Numéro National de Thèse : 2017LYSEN033

Thèse de Doctorat de l'Université de Lyon

opérée par,

l'École Normale Supérieure de Lyon

École doctorale InfoMaths N° 512
École doctorale en Informatique et Mathématiques de Lyon

Spécialité de doctorat : Informatique

Soutenue publiquement le 29 juin 2017 par,

Matthieu Rosenfeld

Avoidability of Abelian Repetitions in Words

Évitabilité des répétitions abéliennes dans les mots

Devant le jury composé de :

Golnaz	Badkobeh	Early Career Fellow, University of Warwick	*Examinatrice*
Valérie	Berthé	Directrice de recherche du CNRS, IRIF, Paris	*Examinatrice*
Julien	Cassaigne	Chargé de recherche du CNRS, I2M, Marseille	*Examinateur*
Maxime	Crochemore	Professeur des Universités, LIGM, Marne-la-Vallée	*Rapporteur*
Michaël	Rao	Chargé de recherche du CNRS, LIP, Lyon	*Directeur de thèse*
Gwenaël	Richomme	Professeur des Universités, LIRMM, Montpellier	*Rapporteur*
Laurent	Vuillon	Professeur des Universités, LAMA, Chambéry	*Rapporteur*

Résumé

Dans ce document, nous étudions l'évitabilité de différentes formes de répétitions dans les mots. En particulier 3 des 6 chapitres sont dédiés aux répétitions abéliennes en lien notamment avec deux questions d'Erdős de 1957 et 1961. Nous commençons par montrer qu'il existe un algorithme décidant, sous certaines conditions, si un mot morphique évite des puissances abéliennes. Cet algorithme élargit la classe sur laquelle les précédents algorithmes pouvaient décider. Une généralisation de cet algorithme nous permet de montrer que les longs carrés abéliens sont évitables sur l'alphabet ternaire et que les carrés additifs sont évitables sur $\mathbb{Z}^2$. Le premier résultat répond à une question ouverte de Mäkelä datant de 2003 alors que le deuxième rappelle la question ouverte de 1994 concernant l'évitabilité des carrés additifs sur $\mathbb{Z}$.

Une autre généralisation de notre algorithme permet d'étudier l'évitabilité des motifs au sens abélien. Nous montrons que les motifs binaires de longueur supérieure à 14 sont évitables sur l'alphabet binaire, améliorant la précédente borne de 118.

Nous donnons des conditions suffisantes pour qu'un morphisme soit sans longues puissances $n^{\text{ème}}$ k-abéliennes. Ce résultat nous permet de calculer, pour tout $k \geq 3$, le nombre minimum de carrés k-abéliens qu'un mot binaire infini doit contenir en facteur. Il permet aussi de montrer que les longs carrés 2-abéliens sont évitables sur l'alphabet binaire et qu'il existe un mot ternaire qui ne contient qu'un seul carré 2-abélien en tant que facteur.

Enfin, nous proposons une classification complète des formules binaires en fonction de la taille d'alphabet qu'il faut pour les éviter et du taux de croissance (exponentiel ou polynomial) du langage les évitant.

Abstract

In this document, we study the avoidability of different kind of repetitions in words. We first show that under some conditions one can decide whether a morphic word avoids abelian n-th powers. This algorithm can decide over a wider class of morphism than the previous algorithms. We generalize this algorithm and use it to show that long abelian squares are avoidable over the ternary alphabet and that additive squares are avoidable over $\mathbb{Z}^2$. The first result answers a weak version of a question formulated by Mäkelä in 2003 and the second one is related to an open question from 1994 about the avoidability of additive squares over $\mathbb{Z}$.

Another generalization of this algorithm can be used to study avoidability of patterns in the abelian sense. In particular, we show that binary patterns of length more than 14 are avoidable over the binary alphabet in the abelian sense. This improves considerably the previous bound of 118.

We give sufficient conditions for a morphism to be long k-abelian n-th power-free. This result allows us to compute for every $k \geq 3$ the number of different k-abelian squares that a binary word must contain. We prove that long 2-abelian squares are avoidable over the binary alphabet and that over the ternary alphabet there exists a word that contains only one 2-abelian square.

We also give a complete classification of binary formulas based on the size of the smallest alphabet over which they are avoidable and on the growth (exponential or polynomial) of the associated language.

Contents

List of Figures

Introduction

Les travaux de Thue sur l'évitabilité des répétitions dans les mots marquent le début de l'étude de la combinatoire des mots [59, 60]. Un carré (resp. cube) est un mot w qui peut s'écrire $w = uu$ (resp. $w = uuu$) avec u un mot non vide. Tout mot binaire de longueur au moins 4 contient un carré en facteur. Thue montra qu'il existe un mot ternaire infini qui évite les carrés en facteur et un mot binaire infini qui évite les cubes (voir [7] pour une traduction en anglais des deux articles de Thue).

Le fameux mot de Thue-Morse (parfois appelé suite de Prouhet-Thue-Morse) est un exemple de mot binaire évitant les cubes, mais est aussi connu pour de nombreuses autres applications allant de la géométrie différentielle à la théorie des nombres (voir [1] pour certaines occurrences de ce mot dans la littérature). Ce mot peut être obtenu en itérant le morphisme h à partir de 0:

$$h : \begin{cases} 0 & \mapsto 01 \\ 1 & \mapsto 10. \end{cases}$$

On peut montrer que si un mot binaire w évite les cubes alors $h(w)$ évite les cubes, et on obtient donc le résultat de Thue par induction.

De manière similaire, on peut obtenir un mot sans carrés en itérant ce morphisme:

$$g : \begin{cases} 0 & \mapsto 012 \\ 1 & \mapsto 02 \\ 2 & \mapsto 1. \end{cases}$$

Depuis les mots sans carrés ont été bien étudiés. On sait par exemple que le nombre de mots sans carrés sur l'alphabet ternaire est exponentiel en fonction de la longueur des mots [9]. Le problème consistant à décider si un morphisme préserve l'évitement des carrés (l'image d'un mot sans carrés est sans carrés) a aussi reçu une certaine attention. Par exemple, Crochemore a montré que, pour tout morphisme h, il existe une constante C_h, qui ne

dépend que des tailles des images des lettres, telle que h préserve l'évitement des carrés si et seulement si il préserve l'évitement des carrés pour les mots de longueur au plus C_h [17].

De nombreuses généralisations de cette question ont été étudiées. On peut citer la conjecture de Dejean [21] qui concerne les répétitions fractionnaires et dont la résolution tient sur de nombreux articles de différents auteurs [12, 41, 47, 48, 18, 52].

L'étude des répétitions abéliennes a été motivée par des questions d'Erdős. Un carré abélien (resp. cube abélien) est un mot uv tel que v est obtenu par permutation des lettres de u (resp. uvw avec v et w obtenus en permutant les lettres de u). Erdős a demandé si les carrés abéliens sont évitables sur 4 lettres [24, 25]. Après des résultats intermédiaires (alphabet de taille 25 par Evdokimov [26] et de taille 5 par Pleasant [50]), Keränen donna une réponse positive à la question d'Erdős en donnant un morphisme 85-uniforme (trouvé par ordinateur) dont le point fixe évite les carrés abéliens [34]. De plus, Dekking a montré qu'on peut éviter les cubes abéliens sur l'alphabet ternaire et les puissances 4-ème abéliennes sur l'alphabet binaire [22].

Les preuves d'existence d'un mot évitant certains types de répétitions sont souvent basées sur des constructions explicites utilisant des mots morphiques. Il existe donc de nombreuses méthodes de preuve, plus ou moins génériques, pour montrer qu'un mot évite certains types de puissances. En particulier, Dekking a fourni des conditions suffisantes pour qu'un morphisme envoie tout mot sans puissances abéliennes sur un mot sans puissances abéliennes [22] et Carpi a donné des conditions plus fortes [10]. Il a été conjecturé que les conditions de Carpi caractérisent les morphismes qui préservent l'évitement des puissances k-ème abéliennes. Mais nous connaissons de nombreux morphismes qui n'ont pas cette propriété, mais dont le point fixe évite les puissances k-ème abéliennes. C'est pourquoi un algorithme qui décide de cette dernière propriété est un outil intéressant pour étudier l'évitabilité des répétitions abéliennes. Currie et Rampersad ont donné un algorithme qui peut décider de cette propriété sous de fortes conditions sur le morphisme [19]. Dans le Chapitre 1, nous généralisons cet algorithme pour pouvoir décider sur une classe plus large de mots morphiques. Cet algorithme s'avère être crucial pour les résultats des trois chapitres qui suivent.

Motivé par le fait que les carrés ne sont pas évitables sur l'alphabet ternaire, Erdős souleva la question de l'évitabilité des longs carrés [25]. Entringer, Jackson et Schatz ont donné une réponse positive à cette question [23]. Dans le même article, ils ont montré qu'au contraire les longs carrés

abéliens ne sont pas évitables sur l'alphabet binaire. Fraenkel et Simpson ont amélioré le premier résultat en montrant qu'il existe un mot binaire infini dont les seuls carrés sont 0^2, 1^2, $(01)^2$ [27]. La construction de Fraenkel et Simpson a depuis été simplifiée et il y a maintenant des preuves de ce résultat basées sur des mots morphiques [28, 51], la plus simple étant la construction donnée par Badkobeh [3].

Mäkelä posa les deux questions suivantes, qui sont assez naturelles après les questions d'Erdős:

Problem 3.1 (Mäkelä (see [35]))**.** *Les carrés abéliens de la forme uv où $|u| \geq 2$ sont-ils évitables sur l'alphabet ternaire ?*

Problem 3.2 (Mäkelä (see [35]))**.** *Les cubes abéliens de la forme uvw où $|u| \geq 2$ sont-ils évitables sur l'alphabet binaire ?*

Dans le Chapitre 3, nous répondons négativement à la seconde question et positivement à une version faible de la première.

Récemment Karhumäki *et al.* ont introduit l'équivalence k-abélienne, une généralisation de l'équivalence abélienne [33]. Cela induit naturellement des notions associées comme la complexité k-abélienne, les classes d'équivalence k-abéliennes ou les répétitions k-abéliennes. En particulier, ils ont montré quelques premiers résultats et posé quelques questions à propos de l'évitabilité des puissances k-abéliennes [30, 31, 33]. Un des premiers résultats concernant l'évitabilité des puissances k-abéliennes est que les carrés 2-abéliens ne sont pas évitables sur l'alphabet ternaire [31]. Après des résultats intermédiaires [29, 39, 40], il a finalement été prouvé qu'on peut éviter les carrés 3-abéliens sur l'alphabet ternaire et les cubes 2-abéliens sur l'alphabet binaire [53]. En considérant les questions de Mäkelä, il semble naturel de demander si les longs carrés 2-abéliens sont évitables sur l'alphabet ternaire ou si les longs carrés k-abéliens sont évitables sur l'alphabet ternaire pour tout k. Nous donnons une réponse positive à ces deux questions au Chapitre 4.

La notion de puissance modulo Φ, introduite par Justin, généralise la notion de puissance et de puissance abélienne [32]. Une question particulièrement intéressante demande s'il existe un alphabet fini $\mathcal{A} \subseteq \mathbb{Z}$ et un mot infini sur $\mathcal{A}$ qui ne contient pas deux facteurs consécutifs de même somme et même longueur [49]. Récemment Cassaigne *et al.* ont montré qu'on peut le faire pour trois facteurs consécutifs [15]. Dans le Chapitre 2, nous considérons l'autre affaiblissement et nous montrons que la réponse est oui si la question est posée avec la contrainte $\mathcal{A} \subseteq \mathbb{Z}^2$.

De nombreux auteurs ont étudié l'évitabilité des motifs, une généralisation des puissances [8, 38, 44, 45, 58, 61]. En fait, Thue lui même était intéressé par savoir si pour tout mot fini u et mot infini w, il existait un morphisme non-écrasant h tel que $h(u)$ est un facteur de w. Dans notre terminologie, il demandait s'il existe des motifs évitables, et il montra que AA et AAA sont évitables. Depuis nous avons une classification complète des motifs binaires selon la taille d'alphabet nécessaire pour les éviter et selon qu'ils sont évitables par un mot purement morphique ou non sur ces alphabets [13, 57]. Cette notion s'étend naturellement a une notion de motif modulo Φ ou de motif au sens abélien. Un motif P est un mot sur un alphabet Δ, et on dit qu'un mot w réalise P (resp. réalise P au sens abélien) si w peut s'écrire $w = w_1 w_2 \dots w_{|P|}$ tel qu'aucun des w_i n'est vide et que pour tout i et j, $P_i = P_j$ implique $w_i = w_j$ (resp. w_j est une permutation de w_i). Currie et Visentin ont montré que les motifs binaires de longueur plus que 118 sont évitables au sens abélien sur l'alphabet binaire [20]. Nous étudions l'évitabilité des motifs binaires au sens abélien dans le Chapitre 6 et nous montrons que les motifs binaires de longueur plus que 14 sont évitables au sens abélien sur l'alphabet binaire.

Résumé des chapitres

Dans le Chapitre 1, nous montrons que, sous certaines conditions, nous pouvons décider si un mot purement morphique évite les puissances n-ème abéliennes. L'algorithme de décision est plus général que l'algorithme de Currie et Rampersad ou que les conditions de Dekking ou de Carpi. En particulier, nous pouvons l'utiliser pour montrer que le morphisme h_6 défini en section 1.4 produit un mot sans carrés abéliens. Ce morphisme est particulièrement intéressant car seulement 3 de ses valeurs propres sont de norme supérieure à 1 et nous avons besoin d'un tel morphisme pour les résultats des Chapitres 2 et 3. Nous introduisons aussi de nombreuses notations au début de ce chapitre.

Dans le Chapitre 2, nous généralisons l'algorithme du Chapitre 1 en un algorithme permettant de décider sous certaines conditions de l'évitement des puissances n-ème additives dans les mots morphiques. Nous utilisons cet algorithme pour montrer qu'il existe un mot infini sur un sous-ensemble fini de $\mathbb{Z}^2$ qui ne contient pas deux facteurs consécutifs de même longueur et même somme. La construction est basée sur le morphisme h_6 et un deuxième

morphisme qui envoie les lettres sur $\mathbb{Z}^2$. Cet algorithme peut aussi être utilisé pour vérifier le résultat de Cassaigne *et al.* à propos de l'évitabilité des cubes sur un ensemble fini de $\mathbb{Z}$ et nous montrons que les cubes additifs sont évitables sur différents sous-ensembles de $\mathbb{Z}$.

Dans le Chapitre 3, nous commençons par montrer que la réponse à la question de Mäkelä 3.2 est négative, c'est-à-dire que chaque mot binaire infini contient au moins un cube abélien de période supérieure à 2. Cela nous mène à considérer une version faible de la question de Mäkelä où 2 est remplacé par "un entier p". Nous montrons ensuite qu'il existe un mot évitant les carrés abéliens de période supérieur à 5 ce qui répond à la version faible de la Question 3.1. La preuve est basée sur une extension de l'algorithme du Chapitre 1 qui permet de décider sous certaines conditions si un mot morphique évite les puissances n-ème abéliennes de période plus que p. Nous montrons aussi que l'évitabilité des longues puissances abéliennes et l'évitabilité des puissances additives sont liées.

Dans le Chapitre 4, nous commençons par donner un ensemble de conditions suffisantes pour qu'un morphisme h ait la propriété suivante: pour tout mot w évitant les puissances n-ème abéliennes, $h(w)$ évite les longues puissances n-ème k-abéliennes. Puis nous utilisons ce résultat pour montrer qu'il existe un mot ternaire infini qui ne contient qu'un seul carré 2-abélien [31]. Nous considérons ensuite la fonction $g : \mathbb{N} \mapsto \mathbb{N} \cup \{\infty\}$ tel que pour tout k, $g(k)$ est le nombre minimum de différents carrés k-abéliens qu'un mot binaire infini doit contenir. D'après le résultat de Fraenkel et Simpson, nous savons que pour tout k, $g(k) \geq 3$ [27], et d'après le résultat d'Entringer, Jackson et Schatz nous savons que $g(1) = \infty$ [23]. Nous montrons que $g(3) = g(4) = 4$ et que $g(k) = 3$ pour $k \geq 5$. Puis, en utilisant une construction plus compliquée basée sur la réponse à la question de Mäkelä, nous montrons que $5 \leq g(2) \leq 734$.

Dans le chapitre 5, nous étudions l'évitabilité des formules, une généralisation des motifs introduite par Cassaigne [14]. En particulier, nous donnons une classification complète des formules binaires en fonction de la taille du plus petit alphabet permettant de les éviter et du taux de croissance du langage associé.

Dans le Chapitre 6, nous généralisons l'algorithme du Chapitre 1 pour pouvoir décider de l'évitement des motifs au sens abélien, et nous utilisons cela pour montrer que les motifs binaires de longueur supérieure à 14 sont évitables au sens abélien sur l'alphabet binaire.

Introduction

The work of Thue on avoidability of repetitions in words initiated the study of combinatorics on words [59, 60]. A square (resp. cube) is a word w that can be written $w = uu$ (resp. $w = uuu$) for some non-empty word u. Every word of length at least 4 over the binary alphabet contains a square. Thue showed, that there exists an infinite word over the ternary alphabet that avoids squares.

He also showed that there exists an infinite binary word that avoids cubes (see [7] for a translation of the two corresponding papers of Thue). The so-called Thue-Morse sequence is an example of binary cube-free word and has many other applications ranging from differential geometry to number theory (see [1] for some of the occurrences of this sequence in the literature). The Thue-Morse sequence also has different equivalent constructions. Thue gave a construction based on a simple morphism:

$$h : \begin{cases} 0 & \mapsto 01 \\ 1 & \mapsto 10. \end{cases}$$

The sequence can be obtained by iterating h over 0. One can show that for any binary word w, if w avoids cubes, then $h(w)$ avoids cubes. Since 0 avoids cubes, it is clear by induction that for any n, $h^n(0)$ avoids cubes. Since the Thue-Morse word can be obtained by iterating a morphism, we say that it is a morphic word.

We can also obtain a square free ternary word by iteration of a simple morphism:

$$g : \begin{cases} 0 & \mapsto 012 \\ 1 & \mapsto 02 \\ 2 & \mapsto 1. \end{cases}$$

Square-free words also received a lot of attention, and it was for instance showed that the number of square-free words over the binary alphabet grows

exponentially with the length of the word [9]. Several people also investigated the problem of deciding whether a given morphism preserves square-freeness. For instance, Crochemore showed that for every morphism h there is a constant C_h that only depends on the length of the images of the letters by h, such that the morphism h preserves square-freeness if and only if it preserves square freeness for words of length less than C_h [17].

Many different generalizations of this results and questions were studied. The ones that received the more attention are probably fractional repetitions, abelian repetitions and patterns. A word $u^n v$ is a $n + \frac{|v|}{|u|}$ power if v is a prefix of u. The famous Dejean conjecture describes for any given alphabet $\mathcal{A}$ the infimum of the x such that x-powers are avoidable over $\mathcal{A}$ [21]. After the work of many different authors [12, 41, 47, 48] the last cases of this conjecture were eventually solved [18, 52].

The study of abelian repetitions was motivated by Erdős. An abelian square (resp. cube) is a word of the form uv where v is a permutation of the letters of u (resp. uvw where v and w are permutations of u). Erdős asked whether abelian squares are avoidable over 4 letters [24, 25]. After some intermediary results (alphabet of size 25 by Evdokimov [26] and size 5 by Pleasant [50]), Keränen answered positively Erdős's question by giving a 85-uniform morphism (found with the assistance of a computer) whose fixed-point is abelian square free [34]. Moreover, Dekking showed that it is possible to avoid abelian cubes on a ternary alphabet and abelian 4-th powers over a binary alphabet [22].

In the literature a common way to show the existence of an infinite word that avoids some kind of repetition is to give an explicit construction based on a morphic word. Because of the interest for this question it is possible to find in the literature different generic ways to show that a pure morphic word avoids abelian k-th powers. In particular, Dekking gave generic sufficient conditions for a morphism to preserve abelian k-th power freeness [22] and Carpi gave stronger conditions [10]. The set of conditions from Carpi is conjectured to be a characterization of morphisms that preserve abelian k-th power freeness. We know many morphisms that do not necessarily preserve abelian k-th power freeness, but that produce abelian k-th power free word when iteratively applied. An algorithm deciding if a morphic word avoids abelian powers is an interesting tool to study avoidability of abelian powers. Currie and Rampersad gave an algorithm that can decide under some conditions if the word generated by iterating a given morphism avoids abelian k-th powers [19]. In Chapter 1, we generalize this algorithm in order to be

able to decide on a larger class of morphic words. We needed to be able to show this property for morphic words from this new class for the results of the following sections.

Motivated by the fact that squares are not avoidable over the binary alphabet, Erdős raised the question of the avoidability of long squares [25]. Entringer, Jackson and Schatz gave a positive answer to this question [23]. They showed in the same paper that long abelian squares are not avoidable over the binary alphabet. Fraenkel and Simpson improved that result and showed that there is an infinite binary word containing only the squares 0^2, 1^2, $(01)^2$ [27]. The construction of Fraenkel and Simpson was simplified and there are proofs of this result based on morphic words [28, 51] amongst which the simplest was given by Badkobeh [3].

Mäkelä asked the two following questions, which are rather natural to ask when looking at the two previous questions from Erdős:

Problem 3.1 (Mäkelä (see [35]))**.** *Can you avoid abelian squares of the form uv where $|u| \geq 2$ over three letters ? - Computer experiments show that you can avoid these patterns at least in words of length 450.*

He also asked the following similar question:

Problem 3.2 (Mäkelä (see [35]))**.** *Can you avoid abelian cubes of the form uvw where $|u| \geq 2$, over two letters ? - You can do this at least for words of length 250.*

In Chapter 3, we give a negative answer to the second question and a positive answer to a weak version of the first question.

Recently Karhumäki *et al.* introduced k-abelian equivalence as a generalization of abelian equivalence [33]. It induces different interesting notions like k-abelian complexity, k-abelian equivalence classes or k-abelian repetitions. In particular, they showed some results and asked some questions about the avoidability of k-abelian powers [30, 31, 33]. One of the first results concerning the avoidability of k-abelian powers was that 2-abelian squares are not avoidable over the ternary alphabet [31]. After some intermediary results [29, 39, 40], it was finally shown that one can avoid 3-abelian squares over the ternary alphabet and 2-abelian cubes over the binary alphabet [53]. Considering Mäkelä's questions it seems natural to ask in a similar fashion whether long 2-abelian squares are avoidable over the ternary alphabet and whether long k-abelian squares are avoidable over the binary alphabet for any k. We give a positive answer to both questions in Chapter 4.

The notion of power modulo Φ was introduced by Justin as a stronger generalization of powers and abelian powers [32]. One of the main questions related to this notion asks whether it is possible to build a word over a finite subset of $\mathbb{Z}$ that does not contains two consecutive factors of same sum and same size [49]. Recently Cassaigne *et al.* showed that you can do this for three consecutive blocks [15]. In Chapter 2, we consider the other weakening of the question and we show that there is an infinite word over a finite subset of $\mathbb{Z}^2$ that does not contain two consecutive factors of same size and same sum.

The avoidability of patterns, a generalization powers, also raised a lot of interest [8, 38, 44, 45, 58, 61]. In fact, Thue himself was interested in knowing whether for every word u and infinite word w there is a non-erasing morphism h such that $h(u)$ is a factor of w. In modern terminology he was asking whether there exist avoidable patterns, and he showed that AA and AAA are avoidable patterns. In particular, we have a precise classification of binary patterns depending on the smallest alphabet on which they are avoidable and on whether they are avoidable by a pure morphic word on those alphabets [13, 57]. This notion extend naturally to a notion of pattern modulo Φ or pattern in the abelian sense. A pattern P is a word over an alphabet Δ, and we say that a word w realizes P (resp. realizes P in the abelian sense) if w can be written $w = w_1 w_2 \dots w_{|P|}$ such that none of the w_i is empty and for all i and j such that $P_i = P_j$, $w_i = w_j$ (resp. w_i is a permutation of w_j). Currie and Visentin showed that binary patterns of length greater than 118 are avoidable in the abelian sense over the binary alphabet [20]. We study avoidability of binary pattern in the abelian sense in Chapter 6 and we show that binary patterns of length greater than 14 are avoidable in the abelian sense over the binary alphabet.

Organization of the manuscript

In Chapter 1, we show that under some mild conditions one can decide whether a pure morphic word avoids abelian n-th powers. The decision procedure is more general than the algorithm from Currie and Rampersad or the conditions from Dekking or Carpi. In particular, we can use it to show that the morphism h_6, defined in Section 1.4, generates an abelian square-free word. This morphism is particularly interesting because it has only 3 eigenvalues greater than 1 and we needed such a morphism for the results

from Chapter 2 and 3. We also introduce many notations at the beginning of this chapter.

In Chapter 2, we generalize the algorithm from Chapter 1 and we get an algorithm deciding additive n-th power freeness for morphic word under some conditions. We use this algorithm to show that there is an infinite word over a finite subset of $\mathbb{Z}^2$ that does not contain two consecutive factors of same size and same sum. The construction is based on the morphism h_6 and a second morphism that maps letters to $\mathbb{Z}^2$. The algorithm can also be used to check the result from Cassaigne *et al.* about avoidability of cubes on a finite subset of $\mathbb{Z}$ and we show that we can avoid additive cubes over different subset of $\mathbb{Z}$.

In Chapter 3, we first show that the answer to Problem 3.2 is negative, that is every infinite binary word contains at least one abelian cube of period more than 2. It leads us to consider weaker versions of Mäkelä's question where 2 is replaced by any integer p. We then show that there is a word avoiding abelian squares of period more than 5 which answers the weak version of the Question 3.1. The proof is based on an extension of the algorithm from Chapter 1 that allows us to decide under some conditions whether a morphic word avoids abelian n-th powers of period more than p. We also explain that we can also deduce some results about avoidability of additive powers from the answer to Mäkelä's questions.

In Chapter 4, we first give a set of sufficient conditions for a morphism h to have the following property: for every word w avoiding abelian n-th powers, $h(w)$ avoids long k abelian n-th powers. Then we use this result to show that there is an infinite ternary word whose only 2-abelian square is 22. This last result is optimal since 2-abelian squares are not avoidable over the ternary alphabet [31]. Then we consider the function $g : \mathbb{N} \mapsto \mathbb{N} \cup \{\infty\}$ such that for all k, $g(k)$ is the minimum number of different k-abelian square that an infinite binary word must contain. From the result of Fraenkel and Simpson, we know that for all k, $g(k) \geq 3$ [27], and from the result of Entringer, Jackson and Schatz we know that $g(1) = \infty$ [23]. We first show that $g(3) = g(4) = 4$ and that $g(k) = 3$ for every $k \geq 5$. Then using a more complicated construction based on the answer to Mäkelä's question we show that $5 \leq g(2) \leq 734$.

In Chapter 5, we study the avoidability of formulas, a generalization of patterns introduced by Cassaigne [14]. In particular we give a complete classification of binary formulas based on the smallest alphabet over which they are avoidable and the growth of the associated language.

In Chapter 6, we generalize the algorithm from Chapter 1 so that we can decide pattern freeness in the abelian sense, and we use that to show that binary patterns of length greater than 14 are avoidable in the abelian sense over the binary alphabet.

Chapter 1

Deciding Whether a Morphic Word is Abelian k-th Power Free

One of the most studied generalizations of repetition is the notion of abelian repetition. We say that two words u and v are *abelian equivalent*, denoted by $u \approx_a v$ if they are permutations of the letters of each other. Clearly $\approx_a$ is an equivalence relation over $\mathcal{A}^*$. A word w is an *abelian n-th power* if there are $u_1, \ldots, u_n \in \mathcal{A}^+$ such that $w = u_1 u_2 \ldots u_n$ and for all $i, j \in [1, n]$, $u_i \approx_a u_j$. Its *period* is $|u_1|$. An *abelian square* (resp. *cube*) is an abelian 2-nd power (resp. abelian 3-rd power). A word is *abelian k-th power-free*, or *avoids abelian k-th powers*, if none of its non-empty factor is an abelian k-th power.

The avoidability of abelian powers has been studied since a question from Erdős in 1957 [24, 25]. He asked whether it is possible to avoid abelian squares in an infinite word over an alphabet of size 4. We give in Figure 1.1 the exhaustive search that shows that abelian squares are not avoidable over the ternary alphabet.

After some intermediary results (alphabet of size 25 by Evdokimov [26] and size 5 by Pleasant [50]), Keränen answered positively Erdős's question:

Theorem 1.1 (Keränen, [34])**.** *Let σ_4 be the morphism such that*

$$\sigma_4 : \begin{cases} a \to abcacdcbcdcadcdbdabacabadbabcbdbcbacbcdcacbabdabacadcbcdcacdbcbacbcdcacdcbdcdadbdcbca \\ b \to bcdbdadcdadbadacabcbdbcbacbcdcacdcbdcdadbdcbcabcbdbadcdadbdacdcbdcdadbdadcadabacadcdb \\ c \to cdacabadabacbabdbcdcacdcbdcdadbdadcadabacadcdbcdcacbadabacabdadcadabacabadbabcbdbadac \\ d \to dabdbcbabcbdcbcacdadbdadcadabacabadbabcbdbadacdadbdcbabcbdbcabadbabcbdbcbacbcdcacbabd. \end{cases}$$

Then any fixed point of σ_4 is abelian square-free.

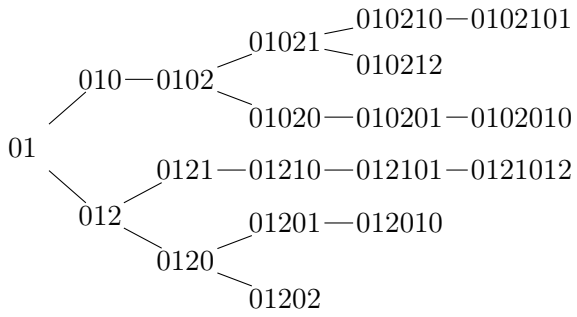

Figure 1.1 – Tree of words avoiding abelian squares over a ternary alphabet (starting w.l.o.g. by 01).

Moreover, Dekking showed that it is possible to avoid abelian cubes on a ternary alphabet and abelian 4-th powers over a binary alphabet [22].

Theorem 1.2 (Dekking, [22])**.** *Fixed points of the following morphism are abelian cube-free:*

$$\sigma_3 : \begin{cases} a \to & aabc \\ b \to & bbc \\ c \to & acc. \end{cases}$$

Theorem 1.3 (Dekking, [22])**.** *The fixed point of the following morphism is abelian 4-th power-free:*

$$\sigma_2 : \begin{cases} a \to & abb \\ b \to & aaab. \end{cases}$$

In the literature a common way to show the existence of an infinite word that avoids some kind of repetition is to give an explicit construction based on a morphic word. An algorithm deciding if a morphic word avoids abelian powers is an interesting tool to study avoidability of abelian powers. In fact, we see in Chapter 2, Chapter 3 and Chapter 4 that it also helps to study the avoidability of different generalizations of abelian powers.

Dekking gave sufficient conditions for a morphism to be *abelian k-th power-free*, that is the image of any abelian k-th power-free word by the given morphism is also abelian k-th power-free [22]. Clearly, if a morphism is abelian k-th power-free then any word generated by applying iteratively this morphism on a letter is abelian k-th power-free. These conditions are used to show Theorem 1.2 and Theorem 1.3, but are not strong enough to show Theorem 1.1.

Carpi gave stronger sufficient conditions for a morphism to be abelian k-th power-free [10]. The set of conditions is conjectured to be a characterization of abelian k-th power-free morphisms, and can be used to show Theorem 1.1. Nonetheless, there are many morphisms that are not abelian k-th power-free, but that generate infinite abelian k-th power-free words. For instance, the infinite word generated by the morphism $h_4 : a \mapsto ac, b \mapsto dc, c \mapsto b, d \mapsto ab$ is abelian cube-free [15], but the image of dcc contains the abelian cube bbb.

Currie and Rampersad gave an algorithm that can decide on a certain class of morphisms if the generated pure morphic words are abelian k-th power-free [19]. Their algorithm can also be used to show Theorem 1.1, but cannot be applied to the morphism h_4. In this Chapter, we will generalize this algorithm and ideas of [15] to give an algorithm that can decide for a wider class of morphisms. In particular, this algorithm can be used to show that h_4 generates abelian cube-free words or that h_6 (defined in Section 1.4) generates abelian square-free word. The fact that our algorithm can decide for morphisms with small eigenvalues is important because we need such morphisms for the results of Chapter 2 and Chapter 3 where we generalize this algorithm to be able to decide other avoidability properties of morphic words.

Our algorithm is based on the notion of template that was introduced by Currie and Rampersad [19], and we need to define this notion and to recall some general results and notations from linear algebra before we give the exact statement of the result and the proof. The results presented in this Chapter are joint work with Michaël Rao and are part of the article [54].

1.1 Definitions

In this section, we recall some classical definitions about words. Most of the terminology and notations come from Lothaire [37].

A finite *alphabet* $\mathcal{A}$ is a finite set whose elements are called *letters*. The *free monoid* generated by $\mathcal{A}$, denoted by $\mathcal{A}^*$, is the set of all finite sequences of elements from $\mathcal{A}$, including the empty sequence, equipped with the *concatenation operation*. The elements of $\mathcal{A}^*$ are called *words* and the *empty word* is denoted by ε. The set of non-empty words is denoted by $\mathcal{A}^+ = \mathcal{A}^* \setminus \varepsilon$. *Infinite words* are infinite sequences over an alphabet.

