^{ℓ_1+1}). Alors (VI.5.3) devient

$$s(2^{\ell_1} + 2^{\ell_2+1}m + 2^{\ell_1+\ell_2+1}m + 2^{\ell_1+2\ell_2}m^2) = 3.$$

Une dernière adaptation de l'algorithme donne que $n \in \{29,\ 157,\ 5793\}$ comme ensemble de solutions.

En résumé, nous trouvons

$$\bigcup_{4\le\lambda\le15} E_{5,\lambda} = \{29,\ 31,\ 51,\ 79,\ 91,\ 95,\ 157,\ 223,\ 279,\ 479,\ 727,\ 1471,\ 5793\},$$

ce qui démontre (VI.2.7).

Notons que la solution avec le plus grand poids est pour la solution 1471, pour un poids égal à 9.

VI.6. Description des algorithmes

VI.6.1. Algorithme `next`. L'objectif de cet algorithme est d'engendrer efficacement tous les entiers impairs plus petits qu'une certaine borne fixée au préalable et à un poids fixé. Ces ensembles sont nécessaires pour la démonstration du Théorème VI.2.1, à savoir les ensembles

$$\Delta_{\ell_1,\ell_2,m} = \{n \in \mathbb{N} :\quad s(n) = \ell_1,\quad s(n^2) \le \ell_2,\quad n < 2^m,\quad n \text{ impair}\}.$$

Le résultat suivant donne, à partir d'un entier donné, le plus petit entier strictement plus grand que cet entier et de même poids.

Lemme VI.6.1. *Soit $n \geq 1$ un entier. On écrit $(n)_2 = x\mathtt{0}\mathtt{1}^{b+1}\mathtt{0}^c$ pour certains $b, c \geq 0$ et x un mot binaire potentiellement vide. Alors l'entier suivant par ordre croissant, noté m, tel que $s(m) = s(n)$ est $(m)_2 = x\mathtt{1}\mathtt{0}^{c+1}\mathtt{1}^b$.*

Preuve. Il est clair que $s(m) = s(n)$ et supposons qu'il existe un entier p tel que $s(p) = s(m)$ et $n < p \leq m$. Comme $p > n$ et $s(p) = s(n)$, un bit d'indice $\geq c+b+1$ de p est `1` car $\mathtt{1}^{b+1}\mathtt{0}^c$ est le plus grand entier de longueur $c+b+1$ de poids $b+1$. Or $p \leq m$, cet indice est exactement $c+b+1$. Puisque $p \leq m$, l'écriture binaire de p commence avec un `1`-bloc de longueur b. Ceci implique alors $p = m$. □

L'algorithme `next` est une transposition du Lemme VI.6.1. D'après un entier donné n, l'algorithme construit l'entier suivant par ordre croissant de même poids.

Algorithme 2 : `next`

```
1 Fonction next(n):
2 c = index of the least significant set bit n ;
3 n = n/2^c ;
4 n = n + 1 ;
5 b = index of the least significant set bit n -1 ;
6 n = 2^c · n ;
7 n = n|2^b ;                                  /* | is the OR operator */
8 return n = n − 1 ;
```

La deuxième étape consiste à calculer le poids du carrés des entiers d'un ensemble d'entiers de poids fixé. Le programme utilise le fait que pour un entier n de la forme $n = m + 2^L p$ et $m < 2^L$, on a $n^2 = m^2 + 2^{L+1}mp + 2^{2L}p^2$. Alors les $L+1$ plus faibles chiffres significatifs de n^2 sont entièrement déterminés par m, i.e. la partie basse de n. Comme dans notre étude on s'intéresse à des entiers avec le poids de leur carré très faible, on peut dans un premier temps regarder le poids de cette partie basse. En effet si cette partie contient déjà trop de `1`-bits, alors on peut rejeter l'entier sans avoir besoin de calculer entièrement tout le carré de l'entier. Cette astuce réduit drastiquement le temps de calcul car le programme rejette très souvent des entiers de cette manière. Pour des raisons d'efficacité et de praticité, nous avons implémenté l'algorithme avec $L = 64$.

Nous avons parallélisé notre programme et distribué sur plusieurs processeurs avec un suffixe prédéterminé avant de faire la fonction `next`. En effet, pour un entier fixé a, on peut considérer les entiers de la forme $n = 1 + 2^a + 2^{a+1}m$ pour un impair m et pour trouver `next`(n), il est suffisant d'exécuter `next`(m). C'est équivalent à fixer un deuxième `1`-bit dans n. Ce découpage est fait via

$$\Delta_{\ell_1,\ell_2,a,m} = \{n \in \mathbb{N} : s(n) = \ell_1,\ s(n^2) \leq \ell_2,\ \text{ le deuxième bit de } n \text{ est } a,\ n < 2^m,\ n \text{ impair}\}$$

et on a

$$\Delta_{\ell_1,\ell_2,m} = \bigcup_{a=1}^{m-\ell_1+1} \Delta_{\ell_1,\ell_2,a,m}.$$

Ce découpage était nécessaire pour conclure pour le cas $\boxed{k = 11}$ de (VI.1.1), pour $m = 1$. En effet il permet de réduire encore le temps de calcul si l'on utilise suffisamment de processeurs.

Cependant, un autre problème apparaît avec cette parallélisation. Le nombre d'entiers dans $\Delta_{\ell_1,\ell_2,a,m}$ n'est pas équivalent pour toutes les valeurs de a. Plus a est faible, plus le cardinal de $\Delta_{\ell_1,\ell_2,a,m}$ est grand. Après de multiples essais et une heuristique donnée par le fait qu'on utilise les mêmes coefficients binomiaux, nous avons supposé que le temps de calcul pour chaque entier était similaire. Alors avec un calcul préliminaire, on désigne spécifiquement à chaque processeur ce qu'il doit calculer pour qu'à la fin chaque processeur prenne un temps de calcul similaire. De cette manière, nous avons encore réduit le temps de calcul global.

VI.6.2. Algorithme `max-integer`. Dans ce paragraphe, nous décrivons l'algorithme pour l'équation $s(n^2) = 4$, qui peut être écrit comme $s(y + 2^{\ell-1}y^2) = 3$ pour un poids fixé $s(y) = k$ (voir le Paragraphe VI.5.1). Les autres cas étant similaires et ayant seulement besoin d'un changement mineur dans l'implémentation, nous décrivons uniquement ce cas.

Notons λ l'unique entier tel que $2^{\lambda-1} \leq y < 2^{\lambda}$ et considérons le schéma suivant pour $y + 2^{\ell-1}y^2$:

$$\begin{array}{ccc|ccccccc|l}
 & & & 1 & \cdots & \varepsilon_\ell & \cdots & \varepsilon_1 & 1 & & = y \\
1 & \cdots & \eta_{\lambda-1} & \cdots & \cdots & 1 & & & & & = 2^{\ell-1}y^2 \\
\hline
\multicolumn{3}{c|}{y_1} & \multicolumn{7}{c|}{y_2} & \\
\end{array}$$

On coupe la somme en deux blocs, y_1 et y_2. On doit allouer au total trois `1`-bits pour y_2 et y_1. On sait que $s(y_1) \geq 1$ car le chiffre le plus significatif de $y + 2^{\ell-1}y^2$ est dans la partie de y_1. On se focalise sur le cas où $\ell = 1$, les autres cas étant similaires.

Comme expliqué précédemment, on attaque ce problème étape par étape en ajoutant successivement des bits dans la décomposition binaire de y. Dans cet algorithme, on considère que les blocs `011` et `11` sont différents, car nous avons plus d'informations sur un entier s'il contient le premier bloc plutôt que le deuxième. En fait, dans ce contexte, il est plus utile de les voir comme des mots plutôt que comme des entiers.

On dit qu'un mot binaire ω est un *candidat* si la partie droite de la somme $\omega + \omega^2$ (avec ici un abus de notation entre le mot et l'entier qui se correspondent) a au plus deux `1`-bits pour une certaine longueur du bloc. Si ω est candidat, alors on peut étendre ω en $\mathtt{0}\omega$ et $\mathtt{1}\omega$ par la gauche et vérifier si ces deux mots sont encore des candidats. Si un mot ω n'est pas un candidat, il n'est pas possible de l'étendre en un mot candidat car les chiffres d'indice faible contiennent déjà trop de `1`-bits et ne peuvent plus être modifiés par une extension par la gauche de ω. L'algorithme commence avec le mot $\omega = \mathtt{1}$, construit des candidats, les traduit en entiers et vérifie s'ils satisfont l'équation $s(n^2) = 4$. Enfin l'algorithme s'arrête lorsque plus aucun candidat ne peut être prolongé.

De plus, pour que l'algorithme s'arrête, nous avons deux conditions supplémentaires. La première est au coeur de l'algorithme et de notre démarche; un mot qui est déjà de poids k ne peut être étendu par d'autres 1-bits, et donc les candidats ont moins que k 1-bits. Sans cette condition, on ne peut pas être sur que le programme s'arrête.

La deuxième condition porte sur la longueur du possible bloc de 0-bits au début du mot $0\cdots0\omega$. Notons que cette condition est également obligatoire pour que l'algorithme s'arrête sinon il pourrait prolonger indéfiniment un candidat avec uniquement des 0-bit à gauche. On a le résultat suivant :

Lemme VI.6.2. *Soit ω un candidat de longueur λ. Alors le mot $10^\lambda\omega$ n'est pas un candidat.*

Preuve. Dans ce cas on a la somme suivante

$$\begin{array}{llllllll} & & & & & 1 & 0\cdots0\,\omega & = m \\ + & 1 & 0\cdots0 & \omega & 0\cdots0 & 0 & \omega^2 & = m^2 \\ \hline & 1 & & \omega & & 1 & \omega^2+\omega & \end{array}$$

La somme contient toujours plus que trois 1-bits. □

La gestion des candidats, il était pratique d'employer une pile. Lorsque la pile est vide, cela veut dire qu'on aura testé tous les candidats possibles. L'algorithme est donné en pseudo-code dans Algorithme 3.

Algorithme 3 : `max-integer`

```
1  Procedure max-integer(k):
2  S=[1] ;                                   /* Stack of all candidates */
3  while S is not empty do
4  |  ω=S.top() ;                            /* Top element of the stack */
5  |  n = 1 + 2ω ;
6  |  if s(n²) = 4 then
7  |  |  print(n)
8  |  S.pop() ;                              /* remove ω from the stack */
9  |  if 1ω is a candidate and s(ω) < k then
10 |  |  S.push(1ω) ;                        /* add 1ω to the top of the stack */
11 |  if 0ω is a candidate and 2*(length of leading zeros of 0ω) < length of ω
   |   then
12 |  |  S.push(0ω) ;                        /* add 0ω to the top of the stack */
```

Pour le Théorème VI.2.3, (VI.2.6), notre implémentation de l'algorithme prend 102 secondes pour terminer pour $s(n) = 16$ et environ 2h 50min pour $s(n) = 17$ avec une machine de bureau Intel(R) Core(TM) i9-9980HK CPU @ 2.40GHz. Le code du programme est disponible à l'adresse :

`https://gitlab.inria.fr/jamet/on-the-binary-digits-of-n-and-n2`.

VI.7. Les cas restants $k = 14, 15$

Dans ce paragraphe, nous considérons le problème de déterminer les solutions impaires de

$$s(n) = s(n^2) \in \{14, 15\},$$

qui constituent les deux seuls cas restants du problème initial. Ces deux cas sont bien plus difficiles que les précédents, car nous ne pouvons pas nous reposer sur les cas précédents. En effet, (VI.1.2)–(VI.1.3), il y a une infinité de solutions pour $s(n) = s(n^2) \in \{12, 13\}$.

Pour attaquer ces derniers cas restants, nous avons amélioré nos programmes qui déterminent les ensembles $\Delta_{\ell_1,\ell_2,m}$. Pour le cas $k = 14$, il y a beaucoup plus de sous-cas que précédemment et la distinction de cas devient très lourde. Nous pouvons toujours nous reposer sur le Lemme VI.3.2, qui donne une décomposition en m blocs avec $1 \leq m \leq 6$ pour des solutions suffisamment grandes.

Pour le cas $m = 1$, nous donnons les constantes données par les Lemme VI.3.7 et Lemme VI.3.8. Comme précédemment, on écrit $n = x_1\mathtt{0}\ldots\mathtt{0}x_0$ et par symétrie on peut supposer que $s(x_1) \geq s(x_0)$. Dans ce cas, on a $s(x_1)^2, s(x_0^2) \leq 10$ et le tableau suivante (voir aussi Paragraphe VI.4.1) :

$s(x_1)$	$s(x_0)$	Ensembles Δ pour $s(x_1x_0) = 2$	Ensembles Δ pour $s(x_1x_0) = 3$
7	7	$\Delta_{7,10,93} \times \Delta_{7,10,93}$	$\Delta_{7,9,182} \times \Delta_{7,9,182}$
8	6	$\Delta_{8,10,91} \times \Delta_{6,10,91}$	$\Delta_{8,9,178} \times \Delta_{6,9,178}$
9	5	$\Delta_{9,10,85} \times \Delta_{5,10,85}$	$\Delta_{9,9,166} \times \Delta_{5,9,166}$
10	4	$\Delta_{10,10,75} \times \Delta_{4,10,75}$	$\Delta_{10,9,146} \times \Delta_{4,9,146}$
11	3	$\Delta_{11,10,61} \times \Delta_{3,10,61}$	$\Delta_{11,9,118} \times \Delta_{3,9,118}$
12	2	$\Delta_{12,10,43} \times \Delta_{2,10,43}$	$\Delta_{12,9,82} \times \Delta_{2,9,82}$

Tableau VI.3. Ensembles Δ pour $m = 1$ et $k = 14$.

L'algorithme `next` ne donne aucune solution possible pour $s(x_1x_0) = 2$ et seulement le couple $(3695, 143)$ pour $s(x_1x_0) = 3$. Comme $s(3695^2) + s(143^2) = 17 > 11$, ce couple n'est pas solution du problème. Ainsi il n'existe aucune famille telle que $m = 1$ comme pour (VI.1.2)–(VI.1.4).

