TUCS

Mari Huova

Combinatorics on Words

New Aspects on Avoidability, Defect Effect, Equations and Palindromes

Turku Centre *for* Computer Science

TUCS Dissertations
No 172, April 2014

Combinatorics on Words

New Aspects on Avoidability, Defect Effect, Equations and Palindromes

Mari Huova

To be presented, with the permission of the Faculty of Mathematics and Natural Sciences of the University of Turku, for public criticism in Auditorium Edu 1 on April 11, 2014, at 12 noon.

University of Turku
Department of Mathematics and Statistics
FI-20014 Turku
Finland

2014

Supervisors

Professor Juhani Karhumäki
Department of Mathematics and Statistics
University of Turku
FI-20014 Turku
Finland

Professor Tero Harju
Department of Mathematics and Statistics
University of Turku
FI-20014 Turku
Finland

Reviewers

Professor Dirk Nowotka
Department of Computer Science
Kiel University
24098 Kiel
Germany

Professor Gwénaël Richomme
UFR VI, Dpt MIAp, Case J11
Université Paul-Valéry Montpellier 3
Route de Mende
34199 Montpellier Cedex 5
France

Opponent

Professor Ion Petre
Department of Information Technologies
Åbo Akademi University
FI-20520 Turku
Finland

ISBN 978-952-12-3034-9
ISSN 1239-1883

Abstract

In this thesis we examine four well-known and traditional concepts of combinatorics on words. However the contexts in which these topics are treated are not the traditional ones. More precisely, the question of avoidability is asked, for example, in terms of *k-abelian squares.* Two words are said to be k-abelian equivalent if they have the same number of occurrences of each factor up to length k. Consequently, k-abelian equivalence can be seen as a sharpening of abelian equivalence. This fairly new concept is discussed broader than the other topics of this thesis.

The second main subject concerns *the defect property.* The defect theorem is a well-known result for words. We will analyze the property, for example, among the sets of *2-dimensional words*, i.e., polyominoes composed of labelled unit squares.

From the defect effect we move to *equations.* We will use a special way to define a product operation for words and then solve a few basic equations over constructed *partial semigroup.* We will also consider the satisfiability question and the compactness property with respect to this kind of equations.

The final topic of the thesis deals with *palindromes.* Some finite words, including all binary words, are uniquely determined up to word isomorphism by the position and length of some of its palindromic factors. The famous *Thue-Morse* word has the property that for each positive integer n, there exists a factor which cannot be generated by fewer than n palindromes. We prove that in general, every non ultimately periodic word contains a factor which cannot be generated by fewer than 3 palindromes, and we obtain a classification of those binary words each of whose factors are generated by at most 3 palindromes. Surprisingly these words are related to another much studied set of words, *Sturmian words.*

Tiivistelmä

Tässä väitöskirjassa tutkitaan neljää perinteistä ja tunnettua sanojen kombinatoriikan kysymystä. Lähestymistavat kysymyksiin eivät kuitenkaan ole perinteisiä. Esimerkiksi toistojen välttämistä sanoissa tarkastellaan käyttäen *k-Abelin ekvivalenssin* käsitettä. Sanat ovat k-Abelin ekvivalentteja, jos niillä on sama lukumäärä jokaista korkeintaan k kirjaimen pituista tekijää. Perinteisen Abelin ekvivalenssin voidaankin ajatella olevan k-Abelin ekvivalenssin erikoistapaus. Uudehkoa k-Abelin ekvivalenssin käsitettä tarkastellaan muita väitöskirjan aiheita laajemmin.

Toinen pääaiheista on *defektiominaisuus.* Sanojen defektiominaisuus on hyvin tunnettu asia, mutta tässä työssä defektiominaisuuden olemassaoloa tarkastellaan muun muassa *kaksiulotteisten sanojen* joukoissa. Kaksiulotteiset sanat ovat *leimatuista yksikköneliöistä* muodostettuja monikulmioita.

Defektiominaisuuden jälkeen käsitellään *yhtälöitä.* Näitä varten määritellään sanojen tulo-operaatio tavallisesta poikkeavalla tavalla ja samalla muodostetaan myös joukko uudenlaisia *osittaisia puoliryhmiä.* Näissä puoliryhmissä ratkotaan sanojen perusyhtälöitä ja tarkastellaan toteutuvuus- ja kompaktisuuskysymyksiä.

Viimeinen väitöskirjan aihe koskee *palindromeja.* Osa äärellisistä sanoista, kuten esimerkiksi kaikki kaksikirjaimiset sanat, voidaan määrittää sanaisomorfiaa vaille yksikäsitteisesti kunkin sanan palindromitekijöiden sijaintien ja pituuksien perusteella. Voidaan osoittaa, että tunnetulla *Thuen-Morsen* sanalla on kaikilla luvun n arvoilla sellainen tekijä, jota ei voida määrittää käyttämättä vähintään n palindromia. Voidaan myös todistaa, että jokainen ääretön sana, joka ei ole lopultakaan jaksollinen, sisältää tekijän, jota ei voida määrittää käyttämättä vähintään kolmea palindromia. Toisaalta sellaiset äärettömät kaksikirjaimiset sanat, jotka eivät ole lopultakaan jaksollisia ja joiden jokaisen tekijän määrittämiseen riittää kolme palindromitekijää, pystytään karakterisoimaan. Nämä sanat ovat yllättäen yhteydessä paljon tutkittuihin *Sturmin sanoihin.*

Acknowledgements

First of all, I want to bestow my gratitude to my great advisors. Professor Juhani Karhumäki introduced this area of discrete mathematics to me when I was doing my Master's Thesis. Since then he has guided me through my doctoral studies, through my first shy steps as a researcher. He has encouraged me and given me many fascinating problems and ideas how to carry on my research. Professor Tero Harju has been a supervisor of my doctoral studies but, in addition, I can see him like an advisor, too. It has been a great pleasure to have such a person as a supervisor and a co-author.

I am grateful for professors Dirk Nowotka and Gwénaël Richomme who have reviewed my thesis and given me many constructive comments. These have improved my manuscript a lot and in many ways. I also want to thank professor Luca Q. Zamboni for reading parts of the manuscript of this thesis and improving the wording significantly. I am greatly thankful for professor Ion Petre. It is an honor for me that he agreed to act as the opponent.

I want to thank all of my co-authors, especially Aleksi Saarela who has had great ideas for our collaboration. I am thankful for the research institute Turku Centre for Computer Science and University of Turku Graduate School for all the support for my studies, research and thesis. I also want to thank Vilho, Yrjö and Kalle Väisälä Foundation and the Academy of Finland for their financial support. I appreciate the good and inspiring working environment the Department of Mathematics and Statistics has offered and I want to express my thanks for that. I want to give my special acknowledgements to the administrative staff for always helping with and taking care of practical matters.

I have been privileged to have the staff of this particular department, Department of Mathematics and Statistics, as my colleagues. I have always felt welcome to every meeting and event. For example, I have good memories of Venla Relay in which I participated with former and present members of the department. I have had many fruitful mathematical and non-mathematical discussions, particularly with my room mates. I want to thank Anne Seppänen, that has been my room mate from the very beginning, for her mental support. I have also found many good friends from the lecture rooms of the department during the last nine years.

For the precious and the most long-lasting support I want to thank my parents and family. They have encouraged me to do my best but at the same time reminded me that I have to enjoy the things I do.

Finally, I give my greatest thanks to Toni for everything, especially believing in me at the times when I did not.

Turku, March 2014 Mari Huova

List of original publications

The thesis is based on the following papers:

1. T. Harju, M. Huova and L. Q. Zamboni. On Generating Binary Words Palindromically, manuscript in arXiv:1309.1886 (submitted).

2. M. Huova. A Note on Defect Theorems for 2-Dimensional Words and Trees. *Journal of Automata, Languages and Combinatorics*, 14(3-4):203–209, 2009.

3. M. Huova. Existence of an Infinite Ternary 64-Abelian Square-free Word. In R. Jungers, V. Bruyère, M. Rigo and R. Hollanders, editors, *RAIRO - Theoretical Informatics and Applications: Selected papers dedicated to Journees Montoises d'Informatique Theorique 2012*, (to appear).

4. M. Huova and J. Karhumäki. Equations in the Partial Semigroup of Words with Overlapping Products. In H. Bordihn, M. Kutrib and B. Truthe, editors, *Dassow Festschrift*, *LNCS* 7300:99–110. Springer, 2012.

5. M. Huova and J. Karhumäki. On k-Abelian Avoidability. In *Combinatorics and graph theory. Part IV, RuFiDiM'11, Zap. Nauchn. Sem. POMI* 402:170–182. POMI, 2012. Translation of the volume in: *J. Math. Sci. (N. Y.)*, 192(3):352–358, 2013.

6. M. Huova and J. Karhumäki. On Unavoidability of k-abelian Squares in Pure Morphic Words. *Journal of Integer Sequences*, 16(2):13.2.2, 2013.

7. M. Huova, J. Karhumäki and A. Saarela. Problems in between words and abelian words: k-abelian avoidability. *Theoret. Comput. Sci.*, 454:172–177, 2012.

8. M. Huova, J. Karhumäki, A. Saarela and K. Saari. Local Squares, Periodicity and Finite Automata. In C. S. Calude, G. Rozenberg and

A. Salomaa, editors, *Rainbow of Computer Science*, *LNCS* 6570:90–101. Springer, 2011.

9. M. Huova and A. Saarela. Strongly k-Abelian Repetitions. In J. Karhumäki, A. Lepistö and L. Q. Zamboni, editors, *Combinatorics on Words* (Proceedings for the 9th International Conference WORDS 2013), *LNCS* 8079:161–168. Springer, 2013.

Contents

Chapter 1

Introduction

In this chapter we present an introduction to combinatorics on words, an area of discrete mathematics and the subject of this thesis. In the first section we describe the development of this area and consider some connections to other fields of mathematics and to other branches of sciences. The next Section 1.2 presents an outline of the structure of this thesis and in the last section of this chapter we give the basic definitions and notions of combinatorics on words relevant to the thesis.

1.1 Background

Combinatorics on words is a fairly new area of discrete mathematics which began in the beginning of the 20'th century. Among the first publications were the Axel Thue's avoidability results which were published in 1906 and 1912, [91, 92]. In those papers Thue showed, for example, the existence of an infinite binary cube-free word and an infinite ternary square-free word. In other words, the existence of an infinite word over a binary (resp. ternary) alphabet which does not contain an occurrence of three (resp. two) consecutive equal blocks. Besides the fact that avoidability questions have been one of the first questions considered in combinatorics on words, the theory of avoidability is also one of the most widely studied topics in the area. Nowadays, the results of Thue are well-known but it has not always been the same. Many of the Thue's results have been rediscovered during the 20'th century like, for example, in the paper by Morse and Hedlund 1944 [80].

Research in combinatorics on words has been active and systematic since the 1950's, much later than the beginning of the study of the area. For example, M.-P. Schützenberger studied the area in connection to information theory and computing, see [8], and P. S. Novikov with S. Adian used combinatorics on words to give a solution to the Burnside problem for groups, see [2]. One important year for this area of discrete mathematics is 1983, the year in which the first book of the field was published. It was the book *Combinatorics on words* by M. Lothaire [68], which was a presentation of the research done so far. Lothaire has also tried to gather the results of newer studies into two other volumes, *Algebraic combinatorics on words* [69] in 2002 and *Applied combinatorics on words* [70] in 2005. The last one concentrates on algorithmical questions.

As the names of the Lothaire's latter books suggest, combinatorics on words has many connections to other areas of mathematics. The connections to automata theory and formal languages are clear because in both fields one operates with words. A set of binary infinite words, so called Sturmian words, have several equivalent definitions and these provide an interesting link between combinatorics, number theory and dynamical systems. Also several algebraic based questions have been reformulated and solved in terms of words. If the words of finite length over a fixed alphabet are considered then the set of these words with the concatenation operation form an algebraic structure, a semi-group. If the empty word, a word without any letters, is added to this set of words as a neutral element, then this set becomes a monoid. In fact, they represent free semi-groups and free monoids over the given alphabet. This means that the elements of these sets can be uniquely factorized by the letters of the alphabet. Combinatorics on words has not only connections to different branches of mathematics but also to other sciences. In computer science the information that computers process is considered to be sequences of two different symbols, 0 and 1, that is binary words. Another example of combining theory of words with an other science is the connection to biology and DNA sequencing. [69, 70]

1.2 Structure of the thesis

In this thesis we consider several of the well-known notions and questions on combinatorics on words. We study *repetitions* and words that avoid repetitions, *defect property*, *word equations*, *palindromes* and to some minor extent, *Sturmian words*. For each of these subjects we have a somewhat new approach. To examine repetitions and *avoidability* we define a set of new equivalence classes, *k-abelian equivalences*. Questions on these newly defined equivalence classes, on repetitions and on avoidance constitute the major part of this thesis reflecting the important role k-abelian equivalence has played in research for this thesis and my postgraduate studies. The defect effect will be considered on the sets of *2-dimensional words* instead of the sets of usual 1-dimensional words. Word equations will be defined over a partial semigroup of words with a new kind of product, so called *overlapping product*. There also exist a few common questions on the number of palindromes, for example, we may ask how many palindromes or different palindromes a word has. Instead, in this thesis we are interested in words that can be determined up to word isomorphism by their palindromic factors. This means that if we know the palindromic factors we can construct the initial word. In this context we discover a new connection between palindromes and Sturmian words and, in fact, a connection to a broader class of words which we call *double Sturmian words.*

Next we introduce the content of this thesis in more details and also give an overview of the settings of more traditional versions of the problems we are interested in. As mentioned, the theory of avoidability is among the oldest and most studied topics in combinatorics on words. The first results, *squares* are avoidable over a ternary alphabet and *cubes* are avoidable over a binary alphabet, were obtained by Thue [91, 92]. Avoidability over an alphabet means that there exists such an infinite word over the alphabet which does not contain a factor of the given form, for example, a square. Thue's results covered the cases in which the words are regarded as ordered sequences of letters. This means that two words are the same if and only if they have the same number of each letter and in the exactly same order.

Since late 1960's commutative variants of similar avoidability problems have been studied. Commutation is achieved if the order of letters in a word is set to be irrelevant. Then two words are considered to be equivalent if they have the same number of each letter. This equivalence relation is called *abelian equivalence.* For example, the words *aab* and *aba* are abelian equivalent and thus the word *aababa* is an abelian square. An infinite word avoids abelian squares (resp. cubes) if it dos not contain two (resp. three) consecutive abelian equivalent factors. The first non-trivial results were obtained by Evdokimov [34] who showed that abelian squares can be avoided in an infinite word over a 25-letter alphabet. The size of the alphabet was reduced to five by Pleasant [85], until the optimal value, four, was found by Keränen [62] in 1992. The optimal value for the size of the alphabet in which abelian cubes are avoidable was proved already earlier by Dekking [27]. He showed that abelian cubes are avoidable over a ternary alphabet.

We consider here new variants of the avoidability problems by defining repetitions via new equivalence relations. These equivalences lie properly in between equality and commutative abelian equality. Thus the original results on avoiding repetitions and abelian repetitions give the background for these new avoidability problems. In abelian equivalence the number of each letter is determinative but in these new equivalences, called *k-abelian equivalences* where k is a natural number, the number of occurrences of each factor of length k is significant. We require that the words in the same k-abelian equivalence class have the same number of each factor of length k and the same prefix and suffix of length $k-1$. Because of the latter requirement two k-abelian equivalent words are abelian equivalent, too. In addition, now this equivalence relation also satisfies the definition of *a congruence*. This means that concatenation is well defined on the level of equivalence classes. Chapter 2 is devoted to the study of this notion of k-abelian equivalence. The main target is avoidability questions but we begin with results on characterizations of these classes and on the numbers and sizes of the classes.

Most of the avoidability results on words and on abelian words are obtained by constructing a desired infinite word by iterating a suitable *morphism*. A morphism is a mapping h that satisfies the condition $h(xy) = h(x)h(y)$. If the morphism is *prefix preserving*, that is there exists a letter a such that $h^i(a)$ is always a prefix of $h^{i+1}(a)$, then the morphism can be iterated infinitely many times and the iteration produces a well defined unique word as the limit of this procedure. A well-known example of this kind of an infinite word is the *Fibonacci word*. It is obtained by iterating a morphism $f(0) = 01, f(1) = 0$ infinitely many times starting at 0. The Fibonacci word begins with 0100101001001010. In Chapter 2 we show, for example, that surprisingly it is not possible to generate an infinite ternary word by iterating one morphism so that the word would avoid k-abelian squares for any $k \geq 1$.

The Fibonacci word is also an example of *a Sturmian words*. These infinite binary words may be characterized in many ways and one possibility is to count the number of different factors of the word. If the word has exactly $n + 1$ different factors of length n, for each natural number n, then the word is a Sturmian word. A characterization can also be given in terms of k-abelian complexity function. By Karhumäki, Saarela and Zamboni, [59], we can use the number of k-abelian equivalence classes of the factors of the word to characterize the Sturmian words. This will be stated in the end of Chapter 2, as well as a few other general results on k-abelian equivalence. We will also come back to Sturmian words in the last chapter of the thesis which deals with connections between Sturmian words and *palindromes*.

An other question that is studied to a large extent in combinatorics on words is periodicity of words. *A period* of a word $w = w_1 \cdots w_m$ is such a natural number $p \leq m$ that w is a factor of $(w_1 \cdots w_p)^m$. Clearly, each finite word has at least one period, namely the length of the word. In periodicity questions for infinite words it is asked whether there exists a finite word v such that the given infinite word w is of the form $w = v^\omega$, where v^ω means that the word v is repeated infinitely many times. If a word w is *ultimately periodic* then there exist finite words u and v such that $w = uv^\omega$. The words that are not ultimately periodic are called *aperiodic*.

A well-known result of Fine and Wilf [36] deals with periods of words. It says that if a word w has periods p and q such that the greatest common divisor (gcd) of p and q is d and the length of the word is at least $p + q - d$ then the word w has a period d, too. In addition, there are words of length $p+q-d-1$ that have periods p and q but not the period d. The corresponding bounds for k-abelian periods can also be considered as in [57]. On the other hand, the theorem of Fine and Wilf can also be stated as: two words u and v are powers of the same word if and only if u^ω and v^ω has a common prefix of length at least $|u| + |v| - \gcd(|u|, |v|)$. Another simple condition for two words to be powers of a same word is *commutation.* If two words x and y commute, i.e., $xy = yx$ then there exists a word z such that $x = z^i$ and $y = z^j$ for some natural numbers i and j. In fact, the commutation is the simplest case of the so called *defect theorem* which is considered in Chapter 3.

In general, the defect theorem says that if a set of n words satisfies a non-trivial relation then there exists a set of $n-1$ words such that each word of the bigger set can be obtained by concatenating some of the words of the smaller set. A natural question is to consider whether there exist different structures, not just sets of words, for which the defect theorem would also hold. This question is considered, for example, in the paper by Harju and Karhumäki, [49]. We can also again consider k-abelian equivalence but this time with respect to defect effect. This question is shortly studied in Chapter 3 but the main subject of the chapter is to study the defect theorem in the context of *two-dimensional words.*

A word $w = a_1a_2\cdots a_n$ where a_i's are letters can be considered to be a one-dimensional word where each point $(i, 0) \in \mathbb{N} \times \{0\}$ has a label a_i for all $1 \leq i \leq n$. For a two-dimensional word we consider also points of the form $(i, j) \in \mathbb{N} \times \mathbb{N}$, where both i and j may be greater than 0. In other words, we analyze different polyominoes constructed from labelled unit squares. In general, the defect theorem does not hold for two-dimensional words but there exist small restricted sets of polyominoes for which the defect theorem holds. The defect theorem for these two-dimensional words has been studied earlier by, for example, Harju and Karhumäki in [49] and Moczurad in [79].

The defect is at times useful in solving word *equations.* A word equation consists of unknowns which are elements of some finite alphabet X and possibly some constants which are letters from the original alphabet Σ. For example, $xab = byb$ is a word equation where $\Sigma = \{a, b\}$ and $X = \{x, y\}$. *A solution* to a word equation is a morphism from the set $(X \cup \Sigma)^*$ to the set Σ^* which maps both sides of the equation to the same word, and for letters in Σ the morphism is an identity mapping. A solution to the given equation $xab = byb$ is a morphism which maps x to a word $b\alpha$ and y to a word αa, where $\alpha \in \Sigma^*$. Another simple word equation is the already mentioned commutation rule $xy = yx$. This is an example of an equation that does not contain any constants.

In Chapter 4 we discuss a few basic word equations and their solutions not only in a way introduced above, but also with a different definition of the word product. This so called *overlapping product* is motivated by bio-operations, see e.g. [83]. We show how these word equations in the partial semigroup with overlapping products can be transformed into problems on ordinary word equations. In addition, we show that *the satisfiability problem* is solvable also for these new kind of word equations. In the satisfiability problem it is asked whether the existence of a solution for a given equation is decidable. It was shown by Makanin in [72] that for ordinary word equations the existence of a solution is decidable. In the end of this 4th chapter we once again give a few simple remarks of k-abelian equivalence, now in connection with solving equations.

In the last chapter we need again the result of Fine and Wilf and the notion of Sturmian words. The central notion for this chapter is *a palindrome*. A palindrome is a word which can be read in both directions, from left to right and from right to left, ending up to the same word. For example, *abaabaaba* is a palindrome. The main subject is to study a new connection between Sturmian words and palindromes. It is known, for example, that the number of different palindromic factors of a finite Sturmian word of length n is $n + 1$ and on the other hand, for an infinite Sturmian word the number of palindromic factors of length m is 1 for even values of m and 2 for odd values of m, see e.g. [31, 32].

We will approach words and their palindromic factors in a bit different way. We examine words that can be generated up to isomorphism by giving the lengths and positions of their palindromic factors. For example, the word *abaabaaba* is uniquely determined up to word isomorphism by the fact that it is a palindromic word of 9 letters and the prefix of length 6 is also a palindrome. Of course, the word *aaaaaaaaa* would also satisfy the conditions but in *abaabaaba* the letters are chosen as freely as possible. It is easy to show that each binary word can be defined by palindromes. On the other hand, we will show that all the binary words that are aperiodic and can be generated from the information of its three palindromic factors are related to Sturmian words.

Because the content of the thesis is diverse and each chapter to some extent has its own independent topic, each chapter is ended with a short conclusion section. In summary, the thesis is organized as follows: Chapter 2 introduces k-abelian equivalences and avoidability questions, Chapter 3 deals with the defect theorem, Chapter 4 concentrates on equations over special partial semigroups of words and the last Chapter 5 studies palindromes and Sturmian words.

1.3 Basic definitions and notions

In this section we introduce basic notions and definitions of combinatorics on words needed for this thesis. Some additional concepts are given in later chapters but the following notions are used throughout the thesis. For a more comprehensive presentation of basic notions see, e.g. [68, 20]. Let Σ be a finite set called *the alphabet* and the elements of the alphabet are called *letters. A word* is a sequence of letters and it can be finite or infinite. In fact, infinite words can be one-way infinite or two-way infinite, meaning that the word either has a starting point and is infinite in one direction or that it is infinite in both directions without any specific starting point. In this thesis we concentrate on finite and one-way infinite words. The finite words are of the form $w = a_1a_2a_3\cdots a_n$ and infinite ones of the form $w = a_1a_2a_3\ldots$ where $a_i \in \Sigma$. The word that does not contain any letters is called the *empty word* and denoted by ϵ.

The set of all finite non-empty words over Σ is denoted by Σ^+. As mentioned it can be viewed as a free semigroup with respect to the product operation defined by concatenation. The empty word is an identity element and if it is added to Σ^+ then a freely generated monoid denoted by Σ^* is obtained. The generators for Σ^* are the letters in Σ. The set of one-way infinite words over the alphabet Σ is denoted by Σ^ω. The mentioned product operation for words is the same as concatenation and because it is associative it is usually not indicated with any sign. For example, the product of words *aab* and *abb* is *aababb*. If the word w is of the form $w = xy$, where $y \neq \epsilon$, then $wy^{-1} = xyy^{-1} = x$ means that the word y is deleted from the end of the word w. This deletion is a partial operation.

A finite word $w \in \Sigma^+$ has *a factor* v if the word can be written as $w = u_1vu_2$ so that $u_1, u_2 \in \Sigma^*$, and if at least one of them is not empty then v is *a proper factor*. For an infinite word w and a finite factor v the corresponding u_2 is infinite. A factor v of the word w is *a prefix* if $u_1 = \epsilon$ and *a suffix* if $u_2 = \epsilon$. A factor is *a proper prefix* or *a proper suffix* if it is a proper factor and a prefix or a suffix respectively. The set of all factors of a word w is denoted by $F(w)$ and the set of all prefixes (resp. suffixes) is $\mathrm{pref}(w)$ (resp. $\mathrm{suf}(w)$). The number of letters in a finite word w is called the *length* of the word and denoted by $|w|$. The number of occurrences of a factor v in a word w is denoted by $|w|_v$. A prefix (resp. suffix) of length n of a word w is denoted by $\mathrm{pref}_n(w)$ (resp. $\mathrm{suf}_n(w)$) and if $n > |w|$ then $\mathrm{pref}_n(w) = \mathrm{suf}_n(w) = w$. The set of factors of length n is $F_n(w) = F(w) \cap \Sigma^n$, where Σ^n denotes all the words of length n over the alphabet Σ.

We say that a word w is *a square* if it equals to $vv = v^2$ for some $v \in \Sigma^+$. Other terms used to describe this kind of words are *a repetition of order 2* and *the second power* of v. A word v^3 is *a cube* or equivalently *a repetition of order 3* or *the third power* of v. Respectively, in general, v^n is *a repetition of order* n or *the* n*th power* of v. The powers can be defined for fractional numbers but we concentrate on powers that are natural numbers. Fractional powers are not so straightforward to determine in the perspectives in which we study the powers. If a word has a repetition of order n as a factor then the word is said to *contain* a repetition of order n or, for example, a square in case $n = 2$. On the other hand, if a word does not contain a repetition of order n then it is said to *avoid* this repetition. We also use terms *square-free* and *cube-free* to denote words that avoid squares and cubes, respectively.

In general, we can define *a pattern* and ask whether a given word avoids the pattern. A pattern is a word over an alphabet containing *variables.* For example, let α and β be variables then the pattern associated with squares is just $\alpha\alpha$ and another example of a pattern is $\alpha\beta\alpha\beta\alpha$. This latter example is related to the notion of *overlap.* For example, a word $w_1abcabcaw_2$ contains an overlap because the first shown occurrence of the factor $abca$ overlaps with the second one. So the factors that have a prefix also as a suffix may overlap. Now in the given example $w_1abcabcaw_2$ a corresponds to α and bc corresponds to β. We say that if a word avoids the pattern $\alpha\beta\alpha\beta\alpha$ then the word is *overlap-free.*

If there exists an infinite word over an alphabet Σ which avoids, for example, squares then it is said that the alphabet avoids squares and that squares are avoidable over the alphabet Σ. In avoidability questions it is usually asked what is the smallest size of the alphabet that avoids a given pattern, if such exists. There exist many variations of avoidability questions. For example, *abelian squares* are defined to be words w that can be written in form $w = uv$ where u and v are *abelian equivalent*, i.e., they have the same number of each letter. In other words abelian equivalent words have the same letters but possibly in different order, i.e., they are anagrams. For instance, $abbabbaa$ is an abelian square. *Abelian cubes* and *abelian nth powers* are defined analogously. Now it is natural to examine words that avoid abelian nth powers.

Let w be a finite word of length n, i.e. $w = a_1a_2\cdots a_n$. *A period* of the word is an integer p for which $1 \leq p \leq n$ and $a_i = a_{i+p}$ for all $1 \leq i \leq n-p$. Every finite word w has at least one period since at least the length of the word is a period. The least period of the word is called *the period* of the word. A finite word can always be written in the form $w = u^n$ for some $u \in \Sigma^+$ and a natural number n. If $n = 1$ is the only possibility then the word is called *primitive.* If an infinite word w can be written in the form $w = u^\omega$ for some $u \in \Sigma^+$ then it is *a periodic word*, and *an ultimately periodic* word can be written in the form $w = uv^\omega$ for some $u \in \Sigma^*$ and $v \in \Sigma^+$. If a word is not ultimately periodic, then it is called *aperiodic.*

A mapping h from a free monoid A^* into a free monoid B^* is called *a morphism* if it satisfies the condition $h(xy) = h(x)h(y)$ for all $x, y \in A^*$. In combinatorics on words we often use morphisms $h : \Sigma^* \to \Sigma^*$. To define how h maps the words of Σ^* it is enough to give the images of the letters. A morphism h is called *uniform* if the images of the letters have the same length, i.e., $|h(a)| = |h(b)|$ for each $a, b \in \Sigma$. *A prefix preserving* (or *prolongable*) *morphism* is a morphism $h : \Sigma^* \to \Sigma^*$ for which there exists a letter $a \in \Sigma$ and a word $\alpha \in \Sigma^*$ such that $h(a) = a\alpha$ and $h^n(\alpha) \neq \epsilon$ for every $n \geq 0$. This means that $h^i(a)$ is always a prefix of $h^{i+1}(a)$. So the word $h^\infty(a)$ is well defined as a limit of iterating $h(a)$ infinitely many times. We call an infinite word *a pure morphic word* if it is obtained by iterating a prefix preserving morphism. *A morphic word* is obtained from a pure morphic word by taking an image of it by a morphism or equivalently under a coding, see [4]. Two well-known morphisms are f and t from the set of binary words into itself defined as follows:

$$f : \begin{cases} 0 \to 01 \\ 1 \to 0 \end{cases} \quad \text{and} \quad t : \begin{cases} 0 \to 01 \\ 1 \to 10 \end{cases}$$

The infinite word $f^\infty(0)$ which is the unique fixed-point of f is called the *Fibonacci word* and the fixed-point of t starting at 0, i.e., the infinite word $t^\infty(0)$ is called the *Thue-Morse word.*

We say that two words u and v *commute* if $uv = vu$. If $u = q^i$ and $v = q^j$ then they clearly commute because $uv = q^{i+j} = vu$. In fact, this property characterizes the words that commute, see e.g. [68]. Another basic property of words is *conjugation.* Words u and v are conjugates if there exist words p and q such that $u = pq$ and $v = qp$. In other words u and v are conjugates if they satisfy $uz = zv$ for some $z \in \Sigma^*$. In fact, now $z = (pq)^i p$ for some $i \geq 0$ in terms of p and q, see e.g. [68]. The words that are conjugates of each other constitute *a conjugacy class.* This set can be composed from the words that are obtained from each other by a cyclic permutation $c : \Sigma^* \to \Sigma^*$, where $c(aw) = wa$ for $a \in \Sigma$ and $w \in \Sigma^*$. For example, $\{aabc, abca, bcaa, caab\}$ is clearly a conjugacy class.