We denote the concatenation of two words $u \in \mathcal{A}^*$ and $v \in \mathcal{A}^*$ by $u.v$ or uv. For instance, if $\mathcal{A} = \{a, b\}$, $u = abba$ is a word over $\mathcal{A}$ and the

concatenation of u with aba is $u.aba = abbaaba$.

A *morphism* from a monoid $(M, .)$ to a monoid $(M', +)$ is a function $f : M \mapsto M'$ such that for all $u, v \in M$, $f(u.v) = f(u) + f(v)$ and $f(\varepsilon_M) = \varepsilon_{M'}$, where ε_M and $\varepsilon_{M'}$ are the respective neutral elements of M and M'. In particular a morphism $h : \mathcal{A}^* \mapsto M$ from the free monoid $\mathcal{A}^*$ is completely characterized by the images of the letters by h and the image of ε is always $h(\varepsilon) = \varepsilon$. For instance, if $\mathcal{A} = \{a, b\}$ and $h : \mathcal{A}^* \mapsto \mathcal{A}^*$ is such that $h(a) = ab$ and $h(b) = ba$ then $h(abba) = h(a)h(b)h(b)h(a) = abbabaab$.

A morphism h is *non-erasing* if there is no letter a such that $h(a) = \varepsilon$. A morphism $h : \mathcal{A}^* \mapsto \mathcal{A}^*$ is *prolongable* at $a \in \mathcal{A}$ if $h(a) = as$ for some $s \in \mathcal{A}^+$. In this case the sequence $(h^i(a))_{i\in\mathbb{N}}$ converges toward the infinite word $w = ash(s)h^2(s)h^3(s)\ldots$ If the morphism is non-erasing, w is infinite and we say that w is a *pure morphic word generated by* h, denoted by $h^\omega(a)$. Note that every pure morphic word generated by a morphism h is a fixed point of h. A *morphic word* is the image of a pure morphic word by a second morphism.

The *length* of a word denoted by $|.|$ is the number of elements of the sequence. Note that $|.| : \mathcal{A}^* \mapsto \mathbb{N}$ is a morphism since for all $u, v \in \mathcal{A}^*$ $|uv| = |u| + |v|$. For instance, if $\mathcal{A} = \{a, b\}$, $|\varepsilon| = 0$ and $|aba| = 3$. We denote by $\mathcal{A}^n = \{w \in \mathcal{A}^* : |w| = n\}$ the set of words of length n. A morphism is said to be *uniform* if all the images of the letters by the morphism have the same length.

A word $u \in \mathcal{A}^*$ is a *factor* of $w \in \mathcal{A}^*$ if there are $p, s \in \mathcal{A}^*$ such that $w = sup$. A word u is a *suffix* (*resp. prefix*) of w if there is $v \in \mathcal{A}^*$ such that $w = vu$ (*resp.* $w = uv$). Let $\mathrm{Suff}(w)$ (resp. $\mathrm{Pref}(w)$, $\mathrm{Fact}(w)$) be the set of suffixes (resp. prefixes, factors) of w. For any morphism h, let $\mathrm{Suff}(h) = \cup_{a\in\mathcal{A}} \mathrm{Suff}(h(a))$, $\mathrm{Pref}(h) = \cup_{a\in\mathcal{A}} \mathrm{Pref}(h(a))$ and $\mathrm{Fact}(h) = \cup_{a\in\mathcal{A}} \mathrm{Fact}(h(a))$.

For any set of words $F \subseteq \mathcal{A}^*$, we say that a word u *avoids* F (or u is F*-free*) if no factor of u belongs to F. A *language* L is a set of words, and we say that it is *factor-closed* (*resp. prefix-closed*) if for every w in L, every factor of w is in L (*resp.* every prefix of w is in L). Note that a language L is factor-closed if and only if there exists a set of words F such that L is the set of words avoiding F.

For any letter a and word w, we denote by $|w|_a$ the number of occurrences of the letter a in w. For any alphabet $\mathcal{A}$ and word $w \in \mathcal{A}^*$ the *Parikh vector* of w over $\mathcal{A}$, denoted by $\Psi_\mathcal{A}(w)$ is the vector indexed over $\mathcal{A}$ such that for all $a \in \mathcal{A}$, $\Psi_\mathcal{A}(w)[a] = |w|_a$. When the alphabet is clear in the context we write $\Psi(w)$ to denote the Parikh vector of w. Note that two words are

abelian equivalent if and only if they have the same Parikh vector. For instance, over the alphabet $\{a, b, c\}$, $\Psi(acbaac) = \Psi(abacac) = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}$ and thus $acbaac \approx_a abacac$.

We associate to every morphism $h : \mathcal{A}^* \mapsto \mathcal{B}^*$ the matrix M_h indexed on $\mathcal{B} \times \mathcal{A}$ such that $(M_h)_{b,a} = |h(a)|_b$. By definition we have that for every $w \in \mathcal{A}^*$, $\Psi(h(w)) = M_h \Psi(w)$. If this is a square matrix, the *eigenvalues of* h are the eigenvalues of M_h.

For any morphism $h : \mathcal{A}^* \mapsto \mathcal{A}^*$, let $\mathrm{Fact}^\infty(h) = \cup_{i=1}^\infty \mathrm{Fact}(h^i)$. We say that h is *primitive* if there exists $k \in \mathbb{N}$ such that for all $a \in \mathcal{A}$, $h^k(a)$ contains all the letters of $\mathcal{A}$ (that is, M_h is primitive). If h is primitive then for any letter $a \in \mathcal{A}$, $\mathrm{Fact}^\infty(h) = \cup_{i=1}^\infty \mathrm{Fact}(h^i(a))$ and we can use that fact to show the following property:

Proposition 1.4. *Let h be a primitive morphism on $\mathcal{A}^*$ prolongable at a, then* $\mathrm{Fact}(h^\omega(a)) = \mathrm{Fact}^\infty(h)$.

Proof. Since h is prolongable at a there is, by definition, a non empty word $s \in \mathcal{A}^+$ such that $h(a) = as$ and $h^\omega(a) = h(a)h(s)h^2(s)\ldots$ Remark that for all i, $h(a)h(s)h^2(s)\ldots h^i(s) = h^{i+1}(a)$. Thus by primitivity of h, $\mathrm{Fact}(h^\omega(a)) = \cup_{i=1}^\infty \mathrm{Fact}(h^i(a)) = \mathrm{Fact}^\infty(h)$. □

In the rest of this section we recall some classical notions from linear algebra.

Jordan decomposition A *Jordan block* $J_n(\lambda)$ is a $n \times n$ matrix with $\lambda \in \mathbb{C}$ on the diagonal, 1 on top of the diagonal and 0 elsewhere.

$$J_n(\lambda) = \begin{pmatrix} \lambda & 1 & & \\ & \lambda & 1 & 0 \\ 0 & & \ddots & 1 \\ & & & \lambda \end{pmatrix}$$

We recall the following well known proposition (see [2]).

Proposition 1.5 (Jordan decomposition)**.** *For any $n \times n$ matrix M on $\mathbb{C}$, there is an invertible $n \times n$ matrix P and a $n \times n$ matrix J such that $M =$*

PJP^{-1}, and the matrix J is as follows:

$$\begin{pmatrix} J_{n_1}(\lambda_1) & & & \\ & J_{n_2}(\lambda_2) & & 0 \\ & & \ddots & \\ 0 & & & J_{n_p}(\lambda_p) \end{pmatrix}$$

where the $J_{n_i}(\lambda_i)$ are Jordan blocks on the diagonal. PJP^{-1} is a Jordan decomposition *of M.*

The λ_i, $i \in \{1, \ldots, p\}$, are the (non necessarily distinct) eigenvalues of M. The set of columns from P are *generalized eigenvectors* of M.

Note that for every $k \geq 0$, $(J_n(\lambda))^k$ is the $n \times n$ matrix M with $M_{i,j} = \binom{k}{j-i}\lambda^{k-j+i}$, with $\binom{a}{b} = 0$ if $a < b$ or $b < 0$. Thus, if $|\lambda| < 1$, $\sum_{k=0}^{\infty}(J_n(\lambda))^k$ is the matrix N where $N_{i,j} = (1-\lambda)^{i-j-1}$ if $j \geq i$, and 0 otherwise. We can easily deduce from these observations the series of k-th powers of a matrix in Jordan normal form, and its sum.

Smith decomposition The Smith decomposition is useful to solve systems of linear Diophantine equations.

Proposition 1.6 (Smith decomposition)**.** *For any matrix $M \in \mathbb{Z}^{n\times m}$, there are $U \in \mathbb{Z}^{n\times n}$, $D \in \mathbb{Z}^{n\times m}$ and $V \in \mathbb{Z}^{m\times m}$ such that:*

- *D is diagonal (*i.e. *$D_{i,j} = 0$ if $i \neq j$),*
- *U and V are unimodular (*i.e.*, their determinant is 1 or -1),*
- *$M = UDV$.*

Since U and V are unimodular, they are invertible over the integers. If one wants to find integer solutions $\mathbf{x}$ of the equation $M\mathbf{x} = \mathbf{y}$, where M is an integer matrix and $\mathbf{y}$ an integer vector, one can use the Smith decomposition UDV of M. One can suppose w.l.o.g. that $n = m$, otherwise, one can fill with zeros. Then $DV\mathbf{x} = U^{-1}\mathbf{y}$. Integer vectors in $\ker(M)$ form a lattice Λ. The set of columns i in V^{-1} such that $D_{i,i} = 0$ gives a basis of Λ. Let $\mathbf{y}' = U^{-1}\mathbf{y}$, which is also an integer vector. Finding the solution $\mathbf{x}'$ of $D\mathbf{x}' = \mathbf{y}'$ is easy, since D is diagonal. The set of solutions is non-empty if and only if for every i, ${\mathbf{y}'}_i$ is a multiple of $D_{i,i}$. One can take $\mathbf{x_0} = V^{-1}\mathbf{x}'_\mathbf{0}$ as a particular solution to $M\mathbf{x_0} = \mathbf{y}$, with $(\mathbf{x}'_\mathbf{0})_i = 0$ if $D_{i,i} = 0$, and $(\mathbf{x}'_\mathbf{0})_i = {\mathbf{y}'}_i / D_{i,i}$ otherwise. The set of solutions is given by $\mathbf{x_0} + \Lambda$.

For any vector $\mathbf{x}$ we denote by $||\mathbf{x}||$ its Euclidean norm. For any matrix complex M, let $||M||$ be its norm induced by the Euclidean norm, that is $||M|| = \sup\left\{\frac{||M\mathbf{x}||}{||\mathbf{x}||} : \mathbf{x} \neq \overrightarrow{0}\right\}$. Let M^* be the conjugate transpose of the matrix M. We will use the following classical Proposition from linear algebra (see [2]).

Proposition 1.7. *Let M be a matrix, and let μ_{min} (resp. μ_{max}) be the minimum (resp. maximum) over the eigenvalues of M^*M (which are all real and non-negative). Then for any $\mathbf{x}$:*

$$\mu_{min}||\mathbf{x}||^2 \leq ||M\mathbf{x}||^2 \leq \mu_{max}||\mathbf{x}||^2.$$

For any vector $\mathbf{x}$, we also denote by $||\mathbf{x}||_1$ its L_1 *norm*, that is the sum of the absolute value of its coordinates. The L_1 norm is usefull for us because of the following property: for any $w \in \mathcal{A}^*$, $|w| = ||\Psi(w)||_1$.

1.2 Templates

The notion of template was first introduced by Currie and Rampersad for their decision algorithm [19]. A *k-template* is a $(2k)$-tuple of the form $t = [a_1, \ldots, a_{k+1}, \mathbf{d}_1, \ldots, \mathbf{d}_{k-1}]$ where for all i, $a_i \in \mathcal{A} \cup \{\varepsilon\}$ and $\mathbf{d}_i \in \mathbb{Z}^n$. A word $w = a_1 w_1 a_2 w_2 \ldots w_k a_{k+1}$, where $w_i \in \mathcal{A}^*$, is a *realization* of (or *realizes*) the template t if for all $i \in \{1, \ldots, k-1\}$, $\Psi(w_{i+1}) - \Psi(w_i) = \mathbf{d}_i$. A template t is *realizable by h* if there is a word in $\mathrm{Fact}^\infty(h)$ which realizes t.

Using the notion of k-templates, we can give an other equivalent definition of abelian k-th powers:

Proposition 1.8. *Let $k \geq 2$ be an integer. A non-empty word is an abelian k-th power if and only if it realizes the k-template $[\varepsilon, \ldots, \varepsilon, \overrightarrow{0}, \ldots, \overrightarrow{0}]$.*

Let $t' = [a'_1, \ldots, a'_{k+1}, \mathbf{d}'_1, \ldots, \mathbf{d}'_{k-1}]$ and $t = [a_1, \ldots, a_{k+1}, \mathbf{d}_1, \ldots, \mathbf{d}_{k-1}]$ be two k-templates and h be a morphism. We say that *t' is a parent by h of t* if there are $p_1, s_1, \ldots, p_{k+1}, s_{k+1} \in \mathcal{A}^*$ such that:

— $\forall i \in \{1, \ldots, k+1\}$, $h(a'_i) = p_i a_i s_i$,

— $\forall i \in \{1, \ldots, k-1\}$, $\mathbf{d}_i = M_h \mathbf{d}'_i + \Psi(s_{i+1}p_{i+2}) - \Psi(s_i p_{i+1})$.

We denote by $\mathrm{Par}_h(t)$ the set of parents by h of t. We will show in Proposition 1.11 that for any $t' \in \mathrm{Par}_h(t)$ if t' is realized by a word w, then t is realized

by a factor of $h(w)$. In Proposition 1.12 we show that if t is realized by a long enough word from $\mathrm{Fact}^\infty(h)$ then there is a realizable template $t' \in \mathrm{Par}_h(t)$.

A template t' is an *ancestor by h* of a template t if there exists $n \geq 1$ and a sequence of templates $t = t_1, t_2, \ldots, t_n = t'$ such that for any i, t_{i+1} is a parent by h of t_i. A template t' is a *realizable ancestor by h* of a template t, if t' is an ancestor by h of t and if t' is realizable by h. For a template t, we denote by $\mathrm{Anc}_h(t)$ (resp. $\mathrm{Ranc}_h(t)$) the set of all the ancestors (resp. realizable ancestors) by h of t. We may omit "by h" when the morphism is clear in the context.

1.3 The decision algorithm

In this section, we show the following theorem.

Theorem 1.9. *For any primitive morphism h with no eigenvalue of absolute value 1 and any template t_0, it is possible to decide whether* $\mathrm{Fact}^\infty(h)$ *realizes t_0.*

Together with the Proposition 1.4, it implies the following corollary:

Corollary 1.10. *For any primitive morphism h with no eigenvalue of absolute value* 1 *it is possible to decide whether the fixed points of h are abelian k-th power-free.*

The main difference with the algorithm from Currie and Rampersad [19] is that we allow h to have eigenvalues of absolute value less than 1.

We first show that for any set S such that $\mathrm{Ranc}_h(t_0) \subseteq S \subseteq \mathrm{Anc}_h(t_0)$, $\mathrm{Fact}^\infty(h)$ realizes t_0 if and only if there is a small factor of $\mathrm{Fact}^\infty(h)$ which realizes a template in S. Then we explain how to compute such a finite set S. Since S is finite we can check for any k-template $t \in S$ whether a small factor realizes t and we can conclude.

1.3.1 Parents and pre-images

The next two lemmas tell that the realizations of the parents of a template t are nearly the pre-images by h of the realizations of h.

Lemma 1.11. *Let t' be a parent of a k-template t_0, and $w \in \mathcal{A}^*$. If w realizes t', $h(w)$ contains a factor that realizes t_0.*

Proof. Let $t_0 = [a_1, \dots, a_{k+1}, \mathbf{d}_1, \dots, \mathbf{d}_{k-1}]$ and $t' = [a'_1, \dots, a'_{k+1}, \mathbf{d'}_1, \dots, \mathbf{d'}_{k-1}]$. Since w realizes t', there are $w_1, \dots, w_k \in \mathcal{A}^*$ such that $w = a'_1 w_1 a'_2 \dots w_k a'_{k+1}$ and for all $i \in \{1, \dots, k-1\}$, $\Psi(w_{i+1}) - \Psi(w_i) = \mathbf{d}'_i$.

Since t' is a parent of t_0, there are $p_1, s_1, \dots, p_{k+1}, s_{k+1} \in \mathcal{A}^*$ such that:

— $\forall i \in \{1, \dots, k+1\}$, $h(a'_i) = p_i a_i s_i$,

— $\forall i \in \{1, \dots, k-1\}$, $\mathbf{d}_i = M_h \mathbf{d'}_i + \Psi(s_{i+1} p_{i+2}) - \Psi(s_i p_{i+1})$.

Thus $h(w) = p_1 a_1 s_1 h(w_1) p_2 a_2 s_2 h(w_2) \dots h(w_k) p_{k+1} a_{k+1} s_{k+1}$. Now let for all i, $u_i = s_i h(w_i) p_{i+1}$ then the word $u = a_1 u_1 a_2 u_2 \dots u_k a_{k+1}$ is a factor of $h(w)$. Moreover for all i,

$$\begin{aligned}
\Psi(u_{i+1}) - \Psi(u_i) &= \Psi(s_{i+1} h(w_{i+1}) p_{i+2}) - \Psi(s_i h(w_i) p_{i+1}) \\
&= \Psi(h(w_{i+1})) - \Psi(h(w_i)) + \Psi(s_{i+1} p_{i+2}) - \Psi(s_i p_{i+1}) \\
&= M_h \left(\Psi(w_{i+1}) - \Psi(w_i)\right) + \Psi(s_{i+1} p_{i+2}) - \Psi(s_i p_{i+1}) \\
&= M_h \mathbf{d}'_i + \Psi(s_{i+1} p_{i+2}) - \Psi(s_i p_{i+1}) \\
\Psi(u_{i+1}) - \Psi(u_i) &= \mathbf{d}_i.
\end{aligned}$$

Thus u realizes t_0. □

Let $\delta = \max_{a \in \mathcal{A}} |h(a)|$ and $\Delta(t) = \max_{i=1}^{k-1} ||\mathbf{d}_i||_1$, for any k-template $t = [a_1, \dots, a_{k+1}, \mathbf{d}_1, \dots, \mathbf{d}_{k-1}]$.

Lemma 1.12. *Let t be a k-template and $w \in \mathcal{A}^*$ be a word which realizes t. If $|w| > k\left(\frac{(k-1)\Delta(t)}{2} + \delta + 1\right) + 1$ then for every w' such that $w \in \mathrm{Fact}(h(w'))$ there is a parent t' of t such that a factor of w' realizes t'.*

The idea is that if the realization is long enough then the part corresponding to each vector is longer than δ. This implies that the a_i are images of different letters and we can then unfold the definitions.

Proof of Lemma 1.12. Let $t = [a_1, \dots, a_{k+1}, \mathbf{d}_1, \dots, \mathbf{d}_{k-1}]$ be a k-template and $w \in \mathrm{Fact}(h(w'))$ such that $|w| > k\left(\frac{(k-1)\Delta(t)}{2} + \delta + 1\right) + 1$ and w realizes t. Then there are $w_1, \dots, w_n \in \mathcal{A}^*$ such that $w = a_1 w_1 a_2 w_2 \dots w_k a_{k+1}$ and $\forall i \in \{1, \dots, k-1\}$, $\Psi(w_{i+1}) - \Psi(w_i) = \mathbf{d}_i$. Thus for any $i, j \in \{1, \dots, k\}$ such that $j < i$, $\Psi(w_i) = \Psi(w_j) + \sum_{m=j}^{i-1} \mathbf{d}_m$ and, by triangular inequality,

we have:

$$\begin{aligned}\big||w_i| - |w_j|\big| &= \big|\,||\Psi(w_i)||_1 - ||\Psi(w_j)||_1\big| \\ &\leq ||\Psi(w_i) - \Psi(w_j)||_1 \\ &\leq \left|\left|\sum_{m=j}^{i-1} \mathbf{d}_m\right|\right|_1 \\ &\leq \sum_{m=j}^{i-1} ||\mathbf{d}_m||_1 \\ &\leq (i-j)\Delta(t).\end{aligned}$$

Therefore for any $i, j \in \{1, \ldots, k\}$, $|w_j| \leq |i-j|\Delta(t) + |w_i|$. Combining this equality with $|w| = k+1+\sum_{m=1}^{k} |w_m|$ we deduce that for any $i \in \{1, \ldots, k\}$, $|w| \leq \sum_{m=1}^{k}(|i-m|\Delta(t) + |w_i|) + k + 1 \leq \frac{k(k-1)}{2}\Delta(t) + k|w_i| + k + 1$. Then, by hypothesis, $k\left(\frac{(k-1)\Delta(t)}{2} + |w_i| + 1\right) + 1 \geq |w| > k\left(\frac{(k-1)\Delta(t)}{2} + \delta + 1\right) + 1$, and consequently $\forall i,\ |w_i| > \delta = \max_{a\in\mathcal{A}} |h(a)|$. We also know that $w \in \text{Fact}(h(w'))$ so there are $a'_1, \ldots, a'_{k+1} \in \mathcal{A}$, $w'_1, \ldots, w'_k \in \mathcal{A}^*$, $p_1, \ldots, p_{k+1} \in \text{Pref}(h)$ and $s_1, \ldots, s_{k+1} \in \text{Suff}(h)$ such that:

— $w'' = a'_1 w'_1 a'_2 \ldots a'_k w'_k a'_{k+1}$ is a factor of w',

— $\forall i,\ h(a'_i) = p_i a_i s_i$,

— $\forall i,\ w_i = s_i h(w'_i) p_{i+1}$.

Then w'' realizes $t' = [a'_1, \ldots, a'_{k+1}, \Psi(w'_2) - \Psi(w'_1), \ldots, \Psi(w'_k) - \Psi(w'_{k-1})]$. Moreover for all i:

$$\begin{aligned}\mathbf{d}_i &= \Psi(w_{i+1}) - \Psi(w_i) \\ \mathbf{d}_i &= \Psi(s_{i+1}h(w'_{i+1})p_{i+2}) - \Psi(s_i h(w'_i)p_{i+1}) \\ \mathbf{d}_i &= M_h\Psi(w'_i) - M_h\Psi(w'_i) + \Psi(s_{i+1}p_{i+2}) - \Psi(s_i p_{i+1}) \\ \mathbf{d}_i &= M_h(\Psi(w'_i) - \Psi(w'_i)) + \Psi(s_{i+1}p_{i+2}) - \Psi(s_i p_{i+1}).\end{aligned}$$

Thus t' is a parent of t and t' is realized by w'' a factor of w'. □

A *small realization* of a k-template t is a realization w of t such that $|w| < k\left(\frac{(k-1)\Delta(t)}{2} + \delta + 1\right) + 1$. Using Lemmas 1.11 and 1.12 we can show the following proposition:

Proposition 1.13. *Let h be a primitive morphism, and t_0 a k-template. Then the following conditions are equivalent:*

1. $\mathrm{Fact}^\infty(h)$ *contains no realization* t_0,
2. $\mathrm{Fact}^\infty(h)$ *contains no small realizations of any elements of* $\mathrm{Anc}_h(t_0)$,
3. $\mathrm{Fact}^\infty(h)$ *contains no small realizations of any elements of* $\mathrm{Ranc}_h(t_0)$.

Proof. **2.** $\iff$ **3.** If a template $t \in \mathrm{Anc}_h(t_0)$ is realized then by definition $t \in \mathrm{Ranc}_h(t_0)$ so 3 $\implies$ 2. The other direction is clear from $\mathrm{Ranc}_h(t_0) \subseteq \mathrm{Anc}_h(t_0)$.

1. $\implies$ **2.** Assume that $\mathrm{Fact}^\infty(h)$ contains a small realization w of $t \in \mathrm{Anc}_h(t_0)$. By definition there are $t_n = t, t_{n-1}, t_{n-2}, \ldots, t_1 \in \mathrm{Anc}_h(t_0)$ such that for all $i \in [0, n-1]$, $t_{i+1} \in \mathrm{Par}_h(t_i)$. Now by applying inductively Lemma 1.11 we get that for all i, t_{n-i} is realized by a factor of $h^i(w) \in \mathrm{Fact}^\infty(h)$. So in particular $\mathrm{Fact}^\infty(h)$ contains a realization of t_0.

2. $\implies$ **1.** Let $w \in \mathrm{Fact}^\infty(h)$ be a realization of t_0. By definition, there is an integer i and a letter $a \in \mathcal{A}$ such that $w \in \mathrm{Fact}(h^i(a))$. If w is a small realization of t_0 then we are done since $t_0 \in \mathrm{Anc}_h(t_0)$. If w is not a small realization, we can apply Lemma 1.12 and we know that there is a parent t_1 of t_0 and $w_1 \in \mathrm{Fact}(h^{i-1}(a))$ such that w_1 realizes t_1. By Lemma 1.12, if w_1 is not a small realization of t_1 there is a parent t_2 of t_1 and $w_2 \in \mathrm{Fact}(h^{i-2}(a))$ such that w_2 realizes t_2.

We can apply this reasoning inductively until we get a w_k which is a small realization of t_k. This happens eventually since $|w_k| \leq |h^{i-k}(a)|$. By construction t_k is an ancestor of t_0, so we have a small realization of an ancestor of t_0. □

We get the following corollary:

Corollary 1.14. *Let h be a primitive morphism prolongable at a, and t_0 a k-template. Let S be a set of k-template such that* $\mathrm{Ranc}_h(t_0) \subseteq S \subseteq \mathrm{Anc}_h(t_0)$. *Then the following conditions are equivalent:*

1. $h^\omega(a)$ *avoids* t_0,
2. $h^\omega(a)$ *avoids every small realizations of every elements of* S.

Any given template only has finitely many small realizations, and we only need to compute small factors of $h^\omega(a)$ to compute them. If we can compute a finite set S such that $\mathrm{Ranc}_h(t_0) \subseteq S \subseteq \mathrm{Anc}_h(t_0)$ then we can decide if $h^\omega(a)$ avoids t_0.

In particular, Currie and Rampersad showed that if M_h^{-1} is defined and has induced euclidean norm smaller than 1, then $\mathrm{Anc}_h(t_0)$ is finite and computable [19]. They deduced a result really similar to the following theorem:

Theorem 1.15. *For any primitive morphism h, if M_h^{-1} is defined and has induced euclidean norm smaller than 1. Then, for any template t_0, it is possible to decide whether $\mathrm{Fact}^\infty(h)$ realizes t_0.*

In the setting of Theorem 1.9 M_h is not necessarily invertible which implies that t_0 could have infinitely many parents and ancestors. Thus we need to find a way to discard many elements of $\mathrm{Anc}_h(t_0)$. In fact, using the Jordan normal form of M_h, we can find conditions on the vectors of the templates of $\mathrm{Ranc}_h(t_0)$.

1.3.2 Finding the set $\mathrm{Ranc}_h(t_0) \subseteq S \subseteq \mathrm{Anc}_h(t_0)$

Let $M = M_h$ be the matrix associated to h, *i.e.* $\forall i, j,\ M_{i,j} = |h(j)|_i$. We recall that we have the following equality:

$$\forall w \in \mathcal{A}^*,\ \Psi(h(w)) = M\Psi(w).$$

We assume that M has no eigenvalue of absolute value 1. Moreover, since it is primitive, it has at least one eigenvalue of absolute value greater than 1. From Proposition 1.5, there is an invertible matrix P and a Jordan matrix J such that $M = PJP^{-1}$. Thus $P^{-1}M = JP^{-1}$, and for any vector $\mathbf{x}$, $P^{-1}M\mathbf{x} = JP^{-1}\mathbf{x}$. We define the map r, such that $r(\mathbf{x}) = P^{-1}\mathbf{x}$ and its projections $\forall i,\ r_i(\mathbf{x}) = (P^{-1}\mathbf{x})_i$. Using this notation we have for any w, $r(\Psi(h(w))) = r(M\Psi(w)) = Jr(\Psi(w))$. Recall that J is as follows:

$$\begin{pmatrix} J_{n_1}(\lambda_1) & & & \\ & J_{n_2}(\lambda_2) & & 0 \\ & & \ddots & \\ 0 & & & J_{n_p}(\lambda_p) \end{pmatrix}$$

where the $J_{n_i}(\lambda_i)$ are Jordan blocks on the diagonal. That is, $J_n(\lambda)$ is a $n \times n$ matrix with $\lambda \in \mathbb{C}$ on the diagonal, 1 on top of the diagonal and 0 elsewhere. Note that it may happen that for $i \neq j$, $\lambda_i = \lambda_j$.

Bounds on the P basis We introduce some additional notations used in Propositions 1.16 and 1.18. Given a square matrix M and PJP^{-1} a Jordan decomposition of M, let $b : \{1, \dots, n\} \to \{1, \dots, p\}$ be the function that associates to an index i of M the number corresponding to its Jordan block in the matrix J, thus $\forall i \in \{1, \dots, n\}$, $\lambda_{b(i)} = J_{i,i}$. Let B be the map that associate to an index i the submatrix corresponding to the Jordan block containing this index, $\forall i \in \{1, \dots, n\}$, $B(i) = J_{n_{b(i)}}(\lambda_{b(i)})$. For any vector $\mathbf{x}$ and $1 \leq i_s \leq i_e \leq n$ such that i_s is the index of the first row of a Jordan block and i_e is the index of the last row of the same block, we denote by $\mathbf{x}_{[i_s,i_e]}$ the sub-vector of $\mathbf{x}$ starting at index i_s and ending at index i_e and then $(J\mathbf{x})_{[i_s,i_e]} = B(i)\mathbf{x}_{[i_s,i_e]}$. Let $E_c(M)$ be the *contracting eigenspace* of M, that is, the subspace generated by columns i of P such that $|\lambda_{b(i)}| < 1$. Similarly let $E_e(M)$ be the *expanding eigenspace* of M, that is, the subspace generated by columns i of P such that $|\lambda_{b(i)}| > 1$. Note that $E_c(M)$ and $E_e(M)$ are independent from the Jordan decomposition we chose.

We show that for any vector $\mathbf{x}$ appearing on a realizable ancestor of any template t_0 and any i, $|r_i(\mathbf{x})|$ is bounded, handling separately generalized eigenvectors of eigenvalues of absolute value less and more than 1. It implies that there are finitely many such integer vectors, since columns of P form a basis of $\mathbb{C}^n$.

Proposition 1.16. *For any i such that $|\lambda_{b(i)}| < 1$, $\{|r_i(\Psi(w))| : w \in \mathrm{Fact}^\infty(h)\}$ is bounded.*

Proof. Take i such that $|\lambda_{b(i)}| < 1$, and let i_s (resp. i_e) be the index that starts (resp. ends) the Jordan block $b(i)$ (thus $i_s \leq i \leq i_e$). Let w be a factor of $\mathrm{Fact}^\infty(h)$. Then there is a factor $w' \in \mathrm{Fact}(h)$, an integer l and for every $j \in \{0, \dots, l-1\}$, a pair of words $(s_j, p_j) \in (\mathrm{Suff}(h), \mathrm{Pref}(h))$ such that:

$$w = \left(\prod_{j=0}^{l-1} h^j(s_j) \right) h^l(w') \left(\prod_{j=l-1}^{0} h^j(p_j) \right).$$

Thus

$$r(\Psi(w)) = \sum_{j=0}^{l-1} J^j r(\Psi(s_j)) + J^l r(\Psi(w')) + \sum_{j=0}^{l-1} J^j r(\Psi(p_j))$$

and

$$r(\Psi(w))_{[i_s,i_e]} = \sum_{j=0}^{l-1} B(i)^j r(\Psi(s_j p_j))_{[i_s,i_e]} + B(i)^l r(\Psi(w'))_{[i_s,i_e]}.$$

Since $\lim_{l\to\infty}\left(\sum_{j=0}^{l} B(i)^j\right)$ exists, $|r_i(\Psi(w))|$ is bounded.