Pour $k = 15$, nous devons déterminer les ensembles suivants :

$s(x_1)$	$s(x_0)$	Ensembles Δ pour $s(x_1x_0) = 2$	Ensembles Δ pour $s(x_1x_0) = 3$
8	7	$\Delta_{8,11,107} \times \Delta_{7,11,107}$	$\Delta_{8,10,210} \times \Delta_{7,10,210}$
9	6	$\Delta_{9,11,103} \times \Delta_{6,11,103}$	$\Delta_{9,10,202} \times \Delta_{6,10,202}$
10	5	$\Delta_{10,11,95} \times \Delta_{5,11,95}$	$\Delta_{10,10,186} \times \Delta_{5,10,186}$
11	4	$\Delta_{11,11,83} \times \Delta_{4,11,83}$	$\Delta_{11,10,163} \times \Delta_{4,10,163}$
12	3	$\Delta_{12,11,67} \times \Delta_{3,11,67}$	$\Delta_{12,10,130} \times \Delta_{3,10,130}$
13	2	$\Delta_{13,11,47} \times \Delta_{2,11,47}$	$\Delta_{13,10,90} \times \Delta_{2,10,90}$

Tableau VI.4. Ensembles Δ pour $m = 1$ et $k = 15$.

Enfin, nous avons réalisé une recherche globale avec $s(n^2) = s(n) = 11, \ldots, 15$ pour $n < 2^{80}$ avec un triple découpage. On trouve les proportions suivantes :

k	Proportion d'entiers impairs tel que $s(n) = s(n^2) = k$ pour $n < 2^{80}$	Temps de calcul	Nombre de coeurs utilisés
11	$4 \cdot 10^{-10}$	2min 19sec	659
12	$1.5 \cdot 10^{-10}$	4min 58sec	480
13	$7.2 \cdot 10^{-11}$	32min 25sec	360
14	$2.6 \cdot 10^{-11}$	3h 34min 43sec	277
15	$1.2 \cdot 10^{-11}$	23h 24min 47sec	218

TABLEAU VI.5. Proportion de solutions et temps de calcul pour $s(n^2) = s(n) = 11, \ldots, 15$.

Ces cinq proportions sont similaires, mais il y a une différence pour le nombre de solutions du problème $s(n^2) = s(n) = k$ entre les cas $k = 11$ et $k = 12, 13$. Cette différence n'est pas visible en étudiant seulement les proportions de solutions. Nous donnons plus de détails sur ces cas.

Pour $k = 11$ notre algorithme trouve que la plus grande solution est $n = 35463511416833$ de longueur binaire 46, sans pour autant affirmer qu'il n'y a pas d'autres solutions. Comme expliqué auparavant, cela semble peu probable de trouver une solution au delà de 2^{80} si la plus grande solution est de longueur 46.

La structure des solutions pour $k = 12$ et $k = 13$ est clairement différente car on peut voir un seuil entre les solutions sporadiques et les solutions appartenant à des familles infinies composées de petits blocs. Ces familles infinies apparaissent également avant ce seuil. Par exemple pour $k = 12$, on a que toutes les solutions de longueur binaire plus grande que 55 est la forme $111 \cdot 2^t + 111$, mais cette forme est déjà valide pour $t \geq 15$.

Pour $k = 14$, nous trouvons des solutions de longueur binaire 80, comme par exemple l'entier

$$n = 605643510452789079965697.$$

Néanmoins, aucune famille infinie n'apparaît clairement lorsque l'on regarde toutes les solutions avec attention. Pour $k = 15$, le constat est similaire; on a une solution de longueur 80, comme par exemple l'entier

$$n = 605642350760526229274625,$$

mais encore une fois aucune famille infinie n'apparaît clairement. Également, les 1-bits dans ces solutions ne suivent pas une règle apparente. Nous pensons alors que si une famille infinie existait pour $k = 14$ ou $k = 15$, elle devrait apparaître déjà pour $n < 2^{80}$, comme pour les cas $k = 12, 13$ et 16. En suivant ces arguments, il y aurait qu'un nombre fini de solutions et nous avons alors formulé la Conjecture VI.2.1.

Cette dichotomie entre le nombre fini et le nombre infini de solutions pour le problème (VI.1.1) est assez surprenante dans un premier temps. Nous pouvons penser que pour les cas $k = 14$ et $k = 15$, une infinité de solutions serait moins surprenant et que la valeur $k = 12$ représenterait un seuil. D'autre part, ce genre de phénomène apparait fréquemment, avec par exemple la différence entre E_3 (voir le Théorème VI.2.2) et l'ensemble conjecturé de solutions E_4 (voir la Conjecture VI.2.2).

Si nous modifions légèrement les données du système, nous trouvons des profils de solutions complètement différents. Par exemple, pour le système

$$s(n) = 14, \quad s(n^2) = 15,$$

nous avons deux familles infinies "évidentes",

$$n = 23 \cdot 2^t + 2943, \text{ avec } t \geq 13,$$

et

$$n = 727 \cdot 2^t + 727, \text{ avec } t \geq 21.$$

Pour réaliser tous les calculs dans ce chapitre, nous avons utilisé le cluster *gros* qui comprends 123 nœuds, Intel Xeon Gold 5220, 18 cœurs / CPU, et 96 GiB de mémoire, voir `https://www.grid5000.fr/w/Nancy:Hardware#gros`. Le code du programme est aussi disponible à l'adresse :

`https://gitlab.inria.fr/jamet/on-the-binary-digits-of-n-and-n2`.

CHAPITRE VII

Corrélations de la suite de Rudin–Shapiro

Ce chapitre porte sur un *travail en cours* sur la corrélation d'ordre k de la suite de Rudin–Shapiro, voir la Définition I.2.9, page 51. Le résultat principal de ce chapitre porte plus précisément sur la suite de Rudin–Shapiro le long des nombres premiers. Le résultat présenté dans cette thèse est partiel et couvre les cas $k = 1, 3$ et k pair. Néanmoins, nous pensons que la méthode employée est adaptée pour résoudre les cas restants.

Notations

Dans tout ce chapitre, nous utilisons les notations suivantes :

- Soient $a, p \geq 0$ et $\alpha \geq 1$ des entiers. On note $p \mid a$ si p divise a et $p^\alpha \parallel a$ si p^α divise exactement a, c'est-à-dire $p^\alpha \mid a$ et $p^{\alpha+1} \nmid a$.
- On note (a, b) le plus grand commun diviseur de deux entiers a et b.
- Pour des entiers $n \geq 1$ et $q \geq 2$, on note $v_q(n) = \max\{\nu \in \mathbb{N} \ : q^\nu \mid n\}$ est la valuation q-adique de n.
- Pour un réel x, on note $\|x\| = \min_{h\in\mathbb{Z}} |x - h|$ la distance de x à $\mathbb{Z}$.
- Pour un entier $M \geq 1$, on note $[M] = \{0, 1, \ldots, M\}$.
- On note φ la fonction indicatrice d'Euler et Λ la fonction de von Mangoldt définie par
$$\Lambda(n) = \begin{cases} \log p, \text{ si } n = p^\nu \text{ pour un certain } \nu \geq 1, \\ 0, \text{ sinon.} \end{cases}$$
- On note $\pi(x)$ le nombre de nombres premiers p tels que $p \leq x$ et $\pi(x; q, a)$ le nombre de nombres premiers p tels que $p \leq x$ et $p \equiv a \pmod q$, pour des entiers $a \geq 0$, $q \geq 1$.
- Soient $k \geq 0$ un entier et $\alpha = (\alpha_0, \ldots, \alpha_k) \in \mathbb{R}^{k+1}$, on note
$$\tilde{\alpha} = \sum_{j=1}^{k} j\alpha_j, \qquad \tilde{\alpha_i} = \sum_{j=i}^{k} \alpha_j, \qquad \text{pour } 0 \leq i \leq k.$$
- Soient $k \geq 0$ un entier, $\alpha = (\alpha_0, \ldots, \alpha_k)$, et $\beta = (\beta_0, \ldots, \beta_k) \in \mathbb{R}^{k+1}$. On note $\alpha \cdot \beta$ le produit scalaire usuel de α et β :
$$\alpha \cdot \beta = \sum_{0\leq i\leq k} \alpha_i\beta_i.$$

VII.1. Introduction

Dans tout ce qui suit, la lettre p désigne un nombre premier, $q \geq 2$ est un entier fixé et s_q est la fonction somme des chiffres en base q.

Aloui, Mauduit et Mkaouar [AMM21] montrent le résultat suivant sur la corrélation de la suite de Thue–Morse $(t'_n)_{n\geq 0}$ sur l'alphabet $\{-1,1\}$ le long des nombres premiers.

Théorème VII.1.1 ([AMM21]). *Soit $k \geq 1$ un entier, on a*

$$\lim_{N\to+\infty} \frac{1}{\pi(N)} \sum_{p\leq N} t'_p t'_{p+1} \cdots t'_{p+k} = \lim_{N\to+\infty} \frac{1}{N} \sum_{\substack{n\leq N \\ n\equiv 1 \pmod 2}} t'_n t'_{n+1} \cdots t'_{n+k}.$$

Nous voulons étendre ce théorème à la suite de Rudin–Shapiro le long des nombres premiers. Nous appelons la suite $\mathcal{R}' = ((-1)^{r_n})_{n\geq 0}$ la *suite de Rudin–Shapiro sur l'alphabet* $\{-1,1\}$, où $\mathcal{R} = (r_n)_{n\geq 0}$ est la suite de Rudin–Shapiro sur l'alphabet $\{0,1\}$. D'après le Chapitre II, pour tout $n \geq 0$, on a $r'_n = (-1)^{r_{11}(n)}$, où $r_{11}(n)$ est le nombre d'occurrences de `11` dans $(n)_2$. De manière équivalente, on note $r_{01}(n)$ le nombre d'occurrences de `01` dans $(n)_2$, y compris le bloc du chiffre le plus significatif.

Définition VII.1.1. Soient des entiers $k, N \geq 1$. On définit la somme suivante :

$$\mathcal{S}_k(N) = \sum_{p\leq N} r'_p r'_{p+1} \cdots r'_{p+k}.$$

L'objectif de ce chapitre est d'étudier la somme $\mathcal{S}_k(N)$. Nous allons comparer le comportement de cette somme avec la somme suivante :

$$\mathcal{S}'_k(N) = \sum_{n\leq N} r'_n r'_{n+1} \cdots r'_{n+k}.$$

Les sommes $\mathcal{S}'_k(N)$ ont été étudiées par Mauduit et Sárközy [MS98]. Ils montrent une nette différence entre le comportement de $S'_1(N)$ et $S'_3(N)$:

$$\left|\mathcal{S}'_1(2^M)\right| = \left|\sum_{n\leq 2^M} r'_n r'_{n+1}\right| = o(2^M), \qquad M \to +\infty$$

$$\left|\mathcal{S}'_3(2^M)\right| = \left|\sum_{n\leq 2^M} r'_n r'_{n+1} r'_{n+2} r'_{n+3}\right| = 2^{M-1},$$

Cependant, leur méthode de démonstration n'est pas adaptable avec les sommes impliquant exclusivement des nombres premiers et c'est pour cette raison que nous faisons appel à des sommes d'exponentielles. Rappelons que nous notons $e(x) = \exp(2i\pi x)$ pour x un réel.

Soient $k \in \mathbb{N}$ et $\alpha = (\alpha_0, \ldots, \alpha_k) \in \mathbb{R}^{k+1}$. Nous définissons les sommes suivantes :

$$\mathcal{A}_\alpha(N) = \sum_{p\leq N} e\left(\alpha_0 r_{11}(p) + \cdots + \alpha_k r_{11}(p+k)\right).$$

Par conséquent, si $\alpha = (1/2, \ldots, 1/2)$, alors

$$\mathcal{S}_k(N) = \mathcal{A}_\alpha(N),$$

car $(-1)^n = e\left(\frac{n}{2}\right)$.

VII.2. Résultats

Théorème VII.2.1. *Soit $k \in \{1,3\} \cup 2\mathbb{N}$, on a*

$$\lim_{N\to+\infty} \frac{1}{\pi(N)} \sum_{p\leq N} r'_p r'_{p+1} \cdots r'_{p+k} = 0.$$

Nous déduisons le Théorème VII.2.1 des deux théorèmes suivants. Le premier théorème donne une majoration de la somme $\mathcal{A}_\alpha(N)$ pour tout $\alpha \in \mathbb{R}^{k+1}$.

Théorème VII.2.2. *Soient $k \geq 1$ un entier et $\alpha = (\alpha_0, \ldots, \alpha_k) \in \mathbb{R}^{k+1}$. Alors pour tout entier N suffisamment grand, on a*

$$\mathcal{A}_\alpha(N) \ll (\log N)^K N^{1-\sigma},$$

où $\sigma = \min(1/2, c\,\|\tilde{\alpha_0}\|^2)$, K ne dépend que de k et la constante c est indépendante de α.

Il est clair que si $\tilde{\alpha_0} \in \mathbb{Z}$, la majoration du théorème précédent est triviale. Le deuxième théorème couvre spécifiquement ce cas. La démonstration de ce théorème est la partie difficile de ce chapitre et n'est démontrée que pour $k \in \{1,3\}$.

Théorème VII.2.3. *Soient $k \in \{1,3\}$ et $\alpha = (\alpha_0, \ldots, \alpha_k) \in \mathbb{R}^{k+1}$ tels que $\tilde{\alpha_0} = h \in \mathbb{Z}$. Il existe $Q(\alpha, k)$ tel que pour tout réel $A > 1$ et pour tout entier N suffisamment grand, on a*

$$\mathcal{A}_\alpha(N) = Q(\alpha,k)\pi(N) + O\left(\frac{N}{\log^A N}\right).$$

Il semble qu'il n'existe pas de formule simple permettant d'exprimer $Q(\alpha, k)$ dans le cas général. Pour terminer ce travail, il reste à trouver une forme générale pour tout k impair au Théorème VII.2.3 afin d'en déduire les cas non couverts par le Théorème VII.2.1, à savoir $k \in 2\mathbb{N}+1 \setminus \{1,3\}$.

Mauduit et Sárközy [MS98] montrent que la corrélation d'ordre 4 de la suite de Rudin–Shapiro est grande : pour tout entier $N \geq 8$,

$$C_4(\mathcal{R}', N) \geq \frac{1}{8}N.$$

Notre quatrième théorème porte sur la corrélation d'ordre d de la suite de Rudin–Shapiro, pour $d \equiv 0 \pmod 4$.

Théorème VII.2.4. *Soit $k \equiv 0 \pmod 4$. Pour tout $N \geq 8k+15$, on a*

$$C_k(\mathcal{R}', N) \geq \frac{1}{8}N.$$

Par conséquent, le théorème précédent fournit un argument supplémentaire pour affirmer que la suite de Rudin–Shapiro ne satisfait pas des bonnes propriétés aléatoires, voir le Paragraphe I.2.7. Par définition de la corrélation d'ordre k, voir la Définition I.2.9, le lemme suivant implique le Théorème VII.2.4.