The expressions $uv = vu$ and $uz = zv$ are also examples of *word equations.* In these equations there are no *constants* and the equations involve only *variables* of the alphabet X. If the equation has constants then the alphabet for the equation is a union of an alphabet of variables and a disjoint alphabet of constants Σ. *A solution* to an equation $u = v$, where $u, v \in (X \cup \Sigma)^+$, is a morphism $e : (X \cup \Sigma)^* \to \Sigma^*$ such that $e(u) = e(v)$ and $e(a) = a$ for all $a \in \Sigma$. For example, a solution to the commutation equation $uv = vu$ is a morphism that maps u to q^i and v to q^j for some $i, j \geq 0$ and $q \in \Sigma^*$.

The Fibonacci word defined above is an example of a *Sturmian word.* As mentioned, if an infinite word w has exactly $n + 1$ different factors of length n for each natural number n, i.e., *the factor complexity* $\rho_w(n)$ of the word is $n + 1$, then the word is a Sturmian word. It is clear that Sturmian words are binary words having two different factors of length one, that is the letters of the alphabet. There exist many equivalent definitions for Sturmian words, see e.g. [69]. For example, if w is an infinite binary word over an alphabet $\{0, 1\}$ then w is Sturmian if and only if w is balanced and aperiodic. Here *a balanced word* means a binary word over $\{0, 1\}$ for which

$|x|_1 - |y|_1 \in \{-1, 0, 1\}$ for each factors x and y of the same length. In other words, in a balanced word the number of letter 1 in each factor of length n is either i or $i+1$, for some $0 \leq i \leq n-1$.

The lexicographic order, denoted by $<_l$, is defined as follows. First, let the letters of alphabet Σ be ordered, i.e., Σ is a finite ordered set. Let $x, y \in \Sigma^+$ be two words and let u be the longest common prefix of x and y. Now we have $x <_l y$ if $x = u$ and $y = uby'$ or if $x = uax'$ and $y = uby'$, for $x', y' \in \Sigma^*$ and $a, b \in \Sigma$ with $a <_l b$. This corresponds to the usual dictionary order. One special class of words is *Lyndon words.* A Lyndon word is a primitive word which is minimal in its conjugacy class with respect to the *lexicographic order.* As an example, we give a few shortest Lyndon words over an alphabet $\{0, 1\}$ with $0 <_l 1$: $\epsilon, 0, 1, 01, 001, 011, 0001, \ldots$ We will need this concept of Lyndon words in the last chapter where we discuss Sturmian words.

As a last notion we mention *a graph* which is, in fact, a concept of graph theory but it may be used as a tool also in combinatorics on words. A graph is an ordered pair $G = (V, E)$, where V is the set of *vertices* and E is the set of *edges.* An edge is a pair of vertices so it can be seen that an edge connects two vertices. If the edges are directed, i.e., they are ordered pairs then the graph is *a directed graph.* In a directed graph each edge has a starting vertex, *a tail*, and an ending vertex, *a head.* An undirected graph which is connected but does not contain any cycles is called *a tree.* This means that from each vertex of a tree there exists exactly one connection via edges to any other vertex of the tree.

Chapter 2

Avoidability with respect to k-abelian equivalence

In this chapter we discuss about avoidability questions and the main concept is *a k-abelian equivalence*. The k-abelian equivalence lies properly between usual equality and abelian equality. By increasing the natural number k for k-abelian equivalence we can move from abelian equality to the direction of equality step by step. Avoidability problems are widely studied in the contexts of usual word power-freeness and abelian power-freeness. So it is natural to study these same questions also for k-abelian power-freeness.

First we introduce this fairly new concept of k-abelian equality and give some basic properties of it as an equivalence class. Then we concentrate on k-abelian repetitions and k-abelian avoidability. The background of this study mostly lies on works of the avoidability by, for example, Thue [91, 92], Evdokimov [34], Pleasant [85], Dekking [27] and Keränen [62]. The content of this chapter relies on papers [51, 48, 50, 49, 52, 45]. We also summarize a few other results related to k-abelian equivalence and the references for these are given separately.

2.1 Definitions

The crucial notion of this whole chapter is *a k-abelian equivalence* of words. Two equivalent words in a usual sense are the words that are exactly the same. Two abelian equivalent words have the same letters but possibly in different order. The main idea behind the k-abelian equivalence is that we generalize the concept of abelian equality to concern factors of length k not just letters. This means that informally, two k-abelian equivalent words have the same number of occurrences of factors of length k. Again we allow the factors to be in different order. Let us now give the precise definition for k-abelian equivalent words.

Definition 1. *Let $k \geq 1$ be a natural number. Words u and v in Σ^n for $n \geq k-1$ are k-abelian equivalent if*

1. $\mathrm{pref}_{k-1}(u) = \mathrm{pref}_{k-1}(v)$ *and* $\mathrm{suf}_{k-1}(u) = \mathrm{suf}_{k-1}(v)$*, and*
2. *for all $w \in \Sigma^k$, the number of occurrences of w in u and v coincide, i.e. $|u|_w = |v|_w$.*

Words of length at most k are k-abelian equivalent if and only if they are equivalent.

For the k-abelian equivalence we use a symbol $\equiv_k$ and for the abelian equivalence $\equiv_a$. Notice that $\equiv_1$ is the same equivalence relation as $\equiv_a$. The first condition about prefixes and suffixes is needed for to make the k-abelian equivalence a sharpening of the abelian equality. This means that if words are k-abelian equivalent for some k then they are also abelian equivalent, i.e., $u = v \Rightarrow u \equiv_k v \Rightarrow u \equiv_a v$. It can also be seen that the equality is a kind of limit of the k-abelian equivalences, i.e., $u = v \Leftrightarrow (u \equiv_k v \ \ \forall\, k \geq 1)$. On the other hand, the second condition would be enough to make this notion an equivalence relation but together with the first condition k-abelian equivalence is a congruence, too. We remind that a congruence is such an equivalence relation R that if $u_1 R u_2$ and $v_1 R v_2$ then $u_1 v_1 R u_2 v_2$ for all elements u_1, u_2, v_1 and v_2. Now $\Sigma^*/\equiv_k$ is a quotient monoid, whose elements are the k-abelian equivalence classes. In most of the problems we consider in this thesis we do not use the properties of a congruence because we deal with combinatorial questions. However, in algebra the congruence and quotient structures are important concepts.

In fact, instead of condition 1 it would be enough to require the words to have either a common prefix of length $k-1$ or a common suffix of the same length. As is easy to see, two words with a common prefix and the same factors of length k have also a common suffix, and vice versa. Before we give another formulation of the definition of k-abelian equivalent words we give an example illustrating this concept of the k-abelian equivalence.

Example 2. Consider the words *aba* and *bab*. They are not 2-abelian equivalent though they have the same factors of length 2. Clearly they are not abelian equivalent either. Instead, the words *abaab* and *aabab* are 2-abelian equivalent as well as abelian equivalent.

Definition 3. *Let $k \geq 1$ be a natural number. Words u and v in Σ^+ are k-abelian equivalent if for each $n = 1, \ldots, k$ and for every $w \in \Sigma^n$ the number of occurrences of w in u and v coincide, i.e. $|u|_w = |v|_w$.*

In fact, for words that have at least k letters Definition 3 can be simplified into the following form:

Definition 4. *Let $k \geq 1$ be a natural number. Words u and v in Σ^+ are k-abelian equivalent if for every $w \in \Sigma^{k-1} \cup \Sigma^k$ the number of occurrences of w in u and v coincide, i.e. $|u|_w = |v|_w$.*

We will show that the definitions 1 and 4 yields the same equivalence and then it is clear that Definition 3 is valid, too. We begin with the fact that all the factors of length k of a word w contains exactly two factors of length $k-1$, one as a prefix and one as a suffix. So if we know the $\text{pref}_{k-1}(w)$ and $\text{suf}_{k-1}(w)$ it is enough to count the numbers of different factors of length k to count the numbers of different factors of length $k-1$. That is, the condition of Definition 4 follows directly from Definition 1. In addition, $|u|_w = |v|_w$ for all $w \in \Sigma^k$ is required in both definitions 1 and 4. To prove the rest we assume on the contrary that for words $u, v \in \Sigma^+$ $|u|_w = |v|_w$ for every $w \in \Sigma^{k-1} \cup \Sigma^k$ but $\text{pref}_{k-1}(u) = x \neq \text{pref}_{k-1}(v)$. The analyzis for suffixes would be similar. Let $|u|_x = r$ so $|v|_x = r$ by the assumption and $\sum_{a \in \Sigma}(|u|_{ax}) = r - 1$. Now from $\text{pref}_{k-1}(v) \neq x$ and $|v|_x = r$ it follows that $\sum_{a \in \Sigma}(|u|_{ax}) = r$ giving a contradiction.

Notions like *k-abelian repetitions* can be now naturally defined. For instance, $w = uv$ is a *k-abelian square* if and only if $u \equiv_k v$. If a word does not contain a factor which is a k-abelian square then it avoids k-abelian squares and it is a k-abelian square-free word. If there exists an infinite k-abelian square-free word over an alphabet Σ then k-abelian squares are said to be avoidable over the alphabet Σ.

From Definition 3 we can easily conclude the next lemma.

Lemma 5. *If two words are k-abelian equivalent then they are k'-abelian equivalent for each $1 \leq k' \leq k$.*

Proof. Follows from the definitions of k-abelian equivalence straightforwardly: If words u and v are k-abelian equivalent they have the same number of occurrences of each factor of length at most k thus u and v are also $(k-1)$-abelian equivalent. Inductively, they are k'-abelian equivalent for each $1 \leq k' \leq k$. $\square$

Corollary 6. *If an infinite word w contains a k-abelian repetition of order m then w contains a k'-abelian repetition of order m for each $1 \leq k' \leq k$.*

We remark that for abelian equivalence the notion of *the Parikh vector* is important. The Parikh vector p is a function from the set of words over m-letter alphabet $\{a_1, a_2, \ldots, a_m\}$ to the set of m-dimensional vectors over natural numbers, where $p(w) = (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_m})$ for a finite word w. The Parikh vector counts the number of each letter in w, so two words are abelian equivalent if and only if they have the same Parikh vector. *Generalized Parikh mappings* has been studied by Karhumäki in [54] and *a k-generalized Parikh vector* is related to k-abelian equivalence. A k-generalized Parikh vector counts the number of occurrences of each factor of length k in a word w. So if two words are k-abelian equivalent then they have the same k-generalized Parikh vector. The work by Karhumäki [54] can be seen as the first introduction to the k-abelian equivalence. In the paper the idea is to use k-generalized Parikh vectors as approximations for the undecidable Post Correspondence Problem (PCP) [86] and to obtain the decidability of these modificated PCP's with respect to k-generalized Parikh vectors.

2.2 Equivalence classes

In this section we concentrate on the properties of k-abelian equivalence classes. We give characterizations of the equivalence classes of 2-abelian and 3-abelian words over a binary alphabet. We count the number of the equivalence classes of 2-abelian words over a binary alphabet and also examine the size of each such an equivalence class. We approximate the number of the equivalence classes for binary 3-abelian words and compare it with the general evaluation of the number of k-abelian equivalence classes obtained in [59].

First we give characterizations for the equivalence classes of 2- and 3-abelian words over a binary alphabet by which we mean that we define a representative for each equivalence class.

Lemma 7. *Over a binary alphabet $\Sigma = \{a, b\}$ the representative of a 2-abelian equivalence class can be given in the form:*

$$aa^k b^l (ab)^m a^n \quad \textit{or} \quad bb^k a^l (ba)^m b^n,$$

where $k, l, m \geq 0$ and $n \in \{0, 1\}$. Characterization is unambiguous if $l + m \geq n$, and $l = 1$ only if $k = 0$.

Proof. Let w be a word over the binary alphabet Σ. It belongs to some 2-abelian equivalence class and the words that are in the same class have the same number of each factor of length 2 and the first and the last letters are the same. There exist four possibilities for the pair of the first and the last letter, namely $(a, a), (a, b), (b, a)$ and (b, b). All these pairs are clearly obtained by the forms given in lemma. If the word w begins and ends with the same letter x then for $y \neq x$ $|w|_{xy} = |w|_{yx}$ and otherwise if the word begins with x and ends with y then $|w|_{xy} = |w|_{yx} + 1$. Thus if the first and the last letters are given then the 2-abelian equivalence class of the word depends only on the number of factors aa, bb and either ab or ba. Thus each 2-abelian equivalence class has a representative of the given form.

To prove the unambiguousness we assume that $l + m \geq n$ and l can be 1 only if $k = 0$. We consider only the classes beginning with a because the case where the words begin with b is symmetric. In the first case let the word contain only a's. The representation is clearly unambiguous because $l = m = n = 0$ and $k + 1$ is the number of a's. Now let us assume that a word contains both a's and b's. If $l \geq 2$ then representative of the 2-abelian class has also only one representation in the given form. Otherwise, if $l < 2$ the words in the equivalence class do not contain the factor bb. Now if $l = 0$ then m has to equal to the number of b's and the number of factors aa has to equal to $k+1$. If $l = 1$ then $k = 0$ and the word does not have any factors aa. The number of b's is now $m + l$ and the representative is unambiguous. □

Next we give the characterization for the equivalence classes of 3-abelian words over a binary alphabet. Without any additional restrictions to the parameters the characterization is again not unambiguous in a few cases.

Lemma 8. *Over a binary alphabet $\Sigma = \{a, b\}$ the representative of a 3-abelian equivalence class of words of length at least 2 can be given as a word xy where $x \in \{x_1, x_2, x_3, x_4\} = X$ and $y \in \{y_1, y_2\} = Y$. The forms of the elements of the sets X and Y are given below:*

$$\begin{array}{l} x_1 = aaa^kb^l(aabb)^m \\ x_2 = bbb^ka^l(aabb)^m \\ x_3 = abb^ka^l(aabb)^m \\ x_4 = baa^kb^l(aabb)^m \end{array} \quad \text{and} \quad \begin{array}{l} y_1 = (aab)^g(ab)^hb^ia^j \\ y_2 = (abb)^g(ab)^hb^ia^j \end{array}$$

where $k, l, m, g, h \geq 0$, $i \in \{0, 1\}$ and $j \in \{0, \dots, 2 - i\}$.

Proof. The four options for x in X cover all the possible prefixes of length 2 the representative may have, namely aa, ab, ba and bb. The same four factors are obtained as a suffix of $y \in Y$ by choosing i and j properly. In the proof of Lemma 7 it was shown that the 2-abelian equivalence class depends on the number of factors aa, bb and either ab or ba if the prefix and suffix are known. Similarly, for 3-abelian equivalence class the number of factors aaa, bbb and the number of 3-letter factors containing aa or bb are significant. Clearly, the total number of letters is necessary, too. Here parameters k and l are chosen according to the numbers of factors aaa and bbb. Then m and g are chosen according to the number of factors of length 3 containing aa or bb and then the rest of the factors are of the form aba or bab which are covered by choosing appropriate values for h, i and j. □

We remark that if, for instance, $k > 1$ and $l > 2$ then all the possible combinations in Lemma 8 give a representative for a different 3-abelian equivalence class. This observation will be used when counting the number of different 3-abelian equivalence classes. Before that we count the number of the equivalence classes of 2-abelian words over a binary alphabet $\Sigma = \{a, b\}$.

Example 9. If the length of a binary word is one then there exist two equivalence classes a and b. If the length is two then there exist four equivalence classes aa, ab, ba and bb.

Theorem 10. *The number of 2-abelian equivalence classes consisting of words of length n over a binary alphabet is $n^2 - n + 2$ and thus the number is in $\Theta(n^2)$.*

Proof. As mentioned in Example 9 for $n \in \{1, 2\}$ the claim holds. Consider next the words of length $n > 2$ and containing k times the letter a and hence letter b occurs $n - k$ times.

We have a correspondence between the number of different letters and the number of the equivalence classes. If the word contains only a's or b's then the only classes are a^n or b^n, respectively. If the word has one occurrence of a (resp. b) and the rest are b's (resp. a's) then there exist classes ab^{n-1}, bab^{n-2} and $b^{n-1}a$ (resp. ba^{n-1}, aba^{n-2} and $a^{n-1}b$). Otherwise there exist at least two occurrences of both letters. Then all the possible equivalence classes can be obtained by constituting the representatives like in Lemma 7. There exist $\min(k,n-k)$ classes of words of the forms $a \dots b$ and $b \dots a$, $\min(k-1,n-k)$ classes of words of the form $a \dots a$ and $\min(k,n-k-1)$ classes of words of the form $b \dots b$. As a summary we have the following correspondence:

number of a's		number of classes
$k \in \{0, n\}$	$\Rightarrow$	1
$k \in \{1, n-1\}$	$\Rightarrow$	3
$1 < k < n-1$	$\Rightarrow$	$2\min(k,n-k) + \min(k-1,n-k) + \min(k,n-k-1)$

From these we obtain the number of the equivalence classes of words with length $n > 2$:

$$\begin{cases} 8 + \sum_{k=2}^{n-2}(4\min(k, n-k) - 1), & \text{if } 2 \nmid n \\ 6 + 2n + \sum_{k=2}^{\frac{n}{2}-1}(4\min(k, n-k) - 1) + \sum_{k=\frac{n}{2}+1}^{n-2}(4\min(k, n-k) - 1), & \text{if } 2 | n \end{cases}$$

By counting the given sums we get the stated number. □

From the characterization of the equivalence classes of binary 3-abelian words in Lemma 8 we can conclude that the number of the equivalence classes in this case is $\Omega(n^4)$.

Theorem 11. *The number of 3-abelian equivalence classes consisting of words of length n over a binary alphabet is in $\Omega(n^4)$.*

Proof. We have five independent variables, k, l, m, g and h in the characterization of 3-abelian equivalence classes. As mentioned if we restrict $k > 1$

and $l > 2$, each combination of these five values gives a different equivalence class. By fixing the length of the words to be n we obtain a relation

$$k + l + 4m + 3g + 2h + \alpha = n,$$

where $\alpha \in \{2, 3, 4\}$ depending on i and j. We may restrict to analyze words long enough and to subsets of the equivalence classes without affecting the order of result. Hence the equation can be modified to the form:

$$12k' + 12l' + 4(3m') + 3(4g') + 2(6h') = 12n'.$$

Now we may count the number of solutions of equation $\sum_{i=1}^{5} x_i = N$, where $x_i > 0$ for all $i \in \{1, \ldots, 5\}$ and N is fixed. The number of solutions is in $\Theta(N^4)$ which implies that the number of 3-abelian equivalence classes of words of length n is in $\Omega(n^4)$. $\square$

Contrary to the 2-abelian case the exact formula for the number of the 3-abelian equivalence classes is not a polynomial, which can be noted by checking few instances of n. In general, in a fixed but arbitrary alphabet the number of k-abelian equivalence classes of words of length n grows polynomially in n but the degree of the polynomial increases exponentially in k. For example, over a binary alphabet the number of 4-abelian equivalence classes consisting of words of length n is already $\Theta(n^8)$. In fact, over a binary alphabet the number of k-abelian equivalence classes of words of length n is $\Theta(n^{2^{k-1}})$ which can be concluded from the following result proved in [59].

Theorem 12 ([59]). *Let $k \geq 1$ and $m \geq 2$ be fixed numbers and let Σ be an m-letter alphabet. The number of k-abelian equivalence classes of Σ^n is $\Theta(n^{m^k - m^{k-1}})$.*

The proof of this theorem generalizes the idea of the proof of Theorem 11 and it is based on counting the number of different functions related on words. The values of these functions depends on the number of factors of length k in the word. We will not give the entire proof but we will introduce some notions and one lemma that are used in the proof. We will use these same tools later in this chapter.

Let $s_1, s_2 \in \Sigma^{k-1}$ and let

$$S(s_1, s_2, n) = \Sigma^n \cap s_1\Sigma^* \cap \Sigma^* s_2$$

be the set of words of length n that start with s_1 and end with s_2. For every word $u \in S(s_1, s_2, n)$ we can define a function

$$f_u : \Sigma^k \to \{0, \ldots, n - k + 1\}, \ f_u(t) = |u|_t.$$

If $u, v \in S(s_1, s_2, n)$, then $u \sim_k v$ if and only if $f_u = f_v$.

On the other hand, if a function $f : \Sigma^k \to \mathbb{N}_0$ is given, then a directed multigraph G_f can be defined as follows:

- The set of vertices is Σ^{k-1}.
- If $t = s_1 a = b s_2$, where $a, b \in \Sigma$, then there are $f(t)$ edges from s_1 to s_2.

If $f = f_u$, then this multigraph is related to *the Rauzy graph* of u. In a Rauzy graph of order n the set of vertices is a set of words of length n such that for each word $x_1 x_2 \cdots x_n$ in the set there also exist words $y_0 x_1 x_2 \cdots x_{n-1}$ and $x_2 x_3 \cdots x_n y_{n+1}$ in the set for some letters y_0 and y_{n+1}. A Rauzy graph is a directed graph and there exists an edge (x, y) from x to y if $x = x_1 x_2 \cdots x_n$ and $y = x_2 x_3 \cdots x_n y_{n+1}$.

In the following lemma $\deg^-$ denotes the indegree, i.e., the number of incoming edges and $\deg^+$ the outdegree, i.e., the number of outgoing edges of a vertex in G_f.

Lemma 13 ([59]). *For a function $f : \Sigma^k \to \mathbb{N}_0$ and words $s_1, s_2 \in \Sigma^{k-1}$, the following are equivalent:*

(i) there is a number n and a word $u \in S(s_1, s_2, n)$ such that $f = f_u$,

(ii) there is an Eulerian path from s_1 to s_2 in G_f,

(iii) the underlying graph of G_f is connected, except possibly for some isolated vertices, and $\deg^-(s) = \deg^+(s)$ for every vertex s, except that if $s_1 \neq s_2$, then $\deg^-(s_1) = \deg^+(s_1) - 1$ and $\deg^-(s_2) = \deg^+(s_2) + 1$,

(iv) the underlying graph of G_f is connected, except possibly for some isolated vertices, and

$$\sum_{a \in \Sigma} f(as) = \sum_{a \in \Sigma} f(sa) + c_s$$

for all $s \in \Sigma^{k-1}$, where

$$c_s = \begin{cases} -1, & \text{if } s = s_1 \neq s_2, \\ 1, & \text{if } s = s_2 \neq s_1, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. (i) $\Leftrightarrow$ (ii): $u = a_1 \dots a_n \in S(s_1, s_2, n)$ and $f = f_u$ if and only if

$$s_1 = a_1 \dots a_{k-1} \to a_2 \dots a_k \to \cdots \to a_{n-k+2} \dots a_n = s_2$$

is an Eulerian path in G_f.

(ii) $\Leftrightarrow$ (iii): This is well known.

(iii) $\Leftrightarrow$ (iv): (iv) is just a reformulation of (iii) in terms of the function f. $\square$

Up to this point we have considered the number of different k-abelian equivalence classes but we can also examine the sizes of these equivalence classes. Here is an example of the number of binary words in a 2-abelian equivalence class depending on the number of different factors. The results were originally published in [51], but there were a few misprints so these are discussed more closely here.

Example 14. We count first the number of binary words that begin with a, end with b and belong to one 2-abelian equivalence class. As mentioned in the proof of Lemma 7 the number of factors ab and ba are dependent on each other. So let the number of factors aa, ab, ba and bb be k, l, $(l-1)$ and m, respectively. Then the equivalence class contains $\binom{k+l-1}{k}\binom{l+m-1}{m}$ such words. The binomial coefficient $\binom{k+l-1}{k}$ refers to the number of choices for the positions of factors aa. There are l factors of ab so there are l positions with respect to these ab factors and the number of factors aa to place into these l positions is k. Similarly, $\binom{l+m-1}{m}$ refers to the number of choices for the positions of factors bb.

Next we consider binary words that begin and end with the same latter a. Let the number of factors aa, ab, ba and bb be k, l, l and m, respectively. Now there are $\binom{k+l}{k}\binom{l+m-1}{m}$ words in this 2-abelian equivalence class. Because the word begins and ends with a there are $k+1$ positions on which the factors aa can be placed. Results for 2-abelian equivalence classes containing words beginning with b are similar and these cases cover all the possible 2-abelian words over binary alphabet.

Example 15. Consider 2-abelian words over a binary alphabet $\{a, b\}$. The following 2-abelian equivalence classes are such that they contain the shortest words so that the size of the class is more than one: $\{aaba, abaa\}$ and $\{babb, bbab\}$. For words of length 5 there exist 2-abelian equivalence classes that has 3 elements: $\{aaaba, aabaa, abaaa\}$ and $\{babbb, bbabb, bbbab\}$. With the formulas given in the previous Example 14 we can count as an example the cases for which $k = 3$, $l = 2$ and $m = 1$. Let us first consider the number of words $a \cdots b$ with the given values of k, l and m. Now the length of the words is 8 and there exist 8 words in the class, namely a^4bab^2, $a^3ba^2b^2$, $a^2ba^3b^2$, aba^4b^2, a^4b^2ab, $a^3b^2a^2b$, $a^2b^2a^3b$ and ab^2a^4b. The words of the form $a \cdots a$ with the same values of k, l and m contain 9 letters and then there exist already 20 words in the class.

2.3 Avoidability questions

In this section we study the avoidability questions with respect to k-abelian equivalence. We ask what are the sizes of the smallest alphabets in which k-abelian squares and cubes can be avoided. These questions are natural problems with respect to earlier studies. Because k-abelian equivalence lies properly between equality and abelian equality we can first concentrate on the known results for avoiding squares, cubes, abelian squares and abelian cubes. Squares are avoidable over ternary alphabets, see e.g. [68], but the maximal length of a ternary word avoiding abelian squares can be easily checked to be seven. It is known, although that is not easy to prove, that there exists an infinite word over a 4-letter alphabet avoiding abelian squares, see [62]. These results indicate that for k-abelian squares the avoidability is obtained either in a alphabet of size three or four.

It is also known that cubes are avoidable over a binary alphabet, for example the infinite word of Thue-Morse accomplishes this property, see [68]. On the other hand, it is easy to see that abelian cubes are not avoidable over a binary alphabet. Dekking has shown that abelian cubes are avoidable over a ternary alphabet, see [27]. So k-abelian cubes can be avoided over an alphabet of size two or three. The problem could also be considered in other way round. Dekking has also proved that repetitions of fourth order are avoidable over a binary alphabet, see [27]. Thus the order of k-abelian repetition that can be avoided over a binary alphabet is either three or four.

The next Table 2.1 summarizes the given results and tells the limits for our k-abelian avoidability problems. We may suppose that $k > 1$ because $\equiv_1$ means the abelian equivalence $\equiv_a$.

Avoidability of squares				Avoidability of cubes			
	type of rep.				type of rep.		
size of the alph.	$=$	$\equiv_k$	$\equiv_a$	size of the alph.	$=$	$\equiv_k$	$\equiv_a$
2	$-$	$-$	$-$	2	$+$	?	$-$
3	$+$	?	$-$	3	$+$	$+$	$+$
4	$+$	$+$	$+$				

Table 2.1: Avoidability of different types of repetitions in infinite words.

For convenience, we appoint our general main problems of this chapter so that we may refer to those later.

Problem 1. *Does there exist an infinite ternary word that would avoid k-abelian squares for some $k \geq 2$?*

Problem 2. *Does there exist an infinite binary word that would avoid k-abelian cubes for some $k \geq 2$?*

If the answer for one of these questions is positive, then the natural additional question is what is the smallest value of k for which the property holds. We start to study these main problems from the case $k = 2$ and we will notice that the problems are not trivial, even in this case.

To examine the existence of an infinite ternary word that would avoid 2-abelian squares we executed a computer program. The program was written down by Java and it can be easily obtained by simplifying the script for producing ternary 3-abelian square-free words. This is given in Appendix B. Some remarks of the code are explained in the end of Section 2.3.2 in which other computational results are presented, too. The basic idea of the code is to generate in a lexicographic order longer and longer words avoiding 2-abelian squares. Once the program has generated a word that contains a 2-abelian square it traces back to the next 2-abelian square-free word in a lexicographic order and continues the search of a longer such word.

The result we obtained was that the maximal length of a 2-abelian square-free word is 537 letters and each of the longer words over the ternary alphabet contains a 2-abelian square. This word is unique up to the permutations of the alphabet and it is given in Example 17. With the earlier results mentioned above this shows that the alphabet to avoid 2-abelian squares have to contain at least four letters, and as mentioned that is enough. So the behaviour of avoidance of 2-abelian squares is similar to the avoidance of abelian squares.

Theorem 16. *The size of the smallest alphabet in which 2-abelian squares can be avoided is 4.*

Example 17. The word of length 537 over a ternary alphabet $\Sigma = \{a, b, c\}$ that avoids 2-abelian squares:

abcbabcacbacabacbabcbacabcbabcabacabcacbacabacbabcbacbcacbabcacbcabcba
bcabacbabcbacbcacbacabacbabcbacabcbabcabacabcacbacabacbabcbacbcacbacab
acbcabacabcacbcabcbacbcacbacabacbabcbacbcacbabcacbcabcbabcabacbabcbacb
cacbacabacbabcbacabcbabcabacabcacbacabacbabcbacbcacbacabacbcabacabcacb
cabcbabcabacabcacbacabacbabcbacabcbabcabacabcacbcabcbabcabacbabcbacbca
cbabcacbcabcbabcabacabcacbcabcbacbcacbacabacbcabacabcacbcabcbabcabacab
cacbacabacbabcbacabcbabcabacabcacbcabcbabcabacbabcbacbcacbabcacbcabcba
bcabacabcacbacabacbabcbacabcbabcabacabcacbabcba.

The result of our computer program was verified by an independent computer program by Aleksi Saarela. The original program was also tested to recognize 2-abelian squares in words that were manually given to it and known to contain a 2-abelian square.

Although, the most of the earlier results on avoiding repetitions and abelian repetitions are obtained by iterating a suitable morphism and showing that the generated word does not contain the repetition, the next example shows that the method of iterating a morphism might not give answers to problems on k-abelian repetitions.