More precisely, a bound for $|r_i(\Psi(w))|$ can be found by the following way. Let $\mathcal{A}^{-1} = \{a^{-1} : a \in \mathcal{A}\}$ be the set of inverses of the letters of $\mathcal{A}$. Recall that the free group generated by $\mathcal{A}$ is the group made of the set of words over $\mathcal{A} \cup \mathcal{A}^{-1}$ where the only non-trivial equalities can be deduced from the fact that for all $a \in \mathcal{A}$, $aa^{-1} = a^{-1}a = \varepsilon$. We can also extend the notion of Parikh vector such that the Parick vector of the inverse of a letter count as a negative occurence of the letter. Now for any $a \in \mathcal{A} \cup \mathcal{A}^{-1}$ and word s, p and f such that $h(a) = pfs$ we have $fsh(a^{-1})pf = f$. For all $a \in \mathcal{A}$, $a \in \mathrm{Fact}(h)$, since h is primitive. It implies that for every $l' > l$ one can find $a \in \mathcal{A} \cup \mathcal{A}^{-1}$ and extend the sequence $(s_j, p_j)_{j\in\{0,\dots,l-1\}}$ to the sequence $(s_j, p_j)_{j\in\{0,\dots,l'-1\}}$ such that:

$$w = \left(\prod_{j=0}^{l'-1} h^j(s_j)\right) h^{l'}(a) \left(\prod_{j=l'-1}^{0} h^j(p_j)\right).$$

Thus there is an infinite sequence $(s_j, p_j)_{j\in\mathbb{N}}$ of elements in $(\mathrm{Suff}(h), \mathrm{Pref}(h))$ such that:

$$r(\Psi(w))_{[i_s,i_e]} = \sum_{j=0}^{\infty} B(i)^j r(\Psi(s_j p_j))_{[i_s,i_e]}.$$

For any i such that $|\lambda_{b(i)}| < 1$, $r_i(\Psi(w))$ is bounded by $\mathbf{u} \cdot \mathbf{v}$, where:

— $\mathbf{u}$ is the vector such that $\mathbf{u}_j = \max\{|r_j(\Psi(sp))| : (s,p) \in (\mathrm{Suff}(h), \mathrm{Pref}(h))\}$,

— $\mathbf{v}$ is the vector such that $\mathbf{v}_j = (1 - |\lambda_{b(i)}|)^{i-j-1}$ if $j \in \{i, \dots, i_e\}$, and zero otherwise. □

Let $r_i^* = 2 \times \max\{|r_i(\Psi(w))| : w \in \mathrm{Fact}^\infty(h)\}$. Let $\mathcal{R}_B$ be the set of templates $t = [a_1, \dots, a_{k+1}, \mathbf{d}_1, \dots, \mathbf{d}_{k-1}]$ such that for every i with $|\lambda_{b(i)}| < 1$ and $j \in \{1, \dots, k-1\}$, $|r_i(\mathbf{d}_j)| \le r_i^*$.

Corollary 1.17. *Every k-template which is realized by h is in $\mathcal{R}_B$.*

We need a tight upper bound on r_i^* for the algorithm corresponding to Theorem 1.9 to be efficient. The bound from the last proposition could be too loose, but we can reach better bounds by considering the fact that (since h is primitive) for any $l > 1$, h^l has the same factors than h. For example, for the abelian square-free morphism h_8 (Section 1.4) the bound for the eigenvalue $\lambda \sim 0.33292, +0.67077i$ is 5.9633, and become 1.4394 for the eigenvalue λ^{20} of $(h_8)^{20}$, while the observed bound on the prefix of size approximately 1 million of a fixed point of $(h_8)^2$ is 1.4341.

For any k-template t_0, we denote by $\mathbf{X}_{t_0}$ the set of all the vectors that appear on an ancestor of t_0.

Proposition 1.18. *For every i such that $|\lambda_{b(i)}| > 1$, for every k-template t_0, $\{|r_i(\mathbf{x})| : \mathbf{x} \in \mathbf{X}_{t_0}\}$ is bounded.*

Proof. The proof is close to the proof of Proposition 1.16. Let $\mathbf{x}$ be a vector of $\mathbf{X}_{t_0}$. If it is not a vector of t_0 then it appears on a template t which is a parent of an ancestors t' of t_0. If x' is the vector at the corresponding position in t' then, by definition of parent, there are $s, s', p, p' \in (\mathrm{Suff}(h), \mathrm{Suff}(h), \mathrm{Pref}(h), \mathrm{Pref}(h))$ such that $x' = Mx + \Psi(sp) - \Psi(s'p')$.

By induction there is a vector $\mathbf{x}_0$ of t_0, an integer l and a sequence of 4-tuple of words $(s_j, s'_j, p_j, p'_j)_{0 \leq i \leq l-1} \in (\mathrm{Suff}(h), \mathrm{Suff}(h), \mathrm{Pref}(h), \mathrm{Pref}(h))^{0 \leq i \leq l-1}$ such that:

$$\mathbf{x}_0 = \sum_{j=0}^{l-1} M^j \Psi(s_j p_j) + M^l \mathbf{x} - \sum_{j=0}^{l-1} M^j \Psi(s'_j p'_j).$$

Thus

$$r(\mathbf{x}_0) = \sum_{j=0}^{l-1} J^j r(\Psi(s_j p_j) - \Psi(s'_j p'_j)) + J^l r(\mathbf{x}).$$

Let i_s (resp. i_e) be the starting (resp. ending) index of the block $b(i)$. Thus

$$B(i)^l r(\mathbf{x})_{[i_s, i_e]} = r(\mathbf{x}_0)_{[i_s, i_e]} + \sum_{j=0}^{l-1} B(i)^j r(\Psi(s'_j p'_j) - \Psi(s_j p_j))_{[i_s, i_e]}.$$

Moreover we know that $B(i)$ is invertible so:

$$r(\mathbf{x})_{[i_s, i_e]} = B(i)^{-l} r(\mathbf{x}_0)_{[i_s, i_e]} + \sum_{j=0}^{l-1} B(i)^{j-l} (r(\Psi(s'_j p'_j) - \Psi(s_j p_j))_{[i_s, i_e]}.$$

The only eigenvalue of $B(i)^{-1}$ is $\lambda_{b(i)}^{-1}$ and has absolute value less than 1, thus $\sum_{j=1}^{\infty} ||B(i)^{-j}||$ converges. Hence $||r(\mathbf{x})_{[i_s, i_e]}||$ can be bounded by a constant depending only on h, P, J and i. Thus there is a constant r^*_{i,t_0} such that for all $\mathbf{x} \in \mathbf{X}_{t_0}$, $|r_i(\mathbf{x})| \leq r^*_{i,t_0}$. □

In paragraph *Computing S efficiently*, we explain why we do not need to compute a value for the bound r^*_{i,t_0}. Since the columns of P is a basis,

Propositions 1.16 and 1.18 imply that the norm of any vector of a k-template from $\mathcal{R}_B \cap \mathrm{Anc}_h(t_0)$ is bounded, and thus $\mathcal{R}_B \cap \mathrm{Anc}_h(t_0)$ is finite. We sum up all the interesting properties about $\mathcal{R}_B \cap \mathrm{Anc}_h(t_0)$ in the next corollary:

Corollary 1.19. *For any template t_0 and any morphism h whose matrix has no eigenvalue of absolute value 1, we have:*

— $\mathrm{Ranc}_h(t_0) \subseteq \mathcal{R}_B \cap \mathrm{Anc}_h(t_0) \subseteq \mathrm{Anc}_h(t_0)$,

— $\mathcal{R}_B \cap \mathrm{Anc}_h(t_0)$ *is finite,*

From Corollary 1.14 and Corollary 1.19, we know that if we can compute $\mathcal{R}_B \cap \mathrm{Anc}_h(t_0)$ then we can decide whether $h_\omega(a)$ avoids abelian k-th powers.

We can deduce from Propositions 1.16 and 1.18 a naive algorithm to compute a set S of templates such that $\mathrm{Ranc}_h(t_0) \subseteq S \subseteq \mathrm{Anc}_h(t_0)$. We first compute a set of templates T_{t_0} whose vectors' coordinates in basis P are bounded by r_i^* or r_{i,t_0}^*, then we compute the parent relation inside T_{t_0} and we select the parents that are accessible from t_0. We explain at the end of this section a more efficient way to compute such a set S.

We summarize the proof of Theorem 1.9. We know from Corollary 1.19 that one can compute a set S such that $\mathrm{Ranc}_h(t_0) \subseteq S \subseteq \mathrm{Anc}_h(t_0)$. Moreover from Corollary 1.14 we know that the followings are equivalent:

1. $h^\omega(a)$ avoids t_0,
2. $h^\omega(a)$ avoids every small realizations of every elements of S.

For any integer l, we can compute every factor of $h^\omega(a)$ of bounded size l. Moreover S is finite so we can check every template of S one by one. So we can check condition 2 with a computer. Hence one can decide whether $h^\omega(a)$ avoids t_0.

Computing S efficiently The naive algorithm we gave is not efficient, since for morphisms whose fixed points avoid abelian powers, the set of ancestors $\mathcal{R}_B \cap \mathrm{Anc}_h(t_0)$ is usually very small relatively to T_{t_0}.

The following algorithm does not necessarily compute $\mathcal{R}_B \cap \mathrm{Anc}_h(t_0)$, but a set S such that $\mathrm{Ranc}_h(t_0) \subseteq S \subseteq \mathcal{R}_B \cap \mathrm{Anc}_h(t_0)$. We compute recursively a set of templates A_{t_0} that we initialize at $\{t_0\}$, and each time that we add a new template t, we compute the set of parents of t which are in $\mathcal{R}_B$ and add them to A_{t_0}. At any time we have $A_{t_0} \subseteq \mathcal{R}_B \cap \mathrm{Anc}_h(t_0)$ which is finite so this algorithm terminates. Moreover if a parent of a template is realizable then this template also is realizable. It implies that, at the end, $\mathrm{Ranc}_h(t_0) \subseteq A_{t_0}$.

We need to be able to compute a finite superset of the set of realizable parents of a template. Let $t = [a_1, \dots, a_{k+1}, \mathbf{d}_1, \dots, \mathbf{d}_{k-1}]$ be a template, and assume that $t' = [a'_1, \dots, a'_{k+1}, \mathbf{d'}_1, \dots, \mathbf{d'}_{k-1}]$ is a parent of t, and t' is realizable by h. Then there are $p_1, s_1, \dots, p_{k+1}, s_{k+1} \in \mathcal{A}^*$ such that:

— $\forall i \in \{1, \dots, k+1\}$, $h(a'_i) = p_i a_i s_i$,

— $\forall i \in \{1, \dots, k-1\}$, $\mathbf{d}_i = M\mathbf{d'}_i + \Psi(s_{i+1}p_{i+2}) - \Psi(s_i p_{i+1})$.

There are finitely many ways of choosing the a'_i in t' and finitely many ways of choosing the s_i and the p_i, so we only need to be able to compute the possible values of the $\mathbf{d'}_i$ of a template with fixed $a'_1, \dots, a'_{k+1}$ and $s_1, p_1, \dots, s_{k+1}, p_{k+1}$. (Note that this is easy if M is invertible.)

Suppose we want to compute $\mathbf{d'}_m$ for some m. That is, we want to compute all the integer solutions $\mathbf{x}$ of $M\mathbf{x} = \mathbf{v}$, where $\mathbf{v} = \mathbf{d}_m - \Psi(s_{m+1}p_{m+2}) + \Psi(s_m p_{m+1})$. Moreover, since we are interested by realizable parents we can restrict ourself to solutions that respect the bounds from Proposition 1.16. The rest is only linear algebra.

First, we can use the smith decomposition of M, as explained after Proposition 1.6, in order to find a particular solution $\mathbf{x_0}$ and a basis $(\beta_1, ..., \beta_\kappa)$ (where $\kappa = \dim \ker(M)$) of the lattice $\Lambda = \ker(M) \cap \mathbb{Z}^n$. If this equation has no integer solution, then the template t has no parents with this choice of a_i, p_i and s_i. We are only interested in parents realizable by h, so we want to compute the set $\mathbf{X} = \{\mathbf{x} \in \mathbf{x_0} + \Lambda : \ \forall i \text{ s.t. } |\lambda_{b(i)}| < 1, \ |r_i(\mathbf{x})| \leq r_i^*\}$. Since Λ is included in the union of the generalized eigenspaces of eigenvalue 0, we know by Proposition 1.16 that $\mathbf{X}$ is finite. Let $\mathcal{B}$ be the matrix whose columns are the elements of the basis $(\beta_1, ..., \beta_\kappa)$, and let $\mathbf{X}_\mathcal{B} = \{\mathbf{x} \in \mathbb{Z}^\kappa : \mathbf{x_0} + \mathcal{B}\mathbf{x} \in \mathbf{X}\}$. $\ker(M)$ is generated by $\mathcal{B}$ but also by the generalized eigenvectors corresponding to a null eigenvalue which are columns of P. So there is a matrix Q made of rows of P^{-1} such that $Q\mathcal{B}$ is invertible. All the rows of Q are rows of P^{-1} thus from Proposition 1.16 there are $c_1, \dots, c_\kappa \in \mathbb{R}$ such that for any $\mathbf{x} \in \mathbf{X}_\mathcal{B}$ and $i \in \{1, \dots, \kappa\}$, $|(Q(\mathcal{B}\mathbf{x} + \mathbf{x_0}))_i| \leq c_i$ thus $|(Q\mathcal{B}\mathbf{x})_i| \leq c_i + |(Q\mathbf{x_0})_i|$. Then:

$$||Q\mathcal{B}\mathbf{x}||^2 \leq \sum_{i=1}^{\kappa} (c_i + |(Q\mathbf{x_0})_i|)^2.$$

Let $c = \sum_{i=1}^{\kappa} (c_i + |(Q\mathbf{x_0})_i|)^2$. From Proposition 1.7, if μ_{min} is the smallest eigenvalue of $(Q\mathcal{B})^*(Q\mathcal{B})$ then $\mu_{min}||\mathbf{x}||^2 \leq ||Q\mathcal{B}\mathbf{x}||^2 \leq c$. Moreover $Q\mathcal{B}$ is invertible, thus $\mu_{min} \neq 0$, and $\mathbf{X}_\mathcal{B}$ contains only integer points in the ball of

radius $\sqrt{\frac{c}{\mu_{min}}}$. We can easily compute a finite super-set of $\mathbf{X}_{\mathcal{B}}$, and thus of $\mathbf{X}$, and then we can select the elements that are actually in $\mathbf{X}$. The choice of $\mathbf{x_0}$ is significant for the sharpness of the bound c: it is preferable to take a $\mathbf{x_0}$ nearly orthogonal to $\ker(M)$.

1.4 Abelian square-free pure morphic words

We implemented the algorithm described in the previous sections. We can use this algorithm to check the following results.

Let h_6 be the following morphism:

$$h_6 : \begin{cases} a \to ace & b \to adf \\ c \to bdf & d \to bdc \\ e \to afe & f \to bce. \end{cases}$$

Theorem 1.20. *$h_6^\omega(a)$ is abelian square-free.*

We provide a computer program [1] that applies the algorithm described in the previous section in order to show Theorem 1.20.

The matrix associated has the following eigenvalues: 0 (with algebraic multiplicity 3), 3, $\sqrt{3}$ and $-\sqrt{3}$. A Jordan decomposition of M_{h_6} is PJP^{-1}, with:

$$J = \begin{bmatrix} 0&1&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&3&0&0 \\ 0&0&0&0&\sqrt{3}&0 \\ 0&0&0&0&0&-\sqrt{3} \end{bmatrix} \text{ and } P = \begin{bmatrix} -\frac{1}{2} & 0 & -1 & 1 & 2+\sqrt{3} & 2-\sqrt{3} \\ \frac{1}{2} & -1 & 0 & 1 & -2-\sqrt{3} & \sqrt{3}-2 \\ -\frac{1}{2} & 1 & -1 & 1 & -1 & -1 \\ 0 & 0 & 1 & 1 & -3-2\sqrt{3} & 2\sqrt{3}-3 \\ 0 & \frac{1}{2} & 1 & 1 & 3+2\sqrt{3} & 3-2\sqrt{3} \\ \frac{1}{2} & -\frac{1}{2} & 0 & 1 & 1 & 1 \end{bmatrix}.$$

The bounds on r_i^*, $i \in \{1,2,3\}$ computed as explained in the proof of Proposition 1.16 on $(h_6)^2$, are respectively 4, $\frac{4}{3}$ and $\frac{4}{3}$. The template $[\varepsilon, \varepsilon, \varepsilon, \overrightarrow{0}]$ has 28514 parents with respect to those bounds, and it has 48459 different ancestors including itself. None of the factors of $h_6^\omega(a)$ is a small realization of a forbidden template so we can conclude that $h_6^\omega(a)$ avoids abelian squares.

1. `https://arxiv.org/abs/1511.05875`

From Proposition 1.16, the Parikh vectors of the factors of $h_6^\omega(a)$ are close to a subspace of dimension 3. In Chapter 3, we need such a morphism in order to apply our proof technique, so finding this morphism is the first step in showing that long abelian squares are avoidable over the ternary alphabet. It seems hard to find simpler morphisms with this property, in particular we are interested by the following question:

Problem 1.21. *Is there an abelian square-free pure morphic word over* 4 *or* 5 *letters generated by a morphism with only* 3 *eigenvalues of norm at least* 1*?*

In fact, for similar reasons, a positive answer to the following question could help to show that additive squares are avoidable over $\mathbb{Z}$:

Problem 1.22. *Is there an abelian square-free pure morphic word generated by a morphism with only* 2 *eigenvalues of norm at least* 1*?*

Let h_8 be the following morphism:

$$h_8 : \begin{cases} a \to h & b \to g \\ c \to f & d \to e \\ e \to hc & f \to ac \\ g \to db & h \to eb. \end{cases}$$

Theorem 1.23. *Words in h_8^∞ (e.g. infinite fixed points of $(h_8)^2$) are abelian square-free.*

This morphism may also be interesting because it is a small morphism which gives an abelian square-free word, its matrix is invertible and it has 4 eigenvalues of absolute value less than 1. In particular, such a morphism could be part of a simpler construction of an abelian square-free word over 4 letters.

Our algorithm can also show that the infinite word generated by $h : a \mapsto ac, b \mapsto dc, c \mapsto b, d \mapsto ab$ is abelian cube-free. It would be interesting for the sake of completeness to be able to decide the abelian k-th power freeness for any morphism. We can get ride of the primitivity condition with some technicalities, but it seems much harder to deal with eigenvalues of absolute value exactly 1.

Problem 1.24. *Is it decidable, for any morphism h, whether the fixed points of h are abelian k-th power-free?*

In fact, we do not know any example of interesting morphism with an eigenvalue of norm 1 generating an abelian k-th power-free word.

Chapter 2

Avoidability of Additive Powers

Let $(G,+)$ be a semigroup and $\Phi : (\mathcal{A}^*,.) \mapsto (G,+)$ be a morphism. Two words u and v are *equivalent modulo* Φ, denoted by $u \approx_\Phi v$, if $\Phi(u) = \Phi(v)$. For any $k \geq 2$, a *k-th power modulo* Φ is a word $w = w_1 w_2 \dots w_k$ such that for all $i \in \{2, \dots, k\}$, $\Phi(w_i) = \Phi(w_1)$. If moreover $|w_1| = |w_2| = \dots = |w_k|$ then it is a *uniform k-th power modulo* Φ. Uniform k-th powers modulo Φ are often called *additive k-th powers*, without mention of the morphism Φ, if the morphism Φ use is clear in the context. A *square modulo* Φ (resp. *cube modulo* Φ) is a 2nd power (resp. 3rd power) modulo Φ. We say that $(G,+)$ is *k-repetitive* (resp. *uniformly k-repetitive*) if for any finite alphabet $\mathcal{A}$ and any morphism $\Phi : (\mathcal{A}^*,.) \mapsto (G,+)$ every infinite word over $\mathcal{A}$ contains a k-th power modulo Φ (resp. a uniform k-th power modulo Φ).

For instance, if G is the free-monoid generated by $\mathcal{A}$ and Φ is the identity, k-th powers modulo Φ are exactly k-th powers. So from the avoidability of squares over the ternary alphabet we deduce that if $|\mathcal{A}| \geq 3$ the free-monoid generated by $\mathcal{A}$ is not 2 repetitive. If G is the free abelian monoid generated by $\mathcal{A}$ and for all $a \in \mathcal{A}$, $\Phi(a) = a$ then k-th powers modulo Φ are exactly abelian k-th powers. So Theorem 1.1 implies that if $|\mathcal{A}| \geq 4$ the free abelian monoid generated by $\mathcal{A}$ is not 2 repetitive.

A direct application of Szemerédi's Theorem shows that for any finite alphabet $\mathcal{A}$, $\Phi : \mathcal{A} \mapsto \mathbb{Z}$ and $k \in \mathbb{N}$, it is not possible to avoid k-th powers modulo Φ over $\mathcal{A}$, that is, $(\mathbb{Z},+)$ is k-repetitive for any k. On the other hand, whether $\mathbb{Z}$ is uniformly 2-repetitive or not is a long standing open question [32, 49], and it was recently showed that $\mathbb{Z}$ is not uniformly 3-repetitive [15]. The main result of this chapter is Theorem 2.5, which states that $\mathbb{Z}^2$ is not uniformly 2-repetitive .

In the rest of the chapter, we only consider $(G, +) = (\mathbb{Z}^d, +)$ for some $d > 0$. Note that, for any integers n and k, if $(\mathbb{Z}^{n+1}, +)$ is k-repetitive then $(\mathbb{Z}^n, +)$ is uniformly k-repetitive. Φ can be seen as a linear map from the Parikh vector of a word to $\mathbb{Z}^d$, therefore we can associate to Φ the matrix F_Φ such that $\forall w \in \mathcal{A}^*$, $\Phi(w) = F_\Phi \Psi(w)$. Clearly, if $d = |\mathcal{A}|$ and F_Φ is invertible then two words are abelian-equivalent if and only if they have the same image by Φ.

In this chapter we show that under some conditions one can decide if a morphism generates a word which is additive k-th power-free. Then we use this result to give some new results about the avoidability of additive powers.

The decidability algorithm is mainly based on the results from the Chapter 1. We also use many notations introduced in Chapter 1. The results presented in this chapter are joint work with Michaël Rao and are part of the article [54].

2.1 Decidability

Let $\Phi : (\mathcal{A}^*, .) \to (\mathbb{Z}^d, +)$ be a morphism with $d \in \mathbb{N}$ and $h : \mathcal{A}^* \mapsto \mathcal{A}^*$ be a primitive morphism. Let the matrix F_Φ be such that $\forall w$, $\Phi(w) = F_\Phi \Psi(w)$.

Let M_h be the matrix associated to h and let $M_h = PJP^{-1}$ be a Jordan decomposition of M_h. Let $E_c(M_h)$ be the *contracting eigenspace* of M_h, that is, the subspace generated by columns i of P such that $|\lambda_{b(i)}| < 1$. Similarly let $E_e(M_h)$ be the *expanding eigenspace* of M_h, that is, the subspace generated by columns i of P such that $|\lambda_{b(i)}| > 1$. Note that $E_c(M_h)$ and $E_e(M_h)$ are independent from the Jordan decomposition we choose.

Proposition 2.1. *If M_h has no eigenvalue of absolute value 1 and $E_e(M_h) \cap \ker(F_\Phi) = \{\vec{0}\}$ then one can compute a finite set of templates S such that each k-th power modulo Φ in* $\mathrm{Fact}^\infty(\mathrm{h})$ *is a realization of a template in S.*

Proof. Let $\kappa = \dim \ker(F_\Phi)$ and $\Lambda = \ker(F_\Phi) \cap \mathbb{Z}^d$. By definition any k-th power modulo Φ realizes at least one template of the form $t = [\varepsilon, \ldots, \varepsilon, \mathbf{d}_1, \ldots, \mathbf{d}_{k-1}]$ where for all i, $\mathbf{d}_i \in \Lambda$. We use the Smith decomposition of F_Φ, as explained after Proposition 1.6, to get the matrix B, whose columns form an integral basis of Λ.

Let Q be the rectangular submatrix of P^{-1} such that the i-th line of P^{-1} is a line of Q if and only if $|\lambda_{b(i)}| < 1$. By definition of B, for every $\mathbf{x} \in \mathbb{C}^\kappa \setminus \{\vec{0}\}$, $B\mathbf{x} \in \ker(F_\Phi)$ then, by hypothesis, $B\mathbf{x} \notin E_e(M_h)$. Since the

lines of Q generate the subspace orthogonal to $E_e(M_h)$, $QB\mathbf{x} \neq \overrightarrow{0}$. Thus we have $\text{rank}(QB) = \kappa$ which implies that there is a submatrix Q' of Q such that $Q'B$ is invertible.

For all $i \in \{1, \ldots, \kappa\}$, let p_i be the function such that for all vector $\mathbf{x}$, $p_i(\mathbf{x}) = (Q'\mathbf{x})_i$. From Proposition 1.16, for all $i \in \{1, \ldots, \kappa\}$, there is $c_i \in \mathbb{R}$ such that for any two factors u and v of $\text{Fact}^\infty(h)$, $|p_i(\Psi(u) - \Psi(v))| \leq c_i$.

Let $\mathbf{X} = \{\mathbf{x} \in \Lambda : \forall i \in \{1, \ldots, \kappa\},\ |p_i(\mathbf{x})| \leq c_i\}$. Since we are only interested in realizable templates for S, we can add the condition: for all i, $\mathbf{d}_i \in \mathbf{X}$.

Let $\mathbf{X}_B = \{\mathbf{x} \in \mathbb{Z}^\kappa :\ B\mathbf{x} \in \mathbf{X}\}$ and $\mathbf{x} \in \mathbf{X}_B$. Then for all i, $|p_i(B\mathbf{x})| \leq c_i$, then $||Q'B\mathbf{x}||^2 \leq \sum_{i=1}^{l} c_i^2 = c$. From Proposition 1.7, if μ_{min} is the smallest eigenvalue of $(Q'B)^*(Q'B)$, we have $\mu_{min}||\mathbf{x}||^2 \leq ||Q'B\mathbf{x}||^2 \leq c$. Since $Q'B$ is invertible, $\mu_{min} \neq 0$ and $||\mathbf{x}|| \leq \sqrt{\frac{c}{\mu_{min}}}$. Then $\mathbf{X}_B$ and $\mathbf{X}$ are finite, and we can easily compute them.

So we can compute $S = \{[\varepsilon, \ldots, \varepsilon, \mathbf{d}_1, \ldots, \mathbf{d}_{k-1}] : \forall i, \mathbf{d}_i \in \mathbf{X}\}$. □

From Theorem 1.9 we know that for any given template we can decide whether it is avoided by a word generated by a primitive morphism with no eigenvalue of absolute value 1. We can deduce the following result:

Theorem 2.2. *Let $h : \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism with no eigenvalue of absolute value 1, and let $\Phi : \mathcal{A}^* \to \mathbb{Z}^d$ a morphism. If $E_e(M_h) \cap \ker(\Phi) = \{\overrightarrow{0}\}$ then one can decide whether every word in $\text{Fact}^\infty(h)$ avoids k-th powers modulo Φ.*

The conditions from Theorem 2.2 seem restrictive, but we can apply this Theorem to every morphic word avoiding additive powers that we found. It seems reasonable to think that the condition $E_e(M_h) \cap \ker(\Phi) = \{\overrightarrow{0}\}$ is necessary in order to generate a word avoiding k-th power modulo Φ. But for the sake of completeness, we ask the following question.

Problem 2.3. *Is there an algorithm deciding k-th power modulo Φ freeness of (pure) morphic words?*

2.2 Results

In this section we use the algorithm described in this chapter to show that additive squares are avoidable over $\mathbb{Z}^2$. We also give some other results

about the avoidability of additive cubes over different alphabets from $\mathbb{Z}$. We provide a computer program [1] that applies the algorithm described in the previous section in order to show Theorem 2.4.

2.2.1 Additive square-free words over $\mathbb{Z}^2$

Let h_6 be the morphism that we already used in Section 1.4:

$$h_6 : \begin{cases} a \to ace & b \to adf \\ c \to bdf & d \to bdc \\ e \to afe & f \to bce. \end{cases}$$

Let Φ be the following morphism:

$$\Phi : \begin{cases} a \to (1,0,0) & b \to (1,1,1) \\ c \to (1,2,1) & d \to (1,0,1) \\ e \to (1,2,0) & f \to (1,1,0). \end{cases}$$

Theorem 2.4. *$h_6^\omega(a)$ does not contains squares modulo Φ.*

In other words, the fixed point $h_{\text{add}}^\omega\left(\begin{pmatrix}0\\0\end{pmatrix}\right)$ of the following morphism does not contain any additive square.

$$h_{\text{add}} : \begin{cases} \begin{pmatrix}0\\0\end{pmatrix} \to \begin{pmatrix}0\\0\end{pmatrix}\begin{pmatrix}2\\1\end{pmatrix}\begin{pmatrix}2\\0\end{pmatrix} & \begin{pmatrix}1\\1\end{pmatrix} \to \begin{pmatrix}0\\0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} \\ \begin{pmatrix}2\\1\end{pmatrix} \to \begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} & \begin{pmatrix}0\\1\end{pmatrix} \to \begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}2\\1\end{pmatrix} \\ \begin{pmatrix}2\\0\end{pmatrix} \to \begin{pmatrix}0\\0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}2\\0\end{pmatrix} & \begin{pmatrix}1\\0\end{pmatrix} \to \begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}2\\1\end{pmatrix}\begin{pmatrix}2\\0\end{pmatrix}. \end{cases}$$

It implies the following result:

Theorem 2.5. *$\mathbb{Z}^2$ is not uniformly 2-repetitive.*

It seems rather natural to ask:

Problem 2.6. *What is the smallest alphabet $\mathcal{A} \subseteq \mathbb{Z}^2$ over which we can avoid additive squares?*

1. `https://arxiv.org/abs/1511.05875`

2.2.2 Additive cubes-free words over $\mathbb{Z}$

Cassaigne *et al.* showed that the fixed point of $f : 0 \to 03, 1 \to 43, 3 \to 1, 4 \to 01$, avoids additive cubes [15]. Our algorithm is able to reach the same conclusion for this morphism. We can also use it to show that additive cubes are avoidable over some other alphabets of size 4.

$$\text{Let } h_4 : \begin{cases} 0 \to 001 \\ 1 \to 041 \\ 2 \to 41 \\ 4 \to 442 \end{cases}, h'_4 : \begin{cases} 0 \to 03 \\ 2 \to 53 \\ 3 \to 2 \\ 5 \to 02. \end{cases} \text{ and } h''_4 : \begin{cases} 0 \to 03 \\ 2 \to 63 \\ 3 \to 2 \\ 6 \to 02. \end{cases}$$

Theorem 2.7. *$h_4^\omega(0)$, $h_4'^\omega(0)$ and $h_4''^\omega(0)$ avoid additive cubes.*

It seems that $\{0, 1, 2, 3\}$ is the only alphabet of 4 integers over which additive cubes are hard to avoid. It gives us a motivation to find a characterization of the alphabets of integers over which additive cubes are avoidable.

Rao conjectured the following property:

Conjecture 2.8 ([53])**.** *For any integers $i < j$ such that i and j are coprime and $j \geq 6$, additive cubes are avoidable over $\{0, i, j\}$.*

He showed that the conjecture is true for $6 \leq j \leq 9$ and that additive cubes are avoidable over $\{0, 1, 5\}$ [53].

Theorem 2.9. *Let $\mathcal{A} \in \mathbb{Z}^3$. If $\mathcal{A} = \{0, 1, 5\}$ or if $\mathcal{A} = \{0, i, j\}$ with i and j coprime and $6 \leq j \leq 9$ then additive squares are avoidable over $\mathcal{A}$.*

Note that if we apply the same affine function of the form $x \mapsto ax+b$ with $a \neq 0$ to each letter of an alphabet $\mathcal{A}$ to get a new alphabet $\mathcal{A}'$ then we can avoid exactly the same additive powers over $\mathcal{A}$ and $\mathcal{A}'$. We say that $\mathcal{A}$ and $\mathcal{A}'$ are equivalent. This is why without loss of generality every alphabet contains 0 and positive integers. It also justifies that, in the previous conjecture, the cases where i and j are not coprime are not interesting.

Using Theorem 2.7 and Theorem 2.9 we can show the following result:

Proposition 2.10. *Let $\mathcal{A} = \{0, i, j, k\}$ be an integer alphabet such that $0 < i < j < k < 10$. If additive cubes are not avoidable over $\mathcal{A}$, then $\mathcal{A}$ is equivalent to $\{0, 1, 2, 3\}$.*

Proof. Let $\mathcal{A} = \{0, i, j, k\}$ be an integer alphabet such that $0 < i < j < k < 10$ and such that additive cubes are not avoidable over $\mathcal{A}$. We distinguish different cases depending on the value of k. Without loss of generality, we can suppose that i, j and k are coprime (we need to consider only one representant by equivalence class).

$k = 3$. Then $\mathcal{A} = \{0, 1, 2, 3\}$.

$k = 4$. $\{0, 1, 2, 4\}$ and $\{0, 2, 3, 4\}$ are equivalent by $x \mapsto 4 - x$ and additive cubes are avoidable over this alphabets by Theorem 2.7. Additive cubes are avoidable over $\{0, 1, 3, 4\}$ [15].

$k = 5$. If $i = 1$ or $j = 4$ then $\mathcal{A}$ contains $\{0, 1, 5\}$ or $\{0, 4, 5\}$ so additive cubes are avoidable by Theorem 2.9. By Theorem 2.7, additive cubes are avoidable over $\{0, 2, 3, 5\}$.

$k = 6$. If $i = 1$ or $j = 5$ then $\mathcal{A}$ contains $\{0, 1, 6\}$ or $\{0, 5, 6\}$ so additive cubes are avoidable by Theorem 2.9. By Theorem 2.7, additive cubes are avoidable over $\{0, 2, 3, 6\}$ (and over $\{0, 3, 4, 6\}$ since they are equivalent). The only remaining possibility is $\{0, 2, 4, 6\}$ which is excluded by coprimality.

$k = 7$. By Theorem 2.9, additive cubes are avoidable over $\{0, i, 7\}$ and thus over $\mathcal{A}$.

$k = 8$. If i (resp. j) is odd then by Theorem 2.9 additive cubes are avoidable over $\{0, i, 8\}$ (resp. $\{0, j, 8\}$). If i and j are both even then i, j, k are not coprime.