Lemme VII.2.1. *Soient $k \geq 0$, $D_1 = (0, 1, \ldots, 8k+3)$ et $D_2 = (0, 1, \ldots, 8k+7)$.*

- *Pour $N \geq 8k+8$, soit M tel que $2^{M+1} \leq N < 2^{M+2}$. Alors on a*
$$\frac{1}{N} \sum_{n<2^M} r'_n r'_{n+1} \cdots r'_{n+8k+3} \leq -\frac{1}{8}.$$
- *Pour $N \geq 8d+15$, soit M tel que $2^{M+1} \leq N < 2^{M+2}$. Alors on a*
$$\frac{1}{N} \sum_{n<2^M} r'_n r'_{n+1} \cdots r'_{n+8k+7} \geq \frac{1}{8}.$$

Le Théorème VII.2.1 et le Lemme VII.2.1 montrent le changement de comportement de la corrélation d'ordre $k \equiv 0 \pmod 4$ lorsque la suite de Rudin–Shapiro est raréfiée le long des nombres premiers.

VII.3. Préliminaires

Le lemme suivant est l'ingrédient clé de la preuve des Théorèmes VII.2.2 et VII.2.3. En effet, il donne une relation de récurrence entre $r_{11}(n+1)$ et $r_{11}(n)$.

Lemme VII.3.1. *Soit $n \geq 0$ un entier. On a*
$$r_{11}(n+1) = r_{11}(n) + 1 - v_2(n+1) + r_{01}(n) - r_{01}(n+1).$$

Preuve. Soit $n = \sum_{i=0}^{\ell} \varepsilon_i 2^i$ la décomposition de n en base 2. D'après la formule de Legendre, on a
$$\begin{aligned} n - r_{11}(n) &= \sum_{i=0}^{\ell} \left(\varepsilon_i 2^i - \varepsilon_{i+1}\varepsilon_i\right) = \sum_{i=0}^{\ell} \varepsilon_i \left(2^i - \varepsilon_{i+1}\right) \\ &= \sum_{i=0}^{\ell} \varepsilon_i \left(2^i - 1\right) + \sum_{\substack{i=0 \\ \varepsilon_{i+1}=0}}^{\ell} \varepsilon_i = v_2(n!) + r_{01}(n). \end{aligned}$$

On en déduit
$$\begin{aligned} n &= r_{11}(n) + v_2(n!) + r_{01}(n), \\ n+1 &= r_{11}(n+1) + v_2((n+1)!) + r_{01}(n+1). \end{aligned}$$

L'identité $v_2((n+1)!) = v_2(n+1) + v_2(n!)$ permet de conclure. □

Remarque VII.3.1. L'identité du lemme précédent est similiaire à l'identité pour s_q donnée par
$$s_q(n+1) = s_q(n) + 1 - (q-1)v_q(n+1).$$
Notons que nous pouvons également démontrer le Lemme VII.3.1 en utilisant l'identité $s_2(n) = r_{11}(n) + r_{01}(n)$.

Soient $n, i \geq 0$ des entiers. En itérant le lemme précédent, on a
$$r_{11}(n+i) = r_{11}(n) + i - \sum_{j=1}^{i} v_2(n+j) + \Delta(n,i), \tag{VII.3.1}$$

où $\Delta(n,i) = r_{\mathtt{01}}(n) - r_{\mathtt{01}}(n+i)$. Au passage, notons que $\Delta(n,i) = \Delta(n+i-1,1) + \Delta(n,i-1)$ pour $i \geq 2$.

Le lemme suivant décrit toutes les valeurs possibles de $\Delta(n,1)$.

Lemme VII.3.2. *Pour tout $n \geq 1$, on a $\Delta(n,1) \in \{-1,0,1\}$. Soient $u = v_2(n+1)$ et $\varepsilon \in \{0,1\}$ tel que $n+1 \equiv 2^u + \varepsilon 2^{u+1} \pmod{2^{u+2}}$. On a :*

- *Si $u = 0$ et $\varepsilon = 0$, alors $\Delta(n,1) = -1$.*
- *Si $u = 0$ et $\varepsilon = 1$, alors $\Delta(n,1) = 0$.*
- *Si $u > 0$ et $\varepsilon = 0$, alors $\Delta(n,1) = 0$.*
- *Si $u > 0$ et $\varepsilon = 1$, alors $\Delta(n,1) = 1$.*

Preuve. On distingue les cas suivants :

- Si $v_2(n+1) = 0$, on distingue les cas suivants :
  - Si $n+1 \equiv 1 \pmod 4$, alors on a $(n)_2 = \star\mathtt{00}$ et $(n+1) = \star\mathtt{01}$ avec $\star$ un mot binaire. Donc on a $\Delta(n,1) = -1$.
  - Si $n+1 \equiv 3 \pmod 4$, alors on a $(n)_2 = \star\mathtt{10}$ et $(n+1) = \star\mathtt{11}$ avec $\star$ un mot binaire. Donc on a $\Delta(n,1) = 0$.
- Si $v_2(n+1) = u > 0$, on distingue les cas suivants :
  - Si $n+1 \equiv 2^u \pmod{2^{u+2}}$, alors on a $(n)_2 = \star\mathtt{001}^u$ et $(n+1)_2 = \star\mathtt{010}^u$, avec $\star$ un mot binaire. Donc on a $\Delta(n,1) = 0$.
  - Si $n+1 \equiv 2^u + 2^{u+1} \pmod{2^{u+2}}$, alors on a $(n)_2 = \star\mathtt{101}^u$ et $(n+1)_2 = \star\mathtt{110}^u$, avec $\star$ un mot binaire. Donc on a $\Delta(n,1) = 1$.

□

Remarque VII.3.2. La parité de $n+1$ joue un rôle important dans la valeur de $\Delta(n,1)$. C'est pourquoi, nous constatons une sorte de biais dans le cadre de notre étude le long des nombres premiers car un nombre premier supérieur ou égal à 3 est impair.

Comme une conséquence directe du précédent lemme, nous déduisons que $\Delta(n,i) \in \{-i,\dots,i\}$ et ne dépend que des valuations 2-adiques $u_j = v_2(n+j)$ pour $1 \leq j \leq i$ et du vecteur $\varepsilon = (\varepsilon_1,\dots,\varepsilon_i)$ tel que $n + j \equiv 2^{u_j} + \varepsilon_j 2^{u_j+1} \pmod{2^{u_j+2}}$.

Le lemme suivant est central dans la théorie des sommes d'exponentielles car il permet de détecter des congruences à l'aide d'outils analytiques. Pour plus de détails sur les sommes d'exponentielles, voir [IK04, Chapter 8].

Lemme VII.3.3. *Soit $a \in \mathbb{Z}$. On a*

$$\sum_{x=0}^{m-1} e\left(\frac{ax}{m}\right) = \begin{cases} m, & \text{si } a \equiv 0 \pmod m, \\ 0, & \text{sinon.} \end{cases}$$

Le lemme suivant se déduit du théorème de Siegel–Walfisz, voir [IK04, Corollary 5.29] par exemple.

Lemme VII.3.4. *Soient $q \geq 1$ et $a \geq 0$ avec $(a,q) = 1$. Soit $A > 1$ un réel. Pour $N \geq 2$, on a*

$$\pi(N;q,a) = \frac{\pi(N)}{\varphi(q)} + O\left(\frac{N}{\log^A N}\right), \qquad N \to +\infty.$$

Notons que pour démontrer le Théorème VII.2.1, le théorème des nombres premiers pour les progressions arithmétiques suffit, voir [IK04, Theorem 2.2].

VII.4. Preuve du Théorème VII.2.2

Nous introduisons dans un premier temps une série de lemmes.

Lemme VII.4.1. *Soient $\alpha, \beta \in \mathbb{R}$ et $(y_n)_{n\geq 0}$ une suite de nombres complexes. Pour tout entier $i \geq 1$ fixé et N suffisamment grand, on a*

$$\sum_{n\leq N} y_n e(\alpha v_2(n+i) + \beta\Delta(n,i)) \ll_i (\log N)^i \max_{r\in[0,1)} \left|\sum_{n\leq N} y_n e(rn)\right|.$$

Preuve. D'après le Lemme VII.3.2, la valeur de $\Delta(n,i)$ est déterminée par un système de congruences. Soit $M = \lfloor \log_2(N+i)\rfloor$. On exprime la somme

$$S = \sum_{n\leq N} y_n e(\alpha v_2(n+i) + \beta\Delta(n,i))$$

de la manière suivante :

$$S = \sum_{(u_1,\dots,u_i)\in[M]^i} \sum_{\substack{n\leq N \\ v_2(n+j)=u_j \\ 1\leq j\leq i}} y_n e(\alpha u_i + \beta\Delta(n,i)).$$

Or $v_2(n+1) = u_1$ si et seulement si $n+1 \equiv 0 \pmod{2^{u_1}}$ et $n+1 \not\equiv 0 \pmod{2^{u_1+1}}$. Donc on a

$$S = S_0 - S_1$$

où

$$S_0 = \sum_{(u_1,\dots,u_i)\in[M]^i} \sum_{\substack{n\leq N \\ v_2(n+j)=u_j \\ 2\leq j\leq i}} \sum_{n+1\equiv 0 \pmod{2^{u_1}}} y_n e(\alpha u_i + \beta\Delta(n,i)),$$

$$S_1 = \sum_{(u_1,\dots,u_i)\in[M]^i} \sum_{\substack{n\leq N \\ v_2(n+j)=u_j \\ 2\leq j\leq i}} \sum_{n+1\equiv 0 \pmod{2^{u_1+1}}} y_n e(\alpha u_i + \beta\Delta(n,i)).$$

On effectue le même découpage sur S_0 et S_1 en fonction de $v_2(n+2) = u_2$. En appliquant ce découpage pour chaque $v_2(n+j)$, on obtient

$$S = T_0 + T_1 + \cdots + T_{2^i-1},$$

où T_λ, $0 \leq \lambda < 2^i$, est de la forme

$$T_\lambda = (-1)^{\tilde{\varepsilon_1}} \sum_{(u_1,\dots,u_i)\in[M]^i} \sum_{\substack{n\leq N \\ n+j\equiv 0 \pmod{2^{u_j+\varepsilon_j}} \\ 1\leq j\leq i}} y_n e(\alpha u_i + \beta\Delta(n,i))$$

avec $\lambda = \sum_{1\le j\le i} \varepsilon_j 2^{j-1}$, pour un certain $\varepsilon = (\varepsilon_1, \dots, \varepsilon_i) \in \{0,1\}^i$ et $\tilde{\varepsilon_1} = \varepsilon_1 + \cdots + \varepsilon_i$. D'après le Lemme VII.3.2, la valeur de $\Delta(n,i)$ ne dépend que de ε dans la somme intérieure de T_λ, noté Δ_ε. Par conséquent, on a

$$T_\lambda = (-1)^{\tilde{\varepsilon_1}} \sum_{(u_1,\dots,u_i)\in[M]^i} e(\alpha u_i + \beta\Delta_\varepsilon) \sum_{\substack{n\le N \\ n+j\equiv 0 \pmod{2^{u_j+\varepsilon_j}} \\ 1\le j\le i}} y_n.$$

D'après le Lemme VII.3.3, nous pouvons traduire le système de congruence

$$\begin{cases} n+1 \equiv 0 \pmod{2^{u_1+\varepsilon_1}}, \\ \vdots \\ n+i \equiv 0 \pmod{2^{u_i+\varepsilon_i}}, \end{cases}$$

par une somme d'exponentielle. Donc on a

$$\begin{aligned} T_\lambda &= (-1)^{\tilde{\varepsilon_1}} \sum_{(u_1,\dots,u_i)\in[M]^i} e\left(\alpha u_i + \beta\Delta_\varepsilon\right) \sum_{n\le N} y_n \left(\frac{1}{2^{u_1+\varepsilon_1}} \sum_{0\le \ell_1 < 2^{u_1+\varepsilon_1}} e\left(\frac{\ell_1(n+1)}{2^{u_1+\varepsilon_1}}\right)\right) \\ &\qquad \times \cdots \times \left(\frac{1}{2^{u_i+\varepsilon_i}} \sum_{0\le \ell_i < 2^{u_i+\varepsilon_i}} e\left(\frac{\ell_i(n+i)}{2^{u_i+\varepsilon_i}}\right)\right) \\ &= (-1)^{\tilde{\varepsilon_1}} \sum_{(u_1,\dots,u_i)\in[M]^i} e\left(\alpha u_i + \beta\Delta_\varepsilon\right) \frac{1}{2^{u_1+\cdots+u_i+\tilde{\varepsilon_1}}} \sum_{\substack{0\le \ell_j < 2^{u_j+\varepsilon_j} \\ 1\le j\le i}} e\left(\frac{\ell_1}{2^{u_1+\varepsilon_1}} + \cdots + \frac{i\ell_i}{2^{u_i+\varepsilon_i}}\right) \\ &\qquad \times \sum_{n\le N} y_n e\left(\left(\frac{\ell_1}{2^{u_1+\varepsilon_1}} + \cdots + \frac{\ell_i}{2^{u_i+\varepsilon_i}}\right) n\right). \end{aligned}$$

Donc pour $0 \le \lambda < 2^i$, on a

$$T_\lambda \ll M^i \max_{r\in[0,i)} \left| \sum_{n\le N} y_n e(rn) \right| = M^i \max_{r\in[0,1)} \left| \sum_{n\le N} y_n e(rn) \right|,$$

par périodicité de la fonction $e(x)$. Le lemme est donc bien démontré. □

Remarque VII.4.1. La différence principale entre la preuve du lemme précédent et celle du [AMM21, Lemma 3.1] est qu'il faut gérer plus de systèmes de congruences, ce qui implique une puissance de $\log N$ dans la majoration finale.

Lemme VII.4.2. *Soient $(\alpha_1, \dots, \alpha_k), (\beta_1, \dots, \beta_k) \in \mathbb{R}^k$ et $(y_n)_{n\ge 0}$ une suite de nombres complexes. Pour tout $k \ge 1$ fixé et pour tout N suffisamment grand, on a*

$$\begin{aligned} &\sum_{n\le N} y_n e\left(\sum_{1\le i\le k} \alpha_i v_2(n+i) + \sum_{1\le i\le k} \beta_i \Delta(n,i) \right) \\ &\qquad \ll_k (\log N)^K \max_{r\in[0,1)} \left| \sum_{n\le N} y_n e(rn) \right|. \end{aligned}$$

avec $K = k(k+1)/2$.