Example 18. In each of the following cases a 2-abelian cube is found fairly early from the beginning. The infinite words that are obtained by iterating a morphism are known to avoid some repetitions. Most of the words in this example can be found in [4] and for the rest of the words the references are given separately.

- Infinite overlap-free Thue-Morse word (by iterating the morphism: $0 \to 01$, $1 \to 10$): $01\overbrace{101001}\overbrace{100101}\overbrace{101001}011...$
- Cube-free infinite word (by iterating the morphism: $0 \to 001$, $1 \to 011$): $001001\overbrace{011001}\overbrace{001011}\overbrace{001011}011...$
- Morphism $0 \to 001011$, $1 \to 001101$, $2 \to 011001$ maps ternary cube-free words to binary cube-free words, see [10], but $001011 \equiv_{a,2} 001101 \equiv_{a,2} 011001$, thus images of all words mapped with this morphism contains 2-abelian cubes.
- A binary overlap-free word w can also be gained in form $w = c_0c_1c_2\ldots$, where c_n means the number of zeros (mod 2) in the binary expansion of n. Again, a 2-abelian cube of length 6 begins as early as from the fifth letter: $w = 0010\overbrace{011010}\overbrace{010110}\overbrace{011010}011...$
- A binary sequence called Kolakoski sequence is cube-free, see [15] and [66], but not 2-abelian cube-free: $122\overbrace{112122}\overbrace{122112}\overbrace{112212}112...$ (It is an open question whether the Kolakoski sequence is a morphic word.)

In the next theorems 21 and 22 we bring out some properties that the infinite words generated by iterating a morphism have. These also support the view that iterating a single morphism may not be a strong enough tool to produce infinite words avoiding k-abelian repetitions. To prove the theorems we give first two lemmas, the latter being just an extension of the former. For the clarity, we prove this special case first. In these lemmas *a 1-free morphism* means a morphism that maps each letter to a word that has length at least two.

Lemma 19. *Let h be a 1-free morphism over an alphabet Σ and let w be a word over Σ. Let $n = \min\{|h(a)| : a \in \Sigma\}$, i.e. $n \geq 2$. If w has 2-abelian equivalent factors u and v then the word $h(w)$ has $(n+1)$-abelian equivalent factors $h(u)$ and $h(v)$.*

Proof. Clearly, $h(u)$ and $h(v)$ are factors of $h(w)$. Let $\text{pref}_1(u) = \text{pref}_1(v) = x$ and $\text{suf}_1(u) = \text{suf}_1(v) = y$ for some $x, y \in \Sigma$. Now $h(x)$ is a prefix of $h(u)$ and $h(v)$ and similarly $h(y)$ is a suffix of $h(u)$ and $h(v)$ where $|h(x)|, |h(y)| \geq n$. Thus the first condition of Definition 1 of k-abelian equivalent words holds.

Each factor of length $(n+1)$ in $h(u)$ (or $h(v)$) is contained in a factor $h(st)$ where st is a factor of u (or v) and $s, t \in \Sigma$. This follows from the choice of n. In fact, a factor of length $(n+1)$ may be already contained in to an image of a single letter. In any case, words u and v are 2-abelian equivalent words and thus abelian equivalent, too. So words u and v have the same number of each letter and the same number of each factor of length two, respectively. Thus the words $h(u)$ and $h(v)$ have the same number of factors $h(s)$ for each $s \in \Sigma$ and $h(st)$ for each $s, t \in \Sigma$. From this we can conclude that the words $h(u)$ and $h(v)$ have the same number of occurrences of each factor of length $(n+1)$. Now the second condition of Definition 1 of k-abelian equivalent words is also satisfied which completes the proof. $\square$

Now we generalize the previous lemma by taking k-abelian factors as a starting point.

Lemma 20. *Let h be a 1-free morphism over an alphabet Σ and let w be a word over Σ. Let $n = \min\{|h(a)| : a \in \Sigma\}$, i.e. $n \geq 2$. If w has k-abelian equivalent factors u and v then the word $h(w)$ has $((k-1)n+1)$-abelian equivalent factors $h(u)$ and $h(v)$.*

Proof. The idea of the proof is the same as in the proof of Lemma 19. Now $\mathrm{pref}_{k-1}(u) = \mathrm{pref}_{k-1}(v)$ and $\mathrm{suf}_{k-1}(u) = \mathrm{suf}_{k-1}(v)$ ensuring $h(u)$ and $h(v)$ to have a common prefix (resp. suffix) of length at least $(k-1)n$.

Correspondingly, each factor of length $((k-1)n+1)$ in $h(u)$ (or $h(v)$) is contained in a factor $h(p)$ where p is a factor of u (or v) and $|p| \leq k$. Because the words u and v are k-abelian equivalent the numbers of each factor of length at most k coincide in these words. From these it follows that $h(u)$ and $h(v)$ have the same number of each factor of length $((k-1)n+1)$, and thus the words are $((k-1)n+1)$-abelian equivalent. $\square$

In the following theorems we assume h to be a prefix preserving morphism over an alphabet Σ and $a \in \Sigma$ to be such that $h^\infty(a)$ is well defined. Instead of requiring that $\min\{|h(a)| : a \in \Sigma\} > 1$ we require the following property:

$$\forall a \in \Sigma \; \exists n > 0 : |h^n(a)| > 1. \tag{2.1}$$

Theorem 21. *The following two conditions are equivalent:*

1. *The infinite word $h^\infty(a)$ contains k-abelian repetition of order m for some $k \geq 2$.*
2. *The infinite word $h^\infty(a)$ contains k-abelian repetition of order m for each $k \geq 1$.*

Proof. It is clear that the Condition 1 follows from 2 straightforwardly.

Let us prove that the first condition implies the second one. First of all, if (2.1) holds we can choose $n' > 0$ such that $\mu = \min\left\{|h^{n'}(a)| : a \in \Sigma\right\} > 1$ and let $\hat{h}$ denote the morphism $h^{n'}$. If $a \in \Sigma$ is a letter for which h is prolongable then $\hat{h}^\infty(a) = h^\infty(a)$. Now we may apply Lemma 20 to the morphism $\hat{h}$.

Let $w = u_1u_2 \cdots u_m$ be a factor of $h^\infty(a)$ such that words u_i are k-abelian equivalent words with each other, then the same holds for $\hat{h}^\infty(a)$. Now $\hat{h}^\infty(a) = h^\infty(a)$ also contains the factor $\hat{h}(w) = \hat{h}(u_1)\hat{h}(u_2)\cdots\hat{h}(u_m)$. From Lemma 20 we know that words $\hat{h}(u_i)$ are $((k-1)\mu+1)$-abelian equivalent words with each other, and thus $h^\infty(a)$ contains a $((k-1)\mu+1)$-abelian repetition of order m. Now we can inductively apply Lemma 20 for the case $h^\infty(a)$ having a $((k-1)\mu+1)$-abelian repetition of order m. It gives us that $h^\infty(a)$ has a $((k-1)\mu^2+1)$-abelian repetition of order m. Finally, we can conclude that $h^\infty(a)$ has a $((k-1)\mu^i+1)$-abelian repetition of order m for each $i \in \mathbb{N}$. In addition, from Lemma 5 we know that $h^\infty(a)$ contains k'-abelian repetition of order m for each $1 \leq k' \leq (k-1)n^i+1$, too. Thus the infinite word $h^\infty(a)$ contains k-abelian repetition of order m for each $k \geq 1$. □

We can also formalize the previous Theorem 21 in the context of k-abelian avoidability.

Theorem 22. *The following two conditions are equivalent:*

1. *The infinite word $h^\infty(a)$ is k-abelian m-free for some $k \geq 1$.*

2. *The infinite word $h^\infty(a)$ is k-abelian m-free for each $k \geq 2$.*

Proof. Follows from Theorem 21. □

Next we mention a few consequences of the above. We remark that h was chosen to be a prefix preserving morphism so that $h^\infty(a)$ is well defined and $\forall a \in \Sigma\ \exists n > 0 : |h^n(a)| > 1$. Let H be the set of morphisms satisfying these conditions.

Remark 23. If each infinite binary word contained 2-abelian cube then from Theorem 21 would follow that for each $h \in H$ over binary alphabet $h^\infty(a)$ would contain k-abelian cube for all $k \geq 1$. This means that if there exists a binary morphism $h \in H$ such that $h^\infty(a)$ is k-abelian cube-free for some $k \geq 2$ then there exists an infinite 2-abelian cube-free word over binary alphabet. The same result can be concluded straightforwardly from Theorem 22. On the other hand, there exists a 2-uniform prefix preserving morphism over two letter alphabet generating a cube-free binary word, for example Thue-Morse word. However, Thue-Morse word contains 2-abelian cube as shown in Example 18, and thus from Theorem 21 it follows that this word is not k-abelian cube-free for any $k \geq 1$.

Remark 24. We will show later in Theorem 38 that we can construct, for example, an infinite binary 8-abelian cube-free word as a morphic image of an infinite word generated by iterating a uniform morphism. It is easy to see that this word contains 2-abelian cube as a factor, which implies by Theorem 22 that the word cannot be obtained by iterating a binary morphism $h \in H$. This also shows how we can use our theorems for deciding whether some infinite word can be obtained by iterating a single morphism $h \in H$.

Now we can combine the results of Theorem 21 and Theorem 16 to once again point out the difficulties we have with words generated by iterating a morphism.

Remark 25. Theorem 16 shows that each infinite word over three letter alphabet contains a 2-abelian square. From Theorem 21 it follows that for each $h \in H$ over ternary alphabet $h^{\infty}(a)$ contains k-abelian square for all $k \geq 1$. This means that k-abelian square-free word over ternary alphabet cannot be generated by iterating a morphism $h \in H$ over ternary alphabet for any $k \geq 1$. Later in the next subsection 2.3.1 we will show that, in fact, k-abelian squares are not avoidable over any pure morphic word for any $k \geq 1$. On the other hand, there exists a 13-uniform prefix preserving morphism over three letter alphabet generating a square-free ternary word, see [65] but as mentioned this word can not be k-abelian square-free for any $k \geq 1$.

2.3.1 Unavoidability of k-abelian squares in ternary pure morphic words

In this section we concentrate on k-abelian square-freeness and continue discussing pure morphic words. The question whether pure morphic words can avoid k-abelian squares over ternary alphabets is challenging and reasonable. For example, Thue already showed that there exists an infinite pure morphic square-free word over ternary alphabet, see [92]. In addition, we have a strong evidence that an infinite ternary 3-abelian square-free word would exist. Our discoveries of the numerical evidence are presented in Section 2.3.2 which concentrates on our computational results.

Actually, in a recent manuscript by Michaël Rao [87] the existence of an infinite ternary 3-abelian square-free word is managed to be proved. This word is obtained as a morphic image of a pure morphic word. So the word is morphic but not pure morphic, as discussed later. Although, iterated morphisms constitute a common tool in avoidability questions, there also exist patterns in the ordinary word case that can be avoided in binary words but not in words produced by only iterating a morphism, as introduced next.

Cassaigne has given a classification of binary patterns with respect to avoidability in binary words, in binary pure morphic words and in ternary pure morphic words, [18]. The patterns $\alpha^2\beta^2\alpha$, $\alpha\beta\alpha^2\beta$ and $\alpha\beta\alpha^2\beta\alpha$ are such that they can be avoided over a binary alphabet, but not in infinite binary pure morphic words. Similarly, we will show that 3-abelian squares can be avoided over a ternary alphabet but not in infinite ternary pure morphic words. A related well-known example is given by the famous (cube-free) Kolakoski word: it is not pure morphic [23], but it is unknown whether it is morphic. Indeed, it is not known whether its subword complexity is quadratic, as would be in the case of a morphic word.

On the other hand, Currie has conjectured (see [24] and [69, Problem 3.1.5, p. 132]), that if a pattern p is avoidable on alphabet Σ, then there exist an alphabet Σ', two morphisms $f : \Sigma'^* \to \Sigma^*$ and $g : \Sigma'^* \to \Sigma'^*$ and a letter $a \in \Sigma'$ such that the infinite word $f(g^\infty(a))$ avoids p. That is, p is avoidable over morphic words. Based on our results and intuition we do not dare to make a related conjecture for k-abelian repetitions, even in the case where the pattern is an integer power.

Our proof of the following theorem showing that an infinite ternary k-abelian square-free word cannot be obtained by iterating a single morphism is divided into two parts. First we will prove the result for 3-abelian squares. In the second part we will generalize the result for every k. A starting point for this theorem is the result of Theorem 16, that is each infinite ternary word contains a 2-abelian square. For binary words and 2-abelian cubes we do not have a similar result. So we cannot use the same idea for proving that an infinite binary k-abelian cube-free word could not be obtained by iterating a single morphism. Indeed, this is an open question.

Theorem 26. *Every ternary infinite pure morphic word contains a k-abelian square for any $k \geq 1$.*

The proof for the case $k = 3$

We start by stating few lemmas which cover some special cases. Combining these results we are able to conclude our avoidability result for 3-abelian squares. Let h be such a prefix preserving morphism over $\Sigma = \{a, b, c\}$ that it is prolongable for a and let $h^\infty(a) = w$. If w is k-abelian square-free then at least one letter has to map to a letter as a consequence of Theorem 21. On the other hand, by Lemma 28 we will show that at most one letter may map to a letter. We continue by remarking that an infinite ternary k-abelian square-free word, in fact a word avoiding ordinary word squares, has to contain each possible factor of length two except aa, bb and cc.

Lemma 27. *Each word of length ≥ 14 over an alphabet $\{a, b, c\}$ contains a square if some of the factors ab, ac, ba, bc, ca or cb do not belong to the* *set of the factors of w, i.e., to the set $F(w)$.*

Proof. Assume that $ab \notin F(w)$. The other cases are symmetric. It is easy to check that $bcbacbcacbaca$ is the longest word avoiding ab and squares. $\square$

By using Lemma 27 we may prove the following:

Lemma 28. *Let h be a morphism*

$$h : \begin{cases} a \mapsto a\alpha \\ b \mapsto x \\ c \mapsto y \end{cases}, \quad \textit{where } \alpha \in \Sigma^+ \textit{ and } x, y \in \Sigma.$$

Now $h^\infty(a) = w$ contains a square.

Proof. If $h(b) = a$ then $h(ba) = aa\alpha$ and by Lemma 27 ba as well as $h(ba)$ are factors of w. Similarly, the case $h(c) = a$ gives a square.

If $h(b) = b = h(c)$ then the image of the factor bc gives a square $h(bc) = bb$ and by Lemma 27 $bc, h(bc) \in F(w)$. The case $h(b) = c = h(c)$ is similar.

Now there are two cases left. First, if $h(b) = c$ and $h(c) = b$ then $h^2(b) = b$ and $h^2(c) = c$ so without loss of generality it is enough to consider the case $h(b) = b$ and $h(c) = c$. If $a^{-1}h(a) = \alpha \in \{b, c\}^+$ then $a^{-1}h^\infty(a) \in \{b, c\}^\omega$ and the binary infinite word cannot be square-free. Thus we have to check the case in which $h(a) = a\alpha_1 a\alpha_2$, where $\alpha_1 \in \{b, c\}^+$ and $\alpha_2 \in \{a, b, c\}^*$. Now

$$a \overset{h}{\mapsto} a\alpha_1 a\alpha_2 \overset{h}{\mapsto} a\alpha_1 a\alpha_2\alpha_1 a\alpha_1 a\alpha_2 h(\alpha_2),$$

because $h(\alpha_1) = \alpha_1$ and thus $h^2(a)$ contains a square $\alpha_1 a\alpha_1 a$. This completes the proof. $\square$

Thus, if there exists a morphism h over a three letter alphabet Σ that generates an infinite k-abelian square-free word it maps exactly one letter to a letter and, in fact, it has to map the letter into itself. Otherwise $|h^2(a)| > 1$ for all $a \in \Sigma$. Without loss of generality, we may assume that $h(b) = b$. By Lemma 27 we may assume that the word contains the images of all the factors of length two except aa, bb and cc. This way we can again restrict the form of the morphism.

Lemma 29. *If h is a prefix preserving morphism over the alphabet $\Sigma = \{a, b, c\}$ that generates an infinite k-abelian square-free word, it has to have one of the following forms:*

$$h : \begin{cases} a \mapsto a\alpha a \\ b \mapsto b \\ c \mapsto c\gamma c \end{cases} \quad \textit{or} \quad h : \begin{cases} a \mapsto a\alpha c \\ b \mapsto b \\ c \mapsto a\gamma c \end{cases}, \quad \textit{where } \alpha, \gamma \in \Sigma^+.$$

Proof. We already have that

$$h : \begin{cases} a \mapsto a\alpha' \\ b \mapsto b \\ c \mapsto \gamma' \end{cases}, \text{ where } \alpha', \gamma' \in \Sigma^+ \text{ and } |\gamma'| \geq 2.$$

We note that if a word avoids k-abelian squares it also avoids usual word squares. Thus $h(ab) = a\alpha' b$ from which it follows that b cannot be a suffix of α'. From $h(cb) = \gamma' b$ and $h(ca) = \gamma' a\alpha'$ it follows that c has to be a suffix of γ'. In addition, b cannot be a prefix of γ' because $h(bc) = b\gamma'$. Now we have that $h(a) = a\alpha a$ or $h(a) = a\alpha c$ and $h(c) = a\gamma c$ or $h(c) = c\gamma c$, where $\alpha, \gamma \in \Sigma^*$. By considering $h(ac)$ we conclude that

$$h : \begin{cases} a \mapsto a\alpha a \\ b \mapsto b \\ c \mapsto c\gamma c \end{cases} \quad \text{or} \quad h : \begin{cases} a \mapsto a\alpha c \\ b \mapsto b \\ c \mapsto a\gamma c \end{cases}, \quad \text{where } \alpha, \gamma \in \Sigma^*.$$

If $\alpha = \epsilon$ (for γ similarly) then $h(a) = aa$ or $h(a) = ac$ and both cases lead to a square. In the latter case $ca \mapsto a\gamma cac \mapsto ach(\gamma)a\gamma caca\gamma c$. Thus neither α nor γ can be the empty word which completes the proof. $\square$

Next we restrict to consider k-abelian square-freeness with $k = 3$.

Lemma 30. *Consider the morphism*

$$h : \begin{cases} a \mapsto a\alpha c \\ b \mapsto b \\ c \mapsto a\gamma c \end{cases}, \text{ where } \alpha, \gamma \in \Sigma^+.$$

Now $h^\infty(a)$ has a 3-abelian square.

Proof. Let $h^\infty(a) = w$ and let us assume that it is square-free (otherwise $h^\infty(a)$ contains 3-abelian squares). As mentioned, each infinite ternary word contains a 2-abelian square, especially $u_1u_2 \in F(w)$ where u_1 and u_2 are 2-abelian equivalent. We have that $u_1 \neq u_2$ and thus $|u_1| > 3$.

Each factor of length three of the word $h(u_1)$ (resp., for $h(u_2)$) is a factor of $h(x_1x_2x_3)$, where $x_1x_2x_3$ is a factor of u_1 and $x_1, x_2, x_3 \in \Sigma$. In fact, the only case where the image of two consecutive letters is not enough is the case where $x_2 = b$ and we take the factor $\mathrm{suf}_1(h(x_1))h(x_2)\mathrm{pref}_1(h(x_3)) = cba$. Thus the factors of length three of the word $h(u_1)$ are determined by the factors of length two of the word u_1 and the number of factors x_1bx_3 in u_1 where $x_1, x_3 \in \{a, c\}$. Because u_1 and u_2 are 2-abelian equivalent now the words $h(u_1)$ and $h(u_2)$ have the same occurrences of the factors of length three.

In addition, if $\text{pref}_1(u_1) = b$ then $\text{pref}_2(h(u_1)) = ba$ because bb cannot be a prefix of u_1 and in other cases $|h(\text{pref}_1(u_1))| \geq 3$. Because $\text{pref}_1(u_1) = \text{pref}_1(u_2)$ we have that $\text{pref}_2(h(u_1)) = \text{pref}_2(h(u_2))$ and correspondingly, $\text{suf}_2(h(u_1)) = \text{suf}_2(h(u_2))$. Now $h(u_1u_2) \in F(w)$ and $h(u_1)$ and $h(u_2)$ are 3-abelian equivalent. □

From the previous lemmas we have that if there exists a prefix preserving morphism over a three letter alphabet generating an infinite 3-abelian square-free word it has to be of the following form:

$$h : \begin{cases} a \mapsto a\alpha a \\ b \mapsto b \\ c \mapsto c\gamma c \end{cases}, \text{ where } \alpha, \gamma \in \Sigma^+.$$

With the following three lemmas we will show that the morphism above always leads to a word containing 3-abelian square. For Lemmas 31, 32 and 33 let us denote $h^\infty(a) = w$.

Lemma 31. *If $aba, cbc \in F(w)$ then w contains a square.*

Proof. Now $h(aba) = a\alpha aba\alpha a$ and $h(cbc) = c\gamma cbc\gamma c$. This means that the only non-trivial case is $h(a) = a\alpha' ca$ and $h(c) = ca\gamma' c$ where $\alpha', \gamma' \in \Sigma^*$. Now $h(ac) = a\alpha' caca\gamma' c$ gives a square $caca$. □

Thus we may restrict the word w not to contain the factor aba or cbc. These cases are symmetric and it is enough to discuss the other one.

Lemma 32. *If $aba \in F(w)$ and $cbc \notin F(w)$ then w contains a square.*

Proof. Now $h(aba) = a\alpha aba\alpha a$. By avoiding trivial squares this gives $h(a) = aca$ or $h(a) = ac\alpha_1 ca$ where $\alpha_1 \in \Sigma^+$. If $h(a) = aca$ then $h(ac) = acac\gamma c$ gives a square $acac$. Thus $h(ac) = ac\alpha_1 cac\gamma c$ and we may conclude that $h(a) = ac\alpha_2 bca$ and $h(c) = cb\gamma_1 c$ where $\alpha_2, \gamma_1 \in \Sigma^*$. The cases $h(a) = acbca$ and $h(c) = cbc$ are not possible because $cbc \notin F(w)$. Further, $h(ca)$ gives $h(a) = acb\alpha_3 bca$ and $h(c) = cb\gamma_2 bc$ where $\alpha_3, \gamma_2 \in \Sigma^+$. The assumption $cbc \notin F(w)$ also gives that $h(a) = acba\alpha_4 bca$ where $\alpha_4 \in \Sigma^*$. Now $cba \in F(h(a)) \subset F(w)$ and $h(cba) = cb\gamma_2 bcbacba\alpha_4 bca$ which contains the square $cbacba$. □

Now we know that w cannot have either aba or cbc as a factor.

Lemma 33. *If $aba, cbc \notin F(w)$ then w contains a 3-abelian square.*

Proof. By [91, 92, 6], the only infinite pure morphic square-free word over a ternary alphabet avoiding aba and cbc is obtained by iterating a morphism

$$g : \begin{cases} a \mapsto abc \\ b \mapsto ac \\ c \mapsto b \end{cases}.$$

Now $g \in H$ and thus $g^\infty(a)$ contains a 3-abelian square. In fact, already $g^5(a)$ contains a 3-abelian square: $g^5(a) = ab\,\underbrace{cacbabcbacab}\,\underbrace{cacbacabcbab}\dots$.

□

Consider now an erasing morphism (the one with $c \mapsto \epsilon$ behaves similarly)

$$e : \begin{cases} a \mapsto a\alpha \\ b \mapsto \epsilon \\ c \mapsto \gamma \end{cases} .$$

The word $e^\infty(a)$ cannot contain aba as well as cbc because $e(aba) = a\alpha a\alpha$ and $e(cbc) = \gamma\gamma$. Thus this case returns to the proof of Lemma 33 and, in fact, the same argument can be used for general k-abelian case, too.

Now we are ready to state the first part of the result.

Lemma 34. *Every ternary infinite pure morphic word contains a 3-abelian square.*

Proof. The proof is clear due to given lemmas. With lemmas 28, 29 and 30 we have restricted the form of the morphism that could produce an infinite ternary 3-abelian square-free word and with lemmas 31, 32 and 33 we have shown that the word obtained by iterating that kind of morphism always contains a 3-abelian square. □

The proof for the general case $k \geq 1$

In this subsection we extend the result of Lemma 34 from 3-abelian squares to arbitrary k-abelian squares. We start by a lemma.

Lemma 35. *If a ternary infinite pure morphic word contains a $(k-1)$-abelian square for $k > 3$ then it contains a k-abelian square.*

Proof. Consider a ternary prefix preserving morphism h and assume that $w = h^\infty(a)$ contains a $(k-1)$-abelian square, i.e., there exist $(k-1)$-abelian equivalent words u_1 and u_2 such that $u_1u_2 \in F(w)$. Assume to the contrary that w is k-abelian square-free. From Lemma 29 it follows that we may assume $|h(a)|, |h(c)| \geq 3$ and $h(b) = b$.

Now every factor of length k in the word $h(u_1)$ (resp., $h(u_2)$) is a factor of $h(v_1)$ (resp., $h(v_2)$) where $v_1 \in F(u_1)$ and $|v_1| \leq \lfloor \frac{k-3}{2} \rfloor + 3$. This is because if the word contains factors of which at most every other has length one and all the others have length at least three we can reach at most $\lfloor \frac{k-3}{2} \rfloor + 3$ factors with a subword of length k. For example, with a subword of length 11 we can reach at most 7 factors as depicted below.

$$\dots - * - - \underbrace{- * - - - * - - - * -} - - * - \dots$$

Thus the factors of $h(u_1)$ (resp., $h(u_2)$) of length k are determined by factors of u_1 (resp., u_2) of length at most $k-1$ because $k-1 \geq \left\lfloor \frac{k-3}{2} \right\rfloor + 3$ for $k > 3$. We know that u_1 and u_2 are $(k-1)$-abelian equivalent thus $h(u_1)$ and $h(u_2)$ have the same number of occurrences of different factors of length k.

In addition, from $\mathrm{pref}_{k-2}(u_1) = \mathrm{pref}_{k-2}(u_2)$ it follows that $\mathrm{pref}_{k-1}(h(u_1)) = \mathrm{pref}_{k-1}(h(u_2))$. The suffixes of length $k-1$ of the words $h(u_1)$ and $h(u_2)$ coincide, too. This shows that $h(u_1)$ and $h(u_2)$ are k-abelian equivalent and clearly $h(u_1)h(u_2) \in F(w)$. □

Now we have all the lemmas for the proof of Theorem 26 which generalizes Lemma 34.

Proof for Theorem 26. Lemma 34 states the result for $k = 3$ and by Lemma 5 the claim holds also for $1 \leq k \leq 3$. Now we can use Lemma 35 for the case $k = 4$ and inductively prove the claim for $k \geq 4$. □

2.3.2 Computational results

We return to analyze our computational results more closely. We have already mentioned that the result of Theorem 16 was obtained by a computer program. In this section we will describe the script which we used to find 2-abelian cube-free words over a binary alphabet. The main idea is the same as in the scripts used to find the finite limit for the length of ternary 2-abelian square-free words and to produce long ternary 3-abelian square-free words. Our observations based on computational tests have supported the conjectures that 3-abelian squares would be avoidable over ternary alphabets as well as 2-abelian cubes would be avoidable over binary alphabets. Later these both conjectures have been shown to be true by Michaël Rao in [87]. Rao also used the computer checking as a tool for the results.

We will first examine the tests made for Problem 1, i.e., tests for 2- and 3-abelian square-freeness over a ternary alphabet. Then we will show the results from tests made for Problem 2, i.e., tests for 2-abelian cube-freeness over a binary alphabet. The codes for generating binary 2-abelian cube-free words and ternary 3-abelian square-free words are given in Appendices A and B as examples of the used scripts. Explanations for the codes are given in the end of this section.

After we had obtained the bound 537 for the length of 2-abelian square-free words we constructed each of these shorter ternary 2-abelian square-free words. We analyzed the number of such words for each of the lengths from 1 to 537, respectively. The sizes of the sets containing ternary 2-abelian square-free words are shown in Figure 2.1. There exist 404 286 words of length 105 and this is the biggest set. The number of words grows monotonically from the length 1 up to 103 and the corresponding numbers of words are 3 and 403 344. After the length 103 the behaviour of the number

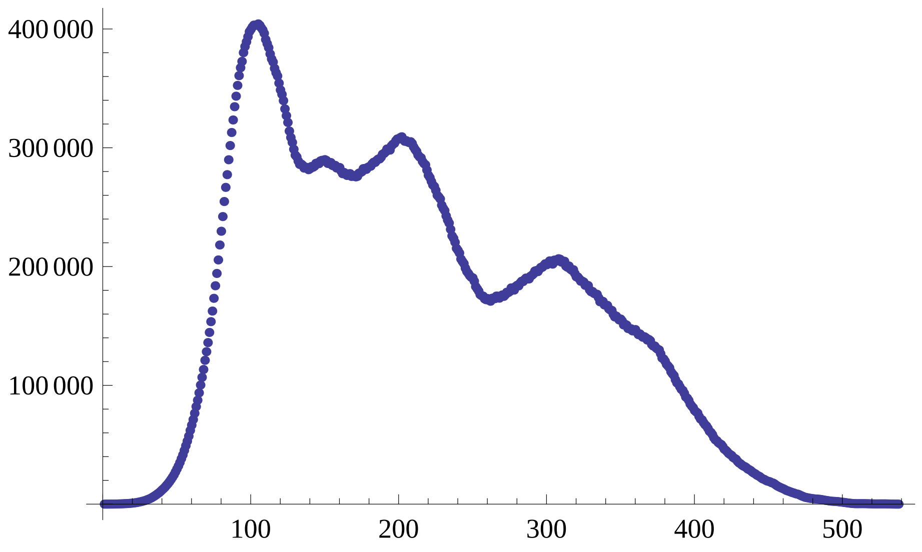


Figure 2.1: The number of ternary 2-abelian square-free words with respect to their lengths.

of words turns out to be strange. There exist surprising peaks and valleys and we do not have an explanation for that. It is also worth to note that it is not typical to have a word longer than 500 letters that would avoid a pattern but not to be able to construct such an infinite word. Another similarly computationally a bit surprising result was found in paper [51]. There it was shown that all the factors of length 24 of an infinite binary word may have the form ux_1x_2v where $|u| = |v|$ and $x_1 \equiv_2 x_2$ without implying the periodicity for the infinite word. These both examples emphasize the unpredictable behaviour of k-abelian repetitions and the computational challenges the notion may lead to.