$k = 9$. If i (resp. j) is not divisible by 3 then by Theorem 2.9 additive cubes are avoidable over $\{0, i, 9\}$ (resp. $\{0, j, 9\}$). If i and j are both divisible by 3 then i, j, k are not coprime. □

If Conjecture 2.8 is true, we can show the following stronger result:

Proposition 2.11. *Conjecture 2.8 implies that for every alphabet $\mathcal{A} = \{0, i, j, k\}$ if additive cubes are not avoidable over $\mathcal{A}$ then $\mathcal{A}$ is equivalent to $\{0, 1, 2, 3\}$.*

Proof. In the proof we say that an alphabet is *good* if additive cubes are avoidable over this alphabet.

In fact Conjecture 2.8 implies that if there is an integer $p \geq 6$ which divides n but is coprime with m then $\{0, n, m\}$ is good. Indeed, since $\frac{n}{\gcd(n,m)} \geq p$, then by Conjecture 2.8 $\left\{0, \frac{n}{\gcd(n,m)}, \frac{m}{\gcd(n,m)}\right\}$ is good which implies that $\{0, n, m\}$ is good. In the following we call this property (P).

Let $\mathcal{A} = \{0, i, j, k\}$ be an alphabet which is not good with i, j and k coprime.

If one of i, j or k is divisible by a prime number $p > 5$, then one of them is not divisible by p by coprimality. Then by (P), $\mathcal{A}$ is good.

Then none of i, j or k is divisible by a prime number $p > 5$. Thus they are only divisible by 2, 3 and 5. At least one of them is not divisible by 5 so if any of them is divisible by $25 = 5^2$, (P) implies that $\mathcal{A}$ is good. For the same reason none of i, j or k is divisible by 3^2 or by 2^3.

This implies that $i, j, k \in \{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}$. W.l.o.g. $i < j < k$. We can do a disjunction over the value of k.

k is divisible by 30. Then i and j are both greater than 5 otherwise, by (P), $\mathcal{A}$ is good. $\{0, 6, 30\}$ is equivalent to $\{0, 1, 5\}$ and $\{0, 6, 60\}$ is equivalent to $\{0, 1, 10\}$ so they are both good. So $i, j \in \{10, 12, 15, 20, 30\}$. By coprimality only one of i or j is divisible by 5 so one of them is equal to 12. By coprimality of i and j, there is no choice left for the last letter.

$k = 20$. Since $\{0, 1, 5\}$ is good and equivalent to $\{0, 4, 20\}$ none of i or j is equal to 4. By (P) none of i or j is equal to 1, 2, 3 or 6. So $i, j \in \{5, 10, 12, 15\}$. By coprimality one of them is 12, and then the other cannot be 10. By (P), $\{0, 5, 12\}$ is good, while $\{0, 12, 15\}$ is equivalent to $\{0, 1, 5\}$ which is good. So every possible alphabet is good for $k = 20$.

$k = 15$. Since $\{0, 1, 5\}$ is good, $\{0, 12, 15\}$ and $\{0, 3, 15\}$ are good. By (P), i and j cannot be equal to 1, 2 or 4. So $i, j \in \{5, 6, 10\}$. By coprimality one of them is equal to 6. Since $\{0, 5, 6\}$ is good, $\{0, 5, 6, 15\}$ is good. The only remaining is possibility $\{0, 6, 10, 15\}$. This alphabet contains $\{6, 10, 15\}$ which is equivalent to $\{0, 4, 9\}$ which is good.

$k = 12$. By (P), i and j are not equal to 1 or 5. Since $\{0, 2, 12\}$ and $\{0, 10, 12\}$ are both equivalent to $\{0, 1, 6\}$ which is good, i and j are not equal to 2 or 10. So $i, j \in \{3, 4, 6\}$. By coprimality the only possible alphabet is $\{0, 3, 4, 12\}$, which contains $\{3, 4, 12\}$ which is equivalent to $\{0, 1, 9\}$ and is good.

$k = 10$. By (P), $\{0, 1, 10\}$ and $\{0, 3, 10\}$ are good, so i and j cannot be equal to 1 or 3. $\{0, 2, 10\}$ is equivalent to $\{0, 1, 5\}$ which is good so i and

j cannot be equal to 2. So $i, j \in \{4, 5, 6\}$. By coprimality one of them is 5. The alphabets $\{0, 5, 6, 10\}$ and $\{0, 4, 5, 10\}$ are equivalent, and the second one contains $\{4, 5, 10\}$ which is good because it is equivalent to $\{0, 1, 6\}$.

$k < 10$**.** See Proposition 2.10. □

Clearly, Rao's conjecture also implies that additive cubes are avoidable over any alphabet of 5 or more integers. On the other hand additive cubes cannot be avoided over two integers since abelian cubes are not avoidable over two letters. Other than proving Rao's Conjecture we are left with some related open questions:

Problem 2.12. *Are additive cubes avoidable over* $\{0, 1, 2, 3\}$*?* $\{0, 1, 4\}$*?* $\{0, 2, 5\}$*?*

The subsets of size 3 of $\{0, 1, 2, 3\}$ are missing from this question because additive cubes are probably not avoidable over $\{0, 1, 2, 3\}$ which implies the same result for any of its subsets.

Finally, recall that one main open question regarding avoidability of additive powers is the 2-repetitivity of $\mathbb{Z}$:

Problem 2.13 ([49])**.** *Are additive squares avoidable over a finite subset of* $\mathbb{Z}$*?*

Chapter 3

Avoidability of Long Abelian Powers

Erdős asked whether it is possible to avoid arbitrarily long squares on binary words [25]. Entringer, Jackson and Schatz showed that it is possible to avoid squares of period at least 3 over the binary alphabet, but that long abelian squares are not avoidable over the binary alphabet [23]. On the other hand Keränen answered positively another Erdős' question by proving that abelian squares are avoidable over 4 letters. This led Mäkelä to ask the following question about the avoidability of long abelian squares on a ternary alphabet:

Problem 3.1 (Mäkelä (see [35]))**.** *Can you avoid abelian squares of the form uv where $|u| \geq 2$ over three letters ? - Computer experiments show that you can avoid these patterns at least in words of length 450.*

He also asked the following similar question:

Problem 3.2 (Mäkelä (see [35]))**.** *Can you avoid abelian cubes of the form uvw where $|u| \geq 2$, over two letters ? - You can do this at least for words of length 250.*

In this section, we will first show that the answer to the second question is negative. It leads us to ask, in Problem 3.6, whether there is an integer p such that abelian cubes of period more than p are avoidable over two letters. In Problem 3.7, we ask whether there is an integer p such that abelian squares of period more than p are avoidable over three letters.

We then give an algorithm, based on the algorithm from Chapter 1, deciding under some conditions whether a morphic word avoids abelian k-th powers of period more than a given p. In Section 3.3, we use this algorithm to solve Problem 3.7 by showing that abelian squares of period more than 5 are avoidable over the ternary alphabet. The results presented in this Chapter are joint work with Michaël Rao and are part of the articles [54] and [55]. We use the notations introduced in Chapter 1.

3.1 Abelian cubes and Mäkelä's Question 3.2

In this section, we show that the answer to Problem 3.2 is negative: there is no infinite word over a binary alphabet avoiding abelian cubes of period at least 2. The usual method to show that a prefix-closed language is finite is to enumerate it by an exhaustive search. Algorithm 1 computes the elements of a set L, if L is a finite prefix-closed language and ISINL is a function that checks if a word is in L.

```
FIND-RIGHT-EXTENSIONS(w)
    for a ∈ A do
        if ISINL(wa) then
            FIND-RIGHT-EXTENSIONS (wa);
    return true;
```

Algorithm 1: Exhaustive computation of L.

For instance, the exhaustive search from Fig. 1.1 shows that abelian squares are not avoidable over the ternary alphabet and corresponds to the execution of Algorithm 1 over the corresponding language. Unfortunately, we cannot run this algorithm, in a reasonable time, over the language of the words avoiding abelian cubes of period at least 2. This language is so large that it would take too much time to do this search on a modern computer. (It is hard to say whether it would take years, centuries or more since we do not know the actual size of the language. But after a few days of computation it seemed that only a tiny fraction of the language had been found.) Thus we need to restrict our search to a smaller language.

A word $w \in \Sigma^*$ is a *Lyndon word* if for all $u, v \in \Sigma^+$ such that $w = uv$, $w <_{lex} vu$, where $<_{lex}$ is the lexicographic order. The Chen-Fox-Lyndon Theorem states that every word can be written uniquely as a concatenation

of non-increasing Lyndon words (see for example [37]). In the following, we refer to this decomposition as the *Lyndon factorization*.

Proposition 3.3. *Any factor-closed language L with arbitrarily long words contains arbitrarily long Lyndon words or repetitions of arbitrarily large exponent.*

Proof. Let us assume that there are no arbitrarily long Lyndon words in L. This implies that there is a finite number n of Lyndon words in L and there is an integer $s \in \mathbb{N}$ such that for every Lyndon word w in L, $|w| \leq s$. Let $w_1, \ldots, w_n \in L$ be the Lyndon words of L ordered by decreasing lexicographic order.

Then using the Lyndon factorization for every $w \in L$ there are some Lyndon words $L_1 \geq_{lex} L_2 \geq_{lex} \ldots \geq_{lex} L_d$ such that $w = L_1 \ldots L_d$. Since L is factor-closed, all the L_i are in L. We get that for every $w \in L$, there are $\alpha_1, \ldots, \alpha_n \in \mathbb{N}$ such that $w = w_1^{\alpha_1} \ldots w_n^{\alpha_n}$. Then $|w| = \sum_i |w_i| \times \alpha_i \leq s \times \sum_i \alpha_i \leq s \times n \times \max_{\mathrm{i}}(\alpha_i)$. So any word w contains at least a repetition of exponent at least $\frac{|w|}{sn}$.

Moreover L contains arbitrarily long words and sn is a constant depending only on L, so L contains repetitions of arbitrarily large exponent. □

The language of the words avoiding abelian cubes of period more than 2 avoids long repetitions and is factor-closed. Moreover, if a language is prefix-closed and contains arbitrarily long Lyndon words it contains arbitrarily long prefixes of Lyndon words. We can then deduce the following Lemma:

Lemma 3.4. *The set L of words avoiding abelian cubes of period more than 2 contains arbitrarily long words if and only if L contains arbitrarily long prefixes of Lyndon words.*

The language of Lyndon word prefixes avoiding abelian cubes is a prefix-closed language. Thus we can use algorithm 1 to show that this set is finite. For instance, Figure 3.1 shows how it shortens the exhaustive search of binary words avoiding abelian squares of period at least 2.

Using this we can give a negative answer to Problem 3.2:

Theorem 3.5. *There is no infinite binary word avoiding abelian cubes of period at least 2.*

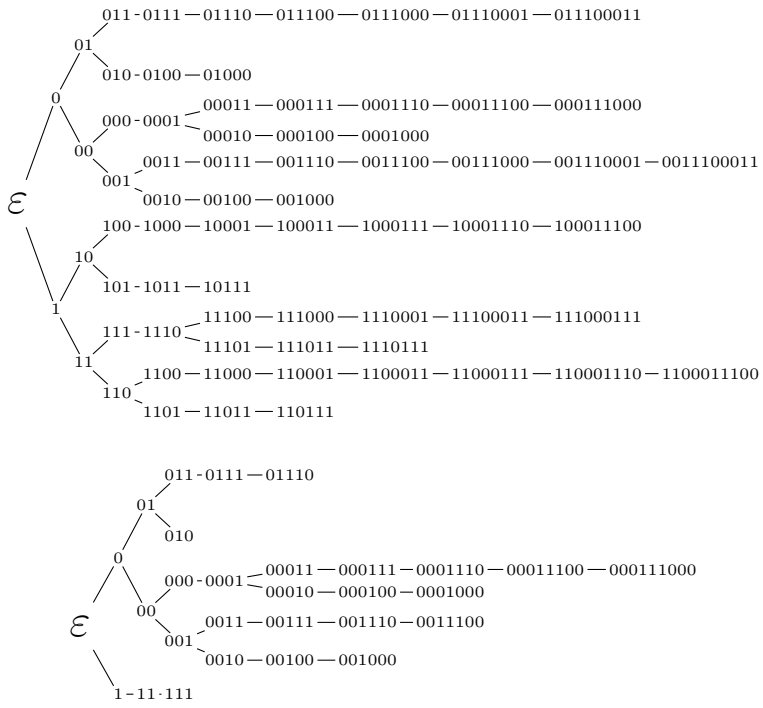

Figure 3.1 – Top: the exhaustive search of binary words avoiding abelian squares of period at least 2. Bottom: the same exhaustive search restricted to the prefixes of Lyndon words.

We checked using a computer program that there are only finitely many binary Lyndon words avoiding abelian cubes of period at least 2. The program takes approximately 3 hours to find all 2 732 711 352 such prefixes of Lyndon words. The longest word has a length of 290. Using Lemma 3.4 we deduce that there is no infinite binary word avoiding abelian cubes of period at least 2.

We can reformulate the question and ask more generally:

Problem 3.6. *Is there a $p \in \mathbb{N}$ such that one can avoid abelian cubes of period at least p over two letters ?*

For $p = 3$, we found a word of length 2 500. Theorem 3.5 also motivates us to study a similar weakening of the Problem 3.1:

Problem 3.7. *Is there a $p \in \mathbb{N}$ such that one can avoid abelian squares of period more than p over three letters ?*

We show in Section 3.3 that the answer is yes for $p = 5$.

3.2 Deciding if a morphic word contains large abelian powers

In this section, we use the results and definitions from Chapter 1 to show that under some conditions one can decide whether a morphic word avoids long abelian repetitions.

Proposition 3.8. *Let $h : \mathcal{A}^* \mapsto \mathcal{A}^*$ and $g : \mathcal{A}^* \mapsto \mathcal{A}'^*$ be two morphisms and M_h and M_g be the matrix associated to those morphisms. If M_h has no eigenvalue of absolute value 1 and $E_e(M_h) \cap \ker(M_g) = \{\vec{0}\}$, then for any template t' one can compute a finite set S that contains any template realizable by h and parent of t' by g.*

Proof. The proof is similar to the computation of parents in Section 1.3. Let $M_h = PJP^{-1}$ be a Jordan decomposition of M_h. Let $\kappa = \dim \ker(M_g)$ and $\Lambda = \ker(M_g) \cap \mathbb{Z}^\kappa$. We use the Smith decomposition of M_g to get the matrix B, whose columns form an integral basis of Λ. Assume $t = [a_1, \ldots, a_{k+1}, \mathbf{d}_1, \ldots, \mathbf{d}_{k-1}]$ is realizable by h and parent of $t' = [a'_1, \ldots, a'_{k+1}, \mathbf{d'}_1, \ldots, \mathbf{d'}_{k-1}]$ by g. Then there are $p_1, s_1, \ldots, p_{k+1}, s_{k+1} \in \mathcal{A}^*$ such that:

— $\forall i,\ g(a_i) = p_i a'_i s_i$

— $\forall i,\ \mathbf{d'}_i = M_g \mathbf{d}_i + \Psi(s_{i+1}p_{i+2}) - \Psi(s_i p_{i+1})$.

There are finitely many choices for the a_i, s_i and p_i. We need to be able to compute all the possible values for $\mathbf{d}_m$ for some m with fixed $a_1, \ldots, a_{k+1}$ and $p_1, s_1, \ldots, p_{k+1}, s_{k+1}$. Then $\mathbf{d}_m$ is an integer solution of $M_g\mathbf{x} = \mathbf{v}$, with $\mathbf{v} = \mathbf{d'}_m + \Psi(s_m p_{m+1}) - \Psi(s_{m+1}p_{m+2})$. We will see that we have only finitely many choices for $\mathbf{d}_m$. As already explained in Section 1.2, if such a solution exists, then $\mathbf{d}_m \in \mathbf{x_0} + \Lambda$, and $\mathbf{x_0}$ can be found with the Smith decomposition of M_g.

Let Q be the rectangular submatrix of P^{-1} such that the ith line of P^{-1} is a line of Q if and only if $|\lambda_{b(i)}| < 1$. For every $\mathbf{x} \in \mathbb{C}^\kappa \setminus \{\vec{0}\}$, $B\mathbf{x} \in \ker(M_g)$ by definition of B. Then, by hypothesis, $B\mathbf{x} \notin E_e(M_h)$ and $QB\mathbf{x} \neq \overrightarrow{0}$ since the lines of Q generate the subspace orthogonal to $E_e(M_h)$. Thus we have $\mathrm{rank}(QB) = \kappa$ which implies that there is a submatrix Q' of Q such that $Q'B$ is invertible.

From Proposition 1.16, for all $i \in \{1, \ldots, \kappa\}$, there is $c_i \in \mathbb{R}$ such that for any two factors u and v of $\mathrm{Fact}^\infty(h)$, $|(Q'(\Psi(u) - \Psi(v)))_i| \leq c_i$.

Let $\mathbf{X} = \{\mathbf{x} \in \mathbf{x_0} + \Lambda : \forall i \in \{1, \ldots, \kappa\},\ |(Q'\mathbf{x})_i| \leq c_i\}$. Since we are only interested in realizable solutions, $\mathbf{d}_m$ has to be in $\mathbf{X}$. Let $\mathbf{X}_B = \{\mathbf{x} \in \mathbb{Z}^\kappa : (\mathbf{x_0} + B\mathbf{x}) \in \mathbf{X}\}$ and $\mathbf{x} \in \mathbf{X}_B$. Then for all i, $|(Q'(B\mathbf{x} + \mathbf{x_0}))_i| \leq c_i$ thus $|(Q'(B\mathbf{x}))_i| \leq c_i + |(Q'\mathbf{x_0})_i|$. Then $||Q'B\mathbf{x}||^2 \leq \sum_{i=1}^{l}(c_i + |(Q'\mathbf{x_0})_i|)^2 = c$. From Proposition 1.7, if μ_{min} is the smallest eigenvalue of $(Q'B)^*(Q'B)$, we have $\mu_{min}||\mathbf{x}||^2 \leq ||Q'B\mathbf{x}||^2 \leq c$. Since $Q'B$ is invertible, $\mu_{min} \neq 0$ and $||\mathbf{x}|| \leq \sqrt{\frac{c}{\mu_{min}}}$. Then $\mathbf{X}_B$ and $\mathbf{X}$ are finite, and we can easily compute them. □

We can easily adapt the proof of Lemma 1.12 to get:

Proposition 3.9. *If no parent of the k-template $[\varepsilon, \ldots, \varepsilon, \overrightarrow{0}, \ldots, \overrightarrow{0}]$ by g is realizable by h then $g(\mathrm{Fact}^\infty(h))$ avoids abelian k-th powers of period larger than $\max_{a\in\mathcal{A}} |g(a)|$.*

The condition of Proposition 3.9 can be easily checked by a computer using Proposition 3.8 and Theorem 1.9. If one wants to decide whether $g(\mathrm{Fact}^\infty(h))$ avoids abelian k-th powers of period at least $p \leq \max_{a\in\mathcal{A}} |g(a)|$, then one can use Proposition 3.9 and check if $g(\mathrm{Fact}^\infty(h))$ does not contain an abelian k-th power of period l for every $p \leq l < \max_{a\in\mathcal{A}} |g(a)|$. If $p > \max_{a\in\mathcal{A}} |g(a)|$, then one can take a large enough integer k such that $p \leq \max_{a\in\mathcal{A}} |g(h^k(a))|$, and do the computation on $g \circ h^k$ instead of g. Note that if $E_e(M_h) \cap \ker(M_g) = \{\overrightarrow{0}\}$, then for every $k \in \mathbb{N}$, $E_e(M_h) \cap \ker(M_{g\circ h^k}) = \{\overrightarrow{0}\}$. Otherwise, for the sake of contradiction let $\mathbf{x} \in (E_e(M_h) \cap \ker(M_{g\circ h^k})) \setminus \{\overrightarrow{0}\}$. Then $M_h^k\mathbf{x} \in \ker(M_g)$. Moreover $\mathbf{x} \in E_e(M_h) \setminus \{\overrightarrow{0}\}$, so $M_h^k\mathbf{x} \in E_e(M_h)$ and $M_h^k\mathbf{x} \neq \overrightarrow{0}$. Thus $M_h^k\mathbf{x} \in E_e(M_h) \cap \ker(M_g) \setminus \{\overrightarrow{0}\}$, and we have a contradiction.

Consequently we have the following theorem.

Theorem 3.10. *Let $h : \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism with no eigenvalue of absolute value 1, let $g : \mathcal{A}^* \to \mathcal{A}'^*$ be a morphism, and let $p, k \in \mathbb{N}$. If $E_e(M_h) \cap \ker(M_g) = \{\overrightarrow{0}\}$ then one can decide whether $g(h^\infty(a))$ avoids abelian k-th powers of period larger than p.*

In Section 3.3, we present a morphic word over 3 letters which avoids abelian squares of period more than 5.

3.3 Results

We use the algorithm described in Section 3.2 to give an answer to Problem 3.7. We also show that this result implies that additive squares are avoidable over $\mathbb{Z}^2$, which gives another proof of Theorem 2.5 from Chapter 2. We provide a computer program [1] that applies the algorithm described in the previous section in order to show Theorem 3.11.

3.3.1 Mäkelä's Problem on squares

Let h_6 be the morphism that we already used in Section 1.4:

$$h_6 : \begin{cases} a \to ace & b \to adf \\ c \to bdf & d \to bdc \\ e \to afe & f \to bce. \end{cases}$$

Let g_3 be the following morphism:

$$g_3 : \begin{cases} a \to \texttt{bbbaabaaac} \\ b \to \texttt{bccacccbcc} \\ c \to \texttt{ccccbbbcbc} \\ d \to \texttt{ccccccccaa} \\ e \to \texttt{bbbbbcabaa} \\ f \to \texttt{aaaaaaabaa}. \end{cases}$$

Theorem 3.11. *The word obtained by applying g_3 to the fixed point of h_6, that is $g_3(h_6^\omega(a))$, avoids abelian squares of period more than 5.*

The kernel of g_3 is of dimension 3, but using the bounds on the 3 null eigenvalues of h_6 we can compute that $[\varepsilon, \ldots, \varepsilon, \overrightarrow{0}, \ldots, \overrightarrow{0}]$ has at most 16214 parents by g_3 realizable by h_6. This is checked using Theorem 3.10. This gives an answer to a weak version of Problem 3.1.

Theorem 3.12. *There is an infinite word over 3 letters avoiding abelian squares of period more than 5.*

The optimal value for this result is probably not 5, so we ask the following question:

1. `https://arxiv.org/abs/1511.05875`

Problem 3.13. *What is the smallest $p \in \mathbb{N}$ such that one can avoid abelian squares of period more than p over three letters ?*

The proof technique presented here could be helpful to solve this problem. Note that we know that $2 \leq p \leq 5$.

In fact, $g_3(h_6^\omega(a))$ contains 34 different abelian squares. We could also ask to minimize the number of different abelian squares.

3.3.2 Link between long abelian powers and additive powers

There is a strong relation between avoidability of long abelian powers and additive powers over $\mathbb{Z}^d$. For instance, the next theorem shows that Theorem 2.5 from Chapter 2 is in fact a corollary of Theorem 3.12.

Theorem 3.14. *For any alphabet $\mathcal{A}$ and integer k, if long abelian k-th powers are avoidable over $\mathcal{A}$ then additive k-th powers are avoidable over a finite subset of $\mathbb{Z}^{|\mathcal{A}|-1}$.*

Proof. Let $f : \mathbb{Z}^{|\mathcal{A}|} \mapsto \mathbb{Z}^{|\mathcal{A}|-1}$ be the projection that takes a vector and forgets the last coordinate. That is f is the left multiplication by the $|\mathcal{A}| \times (|\mathcal{A}|-1)$ matrix:

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}.$$

Then for any $w \in \mathcal{A}^*$ and i, $\Psi(w)_i = \begin{cases} f(\Psi(w))_i & \text{if } i \in [1, |\mathcal{A}|-1] \\ |w| - ||f(\Psi(w))||_1 & \text{if } i = |\mathcal{A}| \end{cases}$

So for any two words $u, v \in \mathcal{A}^*$, $\Psi(u) = \Psi(v)$ if and only if $|u| = |v|$ and $f(\Psi(u)) = f(\Psi(v))$.

Now let $w \in \mathcal{A}^*$ be a word avoiding abelian k-th powers of period at least p. Let $s = (s_i)_{i \in \mathbb{N}}$ be the sequence such that for all $i \in \mathbb{N}$, $s_i = f(\Psi(w_{ip} w_{ip+1} \dots w_{(i+1)p-1}))$. Assume for the sake of contradiction that this sequence contains an additive k-th power. Then there are $u_1, u_2, \dots, u_k$ such that $u_1 u_2 \dots u_k$ is a factor of w, and for all i, j, $|u_j| = |u_i| \geq p$ and $f(\Psi(u_j)) = f(\Psi(u_i))$. This implies that for all i, j, $u_i \approx_a u_j$ and $|u_j| = |u_i| \geq p$. So for any additive k-th power in s there is an abelian k-th power of period at least p in w.

Thus s avoids k-th additive powers. Moreover for all i, $||s_i||_1 \leq p$, so there are finitely many different elements in s. We can deduce that k-th powers are avoidable over a finite subset of $\mathbb{Z}^{|\mathcal{A}|-1}$. □

This theorem by itself cannot be used to show anything new since $\mathbb{Z}^2$ is not 2-repetitive, $\mathbb{Z}$ is not 3-repetitive and that long abelian squares are not avoidable over the binary alphabet. But the proof do not use any of these results and is constructive.

From Theorem 3.12, we know that additive squares are avoidable over $\mathbb{Z}^2$. From the construction from Keränen (Theorem 1.1, [34]), we also get the following corollary:

Corollary 3.15. *There is an infinite additive-square-free word over the alphabet* $\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right\}$.

Since abelian squares are not avoidable over 3 letters, the alphabet size is clearly optimal in this result.

Chapter 4

Avoidability of k-Abelian Powers

The notion of k-abelian repetition was introduced recently by Karhumäki *et al.* as a generalization of both repetitions and abelian repetitions [33, 31]. For $u, w \in \Sigma^*$, we denote by $|u|_w = |\{i : u_i u_{i+1} \dots u_{i+|w|-1} = w\}|$ the number of occurrences of the factor w in u. Two words u and v are *k-abelian equivalent* (for $k \geq 1$), denoted $u \approx_{a,k} v$, if for every $w \in \Sigma^*$ such that $|w| \leq k$, $|u|_w = |v|_w$. Note that 1-abelian equivalence is exactly abelian equivalence and that as k goes to infinity k-abelian equivalence tends toward equality.

A word $u_1 u_2 \dots u_n$ is a *k-abelian n-th power* if it is non-empty, and $u_1 \approx_{a,k} u_2 \approx_{a,k} \dots \approx_{a,k} u_n$. Its *period* is $|u_1|$. A *k-abelian square* (resp. *k-abelian cube*) is a k-abelian 2-nd power (resp. k-abelian 3-rd power). For instance, *abacbacbab* is a 2-abelian square, but not a 3-abelian square.

By definition any $(k+1)$-abelian n-th power is a k-abelian n-th power, and any n-th power is a k-abelian n-th power. By combining that with results about avoidability of usual powers and abelian powers we can deduce some results about k-abelian powers avoidability. For every k:

— over 4 letters or more k-abelian squares are avoidable, since abelian squares are avoidable [34];

— over the ternary alphabet k-abelian cubes are avoidable, since abelian cubes are avoidable [22];

— over the binary alphabet k-abelian 4-th powers are avoidable, since abelian 4-th powers are avoidable [22], but k-abelian squares are not avoidable since usual squares are not avoidable.

It is then natural to ask what is the smallest value of k such that k-abelian

cubes are avoidable over the binary alphabet. It was first proven that $k \leq 8$ [31], then that $k \leq 5$ [40], $k \leq 3$ [39] and finally $k = 2$ [53].

Likewise we can ask what is the smallest k such that k-abelian squares are avoidable over the ternary alphabet. It was proven that $k \geq 3$ [30], then that $k \leq 64$ [29] and finally that $k = 3$ [53].

Entringer, Jackson and Schatz answered a question from Erdős by proving that it is possible to avoid arbitrarily long ordinary squares on binary words, but not arbitrarily long abelian squares. In fact, Fraenkel and Simpson showed that there is an infinite binary word containing only the squares 0^2, 1^2, $(01)^2$ [27] and this result is optimal since every infinite binary word contains at least 3 different squares. It is natural to ask whether this property can be extended to the k-abelian case:

Problem 4.1 (Rao, [53])**.** *What is the smallest k (if any) such that arbitrarily long k-abelian squares can be avoided over a binary alphabet ?*

More generally let $g(k)$ be the minimal number of distinct k-abelian squares that an infinite binary word must contain.

In Subsection 4.2.1, we show that $g(3) = g(4) = 4$ and that $g(k) = 3$ for every $k \geq 5$. Then using a more complicated construction based on Theorem 3.11, we show that $5 \leq g(2) \leq 734$. In Subsection 4.2.2, we show that there is a ternary word containing only the 2-abelian square 22. This result is optimal since 2-abelian squares are not avoidable over the ternary alphabet [30]. This last result is a new result, while the other results presented in this chapter are joint work with Michaël Rao and are part of the articles [54] and [55].

4.1 Proving that a morphic word avoids long k-abelian squares

A word is (l, k)-*abelian n-th power-free* if it does not contain any k-abelian n-th power of period at least l. A morphism is said to be (l, k)-*abelian n-th power-free* if the image of every abelian n-th power-free word by h is (l, k)-abelian n-th power-free. Carpi gave a set of sufficient conditions for a morphism to be abelian n-th power-free (that is $(1, 1)$-abelian n-th power-free) [10]. Rao generalized this result in order to obtain sufficient conditions for a morphism to be k-abelian n-th power-free (that is $(1, k)$-abelian n-th

power-free) [53]. Here we adapt the result and the proof of Rao in order to get sufficient conditions for a morphism to be (l,k)-abelian n-th power-free.

We need to introduce the following notions before the statement of the theorem. For a set $S \subset \Sigma^*$ and a word $w \in \Sigma^*$, we denote by $\Psi_S(w)$ the vector indexed by S such that for every $s \in S$, $\Psi_S(w)[s] = |w|_s$. We may write $\Psi_k(w)$ instead of $\Psi_{\Sigma^k}(w)$ if Σ is clear in the context.

For any matrix M, we denote by $\mathrm{Im}(M)$ the image of the matrix M.

For all $u \in \Sigma^*$, $i \leq |u|$, let $\mathrm{pref}_i(u)$ be the prefix of size i of u and $\mathrm{suff}_i(u)$ be the suffix of size i of u. There are equivalent definitions of k-abelian equivalence (see [33]). Two words of size at most $2k-1$ are k-abelian equivalent if and only if they are equal. For every two words u and v of size at least $k-1$, the following conditions are equivalent:

— u and v are k-abelian equivalent (*i.e.* $u \approx_{a,k} v$),

— $\Psi_k(u) = \Psi_k(v)$ and $\mathrm{pref}_{k-1}(u) = \mathrm{pref}_{k-1}(v)$,

— $\Psi_k(u) = \Psi_k(v)$ and $\mathrm{suff}_{k-1}(u) = \mathrm{suff}_{k-1}(v)$.

Theorem 4.2. *Let $k \geq 1$, $n \geq 2$, $l \geq 1$, two alphabets A and B and a morphism $h : A^* \mapsto B^*$. Suppose that:*

1. *Let $d = \max(k, \max_{a\in A} |h(a)|)$. For every abelian n-th power-free word $w \in A^*$ such that $|h(w[2..|w|-1])| \leq dn$, $h(w)$ is (l,k)-abelian n-th power-free.*
2. *There are $p, s \in B^{k-1}$ such that for every $a \in A$, $p = \mathrm{pref}_{k-1}(h(a)p)$ and $s = \mathrm{suff}_{k-1}(sh(a))$.*
3. *The matrix N indexed by $B^k \times A$, with $N[w,x] = |h(x)p|_w$ has rank $|A|$. Let $S \subseteq B$ be such that the matrix M indexed by $S \times A$, with $M[w,x] = |h(x)p|_w$ is invertible. Let $\Psi_S(v,u) = \Psi_S(vp) + \Psi_S(su) - \Psi_S(sp)$. For every $(a_0, \ldots, a_n) \in A$ and $u_i, v_i \in A^*$ with $h(a_i) = u_i v_i, \forall i \in [0,n]$ such that:*
   (a) $\forall i \in [1, n-1]$, $\mathrm{pref}_{k-1}(v_{i-1}p) = \mathrm{pref}_{k-1}(v_i p)$,
   (b) $\forall i \in [1, n-1]$, $M^{-1}(\Psi_S(v_{i-1}, u_i) - \Psi_S(v_i, u_{i+1}))$ *is an integer vector,*
   (c) $\forall i \in [1, n-1]$, $\Psi_k(v_{i-1}, u_i) - \Psi_k(v_i, u_{i+1}) \in \mathrm{Im}(N)$.

   There are $\alpha_0, \ldots, \alpha_n \in \{0,1\}$ such that for every $i \in [1, n-1]$:

$$M^{-1}(\Psi_S(v_{i-1}, u_i) - \Psi_S(v_i, u_{i+1})) = \\ (1-\alpha_{i-1})\Psi(a_{i-1}) + (2\alpha_i - 1)\Psi(a_i) - \alpha_{i+1}\Psi(a_{i+1}).$$

Then h is (l,k)-abelian n-th power-free.