Preuve. Il suffit d'appliquer k fois le Lemme VII.4.1. □

La fonction r_{11} est une fonction bloc-digitale car elle dépend de blocs de longueur 2. C'est pourquoi les estimations de Martin, Mauduit et Rivat [MMR15] dans le cadre des fonctions digitales employées dans [AMM21] ne peuvent plus être utilisées dans notre contexte. Nous utilisons à la place le théorème suivant de Mauduit et Rivat [MR15].

Lemme VII.4.3 ([MR15, Theorem 4]**).** *Pour tout entier $N \geq 2$ et pour tout $(\alpha, \beta) \in \mathbb{R}^2$, on a*

$$\sum_{n \leq N} \Lambda(n) e(\alpha r_{11}(n) + \beta n) \ll \log(N)^{11/4} N^{1-c\|\alpha\|^2}.$$

Nous pouvons désormais démontrer le Théorème VII.2.2.

Preuve du Théorème VII.2.2. On pose

$$\Psi(N) = \sum_{n \leq N} \Lambda(n) e\left(\alpha_0 r_{11}(n) + \cdots + \alpha_k r_{11}(n+k)\right).$$

D'après l'identité (VII.3.1), on a

$$\sum_{i=0}^{k} \alpha_i r_{11}(n+i) = \tilde{\alpha} + \tilde{\alpha_0} r_{11}(n) - \sum_{i=1}^{k} \tilde{\alpha_i} v_2(n+i) + \sum_{i=1}^{k} \alpha_i \Delta(n,i). \tag{VII.4.1}$$

On déduit de (VII.4.1) et du Lemme VII.4.2 appliqué à $y_n = \Lambda(n) e(\tilde{\alpha_0} r_{11}(n))$ que

$$\begin{aligned}
\Psi(N) &= e(\tilde{\alpha}) \sum_{n \leq N} \Lambda(n) e\left(\tilde{\alpha_0} r_{11}(n) - \sum_{i=1}^{k} \tilde{\alpha_i} v_2(n+i) + \sum_{i=1}^{k} \alpha_i \Delta(n,i)\right), \\
&\ll_k (\log N)^K \max_{r \in [0,1)} \left| \sum_{n \leq N} \Lambda(n) e(\tilde{\alpha_0} r_{11}(n) + rn) \right|, \\
&\ll_k (\log N)^{K+11/4} N^{1-c\|\tilde{\alpha_0}\|^2}.
\end{aligned}$$

Par sommation par parties, on a

$$\mathcal{S}_\alpha(N) \ll \frac{1}{\log N} \max_{m \leq N} |\Psi(m)| + \sqrt{N},$$

et le théorème est prouvé. □

VII.5. Preuve du Théorème VII.2.3

Dans un premier temps, on suppose que $k = 1$. Soit $\alpha = (\alpha_0, \alpha_1) \in \mathbb{R}^2$ tel que $\alpha_0 + \alpha_1 = h \in \mathbb{Z}$. On a

$$\begin{aligned}
\mathcal{S}_\alpha(N) &= \sum_{p \leq N} e(\alpha_0 r_{11}(p) + \alpha_1 r_{11}(p+1)), \\
&= \sum_{p \leq N} e(\alpha_0 r_{11}(p) + \alpha_1 (r_{11}(p) + 1 - v_2(p+1) + \Delta(p,1))),
\end{aligned}$$

d'après (VII.3.1). Donc on a

$$\begin{aligned}\mathcal{S}_\alpha(N) &= e(\alpha_1)\sum_{p\le N} e((\alpha_0+\alpha_1)r_{11}(p)-\alpha_1 v_2(p+1)+\alpha_1\Delta(p,1)))\\ &= e(\alpha_1)\sum_{p\le N} e(-\alpha_1 v_2(p+1)+\alpha_1\Delta(p,1)),\end{aligned}$$

car $e((\alpha_0+\alpha_1)r_{11}(p))=1$. Or $\Delta(p,1)\in\{0,1\}$ d'après le Lemme VII.3.2, on a

$$\mathcal{S}_\alpha(N) = e(\alpha_1)(S_0+S_1),$$

où

$$\begin{aligned}S_0 &= \sum_{\substack{p\le N\\ \Delta(p,1)=0}} e(-\alpha_1 v_2(p+1)),\\ S_1 &= \sum_{\substack{p\le N\\ \Delta(p,1)=1}} e(-\alpha_1 v_2(p+1)+\alpha_1).\end{aligned}$$

Soit $M=\lfloor \log_2(N+1)\rfloor$, on a

$$\begin{aligned}S_0 &= \sum_{1\le u\le M} e(-\alpha_1 u) \sum_{\substack{p\le N\\ v_2(p+1)=u\\ \Delta(p,1)=0}} 1,\\ S_1 &= e(\alpha_1)\sum_{1\le u\le M} e(-\alpha_1 u) \sum_{\substack{p\le N\\ v_2(p+1)=u\\ \Delta(p,1)=1}} 1.\end{aligned}$$

D'après le Lemme VII.3.2 et le fait que $u=v_2(p+1)\ge 1$, nous pouvons en déduire que $\Delta(p,1)=\varepsilon\in\{0,1\}$ si et seulement si $p\equiv 2^u+\varepsilon 2^{u+1}-1 \pmod{2^{u+2}}$. Soient $\gamma_1=2^u-1$ et $\gamma_2=2^u+2^{u+1}-1$. On a

$$\begin{aligned}S_0 &= \sum_{1\le u\le M} e(-\alpha_1 u) \sum_{\substack{p\le N\\ p\equiv 2^u-1 \pmod{2^{u+2}}}} 1,\\ &= \sum_{1\le u\le M} e(-\alpha_1 u)\pi(N;2^{u+2},\gamma_1),\end{aligned}$$

et

$$\begin{aligned}S_1 &= e(\alpha_1)\sum_{1\le u\le M} e(-\alpha_1 u) \sum_{\substack{p\le N\\ p\equiv 2^u+2^{u+1}-1 \pmod{2^{u+2}}}} 1,\\ &= e(\alpha_1)\sum_{1\le u\le M} e(-\alpha_1 u)\pi(N;2^{u+2},\gamma_2),\end{aligned}$$

Or $\varphi(2^{u+2}) = 2^{u+1}$, donc d'après le Lemme VII.3.4, pour $A > 1$, on a

$$S_0 = \pi(N) \sum_{1\le u\le M} \frac{e(-\alpha_1 u)}{2^{u+1}} + O\left(\frac{N}{\log^A N}\right),$$
$$S_1 = e(\alpha_1)\pi(N) \sum_{1\le u\le M} \frac{e(-\alpha_1 u)}{2^{u+1}} + O\left(\frac{N}{\log^A N}\right).$$

Par conséquent on a

$$\begin{aligned}\mathcal{S}_\alpha(N) &= e(\alpha_1)(1+e(\alpha_1))\pi(N) \sum_{1\le u\le M} \frac{e(-\alpha_1 u)}{2^{u+1}} + O\left(\frac{N}{\log^A N}\right),\\ &= \frac{e(\alpha_1)}{2}(1+e(\alpha_1))\pi(N) \sum_{1\le u\le M} \left(\frac{e(-\alpha_1)}{2}\right)^u + O\left(\frac{N}{\log^A N}\right),\\ &= e(\alpha_1)\frac{1+e(\alpha_1)}{2(2e(\alpha_1)-1)}\pi(N) + O\left(\frac{N}{\log^A N}\right).\end{aligned}$$

On pose

$$Q(\alpha,1) = e(\alpha_1)\frac{1+e(\alpha_1)}{2(2e(\alpha_1)-1)}, \tag{VII.5.1}$$

et le théorème est prouvé pour $k = 1$.

Supposons maintenant que $k = 3$. Soit $\alpha = (\alpha_0, \dots, \alpha_3) \in \mathbb{R}^4$ tel que $\alpha_0 + \cdots + \alpha_3 = h \in \mathbb{Z}$ et

$$\mathcal{S}_\alpha(N) = \sum_{p\le N} e(\alpha_0 r_{11}(p) + \cdots + \alpha_3 r_{11}(p+3)).$$

D'après l'identité (VII.3.1), on a

$$\begin{aligned}\alpha_0 r_{11}(p) + \cdots + \alpha_3 r_{11}(p+3) &= (\alpha_0 + \cdots + \alpha_3) r_{11}(p) - (\alpha_1 + 2\alpha_2 + 3\alpha_3)\\ &\quad - (\alpha_1 + \alpha_2 + \alpha_3) v_2(p+1) - (\alpha_2 + \alpha_3) v_2(p+2) - \alpha_3 v_2(p+3)\\ &\quad + \alpha_1\Delta(p,1) + \alpha_2\Delta(p,2) + \alpha_3\Delta(p,3).\end{aligned}$$

Notons que $e((\alpha_0 + \cdots + \alpha_3) r_{11}(p)) = 1$ et $v_2(p+2) = 0$. Pour $1 \le i \le 3$, on pose $b_i = -\tilde{\alpha_i}$. Donc on a

$$\begin{aligned}\mathcal{S}_\alpha(N) &= \sum_{p\le N} e\left(\sum_{1\le i\le 3} b_i v_2(p+i) + \sum_{1\le i\le 3} \alpha_i \Delta(p,i)\right).\\ &= \sum_{p\le N} e\left(b_1 v_2(p+1) + b_3 v_2(p+3) + \alpha_1\Delta(p,1) + \alpha_2\Delta(p,2) + \alpha_3\Delta(p,3)\right).\end{aligned}$$

Soit $\Delta(p) := (\Delta(p,1), \Delta(p,2), \Delta(p,3))$. Rappelons que

$$\begin{cases} \Delta(p,2) = \Delta(p,1) + \Delta(p+1,1), \\ \Delta(p,3) = \Delta(p,1) + \Delta(p+1,1) + \Delta(p+2,1). \end{cases}$$

De plus, $\Delta(p,1), \Delta(p,3) \in \{0,1\}$ et $\Delta(p,2) \in \{-1,0\}$. Donc le vecteur $\Delta(p)$ a $2^3 = 8$ valeurs possibles. Nous listons toutes ces valeurs dans le tableau suivant :

$\Delta(p,1)$	$\Delta(p+1,1)$	$\Delta(p+2,1)$	$\Delta(p)$	
0	-1	0	$(0,-1,-1)$	$=\Delta_1$
1	-1	0	$(1,0,0)$	$=\Delta_2$
0	-1	1	$(0,-1,0)$	$=\Delta_3$
1	-1	1	$(1,0,1)$	$=\Delta_4$
0	0	0	$(0,0,0)$	$=\Delta_5$
0	0	1	$(0,0,1)$	$=\Delta_6$
1	0	0	$(1,1,1)$	$=\Delta_7$
1	0	1	$(1,1,2)$	$=\Delta_8$

TABLEAU VII.1. Valeurs possibles du vecteur $\Delta(p)$ pour $k=3$.

Soient $M=\lfloor \log_2(N+3)\rfloor$ et $\Theta_3=\{\Delta_j\ : 1\le j\le 8\}$. On a

$$\mathcal{S}_\alpha(N)=\sum_{\Delta\in\Theta_3}\sum_{\substack{0\le u_i\le M\\ 1\le i\le 3}} e(b_1u_1+b_3u_3)\sum_{\substack{p\le N\\ 2^{u_i}\|p+i\\ 1\le i\le 3\\ \Delta(p)=\Delta}} e\Big(\sum_{1\le i\le 3}\alpha_i\Delta(p,i)\Big). \tag{VII.5.2}$$

Le terme $e(\sum_{1\le i\le 3}\alpha_i\Delta(p,i))$ ne dépend pas de p dans la somme intérieure. Donc on a

$$S_1+S_2+\cdots+S_8,$$

où

$$S_j=e(\alpha\cdot\Delta_j)\sum_{\substack{0\le u_i\le M\\ 1\le i\le 3}} e(b_1u_1+b_3u_3)\sum_{\substack{p\le N\\ 2^{u_i}\|p+i\\ 1\le i\le 3\\ \Delta(p)=\Delta_j}} 1.$$

Premièrement, nous détaillons le calcul de S_1. D'après le Lemme VII.3.2, $\Delta(p)=\Delta_1$ si et seulement si

$$\begin{cases} p+1\equiv 2^{u_1} & (\mathrm{mod}\ 2^{u_1+2})\\ p+2\equiv 1 & (\mathrm{mod}\ 4)\\ p+3\equiv 2^{u_3} & (\mathrm{mod}\ 2^{u_3+2}).\end{cases}$$

Autrement dit, nous avons le système

$$\begin{cases} p\equiv 2^{u_1}-1 & (\mathrm{mod}\ 2^{u_1+2})\\ p\equiv 3 & (\mathrm{mod}\ 4)\\ p\equiv 2^{u_3}-3 & (\mathrm{mod}\ 2^{u_3+2}).\end{cases}$$

En particulier ce système implique $u_1 \geq 2$. D'après le théorème des restes chinois, il existe une solution au système précédent si et seulement si

$$2^{u_1} - 1 \equiv 2^{u_3} - 3 \pmod{2^{\min(u_1,u_3)+2}}. \tag{VII.5.3}$$

Dans ce cas, s'il y a une solution, elle appartient à une classe modulo $2^{\max(u_1,u_3)+2}$. Nous distinguons les cas suivants :

- Si $u_1 \geq u_3 + 2$, alors (VII.5.3) $\Leftrightarrow 2 \equiv 2^{u_3} \pmod{2^{u_3+2}} \Leftrightarrow u_3 = 1$.
- Si $u_1 = u_3 + 1$, alors (VII.5.3) $\Leftrightarrow 2^{u_3} + 2 \equiv 0 \pmod{2^{u_3+2}}$. Il n'y a pas de solutions.
- Si $u_1 = u_3$, alors (VII.5.3) $\Leftrightarrow 2 \equiv 0 \pmod{2^{u_3+2}}$. Il n'y a pas de solutions.
- Si $u_1 = u_3 - 1$, alors (VII.5.3) $\Leftrightarrow 2^{u_1} \equiv 2 \pmod{2^{u_1+2}} \Leftrightarrow u_1 = 1$. Il n'y a pas de solutions car $u_1 \geq 2$.
- Si $u_1 \leq u_3 - 2$, alors (VII.5.3) $\Leftrightarrow 2 \equiv -2^{u_1} \pmod{2^{u_1+2}}$. Il n'y a pas de solutions.