We used the same technique to produce longer and longer 3-abelian square-free ternary words in a lexicographic order as we had used to produce 2-abelian square-free words (see Appendix B). Now we obtained words containing more than 100 000 letters and the numbers of 3-abelian square-free words over a ternary alphabet seem to grow exponentially, at least for small values of n, see Figure 2.2.

The results for words avoiding 2-abelian cubes over a binary alphabet resembles the computations done for the ternary alphabet and 3-abelian square-free words. We were able to construct a binary word of more than 100 000 letters that still avoids 2-abelian cubes. Again we can search for the numbers of binary 2-abelian cube-free words of different lengths. The numbers of such words with lengths from 1 to 60 grow approximately with a factor 1.3 at each increment of the length, see Figure 2.3. So that the number

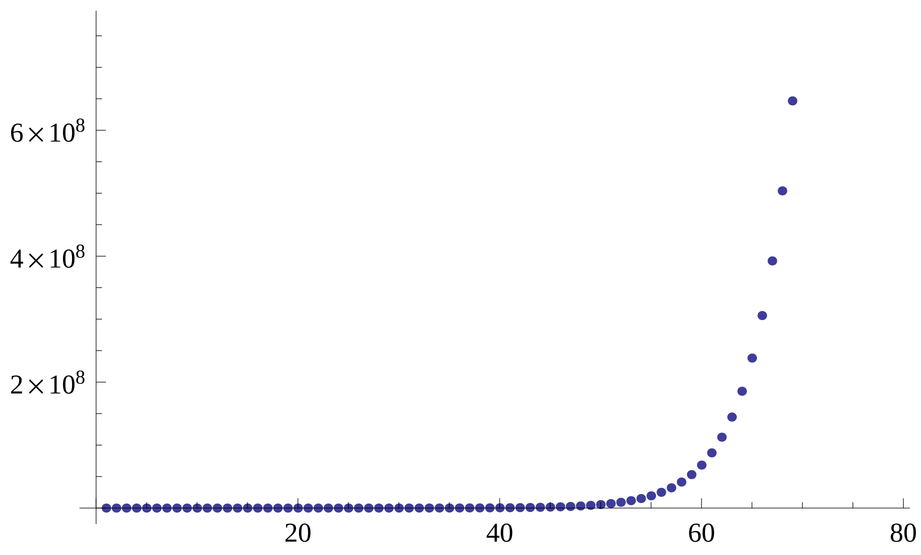


Figure 2.2: The number of ternary 3-abelian square-free words with respect to their lengths.

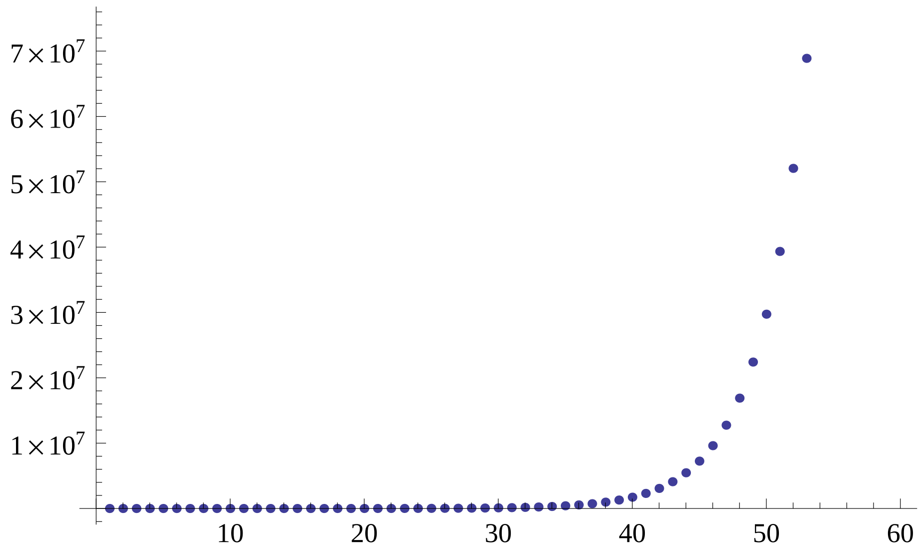


Figure 2.3: The number of binary 2-abelian cube-free words with respect to their lengths.

of binary 2-abelian cube-free words of length 60 is already 478 456 030. Another remark is that there exist more binary 2-abelian cube-free words (254) than ternary 2-abelian square-free words (240) already for length 12.

We have now counted the numbers of ternary 2- and 3-abelian square-free words and binary 2-abelian cube-free words. For comparison, the number of ordinary ternary square-free words of length n is listed as a sequence (A006156) in The On-Line Encyclopedia of Integer Sequences (OEIS) [82] and the number of binary cube-free words of length n is listed as a sequence (A028445). For the three other k-abelian cases presented above there cannot be found a corresponding sequence from OEIS. The numbers of binary 2-abelian cube-free and ordinary cube-free words coincide for the words up to 17 letters. Over a ternary alphabet and for square-free words and 2-abelian square-free words or 3-abelian square-free words the numbers coincide up to words of 11 letters or 23 letters, respectively.

We also chose some binary 2-abelian cube-free prefixes and counted the numbers of binary 2-abelian cube-free words having these fixed prefixes. In this way we can check how many suitable extensions the chosen 2-abelian cube-free word has. As a result, we found examples of binary 2-abelian cube-free words with the property that the number of their extensions again grows approximately with a factor 1.3 when increasing the length of extensions by one. In Figure 2.4 this is done for a fixed prefix of length 2000.

These results were starting points for the searching of infinite words that would avoid 3-abelian squares and 2-abelian cubes. The results we have proved in this thesis up to this point have been unavoidability results. In the next section we will introduce the first avoidability results concerning k-abelian square- and cube-freeness. Before that we will explain a few details of attaining our computational results.

The code for generating 2-abelian cube-free words over a binary alphabet $\{0, 1\}$ is presented in Appendix A. The structure of the code is quite straightforward and it is guaranteed that the execution of the code ends in some point. User sets a limit for the length of words that will be generated. The execution ends when this length is obtained, at the latest. It also ends if there does not exist any new 2-abelian cube-free word. The idea is to build up longer and longer words in lexicographic order and check whether they are 2-abelian cube-free. If the word does not contain 2-abelian cube then we increase the length by one. If the word contains a 2-abelian cube then we go backwards and continue with the next possible word in lexicographic order. The 2-abelian cube-freeness is checked by comparing the first and the last letters and the numbers of factors 00, 11, and the total number of factors 01 and 10. The number of factors 01 and the number of factors 10 are dependent as mentioned in the proof of Lemma 7, so it is enough to count the total number of those factors. All the other codes used for computational results given in this section and in Theorem 16 have the same basic ideas.

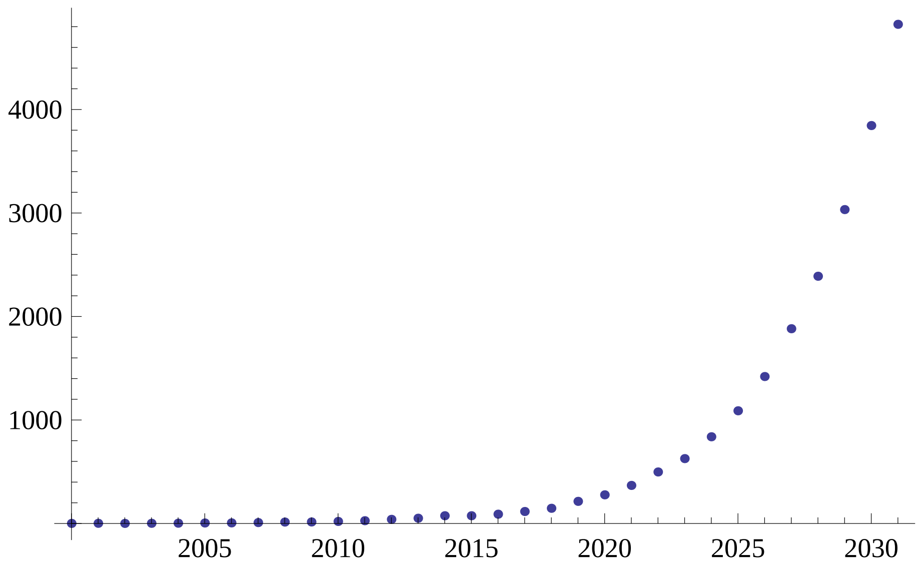


Figure 2.4: The number of binary 2-abelian cube-free words of lengths 2000-2031 having a fixed prefix of length 2000.

The code for generating 3-abelian square-free words over a ternary alphabet $\{0, 1, 2\}$ is presented in Appendix B. The structure of the code is similar to the structure of the code in Appendix A. Now the code is a bit more complicated because of the ternary alphabet and the 3-abelian equivalence relation. To count the number of different factors of length 3 we have to count the numbers of factors 010, 012, 020, 021, 101, 102, 120, 121, 201, 202, 210 and 212. The script for generating 2-abelian square-free words over a ternary alphabet can be obtained easily by simplifying the script given for the 3-abelian case. For 2-abelian case it is enough to count the numbers of factors 01, 02, 10, 12, 20 and 21. So the most significant change is to simplify `factorcount(int beg, int end)` to count the factors of length 2 instead of length 3. Also the lengths of the prefixes and suffixes, that have to be checked, decrease from 2 to 1.

2.3.3 Avoidability results

This section deals with results in which an infinite word avoiding some k-abelian repetitions can be proved to exist. We present the first positive results for problems 1 and 2, i.e. for avoiding k-abelian squares over a ternary alphabet and k-abelian cubes over a binary alphabet. The values for k are 64 and 8 in these results, respectively. So we have a positive result for both of the main questions. The extended versions of the problems ask now what are the minimal values for k to achieve the avoidability. Aleksi

Saarela and Robert Mercas have been able to decrease the value of k all the way to 3 for k-abelian cube-freeness over a binary alphabet, see [77, 78]. Finally, in a manuscript by Michaël Rao [87] the avoidability was obtained for 2-abelian cubes over a binary alphabet and for 3-abelian squares over a ternary alphabet. These reveal the optimal values of k because we have already shown that 2-abelian squares are not avoidable in infinite ternary words.

We start by an example of a ternary square-free word. The technique we use to prove that the given simple morphism generates a ternary square-free word has some similarities with the technique we use to prove the results in k-abelian case. The idea is that the form of the morphism restricts in an obvious way the possible lengths of squares and gives an easy way to determine the positions of some factors in the generated infinite word. The next Example 36 and the following two theorems 37 and 38 were established originally in the paper [50]. Actually, Saarela was responsible for these discoveries and the results are given here for self-containedness.

Example 36. Consider the morphism

$$g : \begin{cases} a \mapsto abcbacbcabcba \\ b \mapsto bcacbacabcacb \\ c \mapsto cabacbabcabac \end{cases} .$$

Let

$$t = g^{\infty}(a) = \prod_{i=1}^{\infty} x_i s_i,$$

where $x_i \in \{a, b, c\}$, $x_i s_i = g(x_i)$ and, moreover,

$$t = g^{-1}(t) = x_1 x_2 \dots . \tag{2.2}$$

It was proved by Leech [65] that the infinite word t is square-free. The words $g(a)$, $g(b)$ and $g(c)$ have equal length, are palindromes and can be obtained from each other by cyclically permuting the three letters. The morphism g is the simplest square-free morphism with these symmetry properties, but there are shorter uniform morphisms that are square-free [93].

To prove the square-freeness there are few details that need to be checked first: it must be verified that t does not contain a square of a word of length less than eight, and that the *starting position* of every factor of length eight is uniquely determined modulo $|g(a)| = 13$. These two conditions can be checked mechanically.

Now, assume that t contains the shortest square $u_1 u_2$ with $u_1 = u_2 = u$. Then $|u| \geq 8$. If $|u|$ would not be divisible by 13, then the prefixes of u_1 and u_2 of length eight would be in different positions modulo 13, and hence they would be different. So $|u|$ must be divisible by 13. Denote by v_i, for $i = 1, 2$, the scattered subword of u_i formed by the occurrences of x_j in u_i. Since $|u|$ is divisible by 13, necessarily $v_1 = v_2$, and hence by (2.2) t contains a shorter square $v_1 v_2$, which is a contradiction.

It is clear that t has a repetition of order 15/8. For instance, $g(aba)$ contains the factor $ag(b)a = abcacbacabcacba$. In fact, the proof above can be modified to show that there are no higher powers.

Now we move on to study k-abelian repetitions. First we will show that 8-abelian cubes can be avoided over a two-letter alphabet. It is known that abelian squares can be avoided over a 4-letter alphabet, see [62]. We show that starting from such a square-free word over a 4-letter alphabet and mapping it to an infinite binary word with a uniform morphism we can produce an infinite 8-abelian cube-free word over a binary alphabet.

We need the following notation. If $u = a_0 \dots a_{n-1}$, where a_i are letters and $0 \le i \le j \le n$, then we let $u[i : j] = a_i \dots a_j$.

Theorem 37. *Let $w \in \{0,1,2,3\}^\omega$ be an abelian square-free word. Let $k \le n$ and $h : \{0,1,2,3\}^* \to \{0,1\}^*$ be an n-uniform morphism that satisfies the following conditions:*

1. *if $u \in \{0,1,2,3\}^4$ is square-free, then $h(u)$ is k-abelian cube-free,*
2. *if $u \in \{0,1,2,3\}^*$ and v is a factor of $h(u)$ of length $2k-2$, then every occurrence of v in $h(u)$ has the same starting position modulo n,*
3. *there is a number i such that $0 \le i \le n-k$ and for at least three letters $x \in \{0,1,2,3\}$, $v = h(x)[i : i+k-1]$ satisfies $|h(u)|_v = |u|_x$ for every $u \in \{0,1,2,3\}^*$.*

Then $h(w)$ is k-abelian cube-free.

Proof. The first condition prohibits short k-abelian cubes in $h(w)$. If $h(w)$ contained a k-abelian cube of length less than $3k$, then this cube would be a factor of $h(u)$ for some $u \in \{0,1,2,3\}^4$, where u is a factor of w and thus square-free.

The second condition restricts the length of every k-abelian cube in $h(w)$ to be divisible by $3n$. If $h(w)$ contained a k-abelian cube pqr, where $|p| = |q| = |r| = m \ge k$, then

$$p[m-k+1 : m-1] \cdot q[0 : k-2] = q[m-k+1 : m-1] \cdot r[0 : k-2].$$

and the starting positions of these factors would differ by m. Now m is divisible by n showing that the length of every k-abelian cube in $h(w)$ is divisible by $3n$.

By using the third condition we show that a k-abelian cube in $h(w)$ would lead to an abelian square in w. Let $a'a_1 \dots a_s b' b_1 \dots b_s c' c_1 \dots c_s d'$ be a factor of w, where $a_j, b_j, c_j, a', b', c' \in \{0,1,2,3\}$, so that pqr is a k-abelian cube in $h(w)$ with

$$p = p_1 h(a_1 \dots a_s) p_2, \quad q = q_1 h(b_1 \dots b_s) q_2, \quad r = r_1 h(c_1 \dots c_s) r_2,$$

where p_1 is a suffix of $h(a')$, $p_2 q_1 = h(b')$, $q_2 r_1 = h(c')$, r_2 is a prefix of $h(d')$, $|p_1| = |q_1| = |r_1|$ and $|p_2| = |q_2| = |r_2|$. Let i be the number and a, b, c the three letters in condition 3. Let $|p| = m$, $|p_2| = j$ and $v_x = h(x)[i : i + k - 1]$ for $x \in \{a, b, c\}$. There are three cases.

If $j \leq i$, then p_2 is too short to contain v_x and $h(a')$ contains v_x if and only if p_1 contains v_x for $x \in \{a, b, c\}$. Similarly for q_2, $h(b')$ and q_1. This gives by condition 3

$$|a'a_1 \dots a_s|_x = |h(a'a_1 \dots a_s)|_{v_x} = |p|_{v_x} = |q|_{v_x} = |h(b'b_1 \dots b_s)|_{v_x} = |b'b_1 \dots b_s|_x$$

for $x \in \{a, b, c\}$. Thus $a'a_1 \dots a_s$ and $b'b_1 \dots b_s$ are abelian equivalent, which contradicts the abelian square-freeness of w.

If $j \geq i + k$, then respectively

$$|a_1 \dots a_s b'|_x = |h(a_1 \dots a_s b')|_{v_x} = |p|_{v_x} = |q|_{v_x} = |h(b_1 \dots b_s c')|_{v_x} = |b_1 \dots b_s c'|_x$$

for $x \in \{a, b, c\}$, so $a_1 \dots a_s b'$ and $b_1 \dots b_s c'$ are abelian equivalent, which is a contradiction.

If $i < j < i + k$, then any of p_1, p_2, q_1 or q_2 cannot contain v_x for $x \in \{a, b, c\}$, which gives

$$|a_1 \dots a_s|_x = |h(a_1 \dots a_s)|_{v_x} = |p|_{v_x} = |q|_{v_x} = |h(b_1 \dots b_s)|_{v_x} = |b_1 \dots b_s|_x$$

for $x \in \{a, b, c\}$. Further, $v_{b'}$ is a factor of $t = p[m - k + 1 : m - 1] q[0 : k - 2]$ and $v_{c'}$ is a factor of $q[m - k + 1 : m - 1] r[0 : k - 2]$, which is the same word as t. Now $v_{b'}$ and $v_{c'}$ have the same starting positions in t, so $v_{b'} = v_{c'}$, and $b' = c'$ by condition 3. Thus $a_1 \dots a_s b'$ and $b_1 \dots b_s c'$ are abelian equivalent. This contradiction completes the proof. □

Now we can prove the existence of an infinite binary word which avoids 8-abelian cubes.

Theorem 38. *Let $w \in \{0,1,2,3\}^\omega$ be an abelian square-free word. Let $h : \{0,1,2,3\}^* \to \{0,1\}^*$ be the morphism defined by*

$$\begin{aligned} h(0) &= 00101\ 0\ 011001\ 0\ 01011, \\ h(1) &= 00101\ 0\ 011001\ 1\ 01011, \\ h(2) &= 00101\ 1\ 011001\ 0\ 01011, \\ h(3) &= 00101\ 1\ 011001\ 1\ 01011. \end{aligned}$$

Then $h(w)$ is 8-abelian cube-free.

Proof. Condition 1 of Theorem 37 is satisfied for $k \geq 4$.

Condition 2 of Theorem 37 is satisfied for $k \geq 6$.

Condition 3 of Theorem 37 is satisfied for $k = 8$ and $i = 5$. The three letters are 0, 1 and 3, and the corresponding factors are 00110010, 00110011 and 10110011.

The claim now follows from Theorem 37. $\square$

The satisfiability of the three conditions in the previous proof can be easily checked by computer as well as by paper and pencil. We remark that for the case $k = 2$ the requirements of the last condition are too strict because there exist only four different binary words of length 2. Thus Theorem 37 cannot be applied for the case $k = 2$.

Next we will introduce the first affirmative result of avoiding k-abelian squares over a ternary alphabet. We have already shown in Theorem 26 that this word cannot be pure morphic. The word we will construct is a morphic word, i.e., a morphic image of a pure morphic word like the 8-abelian cube-free word in Theorem 38. In addition, the pure morphic word which is used as a starting point can be chosen to be the same word introduced by Keränen in [62]. This infinite abelian square-free word is obtained by iterating an 85-uniform morphism over a four letter alphabet. The other morphism we will use is defined by Badkobeh and Crochemore in [5]. For convenience, we will denote that morphism from $\{a, b, c, d\}^*$ into $\{0, 1, 2\}^*$ by g. The morphism g is defined as follows:

$$\begin{cases} a \mapsto & 0102101202102010210121020120210120102120121020120210121 \\ & 0212010210121020102120121020120210121020102101202102012 \\ & 10212010210121020120210120102120121020102101210212, \\ b \mapsto & 0102101202102010210121020120210120102120121020120210121 \\ & 0201021012021020121021201021012102010212012102012021012 \\ & 10212010210121020120210120102120121020102101210212, \\ c \mapsto & 0102101202102010210121020120210120102120121020102101202 \\ & 1020121021201021012102010212012102012021012102010210120 \\ & 21020102120121020120210120102120121020102101210212, \\ d \mapsto & 0102101202102010210121020120210120102120121020102101202 \\ & 1020102120121020120210121020102101202102012102120102101 \\ & 21020102120121020120210120102120121020102101210212. \end{cases}$$

Badkobeh and Crochemore proved the following result for g in [5]:

Theorem 39 ([5]). *The morphism g translates any infinite $7/5^+$-free word on the alphabet $\{a, b, c, d\}$ into a $7/4^+$-free ternary word containing only two $7/4$-powers.*

Before stating Theorem 42 we will define a few notions for the proof of it. *An identifying factor*, *an identifying prefix* and *an identifying suffix* are all defined with respect to a morphism.

Definition 40. *An identifying factor with respect to a morphism $h : \Sigma_0^* \to \Sigma_1^*$ is such a factor $f \in \Sigma_1^+$ that*

(a) there exists a unique $a \in \Sigma_0$ such that $f \in F(h(a))$, and

(b) for this a, $|h(w)|_f = |w|_a \cdot |h(a)|_f$ for all $w \in \Sigma_0^*$.

An identifying prefix with respect to a morphism $h : \Sigma_0^* \to \Sigma_1^*$ is such a prefix $p \in \Sigma_1^+$ of some word $h(i)$ where $i \in \Sigma_0$ that

(a) there exists a unique $a \in \Sigma_0$ such that $p \in \mathrm{pref}(h(a))$, and

(b) for this a, $|h(w)|_p = |w|_a$ for all $w \in \Sigma_0^*$.

An identifying suffix is defined correspondingly.

These identifying objects with respect to a morphism $h : \Sigma_0^* \to \Sigma_1^*$ can be used to track down properties of a word $w \in \Sigma_0^*$ by analysing the factors of the word $h(w)$. If $h(a)$ for some $a \in \Sigma_0$ contains an identifying factor f then we can find out $|w|_a$ by counting $|h(w)|_f$ and $|h(a)|_f$. Identifying prefixes and suffixes can be used to locate letters in w. We give a short example to illustrate these notions and to show how we will use them.

Example 41. Let $h : \{a, b, c\}^* \to \{a, b, c\}^*$ be a morphism defined by

$$\begin{cases} a \mapsto abcab, \\ b \mapsto bcb, \\ c \mapsto cacb. \end{cases}$$

Now the six shortest identifying factors with respect to h are ab, ac, acb, bcb, cab and cac. From these six factors ab and ac are not identifying prefixes or suffixes but cac is an identifying prefix, acb and cab are identifying suffixes and bcb is both an identifying prefix and an identifying suffix. Consider a word $w' = bcbabcabcacbabcab$ which is obtained by taking a h-morphic image of some word w over an alphabet $\{a, b, c\}$. By counting the occurrences of the identifying factor ab in the word w' we can determine that w contains the letter a twice. Because ab is not an identifying prefix or suffix we can not locate the positions of these a's by looking for the positions of ab's in w'. Instead, cab is an identifying suffix related to $h(a)$ and we can determine that $w' = bcb\,h(a)\,cacb\,h(a)$. By continuing analysis with identifying prefixes bcb and cac we can find out that $w = baca$.

We remind also the meaning of *a synchronizing morphism.* A morphism $h : \Sigma^* \to \Sigma'^*$ is said to be synchronizing if for all letters $a, b, c \in \Sigma$ and words $l, r \in \Sigma'^*$ from $h(ab) = lh(c)r$ it follows that either $l = \epsilon$ and $a = c$ or $r = \epsilon$ and $b = c$.

Theorem 42. *Let $\Omega \in \{a, b, c, d\}^\omega$ be an infinite abelian square-free word and $g : \{a, b, c, d\}^* \to \{0, 1, 2\}^*$ the morphism introduced on the page 39. The infinite word $\omega = g(\Omega)$ over $\{0, 1, 2\}$ is 64-abelian square-free.*

Proof. First, we make some remarks about the morphism g and the word Ω. First of all, the word Ω exists and for example, the word constructed by Keränen could be chosen to be Ω. The morphism g is 160-uniform and synchronizing. It is of the form

$$g : \begin{cases} a \mapsto uv\alpha yz \\ b \mapsto uv\beta yz \\ c \mapsto uw\gamma xz \\ d \mapsto uw\delta xz \end{cases}, \text{ where } \begin{cases} |u| = 47, |z| = 40, |v| = |x| = 10, |w| = |y| = 13, \\ |\alpha| = |\beta| = |\gamma| = |\delta| = 50. \end{cases}$$

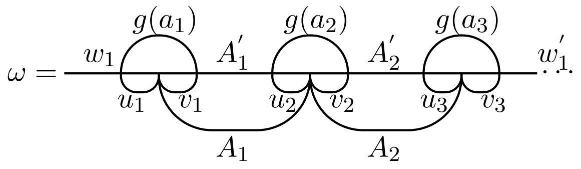
The word Ω can only contain factors ij where $i, j \in \{a, b, c, d\}$ and $i \neq j$. It can be easily checked by computer that now each word $g(ij)$ contains each of its factors of length 63 at most once except in cases $g(bd) = pfqrfs$ and $g(db) = rfspfq$ where $|p| = 32, |f| = 78, |q| = 50, |r| = 57, |s| = 25$. In these cases there are 16 factors of length 63 that occur twice, namely the factors of f. Thus each $g(i)$ for $i \in \{a, b, c, d\}$ contains at least one identifying factor of each length from 63 up to 160 and each $g(i)$ contains also an identifying prefix and suffix of length 63 for any $i \in \{a, b, c, d\}$. In addition, the factor $\text{pref}_{30}(u)$ occurs only as a prefix of $g(a), g(b), g(c)$ and $g(d)$. Respectively for the factor $\text{suf}_{30}(z)$.

If the word ω contains a factor of length 63 twice they can not overlap by the observations above, so $\omega = \omega_0 t \omega_1 t \omega_2$ where $|t| = 63$. Now $|t\omega_1| = n \cdot 160 + \Lambda_1 \cdot 135 + \Lambda_2 \cdot 25$, where $n \in \{0, 1, 2, \ldots\}$ and $(\Lambda_1, \Lambda_2) \in \{(0, 0), (0, 1), (1, 0)\}$. Here $\Lambda_1 \neq \Lambda_2$ if and only if $t \in F(f)$. In most of the cases the two occurrences of t originate from the images of two occurrences of the same letter in Ω. Consequently, the coefficient 160 comes from the length of the morphism. In the case that the first occurrence of t belongs to f in $g(d)$ and the second occurrence of t belongs to f in $g(b)$, i.e. $\Lambda_1 = 1$, there exists extra factor of 135 referring to $|fsp|$. If the occurrences of $t \in F(f)$ are in the opposite order, thus first in $g(b)$ and then in $g(d)$, then we have $\Lambda_2 = 1$ and $|fqr| = 185 = 160 + 25$.

We proceed by showing that if ω contained a 64-abelian square then Ω would not be abelian square-free or $g(\Omega)$ would not be square-free which would give a contradiction. We use identifying factors and prefixes to return our analysis to the properties of Ω. So assume that the word ω contains a 64-abelian square, i.e., $\omega = wA_1A_2w'$ and $A_1 \equiv_{64} A_2$. If $|A_1| \leq 64$ then $A_1 = A_2$ and some of words $g(ij)$, where $ij \in \{a,b,c,d\}\,(\{a,b,c,d\} \setminus \{i\})$, should contain a square which contradicts the result of Theorem 39. So we may assume that $|A_1| > 64$.

Because $A_1 \equiv_{64} A_2$ we have $\mathrm{pref}_{63}(A_1) = \mathrm{pref}_{63}(A_2)$ and ω has the same factor of length 63 twice. Similarly, $\mathrm{suf}_{63}(A_1) = \mathrm{suf}_{63}(A_2)$ and thus $|A_1| = n_p \cdot 160 + \Lambda_{1p} \cdot 135 + \Lambda_{2p} \cdot 25$ and $|A_2| = n_s \cdot 160 + \Lambda_{1s} \cdot 135 + \Lambda_{2s} \cdot 25$. Now $|A_1| = |A_2|$ and the only possibility is that $n_p = n_s, \Lambda_{1p} = \Lambda_{1s}$ and $\Lambda_{2p} = \Lambda_{2s}$. If $\Lambda_{1p} = 1$, then both $\mathrm{suf}_{63}(A_1) \in F(f)$ and $\mathrm{pref}_{63}(A_2) \in F(f)$. In fact, $\mathrm{suf}_{63}(A_1)\mathrm{pref}_{63}(A_2)$ should be a factor of f which is not possible because $|\mathrm{suf}_{63}(A_1)\mathrm{pref}_{63}(A_2)| > |f|$. Similar reasoning holds if $\Lambda_{2p} = 1$. So we have $\Lambda_{1p} = \Lambda_{2p} = \Lambda_{1s} = \Lambda_{2s} = 0$ and $|A_1| = |A_2| = n_p \cdot 160$.

Let $\omega = wA_1A_2w' = w_1g(a_1)A_1'g(a_2)A_2'g(a_3)w_1'$, where a_1, a_2, a_3 are letters in $\{a,b,c,d\}$, $g(a_1) = u_1v_1, g(a_2) = u_2v_2, g(a_3) = u_3v_3$ and $v_1A_1'u_2 = A_1$ and $v_2A_2'u_3 = A_2$. The following graph illustrates the situation.



Because $|A_1| = |A_2| = n_p \cdot 160$ we have $|u_1| = |u_2| = |u_3|$ and $|v_1| = |v_2| = |v_3|$. Now we can divide the study into two cases.

If $|u_1| = 0$ then $A_1 = g(\alpha_1), A_2 = g(\alpha_2)$ for some $\alpha_1\alpha_2 \in F(\Omega)$. Because each $g(i)$ for any $i \in \{a,b,c,d\}$ contains at least one identifying factor of length 64 and $A_1 \equiv_{64} A_2$ so α_1 should be abelian equivalent to α_2. This gives a contradiction because Ω is abelian square-free.