Proof. For the sake of contradiction let us assume that h is not (l,k)-abelian n-th power-free. There is $w \in A^*$ such that $h(w)$ contains an (l,k)-abelian n-th power and w contains no abelian n-th power. There are $q_0, q_1, \ldots, q_n, q_{n+1} \in B^*$ such that $h(w) = q_0 q_1 \ldots q_n q_{n+1}$ and for all $i \in [1,n]$, $q_i \approx_{a,k} q_1$ and $|q_i| \geq l$.

From (1), we know that for all $i \in [1,n]$, $|q_i| \geq d \geq \max_{a \in A} |h(a)|$. Thus there are $r_0, r_1, \ldots, r_{n+1} \in A^*$, $a_0, a_1, \ldots, a_n \in A$ and $u_0, v_0, u_1, v_1 \ldots, u_n, v_n \in B^*$ such that:

— $w = r_0 a_0 r_1 a_1 \ldots a_n r_{n+1}$,
— for all i, $h(a_i) = u_i v_i$,
— for all $i \in [1,n]$, $q_i = v_i h(r_i) u_i$.

For all $i \in [1,n]$, $\Psi_k(q_i) = \Psi_k(v_{i-1} h(r_i) u_i) = \Psi_k(v_{i-1} p) + \Psi_k(h(r_i) p) + \Psi_k(s u_i) - \Psi_k(sp) = \Psi_k(v_{i-1}, u_i) + N\Psi(r_i)$. Moreover for all $i \in [1, n-1]$, $\Psi_k(q_i) = \Psi_k(q_{i+1})$ and thus

$$\Psi_k(v_{i-1}, u_i) - \Psi_k(v_i, u_{i+1}) = N(\Psi(r_{i+1}) - \Psi(r_i)) \tag{4.1}$$

$$M^{-1}(\Psi_S(v_{i-1}, u_i) - \Psi_S(v_i, u_{i+1})) = \Psi(r_{i+1}) - \Psi(r_i) \tag{4.2}$$

Equation 4.2 implies condition (b) and Equation 4.1 implies condition (c). Moreover for all $i \in [1, n-1]$, $\mathrm{pref}_{k-1}(v_{i-1} h(r_i) u_i) = \mathrm{pref}_{k-1}(v_i h(r_{i+1}) u_{i+1})$ which implies condition (a).

Since we have all the conditions of (3.), there are $\alpha_0, \ldots, \alpha_n \in \{0,1\}$ such that for every $i \in [1, n-1]$:

$$M^{-1}(\Psi_S(v_{i-1},u_i) - \Psi_S(v_i,u_{i+1})) = (1-\alpha_{i-1})\Psi(a_{i-1}) + (2\alpha_i - 1)\Psi(a_i) - \alpha_{i+1}\Psi(a_{i+1}) \tag{4.3}$$

For all $i \in [1,n]$, let $r'_i = a_{i-1}^{1-\alpha_{i-1}} r_i a_i^{\alpha_i}$, then clearly $r'_1 r'_2 \ldots r'_n$ is a factor of w. Moreover for all $i \in [1, n-1]$:

$$\begin{aligned}
\Psi(r'_i) - \Psi(r'_{i+1}) &= (1-\alpha_{i-1})\Psi(a_{i-1}) + \Psi(r_i) + (2\alpha_i - 1)\Psi(a_i) - \Psi(r_{i+1}) - \alpha_{i+1}\Psi(a_{i+1}) \\
&= M^{-1}(\Psi_S(v_{i-1},u_i) - \Psi_S(v_i,u_{i+1})) + \Psi(r_i) - \Psi(r_{i+1}) \text{ (From 4.3)} \\
&= 0 \text{ (From 4.2)}
\end{aligned}$$

Thus w contains an abelian k-th power and we get a contradiction. It concludes the proof that h is (l,k)-abelian n-th power-free. □

For any given morphism all the conditions of Theorem 4.2 are easy to check. We wrote a C++ code that checks all the conditions, and we can use it to check whether a morphism is (l,k)-abelian n-th power-free.

4.2 Results

4.2.1 Minimum number of distinct k-abelian squares in binary words

We want to compute for all k, $g(k)$ the minimal number of distinct k-abelian squares that an infinite binary word must contain. Any $(k+1)$-abelian square is a k-abelian square so g is non-increasing. Every binary infinite word contains at least three distinct squares thus for all k, $g(k) \geq 3$ [27].

Proposition 4.3. *The morphism h_3 (defined in Table 4.1) is $(3,5)$-abelian square-free. Moreover, for every abelian square-free word w, $h_3(w)$ contains only 3 distinct 5-abelian squares: 0^2, 1^2 and $(01)^2$.*

The conditions from Theorem 4.2 can be verified on h_3 by a computer program. Then one has to check the small 5-abelian squares that could occur. Since abelian squares are avoidable over 4 letters, $g(k) \leq 3$ for every $k \geq 5$, and so $g(k) = 3$. Note that this result implies the result from Fraenkel and Simpson. Propositions 4.4 and 4.5 give us that $g(3) = g(4) = 4$.

Proposition 4.4. *Every word of size more than 87 over the binary alphabet contains at least 4 different 4-abelian squares.*

This can be verified by an exhaustive computer search using the technique presented in Section 3.1.

Proposition 4.5. *Let:*

$$h_4 : \begin{cases} 0 \to & 000110010100110101100010101000101111010110000101 \\ 1 \to & 000110010100110101100111010101110001110101100010 1 \\ 2 \to & 000110010100111000101000110010110001110101100010 1 \\ 3 \to & 000110010100111001010100111000101100101011000101. \end{cases}$$

Then h_4 is $(3,3)$-abelian square-free. Moreover, for every abelian square-free word w, $h_4(w)$ contains only 4 distinct 3-abelian squares: 0^2, 1^2, $(01)^2$ and $(10)^2$.

The proof that h_4 is $(3,3)$-abelian square-free can be done by checking the conditions from Theorem 4.2. One can then check that $(00)^2$ and $(11)^2$ do not appear as factors of any image of a two-letter word.

Using again an exhaustive search and the technique from Section 3.1, we can give the lower bound $g(2) \geq 5$.

$$
h_3 : \begin{cases}
0 \to & u1001011000101110001100101100010111001011000111001011100011001011000 \\
& 1011100101100111000101100011100101110001100101100010111001011000111 \\
& 0101110001100101100011100101100111000110010110001110010111000110010 \\
& 1000101110010110001110010111000110010110001110010110011100010110011 \\
& 1001011000101110010110011100010111001011000101110001100101100011100 \\
& 1 \\
& 0110011100010111001011000111001011100011v \\
1 \to & u0001110010110011100011001011000111001011001110001011100101100011100 \\
& 1011100011001011000111001011001110001100101100011100101110001011000 \\
& 1100101100111000110010110001110010110011100010110001110010111000101111 \\
& 0001100101100011100101100111000110010110001110010111000110010110001010 \\
& 1110010110001110010111000110010110001110010110011100011001011000111 \\
& 0 \\
& 1011100011001011000101110010110011100010110001110010110001011100101011 \\
2 \to & u0001110010110011100011001011000111001011001110001011100101100011100 \\
& 1011100011001011000101110010110011100010110001110010110001011100101011 \\
& 0011100010111001011000111001011100011001011000101110010110011100010 \\
& 1100101100010111000110010110001011100101100111000101100011100101100 \\
& 1100011001011000111001011100011001011000101110010110011100010111001010 \\
& 1100010111000110010110001011100101100111v \\
3 \to & u0001110010110001011100011001011000101110010110001110010111000110010 \\
& 1100011100101100111000110010110001110010111000101100011100101100111 \\
& 0011001011000111001011001110001011000111001011000101110001100101100 \\
& 1011100101100111000101100011100101110001011001110001100101100011100 \\
& 1 \\
& 0110011100010110001110010110001011100101100111000101110010110011100 \\
& 1011100011001011000101110010110011100v
\end{cases}
$$

Where:

$$
\begin{aligned}
u = \; & 1100011001011000101110010110011100010110001110010110 \\
& 0101110010110011100010111001011000101110001100101100011 \\
& 1110010110011100011001011000101110010110011100010111 \\
v = \; & 0010110001110010110011100011001011100010110011100010111 \\
& 1100101100010111000110010111000101100111000110010110 \\
& 0111001011100010110001110010110001110001011000111001011
\end{aligned}
$$

Table 4.1 – A $(3, 5)$-abelian square-free morphism, with only three distinct 5-abelian squares: 00, 11 and 0101.

Proposition 4.6. *Every word of size more than 92 over the binary alphabet contains at least 5 distinct 2-abelian squares.*

For the upper bound on $g(2)$ we have:

Theorem 4.7. *Let:*

$$h_2 : \begin{cases} a \to & 111111111000 \\ b \to & 101011110100 \\ c \to & 101011000000. \end{cases}$$

h_2 is $(8,2)$-abelian square-free.

h_2 fulfills the conditions from Theorem 4.2 for $l = 8$, $k = 2$. This morphism is less obvious to use since there is no abelian square free ternary word. But recall that we showed in Chapter 3 that there is a ternary word $w = g_3(h_6^\omega(a))$ that avoids abelian squares of length more than 5. By looking at the proof of Theorem 4.2, we also get that 2-abelian squares in $h_2(w)$ that are greater than 8 are factor of the image of an abelian square concatenated with at most 2 letters. Thus there are no 2-abelian square in $h_2(w)$ of period more than $12 \times 7 = 84$.

We can check every factors of $h_2(w)$ of length at most 168 and obtain the exact list of 2-abelian squares. We get that there are only 734 different 2-abelian squares in $h_2(g_3(h_6^\omega(a)))$, all of them of period at most 63. We deduce the following Corollary.

Corollary 4.8. $g(2) \leq 734$.

This Corollary answers Problem 4.1 and we know that long 2-abelian squares are avoidable over the binary alphabet.

For every $k \neq 2$, we know the exact minimal number of different k-abelian square in an infinite binary word so we naturally ask the following question:

Problem 4.9. *What is the minimal number of distinct 2-abelian squares that an infinite binary word must contain?*

Note that any improvement on the upper bound in Problem 3.13 will improve the upper bound for this problem. Nonetheless in order to find the exact value of $g(2)$ one probably has to apply directly a morphism to an abelian square free word. Moreover the proof technique from Chapter 1 could be adapted in order to decide k-abelian powers freeness of morphic words under similar conditions.

4.2.2 2-abelian squares over a ternary alphabet

Rao showed that one can build an infinite word that avoids 3-abelian squares over a ternary alphabet [53]. The longest 2-abelian square-free ternary word has a length of 537 [30]. We showed in Chapter 3 that one can avoid abelian squares of period at least 6 over the ternary alphabet. It implies that 2-abelian square of period at least 6 are avoidable over the ternary alphabet, but we can show that one can avoid 2-abelian squares of period at least 2 over the ternary alphabet.

Let:

$$h_2 : \begin{cases} 0 \to & 00021 \\ 1 \to & 00111 \\ 2 \to & 01121 \\ 3 \to & 01221. \end{cases}$$

Theorem 4.10. *h_2 is $(2,2)$-abelian square-free.*

We can apply Theorem 4.2 with $S = \{00, 01, 02, 11\}$.

In fact we can show the following stronger result:

Theorem 4.11. *Let*

$$h_2' : \begin{cases} 0 \to & 20212210201 \\ 1 \to & 20222010201 \\ 2 \to & 21012010201 \\ 3 \to & 22120122201. \end{cases}$$

The only 2-abelian square contained in the image of any abelian square free word by h_2' is 22.

We use Theorem 4.2 to show that h_2' is 2-abelian square free, and clearly 00 and 11 do not appear on any image by this morphism. Since 2-abelian squares are not avoidable over the ternary aphabet this result is optimal in number of different squares.

Since there are exponentially many abelian square free words over 4 letters (see [11, 36]), we get the following corollary:

Corollary 4.12. *There are exponentially many ternary words that contain at most one 2-abelian square.*

Chapter 5

Avoidability of Binary Formulas

A *pattern* p is a non-empty finite word over an alphabet Δ whose letters are called *variables*. Let $\Delta = \{A, B, \ldots\}$, that is we denote elements of Δ by capital letters. An *occurrence* of a pattern p in a word w is a non-erasing morphism $h : \Delta^* \to \Sigma^*$ such that $h(p)$ is a factor of w. The *avoidability index* $\lambda(p)$ of a pattern p is the size of the smallest alphabet Σ such that there exists an infinite word over Σ avoiding occurrences of p, or ∞ if there is no such finite Σ. We say that a pattern p is *k-avoidable* if $\lambda(p) \leq k$. In fact, the original question that motivated Thue's work was whether or not for any finite word p and infinite word w there is a non-erasing morphism h such that $h(p)$ is a factor of w. In our terminology, he asked whether they are avoidable patterns and he showed that AAA is avoidable over 2 letters, that is $\lambda(AAA) = 2$ [59].

Since the work of Thue on the avoidability of patterns many authors worked on the classification of avoidable patterns [13, 44, 58]. Bean, Ehrenfeucht, and McNulty [6] and Zimin [61] characterized unavoidable patterns, i.e., such that $\lambda(p) = \infty$. Roth proved in [57] that binary patterns of length greater than 6 are avoidable over the binary alphabet. Cassaigne [14] began and Ochem [42] finished the determination of the avoidability index of every pattern with at most 3 variables.

A variable that appears only once in a pattern is said to be *isolated*. Following Cassaigne [14], we associate to a pattern p the *formula* f obtained by replacing every isolated variable in p by a dot. The factors between the dots are called *fragments*.

An *occurrence* of f in a word w is a non-erasing morphism $h : \Delta^* \to \Sigma^*$ such that the h-image of every fragment of f is a factor of w. The avoidability

index $\lambda(f)$ of a formula f is the size of the smallest alphabet over which there is an infinite word containing no occurrence of f. Clearly, every word avoiding f also avoids p, so $\lambda(p) \leq \lambda(f)$. An infinite word is *recurrent* if every finite factor appears infinitely many times. It is a classical result that if there exists an infinite word over Σ avoiding p, then there exists an infinite recurrent word over Σ avoiding p. This recurrent word also avoids f, so that $\lambda(p) = \lambda(f)$. Without loss of generality, a formula is such that no variable is isolated and no fragment is a factor of another fragment. A formula is said to be *binary* if it has at most 2 variables. We say that an avoidable formula f is *exponential* if there are $a > 0$ and $b > 1$ such that the number of words in $\Sigma^n_{\lambda(f)}$ avoiding f is lower bounded by $a.b^n$. Similarly, an avoidable formula f is *polynomial* if there is a polynomial P such that the number of words in $\Sigma^n_{\lambda(f)}$ avoiding f is upper bounded by $P(n)$.

In this chapter, we determine the avoidability index of every binary formula. Formulas are either unavoidable if they divide ABA, or avoidable over the ternary alphabet so we only need to distinguish between avoidability index 2 and 3. We then determine which formulas are exponential and which are polynomial. We also give a formula that characterize the language of the Thue-Morse word in Section 5.5. All the results presented in this Chapter are joint work with Pascal Ochem. The formula characterizing the Thue-Morse language is new and the rest is part of the article [46].

5.1 Classification of binary formulas

We say that a formula f is *divisible* by a formula f' if f does not avoid f', that is, there is a non-erasing morphism h such that the image of any fragment of f' by h is a factor of a fragment of f. If f is divisible by f', then every word avoiding f' also avoids f and thus $\lambda(f) \leq \lambda(f')$. Moreover, the reverse f^R of a formula f satisfies $\lambda(f^R) = \lambda(f)$. For example, the fact that $ABA.AABB$ is 2-avoidable implies that $ABAABB$ and $BAB.AABB$ are 2-avoidable. See Cassaigne [14] and Clark [16] for more information on formulas and divisibility.

First, we check that every avoidable binary formula is 3-avoidable. Since $\lambda(AA) = 3$, every formula containing a square is 3-avoidable. Then, the only square free avoidable binary formula is $ABA.BAB$ with avoidability index 3 ($ABACBAB$ is of avoidability index 3 [14]). Thus, we have to distinguish between avoidable binary formulas with avoidability index 2 and 3. A binary

formula is minimally 2-avoidable if it is 2-avoidable but not divisible by any non-equivalent 2-avoidable binary formula. A binary formula f is maximally 2-unavoidable if it is 2-unavoidable but every non-equivalent binary formula that is divisible by f is 2-avoidable.

We are now ready to give the two main theorems of this chapter:

Theorem 5.1.

Up to symmetry, the maximally 2-unavoidable binary formulas are:

— $AAB.ABA.ABB.BBA.BAB.BAA$

— $AAB.ABBA$

— $AAB.BBAB$

— $AAB.BBAA$

— $AAB.BABB$

— $AAB.BABAA$

— $ABA.ABBA$

— $AABA.BAAB$

Up to symmetry, the minimally 2-avoidable binary formulas are:

— $AA.ABA.ABBA$ *(polynomial)*

— $ABA.AABB$ *(polynomial)*

— $AABA.ABB.BBA$ *(polynomial)*

— $AA.ABA.BABB$ *(exponential)*

— $AA.ABB.BBAB$ *(exponential)*

— $AA.ABAB.BB$ *(exponential)*

— $AA.ABBA.BAB$ *(exponential)*

— $AAB.ABB.BBAA$ *(exponential)*

— $AAB.ABBA.BAA$ *(exponential)*

— $AABB.ABBA$ *(exponential)*

— $ABAB.BABA$ *(exponential)*

— $AABA.BABA$ *(exponential)*

— AAA *(exponential)*

— $ABA.BAAB.BAB$ *(exponential)*

— $AABA.ABAA.BAB$ *(exponential)*

— $AABA.ABAA.BAAB$ *(exponential)*

— *ABAAB (exponential)*

Given a binary formula f, we can use Theorem 5.1 to find $\lambda(f)$. Now, we also consider the problem whether an avoidable binary formula is polynomial or exponential. If $\lambda(f) = 3$, then either f contains a square or $f = ABA.BAB$, so f is exponential. Thus, we consider only the case $\lambda(f) = 2$. If f is divisible by an exponential 2-avoidable formula given in Theorem 5.1, then f is known to be exponential. This leaves open the cases such that f is only divisible by polynomial 2-avoidable formulas. The next result settles every open case.

Theorem 5.2. *The following formulas are polynomial:*

— *BBA.ABA.AABB*

— *AABA.AABB*

The following formulas are exponential:

— *BAB.ABA.AABB*

— *AAB.ABA.ABBA*

— *BAA.ABA.AABB*

— *BBA.AABA.AABB*

To obtain the 2-unavoidability of the formulas in the first part of Theorem 5.1, we use a standard backtracking algorithm. Figure 5.1 gives the maximal length and number of binary words avoiding each maximally 2-unavoidable formula.

In Section 5.3, we consider the polynomial formulas in Theorems 5.1 and 5.2. The proof uses a technical lemma given in Section 5.2. Then we consider in Section 5.4 the exponential formulas in Theorems 5.1 and 5.2.

5.2 The useful lemma

Let b_3 be the ternary Thue-Morse word, that is the fixed point of $\mathtt{0} \mapsto \mathtt{012}$, $\mathtt{1} \mapsto \mathtt{02}$, $\mathtt{2} \mapsto \mathtt{1}$.

Let w and w' be infinite (right infinite or bi-infinite) words. We say that w and w' are *equivalent* if they have the same set of finite factors. We write $w \sim w'$ if w and w' are equivalent. A famous result of Thue [59] can be stated as follows:

Formula	Maximal length of a binary word avoiding this formula	Number of binary words avoiding this formula
$AAB.BBAA$	22	1428
$AAB.ABA.ABB.BBA.BAB.BAA$	23	810
$AAB.BBAB$	23	1662
$AABA.BAAB$	26	2124
$AAB.ABBA$	30	1684
$AAB.BABAA$	42	71002
$AAB.BABB$	69	9252
$ABA.ABBA$	90	31572

Figure 5.1 – The number and maximal length of binary words avoiding the maximally 2-unavoidable formulas.

Theorem 5.3. *[59] Every bi-infinite ternary word avoiding* `010`*,* `212`*, and squares is equivalent to* b_3*.*

Given an alphabet Σ and a set of forbidden factors S, we say that a finite set W of infinite words over Σ *essentially avoids* S if every word in W avoids S and every bi-infinite words over Σ avoiding S is equivalent to one of the words in W. If W contains only one word w, we denote the set W by w instead of $\{w\}$. Then we can restate Theorem 5.3: b_3 essentially avoids `010`, `212`, and squares

The results in the next section involve b_3. We have tried without success to prove them by using Theorem 5.3. We need the following stronger property of b_3:

Lemma 5.4. b_3 *essentially avoids* `010`*,* `212`*,* XX *with* $1 \leq |X| \leq 3$*, and* `2`YY *with* $|Y| \geq 4$*.*

Proof. From Theorem 5.3, b_3 clearly avoids `010`, `212`, XX with $1 \leq |X| \leq 3$, and `2`YY with $|Y| \geq 4$. Thus we only need to show that any word avoiding this set of factors is equivalent to b_3.

We start by checking by computer that b_3 has the same set of factors of length 100 as every bi-infinite ternary word avoiding `010`, `212`, XX with $1 \leq |X| \leq 3$, and `2`YY with $|Y| \geq 4$. On one side we compute the set of factors of a given size of b_3. On the other side we compute the set of words of length 120 that avoid `010`, `212`, XX with $1 \leq |X| \leq 3$, and `2`YY with

$|Y| \geq 4$ and we extract the factor of length 100 in the middle. Then we only need to check the equality of these two sets.

We can check that the set of minimal forbidden factors of b_3 of length at most 4 is $F = \{00, 11, 22, 010, 212, 0202, 2020, 1021, 1201\}$. To finish the proof, we use Theorem 5.3 and we suppose for contradiction that w is a bi-infinite ternary word that contains a large square MM and avoids both F and large factors of the form $2YY$.

Case $M = 0N$. Then w contains $MM = 0N0N$. Since $00 \in F$ and $2YY$ is forbidden, w contains $10N0N$. Since $\{11, 010\} \subset F$, w contains $210N0N$. If $N = P1$, then w contains $210P10P1$, which contains $2YY$ with $Y = 10P$. So $N = P2$ and w contains $210P20P2$. If $P = Q1$, then w contains $210Q120Q12$. Since $\{11, 212\} \subset F$, the factor $Q12$ implies that $Q = R0$ and w contains $210R0120R012$. Moreover, since $\{00, 1201\} \subset F$, the factor $120R$ implies that $R = 2S$ and w contains $2102S01202S012$. Then there is no possible prefix letter for S: 0 gives 2020, 1 gives 1021, and 2 gives 22. This rules out the case $P = Q1$. So $P = Q0$ and w contains $210Q020Q02$. The factor $Q020Q$ implies that $Q = 1R1$, so that w contains $2101R10201R102$. Since $\{11, 010\} \subset F$, the factor $01R$ implies that $R = 2S$, so that w contains $21012S102012S102$. The only possible right extension with respect to F of 102 is 102012. So w contains $21012S102012S102012$, which contains $2YY$ with $Y = S102012$.

Case $M = 1N$. Then w contains $MM = 1N1N$. In order to avoid 11 and $2YY$, w must contain $01N1N$. If $N = P0$, then w contains $01P01P0$. So w contains the large square $01P01P$ and this case is covered by the previous item. So $N = P2$ and w contains $01P21P2$. Then there is no possible prefix letter for P: 0 gives 010, 1 gives 11, and 2 gives 212.

Case $M = 2N$. Then w contains $MM = 2N2N$. If $N = P1$, then w contains $2P12P1$. This factor cannot extend to $2P12P12$, since this is $2YY$ with $Y = P12$. So w contains $2P12P10$. Then there is no possible suffix letter for P: 0 gives 010, 1 gives 11, and 2 gives 212. This rules out the case $N = P1$. So $N = P0$ and w contains $2P02P0$. This factor cannot extend to $02P02P0$, since this contains the large square $02P02P$ and this case is covered by the first item. Thus w contains $12P02P0$. If $P = Q1$, then w contains $12Q102Q10$. Since $\{22, 1021\} \subset F$, the factor $102Q$ implies that

$Q = \mathtt{0}R$, so that w contains $\mathtt{120}R\mathtt{1020}R\mathtt{10}$. Then there is no possible prefix letter for R: $\mathtt{0}$ gives $\mathtt{00}$, $\mathtt{1}$ gives $\mathtt{1201}$, and $\mathtt{2}$ gives $\mathtt{0202}$. This rules out the case $P = Q\mathtt{1}$. So $P = Q\mathtt{2}$ and w contains $\mathtt{12}Q\mathtt{202}Q\mathtt{20}$. The factor $Q\mathtt{202}$ implies that $Q = R\mathtt{1}$ and w contains $\mathtt{12}R\mathtt{1202}R\mathtt{120}$. Since $\{\mathtt{00}, \mathtt{1201}\} \subset F$, w contains $\mathtt{12}R\mathtt{1202}R\mathtt{1202}$, which contains $\mathtt{2}YY$ with $Y = R\mathtt{1202}$. □

5.3 Polynomial formulas

In this section, we show that the formulas announced in Theorems 5.1 and 5.2 to be polynomial are avoidable over 2 letters and are indeed polynomial. In order to do that we first need to introduce the words that avoid this formulas. Let:

$g_x(\mathtt{0}) = \mathtt{01110}$,	$g_y(\mathtt{0}) = \mathtt{0111}$,	$g_z(\mathtt{0}) = \mathtt{0001}$,	$g_t(\mathtt{0}) = \mathtt{01011011010}$,
$g_x(\mathtt{1}) = \mathtt{0110}$,	$g_y(\mathtt{1}) = \mathtt{01}$,	$g_z(\mathtt{1}) = \mathtt{001}$,	$g_t(\mathtt{1}) = \mathtt{01011010}$,
$g_x(\mathtt{2}) = \mathtt{0}$.	$g_y(\mathtt{2}) = \mathtt{00}$.	$g_z(\mathtt{2}) = \mathtt{11}$.	$g_t(\mathtt{2}) = \mathtt{010}$.

Let $\overline{w}$ denote the word obtained from the (finite or bi-infinite) binary word w by exchanging $\mathtt{0}$ and $\mathtt{1}$. Obviously, if w avoids a given formula, then so does $\overline{w}$. A (bi-infinite) binary word w is *self-complementary* if $w \sim \overline{w}$. The words $g_x(b_3)$, $g_y(b_3)$, and $g_t(b_3)$ are self-complementary. Since the frequency of $\mathtt{0}$ in $g_z(b_3)$ is $\frac{5}{9}$, $g_z(b_3)$ is not self-complementary. Then $g_{\overline{z}}$ is obtained from g_z by exchanging $\mathtt{0}$ and $\mathtt{1}$, so that $g_{\overline{z}}(b_3) = \overline{g_z(b_3)}$.

Lemma 5.5.

- $\{g_x(b_3), g_y(b_3), g_z(b_3), g_{\overline{z}}(b_3)\}$ *essentially avoids* $AA.ABA.ABBA$.
- $g_x(b_3)$ *essentially avoids* $AABA.ABB.BBA$.
- *Let* f *be either* $ABA.AABB$, $BBA.ABA.AABB$, *or* $AABA.AABB$. *Then* $\{g_x(b_3), g_t(b_3)\}$ *essentially avoids* f.

We can deduce that the formulas that appear on this lemma are polynomially avoidable over 2 letters.

The rest of the section is devoted to its proof. Let us first state interesting properties of the morphisms and the formulas in Lemma 5.5.

Lemma 5.6. *For every* $p, s \in \Sigma_3$, $Y \in \Sigma_3^*$ *with* $|Y| \geq 4$, *and* $g \in \{g_x, g_y, g_z, g_{\overline{z}}, g_t\}$, *the word* $g(p2YYs)$ *contains an occurrence of* $AABA.AABBA$.

Proof. Since 0 is a prefix and a suffix of the g_x-image of every letter, there are $U, V, W \in \Sigma_3^+$ such that $g_x(p2YYs) = V000U00U00W$, which contains an occurrence of $AABA.AABBA$ with $A = 0$ and $B = 0U0$.

Since 0 is a prefix of the g_y-image of each letter, there are $U, V \in \Sigma_3^+$ such that $g_y(2YYs) = 000U0U0V$. It contains an occurrence of $AABA.AABBA$ with $A = 0$ and $B = 0U$.

Since 1 is a suffix of the g_z-image of each letter, $g_z(p2YY) = 111U1U1$ contains an occurrence of $AABA.AABBA$ with $A = 1$ and $B = 1U$.

Since $g_{\overline{z}}(p2YY) = \overline{g_z(p2YY)}$ and $g_z(p2YY)$ contains an occurrence of $AABA.AABBA$, $g_{\overline{z}}(p2YY)$ contains an occurrence of $AABA.AABBA$.

Since 010 is a prefix and a suffix of the g_t-image of every letter, there are $U, V, W \in \Sigma_3^+$ such that $g_t(p2YYs) = V010010010U010010U010010W$. Thus $g_t(p2YYs)$ contains an occurrence of $AABA.AABBA$ with $A = 010$ and $B = 010U010$. □

It is straightforward to check the following lemma:

Lemma 5.7. *$AABA.AABBA$ is divisible by every formula in Lemma 5.5.*

We are now ready to prove Lemma 5.5. To prove the avoidability, we have implemented Cassaigne's algorithm that decides, under mild assumptions, whether a morphic word avoids a formula [14]. And by running this algorithm we conclude that the formulas are avoided. So we only need to check that any other word is equivalent to one of the word from the list. We have to explain how the long enough binary words avoiding a formula can be split into 4 or 2 distinct incompatible types. A similar phenomenon has been described for $AABB.ABBA$ [43].

First, consider any infinite binary word w avoiding $AA.ABA.ABBA$. A computer check shows by backtracking that w must contain the factor 01110001110. In particular, w contains 00. Thus, w cannot contain both 010 and 0110, since it would produce an occurrence of $AA.ABA.ABBA$. Moreover, a computer check shows by backtracking that w cannot avoid both 010 and 0110. So, w must contain either 010 or 0110 (this is an exclusive or). By symmetry, w must contain either 101 or 1001. There are thus at most 4 possibilities for w, depending on which subset of $\{010, 0110, 101, 1001\}$ appears among the factors of w, see Figure 5.2.

Also, consider any infinite binary word w avoiding f, where f is either $ABA.AABB$, $BBA.ABA.AABB$, or $AABA.AABB$. Notice that the formulas $BBA.ABA.AABB$ and $AABA.AABB$ are divisible by $ABA.AABB$.

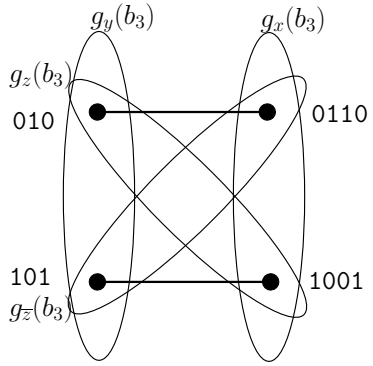

Figure 5.2 – The four infinite binary words avoiding $AA.ABA.ABBA$.

We check by backtracking that no infinite binary word avoids f, 0010, and 00110. A word containing both 0010 and 00110 contains an occurrence of $AABA.AABBA$, and thus an occurrence of f by Lemma 5.7. So w does not contain both 0010 and 00110. Thus, there are two possibilities for w depending on whether it contains 0010 or 00110.

Now, our tasks of the form "prove that a set of morphic words essentially avoids one formula" are reduced to (more) tasks of the form "prove that one morphic word essentially avoids one formula and a set of factors".

Since all the proofs of such reduced tasks are very similar, we only detail the proof that $g_y(b_3)$ essentially avoids $AA.ABA.ABBA$, 0110, and 1001. We check that the set of prolongable binary words of length 100 avoiding $AA.ABA.ABBA$, 0110, and 1001 is exactly the set of factors of length 100 of $g_y(b_3)$. Using Cassaigne's notion of circular morphism [14], this is sufficient to prove that every bi-infinite binary word of this type is the g_y-image of some bi-infinite ternary word w_3. It also ensures that w_3 and b_3 have the same set of small factors. Suppose for contradiction that $w_3 \not\sim b_3$. By Lemma 5.4, w_3 contains a factor $2YY$ with $|Y| \geq 4$. Since w_3 is bi-infinite, w_3 even contains a factor $p2YYs$ with $p, s \in \Sigma_3$. By Lemma 5.6, $g_y(w_3)$ contains an occurrence of $AABA.AABBA$ and by Lemma 5.7, $g_y(w_3)$ contains an occurrence of $AA.ABA.ABBA$. This contradiction shows that $w_3 \sim b_3$. So $g_y(b_3)$ essentially avoids $AA.ABA.ABBA$, 0110, and 1001.

5.4 Exponential formulas

In this section, we show that the exponential formulas from Theorems 5.1 and 5.2 are avoidable over 2 letters and exponential. We first give a construction for each formula and then we explain how to show that the constructions work. We need to introduce a few more definitions.

Given a morphism $g : \Sigma_3^* \to \Sigma_2^*$, a *sqf-g-image* is the image by g of a (finite or infinite) ternary square free word. In this section, with an abuse of language, we say that g avoids a formula f if every sqf-g-image avoids f. The set of word avoiding squares over 3 letters is exponential [9], so if we can show that a morphism g from the ternary alphabet to the binary alphabet avoids a formula f, then we know that f is exponential. For every 2-avoidable exponential formula f from Theorems 5.1 and 5.2, we give below a uniform morphism g that avoids f.