Par conséquent, $\Delta(p) = \Delta_1$ est équivalent à $u_1 \geq 3, u_3 = 1$ et $p \equiv \gamma_1 \pmod{2^{u_1+2}}$ où $\gamma_1 = 2^{u_1} - 1$.

De même, on a $\Delta(p) = \Delta_2$ si et seulement si $u_1 \geq 3, u_3 = 1$ et $p \equiv \gamma_2 \pmod{2^{u_1+2}}$ où $\gamma_2 = 2^{u_1} + 2^{u_1+1} - 1$.

Par conséquent, on a

$$\begin{aligned} S_1 + S_2 = e(-\alpha_2 - \alpha_3) \sum_{3 \leq u_1 \leq M} e(b_1 u_1 + b_3)\pi(N; 2^{u_1+2}, \gamma_1) \\ + e(\alpha_1) \sum_{3 \leq u_1 \leq M} e(b_1 u_1 + b_3)\pi(N; 2^{u_1+2}, \gamma_2). \end{aligned} \tag{VII.5.4}$$

D'après le Lemme VII.3.4, pour $A > 1$, on a

$$\begin{aligned} S_1 + S_2 &= (e(-\alpha_2 - \alpha_3) + e(\alpha_1))e(b_3)\pi(N) \sum_{3 \leq u_1 \leq M} \frac{e(b_1 u_1)}{2^{u_1+1}} + O\left(\frac{N}{\log^A N}\right), \\ &= (e(-\alpha_2 - \alpha_3) + e(\alpha_1))e(b_3)\frac{e(3b_1)}{2^3(2 - e(b_1))}\pi(N) + O\left(\frac{N}{\log^A N}\right). \end{aligned}$$

Ensuite, nous détaillons le calcul de S_3. D'après le Lemme VII.3.2, $\Delta(p) = \Delta_3$ si et seulement si

$$\begin{cases} p + 1 \equiv 2^{u_1} & \pmod{2^{u_1+2}} \\ p + 2 \equiv 1 & \pmod 4 \\ p + 3 \equiv 2^{u_3} + 2^{u_3+1} & \pmod{2^{u_3+2}}. \end{cases}$$

Autrement dit, nous avons le système suivant

$$\begin{cases} p \equiv 2^{u_1} - 1 & \pmod{2^{u_1+2}} \\ p \equiv 3 & \pmod 4 \\ p \equiv 2^{u_3} + 2^{u_3+1} - 3 & \pmod{2^{u_3+2}}. \end{cases}$$

En particulier ce système implique $u_1 \geq 2$. D'après le théorème des restes chinois, il existe une solution au système précédent si et seulement si

$$2^{u_1} - 1 \equiv 2^{u_3} + 2^{u_3+1} - 3 \pmod{2^{\min(u_1,u_3)+2}}. \tag{VII.5.5}$$

Nous distinguons les cas suivants :

- Si $u_1 \geq u_3 + 2$, alors (VII.5.5) $\Leftrightarrow 2 \equiv 2^{u_3} + 2^{u_3+2} \pmod{2^{u_3+2}}$. Il n'y a pas de solutions.
- Si $u_1 = u_3 + 1$, alors (VII.5.5) $\Leftrightarrow 2 \equiv 2^{u_3} \pmod{2^{u_3+2}} \Leftrightarrow u_3 = 1$.
- Si $u_1 = u_3$, alors (VII.5.5) $\Leftrightarrow 2 \equiv 0 \pmod{2^{u_3+2}}$. Il n'y a pas de solutions.
- Si $u_1 = u_3 - 1$, alors (VII.5.5) $\Leftrightarrow 2^{u_1+1} - 2^{u_1} \equiv 2 \pmod{2^{u_1+2}} \Leftrightarrow u_1 = 1$.
- Si $u_1 \leq u_3 - 2$, alors (VII.5.5) $\Leftrightarrow 2 \equiv -2^{u_1} \pmod{2^{u_1+2}}$. Il n'y a pas de solutions.

Par conséquent, $\Delta(p) = \Delta_3$ est équivalent à $u_3 = 1, u_1 = 2$ et $p \equiv \gamma_{3,1} \pmod{2^4}$ ou $u_3 = 2, u_1 = 1$ et $p \equiv \gamma_{3,2} \pmod{2^4}$, pour certains $\gamma_{3,1}, \gamma_{3,2}$ premiers avec 2^4.
De même, on a $\Delta(p) = \Delta_4$ est équivalent à $u_3 = 1, u_1 = 2$ et $p \equiv \gamma_{4,1} \pmod{2^4}$ ou $u_3 = 2, u_1 = 1$ et $p \equiv \gamma_{4,2} \pmod{2^4}$, pour certains $\gamma_{4,1}, \gamma_{4,2}$ premiers avec 2^4.
Par conséquent, on a

$$\begin{aligned} S_3 + S_4 &= e(-\alpha_2)(e(2b_1 + b_3)\pi(N; 2^4, \gamma_{3,1}) + e(b_1 + 2b_3)\pi(N; 2^4, \gamma_{3,2}) \\ &\quad + e(\alpha_1 + \alpha_3)(e(2b_1 + b_3)\pi(N; 2^4, \gamma_{4,1}) + e(b_1 + 2b_3)\pi(N; 2^4, \gamma_{4,2}) \\ &= (e(-\alpha_2) + e(\alpha_1 + \alpha_3))\frac{e(2b_1 + b_3) + e(b_1 + 2b_3)}{8}\pi(N) + O\left(\frac{N}{\log^A N}\right). \end{aligned}$$

En appliquant les mêmes arguments, on a

$$S_5 + S_6 = (1 + e(\alpha_3))\frac{e(b_1 + 2b_3)}{8}\pi(N) + O\left(\frac{N}{\log^A N}\right),$$

et

$$S_7 + S_8 = e(\alpha_1 + \alpha_2)(1 + e(\alpha_3))e(b_1)\frac{e(3b_3)}{2^3(2 - e(b_3))}\pi(N) + O\left(\frac{N}{\log^A N}\right).$$

En regroupant toutes les paires de sommes, on a donc

$$\mathcal{S}_\alpha(N) = Q(\alpha, 3)\pi(N) + O\left(\frac{N}{\log^A N}\right),$$

où

$$\begin{aligned} Q(\alpha, 3) &= \frac{e(2b_1 + b_3)}{8}\left(e(b_1)\,\frac{e(-\alpha_2 - \alpha_3) + e(\alpha_1)}{2 - e(b_1)} + e(-\alpha_2) + e(\alpha_1 + \alpha_3)\right) \\ &\quad + \frac{e(b_1 + 2b_3)}{8}\left(e(b_3)\,\frac{e(\alpha_1 + \alpha_2)(1 + e(\alpha_3))}{2 - e(b_3)} + 1 + e(-\alpha_2) + (1 + e(\alpha_1))e(\alpha_3)\right). \end{aligned} \tag{VII.5.6}$$

Le théorème est donc prouvé pour $k = 3$.

Remarque VII.5.1. Dans le Tableau VII.1, nous avons arrangé les sommes par groupe de 2. En effet, pour un groupe donné, les sommes de ce groupe contribuent de manière similaire à $\mathcal{S}_\alpha(N)$, voir l'égalité (VII.5.4). Pour étendre ce théorème à $k \geq 5$ impair, une compréhension de ces arrangements paraît nécessaire afin d'en déduire une forme concise de $Q(\alpha, k)$.

VII.6. Preuve du Théorème VII.2.1

Soient $k \geq 1$ un entier pair et $\alpha = (1/2, \ldots, 1/2) \in \mathbb{R}^{k+1}$. Donc $\tilde{\alpha_0} = \frac{k+1}{2}$ et $\|\tilde{\alpha_0}\| = 1/2$. D'après le Théorème VII.2.2, on a

$$\mathcal{S}_k(N) \ll (\log N)^K N^{1-\sigma},$$

pour des constantes $K, \sigma > 0$. Par conséquent, on a

$$\lim_{N\to+\infty} \frac{1}{\pi(N)} \mathcal{S}_k(N) = 0,$$

et le théorème est prouvé pour k pair.

Soient $k = 1$ et $\alpha = (1/2, 1/2) \in \mathbb{R}^2$. Donc $\tilde{\alpha_0} = 1$ et $\|\tilde{\alpha_0}\| = 0$. Le Théorème VII.2.2 est alors inutile et nous utilisons le Théorème VII.2.3 à sa place. D'après (VII.5.1), on a

$$Q(\alpha, 1) = e(1/2)\frac{1 + e(1/2)}{2(2e(1/2) - 1)} = 0,$$

car $e(1/2) = -1$. Par conséquent, d'après Théorème VII.2.3, on a

$$S_1(N) = O\left(\frac{N}{\log^A N}\right),$$

pour $A > 1$. On en déduit que

$$\lim_{N\to+\infty} \frac{1}{\pi(N)} \mathcal{S}_1(N) = 0,$$

et le théorème est prouvé pour $k = 1$.

Soit $k = 3$ et $\alpha = (1/2, 1/2, 1/2, 1/2) \in \mathbb{R}^4$. Donc $\|\tilde{\alpha_0}\| = 0$ et nous nous retrouvons dans la même situation que dans le cas $k = 1$. D'après (VII.5.6), on a $Q(\alpha, 3) = 0$ car

$$\begin{aligned}
&e(-\alpha_2 - \alpha_3) + e(\alpha_1) = 1 - 1 = 0,\\
&e(-\alpha_2) + e(\alpha_1 + \alpha_3) = -1 + 1 = 0,\\
&1 + e(\alpha_3) = 1 - 1 = 0,\\
&1 + e(-\alpha_2) + e(\alpha_3)(1 + e(\alpha_1)) = 1 - 1 + 1(1 - 1) = 0.
\end{aligned}$$

Par conséquent, d'après Théorème VII.2.3, on a

$$S_3(N) = O\left(\frac{N}{\log^A N}\right),$$

pour $A > 1$. On en déduit donc que

$$\lim_{N\to+\infty} \frac{1}{\pi(N)} \mathcal{S}_3(N) = 0,$$

et le théorème est prouvé pour $k = 3$.

VII.7. Preuve du Lemme VII.2.1

La preuve de ce lemme est similaire à celle de [MS98] et à celle du Théorème II.1.5, page 67.

PREUVE. Rappelons que pour tout $n \geq 0$, on a

$$\begin{cases} r'_{2n} = r'_n, \\ r'_{2n+1} = (-1)^n r'_n. \end{cases}$$

Dans un premier temps, on a

$$\begin{aligned} V(R_N, 2^M, D_1) &= \sum_{n<2^M} r'_n r'_{n+1} \cdots r'_{n+8k+3} \\ &= \sum_{2n<2^M} r'_{2n} \cdots r'_{2n+8k+3} + \sum_{2n+1<2^M} r'_{2n+1} \cdots r'_{2n+8k+4}. \end{aligned}$$

En regroupant les termes consécutifs par groupes de 2, on a

$$\begin{aligned} r'_{2n} \ldots r'_{2n+8k+3} &= (-1)^n (r'_n)^2 \ (-1)^{n+1} (r'_{n+1})^2 \ cdots \ (-1)^{n+4k+1} (r'_{n+4k+1})^2 \\ &= (-1)^{\sum_{i=0}^{4k+1} i} = -1, \end{aligned}$$

car $\sum_{i=0}^{4k+1} i = (4k+1)(2k+1)$ est un entier impair. Avec le lemme raisonnement, on a

$$\begin{aligned} r'_{2n+1} \cdots r'_{2n+8k+4} &= (-1)^n r'_n \ (-1)^{n+1} (r'_{n+1})^2 \ \cdots \ (-1)^{n+4k+1} (r'_{n+4k+1})^2 \ r'_{n+4k+2} \\ &= (-1)^{\sum_{i=0}^{4k+1} i} r'_n r'_{n+4k+2} = -r'_n r'_{n+4k+2}. \end{aligned}$$

Par conséquent, on a $V(R_N, 2^M, D_1) = -2^{M-1} - \gamma_M(4k+2)$ où $\gamma_M(k) = \sum_{n<2^M} r'_n r'_{n+k}$. D'après [MS98], on a $\gamma_M(2k) = (1 + (-1)^k)\gamma_{M-1}(k)$. Cela mène donc à

$$\begin{aligned} V(R_N, 2^M, D_1) &= -2^{M-1} - \gamma_M(4k+2) \\ &= -2^{M-1} - (1 + (-1)^{2k+1})\gamma_{M-1}(2k+1) = -2^{M-1} < -\frac{1}{8}N. \end{aligned}$$

La preuve de la deuxième partie est similaire. On a

$$\begin{aligned} V(R_N, 2^M, D_2) &= \sum_{n<2^M} r'_n r'_{n+1} \cdots r'_{n+8k+7} \\ &= \sum_{2n<2^M} r'_{2n} \cdots r'_{2n+8k+7} + \sum_{2n+1<2^M} r'_{2n+1} \cdots r'_{2n+8k+8} \\ &= \sum_{n<2^{M-1}} (-1)^{\sum_{i=0}^{4k+3} i} + \sum_{n<2^{M-1}} (-1)^{\sum_{i=0}^{4k+3} i} r'_n r'_{n+4k+4}. \end{aligned}$$

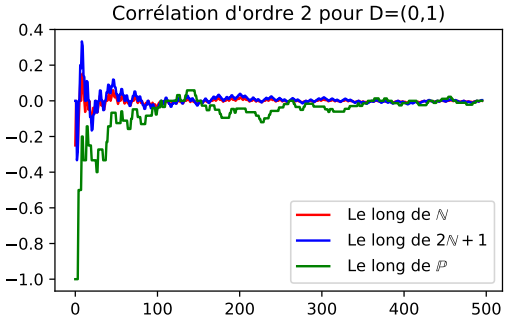
Or $\sum_{i=0}^{4k+3} i = 2(4k+1)(k+1)$ est un entier pair, on a

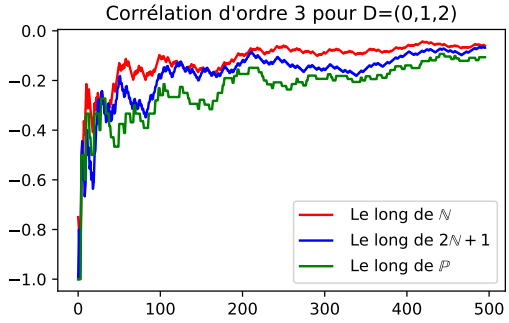
$$V(R_N, 2^M, D_2) = 2^{M-1} + \gamma_M(4k+6) = 2^{M-1} + (1+(-1)^{2k+3})\gamma_{M-1}(4k+6)$$
$$= 2^{M-1} > \frac{1}{8}N.$$