If $|u_1| = m > 0$ then $A_1 = v_1g(\alpha_1)u_2, A_2 = v_2g(\alpha_2)u_3$ for some $\alpha_1a_2\alpha_2 \in F(\Omega)$. We may assume $|u_2| > 63$, the case $|v_2| > 63$ would be similar. Now $\mathrm{suf}_{63}(u_2) = \mathrm{suf}_{63}(A_1) = \mathrm{suf}_{63}(A_2) = \mathrm{suf}_{63}(u_3)$ and $|u_2v_2g(\alpha_2)| = (|\alpha_2| + 1) \cdot 160$ so u_2 have to be equal to u_3 and $a_2 = a_3$, too. Because each $g(i)$ for any $i \in \{a,b,c,d\}$ has an identifying prefix of length 64 and $A_1 \equiv_{64} A_2$, so $g(\alpha_1a_2)$ and $g(\alpha_2a_3) = g(\alpha_2a_2)$ have to have those same identifying factors, so $\alpha_1 \equiv_{\mathrm{a}} \alpha_2$, too. This gives a contradiction because now $\alpha_1a_2\alpha_2a_3 \in F(\Omega)$ and $\alpha_1a_2 \equiv_{\mathrm{a}} \alpha_2a_3$. □

Now we have proved the existence of a 64-abelian square-free word. By choosing the initial abelian square-free word to be morphic we get that the 64-abelian square-free word constructed as in Theorem 42 is also morphic. In general, let Ω be a morphic word over $\{a,b,c,d\}$ then $g(\Omega)$ is a 64-abelian square-free word but by Theorem 26 there does not exist a single morphism that would generate that word directly.

As mentioned, both results of Theorem 38 and Theorem 42 can be improved. In a manuscript [87] by Rao he has been able to show that avoidance can be obtained even for 2-abelian cubes over a binary alphabet and 3-abelian squares over a ternary alphabet. These are the optimal values for k enabling positive answers to problems 1 and 2. First Rao gives a set of conditions which ensure that a given morphism is k-abelian nth power-free. This results is given without a proof in Theorem 43. This resembles the set of conditions that assure a morphism to be abelian nth power-free which was given by Carpi in [14]. Though, these new conditions does not give a characterization for k-abelian nth power-free morphisms. Then Rao claims that there exist such morphisms that meet the requirements of Theorem 43 for the values $k = 2$ and $n = 3$ as well as $k = 3$ and $n = 2$. The checking was done by computer. Applying those morphism respectively to an infinite abelian cube-free word over a ternary alphabet and an infinite abelian square-free word over a four-letter alphabet the asked 2-abelian cube-freeness and 3-abelian square-freeness are obtained. For example, the word introduced by Keränen can again be used as the abelian square-free word over a four-letter alphabet.

For the next theorem we denote by $\Psi(u)$ the Parikh vector of u. For a set $S \subseteq \Sigma^*$, $\Psi_S(u)$ denotes the vector indexed by S such that $\Psi_S(u)[w] = |u|_w$ for every $w \in S$. For convenience, if the alphabet Σ is clear from the context then $\Psi_{\Sigma^k}(u)$ is denoted by $\Psi_k(u)$, for $k \geq 1$. So Ψ_k is now a k-generalized Parikh vector.

Theorem 43 ([87])**.** *We fix $k \geq 1$ and $n \geq 2$, and two alphabets Σ and Σ'. Let $h : \Sigma^* \to \Sigma'^*$ be a morphism. Suppose that:*

1. *For every abelian nth power-free word $w \in \Sigma^*$ with $|w| \leq 2$ or $|h(w[2 : |w| - 1])| \leq (k-2)n - 2$, $h(w)$ is k-abelian nth power-free.*

2. *There are $p, s \in \Sigma'^{k-1}$ such that for every $a \in \Sigma$, $p = \mathrm{pref}_{k-1}(h(a)p)$ and $s = \mathrm{suf}_{k-1}(sh(a))$.*

3. *The matrix N indexed by $\Sigma'^k \times \Sigma$, with $N[w, x] = |h(x)p|_w$, has rank $|\Sigma|$.*

4. *Let $S \subseteq \Sigma'^k$, with $|S| = |\Sigma|$, such that the matrix M indexed by $S \times \Sigma$, with $M[w, x] = |h(x)p|_w$, is invertible. Let*

$$\Psi_S(v, u) = \Psi_S(vp) + \Psi_S(su) - \Psi_S(sp)$$

and $\Psi_k(v, u) = \Psi_{\Sigma'^k}(v, u)$. For every $a_i \in \Sigma$, $u_i, v_i \in \Sigma^*$ such that $u_i v_i = h(a_i)$; $0 \leq i \leq n$; such that:

- $|\{\mathrm{pref}_{k-1}(v_i p) : 0 \leq i \leq n\}| = 1$,
- $M^{-1}(\Psi_S(v_{i-1}, u_i) - \Psi_S(v_i, u_{i+1}))$ *is an integer vector, for every* $1 \leq i < n$,
- $\Psi_k(v_{i-1}, u_i) - \Psi_k(v_i, u_{i+1}) \in im(N)$ *for every* $1 \leq i < n$,

there is a $(\alpha_0 \cdots \alpha_n) \in \{0,1\}^{n+1}$ *such that for every* $1 \leq i < n$:

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

Then h is k-abelian nth power-free.

The following morphism were given in [87] as examples of 2-abelian cube-free and 3-abelian square-free morphisms. These are not the only that were given in the paper. These were chosen because they are uniform morphisms and in addition, h_2 is the smallest uniform 2-abelian cube-free morphism that was found in the paper.

$$h_2 : \begin{cases} 0 \mapsto 001001010010110010010100100110010011001011010 11 \\ 1 \mapsto 00100110010011001101100110110010011001101101011 \\ 2 \mapsto 00110110101101001011010110100101001001101101011 \end{cases}$$

$$h_3 : \begin{cases} 0 \mapsto 010201202101201020121021 2 \\ 1 \mapsto 0102101201021201210120212 \\ 2 \mapsto 0102101210212021020120212 \\ 3 \mapsto 0121020120210201210120212 \end{cases}$$

Theorem 44. *There exists an infinite 2-abelian cube-free word over a binary alphabet, i.e., 2-abelian cubes are avoidable over binary alphabets.*

Proof. By conditions of Theorem 43 h_2 is a 2-abelian cube-free morphism and maps an abelian cube-free word to a 2-abelian cube-free word. For example, a fixed point of morphism $\mu : 0 \mapsto 0012, 1 \mapsto 112, 2 \mapsto 022$ is an infinite abelian cube-free word by Dekking [27]. Now $h_2(\mu^\infty(0))$ is an infinite 2-abelian cube-free word over a binary alphabet. $\square$

Theorem 45. *There exists an infinite 3-abelian square-free word over a ternary alphabet, i.e., 3-abelian squares are avoidable over ternary alphabets.*

Proof. By conditions of Theorem 43 h_3 is a 3-abelian square-free morphism and maps an abelian square-free word to a 3-abelian square-free word. For example, a fixed point of the morphism γ given by Keränen [62] is an infinite abelian square-free word. Now $h_3(\gamma^\infty(0))$ is an infinite 3-abelian square-free word over a ternary alphabet. $\square$

As a conclusion we can present the completed version of Table 2.1 that in the beginning summarized the older results and told us the limits for our k-abelian avoidability problems. Now Table 2.2 shows how k-abelian square-freeness and k-abelian cube-freeness behave in comparison with square- and abelian square-freeness and cube- and abelian cube-freeness.

Avoidability of squares					Avoidability of cubes				
size of	type of rep.				size of	type of rep.			
the alph.	$=$	$\equiv_{k>2}$	$\equiv_2$	$\equiv_a$	the alph.	$=$	$\equiv_{k>2}$	$\equiv_2$	$\equiv_a$
2	$-$	$-$	$-$	$-$	2	$+$	$+$	$+$	$-$
3	$+$	$+$	$-$	$-$	3	$+$	$+$	$+$	$+$
4	$+$	$+$	$+$	$+$					

Table 2.2: Avoidability of different types of repetitions in infinite words.

We can note from Table 2.2 that the behaviour of k-abelian power-freeness depends on the value of k. If we talk about k-abelian cube-freeness the behaviour is similar to the cube-freeness for all $k \geq 2$. For k-abelian square-freeness the behaviour is similar to the abelian square-freeness for $k = 2$ and for the bigger values of k the behaviour is similar to the square-freeness. These results were not guessed in the beginning of the study.

2.3.4 Unavoidability of weakly k-abelian squares

In this section we talk about k-abelian squares but we will define the notion of a square in a bit unusual way. Carpi and De Luca have studied square-freeness in partially commutative monoids earlier in [16]. Their approach to square-freeness is similar but not identical to our interpretation. Another related concept is that of *approximate squares*, which can be defined, for example, as words of the form uv, where the so called Hamming distance of u and v is "small enough" or equivalently as words w such that the Hamming distance of w and some square is "small enough". The first definition is analogous to the definition of *R-squares* we will give shortly and the latter definition is analogous to the definition of *weakly R-squares* which is the definition we concentrate on in this section. The avoidability of approximate squares has been studied by Ochem, Rampersad and Shallit [81].

Originally the results of this section were introduced in [52] and the ideas for the proofs of theorems were by Saarela. In the paper we used the term *strongly k-abelian repetitions* instead of *weakly k-abelian repetitions.* This weakly k-abelian repetition is more natural because if a word is a weakly (earlier strongly) k-abelian repetition it does not imply that the word would also be a k-abelian repetition. Though, when we discuss about words that avoid these weakly k-abelian repetitions we could say that these words are, for example, *strongly k-abelian square-free.* Whereas, this is still reasonable because a strongly k-abelian square-free word is also k-abelian square-free. So when talking about repetitions we define a weaker concept than k-abelian repetitions but when talking about repetition-freeness we define a stronger notion.

As mentioned in Section 2.1 k-abelian equivalence is, in fact, a congruence of words. This means that k-abelian equivalence is such an equivalence relation R that $uvRu'v'$ whenever uRu' and vRv'. As in the previous sections we are interested in the products of words which are k-abelian equivalent but we will first define squares for all congruences R. Higher powers can be then defined analogously.

If u and v are congruent words, then their product uv is an *R-square*. This is the definition we have used this far and which is a common definition in the study of abelian and k-abelian repetition-freeness, in general. In this section, however, we concentrate on another definition:

Definition 46. *A word w is a* weakly R-square *if it is congruent to a square of some non-empty word v, i.e. $wRvv$.*

For example, $aabb$ is not an abelian square because aa and bb are not abelian equivalent, but it is a weakly abelian square because it is abelian equivalent to $(ab)^2$. Next we will show an easy lemma that it does not matter whether the word is congruent to a square or to a R-square. In both cases the word is weakly R-square and, in fact, all of these words belong to the same equivalence class.

Lemma 47. *A word is a weakly R-square if and only if it is congruent to an R-square.*

Proof. It is clear that if word w is a weakly R-square then it is congruent to an R-square. Then assume for the other direction that w is congruent to an R-square, say $wRuv$ and uRv. Now we use the assumption that R is not just an equivalence relation but a congruence. So $wRuu$, because uRv implies $uvRuu$. □

It could be said that weakly R-squares take the concept of squares farther away from words and closer to the monoid defined by R.

Let us now state the definitions of weakly abelian and k-abelian nth powers for any $n \geq 1$.

Definition 48. *A word w is a weakly abelian nth power if it is abelian equivalent to a word which is an nth power.*

Definition 49. *A word w is a weakly k-abelian nth power if it is k-abelian equivalent to a word which is an nth power.*

The basic problem we are considering is avoidability of weakly abelian and weakly k-abelian nth powers. We prove that, for all k and n, they are unavoidable on all finite alphabets. First we show that in abelian case it is easy to see that there does not exist infinite word which would avoid a weakly abelian nth power. Recall that two words are abelian equivalent if and only if they have the same Parikh vectors.

Theorem 50. *Let Σ be an alphabet and let $n \geq 2$. Every infinite word $w \in \Sigma^\omega$ contains a non-empty factor that is abelian equivalent to an nth power.*

Proof. A word is abelian equivalent to an nth power if and only if its Parikh vector is zero modulo n. The number of different Parikh vectors modulo n is finite, so w has two prefixes u and uv such that their Parikh vectors are the same modulo n. Then the Parikh vector of v is zero modulo n, so v is abelian equivalent to an nth power. □

Theorem 50 can be generalized for k-abelian equivalence, but this is not straightforward. One important difference between abelian and k-abelian equivalence is that if a vector with non-negative elements is given, then a word having that Parikh vector can be constructed, but if for every $t \in \Sigma^k$ a non-negative number n_t is given, then there need not exist a word u such that $|u|_t = n_t$ for all t (see Example 53).

Perhaps the biggest difficulty in generalizing Theorem 50 lies in finding an analogous version of the fact that a word is abelian equivalent to an nth power if and only if its Parikh vector is zero modulo n. On the one direction we have:

Lemma 51. *If a word v of length at least $k-1$ is k-abelian equivalent to an nth power, then*

$$|v|_t + |\mathrm{suf}_{k-1}(v)\mathrm{pref}_{k-1}(v)|_t \equiv 0 \pmod{n} \tag{2.3}$$

for all $t \in \Sigma^k$.

Proof. Let v be k-abelian equivalent to u^n. Then

$$\begin{aligned} &|v|_t + |\mathrm{suf}_{k-1}(v)\mathrm{pref}_{k-1}(v)|_t = |v\mathrm{pref}_{k-1}(v)|_t \\ =&|u^n\mathrm{pref}_{k-1}(v)|_t = |u^n\mathrm{pref}_{k-1}(u^n)|_t = n|u\mathrm{pref}_{k-1}(u^n)|_t \equiv 0 \pmod{n} \end{aligned}$$

for all $t \in \Sigma^k$. □

The problem is that the converse does not hold. For example, $v = babbbbab$ satisfies (2.3) for $n = 2$ and $k = 3$ but it is not 3-abelian equivalent to any square. However, the converse does hold if for every t either $|v|_t$ is large enough or $|v\mathrm{pref}_{k-1}v|_t$ is zero. This is formulated precisely in

Lemma 54. To prove this we need the definitions and Lemma 13 introduced in Section 2.2 in connection with estimation of the number of k-abelian equivalence classes, see [59]. We continue by few examples which demonstrates these notions and the use of that lemma.

Example 52. Let $k = 3$ and consider the word $u = aaabaab$. The multigraph G_{f_u} is

$$\begin{tikzcd}
 & ab \arrow[d] \\
aa \arrow[loop left] \arrow[ur, bend left] \arrow[ur] & ba \arrow[l]
\end{tikzcd}$$

The word u corresponds to the Eulerian path

$$aa \to aa \to ab \to ba \to aa \to ab.$$

There is also another Eulerian path from aa to ab:

$$aa \to ab \to ba \to aa \to aa \to ab.$$

This corresponds to the word $aabaaab$, which is 3-abelian equivalent to u.

Example 53. Let us consider some functions $f : \{a, b\}^2 \to \mathbb{N}_0$.

If $f(aa) = f(bb) = 1$ and $f(t) = 0$ otherwise, then the underlying graph of G_f is not connected, so there does not exist a word u such that $f = f_u$.

If $f(ab) = 2$ and $f(t) = 0$ otherwise, then the indegree of a in G_f is zero but the outdegree is two, so there does not exist a word u such that $f = f_u$.

In the original version of the next lemma in [52] there was a little mistake which actually does not affect the essential parts of the proof. Now the requirement $|v|_t > (n-1)(k-1)$ or $|v|_t = 0$ for all $t \in \Sigma^k$ is replaced by $|v|_t > (n-1)(k-1)$ or $|v\text{pref}_{k-1}(v)|_t = 0$ for all $t \in \Sigma^k$.

Lemma 54. *If*

$$|v|_t + |\text{suf}_{k-1}(v)\text{pref}_{k-1}(v)|_t \equiv 0 \pmod{n} \tag{2.4}$$

and either $|v|_t > (n-1)(k-1)$ *or* $|v\text{pref}_{k-1}(v)|_t = 0$ *for all* $t \in \Sigma^k$*, then* v *is* k*-abelian equivalent to an* n*th power.*

Proof. Let $s_1 = \text{pref}_{k-1}(v)$ and $s_2 = \text{suf}_{k-1}(v)$. By Lemma 13 the underlying graphs G_{f_v} and $G_{f_{s_2 s_1}}$ are connected and,

$$\sum_{a\in\Sigma} f_v(as) = \sum_{a\in\Sigma} f_v(sa) + c_s \qquad \text{and} \qquad \sum_{a\in\Sigma} f_{s_2s_1}(as) = \sum_{a\in\Sigma} f_{s_2s_1}(sa) - c_s \tag{2.5}$$

for all $s \in \Sigma^{k-1}$, where

$$c_s = \begin{cases} -1, & \text{if } s = s_1 \neq s_2, \\ 1, & \text{if } s = s_2 \neq s_1, \\ 0, & \text{otherwise.} \end{cases}$$

In the latter equality the term $-c_s$ has negative sign because s_1 is now a suffix of $s_2 s_1$ and s_2 is a prefix of it. By assumptions of the lemma, a function $f : \Sigma^k \to \mathbb{N}_0$ can be defined by

$$f(t) = \frac{f_v(t) - (n-1) f_{s_2 s_1}(t)}{n}.$$

Clearly, $f(t) \in \mathbb{N}_0$ for all $t \in \Sigma^k$. If $|v\text{pref}_{k-1}(v)|_t = 0$ then $f_v(t) = f_{s_2 s_1}(t) = 0$, thus $f(t) = 0$. Otherwise $|v|_t > 0$, and then $f_v(t) = |v|_t > (n-1)(k-1) \geq (n-1) f_{s_2 s_1}(t)$ and thus $f(t) > 0$.

Because the underlying graph of G_{f_v} is connected and from $f_v(t) > 0$ it follows that $f(t) > 0$, thus the underlying graph of G_f must also be connected, except possibly for some isolated vertices. By using (2.5) we get,

$$\sum_{a \in \Sigma} f(as) = \sum_{a \in \Sigma} f(sa) + c_s$$

for all $s \in \Sigma^{k-1}$. So by Lemma 13, there is a word $u \in S(s_1, s_2, |u|)$ such that $f = f_u$. Then u^n begins with s_1 and ends with s_2 and

$$|u^n|_t = n|u|_t + (n-1)|s_2 s_1|_t = nf(t) + (n-1) f_{s_2 s_1}(t) = f_v(t) = |v|_t$$

for all $t \in \Sigma^k$, so u^n is k-abelian equivalent to v. □

Now we can use this lemma to prove the result of weakly k-abelian unavoidability. The idea of the proof is quite similar to the proof of abelian case in Theorem 50. Now the pair $(f_u \bmod n, \text{suf}_{k-1}(u))$. plays similar role than the Parikh vector played in the abelian case.

Theorem 55. *Let Σ be an alphabet and let $k, n \geq 2$. Every infinite word $w \in \Sigma^\omega$ contains a non-empty factor that is k-abelian equivalent to an nth power.*

Proof. For a prefix u of w, consider the pair $(f_u \bmod n, \text{suf}_{k-1}(u))$. The number of different pairs is finite, so w has infinitely many prefixes $u_1, u_2, \ldots$ such that their pairs are the same. Let i be such that no factor of length k appearing only finitely many times in w appears after u_i. Let $j > i$ be such that if $u_j = u_i v$, then each other factor of length k appears at least $(n-1)(k-1)$ times in v. Then for all $t \in \Sigma^k$

$$\begin{aligned} &|v|_t + |\text{suf}_{k-1}(v)\text{pref}_{k-1}(v)|_t = |\text{suf}_{k-1}(v)v|_t = |\text{suf}_{k-1}(u_i)v|_t \\ =&|u_i v|_t - |u_i|_t = f_{u_j}(t) - f_{u_i}(t) \equiv 0 \pmod{n}. \end{aligned}$$

Thus v satisfies the conditions of Lemma 54 and v is k-abelian equivalent to an nth power. $\square$

Some further questions that might be asked on weakly k-abelian powers are:

- How many k-abelian equivalence classes of words of length l contain an nth power?
- How many words there are in those equivalence classes, i.e. how many words of length l are weakly k-abelian nth powers?
- What is the length of the longest word avoiding weakly k-abelian nth powers?
- How many words avoid weakly k-abelian nth powers?

The answers depend on k, n, l and the size of the alphabet. We will make a few remarks about those questions.

First, it is easy to prove that two squares uu and vv are k-abelian equivalent if and only if u and v are. Thus the number of k-abelian equivalence classes of words of length $2l$ containing a square is the number of k-abelian equivalence classes of words of length l. The estimation of this number has been already presented in Theorem 12 and the number is polynomial with respect to l.

Second, some of the equivalence classes contain exponentially many words. For example, a word over the alphabet $\{a, b\}$ is 2-abelian equivalent to $(a^m(ab)^m)^2$ if and only if it begins with a, ends with b, contains $4m$ a's and $2m$ b's but does not contain consecutive b's, see Lemma 7. The number of such words is 3, 35, 462, 6435, 92378, ..., $\binom{4m-1}{2m}$ with respect to $m = 1, 2, 3, \dots$ by Example 14.

As an example we will consider binary words of length 12 , i.e., the words $\{a, b\}^{12}$ and count how many differently defined squares there exist. This also illustrates the behaviour of these concepts. In the set of 4096 words there are

- 64 squares,
- 168 2-abelian squares,
- 924 abelian squares,
- 1024 weakly 2-abelian squares,
- 2048 weakly abelian squares.

Those 1024 weakly 2-abelian squares belong to 32 different equivalence classes and weakly abelian squares belong to 7 different equivalence classes. Representatives for each of these seven classes over a binary alphabet are as follows: $a^{12}, a^{10}b^2, a^8b^4, a^6b^6, a^4b^8, a^2b^{10}, b^{12}$.

2.4 Overview of other results on k-abelian equivalence

In this last section of this chapter we present a selection of other results related to k-abelian equivalence. We will not go through the proofs or other details because results of this section are mostly based on works of other researchers and the purpose is just to give an overview of the general study. First we will point out a general remark we made with Karhumäki in [48] based on his earlier results.

With *Parikh properties* we mean abelian properties of words, that is, properties that are dependent on number of each letter in the word. The *k-generalized Parikh properties* then refer to properties that are dependent on numbers of occurrences of factors of length k. As stated in [54] these are closely related in problems defined by morphisms. Problems on 1-free morphisms and k-generalized Parikh properties can be reduced to problems on 1-free morphisms and usual Parikh properties over a bigger alphabet. A 1-free morphism means a morphism which maps each letter to a word at least two letters. The action to check the factors of length k is denoted by a mapping $\bigwedge_k : \Sigma^* \to \hat{\Sigma}^*$ and we use $\bigwedge_k(x) = \hat{x}$, for convenience. Details of the notions and the proof of the following lemma are given in [54].

Lemma 56 ([54]). *Let $h : \Sigma^* \to \Sigma^*$ be a 1-free morphism and $k \geq 1$. Then there exists a morphism $\hat{h} : \hat{\Sigma}^* \to \hat{\Sigma}^*$ such that $\bigwedge_k h = \hat{h} \bigwedge_k$, i.e., the following diagram holds true for all $x \in \Sigma^*$*

$$\begin{tikzcd}
x \arrow[r, "\wedge_k"] \arrow[d, "h"'] & \hat{x} \arrow[d, "\hat{h}"] \\
h(x) \arrow[r, "\wedge_k"] & \widehat{h(x)} = \hat{h}(\hat{x}).
\end{tikzcd}$$

From Lemma 56 we can conclude, for instance, that the avoidability questions on k-abelian repetitions for a morphism h can be reduced to problems on abelian avoidability for a morphism $\hat{h}$, although in bigger alphabets.

Other things we will present here are related to periodicity, Sturmian words and Fine and Wilf's theorem. First we define *a k-abelian complexity* $\mathcal{P}_w^{(k)}(n)$ of w, which counts the number of k-abelian equivalence classes of factors of w of length n.

$$\mathcal{P}_w^{(k)}(n) = Card(F_n(w)/\equiv_k).$$

There exist many other complexity functions, for example, *the factor complexity* of w is $\rho_w(n) = Card(F_n(w))$ and *the abelian complexity* of w is the function that counts the number of pairwise non abelian equivalent factors of w of length n, i.e. the abelian complexity is the same as 1-abelian complexity, see e.g. [88]. Most of the complexity functions can be used to reveal that a word is ultimately periodic. On the other hand, usually Sturmian words have the lowest possible complexity among the aperiodic words. In [59] it is shown that k-abelian complexity function does not make a difference. They show in [59], for example, the following:

Theorem 57 ([59]). *Let $w \in \Sigma^\omega$, $k \in \mathbb{Z}^+ \cup \{+\infty\}$ and $q^{(k)} : \mathbb{N} \to \mathbb{N}$ a function*

$$q^{(k)}(n) = \begin{cases} n+1 \text{ for } n \leq 2k-1 \\ 2k \text{ for } n \geq 2k. \end{cases}$$

If $\mathcal{P}_w^{(k)}(n_0) < q^{(k)}(n_0)$ for some $n_0 \geq 1$ then w is ultimately periodic.

Theorem 58 ([59]). *Fix $k \in \mathbb{Z}^+ \cup \{+\infty\}$. Let $w \in \Sigma^\omega$ be an aperiodic word and let $q^{(k)} : \mathbb{N} \to \mathbb{N}$ be the same function as in Theorem 57. The following conditions are equivalent:*

- *w is a balanced binary word, i.e., w is Sturmian word.*
- *$\mathcal{P}_w^{(k)}(n) = q^{(k)}(n)$.*

In another paper [58] from the same authors they have studied, in contrast, the differences between the results obtained by using factor complexity, abelian complexity or k-abelian complexity. They also study the asymptotic lower and upper complexities and show that they may differ significantly from each other.

One important question is when local properties imply global properties. This is studied in context of local squares and global periodicity in paper [51]. There are given minimal values of n for which there exist aperiodic infinite words which contain a left (or right or centered)square (or 2-abelian square or abelian square) of length at most n everywhere. Here a word w contains everywhere, for example, a left square of length at most n, if every factor of w of length $2n$ has a non-empty square as a suffix. Another conclusion concerning the study from local to global is the result from Fine and Wilf [36]. This periodicity result tells how long a word can be having periods p and q without having a period of common greatest divisor of p and q. In [57] it is shown that if k-abelian periods are considered instead of usual word periods then there does not exist corresponding bounds, in general. On the other hand, the cases when such bounds exist can be characterized and some non-trivial upper bounds and lower bounds can be given.

Let $k \geq 1$ and let p and q be such that $p, q \geq 2$ and $\gcd(p, q) = d < \min\{p, q\}$. The *initial k-abelian periods* of a word w are defined as follows: If there are k-abelian equivalent words $u_1, \ldots, u_{n+1}$ of length p such that $w = u_1 \cdots u_n \mathrm{pref}_{|w|-np}(u_{n+1})$, then w has an initial k-abelian period p. Other k-abelian periods can be defined by shifting the initial point of periods. Let $L_k(p, q)$ denote the length of the longest word that has initial k-abelian periods p and q but does not have initial k-abelian period d. In [57] it is shown, for instance, that:

Theorem 59 ([57])**.** *Let $p, q > gcd(p, q) = d > k$. There exists an infinite word that has initial k-abelian periods p and q but that does not have k-abelian period d, i.e., $L_k(p, q) = \infty$.*

Theorem 60 ([57])**.** *Let $p, q > gcd(p, q) = d$ and $d \leq k$. If w has initial k-abelian periods p and q and $|w| \geq lcm(p, q)$, then w has period d.*

2.5 Conclusions and perspectives

We have defined a set of equivalence relations for words which build a bridge between usual equality and abelian equality. One of the important and traditional questions in combinatorics on words is the avoidability question. In this chapter we have studied this problem in context of k-abelian equivalence. We have shown that 2-abelian squares cannot be avoided over a ternary alphabet and in addition, k-abelian squares cannot be avoided over ternary pure morphic words for any $k \geq 1$. On the other hand, we have obtained computational evidences that 3-abelian squares could be avoided over ternary alphabets and 2-abelian cubes already over binary alphabets. These questions have interested also other researchers and lately Rao has managed to prove those conjectures to be true in a manuscript [87].

We have also studied a question of avoidability by defining the notion of square in a new way and obtained two unavoidability results. Other general or traditional questions on combinatorics on words can be also examined with respect to k-abelian equivalences. We have introduced several results of this kind in the end of this chapter but we will also come back to k-abelian equivalences in the next chapters with such notions as the defect property and equations. Thus there exist many questions that can be asked in terms of k-abelian equivalences and even for avoidability there exist many open problems, see e.g. [87].

Chapter 3

Defect effect for two-dimensional words and trees

In this chapter we study the well-known property of words, the so called defect property, see for example [68, 69]. The defect property states that *if a set of $n+1$ words satisfies a non-trivial equation then these words can be expressed by concatenating at most n words.* The validity of the defect property has been studied also for other sets. For example, in [41, 55, 73, 79] the defect property has been analyzed in the contexts of trees and figures. These are also in the interest of this chapter. The defect property holds for trees but does not hold for figures, in general. After the papers [41] and [79] there remained two specific open problems. We will give counterexamples of sets of figures that do not have the defect property for these two problems and thus complete this survey. The content of this chapter is modified from the paper [46]. Before we go into the main topic, i.e. figures, we give an example showing that the defect property falls down with the concept of the previous chapter, i.e. k-abelian equivalence.

Example 61. Let $S = \{a^{k-1}b, a^kb\}$ be a set of words representing different k-abelian equivalence classes. Clearly $a^{k-1}b.a^kb \equiv_k a^kb.a^{k-1}b$ so S satisfies a non-trivial equation. If the defect property held then the words of S or the k-abelian equivalence classes they represent should be possible to be expressed by a single word. This single word should contain both letters a and b and be a factor of $a^{k-1}b$. So $a^{k-1}b$ is the only possibility for the word. Now a^kb cannot be expressed by that word and thus the set S cannot be expressed by one word and the defect property does not hold.

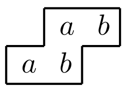
3.1 Definitions

In this section we define essential notions for this chapter and present the older results known about the subject. Here *figure* means *a labelled polyomino*, also called *a brick* or *a 2-dimensional word.* A figure is a partial mapping $x : \mathbb{Z}^2 \to \Sigma$, where Σ is a finite alphabet like for words. The domain of x is a finite and connected subset of $\mathbb{Z}^2$. Every element of the domain, i.e. unit square, corresponds to a labelled *cell* of the figure. The position of the figure with respect to the point (0,0) is not significant. In fact, only the shape and labels of the figure are determining.