For the sake of completeness, we detail some stronger properties of this morphisms. If possible, we simultaneously avoid the reverse formula f^R of f. We also avoid large squares. Let SQ_t denote the pattern corresponding to squares of period at least t, that is, $SQ_1 = AA$, $SQ_2 = ABAB$, $SQ_3 = ABCABC$, and so on. The morphism g avoids SQ_t with t as small as possible. Since $\lambda(SQ_2) = 3$, a binary word avoiding SQ_3 is necessarily best possible in terms of length of avoided squares.

$f = AA.ABA.BABB$. This 22-uniform morphism avoids $\{f, f^R, SQ_6\}$:

$$\begin{aligned}
0 &\mapsto 00011011011100111000 11\\
1 &\mapsto 00011011011100011000 11\\
2 &\mapsto 00011011011000111001 11
\end{aligned}$$

This 44-uniform morphism avoids $\{f, SQ_5\}$:

$$\begin{aligned}
0 &\mapsto 00010010011000111001001100010011100100100111\\
1 &\mapsto 00010010011000100111001001100011100100100111\\
2 &\mapsto 00010010011000100111001001001100011100100111
\end{aligned}$$

Notice that $\{f, f^R, SQ_5\}$ is 2-unavoidable and $\{f, SQ_4\}$ is 2-unavoidable.

$f = AA.ABB.BBAB$. This 60-uniform morphism avoids $\{f, f^R, SQ_{11}\}$:

$$\begin{aligned}
0 &\mapsto 000110011100011001110011000111000110011100011100110001110011\\
1 &\mapsto 000110011100011001110001110011000111000110011100110001110011\\
2 &\mapsto 000110011100011001110001100111000111001100011100110001110011
\end{aligned}$$

Notice that $\{f, SQ_{10}\}$ is 2-unavoidable.

$f = AA.ABAB.BB$. f is self-reverse. This 11-uniform morphism avoids $\{f, SQ_4\}$:

$$0 \mapsto 00100110111$$
$$1 \mapsto 00100110001$$
$$2 \mapsto 00100011011$$

Notice that $\{f, SQ_3\}$ is 2-unavoidable.

$f = AA.ABBA.BAB$. $f$is self-reverse. This 30-uniform morphism avoids $\{f, SQ_6\}$:

$$0 \mapsto 000110001110011000110011100111$$
$$1 \mapsto 000110001100111001100011100111$$
$$2 \mapsto 000110001100011001110011100111$$

Notice that $\{f, SQ_5\}$ is 2-unavoidable.

$f = AAB.ABB.BBAA$. f is self-reverse. This 30-uniform morphism avoids $\{f, SQ_5\}$:

$$0 \mapsto 000100101110100010110111011101$$
$$1 \mapsto 000100101101110100010111011101$$
$$2 \mapsto 000100010001011101110111010001$$

Notice that $\{f, SQ_4\}$ is 2-unavoidable.

$f = AAB.ABBA.BAA$. f is self-reverse. This 38-uniform morphism avoids $\{f, SQ_5\}$:

$$0 \mapsto 00010001000101110111010001011100011101$$
$$1 \mapsto 00010001000101110100011100010111011101$$
$$2 \mapsto 00010001000101110001110100010111011101$$

Notice that $\{f, SQ_4\}$ is 2-unavoidable.

$f = AABB.ABBA$. This 193-uniform morphism avoids $\{f, SQ_{16}\}$:

```
0 ↦ 00010001011011101100010110111000101101110111000101100010001011
01110110001011011101110001011011101100010110111000101101110111000001
01100010001011011100010110111011100010110111011000101101110001011
1 ↦ 00010001011011101100010110111000101101110111000101100010001011
01110001011011101110001011011101100010110111000101101110111000001011
00010001011011101100010110111011100010110111011000101101110001011
2 ↦ 00010001011011100010110111011100010110001000101101110110001011
01110111000101101110110001011011100010110111011100010110001000101l
01110110001011011100010110111011100010110111011000101101110001011
```

Notice that $\{f, f^R\}$ is 2-unavoidable and $\{f, SQ_{15}\}$ is 2-unavoidable. Previous papers [42, 43] have considered a 102-uniform morphism to avoid $\{f, SQ_{27}\}$.

$f = ABAB.BABA$. f is self-reverse. This 50-uniform morphism avoids $\{f, SQ_3\}$, see [42]:

```
0 ↦ 00011001011000111001011001110001011100101100010111
1 ↦ 00011001011000101110010110011100010110001110010111
2 ↦ 00011001011000101110010110001110010111000101100111
```

Notice that a binary word avoiding $\{f, SQ_3\}$ contains only the squares `00`, `11`, and `0101` (or `00`, `11`, and `1010`).

$f = AABA.BABA$. A case analysis of the small factors shows that a recurrent binary word avoids $\{f, f^R, SQ_3\}$ if and only if it contains only the squares `00`, `11`, and `0101` (or `00`, `11`, and `1010`). Thus, the previous 50-uniform morphism that avoids $\{ABAB.BABA, SQ_3\}$ also avoids $\{f, f^R, SQ_3\}$.

$f = AAA$. f is self-reverse. This 32-uniform morphism avoids $\{f, SQ_4\}$:

```
0 ↦ 00101001101101001011001001101011
1 ↦ 00101001101100101101001001101011
2 ↦ 00100101101001001101101001011011
```

Notice that $\{f, SQ_3\}$ is 2-unavoidable.

$f = ABA.BAAB.BAB$. f is self-reverse. This 10-uniform morphism avoids $\{f, SQ_3\}$:

$$\begin{aligned} 0 &\mapsto 0001110101 \\ 1 &\mapsto 0001011101 \\ 2 &\mapsto 0001010111 \end{aligned}$$

$f = AABA.ABAA.BAB$. f is self-reverse. This 57-uniform morphism avoids $\{f, SQ_6\}$:

$$\begin{aligned} 0 &\mapsto 000101011100010110010101100010111001011000101011100101011 \\ 1 &\mapsto 000101011100010110010101100010101110010110001011100101011 \\ 2 &\mapsto 000101011100010110010101100010101110010101100010111001011 \end{aligned}$$

Notice that $\{f, SQ_5\}$ is 2-unavoidable.

$f = AABA.ABAA.BAAB$. f is self-reverse. This 30-uniform morphism avoids $\{f, SQ_3\}$:

$$\begin{aligned} 0 &\mapsto 000101110001110101000101011101 \\ 1 &\mapsto 000101110001110100010101110101 \\ 2 &\mapsto 000101110001010111010100011101 \end{aligned}$$

$f = ABAAB$. This 10-uniform morphism avoids $\left\{f, f^R, SQ_3\right\}$, see [42]:

$$\begin{aligned} 0 &\mapsto 0001110101 \\ 1 &\mapsto 0000111101 \\ 2 &\mapsto 0000101111 \end{aligned}$$

$f = BAB.ABA.AABB$. f is self-reverse. This 16-uniform morphism avoids $\{f, SQ_5\}$:

$$\begin{aligned} 0 &\mapsto 0101110111011101 \\ 1 &\mapsto 0100010111010001 \\ 2 &\mapsto 0001010111010100 \end{aligned}$$

Notice that $\{f, SQ_4\}$ is 2-unavoidable.

$f = AAB.ABA.ABBA$**.** f is avoided with its reverse. This 84-uniform morphism avoids $\{f, f^R, SQ_5\}$:

$0 \mapsto$ 000100010111000111010001000101110111010001011100011101000101110100011101110100001011100011101000101110111010001011100011101000101110100011101110100001011100011101000101110111010001011100011101000101110100011101110100001011100011101

$f = BBA.AABA.AABB$. This 160-uniform morphism avoids $\{f, f^R, SQ_{21}\}$:

```
0 ↦ 00010110010111000101110010110001011100010110010111001011000101
110010110001011001011100101100010111000101100101110001011100101100
0101100101110010110001011100101 1
1 ↦ 00010110010111000101110010110001011100010110010111001011000101
110010110001011001011100010110010111001011000101110001011100101100
0101100101110010110001011100101 1
2 ↦ 00010110010111000101110010110001011100010110010111001011000101
110010110001011001011100101100010111000101100101110001011001011100
0101100101110010110001011100101 1
```

This 202-uniform morphism avoids $\{f, SQ_5\}$:

```
0 ↦ 00011010011101101000110101000111011010011011010100011101101000
110101000111011010100011010011101101001101101010001101001110110101
000111011010001101010001110110101000110100111011010100011101101001
10110101
1 ↦ 00011010011101101000110101000111011010011011010100011010011101
101010001110110100011010100011101101010001101001110110101000111011
010011011010100011010011101101001101101010001110110100011010100011
10110101
2 ↦ 00011010011101101000110101000111011010011011010100011010011101
101010001110110100011010100011101101010001101001110110100110110101
000111011010001101010001110110101000110100111011010100011101101001
10110101
```

Notice that $\{f, f^R, SQ_{20}\}$ is 2-unavoidable and $\{f, SQ_4\}$ is 2-unavoidable.

In the rest of this section we explain how to show that for every formula f above and corresponding morphism g, g avoids f. We start by checking that every morphism is synchronizing, that is, for every letters $a, b, c \in \Sigma_3$, the factor $g(a)$ only appears as a prefix or a suffix in $g(bc)$.

For every q-morphism g, the sqf-g-images are claimed to avoid SQ_t with $2t < q$. Let us prove that SQ_t is avoided. We check exhaustively that the sqf-g-images contain no square uu such that $t \leq |u| \leq 2q-2$. Now suppose for contradiction that an sqf-g-image contains a square uu with $|u| \geq 2q-1$. The condition $|u| \geq 2q-1$ implies that u contains a factor $g(a)$ with $a \in \Sigma_3$. This factor $g(a)$ only appears as the g-image of the letter a because g is synchronizing. Thus the distance between any two factors u in an sqf-g-image is a multiple of q. Since uu is a factor of an sqf-g-image, we have

q divides $|u|$. Also, the center of the square uu cannot lie between the g-images of two consecutive letters, since otherwise there would be a square in the pre-image. The only remaining possibility is that the ternary square free word contains a factor $aXbXc$ with $a, b, c \in \Sigma_3$ and $X \in \Sigma_3^+$ such that $g(aXbXc) = bsYpsYpe$ contains the square $uu = sYpsYp$, where $g(X) = Y$, $g(a) = bs$, $g(b) = ps$, $g(c) = pe$. Then, we also have $a \neq b$ and $b \neq c$ since $aXbXc$ is square free. Then abc is square free and $g(abc) = bspspe$ contains a square with period $|s| + |p| = |g(a)| = q$. This is a contradiction since the sqf-g-images contain no square with period q.

Notice that f is not square free, since the only avoidable square free binary formula is $ABA.BAB$, which is not 2-avoidable. We distinguish two kinds of formula.

A formula is *easy* if every appearing variable is contained in at least one square. Every potential occurrence of an easy formula then satisfies $|A| < t$ and $|B| < t$ since SQ_t is avoided. The longest fragment of every easy formula has length 4. So, to check that g avoids an easy formula, it is sufficient to consider the set of factors of the sqf-g-images with length at most $4(t-1)$.

A formula is *tough* if one of the variables is not contained in any square. The tough formulas are $ABA.BAAB.BAB$, $AABA.ABAA.BAAB$, $ABAAB$ and $AABA.ABAA.BAB$. They have been named so that the variable that does not appear in any square is B. As before, every potential occurrence of a tough formula satisfies $|A| < t$ since SQ_t is avoided. Suppose for contradiction that $|B| \geq 2q - 1$. By previous discussion, the distance between any two occurrences of B in an sqf-g-image is a multiple of q. The case of $ABA.BAAB.BAB$ can be settled as follows. The factor $BAAB$ implies that q divides $|BAA|$ and the factor BAB implies that q divides $|BA|$. This implies that q divides $|A|$, which contradicts $|A| < t$. For the other formulas, only one fragment contains B twice. This fragment is said to be *important*. Since $|A| < t$, the important fragment is a repetition which is "almost" a square. The important fragment is $\boldsymbol{B}A\boldsymbol{B}$ for $AABA.ABAA.BAB$, $\boldsymbol{B}AA\boldsymbol{B}$ for $AABA.ABAA.BAAB$, and $\boldsymbol{AB}A\boldsymbol{AB}$ for $ABAAB$. Informally, this almost square implies a factor $aXbXc$ in the ternary pre-image, such that $|a| = |c| = 1$ and $1 \leq |b| \leq 2$. If $|X|$ is small, then $|B|$ is small and we check exhaustively that there exists no small occurrence of f. If $|X|$ is large, there would exist a ternary square free factor $aYbYc$ with $|Y|$ small, such that $g(aYbYc)$ contains the important fragment of an occurrence of f if and only if $g(aXbXc)$ contains the important fragment of a smaller occurrence of f.

5.5 A formula characterizing b_3

In Lemma 5.4, we gave a characterization of b_3. In the proof of Theorem 5.1, every minimally 2-avoidable binary formula, and thus every 2-avoidable binary formula, is avoided by some morphic image of b_3. The next theorem is also a characterization of b_3, but is based on a formula. Let $b_{3(0\leftrightarrow 1)}$ be b_3 where we exchange 0 and 1, that is we applied the morphism $0 \mapsto 1,\ 1 \mapsto 0,\ 2 \mapsto 2$. Similarly let $b_{3(2\leftrightarrow 1)}$ be the image of b_3 by $0 \mapsto 0,\ 1 \mapsto 2,\ 2 \mapsto 1$.

Theorem 5.8. *Let w be a recurrent ternary word avoiding the formula $F = ABCAB.BAC.ACA$. Then w is equivalent to a word from $\{b_3, b_{3(0\leftrightarrow 1)}, b_{3(2\leftrightarrow 1)}\}$.*

Proof. Let w be ternary recurrent word avoiding F. For the sake of contradiction assume that F contains a square uu. Then since w is recurrent there is a non empty factor v such that $uuvuu$ is a factor of w. Clearly $uuvuu$ does not avoid F, so we reach a contradiction. Thus w has to be square-free.

The longest ternary word avoiding 012 and squares has length 29, so w contains 012, 021, 102, 120, 201 and 210. Since w contains 102 but avoids F we deduce that it avoids 01201 or 020 (by considering $A \mapsto 0$, $B \mapsto 1$, $C \mapsto 2$ in F). Thus 01201 and 020 cannot be both factors of w, and we can reproduce this for any permutation of the letters.

Moreover, if the factor 01201 appears then the factor 1012010 appears because any other extension of 012010 contains a square. So if 01201 appears 101 appears and thus 12012 cannot appear. Finally we get that 01201 and 12012 cannot both be factors of w. If we do that for every permutation of the letter we get the graph from Fig. 5.3.

An edge between two factors in Fig. 5.3 means that the 2 factors cannot both appear in w because they imply an occurrence of the formula. A dashed edge means that the 2 factors cannot both appear in w because one of them contains a factors that has an edge with the other one.

One can check, that if we forbid the 6 long factors from Fig.5.3, F and squares the longest ternary word has length 32. So w contains at least one of the long factors. But again we can show that if we allow 10210, but forbid all the other long factors, F and squares the longest ternary word has length 43. So we know that we have to allow at least 2 long factors. Up to a permutation of the alphabet there are only 2 possibilities:

— w contains the factors 10210 and 01201, then w avoids 121 and 020 so from Theorem 5.3, w is equivalent to $b_{3(2\leftrightarrow 1)}$.

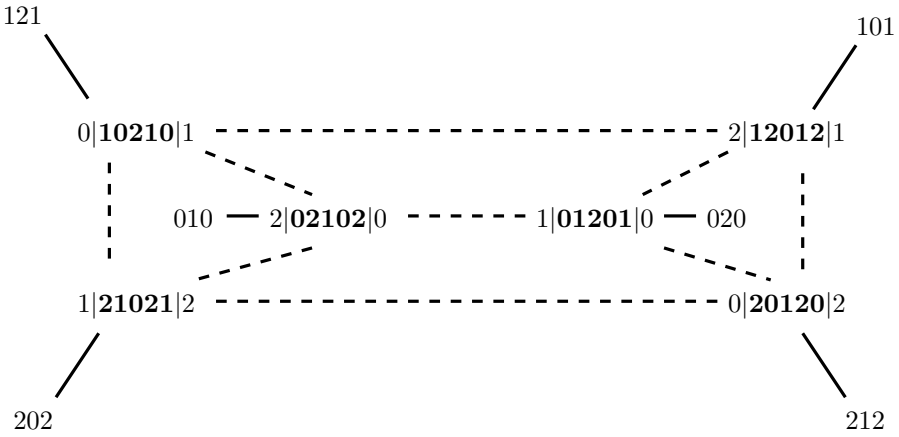

Figure 5.3 – The incompatibility graph.

— w contains the factors 10210 and 20120, then w avoids 121 and 212, and one can check that the longest word avoiding 121, 212, squares and F has length 21.

Thus w avoids squares and one of the 3 sets $\{121, 020\}$, $\{101, 202\}$ or $\{010, 212\}$. By Theorem 5.3, w is equivalent to one of $\{b_3, b_{3(0\leftrightarrow 1)}, b_{3(2\leftrightarrow 1)}\}$. □

Note that this implies that $ABCAB.BAC.ACA$ is a polynomial formula.

5.6 Remarks about polynomial languages

It seems interesting to find more polynomial languages defined by simple sets of forbidden factors. They are different examples in the literature [4, 5, 59]. Two of them are:

— b_3 essentially avoids AA, `010`, and `212`.

— The binary Thue-Morse word essentially avoids $ABABA$, `000`, and `111` (or equivalently $ABABA$ and AAA).

Now we can extend this list:

— $g_x(b_3)$ essentially avoids $AAB.BAA.BBAB$.

— $\{g_x(b_3), g_t(b_3)\}$ essentially avoids $ABA.AABB$ (or equivalently $ABACAABB$), $BBA.ABA.AABB$, or $AABA.AABB$ (or equivalently $AABACAABB$).

— $\{g_x(b_3), g_y(b_3), g_z(b_3), g_{\overline{z}}(b_3)\}$ essentially avoids $AA.ABA.ABBA$.

— $ABCAB.BAC.ACA$ is polynomial.

The last formula gave a new characterization of the polynomial language of the factors of the ternary Thue-Morse word. It would be interesting to have a similar characterization for the binary Thue-Morse word.

Problem 5.9. *Is there a formula F such that any (recurrent) word avoiding F is equivalent to the Thue-Morse word?*

Chapter 6

Avoidability of Binary Patterns in the Abelian Sense

We gave in Chapter 5 the usual definition of the realization of a pattern, that is a word w *realizes* a pattern $P \in \Delta^*$, if there is a non-erasing morphism h such that $h(P) = w$. Equivalently, w realizes the pattern P if $w = w_1 w_2 \dots w_{|P|}$ such that $\forall i, j,\ P_i = P_j \implies w_i = w_j$. Based on this second definition, there is a natural generalization of patterns to abelian equivalence or any other equivalence relation.

Let $P = P_1 P_2 \dots P_n$ be a pattern, where the P_i are letters. Then we say that a word $w \in \mathcal{A}^*$ *realizes* P *in the abelian sense* if there are $w_1, \dots, w_n \in \mathcal{A}^+$ such that $w = w_1 w_2 \dots w_n$ and $\forall i, j,\ P_i = P_j \implies w_i \approx_a w_j$. If a word w has no factor that realizes a pattern P in the abelian sense, then w *avoids* P in the abelian sense. We say that a pattern is *abelian-k-avoidable* if there is a word from an alphabet of size k that avoids this pattern. For any pattern $P \in \Delta^*$, the *abelian-avoidability index* of P, denoted by $\lambda_a(P)$, is the smallest integer k such that P is abelian-k-avoidable or ∞ if there is no such k. It is an abelian analog of the usual avoidability index of a pattern P.

For example, $\lambda_a(AA) = 4$ since abelian squares are avoidable over 4 letters [34]. Likewise we can deduce from Dekking's results that $\lambda_a(AAA) = 3$ and $\lambda_a(AAAA) = 2$ [22]. We can also deduce $\lambda_a(ABA) = \lambda_a(ABACABA) = \infty$ from the fact that these two patterns are not avoidable in the usual sense [61]. In [20] authors showed that binary pattern of length greater than 118 are abelian-2-avoidable and asked for a more precise characterization. In Section 6.2, we show that under some conditions one can decide if the fixed

point of a morphism avoids a given pattern. The results presented in this chapter are part of the article [56].

6.1 Divisibility and easy results

This section contains simple results about λ_a, the abelian-avoidability index function. The first proposition tells us that an abelian realization of a pattern can also be given by a morphism.

Proposition 6.1. *For any word u and pattern $P = P_1P_2 \dots P_{|P|}$ the 2 following conditions are equivalent:*

- *u is an abelian realization of P,*
- *there exist a morphism $\psi : \Delta^* \mapsto \mathcal{A}^*$ and $u_1, u_2, \dots, u_{|P|} \in \mathcal{A}^*$ such that $u = u_1u_2 \dots u_{|P|}$ and for all $i \in [1, |P|]$, $u_i \approx_a \psi(P_i)$.*

Patterns are words so we can extend the definition of the occurrence of a pattern in a word to the occurrence of a pattern in a pattern. For any alphabet Σ (in particular one can take $\Sigma = \mathcal{A} \cup \Delta$), $\leq_a$ is the relation over Σ^* such that for all $u, v \in \Sigma^*$, $u \leq_a v$ if and only if v contains an abelian occurrence of u.

Theorem 6.2. *$\leq_a$ is a preorder on Σ^*.*

Proof. The reflexivity of $\leq_a$ is clear, we need to show the transitivity. Let $x, y, z \in \Sigma^*$ such that $x \leq_a y$ and $y \leq_a z$. From Proposition 6.1, we have:

- There are $i_1 < \dots < i_{|x|+1} \in [1, |y| + 1]$ and a morphism $\phi_1 : \Sigma^* \mapsto \Sigma^*$ such that for all $k \in [1, |x|]$, $y_{i_k}y_{i_k+1} \dots y_{i_{k+1}-1} \approx_a \phi_1(x_k)$.
- There are $j_1 < \dots < j_{|y|+1} \in [1, |z| + 1]$ and a morphism $\phi_2 : \Sigma^* \mapsto \Sigma^*$ such that for all $k \in [1, |y|]$, $z_{j_k}z_{j_k+1} \dots z_{j_{k+1}-1} \approx_a \phi_2(y_k)$.

Then, for all $k \in [1, |x|]$:

$$\Psi(z_{j_{i_k}} z_{j_{i_k}+1} \dots z_{j_{i_{k+1}}-1}) = \sum_{l=i_k}^{i_{k+1}-1} \Psi(z_{j_l} z_{j_l+1} \dots z_{j_{l+1}-1})$$

$$\Psi(z_{j_{i_k}} z_{j_{i_k}+1} \dots z_{j_{i_{k+1}}-1}) = \sum_{l=i_k}^{i_{k+1}-1} \Psi(\phi_2(y_l))$$

$$\Psi(z_{j_{i_k}} z_{j_{i_k}+1} \dots z_{j_{i_{k+1}}-1}) = \Psi(\phi_2(y_{i_k})\phi_2(y_{i_k+1}) \dots \phi_2(y_{i_{k+1}-1}))$$

$$\Psi(z_{j_{i_k}} z_{j_{i_k}+1} \dots z_{j_{i_{k+1}}-1}) = \Psi(\phi_2(y_{i_k} y_{i_k+1} \dots y_{i_{k+1}-1}))$$

$$\Psi(z_{j_{i_k}} z_{j_{i_k}+1} \dots z_{j_{i_{k+1}}-1}) = \Psi(\phi_2(\phi_1(x_k))).$$

Thus z contains an abelian occurrence of x given by the morphism $\phi_2 \circ \phi_1$. Hence $x \leq_a z$ and $\leq_a$ is transitive. □

The transitivity of $\leq_a$ can be used to show the following property:

Theorem 6.3. *The function $\lambda_a : \Delta^* \mapsto \mathbb{N} \cup \{\infty\}$ is anti-monotone, that is, for any pattern P and P' such that $P \leq_a P'$, $\lambda_a(P) \geq \lambda_a(P')$.*

Proof. Let $P, P' \in \Delta^*$ such that $P \leq_a P'$. If $\lambda_a(P) = \infty$ then by definition $\lambda(P') \leq \lambda_a(P)$. Otherwise there is an infinite word w over the alphabet $\mathcal{A} = \{1, 2, \dots, \lambda_a(P)\}$ that avoids P in the abelian sense. If w contains an abelian occurrence of P', w also contains an abelian occurrence of P by transitivity of $\leq_a$, which is a contradiction. Thus w avoids P' in the abelian sense, and $\lambda_a(P') \leq \lambda_a(P)$. □

The next corollary is a simple example of application of Theorem 6.3.

Corollary 6.4. *For any $n \in \mathbb{N}$, if there is a pattern P such that $\lambda_a(P) = n$ then there is an integer N such that every pattern of length at least N over an alphabet of size $n-1$ has abelian avoidability index at most n.*

Proof. Let P be such a pattern, then there is an integer N such that every pattern P' of length at least N over an alphabet of size $\lambda_a(P) - 1$ contains an abelian occurrence of P, $P \leq_a P'$. Thus by Theorem 6.3, $\lambda_a(P') \leq \lambda_a(P)$. □

From this corollary combined with the fact that $\lambda_a(AA) = 4$ and $\lambda_a(AAA) = 3$ we deduce that long enough binary patterns are abelian-3-avoidable and long enough ternary pattern are abelian-4-avoidable.

In fact, one can check by exhaustive search that every binary pattern of length greater than 9 contains an abelian cube, and every ternary pattern of length greater than 7 contains an abelian square. We can deduce the following:

Proposition 6.5. *Binary patterns of length greater than* 9 *are abelian-*3*-avoidable. Ternary patterns of length greater than* 7 *are abelian-*4*-avoidable.*

We can also deduce the exact list of binary patterns that are not abelian-4-avoidable:

Proposition 6.6. *Let* $P \in \{A, B\}^+$. *If* $P \in \{A, B, AB, BA, ABA, BAB\}$, *then* $\lambda_a(P) = \infty$ *otherwise* $\lambda_a(P) \leq 4$.

Proof. It is easy to check that the list $\{A, B, AB, BA, ABA, BAB\}$ is the complete list of binary patterns avoiding AA. Moreover, $\lambda_a(AA) = 4$ and thus by Theorem 6.3 if $P \notin \{A, B, AB, BA, ABA, BAB\}$, $\lambda_a(P) \leq 4$. □

This leads to ask which binary patterns are abelian-3-avoidable and which are abelian-2-avoidable. It was proven in [20] that binary pattern of length greater than 118 are abelian-2-avoidable. We can show that binary pattern of length greater than 14 are abelian-2-avoidable. For that, we need to find a list of abelian-2-avoidable patterns and then to apply Theorem 6.3.

6.2 Decidability of pattern freeness in the abelian sense

In this section we present an algorithm do decide under some conditions whether the fixed-point of a morphism avoids a given pattern in the abelian sense. The basic ideas are similar to the ideas used in Chapter 1. We generalize the notion of k-templates introduced in [19] that was also used in Chapter 1.

Let Δ and $\mathcal{A}$ be two alphabets, and P a pattern over Δ. A *P-template* over $\mathcal{A}$ is a $2(|P|+1)$-tuple of the form: $[w_0, v_1, w_1, v_2 \ldots, v_{|P|}, w_{|P|}]$ where for all i, $w_i \in \mathcal{A}^*$ and $v_i \in \mathbb{Z}^{|\mathcal{A}|}$. A word $w \in \mathcal{A}^*$ *realizes* (or *is a realization* of) a P-template $t = [w_0, v_1, \ldots, v_{|P|}, w_{|P|}]$ if there are $u_1, \ldots, u_{|P|} \in \mathcal{A}^+$ such that $w = w_0 u_1 w_1 u_2 \ldots u_{|P|} w_{|P|}$ and $P_i = P_j \implies \Psi(u_i) - \Psi(u_j) = v_i - v_j$, for all $i, j \in [1, |P|]$.

Each template can be associated to its set of realizations, but many templates are associated to the same set. We associate to any pattern P the function $\varphi_P : [1, |P|] \mapsto [1, |P|]$ such that $\varphi_P(i) = \min\{j : P_j = P_i\}$ is the position of the first occurrence of the letter P_i in P. We say that a P-template is *normalized* if for all $i \in [1, |P|]$, $\varphi_P(i) = i \implies v_i = \overrightarrow{0}$. For any P-template, we can compute a normalized template that is realized by the same set of words. Since one doesn't change the set of realizations by adding the same vector to all vectors corresponding to the same letter, one get the *normalization* of a template by taking for all i, $v'_i = v_i - v_{\varphi_P(i)}$ and $w'_i = w_i$. Note that there is a natural bijection between the set of k-templates from Chapter 1 and the set of normalized A^k-templates. In the following, we only use normalized templates.

We say that a morphism $h : \mathcal{A}^* \to \mathcal{A}^*$ is *convenient* if its associated matrix M_h is invertible, $\|M_h^{-1}\|_2 < 1$ and $\forall a \in \mathcal{A}$, $|h(a)| > 1$. We can now state the main theorem:

Theorem 6.7. *For any alphabets Δ and $\mathcal{A}$, pattern $P \in \Delta^*$, P-template t, convenient morphism $h : \mathcal{A}^* \mapsto \mathcal{A}^*$ and any letter $a \in \mathcal{A}$ such that $h(a) = as$ for some non-empty s, it is possible to decide if $h^\omega(a)$ avoids t.*

By definition, w avoids the P-template $[\varepsilon, \overrightarrow{0}, \varepsilon, \ldots, \overrightarrow{0}, \varepsilon, \overrightarrow{0}, \varepsilon]$, if and only if w avoids P. From that we can deduce the following corollary:

Corollary 6.8. *For any alphabets Δ and $\mathcal{A}$, pattern $P \in \Delta^*$, any convenient morphism $h : \mathcal{A}^* \mapsto \mathcal{A}^*$ and any letter $a \in \mathcal{A}$ such that $h(a) = as$ for some non-empty s, it is possible to decide if $h^\omega(a)$ is avoids P in the abelian sense.*

The rest of this section is devoted to the proof of Theorem 6.7. The main idea of the proof is that we can compute S, a set of templates, such that $t \in S$ and $h^{n+1}(a)$ avoids any template of S if and only if $h^n(a)$ avoids any template of S. Thus $h^\omega(a)$ avoids t if and only if a avoids any template of S which is easy to check. The set S corresponds to what we call *the set of special ancestors*. In the following Δ will always be the alphabet of patterns and $\mathcal{A}$ the alphabet of words and templates.

6.2.1 Parents, ancestors and specialization of a template

Let $t = [w_0, v_1, \ldots, v_{|P|}, w_{|P|}]$ and $t' = [w'_0, v'_1, \ldots, v'_{|P|}, w'_{|P|}]$ be two normalized P-templates. We say that t' *is a parent of t by h* if there are

$p_0, s_0, \ldots, p_{|P|}, s_{|P|} \in \mathcal{A}^*$ such that:

— $\forall i \in [0, |P|]$, p_i is a prefix of the image of the first letter of w'_i, s_i is a suffix of the image of the last letter of w'_i and $h(w'_i) = p_i w_i s_i$,

— $\forall i, j \in [1, |P|]$, $P_i = P_j \implies v_i - v_j = (\Psi(s_{i-1}) + M_h v'_i + \Psi(p_i)) - (\Psi(s_{j-1}) + M_h v'_j + \Psi(p_j))$.

For any normalized template t we denote $\mathrm{Par}_h(t)$ the set of parents of t by h. The *ancestors of t by h* is the set $\mathrm{Ancestors}_h(t) = \cup_{i=0}^{\infty} \mathrm{Par}_h^i(t)$. The relation "is an ancestor" is the transitive and reflexive closure of the relation "is a parent". Note that by definition a pattern is an ancestor of itself.

Intuitively this notion of parent and ancestor is similar to what we have for k-template. In particular, the idea is that the set of parent of a P-template t correspond to the preimages of the long realizations of t. We can show the first direction:

Lemma 6.9. *For any word w and any P-templates t and $t' \in \mathrm{Par}_h(t)$, if w is a realization of t' then $h(w)$ contains a realization of t.*

Proof. Let $t = [w_0, v_1, \ldots, v_{|P|}, w_{|P|}]$ and $t' = [w'_0, v'_1, \ldots, v'_{|P|}, w'_{|P|}] \in \mathrm{Par}_h(t)$. Then by definition there are $p_0, s_0, \ldots, p_{|P|}, s_{|P|} \in \mathcal{A}^*$ such that:

— $\forall i \in [1, |P|]$, $h(w'_i) = p_i w_i s_i$,

— $\forall i, j \in [1, |P|]$, $P_i = P_j \implies v_i - v_j = (\Psi(s_{i-1}) + M_h v'_i + \Psi(p_i)) - (\Psi(s_{j-1}) + M_h v'_j + \Psi(p_j))$.