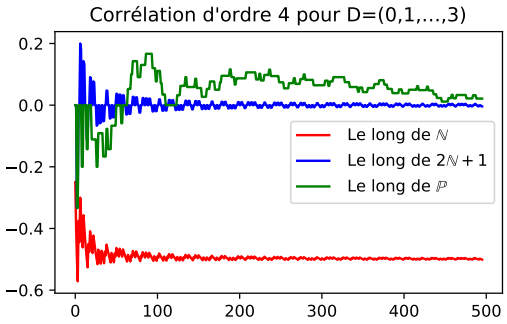
□

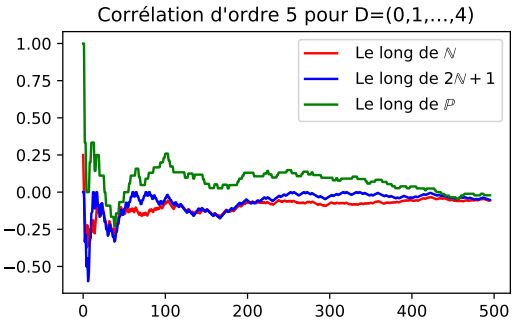
VII.8. Simulations

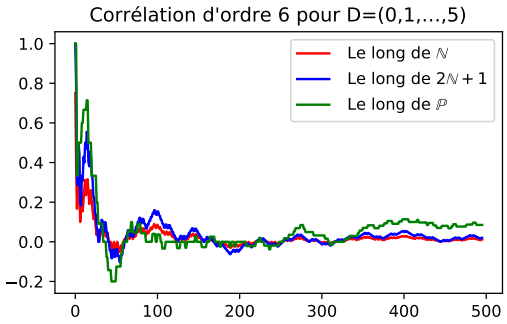
Nous avons réalisé les simulations suivantes des corrélations d'ordre k du vecteur $D = (0, 1, \ldots, k-1)$ de la suite $\mathcal{R}'$, de la suite $\mathcal{R}'$ le long des nombres impairs et de la suite $\mathcal{R}'$ le long des nombres premiers, pour $N \leq 500$. Nous avons normalisé les sommes respectivement par N, $N/2$ et $\pi(N)$.

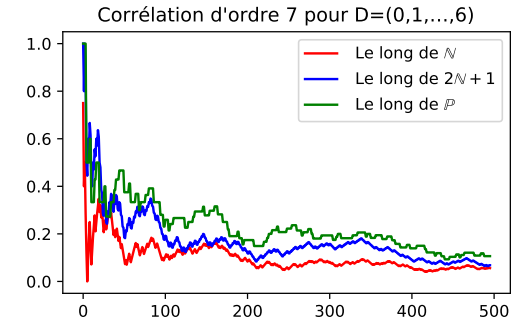


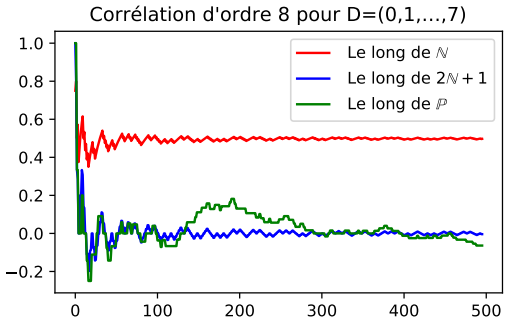


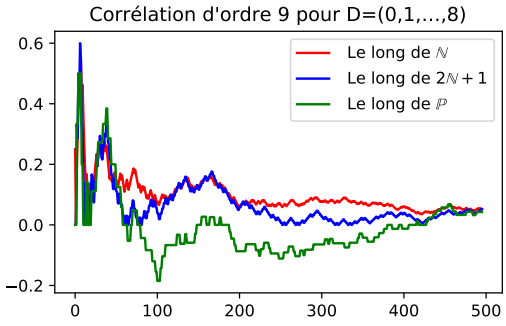


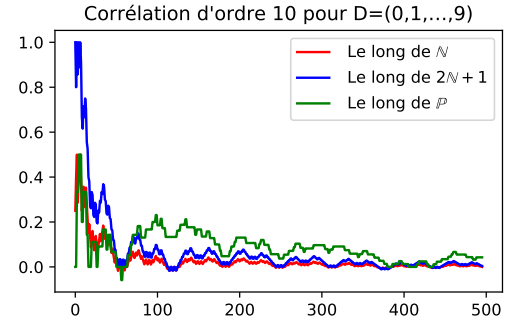


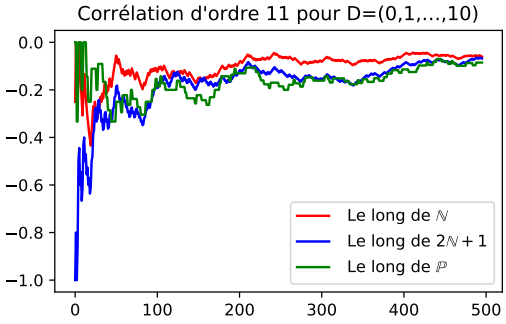


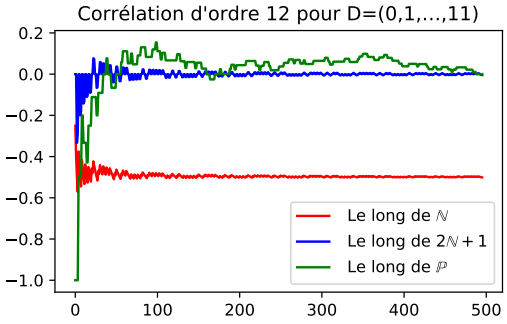


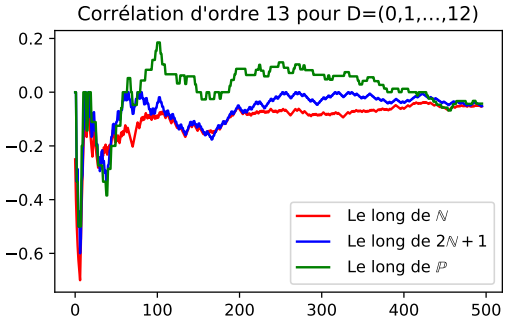

Corrélation d'ordre 8 pour D=(0,1,...,7)
Le long de $\mathbb{N}$
Le long de $2\mathbb{N}+1$
Le long de $\mathbb{P}$

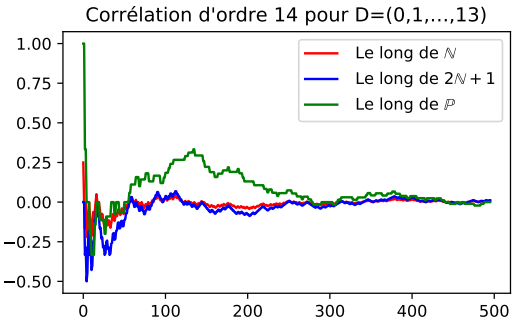

Corrélation d'ordre 9 pour D=(0,1,...,8)
Le long de $\mathbb{N}$
Le long de $2\mathbb{N}+1$
Le long de $\mathbb{P}$

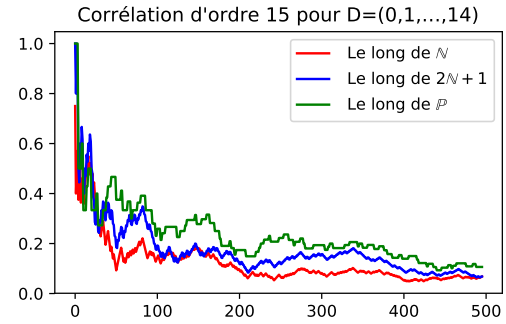

Corrélation d'ordre 10 pour D=(0,1,...,9)
Le long de $\mathbb{N}$
Le long de $2\mathbb{N}+1$
Le long de $\mathbb{P}$

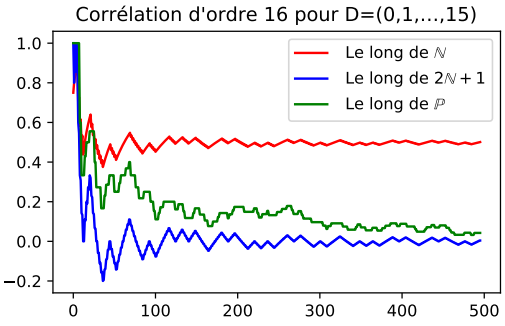

Corrélation d'ordre 11 pour D=(0,1,...,10)
Le long de $\mathbb{N}$
Le long de $2\mathbb{N}+1$
Le long de $\mathbb{P}$

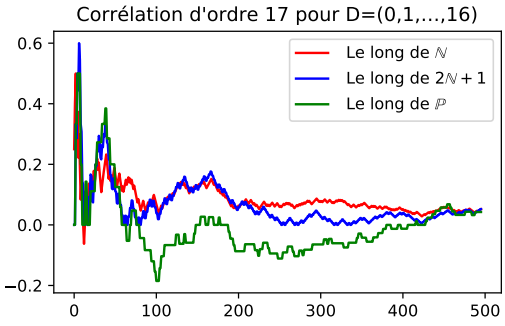

Corrélation d'ordre 12 pour D=(0,1,...,11)
Le long de $\mathbb{N}$
Le long de $2\mathbb{N}+1$
Le long de $\mathbb{P}$


Corrélation d'ordre 13 pour D=(0,1,...,12)
Le long de $\mathbb{N}$
Le long de $2\mathbb{N}+1$
Le long de $\mathbb{P}$


Corrélation d'ordre 14 pour D=(0,1,...,13)
Le long de $\mathbb{N}$
Le long de $2\mathbb{N}+1$
Le long de $\mathbb{P}$


Corrélation d'ordre 15 pour D=(0,1,...,14)
Le long de $\mathbb{N}$
Le long de $2\mathbb{N}+1$
Le long de $\mathbb{P}$





Nous retrouvons bien le fait que la corrélation d'ordre k de la suite $\mathcal{R}'$ est grande pour $k \equiv 0 \pmod 4$. Nous constatons également que les corrélations le long des nombres premiers et le long des nombres impairs semblent avoir la même limite pour tout $k \geq 1$. Nous formulons alors la conjecture suivante, analogue du Théorème VII.1.1.

Conjecture VII.8.1. *Soit $k \geq 1$ un entier, on a*

$$\lim_{N\to+\infty} \frac{1}{\pi(N)} \sum_{p\leq N} r'_p r'_{p+1} \cdots r'_{p+k} = \lim_{N\to+\infty} \frac{1}{N} \sum_{\substack{n\leq N \\ n\equiv 1 \pmod 2}} r'_n r'_{n+1} \cdots r'_{n+k}.$$

VII.9. Ouvertures

Nous finissons ce chapitre en évoquant des problèmes ouverts. Soit $\ell > 1$, étant donné l'entier n, on note $r'_{\ell,n} = (-1)^{r_{1^\ell}(n)}$ où $r_{1^\ell}(n)$ est le nombre d'occurrences de 1^ℓ dans $(n)_2$.

Problème VII.9.1. Soit $k \geq 1$ un entier. Étudier la somme

$$\sum_{p\leq N} r'_{\ell,p} r'_{\ell,p+1} \cdots r'_{\ell,p+k}.$$

Ce problème semble plus difficile que l'étude réalisée dans ce chapitre, qui correspond au cas $\ell = 2$.

Les problèmes suivants sont ardus et sont formulés dans [DMR19].

Problème VII.9.2. Soient $P \in \mathbb{Z}[X]$ tel que $P(\mathbb{N}) \subset \mathbb{N}$ de degré $d \geq 3$. Montrer que la suite $\mathcal{T}_P$, la suite de Thue–Morse le long du polynôme P, est normale.

Problème VII.9.3. Montrer que la suite $\mathcal{T}_{\mathbb{P}}$, la suite de Thue–Morse le long des nombres premiers, est normale.

Nous pouvons également formuler ces problèmes en remplaçant la suite de Thue–Morse par la suite de motifs d'ordre k, voir la Définition I.3.8.