Example 62. Consider a figure $x : \{(0,0),(1,0),(1,1)(2,1)\} \to \{a,b\}$ defined by

$$x : \begin{cases} (0,0) \mapsto a \\ (1,0) \mapsto b \\ (1,1) \mapsto a \\ (2,1) \mapsto b \end{cases} .$$

An illustration of the figure is the following:



The set of all figures over Σ is denoted by $\Sigma^\#$. If $X \subseteq \Sigma^\#$ is a set of figures then the set of all the figures tilable with the elements of X is denoted by $X^\#$. In the tilings of the figures of the set $X^\#$ we do not allow rotations of the original figures of X. A set of figures $X \subseteq \Sigma^\#$ is *a code* if every element of $X^\#$ admits exactly one tiling with the elements of X.

Because the shape of the domain qualifies figures so they can be divided into different classes according to the shape. For example, the concepts *dominoes*, *squares* and *rectangles* refer to the classes of figures with the respective shape of domain. With a domino we mean a figure which has a domain of $1 \times n$ or a $n \times 1$ rectangle with $n \geq 1$. The defect theorem for figures is usually examined over one class of figures at a time. The defect property for figures of a class $\mathcal{C} \subseteq \mathbb{Z}^2$ can be expressed as follows:

Formulation 63. *Let $X \subseteq \Sigma^\#$ be such a finite non-code that the domains of all $x \in X$ belong to $\mathcal{C}$. Then there exists such a code $Y \subseteq \Sigma^\#$, with the domains of all $y \in Y$ belonging to $\mathcal{C}$, that $X \subseteq Y^\#$ and $|Y| < |X|$, i.e., the size of the set Y is smaller than the size of the set X.*

As mentioned earlier the defect property does not hold for the figures in general. In addition, it does not hold even if we restrict to examine the sets of figures with an arbitrary cardinality within a specific class $\mathcal{C}$ corresponding to dominoes, squares or rectangles. So, in general, the answer to the question, whether there exists the defect property for figures as stated in

Formulation 63, would be negative. On the other hand, in [79] it was shown that the sets of two figures possess the defect property within these three classes. For dominoes the property also holds for the sets of size three as shown in [79]. For example, in [41, 79] there were given counterexamples to indicate that the defect property fails with figures with unrestricted shape and with squares already for the sets of size three. For dominoes and rectangles it is enough to consider the sets of size four to detect the fail of the defect property. We remark that if the defect property fails within a certain class of figures at the certain size of a set it also fails for bigger sets. So after these studies by Harju and Karhumäki and Moczurad two questions remained open: whether the defect property holds or does not hold for three rectangles and for two figures with unrestricted shape.

In the next section we give the examples which show that the answer in both cases is that the property fails. Thus, the validity of the defect property in the classes corresponding to dominoes, squares, rectangles and figures with unrestricted shape is now determined for each size of the set.

3.2 Analysis of the defect property for figures

In this section we give two counterexamples for the defect property over the sets of three rectangles and for the two figures of unrestricted shape. These examples show that there exist non-codes composed of figures of the given shape but these words cannot be tiled with fewer figures of the same class. These two cases make the analysis of the defect property of the figures of unrestricted shape, rectangles, squares and dominoes complete.

Example 64. A non-code consisting of the three rectangles which are shown in Figure 3.1 cannot be tiled with two rectangles. A figure that has two different tilings over the set of three considered rectangles is shown in Figure 3.2.

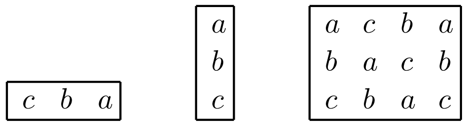


Figure 3.1: The set of three rectangles not possessing the defect property.

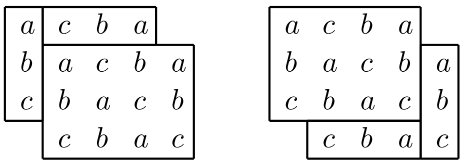


Figure 3.2: There is a figure having two different tilings.

To show that the set does not have the defect property it has to be shown that there are not any two rectangles that would tile the three original ones. We start by examining the two dominoes at first. In the set of two figures tiling all the original rectangles there have to be both original dominoes or otherwise the size of the tiling set becomes larger than two. With these two dominoes it is not possible to tile the third rectangle which is demonstrated in Figure 3.3. The four lines in the figure denote all the cells that may be covered with the dominoes and at least two cells remain uncovered. This means that the defect property fails over the sets of rectangles if the size of the set is three or greater.

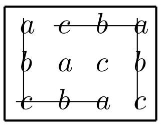


Figure 3.3: The third rectangle cannot be tiled with the other rectangles of the set.

Although, it is not required that the composition of the rectangles should also be a rectangle we can construct a rectangle having two different tilings as shown in Figure 3.4.

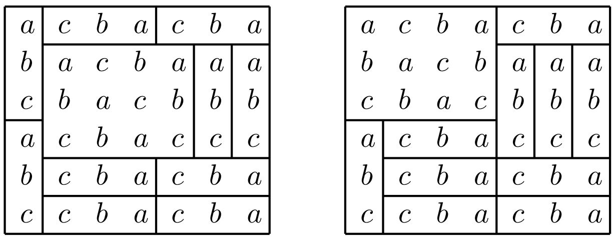


Figure 3.4: The figure with two tilings can be completed to a rectangle.

Example 65. A non-code consisting of the two figures of an unrestricted shape shown in Figure 3.5 cannot be tiled with one figure. In Figure 3.6 there is shown a figure having two different tilings over the considered figures.

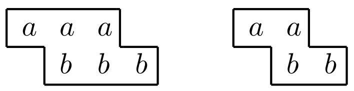


Figure 3.5: The set of two figures not possessing the defect property.

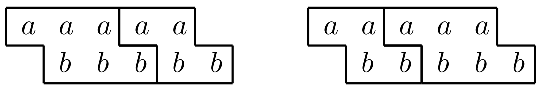


Figure 3.6: There is a figure having two different tilings.

To show that the set does not have the defect property we have to show that there are not any figure that would tile the two original ones. We consider the figure with four cells at first. This figure cannot be tiled with any smaller factor hence the figure itself should be the figure that would also tile the other one. The figure with six cells cannot be tiled with the figure of four cells so these two figures of unrestricted shape cannot be tiled with only one figure. This shows that the defect property already fails for two figures if the shape of the figures is not restricted.

We can summarize the analysis of the defect property of the figures of unrestricted shape, rectangles, squares and dominoes with the following Theorem 66. In this theorem we give a table representing the validity of the defect property which is a completed version of the corresponding table in [79]. It is marked on the table whether the defect property holds for (H) or fails over (F) the specific shape and the size of the set.

Theorem 66. *The complete analysis of the defect property of the figures of unrestricted shape, rectangles, squares and dominoes gives the following results:*

Shape $\diagdown$ *Size*	2	3	≥ 4
Dominoes	H	H	F
Squares	H	F	F
Rectangles	H	F	F
Unrestricted	F	F	F

The last row of the table in Theorem 66 indicates that the defect property does not hold for the figures at all if the shape of the figures is not restricted.

3.3 The defect effect and the prefix rank for a set of trees

Another example of two-dimensional objects for which survey of the defect property has been done is *a k-ary tree*, see e.g. [41, 55, 74, 73]. We remind that a tree is a certain type of graph and a few examples of trees can be seen in Figure 3.7. A finite k-ary tree over an alphabet Σ can be defined as a partial mapping $\{1, 2, \ldots, k\}^* \to \Sigma$. *The domain* of a finite k-ary tree is a finite and prefix closed subset of $\{1, 2, \ldots, k\}^*$ and the elements of the domain are called *nodes*. In this section we give a short overview of the validity of the defect property with trees as we did in the previous section with figures. The basic results for the defect property for trees are already studied in earlier papers, for example, in [55, 74] but we will point out a few special properties.

In combinatorics on words *the hull* of a set of words X means the smallest semigroup that satisfies some special properties and contains the set X. The size of the minimal generating set of this hull is called *the rank.* Now by setting different conditions that the semigroup has to satisfy we can define different hulls and ranks, respectively. For example, *free rank, prefix rank* and *suffix rank* are common and much studied ranks. For example, the prefix hull corresponds to the hull of which minimal generating set is *a prefix set.* A prefix set of words means such a set that it does not contain a word which would be a prefix of another word of the set. For more details see for example [20].

To define *a prefix set for trees* we need a notion of *compatibility,* which means that every pair of trees agrees on the intersection of their domains. A set of trees is a prefix set if the trees of the set are pairwise *noncompatible.* In this section we concentrate on the notions of prefix hull and prefix rank for tree sets. For more about different notions of hulls and ranks that are extended to trees and the exact definitions are given, for example, in [55].

In general, the defect theorem holds for the sets of trees as shown, for example, in [74]. The defect property holds when stated by using the free rank or the suffix rank for trees as it holds for words, too, see [55]. However, because of the differences between the definitions of the prefix set of trees and the prefix set of words, the defect theorem for trees cannot be expressed with a prefix rank. As stated in [55] there is a counterexample showing that this property fails. In the example there were a few mistakes and we give a corrected version of the counterexample as Example 67. It shows that a tree set of size three has the prefix rank of four which means that the defect property in this content is not valid, in general. In the previous section we noticed that it was possible to define the boundaries for the size of the sets of figures for which the defect property fails. This boundary was dependent on the shape of the figures. Now we will examine the minimal size of the set for which there is a tree set with a prefix rank greater than the size of the set.

Example 67. A non-code consisting of the following three trees shown in Figure 3.7 has the prefix rank of four.

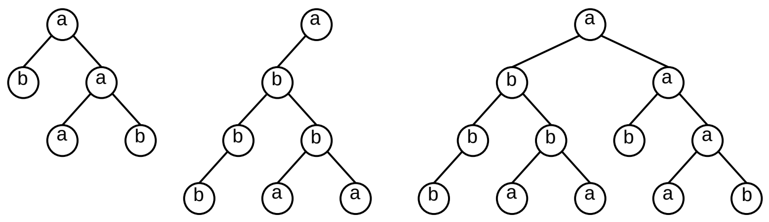


Figure 3.7: The third tree of the set can be factorized with two others.

The following four trees shown in Figure 3.8 form the minimal generating set of the prefix hull of the considered tree set indicating that the prefix rank is four.

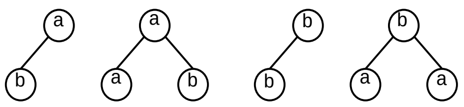


Figure 3.8: The prefix hull of the set consists of four trees.

In [55] it is shown that the free rank of a non-code tree set is always smaller than the size of the set. This means that the free hull of a non-code consisting of two trees can be generated by just one tree. This implies that the prefix hull of the set can be generated with the same tree, too. Thus we know that the defect property holds for tree sets of size two expressed with free rank as well as prefix rank. We remark that, likewise with figures, if the defect property fails at the certain size of a set, it also fails for bigger sets. Clearly, we can always increase the size of the set by adding a tree with a node labelled with a new letter. This can be done to the set for which the property already fails. We conclude that the defect property holds for the sets of two trees in the context of prefix rank and fails over bigger sets. So the result resembles the validity of the defect property for squares and rectangles.

3.4 Conclusions and perspectives

The defect theorem is often thought as folklore. It states that if a set of n words satisfy a non-trivial equation then these words can be expressed by $n-1$ words. The first known formulation of this property is by Skordev and Sendov in 1961, [90], and the first fundamental paper of the defect theorem is from 1979 by Berstel, Perrin, Perrot and Restivo, [9]. In 2004, 25 years later, Harju and Karhumäki presented a detailed analyzis of the different aspects of the defect theorem in [41]. In this chapter we have considered the defect property within the sets of so called 2-dimensional words. These are sort of generalizations of usual words seen as 1-dimensional words. The defect property holds just for small sets of these 2-dimensional words, also called figures. We have introduced two examples in which the defect theorem fails and these give the answers to two open problems that remained after the earlier studies of these basic cases for figures. Now we have that for figures of unrestricted shape the defect theorem does not hold at all and for figures that are rectangles the defect theorem holds for pairs but not for larger sets.

In [64] the concept of figures is discussed with additional properties enclosed to the figures. These *directed figures* have two special points, *a starting point* and *an ending point*. Two directed figures can be concatenated by joining the figures together so that the starting point of the latter is placed on the top of the ending point of the former. When concatenating directed figures we need *a merging function* which determines the labels of the cells that overlap in concatenation. In [64] it is shown that even in very simple cases the defect property fails in the sets of directed figures. On the other hand, if the set of directed figures is chosen so that the directed figures have a correspondence with words then the set clearly satisfies the defect property. An open problem is to find such restrictions for figures that they would satisfy the defect property but would not have a correspondence to words.

We have also considered here other 2-dimensional objects, namely trees. This time we add to the question of validity of the defect theorem the notion of a prefix rank. The result is that the defect property holds for the sets of two trees in the context of prefix rank and fails over bigger sets. So the behaviour is similar to the rectangular figures. The defect theorem can be studied in many different connections and here it is checked also for k-abelian equivalences. A simple example shows that the defect property for k-abelian equivalences does not hold at all.

Chapter 4

Word equations over partial semigroups

This chapter contains a discussion of word equations and the solutions of those. A simple example of a word equation is $x^2 = abab$ for which the only solution is that the unknown word x equals to the word ab. This equation has both unknowns and constants. Actually, the other side consists of unknowns and the other side of constants. Another, simple equation $xy = yx$ consists only of unknowns and this equation is one of the basic word equations. The solution for this can be easily characterized and it can be applied in solving many other equations. In fact, by the defect property, discussed in the previous chapter, we already know that there exists a word z that both of the words x and y can be represented by it, i.e., $x = z^i$ and $x = z^j$ for some $i, j \geq 0$. Although, word equations have been studied a lot there exist quite simple looking questions about equations and their solutions but the answers are still not known. In this thesis we concentrate on few basic equations but we define the product of words in an exceptional way. For more information of usual word equations, see for example [68].

The specific way to concatenate words we will use in this chapter is motivated by DNA computing. This operation of *overlapping product* or its extensions have been studied in a number of articles associated with bio-operations in DNA strands, see e.g., [83, 22, 13, 29, 30, 53, 75, 76]. The operation of overlapping product of words is locally controlled and a (partial) associative operation on the set of non-empty words Σ^+. The descriptional complexity of this operation has been analyzed in the case of regular languages in [44]. As mentioned we will consider this operation in connection with word equations. It turns out that many questions on equations can be transformed, and finally solved, by translating these to related problems on ordinary word equations. The translation is made because, for example, the simple operation of cancellation does not work for equations over these partial semigroups. Thus, for example, $x \bullet y = x \bullet z \bullet x$ is not equivalent to $y = z \bullet x$, as explained more closely in Section 4.2.

More concretely, we solve a few basic equations over overlapping product, introduce a general translation of such equations to a Boolean system of ordinary equations, and as a consequence establish, e.g., that the fundamental result of solvability of the satisfiability problem extends to these new types of equations. The content of this chapter is based on the paper [47].

4.1 Basic word equations

We have already mentioned one of the basic equations, that is the commutation equation. There also exist a few other equations that can be considered as elementary equations on combinatorics on words. These equations are elementary in a sense that they can be used in solving other equations and they define some simple properties the words may have. We will shortly present some of these equations and give the set of solutions for them. We will not give the proofs for the solutions because these are quite well-known and can be found, for example, in [68]. In addition, it is easy to verify that the given solutions satisfy the equations. In the next section we will compare these solutions to the solutions we get for the corresponding equations with overlapping products. The equations with overlapping products are considered in more details, too.

In what follows, let X be the set of unknowns and let Σ be the alphabet of constants and thus also the alphabet for the solutions. We will use Greek letters to denote the words over Σ to distinguish them from the unknowns in X and the letters in Σ. In the first chapter a solution of an equation $u = v$, where $u, v \in (X \cup \Sigma)^+$, was defined to be a morphism $e : (X \cup \Sigma)^* \to \Sigma^*$ that satisfies $e(u) = e(v)$ and $e(a) = a$ for all $a \in \Sigma$. Though, the solution is often given in the form $x = \alpha, y = \beta$ and $z = \gamma$ for the unknowns $x, y, z \in X$ of the equation and $\alpha, \beta, \gamma \in \Sigma^*$ meaning that the morphism maps x to α, y to β and z to γ.

Arguments and tools that are commonly used for solving word equations are *Levi's lemma*, *splitting the equations* and *the length argument.* The length argument means that it is possible to derive from the equation that some parts in the equation have to have, for example, the same length. For instance, it is clear that from the equation $xyxz = z^2xy$ it follows that $|x| = |z|$ and thus $x = z$. Splitting the equation means that it is possible to split the equation to a pair of equations because of, for example, the length argument or some other additional information. As an example, consider the equation $xyx^3y = uv^2u$. The middle point can be located on both sides of the equation. Thus we get a pair of equations $xyx = uv$ and $xxy = vu$. The third property that can be used to solve equations or to make them more simple is the following lemma, so called Levi's lemma, see [67]:

Lemma 68 ([67]). *Let $x, y, u, v \in \Sigma^+$ be words such that $xy = uv$. Then there exists a word $w \in \Sigma^*$ such that*

$$x = uw \text{ and } wy = v \text{ , or}$$

$$xw = u \text{ and } y = wv$$

.

Now we go to the basic word equations.

Equation 1. *Consider the equation $xy = yx$, for $x, y \in X$. This equation defining the commutative words has the solution $x = \alpha^i, y = \alpha^j$ for $\alpha \in \Sigma^*$ and $i, j \geq 0$.*

Equation 2. *Consider the equation $xz = zy$, for $x, y, z \in X$. This equation defines the conjugation, i.e., words x and y are conjugates if and only if they satisfy the given equation. In other words, the words x and y can be obtained from each other by cyclically shifting the letters and the word z corresponds to those shifted letters. The solution for this equation is $x = \alpha\beta, y = \beta\alpha, z = (\alpha\beta)^i\alpha^j$ for $\alpha, \beta \in \Sigma^*$ and $i, j \geq 0$.*

The third basic equation we consider asks when the product of two squares is a square, a problem first studied in [71]. The answer is that the equation has only periodic solutions.

Equation 3. *Consider the equation $x^2y^2 = z^2$, for $x, y, z \in X$. Clearly, the length of z equals to the length of xy and z begins with x and ends with y. Thus $z = xy$ which gives commutation $xy = yx$. The solution for the equation $x^2y^2 = z^2$ is $x = \alpha^i, y = \alpha^j, z = \alpha^{i+j}$ for $\alpha \in \Sigma^*$ and $i, j \geq 0$.*

4.2 Concatenation of words with overlap

We define first the *partial* binary operation, so-called *overlapping product*, on Σ^+ as follows: For two words ua and bv, with $a, b \in \Sigma$, we set

$$ua \bullet bv = \begin{cases} uav & \text{if } a = b \text{ ,} \\ \text{undefined} & \text{if } a \neq b \text{ .} \end{cases}$$

Clearly, the operation $\bullet$ is an associative partial operation so that $(\Sigma^+, \bullet)$ constitutes *a partial semigroup.* In addition, letters can be seen in $(\Sigma^+, \bullet)$ as partial (non-unique) left and right units. Indeed, $a \bullet u$ with any $u \in \Sigma^+$ is equal to u whenever the product is defined.

The product can be written without parenthesis due to the associativity:

$$\alpha = \alpha_1 \bullet \alpha_2 \bullet \cdots \bullet \alpha_n \ , \ \text{ for any } \alpha_i \in \Sigma^+. \tag{4.1}$$

If the word α, as an element of Σ^+, is defined as on (4.1) then for each $i = 1, \ldots, n-1$, necessarily

$$\text{last } \alpha_i = \text{first } \alpha_{i+1} \ .$$

Now α is deduced from (4.1) as follows

$$\begin{aligned} \alpha &= \alpha_1(\text{last } \alpha_1)^{-1}\alpha_2(\text{last } \alpha_2)^{-1} \cdots \alpha_{n-1}(\text{last } \alpha_{n-1})^{-1}\alpha_n \\ &= \alpha_1(\text{first } \alpha_2)^{-1}\alpha_2(\text{first } \alpha_3)^{-1} \cdots \alpha_{n-1}(\text{first } \alpha_n)^{-1}\alpha_n \ . \end{aligned}$$

On the other hand any word

$$\alpha = \alpha_1\alpha_2 \cdots \alpha_n \ \text{ with } \alpha_i \in \Sigma^+$$

can be written as an element of the partial semigroup $(\Sigma^+, \bullet)$ as follows:

$$\alpha = \alpha_1(\text{first } \alpha_2) \bullet \alpha_2(\text{first } \alpha_3) \bullet \cdots \bullet \alpha_{n-1}(\text{first } \alpha_n) \bullet \alpha_n \ .$$

It is worth noting that the latter translation is always defined.

Common tools for solving word equations such as Levi's Lemma, splitting of equation and length argument are not so straightforward to use with equations containing overlapping products. Problems for using these tools arise from the facts that for overlapping products to be defined the last and the first letters of the adjacent factors have to coincide and when a product is conducted these two letters are unified to a single letter. The next example shows one problem that may occur.

Example 69. Consider an equation $x \bullet y = x \bullet z \bullet x$ with overlapping products and an equation $xy = xzx$. Equation $xy = xzx$ can be reduced into the form $y = zx$. Accordingly we could suppose that $x \bullet y = x \bullet z \bullet x$ equals with equation $y = z \bullet x$. However, for example, $y = abb, z = ab, x = bb$ is a solution for $y = z \bullet x$ but not for the original equation because the overlapping product $x \bullet y = bb \bullet abb$ is not defined.

Example 69 shows that we cannot use Levi's Lemma straightforwardly to eliminate the leftmost or the rightmost unknowns. The same problem arises if we split an equation. Again we may loose the information of the requirements that originated from the overlapping product that was located at the point of splitting.

Unification of the last and the first letters of the adjacent factors complicates also the use of length argument. The total length of an expression containing overlapping products depends on the lengths of the factors and on the number of factors, i.e. $|x_1 \bullet \cdots \bullet x_k| = |x_1| + \cdots + |x_k| - (k-1)$. Although, for some equations it may be easy to detect, for example, the middle of both sides as for example in the equation $x \bullet y \bullet y \bullet x = z \bullet z$. From this we can conclude that $x \bullet y = z$, $y \bullet x = z$. The consequences of splitting the equation, that is last $y =$ first y, have to be taken into account, too.

Now we proceed by solving the same basic equations as we did in the previous section but here we consider them over the partial semigroup with overlapping product. First we consider the equation $x \bullet y = y \bullet x$, which corresponds to commutation and the Equation 1.

Example 70. To solve the equation $x \bullet y = y \bullet x$ we first assume that $|x|, |y| > 1$. For the overlapping product to be defined we can assume that $x = ax'a$ and $y = ay'a$, where $a \in \Sigma$ and x', y' are new unknowns of the set X'. Now we can reduce the equation $x \bullet y = y \bullet x$ into an ordinary word equation $x \bullet y = ax'a \bullet ay'a = ax'ay'a = ay'ax'a = ay'a \bullet ax'a = y \bullet x$. From the equation $ax'ay'a = ay'ax'a$ we can notice that $ax'ay' = ay'ax'$, and hence ax' and ay' commute. Now we can write $ax' = t^i$ and $ay' = t^j$, where $t = a\alpha$ with $\alpha \in \Sigma^*$ and $i, j > 0$. From this we get $x = ax'a = t^i a = (a\alpha)^i a$ and $y = ay'a = t^j a = (a\alpha)^j a$, where $a \in \Sigma, \alpha \in \Sigma^*$ and $i, j > 0$. In the case that $|x| = 1$ (resp. $|y| = 1$) we have $x = a$ (resp. $y = a$), with $a \in \Sigma$ and $y = a\alpha a$ or $y = a$ (resp. $x = a\alpha a$ or $x = a$), with $\alpha \in \Sigma^*$. Thus the equation $x \bullet y = y \bullet x$ has solutions

$$\begin{cases} x = (a\alpha)^i a \\ y = (a\alpha)^j a \end{cases}, \quad \text{where } a \in \Sigma, \alpha \in \Sigma^* \text{ and } i, j \geq 0.$$

We remark that the answer of the equation of the previous example could also be written with the help of the overlapping product. For example, if $x = (a\alpha)^2 a$, $y = (a\alpha)^3 a$ we could also write $x = (a\alpha a) \bullet (a\alpha a)$, $y = (a\alpha a) \bullet (a\alpha a) \bullet (a\alpha a)$. Thus, the words that are solutions of this equation referring to commutation are, in fact, overlapping products of words of the form $a\alpha a$ or letters as a special case.

The second equation we will examine is associated with conjugation, i.e. the Equation 2 $xz = zy$.

Example 71. We first check few special cases for the equation $x \bullet z = z \bullet y$. If $x = a$, with $a \in \Sigma$, then $y = b$, $b \in \Sigma$, and $z = a\alpha b$, $\alpha \in \Sigma^*$, or if $a = b$, then $z = a$ is possible, too. If $|x| = 2$ then $x = aa$, with $a \in \Sigma$, and then $y = bb$ and $z = a\alpha b$, where $\alpha \in \Sigma^*$ or if $a = b$, then $z = a^i$ for $i > 0$ is possible, too. In fact, if $a \neq b$ then $x = aa$, $y = bb$ and $z = a\alpha b$ would give an equation $aa\alpha b = a\alpha bb$, which does not have a solution. So if $|x| = 2$ then $x = y = aa$ and $z = a^i$ with $a \in \Sigma$ and $i > 0$.

If $|z| = 1$ and $|x|, |y| > 2$ then $z = a$ for some $a \in \Sigma$ and $x = y = a\alpha a$ for some $\alpha \in \Sigma^+$. If $|z| = 2$ and $|x|, |y| > 2$ then $z = aa$ or $z = ab$ for some

$a, b \in \Sigma$. If $z = aa$ then $x = y = a^i$, for some $i > 2$. If $z = ab$ then $x = ab\alpha a$ and $y = b\alpha ab$ for some $\alpha \in \Sigma^*$.

Now we can assume that $|x|, |y|, |z| > 2$ in equation $x \bullet z = z \bullet y$. As in Example 70 we may assume $x = ax'a$, $y = by'b$ and $z = az'b$, where $a, b \in \Sigma$ and $x', y', z' \in X'$. These assumptions are due to the facts that overlapping products have to be defined and x and z have a common first letter and y has a common last letter with z. Reduction gives now $x \bullet z = ax'az'b = az'by'b = z \bullet y$. From the word equation $x'az' = z'by'$ we can conclude that $x'a$ and by' conjugate. The conjugation property gives that there exist $p, q' \in \Sigma^*$ so that $x'a = pq'$, $by' = q'p$ and $z' = p(q'p)^i$, where $i \geq 0$. In addition, if $|q'| \geq 2$ then $q' = bqa$ with $q \in \Sigma^*$. Now with these assumptions we have a solution

$$\begin{cases} x = ax'a = apq' = apbqa \\ y = by'b = q'pb = bqapb \\ z = az'b = ap(q'p)^i b = ap(bqap)^i b \end{cases},$$

where $a, b \in \Sigma, p, q \in \Sigma^*$ and $i \geq 0$.
If $q' = a$ then $a = b$ and solutions are of the form

$$\begin{cases} x = ax'a = apa \\ y = ay'a = apa \\ z = az'a = (ap)^{i+1} a \end{cases},$$

where $a \in \Sigma, p \in \Sigma^+$ and $i \geq 0$.
Notice that we can incude the special solution in which $|z| = 1$ into this formula by changing $z = (ap)^i a$. Now the solution of the special case in which $|x| = 2$ can also be included in this formula by allowing $p = \epsilon$.

We have one case left. If $q' = \epsilon$ then $p = bp'a$, where $p' \in \Sigma^*$ and solutions are of the form

$$\begin{cases} x = ax'a = ap = abp'a \\ y = by'b = pb = bp'ab \\ z = az'b = a(p)^{i+1} b = a(bp'a)^{i+1} b \end{cases},$$

where $a, b \in \Sigma, p, p' \in \Sigma^*$ and $i \geq 0$.
In fact, these last solutions and the solutions for the case $|z| = 2$ and $|x| = |y| > 2$ are included in the the first formula of the three formulas above. Thus equation $x \bullet z = z \bullet y$ has solutions

$$\begin{cases} x = apbqa \\ y = bqapb \\ z = ap(bqap)^i b \end{cases}, \text{ where } a, b \in \Sigma, p, q \in \Sigma^* \text{ and } i \geq 0$$

and the special solutions given below, where $a, b \in \Sigma, \alpha \in \Sigma^*$, $i \geq 0$:

$$\begin{cases} x = a \\ y = b \\ z = a\alpha b \end{cases}, \begin{cases} x = a \\ y = a \\ z = aa^i \end{cases} \text{ and } \begin{cases} x = a\alpha a \\ y = a\alpha a \\ z = (a\alpha)^i a \end{cases}.$$

The third basic equation discusses squares. In the case of usual word equations the answer for Equation 3 was that the equation has only periodic solutions. If we consider the equation with overlapping product we get a corresponding result.

Example 72. We first assume that $|x|, |y|, |z| > 1$ in the equation $x \bullet x \bullet y \bullet y = z \bullet z$. Because overlapping products have to be defined we can again assume that $x = ax'a$, $y = ay'a$ and $z = az'a$, where $a \in \Sigma$ and $x', y', z' \in X'$. Reduction of overlapping products into usual word products gives an equation $ax'ax'ay'ay'a = az'az'a$ from which we get a more simple equation $(ax')^2(ay')^2 = (az')^2$. From this we can conclude that $ax' = t^i$ and $ay' = t^j$ and $az' = t^{i+j}$ with $t = a\alpha$, $\alpha \in \Sigma^*$ and $i, j > 0$ and hence $x = ax'a = (a\alpha)^i a$, $y = ay'a = (a\alpha)^j a$ and $z = az'a = (a\alpha)^{i+j}a$.

Again if some of the unknowns equal to a letter, then the solution is gained from the following general formula by allowing $i, j \geq 0$. The equation $x \bullet x \bullet y \bullet y = z \bullet z$ has solutions

$$\begin{cases} x = (a\alpha)^i a \\ y = (a\alpha)^j a \\ z = (a\alpha)^{i+j} a \end{cases}, \text{ where } a \in \Sigma, \alpha \in \Sigma^* \text{ and } i, j \geq 0.$$

We yet give one example of a basic equation which leads us to analyze the defect property once again.