Assume that there is a word w realizing t'. Then there are $u'_1, \ldots, u'_{|P|} \in \mathcal{A}^+$ such that $w = w'_0 u'_1 w'_1 u'_2 w'_2 \ldots u'_{|P|} w'_{|P|}$ and $\forall i, j$, $P_i = P_j \implies \Psi(u'_i) - \Psi(u'_j) = v'_i - v'_j$. Then $h(w) = p_0 w_0 s_0 h(u'_1) p_1 w_1 \ldots s_{|P|-1} h(u'_{|P|}) p_{|P|} w_{|P|} s_{|P|}$. For all i let $u_i = s_{i-1} h(u'_i) p_i$, then $W = w_0 u_1 w_1 u_2 w_2 \ldots u_{|P|} w_{|P|}$ is a factor of $h(w)$.

Moreover for all $i, j \in [1, |P|]$, if $P_i = P_j$ then:

$$\begin{aligned}
\Psi(u_i) - \Psi(u_j) &= \Psi(s_{i-1} h(u'_i) p_i) - \Psi(s_{j-1} h(u'_j) p_j) \\
&= (\Psi(s_{i-1}) + \Psi(h(u'_i)) + \Psi(p_i)) - (\Psi(s_{j-1}) + \Psi(h(u'_j)) + \Psi(p_j)) \\
&= (\Psi(s_{i-1}) + M_h v'_i + \Psi(p_i)) - (\Psi(s_{j-1}) + M_h v'_j + \Psi(p_j)) \\
&= v_i - v_j.
\end{aligned}$$

So W realizes t and is a factor of $h(w)$. □

By induction, we know that if one of the ancestors of t is not avoided by $h^n(a)$ for some $n \in \mathbb{N}$, then there is $m \geq n$ such that t is not avoided by $h^m(a)$. Lemma 6.9 has similar statement and proof as Lemma 1.11. On the other side, the notion of long realization is more complicated than in the k-template case since the w_i corresponding to different letters in P can have arbitrarily different size. We need to introduce the notion of P-template specialization to give an analog of Theorem 1.14.

Let $P \in \Delta^*$ and $L \subseteq \Delta$ then we denote by $P_{|L}$ the pattern which is obtained by deleting from P every letter from $\Delta - L$. For instance, $ABCBBCCA_{|\{A,C\}} = ACCCA$.

Let $\mathrm{Pos}_{(P,L)} : [1, |P_{|L}|] \mapsto [1, |P|]$ be such that $\mathrm{Pos}_{(P,L)}(i) = \min\{j : |(P_1 \dots P_j)_{|L}| = i\}$, where P_i is the i-th letter of P. That is, $\mathrm{Pos}_{(P,L)}(i)$ is the position of the letter in P that ends in position i in $P_{|L}$.

For any P-template $t = [w_0, v_1, w_1, \dots, v_{|P|}, w_{|P|}]$ and any $P_{|L}$-template $t_{|L} = [w'_0, v'_1, w'_1, \dots, v'_{|P_{|L}|}, w'_{|P_{|L}|}]$, we say that $t_{|L}$ is a *L-specialization* of t if there are $(u_i)_{i:P_i \notin L} \in \mathcal{A}^+$ such that:

— $\forall i,\ v'_i = v_{\mathrm{Pos}_{(P,L)}(i)}$,

— $\forall i, j,\ P_i = P_j \notin L \implies \Psi(u_i) - \Psi(u_j) = v_i - v_j$,

— $\forall i,\ w'_i = w_{i_b} u_{i_b+1} w_{i_b+1} \dots w_{i_e-1} u_{i_e} w_{i_e}$, where $i_b = \mathrm{Pos}_{(P,L)}(i)$ and $i_e = \mathrm{Pos}_{(P,L)}(i+1) - 1$.

The Lemma 6.10 states that the set of realizations of a specialization of a P-template t is a subset of the realizations of t. We omit the proof of this lemma which is technical but straightforward.

Lemma 6.10. *Let $P \in \Delta^*$ be a pattern and $L \subseteq \Delta$. For any P-template t and any L-specialization $t_{|L}$ of t if there is a word w realizing $t_{|L}$ then w realizes t.*

With this definition a P-template t has infinitely many L-specializations, but in most cases the parents of a given L-specialization are included in the L-specializations of the parents. We introduce the set of *small L-specializations* in order to keep a finite subset of them. A L-specialization of a P-template t is a *small L-specialization* if, with the notations from the definition of L-specialization, for any $A \in \Delta - L$ there is $i \in [1, |P|]$ such that $P_i = A$ and $|u_i| \leq 2 \cdot \max_{a \in \mathcal{A}} |h(a)|$.

The set of *special ancestors* of a P-template t by h is the smallest set of templates containing t and any ancestor and any small L-specialization of any of its element.

Now we show that this set allows us to decide if the morphism's fixed point avoids the template.

Theorem 6.11. *For any pattern $P \in \Delta^+$, any normalized P-template t, any convenient morphism h and any word $w \in \mathcal{A}^+$, if there is a factor f of $h(w)$ that realizes t, then there is a factor f' of w that realizes a special ancestor of t.*

In fact we show that f' realizes the parent of an L-specialization of t for some well chosen set L. The only thing we do is to unfold the definitions with this set L.

Proof. Let $t = [w_0, v_1, w_1, \ldots, v_{|P|}, w_{|P|}]$ be a normalized P-template and assume there is a factor f of $h(w)$ that realizes t. Then by definition there are $u_1, \ldots, u_{|P|} \in \mathcal{A}^+$ such that $f = w_0 u_1 w_1 u_2 w_2 \ldots u_{|P|} w_{|P|}$ and

$$\forall i, j \ P_i = P_j \implies \Psi(u_i) - \Psi(u_j) = v_i - v_j. \tag{6.1}$$

Let us introduce the set L:

$$L = \left\{ A \in \Delta : \forall i, P_i = A \implies |u_i| > 2 \cdot \max_{a \in \mathcal{A}} |h(a)| \right\}. \tag{6.2}$$

Take the $P_{|L}$-template $t_{|L} = [w'_0, v'_1, w'_1, \ldots, v_{|P_{|L}|}, w_{|P_{|L}|}]$ such that:

- $\forall i, \ v'_i = v_{\mathrm{Pos}_{(P,L)}(i)}$,
- $\forall i, \ w'_i = w_{i_b} u_{i_b+1} w_{i_b+1} \ldots w_{i_e-1} u_{i_e} w_{i_e}$, where $i_b = \mathrm{Pos}_{(P,L)}(i)$ and $i_e = \mathrm{Pos}_{(P,L)}(i+1) - 1$.

From the equality (6.1) and the definition of L, $t_{|L}$ is a small L-specialization of t. Let $(u'_i)_{1 \le i \le |P_{|L}|}$ be such that for all i, $u'_i = u_{\mathrm{Pos}_{(P,L)}(i)}$. Then $f = w'_0 u'_1 w'_1 \ldots u'_{|P_{|L}|} w'_{|P_{|L}|}$. Then from the equality (6.1) we can deduce:

$$\forall i, j \ [P_{|L}]_i = [P_{|L}]_j \implies \Psi(u'_i) - \Psi(u'_j) = v'_i - v'_j. \tag{6.3}$$

So f is a realization of the $P_{|L}$-template $t_{|L}$.

Since f is a factor of $h(w)$ there is a factor f' of w such that $h(f') = p_0 f s_{|P_{|L}|}$, where $p_0 \in \mathrm{prefixes}(h(f'_1))$ and $s_{|P_{|L}|} \in \mathrm{suffixes}(h(f'_{|f'|}))$. By construction, for all i, $|u'_i| > 2 \cdot \max_{a \in \mathcal{A}} |h(a)|$ so we know that each of the u'_i contains at least the full image of one letter. So there are $u''_1, \ldots, u''_{|P_{|L}|} \in \mathcal{A}^+$, $w''_0, \ldots, w''_{|P_{|L}|} \in \mathcal{A}^*$ and $s_0, p_1, s_1, \ldots, s_{|P_{|L}|-1}, p_{|P_{|L}|} \in \mathcal{A}^*$ such that $f' = w''_0 u''_1 w''_1 u''_2 \ldots u''_{|P_{|L}|} w''_{|P_{|L}|}$ and for all $i \in [0, |P|]$:

— p_i is a prefix of the image of the first letter of w''_i,

— s_i is a suffix of the image of the last letter of w''_i,

— $h(w''_i) = p_i w'_i s_i$,

— $u'_i = s_{i-1} h(u''_i) p_i$.

For all $i \in [1, |P_{|L}|]$, let $v''_i = \Psi(u''_i) - \Psi(u''_{\varphi_{P_{|L}}(i)})$ and let t'' be the $P_{|L}$-template $t'' = [w''_0, v''_1, w''_1, v''_2, \ldots, v''_{|P_{|L}|}, w''_{|P_{|L}|}]$. Then t'' is the normalization of the $P_{|L}$-template $[w''_0, \Psi(u''_1), w''_1, \Psi(u''_2), \ldots, \Psi(u''_{|P_{|L}|}), w''_{|P_{|L}|}]$ which is realized by f', thus f' is a realization of t''.

From $u'_i = s_{i-1} h(u''_i) p_i$ we get:

$$\Psi(u'_i) = \Psi(s_{i-1}) + M_h \Psi(u''_i) + \Psi(p_i). \tag{6.4}$$

Let $i, j \in [1, |P_{|L}|]$ such that $[P_{|L}]_i = [P_{|L}]_j$ then $\varphi_{P_{|L}}(i) = \varphi_{P_{|L}}(j)$ and hence

$$\Psi(u''_{\varphi_{P_{|L}}(i)}) = \Psi(u''_{\varphi_{P_{|L}}(j)}). \tag{6.5}$$

Now if we put all of that together we get:

$$\begin{aligned} v'_i - v'_j =& \Psi(u'_i) - \Psi(u'_j) \text{ (from (6.3))} \\ =& (\Psi(s_{i-1}) + M_h \Psi(u''_i) + \Psi(p_i)) - (\Psi(s_{j-1}) + M_h \Psi(u''_j) + \Psi(p_j)) \text{ (from (6.4))} \\ =& (\Psi(s_{i-1}) + M_h v''_i + \Psi(p_i)) - (\Psi(s_{j-1}) + M_h v''_j + \Psi(p_j)) \text{ (from (6.5)).} \end{aligned}$$

Thus t'' is a parent of $t_{|L}$. So t'' is a parent of a specialization of t and is realized by a factor f' of w. □

Theorem 6.11 together with the fact that, by definition, a special ancestor of a special ancestor of t is itself a special ancestor of t gives:

Theorem 6.12. *For any pattern $P \in \Delta^*$, any normalized P-template t, any convenient morphism h and any letter $a \in \mathcal{A}$, if there is a positive integer n and a factor of $h^n(a)$ that realizes t, then a realizes a special ancestor of t.*

We also need the converse, that is:

Theorem 6.13. *For any pattern $P \in \Delta^*$, any normalized P-template t, any convenient morphism h and any letter $a \in \mathcal{A}$, if a realizes a special ancestor t' of t, then there is a positive integer n and a factor of $h^n(a)$ that realizes t'.*

Proof. We first take the sequence of parent and L-specialization that reaches t' from t. Then we use Lemmas 6.9 and 6.10 to reverse operations on a and we reach the factor of $h^n(a)$ that realizes t. □

From Theorem 6.12 and Theorem 6.13 we deduce the following one:

Theorem 6.14. *For any pattern $P \in \Delta^*$, any normalized P-template t, any convenient morphism h and any letter $a \in \mathcal{A}$, $h^\omega(a)$ avoids t if and only if a does not realize any special ancestor of t.*

Now we have to show that the set of special ancestors of a template is finite and computable under the conditions from Theorem 6.7.

6.2.2 Computing the set of special ancestors

Lemmas 6.15, 6.16 and 6.17 tell us that the set of ancestors of a given template is computable.

Lemma 6.15. *For any convenient morphism $h : \mathcal{A}^* \mapsto \mathcal{A}^*$ and normalized P-template t, the set $\mathrm{Par}_h(t)$ is finite and computable.*

Proof. Since the template t is normalized we know that:

$$M_h v'_i = \begin{cases} 0 & \text{if } \varphi_P(i) = i \\ v_i - \Psi(s_{i-1}) - \Psi(p_i) + \Psi(s_{\varphi_P(i)-1}) + \Psi(p_{\varphi_P(i)}) & \text{if } \varphi_P(i) \neq i. \end{cases}$$

Since M_h is invertible, there is at most one parent for a given valuation of $(w'_i)_{0\leq i\leq |P|}$, $(s_i)_{0\leq i\leq |P|}$ and $(p_i)_{0\leq i\leq |P|}$. Moreover the possibilities for the s_i, p_i and hence for the w'_i are finite (if h is injective there is at most one possibility for each w'_i). So we can try all the valuations for $(w'_i)_{0\leq i\leq |P|}$, $(s_i)_{0\leq i\leq |P|}$ and $(p_i)_{0\leq i\leq |P|}$ and we get all the parents. □

Lemma 6.16. *For any convenient morphism $h : \mathcal{A}^* \mapsto \mathcal{A}^*$ and normalized P-template t there are $(r_1, \ldots, r_{|P|}) \in \mathbb{R}^+$ such that for any normalized ancestor $t' = [w'_0, v'_1, w'_1, v'_2 \ldots, v_{|P|}, w_{|P|}]$ of t by h then for all i, $\|v'_i\|_2 < r_i$.*

We omit the details of the proof of Lemma 6.16 which is similar to the proof of Lemma 4 in [19], or can be derived from Proposition 1.18. Let v_i be the i-th vector of t, then $v'_i = M^{-n} v_i + \sum_{j=0}^{n-1} M^{-j}(\Psi(s_j) + \Psi(p_j) - \Psi(s'_j) - \Psi(p'_j))$ for some s_j, s'_j and p_j, p'_j being respectively suffixes and prefixes of images of letters by h. Moreover $\|M_h^{-1}\|_2 < 1$, so $\|v'_i\|_2$ is bounded.

Lemma 6.17. *For any normalized P-template t the set of ancestors of t by h is finite and computable.*

Proof. Let $t' = [w_0, v_1, w_1, v_2 \ldots, v_{|P|}, w_{|P|}]$ be an ancestor of t by h. From Lemma 6.16, each of the v_i is bounded and since $v_i \in \mathbb{Z}^{|\mathcal{A}|}$ there are finitely many choices for each of the v_i. Moreover, since for all $a \in \mathcal{A}$, $|h(a)| > 1$, the length of the w_i is bounded and there are finitely many different values for the w_i. It implies that there are only finitely many possible ancestors.

In order to compute the set of ancestors, one starts with the singleton $S = \{t\}$ and repeats the operation $S = S \cup \mathrm{Par}_h(S)$ (computable thanks to Lemma 6.15) until S reaches a fixed point, which will eventually happen since the set of ancestors is finite. □

Now we need to show that the set of specialization of a P-template is computable.

Lemma 6.18. *For any pattern $P \in \Delta^*$, P-template t and $L \subseteq \Delta$ the set of small L-specializations of t is finite and computable.*

Proof. Let $P \in \Delta^*$, $t = [w_0, v_1, w_1, \ldots, v_{|P|}, w_{|P|}]$ be a P-template and $L \subseteq \Delta$. Let $t_{|L} = [w'_0, v'_1, w'_1, \ldots, v_{|P_{|L}|}, w_{|P_{|L}|}]$ be a small L-specialization of t. Since $t_{|L}$ is a small L-specialization of t, for any letter $A \notin L$ there is an index i_A such that $P_{i_A} = A$ and $|u_{i_A}| \leq 2 \cdot max_{a \in \mathcal{A}} |h(a)|$, and there are only finitely many possible values for the u_{i_A}. Then from the definition for all j, $(P_j = A \text{ and } j \neq i_A) \implies \Psi(u_j) = \Psi(u_{i_A}) + v_{P_j} - v_{P_{i_A}}$. So there are only finitely many possible values for each u_j.

Once we have chosen the $(u_i)_{i:P_i \notin L}$ the w'_i and v'_i are fixed. Hence by trying all the possible values for $(u_i)_{i:P_i \notin L}$ we can compute the set of all small L-specializations of t. □

We denote by $\mathrm{SmallSpec}_L(t)$ the set of small L-specializations of a P-template t. Now using Lemma 6.17 and Lemma 6.18 we can explain how to compute the set of special factors of a template.

Theorem 6.19. *For any alphabets Δ and $\mathcal{A}$, pattern $P \in \Delta^*$, normalized P-template t and convenient morphism $h : \mathcal{A}^* \mapsto \mathcal{A}^*$, one can compute the set of special ancestors of t by h.*

Proof. Algorithm 2 computes this set for any P, t and h.

Input : P, t, h.
Output: The set S of special ancestors of t.
$S = \text{Ancestors}_h(t)$;
for $i = |\Delta| - 1 \dots 0$ **do**
  for $L \subseteq \Delta$, $|L| = i$ **do**
    $S = S \cup \text{SmallSpec}_L(S)$;
  $S = S \cup \text{Ancestors}_h(S)$;

Algorithm 2: How to compute special ancestors.

This algorithm halts because if S is finite then by Lemma 6.17 and 6.18 one can execute $S = S \cup \text{Ancestors}(S)$ and $S = S \cup \text{SmallSpec}_L(S)$ and keep S finite.

For any $D \subseteq L \subseteq \Delta$ and any pattern $P \in \Delta^*$, $(P_{|D})_{|L} = P_{|D}$. So for any L-specialization t_{DL} of a D-specialization t_D of a P-template t, $t_{DL} = t_D$. It implies that at the end for any $L \subseteq \Delta$, every element of S has all of its small L-specializations in S. Since the last operation of the algorithm adds the ancestors, every ancestor of any element of S is in S. □

In some reasonable implementation of the algorithm, it is important to use for S a data-structure that allows to check if a template is already in S in logarithmic time. Moreover, we are careful with specialization so that we do not obtain twice the same template by two different paths of specialization (dropping the letter A and then the letter B is the same than dropping B and then A).

Under the conditions of Theorem 6.7, one can compute the set of special ancestors of a template, thanks to Theorem 6.19. Since we can compute the set of special ancestors and compare it to the letter a, we can decide if $h^\omega(a)$ avoids t by Theorem 6.14. This concludes the proof of Theorem 6.7.

We implemented this algorithm in C++ and thus we can check if a pattern is avoided by the fixed point of a morphism.

6.2.3 Small eigenvalues and morphic words

All the techniques from Chapter 1 can be used in order to improve this algorithm. In Theorem 6.7, we need M_h to be invertible and $\|M_h^{-1}\|_2 < 1$, but this can be replaced by "M_h has no eigenvalues of norm 1". We can use Proposition 1.16 and Proposition 1.18 to compute a subset of the realizable ancestors and the rest stays unchanged.

Similarly, we can use the techniques from Chapter 3 in order to be able to decide whether a morphic word avoids a pattern in the abelian sense. We need the same conditions than in Theorem 3.10. In particular, there are probably some binary patterns that we were not able to avoid, but can be avoided by the image of the fixed point of $h : a \mapsto ac, b \mapsto dc, c \mapsto b, d \mapsto ab$ by a second morphism. Our implementation of this algorithm was not able to terminate on the candidates pair of morphisms that we found due to the huge number of special ancestors between the bounds from Proposition 1.16. This would be interesting to find a way to improve this algorithm, maybe by finding a way to discard most of the special ancestors by using a second criteria or better bounds.

6.3 Results and open questions

In this section we use Corollary 6.8 as a black box that can decide if a pattern is avoided by the fixed-point of a morphism.

6.3.1 Abelian-3-avoidability of binary patterns

The first application is to improve Theorem 6.5 :

Theorem 6.20. *Binary patterns of length at least* 9 *are abelian-*3*-avoidable. More precisely every pattern that does not appear up to symmetry on the following list is abelian-3-avoidable:*

A, AA, AB, AAB, ABA, AABA, AABB, ABAB, ABBA, AABAA, AABAB, AABBA, ABAAB, ABABA, AABAAB, AABABA, AABABB, AABBAA, ABAABA, AABAABA, AABABAA, ABBABBA, AABAABAA, ABAABAAB.

Proof. It is well known that AAA is abelian-3-avoidable [22] and it is already enough to show the upper bound. Moreover, we can use the algorithm from Theorem 6.7 to show that any fixed point of $a \mapsto aabaac$, $b \mapsto cbbbab$, $c \mapsto cbccac$ avoids $AABBAB$ in the abelian sense. So we only need to find exhaustively all the words that avoid AAA, $AABBAB$ and $ABAABB$. We get the list of Theorem 6.20. □

Conversely, if there is a word that avoids $AABAA$, there is also a recurrent word w that avoids $AABAA$ and then w avoids AA, thus $\lambda_a(AABAA) = 4$.

So the patterns $A, AA, AB, AAB, ABA, AABA, AABAA$ are not abelian-3-avoidable. But, for the rest of the list, we do not know which of them are abelian-3-avoidable

Problem 6.21. *Which of the following patterns are abelian-3-avoidable?*

AABB, ABAB, ABBA, AABAB, AABBA, ABAAB, ABABA, AABAAB, AABABA, AABABB, AABBAA, ABAABA, AABAABA, AABABAA, ABBABBA, AABAABAA, ABAABAAB.

There is a direct link with the first Mäkelä's question. In Chapter 3 we showed that abelian squares of the form uv where $|u| \geq 6$ are avoidable over the ternary alphabet. If we can avoid abelian squares of period more than 2 over the binary alphabet then all the patterns from Problem 6.21 other than $AABB$ are abelian-3-avoidable.

6.3.2 Abelian-2-avoidability of binary patterns

For the binary case it was shown in [20] that:

Theorem 6.22 (J. D. Currie, T. I. Visentin)**.** *Binary patterns of length greater than 118 are abelian-2-avoidable.*

They also asked:

Problem 6.23 (J. D. Currie, T. I. Visentin)**.** *Characterize which binary patterns are abelian-2-avoidable.*

Using the algorithm from Theorem 6.7 we can improve this result and lower the 118 to 14. First we use the algorithm to check that:

Lemma 6.24. *The fixed points of the morphisms on the left avoid in the abelian sense the corresponding patterns in the right:*

morphisms	*avoided patterns*
$a \mapsto aabaa$ $b \mapsto bbabb$	*AABBBAAAB, ABAAABBBA, AAABABABBB, AAABABBABB,AAABABBBAB, AABBBABAAB, AABBBABABA, ABAABABBBA, ABAABBBABA, ABABAABBBA, ABBBABAAAB, AABAABBBAB, AABBBAABAB, AABBBAABAAB, AAABABBBAAAB, AABBBABBBAA, ABABABBBABA, ABABBABBABA, AAABAAABBAB, AAABBABAAAB, AAABAABAABAB, AAABABAAABAB, AABAAABABAAB, AAABAAABABBA, AAABABAABAAB, AAABABAABAAB, ABBABAAABAAB, ABABBBABBBABA.*
$a \mapsto aaaab$ $b \mapsto abbab$	*ABAABBBAAB, AAABBABABB, AAABBABBAB, AABAABBABB, AABABABBBA, AABABBABBA, AABABBBAAB, AABABBBABA, AABBAABBBA, AABBABABBA, AABBABBAAB, AABBABBABA, AABBBAABBA, ABAABBABBA, AABBABABBBA, AABABBBABBBA,*
$a \mapsto abb$ $b \mapsto aaab$	*AAAA, AAABAABBB, AAABBBABB, AABBABBBA, AABBBABBA, AAABBAAABB, AABABAAABB, ABBBAABBBA, AAABAABBAB, AAABAABAABB, AAABBAABAAB, AABAABAABBA, AABAABBAAAB, AABABABAAAB, AAABBAAABAB, AABAAABABAB, AABAAABBAAB, AAABAABAAABAB,*
$a \mapsto aaab$ $b \mapsto bbba$	*AAABABBBAA, AAABBAABBB, AAABBABBAA, ABABAAABBB, ABABBBAABBA, AABABBAAABA, AABBABAAABA,*
$a \mapsto abaa$ $b \mapsto babb$	*AABBABBABBA, AABABBABBBA, AABBBABBBABA, ABABBABBABBA, ABABBABBBABA, ABABBBABABBA, ABBABABBABBA,*
$a \mapsto aaaba$ $b \mapsto babbb$	*ABAABBBAAA, AABABBBAAA,*
$a \mapsto aababbaaaba$ $b \mapsto babbbaababb$	*AABAAABAAABAB, ABBBABBBABBBA, AAABAAABAAABAAA,*

It implies that, if a pattern contains any of the patterns from Lemma 6.24, then it can be avoided by a binary word. One can check by an exhaustive search that any binary pattern of length greater than 14 contains at least one of the patterns from the Lemma 6.24. It implies:

Theorem 6.25. *Binary patterns of length greater than 14 are abelian-2-avoidable.*

In fact, up to symmetry, there are only 284 patterns that avoid all patterns of Lemma 6.24. We give the list of the 284 patterns in Appendix A.

Theorem 6.26. *The patterns from the following list are abelian-2-unavoidable: A, AA, AB, AAA, AAB, ABA, AAAB, AABA, AABB, ABAB, ABBA, AAABA, AAABB, AABAA, AABAB, AABBA, ABAAB, ABABA, ABBBA, AAABAA, AAABAB, AABAAB, ABAAAB, AAABAAA.*

Proof. Let assume that $AAABAAA$ is abelian-2-avoidable, then we can find a recurrent word that avoids $AAABAAA$ in the abelian sense and this words necessarily avoids AAA which is not possible. Thus $AAABAAA$ is abelian-2-unavoidable.

For all the other patterns one can do an exhaustive search using Proposition 3.3 and check that they are abelian-2-unavoidable. □

For the 260 other patterns we don't know which are abelian-2-avoidable and which are not.

We are left with some interesting questions:

Problem 6.27. *What is the length of the longest abelian-2-unavoidable binary pattern?*

We know that the answer is between 7 and 14.

Problem 6.28. *What is the exact list of the abelian-2-unavoidable binary patterns?*

It is probably related somehow to the second question from Mäkelä (Problem 3.2) which seems hard. For instance, if we can avoid abelian cubes of period more than 4 we can add $AAABAAABAAAB$ and many other formulas to the list of abelian-2-avoidable patterns. A proof that abelian cubes of the form uvw where $|u| \geq 3$ are avoidable over two letters would imply that many of the 260 patterns are also abelian-2-avoidable.

Finally we have some more general questions:

Problem 6.29. *For any finite alphabet Δ is it true that:*

- $\exists N \in \mathbb{N}$*, such that any pattern over* Δ *of length greater than* N *is abelian-avoidable?*
- $\exists N \in \mathbb{N}$*, such that any pattern over* Δ *of length greater than* N *is abelian-*$|\Delta|$*-avoidable?*
- $\exists N \in \mathbb{N}$*, such that any pattern over* Δ *of length greater than* N *is abelian-2-avoidable?*

6.3.3 Possible generalizations

We can also generalize the algorithm form Theorem 6.7 with the techniques from Chapter 2. We say that a word w over the alphabet $S \subseteq \mathbb{Z}^k$ realizes the pattern P in the additive sense if $w = w_1 w_2 \dots w_{|P|}$ such that for all i and j such that $P_i = P_j$, w_j and w_i are additive equivalent. We can then ask:

Problem 6.30. *What are the binary patterns (or ternary patterns) that are avoidable in the additive sense over the alphabet* $\{0, 1, 2, \dots, n-1\}$*?*

Since we can avoid additive cubes over $\{0, 1, 3, 4\}$ and we can avoid abelian cubes over the binary alphabet we know that every long enough binary pattern is additive avoidable over $\{0, 1, 3, 4\}$. We could use the technique from this chapter to show more precise results.

We could also ask the same question for pattern avoidability in the k-abelian sense and again the algorithm presented in this section can be adapted to this problem.

Chapter 7

Conclusion

In the first part of this document, we study the avoidability of different generalizations of abelian powers in words. We first show that abelian n-th power freeness of morphic word is decidable as long as the matrix associated to the morphism has no eigenvalues of norm 1. Our result is a generalization of a result from Currie and Rampersad for matrices with every eigenvalues greater than 1. Being able to decide when the morphism has eigenvalues of norm less than 1 is really useful for Chapter 2 and Chapter 3. Using properties of this kind of morphisms, we show that we can decide whether a morphic word avoids long abelian powers or additive powers.

We use these results to show that additives squares are avoidable over $\mathbb{Z}^2$. We then show that abelian squares of period more than 5 are avoidable over the ternary alphabet. In fact, this second result implies the first one, and there is a strong link between long abelian powers and additive powers. We also show using an exhaustive search that abelian cubes of period more than 2 are not avoidable over the binary alphabet.

In Chapter 4, we give sufficient conditions for a morphism to be (l, k)-abelian n-th power free. Using this and the exhaustive search technique from Chapter 3, we provide new results about the avoidability of long k-abelian squares. We first compute the minimal number of k-abelian squares in an infinite binary word for every $k \geq 3$, and we show that long 2-abelian squares are avoidable over the binary alphabet. Finally, we show that there is an infinite ternary word that contains only a single 2-abelian square.

In tables 7.1 and 7.2, we summarize what we know about avoidability of (long) k-abelian n-th powers, for every k and every n. In particular, it includes most results from Chapter 3 and Chapter 4. We identify "∞-abelian

n-th powers" with n-th powers. From Dekking result about the avoidability of abelian 4-th powers over 2 letters [22], we know that for any $n \geq 4$, $k \geq 1$ and alphabet $\mathcal{A}$ of size at least 2, k-abelian n-th powers are avoidable over $\mathcal{A}$.

k \ $\lvert\mathcal{A}\rvert$	2	3	4
1	∞ ([23])	$1 \leq ? \leq 34$ (Th. 3.12, Pb. 3.13)	0 ([34])
2	$5 \leq ? \leq 734$ (Prop. 4.6,Th. 4.7, Pb. 4.9)	1 (Th. 4.11)	0
3, 4	4 (Prop. 4.4, 4.5)	0 ([53])	0 0
≥ 5	3 (Th. 4.3)	0	0
∞	3 ([23])	0	0

Figure 7.1 – The minimal number of different k-abelian squares in infinite words over $\mathcal{A}$.

k \ $\lvert\mathcal{A}\rvert$	2	3
1	$3 \leq ?$ (Th. 3.5, Pb. 3.6)	0 ([22])
≥ 2	0 ([53])	0
∞	0	0

Figure 7.2 – The minimal number of different k-abelian cubes in infinite words over $\mathcal{A}$.

There are 3 open questions in these tables:

Problem 3.6. *Is there a $p \in \mathbb{N}$ such that one can avoid abelian cubes of period at least p over two letters ?*

Problem 3.13. *What is the smallest $p \in \mathbb{N}$ such that one can avoid abelian squares of period more than p over three letters ?*

Problem 4.9. *What is the minimal number of distinct 2-abelian squares that an infinite binary word must contain?*

The main open question regarding avoidability of additive powers is:

Problem 2.13 ([49]). *Are additive squares avoidable over* $\mathbb{Z}$*?*

It seems also interesting to find a characterization of the subsets of $\mathbb{Z}$ over which additive cubes are avoidable. It leads to the following question:

Problem 2.12. *Are additive cubes avoidable over* $\{0,1,2,3\}$*?* $\{0,1,4\}$*?* $\{0,2,5\}$*?*

In the second part of the document, we studied two different variations of patterns. First we studied formulas, and were able to give a complete characterization of the formulas that are avoidable over the binary alphabet, and we distinguished formulas which are exponentially avoidable from formulas that are polynomially avoidable. We were also able to find a new characterization for the language of the ternary Thue-Morse word.

In the last chapter, we studied the avoidability of patterns in the abelian sense. First we showed that the decision algorithm for abelian powers can be adapted to decide pattern freeness of morphic words in the abelian sense. Then, using this algorithm and an exhaustive search among patterns, we showed that every binary pattern of length more than 14 is avoidable over the binary alphabet. It is a significant improvement of the previous bound which was 118 but we still don't know if 14 is optimal. We note that this algorithm could be adapted to decide pattern freeness in the additive sense or in the k-abelian sense and there are probably interesting results about the avoidability of these patterns.

Chapter 8

Conclusion

Dans la première partie de ce document, nous étudions l'évitabilité de différentes généralisations des puissances abéliennes. Nous commençons par montrer que l'évitement des puissances n-ème abéliennes par un mot morphique est décidable si la matrice associée ne possède pas de valeur propre de norme 1. Notre résultat est une généralisation d'un résultat de Currie et Rampersad avec des matrices dont toutes les valeurs propres sont de norme supérieure à 1. Être capable de décider dans le cas où le morphisme possède des valeurs propres de norme inférieure à 1 est crucial pour les Chapitres 2 et 3. En utilisant les propriétés de tels morphismes, nous expliquons comment décider si un mot morphique évite les longues puissances abéliennes ou les puissances additives.

Nous commençons par utiliser ce résultat pour montrer que les carrés additifs sont évitables sur $\mathbb{Z}^2$. Nous montrons ensuite que les carrés abéliens de période plus que 5 sont évitables sur l'alphabet ternaire. En fait, ce second résultat implique le premier et les longues puissances abéliennes sont très liées aux puissances additives. Nous montrons aussi par une recherche exhaustive que les cubes abéliens de période plus que 2 ne sont pas évitables sur l'alphabet binaire.