ANNEXE A

Programmes Python et SageMath

Dans cette annexe, nous donnons le code SageMath qui nous a permis de réaliser certaines des figures et simulations du manuscrit.

```
import numpy

def t(n): #Suite de Thue--Morse
    if sum(n.digits(2))%2==0: return(0)
    return(1)

def r(n): #Suite de Rudin--Shapiro
    x=0
    while n>1:
        if (n+1)%4==0:
            x=x+1
            n=int(n/2)
        else:
            n=int(n/2)
    return x%2

def z(n): #Decomposition en base de Zeckendorf
     if n == 0 : return [0]
     fib = [2,1]
     while fib[0] < n: fib[0:0] = [sum(fib[:2])]
     dig = []
     for f in fib:
         if f <= n:
             dig, n = dig + [1], n - f
         else:
             dig += [0]
     return dig if dig[0] else dig[1:]

def sZ(n): #Somme des chiffres en base de Zeckendorf modulo 2
     return mod(add(z(n)),2)
```

Le programme suivant utilise l'implémentation des outils de la combinatoire des mots dans Sage.

```
phi = WordMorphism('a->ab,b->a') #Mot infini de Fibonacci
h = WordMorphism('a->0,b->1')
w = phi.fixed_point('a')
v=(h(w)[0:10000])
def f(n):
    return(v[n])

psi = WordMorphism('0->03,1->43,3->1,4->01') #Morphisme de la Figure
                                             III.2
w2 = psi.fixed_point('0')
v2=w2[0:10000]
def f2(n):
    return(v2[n])
```

Puis pour les représenter on utilise le code suivant, où f désigne la suite qu'on étudie et C le nombre de colonnes/lignes fixé (ici fixé à 64 par exemple).

```
import matplotlib.pyplot as plt
C=64 #Nombre de colonnes
R=C #Nombre de lignes
L=numpy.zeros([R,C])

for i in numpy.arange(0,R):
    for j in numpy.arange(0,C):
        x=int(C*i+j)+0
        L[i,j]=f(x)

plt.tick_params(axis='x', which='both', bottom=False,
                top=False, labelbottom=False)
plt.tick_params(axis='y', which='both', right=False,
                left=False, labelleft=False)

fig=plt.imshow(L,cmap='Greys') #pour une suite binaire
fig=plt.imshow(L,cmap='viridis') #pour une suite sur un autre alphabet
```

Pour la complexité linéaire, nous avons utilisé le programme suivant, qui convient pour les suites binaires.

```
def BM(s):
    n = len(s)
    C = [GF(2)(1)]
    B = [GF(2)(1)]
    temp = []
    T = []
    L,N=0,0
    m = -1
    y=[]
    while N < n:
        temp = B
        d = s[N]
        for i in range(1,L+1):
            d = d + C[i]*s[N-i]
        if d == 1:
            T = C
            temp = [ 0 for i in range(int(N-m))] + temp
            if len(C) < len(temp):
                C = C + [0 for i in range(len(temp)-len(C))]
            else:
                temp = temp + [0 for i in range(len(C)-len(temp))]

            for i in range(len(C)):
                C[i] = C[i] + temp[i]

            if L <= N/2:
                L = int(N + 1 - L)
                m = int(N)
                B = T
        y.append(L)
        N = N + 1
    return(y)
```

Pour les estimations des corrélations d'ordre k de la suite $\mathcal{R}'$ du Paragraphe VII.8, nous avons utilisé le programme suivant pour le calcul des sommes.

```
def pi(N):
    c=0
    for n in range(N):
        if is_prime(n):
            c=c+1
    return(c)

def r11(n):
     return sum([((n>>i)&3==3) for i in range(len(bin(n)[2:]) - 1)])

def S(N,D,b):
    s=0
    if b=1:
      for n in range(1,N):
          m=0
          for i in range(len(D)):
              m=m+r11(n+D[i])
          s=s+(-1)**(m)
      s=float(s/(N))
      return(s)
    else:
      for n in range(1,N,2):
          m=0
           for i in range(len(D)):
              m=m+r11(n+D[i])
          s=s+(-1)**(m)
       s=float(2*s/(N))
      return(s)

def SP(N,D):
    s=0
    for n in range(N):
        if is_prime(n):
            m=0
            for i in range(len(D)):
                m=m+r11(n+D[i])
            s=s+(-1)**(m)
    s=float(s/(pi(N)))
    return(s)
```

Nous avons utilisé le programme suivant pour réaliser les simulations.

```
for k in range(2,17):
u,v,w=[],[],[]
D=[i for i in range(k)]
for N in range(4,500):
    u.append(S(N,D,1))
    v.append(S(N,D,2))
    w.append(SP(N,D))
fig = plt.figure(1, figsize=(5, 3))
ax=plt.gca()
ax.plot(u,label=' Le long de $\mathbb{N}$',color='r')
ax.plot(v,label=' Le long de $2\mathbb{N}+1$',color='b')
ax.plot(w,label=' Le long de $\mathbb{P}$ ',color='g')
ax.legend()
```

Bibliographie

[ACSZ17] J-P. Allouche, J. Cassaigne, J. O. Shallit, and L. Q. Zamboni. A taxonomy of morphic sequences. 2017. `https://arxiv.org/abs/1711.10807`.

[ADM22] B. Adamczewski, M. Drmota, and C. Müllner. (Logarithmic) densities for automatic sequences along primes and squares. *Trans. Amer. Math. Soc.*, 375(1):455–499, 2022.

[ADQ21] J-P. Allouche, M. Dekking, and M. Queffélec. Hidden automatic sequences. *Comb. Theory*, 1:Paper No. 20, 15, 2021.

[AHN20] J-P. Allouche, G.-N. Han, and H. Niederreiter. Perfect linear complexity profile and apwenian sequences. *Finite Fields Appl.*, 68:101761, 13, 2020.

[AJK+22] K. Aloui, D. Jamet, H. Kaneko, S. Kopecki, P. Popoli, and T. Stoll. On the binary digits of n and n^2, 2022. `https://arxiv.org/abs/2203.05451`, accepted in *Theoret. Comput. Sci.* (10/10/2022).

[AKM+07] N. Alon, Y. Kohayakawa, C. Mauduit, C. G. Moreira, and V. Rödl. Measures of pseudorandomness for finite sequences: typical values. *Proc. Lond. Math. Soc. (3)*, 95(3):778–812, 2007.

[All82] J-P. Allouche. Somme des chiffres et transcendance. *Bull. Soc. Math. France*, 110(3):279–285, 1982.

[All16] J-P. Allouche. On a Golay-Shapiro-like sequence. *Unif. Distrib. Theory*, 11(2):205–210, 2016.

[AMM21] K. Aloui, C. Mauduit, and M. Mkaouar. On the correlation of the sum of digits along prime numbers. *Acta Arith.*, 199(3):275–301, 2021.

[AS93] J-P. Allouche and O. Salon. Sous-suites polynomiales de certaines suites automatiques. *J. Théor. Nombres Bordeaux*, 5(1):111–121, 1993.

[AS99] J-P. Allouche and J. O. Shallit. The ubiquitous Prouhet-Thue-Morse sequence. In *Sequences and their applications (Singapore, 1998)*, Springer Ser. Discrete Math. Theor. Comput. Sci., pages 1–16. Springer, London, 1999.

[AS03] J-P. Allouche and J. O. Shallit. *Automatic Sequences : Theory, Applications, Generalizations.* Cambridge University Press, Cambridge, 2003.

[ASY22] J-P. Allouche, J. O. Shallit, and R. Yassawi. How to prove that a sequence is not automatic. *Expo. Math.*, 40(1):1–22, 2022.

[AUFP13] C. Ahlbach, J. Usatine, C. Frougny, and N. Pippenger. Efficient algorithms for Zeckendorf arithmetic. *Fibonacci Quart.*, 51(3):249–255, 2013.

[BB14] M. A. Bennett and Y. Bugeaud. Perfect powers with three digits. *Mathematika*, 60(1):66–84, 2014.

[BBE+83] A. Blumer, J. Blumer, A. Ehrenfeucht, D. Haussler, and R. M. McConnell. Linear size finite automata for the set of all subwords of a word - an outline of results. *Bull. EATCS*, 21:12–20, 1983.

[BBH+85] A. Blumer, J. Blumer, D. Haussler, A. Ehrenfeucht, M. T. Chen, and J. Seiferas. The smallest automaton recognizing the subwords of a text. volume 40, pages 31–55. 1985. Special issue: Eleventh international colloquium on automata, languages and programming (Antwerp, 1984).

[BBM12] M. A. Bennett, Y. Bugeaud, and M. Mignotte. Perfect powers with few binary digits and related Diophantine problems, II. *Math. Proc. Cambridge Philos. Soc.*, 153(3):525–540, 2012.

[Ben17] M. A. Bennett. The polynomial-exponential equation $1 + 2^a + 6^b = y^q$. *Period. Math. Hungar.*, 75(2):387–397, 2017.

[Ber79] J. Berstel. Sur la construction de mots sans carré. (18):15, 1979.

[BHMP16] A. Bérczes, L. Hajdu, T. Miyazaki, and I. Pink. On the Diophantine equation $1+x^a+z^b = y^n$. *J. Comb. Number Theory*, 8(2):145–154, 2016.

[BK95] N. L. Bassily and I. Kátai. Distribution of the values of q-additive functions on polynomial sequences. *Acta Math. Hung.*, 68(4):353–361, 1995.

[BMS06] Y. Bugeaud, M. Mignotte, and S. Siksek. Classical and modular approaches to exponential Diophantine equations. I. Fibonacci and Lucas perfect powers. *Ann. of Math. (2)*, 163(3):969–1018, 2006.

[Bor09] E. Borel. Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 27(1):247–271, 1909.

[Brl89] S. Brlek. Enumeration of factors in the Thue-Morse word. volume 24, pages 83–96. 1989. First Montreal Conference on Combinatorics and Computer Science, 1987.

[BRS+10] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, and S. Vo. Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, Gaithersburg, MD, USA, 2010.

[Bru95] V. Bruyère. Automata and numeration systems. *Sém. Lothar. Combin.*, 35:Art. B35b, approx. 19, 1995.

[BW06] N. Brandstätter and A. Winterhof. Linear complexity profile of binary sequences with small correlation measure. *Period. Math. Hungar.*, 52(2):1–8, 2006.

[CCSS14] J. Cassaigne, J. D. Currie, L. Schaeffer, and J. O. Shallit. Avoiding three consecutive blocks of the same size and same sum. *J. ACM*, 61(2):Art. 10, 17, 2014.

[CE46] A. H. Copeland and P. Erdős. Note on normal numbers. *Bull. Amer. Math. Soc.*, 52:857–860, 1946.

[Cha33] D. G. Champernowne. The construction of decimals normal in the scale of ten. *J. London Math. Soc.*, 8(4):254–260, 1933.

[Chr79] G. Christol. Ensembles presque periodiques k-reconnaissables. *Theoret. Comput. Sci.*, 9(1):141–145, 1979.

[CMS02] J. Cassaigne, C. Mauduit, and A. Sárközy. On finite pseudorandom binary sequences. VII. The measures of pseudorandomness. *Acta Arith.*, 103(2):97–118, 2002.

[Cob69] A. Cobham. On the base-dependence of sets of numbers recognizable by finite automata. *Math. Systems Theory*, 3:186–192, 1969.

[Cob72] A. Cobham. Uniform tag sequences. *Math. Systems Theory*, 6:164–192, 1972.

[CZ13] P. Corvaja and U. Zannier. Finiteness of odd perfect powers with four nonzero binary digits. *Ann. Inst. Fourier (Grenoble)*, 63(2):715–731, 2013.

[Dav08] H. Davenport. *The Higher Arithmetic*. Cambridge University Press, Cambridge, eighth edition, 2008. An introduction to the theory of numbers, With editing and additional material by James H. Davenport.

[Dek21] F. Michel Dekking. The sum of digits functions of the Zeckendorf and the base phi expansions. *Theoret. Comput. Sci.*, 859:70–79, 2021.

[Die12] C. Diem. On the use of expansion series for stream ciphers. *LMS J. Comput. Math.*, 15:326–340, 2012.

[dLV89] Aldo de Luca and Stefano Varricchio. Some combinatorial properties of the Thue-Morse sequence and a problem in semigroups. *Theoret. Comput. Sci.*, 63(3):333–348, 1989.

[DMR19] M. Drmota, C. Mauduit, and J. Rivat. Normality along squares. *J. Eur. Math. Soc. (JEMS)*, 21(2):507–548, 2019.

[DMS18] M. Drmota, C. Müllner, and L. Spiegelhofer. Möbius orthogonality for the Zeckendorf sum-of-digits function. *Proc. Amer. Math. Soc.*, 146(9):3679–3691, 2018.

[DMS21] M. Drmota, C. Müllner, and L. Spiegelhofer. Primes as sums of Fibonacci numbers, 2021. Accepted for publication in Mem. Amer. Math. Soc. `https://arxiv.org/abs/2109.04068`.

[DT97] M. Drmota and R. F. Tichy. *Sequences, Discrepancies and Applications*, volume 1651 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1997.

[DT05] C. Dartyge and G. Tenenbaum. Sommes des chiffres de multiples d'entiers. *Ann. Inst. Fourier (Grenoble)*, 55(7):2423–2474, 2005.

[DT06] C. Dartyge and G. Tenenbaum. Congruences de sommes de chiffres de valeurs polynomiales. *Bull. London Math. Soc.*, 38(1):61–69, 2006.

[Fog02] N. P. Fogg. *Substitutions in Dynamics, Arithmetics and Combinatorics*, volume 1794 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2002.

[Gel68] A. O. Gelfond. Sur les nombres qui ont des propriétés additives et multiplicatives données. *Acta Arith.*, 13:259–265, 1967/68.

[Gol51] M. J. E. Golay. Static multislit spectrometry and its application to the panoramic display of infrared spectra∗. *J. Opt. Soc. Am.*, 41(7):468–472, 1951.

[Gol67] S. W. Golomb. *Shift Register Sequences*. With portions co-authored by Lloyd R. Welch, Richard M. Goldstein, and Alfred W. Hales. Holden-Day, Inc., San Francisco, Calif.-Cambridge-Amsterdam, 1967.

[GPM20] D. Gómez-Pérez and L. Mérai. Algebraic dependence in generating functions and expansion complexity. *Adv. Math. Commun.*, 14(2):307–318, 2020.

[GPMN18] D. Gómez-Pérez, L. Mérai, and H. Niederreiter. On the expansion complexity of sequences over finite fields. *IEEE Trans. Inform. Theory*, 64(6):4228–4232, 2018.

[Gya13] K Gyarmati. Measures of pseudorandomness. In *Finite fields and their applications*, volume 11 of *Radon Ser. Comput. Appl. Math.*, pages 43–64. De Gruyter, Berlin, 2013.

[HLS11a] K. G. Hare, S. Laishram, and T. Stoll. Stolarsky's conjecture and the sum of digits of polynomial values. *Proc. Amer. Math. Soc.*, 139(1):39–49, 2011.

[HLS11b] K. G. Hare, S. Laishram, and T. Stoll. The sum of digits of n and n^2. *Int. J. Number Theory*, 7(7):1737–1752, 2011.

[HP14] L. Hajdu and I. Pink. On the Diophantine equation $1 + 2^a + x^b = y^n$. *J. Number Theory*, 143:1–13, 2014.

[HW08] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, Oxford, sixth edition, 2008. Revised by D. R. Heath-Brown and J. H. Silverman, With a foreword by Andrew Wiles.

[IK01] K-H. Indlekofer and I. Kátai. Investigations in the theory of q-additive and q-multiplicative functions. I. *Acta Math. Hungar.*, 91(1-2):53–78, 2001.

[IK04] H. Iwaniec and E. Kowalski. *Analytic Number Theory*, volume 53 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, RI, 2004.

[IW17] L Işık and A Winterhof. Maximum-order complexity and correlation measures. *Cryptography*, 1, 03 2017.

[Jan89] C. J. A. Jansen. *Investigations on nonlinear streamcipher systems: Construction and evaluation methods*. ProQuest LLC, Ann Arbor, MI, 1989. Thesis (Dr.)–Technische Universiteit Delft (The Netherlands).

[Jan91] C. J. A. Jansen. The maximum order complexity of sequence ensembles. In Donald W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, pages 153–159, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.