Example 73. To solve an equation $x \bullet y = u \bullet v$ we may assume $x = x'a$, $y = ay'$, $u = u'b$ and $v = bv'$ where $a, b \in \Sigma$ and $x', y', u', v' \in X'$. With these assumptions we have an ordinary word equation $x'ay' = u'bv'$. We consider only the case $|x'| < |u'|$, the case $|u'| < |x'|$ being symmetric and the case $|x'| = |u'|$ being clear. The equation $x'ay' = u'bv'$ has now a solution $x' = \alpha$, $y' = \beta b\gamma$, $u' = \alpha a\beta$ and $v' = \gamma$ where $\alpha, \beta, \gamma \in \Sigma^*$. The solution for the original equation with the assumption $|x| < |u|$ can now be given:

$$\begin{cases} x = \alpha a \\ y = a\beta b\gamma \\ u = \alpha a\beta b \\ v = b\gamma \end{cases}, \text{ where } a, b \in \Sigma, \alpha, \beta, \gamma \in \Sigma^*.$$

We remark that these four words x, y, u and v of the previous example can be expressed in the form $x = \alpha a$, $y = a\beta b \bullet b\gamma$, $u = \alpha a \bullet a\beta b$ and $v = b\gamma$, thus they can be formed from three words by overlapping product. This implies, as stated in Theorem 75 that defect property is also valid in $(\Sigma^+, \bullet)$. Before the theorem we give an example illustrating the behaviour of a set containing letters. It explains why the words that are letters are excluded from the defect theorem.

Example 74. Consider the set of words $\{a, b, ab\}$. Now the words satisfy a non-trivial equation with overlapping products $a \bullet ab = ab \bullet b$. But these three words cannot be obtained from any two words by concatenating those with overlapping products. Thus the sets that contain letters, i.e., partial left and right units, do not possibly have the defect effect.

Theorem 75. *Let S be a set of n words with $S \cap \Sigma = \emptyset$, i.e., each word in S has length at least 2. If S satisfies a non-trivial equation with overlapping products, then these words can be expressed with $n-1$ words by using overlapping products.*

Proof. Let $x_1 \bullet x_2 \bullet \cdots \bullet x_k = y_1 \bullet y_2 \bullet \cdots \bullet y_l$ be a non-trivial equation such that $x_i, y_j \in S$ for all $i = 1, \ldots, k$ and $j = 1, \ldots, l$. We may assume that $|x_1| < |y_1|$ and hence y_1 can be written in the form $y_1 = x_1 \bullet (\text{last } x_1)\, y_1'$, for some word y_1'. Thus, the words of the set S can be expressed with words $S_1 = (S - \{y_1\}) \cup \{(\text{last } x_1)\, y_1'\}$. The number of words in S_1 is clearly at most n and $S_1 \cap \Sigma = \emptyset$. Now the equation corresponding to the original equation can be reduced at least from the beginning with a factor x_1 and hence, the new (non-trivial) equation will be shorter in terms of the total length of an expression which is given by $|x_1 \bullet \cdots \bullet x_k| = |x_1| + \cdots + |x_k| - (k-1)$. We divide the analyzis into two cases.

Case 1. Inductively, with respect to the length of the non-trivial equation, we will proceed step by step into an equation $u = v_1 \bullet \cdots \bullet v_m$ with $u, v_1, \ldots v_m$ words from the processed set of at most n words. Now it is clear that the word u may be removed from the set and the original words can be expressed with $n-1$ words as claimed.

Case 2. If in some point of the procedure described above the equation will reduce into a trivial equation, the constructed set of words corresponding to that situation contains already at most $n-1$ words. This follows from the fact that the reduction from a non-trivial equation into a trivial equation is possible only if some factor replacing an old word already exists in the considered set of words. $\square$

As a conclusion, the above examples and Theorem 75 show that results for word equations over overlapping product are often similar, but not exactly the same, as in the case of ordinary word equations. Moreover, the proofs reduce to that of ordinary words. This reduction is the subject of the next section.

4.3 Reduction into word equations

In this section the reduction of equations over overlapping products to that of ordinary word equations is analyzed in general. The reduction leads to a Boolean combination of word equations, as we shall see in the next result.

Theorem 76. *Let Σ be a finite alphabet, X be the set of unknowns and $e : u = v$ be an equation over $\Sigma \cup X$ with overlapping products. Then the equation e can be reduced into a Boolean combination of ordinary word equations. The Boolean combination contains only disjunctions.*

Proof. Consider the equation $u = x_1 \bullet x_2 \bullet \cdots \bullet x_l = y_1 \bullet y_2 \bullet \cdots \bullet y_m = v$, where $x_i, y_j \in X$ for all $i = 1, \ldots, l$ and $j = 1, \ldots, m$.

Part 1. Assume that the solutions u_i for x_i and v_j for y_j have $|u_i|, |v_j| > 1$, for all $i = 1, \ldots, l$ and $j = 1, \ldots, m$, and hence we can mark the first and the last letters of the words and write

$$x_1 = a_1 x_1' a_2 \ , \ x_2 = a_2 x_2' a_3 \ , \ \ldots \ , \ x_l = a_l x_l' a_{l+1} \ ,$$

$$y_1 = b_1 y_1' b_2 \ , \ y_2 = b_2 y_2' b_3 \ , \ \ldots \ , \ y_m = b_m y_m' b_{m+1} \ ,$$

where $a_i, b_j \in \Sigma$ and x_i' and y_j' are new unknowns from the set X'.

Now we have some restrictions for choosing the letters a_i, b_j. If $x_i = x_j$ then $a_i = a_j$ and $a_{i+1} = a_{j+1}$, and similarly if $y_i = y_j$, then $b_i = b_j$ and $b_{i+1} = b_{j+1}$. Comparing unknowns of the equation e on both sides we have that if $x_i = y_j$, then $a_i = b_j$ and $a_{i+1} = b_{j+1}$, and in addition, $a_1 = b_1$ and $a_{l+1} = b_{m+1}$ always hold.

With these assumptions and markings we have a reduced word equation $e' : u' = v'$ without overlapping products where u' and v' are defined as follows:

$$u = x_1 \bullet x_2 \bullet \cdots \bullet x_l = a_1 x_1' a_2 x_2' a_3 \cdots a_l x_l' a_{l+1} = u'$$

$$v = y_1 \bullet y_2 \bullet \cdots \bullet y_m = b_1 y_1' b_2 y_2' b_3 \cdots b_m y_m' b_{m+1} = v' \ .$$

In fact, to solve the original equation e we have to solve the reduced equation e' with all possible combinations of values for letters a_i and b_j from the set Σ. In other words, the set of solutions of the original equation $u = v$ equals to the set of solutions of a Boolean set of equations which is a disjunction of equations without overlapping products.

Part 2. In Part 1 we assumed that each unknown corresponds to a word of length at least two. Now we assume that at least one of the unknowns corresponds to a letter. We proceed as in Part 1 but with a bit different markings. Let $x_i = a_{i,1} x_i' a_{i,2}$ or $x_i = a_{i,12}$, with $a_{i,1}, a_{i,2}, a_{i,12} \in \Sigma$, depending on the length of the solution corresponding to x_i. Because overlapping products have to be defined we have $a_{i,2} = a_{i+1,1}$ or $a_{i,2} = a_{i+1,12}$ and $a_{i,12} = a_{i+1,1}$ or $a_{i,12} = a_{i+1,12}$. We process similarly with y_j's and b's. As in Part 1, we have some apparent additional restrictions for letters a's and b's depending on equation e. With these assumptions and markings we can again form a corresponding reduced word equation $e' : u' = v'$ without overlapping products.

To solve the original equation with assumptions of Part 2 we have again a Boolean combination of word equations to solve. This set is a disjunction of equations of the form e' with all possible combinations such that at least one unknown corresponds to a letter and values of corresponding a's and b's vary in the set Σ.

Part 3. In Part 1 and Part 2 we have only discussed the cases of constant free equations. If some factors in the equation $u = x_1 \bullet x_2 \bullet \cdots \bullet x_l = y_1 \bullet y_2 \bullet \cdots \bullet y_m = v$ are constants we proceed as previously in Parts 1 and 2 but with the additional knowledge of constants. If, for example, x_i is a constant in e and we have marked $x_i = a_i x_i' a_{i+1}$ we treat a_i, a_{i+1} and x_i' in equation e' as constants, too.

As a conclusion we remark that the considered Boolean sets are finite and the set of solutions of the original equation e is the set of solutions of a disjunction of Boolean sets of Part 1 and Part 2, the observations of the third part taken into account if necessary. Equations in this combined Boolean set do not contain overlapping products, and this proves the claim. $\square$

We remark that regardless of equation e having constants or not the equations in the constructed Boolean set have constants. Constants appear because the given reduction takes into account the fact that overlapping products have to be defined. The property that the overlapping product is only partially defined also makes it difficult to convert equations to the other direction. As mentioned in Section 4.2 it is easy to write a word as the element of partial semigroup $(\Sigma^+, \bullet)$. But if we try to convert, for example, an equation $xy = z$ we cannot just write $x \bullet y = z$. Instead, the equation $x \bullet y' = z$ with requirements $y' = ay, x = x'a$, with $a \in \Sigma$, would correspond the original equation.

4.4 Consequences of the reduction

It is known that any Boolean combination of word equations can be transformed into a single equation, see [56, 20] or [12] as the original source. Another well known result concerning word equations is the satisfiability problem, that is decidability of whether a word equation has a solution or not. The satisfiability problem is shown to be decidable by Makanin [72], see also [84]. We will show that corresponding results are also valid for equations with overlapping products.

Theorem 77. *For any Boolean combination of equations with overlapping products we can construct a single equation without overlapping products such that the sets of solutions of the Boolean combination and the single equation are equal when restricted to unknowns of the original equations.*

Proof. The result of the previous section shows that an equation with overlapping products can be reduced into a Boolean combination of usual word equations. From this it follows that any Boolean combination of equations with overlapping products can be reduced into another Boolean combination of ordinary word equations. This, in turn, as stated in [56] can be transformed into a single equation without overlapping products. □

We remind that combining a conjunction of two word equations into a single equation does not require any extra unknowns but in a case of disjunction two additional unknowns are required in the construction given in [56], see also [20]. Thus, the single equation constructed from the Boolean combination of equations is likely to contain many more unknowns than the original equations because of the disjunctions derived from the reduction method.

We next slightly modify this old proof for the result of [56] concerning a disjunction of two equations. The new result shows that, in fact, two additional unknowns are enough to combine a disjunction of a finite set of equations into a single equation.

Theorem 78. *Let $e_1 : u_1 = v_1, \ldots, e_n : u_n = v_n$ be a finite set of equations. A disjunction of these equations, i.e. the property expressible by e_1 or e_2 or $\ldots$ or e_n, can be transformed into a single equation with only two additional unknowns.*

Proof. We may assume that the right hand sides of the equations are the same because the disjunctions of the equations of the following two sets S_1 and S_2 are equivalent:

$$S_1 : \begin{array}{rcl} u_1 & = & v_1 \\ u_2 & = & v_2 \\ & \vdots & \\ u_n & = & v_n \end{array} \quad \text{and} \quad S_2 : \begin{array}{rcl} u_1v_2v_3\cdots v_n & = & v_1v_2\cdots v_n \\ v_1u_2v_3\cdots v_n & = & v_1v_2\cdots v_n \\ & \vdots & \\ v_1v_2\cdots v_{n-1}u_n & = & v_1v_2\cdots v_n \ . \end{array}$$

Thus, we may assume that $v_1 = v_2 = \cdots = v_n = v$ holds for equations $e_1, \ldots, e_n$.

To complete the proof we will outline the necessary constructions, and the justifications can be deduced as in [56]. First we define a function $\langle\ \rangle$ by

$$\langle\alpha\rangle = \alpha a \alpha b \ , \ \text{ where } a, b \in \Sigma,\ a \neq b.$$

We will use the properties that for each α the shortest period of $\langle\alpha\rangle$ is longer than half of its length and $\langle\alpha\rangle$ is primitive. We remark that now $\langle\alpha\rangle$ can occur in $\langle\alpha\rangle^2$ only as a prefix and a suffix. Let us denote $u_1 \cdots u_n = u$. With these observations we may deduce that

$$u_1 = v \text{ or } u_2 = v \text{ or } \cdots \text{ or } u_n = v \ \Leftrightarrow\ \exists Z, Z' :\ X = ZYZ' \ ,$$

where

$$Y = \langle u \rangle^2 v \langle u \rangle v \langle u \rangle^2$$

and

$$X = \langle u \rangle^2 u_1 \langle u \rangle u_1 \langle u \rangle^2 u_2 \langle u \rangle u_2 \langle u \rangle^2 \cdots \langle u \rangle^2 u_n \langle u \rangle u_n \langle u \rangle^2 \quad .$$

The proof of the previous equivalence is based on the facts that the word $\langle u \rangle^2$ is a prefix and a suffix of Y and that it occurs in X in exactly $n+1$ places. We concentrate on the non-trivial part of the proof. Thus, if $X = ZYZ'$ holds there are essentially two possibilities for $v \langle u \rangle v$:

$$v \langle u \rangle v = u_i \langle u \rangle u_i, \text{ for some } i \qquad \text{or}$$
$$v \langle u \rangle v = u_i \langle u \rangle u_i \langle u \rangle^2 u_{i+1} \langle u \rangle u_{i+1} \langle u \rangle^2 \cdots u_{j-1} \langle u \rangle u_{j-1} \langle u \rangle^2 u_j \langle u \rangle u_j,$$

for some i and j with $i < j$.

In the first case $v = u_i$ as required. In the second case we can use the positions of factors $\langle u \rangle$ and $\langle u \rangle^2$ to conclude that this case is not possible, which completes the proof. We separate the analyzis into two cases depending on whether $v \langle u \rangle v$ equals to an expression containing an odd number of factors $\langle u \rangle^2$ or an even number of those. The following two examples illustrate the argumentation in each case. We leave it to the reader to apply corresponding arguments for the other values of i and j.

Let $w = u_1 \langle u \rangle u_1 \langle u \rangle^2 u_2 \langle u \rangle u_2$ and assume $v \langle u \rangle v = w$. Now the factor $\langle u \rangle$ in the middle of $v \langle u \rangle v$ has to overlap with the factor $\langle u \rangle^2$ of w, otherwise one of the v's would contain a factor $\langle u \rangle^2$. In a general case the overlapping concerns the centermost occurrence of factors $\langle u \rangle^2$. Now the factor preceding (or succeeding) the mentioned $\langle u \rangle$ has the length at least $2|u_1| + 2|\langle u \rangle|$ (or $2|u_2| + 2|\langle u \rangle|$). We may assume $|v| \geq 2|u_1| + 2|\langle u \rangle|$, the other case being similar. Now $|v \langle u \rangle v| \geq 4|u_1| + 5|\langle u \rangle| > |w|$ because $|\langle u \rangle| > 2|u_2|$. This gives a contradiction.

Let $w' = u_1 \langle u \rangle u_1 \langle u \rangle^2 u_2 \langle u \rangle u_2 \langle u \rangle^2 u_3 \langle u \rangle u_3$ and assume $v \langle u \rangle v = w'$. Now the factor $\langle u \rangle^2$ has to be located in the same place on both occurrences of v in the word $v \langle u \rangle v$. This gives $v = u_1 \langle u \rangle u_1 \langle u \rangle^2 u_3 \langle u \rangle u_3$ and thus $|v \langle u \rangle v| = 9|\langle u \rangle| + 4|u_1| + 4|u_3| > |w'|$ giving a contradiction.

□

With a *positive* Boolean combination we refer to a Boolean combination that does not contain any negations, e.g. a Boolean combination of equations without inequalities. Now we can show that the conversion of a finite positive Boolean combination of equations over overlapping products into a single ordinary word equation requires only two extra unknowns.

Theorem 79. *For any finite positive Boolean combination of equations with overlapping products we can construct a single ordinary word equation with* *two additional unknowns such that the sets of solutions of the Boolean combination and the single equation are equal for some choice of these additional unknowns.*

Proof. For each equation over overlapping products we have a corresponding finite disjunction of ordinary equations based on reduction of Theorem 76. Thus, any finite positive Boolean combination of equations with overlapping products can be transformed into a finite positive Boolean combination of ordinary word equations. We may write the constructed Boolean combination in a disjunctive normal form and replace each conjunction of equations by a single equation. Thus, we have formed a finite disjunction of word equations without any additional unknowns. By Theorem 78 we can transform this disjunction into a single equation with two additional unknowns which proves the claim. □

The compactness theorem for words says that each system of equations over Σ^+ and with a finite number of unknowns is equivalent to some of its finite subsystems, see [3], [38] and also [42]. We remark that the analogical result concerning equations with overlapping products is not an obvious consequence of the reduction whereas the satisfiability theorem is as analyzed a few lines later. In fact, we do not even know whether the compactness theorem holds in this context. If we use the reduction on an infinite system of equations with overlapping products in order to be able to use the compactness theorem of ordinary word equations, we will end up with an infinite number of finite systems of disjunctions connected with conjunctions. Although, a finite positive Boolean combination of equations over overlapping products can be reduced into a single ordinary word equation with only two additional unknowns, a corresponding reduction of an infinite positive Boolean combination would require an infinite number of unknowns. Thus, we cannot use the original compactness theorem because of the infinite number of unknowns and the question about validity of the compactness theorem for equations over overlapping products remains open.

The decidability result for equations with overlapping products is instead obtained easily.

Theorem 80. *The satisfiability problem for a finite positive Boolean combination of equations with overlapping products is decidable.*

Proof. Theorem 77 shows that an equation with overlapping products can be reduced into a single equation without overlapping products. With Makanin's algorithm we can decide whether this equation without overlapping products has solutions or not and the existence of solutions is not affected by the additional unknowns in a sense that they would restrict the existence. Thus, we can straightforwardly decide the existence of solutions of the original equation with overlapping products, too. □

4.5 Conclusions and perspectives

In this chapter we have considered a few basic equations with a new concept of overlapping product. In addition we have introduced a general reduction from a equation with overlapping products to a Boolean combination of usual word equations. By this reduction we can show that the satisfiability problem for a finite positive Boolean combination of equations with overlapping products is decidable. Whereas the validity of the compactness theorem for equations with overlapping products remains open.

This setting resembles another open problem discussed in [19]. The problem states that the isomorphism problem for finitely generated F-semigroups (i.e. subsemigroups of free semigroups) is decidable whereas it is an open problem whether the result can be extended to F-semigroups generated by rational sets. The isomorphism problem asks whether two F-semigroups are isomorphic. For a finitely generated F-semigroup we can form a finite F-presentation which is, in fact, essentially based on equations. The finiteness of this F-presentation is obtained by the compactness theorem. Now it is decidable whether two finitely generated F-semigroups are isomorphic, i.e. whether they have a common F-presentation. For the F-semigroups generated by rational sets we cannot use the same approach because we have now infinitely many unknowns and we cannot apply the compactness theorem. On the other hand, the freeness problem of F-semigroups is a special case of the isomorphism problem and it is decidable for both F-semigroups generated by finite sets and by rational sets, see [8]. In addition, a free monoid can be embedded into a multiplicative monoid of integer matrices and for those already the freeness problem is undecidable, see [63, 40]. So these isomorphism and freeness problems seem to lay on the interface of decidable problems and undecidable problems. The situation may be the same for the compactness theorem we were dealing with in this chapter.

As the last perspective of this section we again return to k-abelian equivalences. It is a natural question what we can say about solving equations over k-abelian equivalence classes. As an example, consider the commutation equation $xy = yx$. Now we can ask which k-abelian equivalence classes x and y satisfy $xy \equiv_k yx$. Let us consider just equivalence classes which contain words of at least $k-1$ letters. Clearly, $\mathrm{pref}_{k-1}(x) = \mathrm{pref}_{k-1}(y)$ and $\mathrm{suf}_{k-1}(x) = \mathrm{suf}_{k-1}(y)$ are now the only requirements that are needed. For shorter words the requirements are a mixture of usual commutation conditions and demands based on generalized Parikh properties. Let us consider the conjugation, too. What are the requirements for words of length at least $k-1$ to satisfy $xz \equiv_k zy$? It is clear that $\mathrm{pref}_{k-1}(x) = \mathrm{pref}_{k-1}(z)$ and $\mathrm{suf}_{k-1}(z) = \mathrm{suf}_{k-1}(y)$. It is easy to see that $x\mathrm{pref}_{k-1}(x)\mathrm{suf}_{k-1}(y)$ have to be k-abelian equivalent to $\mathrm{pref}_{k-1}(x)\mathrm{suf}_{k-1}(y)y$, too. Because in both cases, for overlapping products and for k-abelian equivalence classes, the prefixes and suffixes are significant, they also play an important role when solving equations. In fact, for solving equations with respect to k-abelian equivalence the main focus seems to be in prefixes and suffixes, and thus the results are of a little interest at this level.

Chapter 5

On palindromes and Sturmian words

As mentioned already in the first chapter, combinatorics on words has many connections to other areas of mathematics as well as other sciences. Palindrome is an example of a concept that plays an important role in various areas of mathematics including diophantine approximation and number theory (e.g. [1, 21]), discrete mathematics (e.g. [37, 11]), algebra (e.g. [61, 28]), biomathematics (e.g. [60]), geometric symmetry in translation surfaces associated with various dynamical systems including interval exchange transformations (e.g. [35]) and theoretical physics in the spectral properties of discrete Schrödinger operators defined on quasicrystals (e.g. [43]).

In this chapter we investigate the connection with palindromes and Sturmian words. This chapter is based on the paper [39]. The original purpose was to consider binary words that can be defined up to word isomorphism by a set of its palindromic factors. What we mean by defining words by palindromes is explained later. The main result we ended up was the characterization of infinite binary not ultimately periodic words whose factors can always be defined by three palindromes. We found that this characterization is based on Sturmian words. On the other hand Sturmian words are known to be a set of words that can be defined in many different ways. Remark that, for example, in Chapter 2 it was mentioned that even k-abelian complexity can be used to define Sturmian words.

5.1 Definitions and notations

To define what means that a word is generated palindromically we define a set $S(n) = \{(i,j) \mid 1 \leq i \leq j \leq n\}$ for each positive integer n. We remind that for a word $u = u_1u_2 \cdots u_n \in \mathbb{A}^n$ and $(i,j) \in \mathcal{S}(n)$, we denote the factor $u_iu_{i+1} \cdots u_j$ by $u[i,j]$. In case $i = j$, we write $u[i]$ instead of

$u[i,i]$. Let $\mathrm{Alph}(u) = \{u_i \mid 1 \leq i \leq n\}$ denote the subset of Σ consisting of all letters occurring in u. Remark that a word w is a *palindrome* if $w = w^R$, where w^R denotes the reversal (or mirror image) of the word. That is, if $w = a_1a_2 \cdots a_n \in \Sigma^n$ then $w^R = a_n \cdots a_2a_1$. Finally, let $\mathcal{P}$ denote the collection of all palindromes (over any alphabet).

Definition 81. *Fix $u \in \Sigma^n$ and $S \subseteq \mathcal{S}(n)$. We say that S palindromically generates u if the following three conditions are verified:*

- *$u[i,j] \in \mathcal{P}$ for all $(i,j) \in S$,*
- *for all $k \in \{1, 2, \ldots, n\}$, there exists $(i,j) \in S$ with $i \leq k \leq j$,*
- *for each non-empty set Σ' and word $v \in \Sigma'^n$, if $v[i,j] \in \mathcal{P}$ for all $(i,j) \in S$ then there exists a mapping $c\colon \mathrm{Alph}(u) \to \Sigma'$ which extends to a morphism $c\colon \mathrm{Alph}(u)^* \to \Sigma'^*$ of words such that $c(u) = v$.*

We call the elements of S *generators* (or *palindromic generators*). It follows from the definition that if a set $S \subseteq \mathcal{S}(n)$ palindromically generates two words $u \in \Sigma^n$, $v \in \Sigma'^n$ then u and v are word isomorphic, i.e., there is a bijection $\nu\colon \mathrm{Alph}(u) \to \mathrm{Alph}(v)$ which extends to a morphism of words such that $\nu(u) = v$. In particular, the last condition in Definition 81 means that $\mathrm{Alph}(u)$ has the largest cardinality such that the first two conditions are satisfied.

Example 82. For each letter $a \in \Sigma$, the singleton set $S = \{(1,1)\}$ palindromically generates a. The set $S = \{(1,2)\}$ palindromically generates the word a^2, while $S = \{(1,1),(2,2)\}$ palindromically generates the word ab. For $n \geq 3$, the sets $S = \{(1,n-1),(1,n)\}$ and $S = \{(1,n),(2,n)\}$ each palindromically generate a^n. Let $a, b, c \in \Sigma$ and $S_1 = \{(1,5),(3,8),(7,9)\}$ and $S_2 = \{(1,5),(2,9)\}$. Now S_1 and S_2 both generate the same word $abcbaabcb$.

Given a word $u \in \Sigma^+$, we let $\mu(u)$ denote the infimum of the cardinality of all sets $S \subseteq \mathcal{S}(|u|)$ which palindromically generate u, i.e.,

$$\mu(u) = \inf\{\#S \mid S \subseteq \mathcal{S}(|u|) \text{ palindromically generates } u\}.$$

As usual, we let $\inf(\emptyset) = +\infty$. For instance, it is easily checked that $\mu(a^n) = 1$ for $n = 1, 2$ and $\mu(a^n) = 2$ for $n \geq 3$. Also $\mu(u) < +\infty$ whenever $u \in \Sigma^+$ and $\#\Sigma = 2$, i.e., whenever u is a binary word. Indeed, it is readily verified that for each $u \in \{0,1\}^+$, the set

$$S_u = \{(i,j) \mid u[i,j] = ab^ka \ \text{ for } \{a,b\} = \{0,1\},\ k \geq 0\}.$$

palindromically generates u. Typically $\mu(u) < \#S_u$, e.g., if $u = 00101100$, then the set $S = \{(1,2),(2,4),(3,5),(4,7),(7,8)\}$ palindromically generates u. In this example $S_u = S \cup \{(5,6)\}$, but the palindrome (5,6) is not needed in the generating set. The set S is the smallest palindromic generating set whence $\mu(00101100) = 5$. In contrast, for the ternary word *abca* no subset S of $\mathcal{S}(4)$ palindromically generates *abca*. For this reason, we shall primarily restrict ourselves to binary words.

Given an infinite word $x \in \Sigma^\omega$, we are interested in the quantity

$$\psi(x) = \sup\{\mu(u) \mid u \text{ is a factor of } x\}.$$

This means that we try to find the minimal number of palindromes for which each factor of x can be generated.

Recall that a binary word $x \in \{0,1\}^\omega$ is called *Sturmian* if x contains exactly $n+1$ factors of each given length n. It is well known that each Sturmian word x is aperiodic and uniformly recurrent, i.e., each factor of x occurs in x with bounded gap. Sturmian words are also closed under reversal (i.e., if u is a factor of x then so is u^R) and balanced (for any two factors u and v of the same length, $||u|_a - |v|_a| \leq 1$ for each $a \in \{0,1\}$). In fact an infinite binary word $x \in \{0,1\}^\omega$ is Sturmian if and only if it is aperiodic and balanced. For more on Sturmian words we refer the reader to [7, 68].

A factor u of a Sturmian word x is called *right special* (resp. *left special*) if both $u0$ and $u1$ (resp. $0u$ and $1u$) are factors of x. Thus x contains exactly one right special (resp. left special) factor of each given length, and u is right special if and only if u^R is left special. A factor u of x is called bispecial if u is both right and left special. Thus, if u is a bispecial factor of x, then u is a palindrome. A binary word $y \in \{0,1\}^*$ is called a *central* word if and only if $y \in \mathcal{P}$ and $y0$ and $y1$ are both balanced.

We will use several times the notion of *a bordered word.* Given two nonempty words u and v, we say u is a *border* of v if u is both a proper prefix and a proper suffix of v. If v has no borders then it is said to be *unbordered.*

We will later state our main result by making use of the following morphisms: For each subset $A \subseteq \{0,1\}$ we denote by $d_A \colon \{0,1\}^* \to \{0,1\}^*$ the *doubling morphism* defined by the rule

$$d_A(a) = \begin{cases} aa & \text{if } a \in A \\ a & \text{if } a \notin A. \end{cases}$$

Definition 83. *A word $y \in \{0,1\}^\omega$ is called double Sturmian if y is a suffix of $d_A(x)$ for some Sturmian word x and $A \subseteq \{0,1\}$. In particular, taking $A = \emptyset$, it follows that every Sturmian word is double Sturmian.*

Clearly, if y is double Sturmian, then there exists a Sturmian word x, a subset $A \subseteq \{0,1\}$ and $a \in A$ such that $d_A(x) \in \{y, ay\}$.

For each pair $I = (i,j) \in \mathcal{S}(n)$, we define a function

$$\rho_I : \{i, i+1, \ldots, j\} \to \{i, i+1, \ldots, j\}$$

called a *reflection* by $\rho_I(k) = i + j - k$. The function ρ_I is an involution, i.e., it is a permutation that satisfies $\rho_I(\rho_I(k)) = k$ for all $i \leq k \leq j$. For $u \in \Sigma^n$ we have that $u[i, j] \in \mathcal{P}$ if and only if $u[k] = u[\rho_I(k)]$ for each $i \leq k \leq j$. If $J = (i', j')$ with $i \leq i' \leq j' \leq j$, we denote by $\rho_I(J)$ the reflected pair $(\rho_I(j'), \rho_I(i'))$.

Definition 84. *Suppose $S \subseteq \mathcal{S}(n)$ palindromically generates a word $u \in \Sigma^n$, and let $m \in \{1, 2, \ldots, n\}$. We say m is a leaf with respect to S if there exists at most one pair $I = (i, j) \in S$ for which $i \leq m \leq j$ and $\rho_I(m) \neq m$.*

Example 85. Consider a word *abbabab* which is palindromically generated by $S = \{(1, 4), (3, 7), (5, 7)\}$. Now 1, 2, 5 and 6 are leaves with respect to S. Note also that two of the leaves has label a and two of them has label b.

5.2 Preliminary results

In this section we give preliminary results that we will need in our main results. First we give a simplified proof for the result shown by Saari in [89]. It improves an earlier result of Ehrenfeucht and Silberger in[33].

Lemma 86. *Each aperiodic infinite word x contains an infinite number of Lyndon words. In particular x has arbitrarily long unbordered factors.*

Proof. Let $\preceq$ be a lexicographic ordering of words, and suppose to the contrary that x contains only finitely many Lyndon factors. We write $x = u_1 u_2 \cdots$ where for each $i \geq 2$ we have that u_i is the longest Lyndon word that is a prefix of the suffix $(u_1 \cdots u_{i-1})^{-1} x$. Then, for all i, we have $u_{i+1} \preceq u_i$ since otherwise $u_i u_{i+1}$ would be a Lyndon prefix longer than u_i. Thus there exists a positive integer j such that $x = u_1 \cdots u_{j-1} u_j^\omega$, contradicting that x is aperiodic. The last claim now follows since every Lyndon word is unbordered. Indeed, if $u = vuv$ is a Lyndon word with respect to the order $\preceq$, then $vvu \preceq vuv$ (since $vu \preceq uv$), and hence v is the empty word. $\square$

We will also need the following result.