Dans le Chapitre 4, nous donnons des conditions suffisantes pour qu'un morphisme soit "sans puissances n-ème (l,k)-abéliennes" ((l,k)-abelian n-th power free). En utilisant ce résultat et la technique de recherche exhaustive présentée au Chapitre 3, nous donnons de nouveaux résultats concernant l'évitabilité des longs carrés k-abéliens. Nous commençons par calculer le nombre minimal de carrés k-abéliens qu'un mot binaire infini doit contenir pour tout $k \geq 3$ et nous montrons que les longs carrés 2-abéliens sont évita-

bles sur l'alphabet binaire. Enfin, nous montrons qu'il existe un mot ternaire qui ne contient qu'un seul carré 2-abélien, ce qui est le minimum possible.

Nous résumons dans les tables 8.1 et 8.2, ce qui est connu de l'évitabilité des (longues) puissances n-ème k-abéliennes, pour tout k et n. Elles contiennent les résultats des Chapitres 3 et 4. Nous identifions les puissances n-ème ∞-abéliennes avec les puissances n-ème. D'après le résultat de Dekking à propos de l'évitabilité des puissances 4-ème abéliennes sur l'alphabet binaire [22], nous savons que pour tout $n \geq 4$, $k \geq 1$ et alphabet $\mathcal{A}$ de taille au moins 2, les puissances n-ème k-abéliennes sont évitables sur $\mathcal{A}$.

k \ $\lvert\mathcal{A}\rvert$	2	3	4
1	∞ ([23])	$1 \leq ? \leq 34$ (Th. 3.12, Pb. 3.13)	0 ([34])
2	$5 \leq ? \leq 734$ (Prop. 4.6,Th. 4.7, Pb. 4.9)	1 (Th. 4.11)	0
3, 4	4 (Prop. 4.4, 4.5)	0 ([53])	0 0
≥ 5	3 (Th. 4.3)	0	0
∞	3 ([23])	0	0

Figure 8.1 – Le nombre minimal de carrés k-abéliens différents que contient un mot infini sur $\mathcal{A}$.

k \ $\lvert\mathcal{A}\rvert$	2	3
1	$3 \leq ?$ (Th. 3.5, Pb. 3.6)	0 ([22])
≥ 2	0 ([53])	0
∞	0	0

Figure 8.2 – Le nombre minimal de cubes k-abéliens différents que contient un mot infini sur $\mathcal{A}$.

Il reste trois questions ouvertes dans cette table :

Problème 3.6. *Existe-t-il un entier $p \in \mathbb{N}$ tel que les cubes abéliens de période au moins p sont évitables sur l'alphabet binaire?*

Problème 3.13. *Quel est le plus petit entier $p \in \mathbb{N}$ tel que les carrés abéliens de période au moins p sont évitables sur l'alphabet ternaire?*

Problème 4.9. *Quel est le nombre minimal de carrés 2-abéliens différents qu'un mot binaire infini doit contenir?*

La question principale à propos de l'évitabilité des puissances additives est :

Problème 2.13 ([49])**.** *Les carrés additifs sont-ils évitables sur* $\mathbb{Z}$ *?*

Il semble aussi intéressant de caractériser les sous-ensembles de $\mathbb{Z}$ sur lesquels les cubes additifs sont évitables. Cela mène à la question suivante :

Problème 2.12. *Les cubes additifs sont-ils évitables sur* $\{0,1,2,3\}$*?* $\{0,1,4\}$*?* $\{0,2,5\}$*?*

Dans la seconde partie du document, nous avons étudié deux variations des motifs. Nous avons d'abord étudié les formules, et avons pu donner une classification complète des formules en fonction de leur évitabilité sur l'alphabet binaire et de la croissance exponentielle ou polynomiale du langage associé. Nous avons aussi donné une nouvelle caractérisation du langage du mot ternaire de Thue-Morse basée sur une unique formule.

Dans le dernier chapitre, nous avons étudié l'évitabilité des motifs au sens abélien. Nous avons montré que l'algorithme de décision du premier chapitre peut s'adapter pour décider si un mot morphique évite un motif au sens abélien. Puis, en utilisant cet algorithme et une recherche exhaustive, nous avons montré que tous les motifs binaires de longueur plus que 14 sont évitables sur l'alphabet binaire. C'est une amélioration significative de la précédente borne de 118, mais nous ne savons pas si 14 est optimale. Nous remarquons que cet algorithme peut être adapté pour l'étude des motifs au sens additif ou au sens k-abélien.

Appendix A

Abelian Avoidability Index of Binary Patterns

In Chapter 6, we gave different results about the abelian avoidability index of binary patterns. The following table contains the results from Theorem 6.20, Theorem 6.25 and Theorem 6.26.

The *mirror* of a word $w = w_1 w_2 \dots w_n$ is the word ${}^r w = w_n w_{n-1} \dots w_2 w_1$. Let $\phi : \{A, B\}^* \mapsto \{A, B\}^*$ be the morphism such that $\phi(A) = B$ and $\phi(B) = A$.

For any pattern $P \in \{A, B\}$, we clearly have: $\lambda_a(P) = \lambda_a(\phi(P)) = \lambda_a({}^r P) = \lambda_a(\phi({}^r P))$. And we only include P in the following list if it is the lexicographically smallest pattern from this list of 4 patterns. If none of the 4 equivalent patterns appears in the list, the pattern is abelian-2-avoidable.

For instance, if $P = ABBBABBB$, $\phi({}^r P) = AAABAAAB \leq_{lex} P$ and we can find $AAABAAAB$ in the list and so $2 \leq \lambda_a(P) \leq 3$. For $P = BBBAAABBBA$, we have to look for $\phi(P) = AAABBBAAAB$ which is not in the list so we know that P is abelian-2-avoidable, and we can even use Lemma 6.24 to find a word avoiding it.

A	$\lambda_a = \infty$	AA	$\lambda_a = 4$
AAA	$\lambda_a = 3$	AAAB	$\lambda_a = 3$
AAABA	$\lambda_a = 3$	AAABAA	$\lambda_a = 3$
AAABAAA	$\lambda_a = 3$	AAABAAAB	$2 \leq \lambda_a \leq 3$
AAABAAABA	$2 \leq \lambda_a \leq 3$	AAABAAABAA	$2 \leq \lambda_a \leq 3$
AAABAAABAAA	$2 \leq \lambda_a \leq 3$	AAABAAABAAAB	$2 \leq \lambda_a \leq 3$
AAABAAABAAABA	$2 \leq \lambda_a \leq 3$	AAABAAABAAABAA	$2 \leq \lambda_a \leq 3$

AAABAAABAAABB	$2 \leq \lambda_a \leq 3$	AAABAAABAAABBA	$2 \leq \lambda_a \leq 3$
AAABAAABAAB	$2 \leq \lambda_a \leq 3$	AAABAAABAABA	$2 \leq \lambda_a \leq 3$
AAABAAABAABAA	$2 \leq \lambda_a \leq 3$	AAABAAABAABAAA	$2 \leq \lambda_a \leq 3$
AAABAAABAB	$2 \leq \lambda_a \leq 3$	AAABAAABABA	$2 \leq \lambda_a \leq 3$
AAABAAABABAA	$2 \leq \lambda_a \leq 3$	AAABAAABABAAA	$2 \leq \lambda_a \leq 3$
AAABAAABABB	$2 \leq \lambda_a \leq 3$	AAABAAABB	$2 \leq \lambda_a \leq 3$
AAABAAABBA	$2 \leq \lambda_a \leq 3$	AAABAAABBAA	$2 \leq \lambda_a \leq 3$
AAABAAABBAAA	$2 \leq \lambda_a \leq 3$	AAABAAB	$2 \leq \lambda_a \leq 3$
AAABAABA	$2 \leq \lambda_a \leq 3$	AAABAABAA	$2 \leq \lambda_a \leq 3$
AAABAABAAA	$2 \leq \lambda_a \leq 3$	AAABAABAAAB	$2 \leq \lambda_a \leq 3$
AAABAABAAABA	$2 \leq \lambda_a \leq 3$	AAABAABAAABAA	$2 \leq \lambda_a \leq 3$
AAABAABAAB	$2 \leq \lambda_a \leq 3$	AAABAABAABA	$2 \leq \lambda_a \leq 3$
AAABAABAABAA	$2 \leq \lambda_a \leq 3$	AAABAABAABAAA	$2 \leq \lambda_a \leq 3$
AAABAABAB	$2 \leq \lambda_a \leq 3$	AAABAABABA	$2 \leq \lambda_a \leq 3$
AAABAABABAA	$2 \leq \lambda_a \leq 3$	AAABAABABAAA	$2 \leq \lambda_a \leq 3$
AAABAABABB	$2 \leq \lambda_a \leq 3$	AAABAABB	$2 \leq \lambda_a \leq 3$
AAABAABBA	$2 \leq \lambda_a \leq 3$	AAABAABBAA	$2 \leq \lambda_a \leq 3$
AAABAABBAAA	$2 \leq \lambda_a \leq 3$	AAABAB	$\lambda_a = 3$
AAABABA	$2 \leq \lambda_a \leq 3$	AAABABAA	$2 \leq \lambda_a \leq 3$
AAABABAAA	$2 \leq \lambda_a \leq 3$	AAABABAAAB	$2 \leq \lambda_a \leq 3$
AAABABAAABA	$2 \leq \lambda_a \leq 3$	AAABABAAABAA	$2 \leq \lambda_a \leq 3$
AAABABAAB	$2 \leq \lambda_a \leq 3$	AAABABAABA	$2 \leq \lambda_a \leq 3$
AAABABAABAA	$2 \leq \lambda_a \leq 3$	AAABABAABB	$2 \leq \lambda_a \leq 3$
AAABABAB	$2 \leq \lambda_a \leq 3$	AAABABABA	$2 \leq \lambda_a \leq 3$
AAABABABAA	$2 \leq \lambda_a \leq 3$	AAABABABAAA	$2 \leq \lambda_a \leq 3$
AAABABABB	$2 \leq \lambda_a \leq 3$	AAABABB	$2 \leq \lambda_a \leq 3$
AAABABBA	$2 \leq \lambda_a \leq 3$	AAABABBAA	$2 \leq \lambda_a \leq 3$
AAABABBAAA	$2 \leq \lambda_a \leq 3$	AAABABBAB	$2 \leq \lambda_a \leq 3$
AAABABBB	$2 \leq \lambda_a \leq 3$	AAABABBBA	$2 \leq \lambda_a \leq 3$
AAABB	$\lambda_a = 3$	AAABBA	$2 \leq \lambda_a \leq 3$
AAABBAA	$2 \leq \lambda_a \leq 3$	AAABBAAA	$2 \leq \lambda_a \leq 3$
AAABBAAAB	$2 \leq \lambda_a \leq 3$	AAABBAAABA	$2 \leq \lambda_a \leq 3$
AAABBAAABAA	$2 \leq \lambda_a \leq 3$	AAABBAAB	$2 \leq \lambda_a \leq 3$
AAABBAABA	$2 \leq \lambda_a \leq 3$	AAABBAABAA	$2 \leq \lambda_a \leq 3$
AAABBAABB	$2 \leq \lambda_a \leq 3$	AAABBAB	$2 \leq \lambda_a \leq 3$
AAABBABA	$2 \leq \lambda_a \leq 3$	AAABBABAA	$2 \leq \lambda_a \leq 3$
AAABBABAB	$2 \leq \lambda_a \leq 3$	AAABBABB	$2 \leq \lambda_a \leq 3$
AAABBABBA	$2 \leq \lambda_a \leq 3$	AAABBB	$2 \leq \lambda_a \leq 3$

AAABBBA	$2 \leq \lambda_a \leq 3$	AAABBBAA	$2 \leq \lambda_a \leq 3$
AAABBBAAA	$2 \leq \lambda_a \leq 3$	AAABBBAAB	$2 \leq \lambda_a \leq 3$
AAABBBAABB	$2 \leq \lambda_a \leq 3$	AAABBBAB	$2 \leq \lambda_a \leq 3$
AAABBBABA	$2 \leq \lambda_a \leq 3$	AAB	$\lambda_a = 4$
AABA	$\lambda_a = 4$	AABAA	$\lambda_a = 4$
AABAAAB	$2 \leq \lambda_a \leq 3$	AABAAABA	$2 \leq \lambda_a \leq 3$
AABAAABAA	$2 \leq \lambda_a \leq 3$	AABAAABAAAB	$2 \leq \lambda_a \leq 3$
AABAAABAAABA	$2 \leq \lambda_a \leq 3$	AABAAABAAABAA	$2 \leq \lambda_a \leq 3$
AABAAABAAABB	$2 \leq \lambda_a \leq 3$	AABAAABAAABBA	$2 \leq \lambda_a \leq 3$
AABAAABAAB	$2 \leq \lambda_a \leq 3$	AABAAABAABA	$2 \leq \lambda_a \leq 3$
AABAAABAABAA	$2 \leq \lambda_a \leq 3$	AABAAABAABAB	$2 \leq \lambda_a \leq 3$
AABAAABAB	$2 \leq \lambda_a \leq 3$	AABAAABABA	$2 \leq \lambda_a \leq 3$
AABAAABABAA	$2 \leq \lambda_a \leq 3$	AABAAABABB	$2 \leq \lambda_a \leq 3$
AABAAABABBA	$2 \leq \lambda_a \leq 3$	AABAAABB	$2 \leq \lambda_a \leq 3$
AABAAABBA	$2 \leq \lambda_a \leq 3$	AABAAABBAA	$2 \leq \lambda_a \leq 3$
AABAAABBAB	$2 \leq \lambda_a \leq 3$	AABAAABBABA	$2 \leq \lambda_a \leq 3$
AABAAB	$3 \leq \lambda_a \leq 4$	AABAABA	$2 \leq \lambda_a \leq 4$
AABAABAA	$2 \leq \lambda_a \leq 4$	AABAABAAAB	$2 \leq \lambda_a \leq 3$
AABAABAAABA	$2 \leq \lambda_a \leq 3$	AABAABAAABAB	$2 \leq \lambda_a \leq 3$
AABAABAAABABB	$2 \leq \lambda_a \leq 3$	AABAABAAB	$2 \leq \lambda_a \leq 3$
AABAABAABA	$2 \leq \lambda_a \leq 3$	AABAABAABAA	$2 \leq \lambda_a \leq 3$
AABAABAABAB	$2 \leq \lambda_a \leq 3$	AABAABAABB	$2 \leq \lambda_a \leq 3$
AABAABAB	$2 \leq \lambda_a \leq 3$	AABAABABA	$2 \leq \lambda_a \leq 3$
AABAABABAA	$2 \leq \lambda_a \leq 3$	AABAABABAAB	$2 \leq \lambda_a \leq 3$
AABAABABB	$2 \leq \lambda_a \leq 3$	AABAABB	$2 \leq \lambda_a \leq 3$
AABAABBA	$2 \leq \lambda_a \leq 3$	AABAABBAA	$2 \leq \lambda_a \leq 3$
AABAABBAB	$2 \leq \lambda_a \leq 3$	AABAABBBA	$2 \leq \lambda_a \leq 3$
AABAABBBAA	$2 \leq \lambda_a \leq 3$	AABAB	$3 \leq \lambda_a \leq 4$
AABABA	$2 \leq \lambda_a \leq 4$	AABABAA	$2 \leq \lambda_a \leq 4$
AABABAAAB	$2 \leq \lambda_a \leq 3$	AABABAAABA	$2 \leq \lambda_a \leq 3$
AABABAAABAB	$2 \leq \lambda_a \leq 3$	AABABAAB	$2 \leq \lambda_a \leq 3$
AABABAABA	$2 \leq \lambda_a \leq 3$	AABABAABAAB	$2 \leq \lambda_a \leq 3$
AABABAABB	$2 \leq \lambda_a \leq 3$	AABABAB	$2 \leq \lambda_a \leq 3$
AABABABA	$2 \leq \lambda_a \leq 3$	AABABABAA	$2 \leq \lambda_a \leq 3$
AABABABB	$2 \leq \lambda_a \leq 3$	AABABB	$2 \leq \lambda_a \leq 4$
AABABBA	$2 \leq \lambda_a \leq 3$	AABABBAA	$2 \leq \lambda_a \leq 3$
AABABBAAAB	$2 \leq \lambda_a \leq 3$	AABABBAB	$2 \leq \lambda_a \leq 3$
AABABBBA	$2 \leq \lambda_a \leq 3$	AABABBBAA	$2 \leq \lambda_a \leq 3$

AABABBBAB	$2 \leq \lambda_a \leq 3$	AABABBBABBA	$2 \leq \lambda_a \leq 3$
AABABBBABBAB	$2 \leq \lambda_a \leq 3$	AABB	$3 \leq \lambda_a \leq 4$
AABBA	$3 \leq \lambda_a \leq 4$	AABBAA	$2 \leq \lambda_a \leq 4$
AABBAAAB	$2 \leq \lambda_a \leq 3$	AABBAAABA	$2 \leq \lambda_a \leq 3$
AABBAAABAB	$2 \leq \lambda_a \leq 3$	AABBAAABABA	$2 \leq \lambda_a \leq 3$
AABBAAABB	$2 \leq \lambda_a \leq 3$	AABBAAABBA	$2 \leq \lambda_a \leq 3$
AABBAAABBAA	$2 \leq \lambda_a \leq 3$	AABBAAB	$2 \leq \lambda_a \leq 3$
AABBAABA	$2 \leq \lambda_a \leq 3$	AABBAABAAB	$2 \leq \lambda_a \leq 3$
AABBAABAABA	$2 \leq \lambda_a \leq 3$	AABBAABB	$2 \leq \lambda_a \leq 3$
AABBAB	$2 \leq \lambda_a \leq 3$	AABBABA	$2 \leq \lambda_a \leq 3$
AABBABAAAB	$2 \leq \lambda_a \leq 3$	AABBABAB	$2 \leq \lambda_a \leq 3$
AABBABBA	$2 \leq \lambda_a \leq 3$	AABBABBAA	$2 \leq \lambda_a \leq 3$
AABBABBAB	$2 \leq \lambda_a \leq 3$	AABBBA	$2 \leq \lambda_a \leq 3$
AABBBAA	$2 \leq \lambda_a \leq 3$	AABBBAAB	$2 \leq \lambda_a \leq 3$
AABBBAABA	$2 \leq \lambda_a \leq 3$	AABBBAB	$2 \leq \lambda_a \leq 3$
AABBBABA	$2 \leq \lambda_a \leq 3$	AABBBABAB	$2 \leq \lambda_a \leq 3$
AABBBABBBA	$2 \leq \lambda_a \leq 3$	AABBBABBBAB	$2 \leq \lambda_a \leq 3$
AB	$\lambda_a = \infty$	ABA	$\lambda_a = \infty$
ABAAAB	$\lambda_a = 3$	ABAAABA	$2 \leq \lambda_a \leq 3$
ABAAABAAAB	$2 \leq \lambda_a \leq 3$	ABAAABAAABA	$2 \leq \lambda_a \leq 3$
ABAAABAAABAB	$2 \leq \lambda_a \leq 3$	ABAAABAAABBA	$2 \leq \lambda_a \leq 3$
ABAAABAAB	$2 \leq \lambda_a \leq 3$	ABAAABAABA	$2 \leq \lambda_a \leq 3$
ABAAABAABAB	$2 \leq \lambda_a \leq 3$	ABAAABAB	$2 \leq \lambda_a \leq 3$
ABAAABABA	$2 \leq \lambda_a \leq 3$	ABAAABABAAB	$2 \leq \lambda_a \leq 3$
ABAAABABAB	$2 \leq \lambda_a \leq 3$	ABAAABABABA	$2 \leq \lambda_a \leq 3$
ABAAABABBA	$2 \leq \lambda_a \leq 3$	ABAAABBA	$2 \leq \lambda_a \leq 3$
ABAAABBAAB	$2 \leq \lambda_a \leq 3$	ABAAABBAABA	$2 \leq \lambda_a \leq 3$
ABAAABBAB	$2 \leq \lambda_a \leq 3$	ABAAABBABA	$2 \leq \lambda_a \leq 3$
ABAAB	$3 \leq \lambda_a \leq 4$	ABAABA	$2 \leq \lambda_a \leq 4$
ABAABAAAB	$2 \leq \lambda_a \leq 3$	ABAABAAABAB	$2 \leq \lambda_a \leq 3$
ABAABAAB	$2 \leq \lambda_a \leq 4$	ABAABAABA	$2 \leq \lambda_a \leq 3$
ABAABAABAAB	$2 \leq \lambda_a \leq 3$	ABAABAABAB	$2 \leq \lambda_a \leq 3$
ABAABAABABA	$2 \leq \lambda_a \leq 3$	ABAABAABBA	$2 \leq \lambda_a \leq 3$
ABAABAB	$2 \leq \lambda_a \leq 3$	ABAABABA	$2 \leq \lambda_a \leq 3$
ABAABABAAAB	$2 \leq \lambda_a \leq 3$	ABAABABAAB	$2 \leq \lambda_a \leq 3$
ABAABABAABA	$2 \leq \lambda_a \leq 3$	ABAABBA	$2 \leq \lambda_a \leq 3$
ABAABBAAAB	$2 \leq \lambda_a \leq 3$	ABAABBAB	$2 \leq \lambda_a \leq 3$
ABAABBBA	$2 \leq \lambda_a \leq 3$	ABAB	$3 \leq \lambda_a \leq 4$

ABABA	$3 \leq \lambda_a \leq 4$	ABABAAAB	$2 \leq \lambda_a \leq 3$
ABABAAABAAB	$2 \leq \lambda_a \leq 3$	ABABAAABAB	$2 \leq \lambda_a \leq 3$
ABABAAABABA	$2 \leq \lambda_a \leq 3$	ABABAAABBA	$2 \leq \lambda_a \leq 3$
ABABAAB	$2 \leq \lambda_a \leq 3$	ABABAABAAAB	$2 \leq \lambda_a \leq 3$
ABABAABAAB	$2 \leq \lambda_a \leq 3$	ABABAB	$2 \leq \lambda_a \leq 3$
ABABABA	$2 \leq \lambda_a \leq 3$	ABABABAAAB	$2 \leq \lambda_a \leq 3$
ABABABBBA	$2 \leq \lambda_a \leq 3$	ABABBA	$2 \leq \lambda_a \leq 3$
ABABBAAAB	$2 \leq \lambda_a \leq 3$	ABABBABBA	$2 \leq \lambda_a \leq 3$
ABABBABBBA	$2 \leq \lambda_a \leq 3$	ABABBBA	$2 \leq \lambda_a \leq 3$
ABABBBAAB	$2 \leq \lambda_a \leq 3$	ABABBBABA	$2 \leq \lambda_a \leq 3$
ABABBBABBA	$2 \leq \lambda_a \leq 3$	ABABBBABBBA	$2 \leq \lambda_a \leq 3$
ABBA	$3 \leq \lambda_a \leq 4$	ABBAAAB	$2 \leq \lambda_a \leq 3$
ABBAAABAAAB	$2 \leq \lambda_a \leq 3$	ABBAAABBA	$2 \leq \lambda_a \leq 3$
ABBAAB	$2 \leq \lambda_a \leq 3$	ABBAABAAB	$2 \leq \lambda_a \leq 3$
ABBAABBBA	$2 \leq \lambda_a \leq 3$	ABBABAAAB	$2 \leq \lambda_a \leq 3$
ABBABABBA	$2 \leq \lambda_a \leq 3$	ABBABABBBA	$2 \leq \lambda_a \leq 3$
ABBABBA	$2 \leq \lambda_a \leq 4$	ABBABBAAAB	$2 \leq \lambda_a \leq 3$
ABBABBABBA	$2 \leq \lambda_a \leq 3$	ABBABBABBBA	$2 \leq \lambda_a \leq 3$
ABBABBBA	$2 \leq \lambda_a \leq 3$	ABBABBBABBA	$2 \leq \lambda_a \leq 3$
ABBBA	$\lambda_a = 3$	ABBBAAAB	$2 \leq \lambda_a \leq 3$
ABBBABABBBA	$2 \leq \lambda_a \leq 3$	ABBBABBBA	$2 \leq \lambda_a \leq 3$

Table A.1 – List of the 284 patterns from Theorem 6.25.

Note that there are 3 patterns for which there are 3 possible values for the abelian avoidability index: $ABAABA$, $AABAABA$, $AABABA$, $AABABB$, $AABAABAA$, $ABAABAAB$, $ABBABBA$, $AABABAA$, $AABABA$, $AABABB$, $AABBAA$.

Index

Bibliography

[1] J. P. Allouche and J. Shallit. *The Ubiquitous Prouhet-Thue-Morse Sequence*. Springer London, London, 1999.

[2] K. E. Atkinson. *An Introduction to Numerical Analysis*, page 488. J. Wiley, second edition, 1989.

[3] G. Badkobeh. Infinite words containing the minimal number of repetitions. *Journal of Discrete Algorithms*, 20:38 – 42, 2013. StringMasters 2011 Special Issue.

[4] G. Badkobeh and P. Ochem. Characterization of some binary words with few squares. *Theoretical Computer Science*, 588:73–80, 2015.

[5] K. A. Baker, G. F. McNulty, and W. Taylor. Growth problems for avoidable words. *Theoretical Computer Science*, 69(3):319 – 345, 1989.

[6] D.R. Bean, A. Ehrenfeucht, and G.F. McNulty. Avoidable patterns in strings of symbols. *Pacific J. of Math.*, 85:261–294, 1979.

[7] J. Berstel. Axel Thue's papers on repetitions in words: a translation. *Publications du LaCIM 20, Université du Québec à Montréal*, 1995.

[8] F. Blanchet-Sadri and B. Woodhouse. Strict bounds for pattern avoidance. *Theoretical Computer Science*, 506:17–27, 2013.

[9] F. Brandenburg. Uniformly growing k-th power-free homomorphisms. *Theoretical Computer Science*, 23(1):69 – 82, 1988.

[10] A. Carpi. On abelian power-free morphisms. *International Journal of Algebra and Computation*, 03(02):151–167, 1993.

[11] A. Carpi. On the number of abelian square-free words on four letters. *Discrete Applied Mathematics*, 81(1-3):155–167, 1998.

[12] A. Carpi. On Dejean's conjecture over large alphabets. *Theoretical Computer Science*, 385(1):137 – 151, 2007.

[13] J. Cassaigne. Unavoidable binary patterns. *Acta Informatica*, 30(4):385–395.

[14] J. Cassaigne. *Motifs évitables et régularité dans les mots.* PhD thesis, Université Paris VI, 1994.

[15] J. Cassaigne, J. D. Currie, L. Schaeffer, and J. Shallit. Avoiding three consecutive blocks of the same size and same sum. *Journal of the ACM*, 61(2):10:1–10:17, 2014.

[16] R. J. Clark. *Avoidable formulas in combinatorics on words.* PhD thesis, University of California, Los Angeles, 2001.

[17] M. Crochemore. Sharp characterizations of squarefree morphisms. *Theoretical Computer Science*, 18(2):221 – 226, 1982.

[18] J. Currie and N. Rampersad. A proof of Dejean's conjecture. *Math. Comp. 80 (2011), 1063-1070.*

[19] J. D. Currie and N. Rampersad. Fixed points avoiding abelian k-powers. *Journal of Combinatorial Theory, Series A*, 119(5):942–948, July 2012.

[20] J. D. Currie and T. I. Visentin. Long binary patterns are abelian 2-avoidable. *Theoretical Computer Science*, 409(3):432 – 437, 2008.

[21] F. Dejean. Sur un théorème de Thue. *Journal of Combinatorial Theory, Series A*, 13(1):90 – 99, 1972.

[22] F. M. Dekking. Strongly non-repetitive sequences and progression-free sets. *Journal of Combinatorial Theory, Series A*, 27(2):181 – 185, 1979.

[23] R. C. Entringer, D. E. Jackson, and J. A. Schatz. On nonrepetitive sequences. *Journal of Combinatorial Theory, Series A*, 16(2):159 – 164, 1974.

[24] P. Erdős. Some unsolved problems. *The Michigan Mathematical Journal*, 4(3):291–300, 1957.

[25] P. Erdős. Some unsolved problems. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 6:221–254, 1961.

[26] A. A. Evdokimov. Strongly asymmetric sequences generated by a finite number of symbols. *Dokl. Akad. Nauk SSSR*, 179:1268–1271, 1968.

[27] A. S. Fraenkel and R. J. Simpson. How many squares must a binary sequence contain? *The Electronic Journal of Combinatorics*, 2, 1995.

[28] T. Harju and D. Nowotka. Binary words with few squares. *Bulletin of the EATCS*, page 2006.

[29] M. Huova. Existence of an infinite ternary 64-abelian square-free word. *RAIRO - Theoretical Informatics and Applications*, 48(3):307–314, 007 2014.

[30] M. Huova, J. Karhumäki, A. Saarela, and K. Saari. *Local Squares, Periodicity and Finite Automata*, pages 90–101. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[31] M. Huova, J. Karhumäki, and A. Saarela. Problems in between words and abelian words: k-abelian avoidability. *Theoretical Computer Science*, 454:172 – 177, 2012. Formal and Natural ComputingHonoring the 80th Birthday of Andrzej Ehrenfeucht.

[32] J. Justin. Généralisation du théorème de van der Waerden sur les semigroupes répétitifs. *Journal of Combinatorial Theory, Series A*, 12(3):357 – 367, 1972.

[33] J. Karhumaki, A. Saarela, and L. Q. Zamboni. On a generalization of Abelian equivalence and complexity of infinite words. *Journal of Combinatorial Theory, Series A*, 120(8):2189 – 2206, 2013.

[34] V. Keränen. Abelian squares are avoidable on 4 letters. In *ICALP*, pages 41–52, 1992.

[35] V. Keränen. New abelian square-free DT0L-languages over 4 letters. *Manuscript*, 2003.

[36] V. Keränen. A powerful abelian square-free substitution over 4 letters. *Theoretical Computer Science*, 410(38):3893 – 3900, 2009.

[37] M. Lothaire. *Combinatorics on Words*. Cambridge University Press, 1997.

[38] R. Mercaş, P. Ochem, A. V. Samsonov, and A. M. Shur. Binary patterns in binary cube-free words: Avoidability and growth. *RAIRO - Theoretical Informatics and Applications*, 48(4):369–389, 2014.

[39] R. Mercaş and A. Saarela. *3-Abelian Cubes Are Avoidable on Binary Alphabets*, pages 374–383. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[40] R. Mercaş and A. Saarela. 5-abelian cubes are avoidable on binary alphabets. *RAIRO - Theoretical Informatics and Applications*, 48(4):467–478, 10 2014.

[41] M. Noori and J. D. Currie. Dejean's conjecture and Sturmian words. *European Journal of Combinatorics*, 28(3):876 – 890, 2007.

[42] P. Ochem. A generator of morphisms for infinite words. *RAIRO - Theoretical Informatics and Applications*, 40:427–441, 2006.

[43] P. Ochem. Binary words avoiding the pattern AABBCABBA. *RAIRO - Theoretical Informatics and Applications*, 44(1):151–158, 2010.

[44] P. Ochem. Doubled patterns are 3-avoidable. *The Electronic Journal of Combinatorics.*, 23(1), 2016.

[45] P. Ochem and A. Pinlou. Application of entropy compression in pattern avoidance. *The Electronic Journal of Combinatorics.*, 21(2), 2014.

[46] P. Ochem and M. Rosenfeld. *Avoidability of Formulas with Two Variables*, pages 344–354. Springer Berlin Heidelberg, 2016.

[47] J. Ollagnier. Proof of Dejean's conjecture for alphabets with 5, 6, 7, 8, 9, 10 and 11 letters. *Theoretical Computer Science*, 95(2):187 – 205, 1992.

[48] J. Pansiot. A propos d'une conjecture de F. Dejean sur les répétitions dans les mots. *Discrete Applied Mathematics*, 7(3):297 – 311, 1984.

[49] G. Pirillo and S. Varricchio. On uniformly repetitive semigroups. *Semigroup Forum*, 49(1):125–129, 1994.

[50] P. A. B. Pleasants. Non-repetitive sequences. *Mathematical Proceedings of the Cambridge Philosophical Society*, 68:267–274, 9 1970.

[51] N. Rampersad, J. Shallit, and M. Wang. Avoiding large squares in infinite binary words. *Theoretical Computer Science*, 339(1):19 – 34, 2005.

[52] M. Rao. Last Cases of Dejean's Conjecture. *Theoretical Computer Science*, 412(27):3010–3018, June 2011.

[53] M. Rao. On some generalizations of abelian power avoidability. *Theoretical Computer Science*, 601:39 – 46, 2015.

[54] M. Rao and M. Rosenfeld. Avoiding two consecutive blocks of same size and same sum over $\mathbb{Z}^2$. *ArXiv e-prints*, November 2015.

[55] M. Rao and M. Rosenfeld. Avoidability of long k-abelian repetitions. *Mathematics of Computation*, 2016.

[56] M. Rosenfeld. Every binary pattern of length greater than 14 is abelian-2-avoidable. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, pages 81:1–81:11, 2016.

[57] P. Roth. Every binary pattern of length six is avoidable on the two-letter alphabet. *Acta Informatica*, 29(1):95–107.

[58] U. Schmidt. Avoidable patterns on two letters. *Theoretical Computer Science*, 63(1):1 – 17, 1989.

[59] A. Thue. Über unendliche Zeichenreihen. *'Norske Vid. Selsk. Skr. I. Mat. Nat. Kl. Christiania*, 7:1–22, 1906.

[60] A. Thue. Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen. *Norske Vid. Selsk. Skr. I. Mat. Nat. Kl. Christiania,*, 10:1–67, 1912.

[61] A. I. Zimin. Blocking sets of terms. *Sbornik: Mathematics*, 47(2):353–364, 1984.