[JPS21] D. Jamet, P. Popoli, and T. Stoll. Maximum order complexity of the sum of digits function in Zeckendorf base and polynomial subsequences. *Cryptogr. Commun.*, 13(5):791–814, 2021. `https://link.springer.com/article/10.1007/s12095-021-00507-w`.

[Jun93] D. Jungnickel. *Finite Fields*. Bibliographisches Institut, Mannheim, 1993. Structure and arithmetics.

[KN74] L. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences*. Pure and Applied Mathematics. Wiley-Interscience [John Wiley & Sons], New York-London-Sydney, 1974.

[Knu97] D. E. Knuth. *The Art of Computer Programming. Vol. 2*. Addison-Wesley, Reading, MA, 1997. Fundamental algorithms, Third edition.

[KS22] H. Kaneko and T. Stoll. Products of integers with few nonzero digits. *Unif. Distrib. Theory*, 17(1):11–28, 2022.

[Kur03] P Kurka. *Topological and Symbolic Dynamics*, volume 11 of *Cours Spécialisés [Specialized Courses]*. Société Mathématique de France, Paris, 2003.

[Leh51] D. H. Lehmer. Mathematical methods in large-scale computing units. In *Proceedings of a Second Symposium on Large-Scale Digital Calculating Machinery, 1949*, pages 141–146. Harvard University Press, Cambridge, Mass., 1951.

[Lek51] C. G. Lekkerkerker. *Voorstelling van natuurlijke getallen door een som van getallen van fibonacci*. Stichting Mathematisch Centrum, 1951. Issue: ZW 30/51 Publication Title: Stichting Mathematisch Centrum. Zuivere Wiskunde Type: urn:NBN:nl:ui:18-6922.

[Lin97] B. Lindström. On the binary digits of a power. *J. Number Theory*, 65(2):321–324, 1997.

[LN96] R. Lidl and H. Niederreiter. Finite fields and their applications. In *Handbook of algebra, Vol. 1*, volume 1 of *Handb. Algebr.*, pages 321–363. Elsevier/North-Holland, Amsterdam, 1996.

[Lot97] M. Lothaire. *Combinatorics on Words*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 1997.

[Lot02] M. Lothaire. *Algebraic Combinatorics on Words*, volume 90 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2002.

[LS07] P. L'Ecuyer and R. Simard. TestU01: a C library for empirical testing of random number generators. *ACM Trans. Math. Software*, 33(4):Art. 22, 40, 2007.

[Luc04] F. Luca. The Diophantine equation $x^2 = p^a \pm p^b + 1$. *Acta Arith.*, 112(1):87–101, 2004.

[Mah27] K. Mahler. The spectrum of an array and its application to the study of the translation properties of a simple class of arithmetical functions: Part two on the translation properties of a simple class of arithmetical functions. *Journal of Mathematics and Physics*, 6(1-4):158–163, 1927.

[Mau01] C. Mauduit. Multiplicative properties of the Thue-Morse sequence. *Period. Math. Hungar.*, 43(1-2):137–153, 2001.

[Mei15] S.-Y. Mei. The sum of digits of polynomial values. *Integers*, 15:Paper No. A32, 12, 2015.

[Mel05] G. Melfi. On simultaneous binary expansions of n and n^2. *J. Number Theory*, 111(2):248–256, 2005.

[MH38] M. Morse and G. A. Hedlund. Symbolic Dynamics. *Amer. J. Math.*, 60(4):815–866, 1938.

[MMR15] B. Martin, C. Mauduit, and J. Rivat. Fonctions digitales le long des nombres premiers. *Acta Arith.*, 170(2):175–197, 2015.

[MNW17] L. Mérai, H. Niederreiter, and A. Winterhof. Expansion complexity and linear complexity of sequences over finite fields. *Cryptogr. Commun.*, 9(4):501–509, 2017.

[Mor21] H. M. Morse. Recurrent geodesics on a surface of negative curvature. *Trans. Amer. Math. Soc.*, 22(1):84–100, 1921.

[Mos07] Y. Moshe. On the subword complexity of Thue-Morse polynomial extractions. *Theoret. Comput. Sci.*, 389(1-2):318–329, 2007.

[Mou16] H. Mousavi. Automatic theorem proving in Walnut. *CoRR*, abs/1603.06017, 2016.

[MR09] C. Mauduit and J. Rivat. La somme des chiffres des carrés. *Acta Math.*, 203(1):107–148, 2009.

[MR10] C. Mauduit and J. Rivat. Sur un problème de Gelfond: la somme des chiffres des nombres premiers. *Ann. of Math. (2)*, 171(3):1591–1646, 2010.

[MR15] C. Mauduit and J. Rivat. Prime numbers along Rudin-Shapiro sequences. *J. Eur. Math. Soc. (JEMS)*, 17(10):2595–2642, 2015.

[MR18] C. Mauduit and J. Rivat. Rudin-Shapiro sequences along squares. *Trans. Amer. Math. Soc.*, 370(11):7899–7921, 2018.

[MS97] C. Mauduit and A. Sárközy. On finite pseudorandom binary sequences. I. Measure of pseudorandomness, the Legendre symbol. *Acta Arith.*, 82(4):365–377, 1997.

[MS98] C. Mauduit and A. Sárközy. On finite pseudorandom binary sequences. II. The Champernowne, Rudin-Shapiro, and Thue-Morse sequences, a further construction. *J. Number Theory*, 73(2):256–276, 1998.

[MS14] M. Madritsch and T. Stoll. On simultaneous digital expansions of polynomial values. *Acta Math. Hungar.*, 143(1):192–200, 2014.

[MS17] C. Müllner and L. Spiegelhofer. Normality of the Thue-Morse sequence along Piatetski-Shapiro sequences, II. *Israel J. Math.*, 220(2):691–738, 2017.

[Mül17] C. Müllner. Automatic sequences fulfill the Sarnak conjecture. *Duke Math. J.*, 166(17):3219–3290, 2017.

[Mül18] C. Müllner. The Rudin-Shapiro sequence and similar sequences are normal along squares. *Canad. J. Math.*, 70(5):1096–1129, 2018.

[MW18a] L. Mérai and A. Winterhof. On the Nth linear complexity of automatic sequences. *J. Number Theory*, 187:415–429, 2018.

[MW18b] L. Mérai and A. Winterhof. On the pseudorandomness of automatic sequences. *Cryptogr. Commun.*, 10(6):1013–1022, 2018.

[MW22] L. Mérai and A. Winterhof. Pseudorandom sequences derived from automatic sequences. *Cryptogr. Commun.*, 14(4):783–815, 2022.

[Nie88] H. Niederreiter. The probabilistic theory of linear complexity. In *Advances in cryptology—EUROCRYPT '88 (Davos, 1988)*, volume 330 of *Lecture Notes in Comput. Sci.*, pages 191–209. Springer, Berlin, 1988.

[Nie90] H. Niederreiter. A combinatorial approach to probabilistic results on the linear-complexity profile of random sequences. *J. Cryptology*, 2(2):105–112, 1990.

[Nie92] H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[NX14] H. Niederreiter and C. Xing. Sequences with high nonlinear complexity. *IEEE Trans. Inform. Theory*, 60(10):6696–6701, 2014.

[OEI22] OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences, 2022. Published electronically at `http://oeis.org`.

[Pet02] M. Peter. The summatory function of the sum-of-digits function on polynomial sequences. *Acta Arith.*, 104(1):85–96, 2002.

[PH12] I. Palacios-Huerta. Tournaments, Fairness And The Prouhet-Thue-Morse Sequence. *Economic Inquiry*, 50(3):848–849, July 2012.

[Pop20] P. Popoli. On the maximum order complexity of Thue-Morse and Rudin-Shapiro sequences along polynomial values. *Unif. Distrib. Theory*, 15(2):9–22, 2020. `https://sciendo.com/fr/article/10.2478/udt-2020-0008`.

[Pro51] E. Prouhet. Mémoire sur quelques relations entre les puissances des nombres. *C. R. Acad. Sc. Paris*, 33:225, 1851.

[Que87] M. Queffélec. *Substitution Dynamical Systems — Spectral Analysis*, volume 1294 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1987.

[Rig14] M. Rigo. *Formal Languages, Automata and Numeration Systems. 1*. ISTE, London; John Wiley & Sons, Inc., Hoboken, NJ, 2014. Introduction to combinatorics on words, With a foreword by Valérie Bethé.

[Rit63] R. W. Ritchie. Finite automata and the set of squares. *J. Assoc. Comput. Mach.*, 10:528–531, 1963.

[Rud59] W. Rudin. Some theorems on Fourier coefficients. *Proc. Amer. Math. Soc.*, 10:855–859, 1959.

[Rue86] R. A. Rueppel. *Analysis and design of stream ciphers*. Communications and Control Engineering Series. Springer-Verlag, Berlin, 1986. With a foreword by James L. Massey.

[Saf86] B. Saffari. Une fonction extrémale liée à la suite de Rudin-Shapiro. *C. R. Acad. Sci. Paris Sér. I Math.*, 303(4):97–100, 1986.

[Sar11] P. Sarnak. Three lectures on the Möbius function randomness and dynamics. 2011. `https://publications.ias.edu/sites/default/files/MobiusFunctionsLectures(2).pdf`.

[Sau15] J. C. Saunders. Sums of digits in q-ary expansions. *Int. J. Number Theory*, 11(2):593–611, 2015.

[Sha53] H. S. Shapiro. *Extremal Problems for Polynomials and Power Series*. ProQuest LLC, Ann Arbor, MI, 1953. Thesis (Ph.D.)–Massachusetts Institute of Technology.

[Sha21] J. O. Shallit. Subword complexity of the Fibonacci-Thue-Morse sequence: the proof of Dekking's conjecture. *Indag. Math. (N.S.)*, 32(3):729–735, 2021.

[Sha22a] J. O. Shallit. *The Logical Approach to Automatic Sequences: Exploring Combinatorics on Words with Walnut*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2022.

[Sha22b] J. O. Shallit. Note on a Fibonacci parity sequence. *Cryptography and Communications*, July 2022. `https://arxiv.org/abs/2203.10504`.

[She22] Z. Shen. The subword complexity of polynomial subsequences of the Thue-Morse sequence. *C. R. Math. Acad. Sci. Paris*, 360:503–511, 2022.

[Shp03] I. Shparlinski. *Cryptographic Applications of Analytic Number Theory*, volume 22 of *Progress in Computer Science and Applied Logic*. Birkhäuser Verlag, Basel, 2003.

[SM93] M. Stamp and C. F. Martin. An algorithm for the k-error linear complexity of binary sequences with period 2^n. *IEEE Trans. Inform. Theory*, 39(4):1398–1401, 1993.

[Sme88] B. Smeets. Linear complexity profile of sequences over the field GF(Q). *International symposium on information theory*, 1988.

[Spi20] L. Spiegelhofer. The level of distribution of the Thue-Morse sequence. *Compos. Math.*, 156(12):2560–2587, 2020.

[Sto78] Kenneth B. Stolarsky. The binary digits of a power. *Proc. Amer. Math. Soc.*, 71(1):1–5, 1978.

[Sto12] T. Stoll. The sum of digits of polynomial values in arithmetic progressions. *Funct. Approx. Comment. Math.*, 47(part 2):233–239, 2012.

[Sto13] T. Stoll. Combinatorial constructions for the Zeckendorf sum of digits of polynomial values. *Ramanujan J.*, 32(2):227–243, 2013.

[Sto16] T. Stoll. On digital blocks of polynomial values and extractions in the Rudin-Shapiro sequence. *RAIRO Theor. Inform. Appl.*, 50(1):93–99, 2016.

[SW19a] Z. Sun and A. Winterhof. On the maximum order complexity of subsequences of the Thue-Morse and Rudin-Shapiro sequence along squares. *Int. J. Comput. Math. Comput. Syst. Theory*, 4(1):30–36, 2019.

[SW19b] Z. Sun and A. Winterhof. On the maximum order complexity of the Thue-Morse and Rudin-Shapiro sequence. *Unif. Distrib. Theory*, 14(2):33–42, 2019.

[SZ54] R. Salem and A. Zygmund. Some properties of trigonometric series whose terms have random signs. *Acta Mathematica*, 91(1):245–301, December 1954.

[Sza02] L. Szalay. The equations $2^n \pm 2^m \pm 2^l = z^2$. *Indag. Math. (N.S.)*, 13(1):131–142, 2002.

[Sza20] L. Szalay. Computational algorithm for solving the diophantine equations $2^n \pm \alpha \cdot 2^m + \alpha^2 = x^2$. *Houston J. Math.*, 46(2):295–306, 2020.

[SZL20] Z. Sun, X. Zeng, and D. Lin. On the Nth maximum order complexity and the expansion complexity of a Rudin-Shapiro-like sequence. *Cryptogr. Commun.*, 12(3):415–426, 2020.

[Thu06] A. Thue. *Über unendliche Zeichenreihen*. Skrifter udg. af Videnskabs-Selskabet i Christiania : 1. Math.-Naturv. Klasse. Dybwad [in Komm.], 1906.

[Thu12] A. Thue. *Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen*. Skrifter udgivne af Videnskabsselskabet i Christiania. 1,Math.Nat.wiss.Kl.1912,1. Jacob Dybwad, 1912.

[Tit86] E. C. Titchmarsh. *The Theory of the Riemann Zeta-function*. The Clarendon Press, Oxford University Press, New York, second edition, 1986. Edited and with a preface by D. R. Heath-Brown.

[TW07] A. Topuzoğlu and A. Winterhof. Pseudorandom sequences. In *Topics in geometry, coding theory and cryptography*, volume 6 of *Algebr. Appl.*, pages 135–166. Springer, Dordrecht, 2007.

[TWN20] TWN. LFSR and Berlekamp-Massey algorithm, 2020. `https://github.com/thewhiteninja/lfsr-berlekamp-massey`.

[vN51] J. v. Neumann. Various techniques used in connection with random digits. In A.S. Householder, G.E. Forsythe, and H.H. Germond, editors, *Monte Carlo Method*, pages 36–38. National Bureau of Standards Applied Mathematics Series, 12, Washington, D.C.: U.S. Government Printing Office, 1951.

[Wan99] Y. Wang. Linear complexity versus pseudorandomness: on Beth and Dai's result. In *Advances in cryptology—ASIACRYPT'99 (Singapore)*, volume 1716 of *Lecture Notes in Comput. Sci.*, pages 288–298. Springer, Berlin, 1999.

[Zec72] E. Zeckendorf. Représentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres de Lucas. *Bull. Soc. Roy. Sci. Liège*, 41:179–182, 1972.