Lemma 87. *Let u be an unbordered factor of a Sturmian word $x \in \{0, 1\}^\omega$. Then either $u \in \{0, 1\}$ or $u = ayb$, where $\{a, b\} = \{0, 1\}$, and y is a central word.*

Proof. If $u \notin \{0, 1\}$, then we can write $u = ayb$ with $\{a, b\} = \{0, 1\}$. We claim that y is a palindrome. In fact, suppose vc (resp. dv^R) is a prefix (resp. suffix) of y with $v \in \{0, 1\}^*$ and $\{c, d\} = \{0, 1\}$. Since u is balanced it follows that $c = b$ and $d = a$. Since $av^R b$ and avb are both factors of x, and Sturmian words are closed under reversal, it follows that v is a bispecial factor of x, and hence a palindrome. Thus avb is both a prefix and a suffix of u. Since u is unbordered, it follows that $u = avb$, and hence $y = v$. Thus y is central. □

The following lemma gives a reformulation of Definition 81:

Lemma 88. *Let $u \in \Sigma^n$ be such that all letters of Σ occur in u. Let $S \subseteq \mathcal{S}(n)$. The following conditions are equivalent:*

(i) S palindromically generates u.

(ii) for each $k \in \{1, 2, \ldots, n\}$, there exists an $(i, j) \in S$ such that $i \le k \le j$, and for each $1 \le i, j \le n$, we have $u[i] = u[j]$ if and only if

$$\text{there exists a finite sequence (or path) } (I_t)_{t=1}^r \in S^r \text{ such that } j = \rho_{I_r}\rho_{I_{r-1}} \cdots \rho_{I_1}(i). \qquad (\star)$$

Proof. We first define relation θ as follows. Let $i\theta j$ if and only if there exists an $I \in S$ such that $j = \rho_I(i)$. Denote by θ^* the reflexive and transitive closure of θ. The relation θ is symmetric by the definition of the mappings ρ_I and thus θ^* is an equivalence relation. Here $i\theta^*j$ if and only if $(\star)$ holds for some sequence $(I_t)_{t=1}^r$ of elements from S. Let then $v \in \Sigma'^n$ be any word such that the factor $v[i, j]$ is a palindrome for each $(i, j) \in S$. Now, $i\theta j$ implies $v[i] = v[j]$. Consequently, $i\theta^*j$ implies $u[i] = u[j]$ by transitivity. It follows that the cardinality of Σ' is at most the number of equivalence classes of θ^*.

If S palindromically generates the given word u, then, by the last condition of Definition 81, each equivalence class of θ^* corresponds to a different letter in Σ. Therefore $u[i] = u[j]$ if and only if $i\theta^*j$. This proves the claim from (i) to (ii).

Suppose then that S satisfies (ii). First, let $I = (i, j) \in S$, and let $i \le k \le j$. Denote $k' = \rho_I(i)$. By (ii), we have that $u[k] = u[k']$, and therefore $u[i, j] \in \mathcal{P}$. Hence the first condition of Definition 81 holds. The second condition is part of (ii). For the third condition, by the beginning of the proof, any word $v \in \Sigma'^n$ for which $v[i, j]$ is a palindrome for each $(i, j) \in S$, the cardinality of Σ' is at most the cardinality of Σ. By (ii), we have that $v[i] = v[j]$ implies $u[i] = u[j]$ which proves the claim. □

In the next lemma we show a certain heritage property for the number of palindromes that are needed to generate factors of a word.

Lemma 89. *Let $u \in \Sigma^+$. Then $\mu(v) \le \mu(u)$ for all factors v of u.*

Proof. The result is clear in case $\mu(u) = +\infty$. So suppose $S \subseteq \mathcal{S}(|u|)$ palindromically generates u and set $k = \#S$. It suffices to show that if

$u = ax = yb$, where $a, b \in \Sigma$, then $\max\{\mu(x), \mu(y)\} \leq k$. We prove only that $\mu(x) \leq k$ as the proof that $\mu(y) \leq k$ is completely symmetric.

Suppose $S = \{I_1, I_2, \ldots, I_k\}$ palindromically generates u and let $m \in \mathbb{N}$ be the largest integer such that $I = (1, m) \in S$. Let $D = \{r \in \{1, 2, \ldots, k\} \mid I_r = (1, q) \text{ with } q < m\}$. Let

$$S' = S \cup \{I'_r \mid r \in D\} \setminus \{I_r \mid r \in D\}$$

where for each $r \in D$ we set $I'_r = \rho_I(I_r) = (m - q + 1, m)$ (see Fig. 5.1).

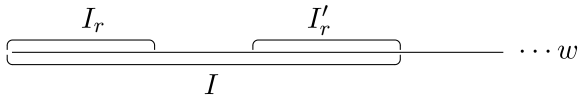


Figure 5.1: Reflecting the generators in S.

It follows that S' also palindromically generates u and I is the only generator in S' containing the initial position 1. Whence 1 is a leaf w.r.t. S', and hence

$$S'' = S' \cup \{(2, m-1)\} \setminus \{I\}$$

palindromically generates the suffix $x = a^{-1}u$. This proves the claim. □

Next we will prove an important lemma for doubling letters. We will show that if the word has a palindrome of odd length in a palindromic generating set and we double the letters corresponding to the letter in the middle of that odd palindrome, it is enough to have the same number of palindromes in the palindromic generating set of this new doubled word.

Lemma 90. *Suppose $u \in \Sigma^n$ is palindromically generated by a set $S \subseteq \mathcal{S}(n)$. Suppose further that there exist $p, q \in \mathbb{N}$ such that $(p, p + 2q) \in S$. Then for $A = \{u[p + q]\}$ we have $\mu(d_A(u)) \leq \#S$.*

Proof. Let $a = u[p + q]$, and write d_a for $d_{\{a\}}$. We define first a mapping $p_{d_a} \colon \{1, 2, \ldots, n\} \to \mathbb{N} \cup \{(j, j + 1) : j \in \mathbb{N}\}$ for the positions of $u \in \Sigma^n$. It maps every position of u to a corresponding position of $d_a(u)$ or to a pair of positions of $d_a(u)$ depending on whether the position of u has a label a that will be doubled or some other label. For convenience let us denote the prefix $u[1, i]$ of u by u_i and let u_0 be the empty word.

$$p_{d_a}(i) = \begin{cases} |d_a(u_i)| & \text{if } u[i] \neq a \\ (|d_a(u_i)| - 1, |d_a(u_i)|) & \text{if } u[i] = a. \end{cases}$$

Let $S' = \{(i', j') \mid (i, j) \in S\}$ where $i' = |d_a(u_{i-1})| + 1$ and $j' = |d_a(u_j)|$. In other words, we dilate each generator (i, j) by applying d_a to the corresponding factor $u[i, j]$. Clearly $\#S' = \#S$. We shall show that S' palindromically generates $d_a(u)$. We claim first that if a position i_1 of u is reflected to i_2 by a generator $(i_S, j_S) \in S$ then $p_{d_a}(i_1)$ is reflected to $p_{d_a}(i_2)$ by a generator $(i_{S'}, j_{S'}) \in S'$. Here $(i_{S'}, j_{S'})$ is the generator in S' corresponding to the generator (i_S, j_S) in S. Now $U = u[i_S, i_1 - 1] = u[i_2 + 1, j_S]^R$, and thus $|d_a(U)| = |d_a(U^R)| = |d_a(u[i_2 + 1, j_S])|$; see Figure 5.2. So if $u[i_1] \neq a$ then $p_{d_a}(i_1) - i_{S'} = j_{S'} - p_{d_a}(i_2)$ and otherwise $p_{d_a}(i_1) - (i_{S'}, i_{S'} + 1) = (j_{S'} - 1, j_{S'}) - p_{d_a}(i_2)$. Thus $p_{d_a}(i_1)$ is reflected to $p_{d_a}(i_2)$ as a single or as a pair of positions.

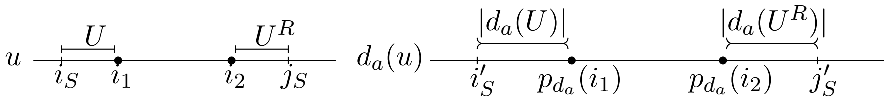


Figure 5.2: Reflection complies with doubling morphism d_a.

Let us denote, for any $x \in \Sigma$,

$$\Omega_{u,x} = \{i : 1 \leq i \leq n \text{ and } u[i] = x\},$$

i.e., $\Omega_{u,x}$ is the set of occurrences of x in u. By Lemma 88, for each $i, j \in \Omega_{u,x}$ there exists a sequence $i = i_1, i_2, \ldots, i_l = j$ of positions $i_m \in \Omega_{u,x}$ such that i_m is reflected to i_{m+1} by some generator in S. As we just shown, there also exists a sequence $i' = p_{d_a}(i_1), p_{d_a}(i_2), \ldots, p_{d_a}(i_l) = j'$ such that $p_{d_a}(i_m)$ is reflected to $p_{d_a}(i_{m+1})$ by some generator in S'. Let us define

$$\hat{\Omega}_{d_a(u),x} = \{p_{d_a}(i) : i \in \Omega_{u,x}\}.$$

In fact, if $x \neq a$ then $\hat{\Omega}_{d_a(u),x}$ is the same set as $\Omega_{d_a(u),x}$. The only problematic set is $\hat{\Omega}_{d_a(u),a} = \{(i', i' + 1) : 1 \leq i' < |d_a(u)| \text{ and for which } \exists i \in \Omega_{u,a} \text{ s.t. } p_{d_a}(i) = (i', i' + 1)\}$.

Now, consider the generator $(p', q') \in S'$ obtained from $(p, p + 2q) \in S$, i.e. $(p', q') = (|d_a(u_{p-1})| + 1, |d_a(u_{p+2q})|)$. The length of the palindrome determined by the generator (p', q') is even because $|d_a(u[p, p + q - 1])| = |d_a(u[p+q+1, p+2q])|$ and $|d_a(u[p+q])| = 2$. Hence a pair of two consecutive positions of $d_a(u)$, namely the positions of $p_{d_a}(p + q) \in \hat{\Omega}_{d_a(u),a}$, is now in the middle of this palindrome (p', q') thus these two positions reflect to each other and they have to have the same letter in word $d_a(u)$. Because of the pairwise reflections among the set $\hat{\Omega}_{d_a(u),a}$, the positions of u according to the pairs in this set contain the same letter a.

So we have that $\hat{\Omega}_{d_a(u),x} = \Omega_{d_a(u),x}$ for $x \neq a$. In the case $x = a$ the positions covered by the pairs in $\hat{\Omega}_{d_a(u),a}$ are exactly the positions in $\Omega_{d_a(u),a}$. This shows that there exists a path as described in Lemma 88 between each positions in $\Omega_{d_a(u),x}$ for any $x \in \Sigma$ thus S' palindromically generates the doubled word $d_a(u)$ and this ends the proof. □

Corollary 91. *Let $a \in \Sigma$ and $u \in \Sigma^n$. Then for $A = \{a\}$ we have*

$$\mu(d_A(u)) \leq \mu(u) + 1.$$

Proof. The result is clear if the symbol a does not occur in u since in this case $d_A(u) = u$. Thus we can assume that a occurs in u. Suppose $S \subseteq \mathcal{S}(|u|)$ palindromically generates u and $\mu(u) = \#S$. If S contains a generator that determines a palindrome of odd length whose center is equal to a, then by Lemma 90 we deduce that $\mu(d_A(u)) \leq \mu(u) < \mu(u) + 1$. If no such generator exists, then we can add a "trivial" generator (i, i) where i is such that $u[i] = a$. Now $S \cup \{(i, i)\}$ has $\mu(u) + 1$ elements and the result follows again from Lemma 90. □

Next we give an example of this result and as another example we refer to the end of this chapter, to Example 108.

Example 92. Consider a palindrome $u = aba$ which can be generated by one palindromic generator (1,3). The doubled word $d_{\{a\}}(u) = aabaa$ can be generated by the set $\{(1, 5), (1, 2)\}$ and $\mu(d_{\{a\}}(u)) = \mu(u) + 1$. If we instead double the letter b then we do not need any additional generators, i.e., the doubled word $d_{\{b\}}(u) = abba$ can be generated by $(1, 4)$ and $\mu(d_{\{b\}}(u)) = \mu(u)$.

The next lemma will be used several times, especially, in the proof of Lemma 104.

Lemma 93. *Let $w \in \{0, 1\}^*$ be an unbordered word of length n which is palindromically generated by a set $S \subseteq \mathcal{S}(n)$ with $\#S = 3$. Then w has at most four leaves. Moreover, there are at most two leaves with label 0, and at most two leaves with label 1.*

Proof. We note first that since w is unbordered, the longest palindromic prefix U and the longest palindromic suffix V of w do not overlap. Indeed, if $U = ux$ and $V = xv$ then w has a border x^R; a contradiction.

Let $S = \{(1, i), (j, n), (k, m)\}$. Since the maximal palindromes U and V do not overlap, we have $i < j$, and hence every path starting from a leaf has a unique continuation to a new position until the path enters another leaf. Therefore, there is at most one path for the letter 0 and at most one for the letter 1. These two paths necessarily consume every position of w, and the endpoints of these paths are the leaves of S. □

5.3 Main results

Finally we are ready to express and prove our main results concerning palindromic generating sets of words. First we give a result that in a sense justifies our study. This first main result concerns the well known Thue-Morse infinite word

$$\mathbb{T} = 01101001100101101001\ldots$$

defined as the fixed point beginning in 0 of the morphism $\tau\colon \{0,1\}^* \to \{0,1\}^*$ given by $\tau(0) = 01$ and $\tau(1) = 10$. We prove that such a number of palindromes in a palindromic generating set that would be enough for all factors of Thue-Morse word is not finite.

Theorem 94. *For each positive integer n there exists a factor u of $\mathbb{T}$ with $\mu(u) \geq n$, i.e., $\psi(\mathbb{T}) = +\infty$.*

Proof. Set $t_k = \tau^k(0)$ for $k \geq 0$. We will show that $\mu(t_{2k}) > \mu(t_{2k-2})$ for each $k > 1$ from which it follows immediately that $\psi(\mathbb{T}) = +\infty$. For each $k \geq 0$ there exists $S_{2k} \subseteq \mathcal{S}(2^{2k})$ which palindromically generates t_{2k} and $\mu(t_{2k}) = \#S_{2k}$. We first observe that the prefix t_{2k} of $\mathbb{T}$ of length 2^{2k} is a palindrome, since $t_{2k} = t_{2k-1}t_{2k-1}^R$. Also, since $\mathbb{T}$ is overlap-free (see for instance [68]), it follows that if v is a palindromic factor of t_{2k}, either v lies completely in the prefix t_{2k-1} or completely in the suffix t_{2k-1}^R, or its midpoint is the midpoint of t_{2k}. There necessarily exists a generator in S_{2k} that shares the middle point with t_{2k} in order to relate an occurrence of 0 in the prefix t_{2k-1} with an occurrences of 0 in the suffix t_{2k-1}^R of t_{2k}. Such a generator can always be replaced by the full palindrome $F = (1, 2^{2k})$, and thus without loss of generality we can assume that $F \in S_{2k}$.

If $I = (i,j) \in S_{2k}$ lies in the suffix t_{2k-1}^R of t_{2k}, i.e., if $i > 2^{2k-1}$, then we replace I by its reflection $I' = \rho_F(I)$ which lies entirely in the first half of t_{2k}. Since $\rho_I = \rho_F\rho_{I'}\rho_F$ on the domain of ρ_I, the set $(S_{2k} \setminus \{I\}) \cup \{I'\}$ generates t_{2k}. In this fashion we obtain a generator set S'_{2k} consisting of F and a set of pairs (i,j) where $j \leq 2^{2k-1}$. Thus $S'_{2k} \setminus \{F\}$ palindromically generates the prefix t_{2k-1} of t_{2k}. Since t_{2k-2} is a factor of t_{2k-1}, it follows from Lemma 89 that $\mu(t_{2k}) > \mu(t_{2k-1}) \geq \mu(t_{2k-2})$ as required. □

So there exist infinite binary words x that have $\psi(x) = +\infty$. On the other hand, we we will prove the existence of aperiodic binary words x' for which $\psi(x')$ is just three. In fact, we obtain a complete classification of such words. Next we show that $\psi(y) \geq 3$ always for aperiodic words y.

Lemma 95. *If $x \in \Sigma^\omega$ is aperiodic, then $\psi(x) \geq 3$.*

Proof. Let $r = \#\Sigma$ and let $x \in \Sigma^\omega$ be aperiodic. Suppose to the contrary that $\psi(x) \leq 2$. By Lemma 86, x contains an unbordered factor w of length $|w| \geq 2r + 1$. If w is palindromically generated by a singleton set $\{I\} \subseteq \mathcal{S}(|w|)$, then w contains at least $r + 1$ distinct symbols, a contradiction. If w is palindromically generated by a set $\{(1,p),(q,|w|)\} \subseteq \mathcal{S}(|w|)$ of size 2, then as w is unbordered, it follows that the palindromic prefix $w[1,p]$ does not overlap the palindromic suffix $w[q,|w|]$ (i.e., $p < q$). It follows again that w must have at least $r+1$ distinct symbols, a contradiction. Hence, $\psi(x) \geq 3$. $\square$

Now we are ready to express the main result of this chapter. The proof of this theorem is given in the following subsection.

Theorem 96. *Let $u \in \{0,1\}^+$. Then $\mu(u) \leq 3$ if and only if u is a factor of a double Sturmian word.*

Combining Lemma 95 and Theorem 96 we deduce that:

Corollary 97. *Let $x \in \{0,1\}^\omega$ be aperiodic. Then $\psi(x) = 3$ if and only if x is double Sturmian. In particular, if $\psi(x) = 3$ then x is uniformly recurrent.*

5.3.1 Proof of Theorem 96

We begin by showing that every factor of a double Sturmian word is palindromically generated by a set S of size at most 3. We recall the following fact which is a consequence of a result in [25] (see also [17]):

Lemma 98 (Proposition 22 in [7]). *A word $x \in \{0,1\}^*$ is a central word if it is a power of a single letter or if it satisfies the equations $x = u01v = v10u$ with $u, v \in \{0,1\}^*$. Moreover in the latter case u and v are central words and setting $p = |u| + 2$ and $q = |v| + 2$, we have that p and q are relatively prime periods of x and $\min\{p,q\}$ is the least period of x.*

We also recall the following extremal property of the Fine and Wilf theorem [36] due to de Luca and Mignosi [26]

Lemma 99 (Proposition 23 in [7]). *A word x is a central word if and only if there exist relatively prime positive integers p and q with $|x| = p + q - 2$ such that x has periods p and q.*

We will also use the following property by Lothaire [69] several times to find a period of a palindrome which has a palindromic prefix (and suffix).

Lemma 100 ([69]). *If $uv = vu'$, then $|u|$ is a period of uv.*

Proposition 101. *Let y be a factor of a double Sturmian word $\omega \in \{0,1\}^\omega$. Then $\mu(y) \leq 3$.*

Proof. Let y be a factor of a double Sturmian word ω. Thus there exists a Sturmian word $\omega' \in \{0,1\}^\omega$ and a subset $A \subseteq \{0,1\}$ such that ω is a suffix of $d_A(\omega')$. Let y' be a shortest unbordered factor of ω' such that y is a factor of $d_A(y')$. Because ω' is aperiodic and uniformly recurrent, by Lemma 86 such a factor y' always exists. Now by Lemma 89 it is enough to show that $d_A(y')$ is palindromically generated by a set with at most three generators.

By Lemma 90 it suffices to show that y' is palindromically generated by a set S' with $\#S' \leq 3$ and containing two generators which determine two palindromes of odd length in y' with distinct central symbols .

Since y' is unbordered, by Lemma 87, we may write $y' = axb$ where $x \in \{0,1\}^*$ is a central word and $\{a,b\} = \{0,1\}$. Without loss of generality we can assume that $a = 0$ and $b = 1$. We first consider the case where x is a power of a single letter. If x is empty, then $y' = 01$ is palindromically generated by $S' = \{(1,1),(2,2)\}$. If $x = 0^n$ with $n \geq 1$, then $y' = 0^{n+1}1$ is palindromically generated by $S' = \{(1,n),(1,n+1),(n+2,n+2)\}$. If $x = 1^n$ with $n \geq 1$, then $y' = 01^{n+1}$ is palindromically generated by $S' = \{(1,1),(2,n+1),(2,n+2)\}$. In all of these cases there exist two generators which determine odd length palindromes with distinct central symbols.

Next we assume x is not a power of a single letter. By Lemma 98 there exist central words u and v such that $y' = 0u01v1$. Put $U = 0u0$ and $V = 1v1$. We claim that y' is palindromically generated by the set

$$S' = \{(1,|U|),(|y'|-|V|+1,|y'|),(2,|x|+1)\}.$$

We first note that $y'[1,|U|] = U$, $y'[|y'|-|V|+1,|y'|] = V$ and $y'[2,|x|+1] = x$, whence $y'[1,|U|]$, $y'[|y'|-|V|+1,|y'|]$, $y'[2,|x|+1] \in \mathcal{P}$. Next let $\mathbb{B}$ be a new alphabet and let $w \in \mathbb{B}^*$ be a word with $|w| = |y'|$. Set $U' = w[1,|U|]$, $V' = w[|y'|-|V|+1,|y'|]$, and $x' = w[(2,|x|+1)]$ so that $w = a'x'b'$ with $a', b' \in \mathbb{B}$. Suppose $U', V', x' \in \mathcal{P}$. It follows, e.g., from Lemma 100 that x' has now periods $|U'| = |U|$ and $|V'| = |V|$ and by Lemma 98 they are relatively prime because $|U|$ and $|V|$ are.

Since $|x'| = |U'| + |V'| - 2$, we deduce by Lemma 99 that x' is a central word. Thus x' is either a power of a single letter or it is isomorphic to x. In the first case w is also a power of a single letter. In the second case, w is word isomorphic to y'. Thus in either case there exists a mapping $\nu : \Sigma \to \Sigma'$ with $\nu(y') = w$. So y' is palindromically generated by $S' = \{(1,|U|),(|y'|-|V|+1,|y'|),(2,|x|+1)\}$ where two of the associated palindromes have odd length. Indeed, by Lemma 98 $|U|$ and $|V|$ are relatively prime and $|x| = |U|+|V|-2$. So either $|U|$ and $|V|$ are odd or only the other one is odd but then $|x|$ is odd, too. Because $y' \in \{0,1\}^*$ is unbordered and palindromically generated by a set of size 3, Lemma 93 gives that y' has at most two leaves with label 0 and at most two leaves with label 1. Thus the central symbols of these odd palindromes are necessarily distinct since the first and the last letter of y' are also leaves with different symbols. □

It follows from the following refinement of Lemma 98 that the generating set obtained in Proposition 101 for $y = axb$, where x is a central word, includes both the longest palindromic prefix and the longest palindromic suffix of y.

Proposition 102. *Let $x \in \{0,1\}^*$ be a central word which is not a power of a single letter. Let u and v be as in Lemma 98. Then $0u0$ (resp. $1v1$) is the longest palindromic prefix (resp. suffix) of $0x1$. Moreover, two of the three palindromes $\{x, 0u0, 1v1\}$ are of odd length and have distinct central symbols.*

Proof. Since u is a central word and hence a palindrome, we have that $0u0$ is a palindromic prefix of $0x1$. It remains to show that it is the longest such prefix. Suppose to the contrary that $0x1$ admits a palindromic prefix $0u'0$ with $|u'| > |u|$. Then by Lemma 98, we have that $p = |u|+2, q = |v|+2$ and $p' = |x| - |u'|$ are each periods of x. Since $p' = |x| - |u'| < |x| - |u| = q$, it follows from Lemma 98 that the $\min\{p,q\} = p$. Also, as $|x| = |u| + |v| + 2 \leq |u'| + |v| + 1$, we deduce that

$$|x| \geq |x| - |u'| + |x| - |v| - 1 = p' + p - 1 \geq p' + p - \gcd(p, p').$$

But since both p and p' are periods of x, it follows from the Fine and Wilf Theorem [36] that x has period $\gcd(p,p')$ which by Lemma 98 is equal to p. Whence p divides p'. Let z denote the suffix of x of length p'. Since $0u'0$ is a palindromic prefix of $0x1$ it follows that z begins in 0. On the other hand, since $10u$ is a suffix of x of length p and p divides p', it follows that z begins in 1. This contradiction proves that $0u0$ is the longest palindromic prefix of $0x1$. Similarly one deduces that $1v1$ is the longest palindromic suffix of $0x1$.

By Lemma 98 p and q are relatively prime so two of the three palindromes $\{x, 0u0, 1v1\}$ have odd length and by Lemma 93 the central symbols of these have to be different. □

We will next give a proof for the converse part of Theorem 96, namely:

Proposition 103. *Suppose $w \in \{0,1\}^+$ and $\mu(w) \leq 3$. Then w is a factor of a double Sturmian word ω.*

For the proof of the proposition we will first give a few essential lemmas. In the following lemmas we use a and b as variables of letters such that $\{a,b\} = \{0,1\}$. Lemma 93 on the number of leaves entails some immediate restrictions on w.

Lemma 104. *Suppose $w \in \{a,b\}^*$ be such that $\mu(w) \leq 3$. Then*

(i) The words a^3 and b^3 do not both occur in w.

(ii) For odd k, ba^kb and a^{k+2} do not both occur in w.

(iii) The words ba^kb and a^{k+3} do not both occur in w for any $k \geq 1$.

(iv) All three words a^{k+2}, $ba^{k+1}b$ and ba^kb do not occur in w for any $k \geq 1$.

Proof. In each of the cases we assume that w with $|w| = n$ is a minimal counter example, i.e., none of its proper factors is a counter example. By the minimality assumption, w begins and ends in the expressed forbidden words. Since also the reverse w^R is a counter example, we may assume that the first letter of w is b.

We remind that if w is palindromically generated by S' then all the positions of w have to be covered by palindromic generators of S'. So S' has to contain generators corresponding to a prefix (resp. a suffix) of the longest palindromic prefix (resp. suffix) of w.

For Case (i), let $w = b^3ua^3$ for some u. By the minimality assumption, w is unbordered, and w is palindromically generated by the set $S = \{(1,p),(n-s,n),(i,j)\}$ for some $1 \le p \le 3$, $0 \le s \le 2$ and some (i,j) that determines the palindrome $w[i,j]$. Here either $i > 3$ or $j < n-2$, since the palindrome $w[i,j]$ starts and ends with the same letter. So there are at least three leaves for a or at least three leaves for b which contradicts Lemma 93.

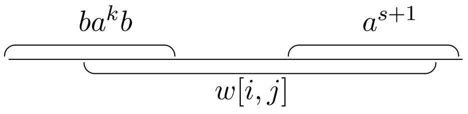


Figure 5.3: The palindromic factors of w in Case (ii).

For Case (ii), let $w = ba^kbua^{k+2}$ for some u. Since w is minimal, it is unbordered, and it is palindromically generated by $S = \{(1,k+2),(n-s,n),(i,j)\}$ for some $1 \le s \le k+1$ and some (i,j). These generators determine the factors $w[1,k+2] = ba^kb$, $w[n-s,n] = a^{s+1}$ and $w[i,j]$, respectively, where the palindrome $w[i,j]$ necessarily misses the first b of $w[1,k+2]$ and the last two a's of $w[n-s,n]$; see Figure 5.3. Also, $w[1,k+2]$ is of odd length, since k is assumed to be odd, and thus its middle position is the third leaf with a letter a. This contradicts Lemma 93.

For Case (iii), let $w = ba^kbua^{k+3}$. Since w is minimal, w is unbordered, and it is palindromically generated by $S = \{(1,k+2),(n-s,n),(i,j)\}$ for some $2 \le s \le k+2$ and some (i,j). The palindrome $w[i,j]$ misses at least the last three a's of w. Again Lemma 93 yields a contradiction.

For Case (iv), consider first any factor of the form $v = ba^kbua^{k+1}$ of w, where ba^kb occurs only as a prefix and a^{k+1} only as a suffix of v. Again v is unbordered, and S contains a generator (i,j), where $i > 1$ and $j < |v|$, i.e., the palindrome $v[i,j]$ misses at least the suffix a and the prefix b of v. By Case (ii), $u \neq ab$ (for otherwise $k+1 \ge 3$ and w contains both bab and a^3), and hence, by Lemma 93, $u \in \{\varepsilon, b\}$. Similarly, if $v = a^{k+1}uba^kb$ then $u \in \{\varepsilon, b\}$. This proves the case since if a^{k+2} and ba^kb occur in w, by the above $ba^{k+1}b$ does not occur in w. $\square$

The key result for the converse is stated in Lemma 106 which gives a simplified criterion for pairs witnessing that a word is unbalanced when compared to the general case of Lemma 105. By Proposition 2.1.3 of Lothaire [69], we have

Lemma 105 ([69])**.** *If $w \in \{a, b\}^*$ is unbalanced then there is a palindrome u such that both aua and bub are factors of w.*

Lemma 106. *Suppose $w \in \{a, b\}^*$ is unbalanced and $\mu(w) \leq 3$. Let aua and bub be any palindromic factors of w. Then $u = a^k$ or $u = b^k$ for some even $k \geq 0$.*

Proof. By Lemma 104 we can assume without loss of generality that b^3 is not a factor of w. Also, the case for $u = \varepsilon$ is clear, and thus we can assume that $u \neq \varepsilon$.

Consider a shortest factor z of w containing both aua and bub. Then z begins with aua and ends with bub, or vice-versa, and z has unique occurrences of the factors aua and bub. We can suppose that z starts with a, since otherwise we take the reverse of z. Now, aua is a maximal palindromic prefix of z, since a longer palindromic prefix of z would contain aua as a suffix. Similarly bub is a maximal palindromic suffix of z.

By Lemma 89, $\mu(z) \leq 3$. Since u is a palindrome, the factors aua and bub do not overlap. Indeed, if $z = au_1bu_2au_3b$, where $u = u_1bu_2 = u_2au_3$, then by taking a reverse of u, we have $u_2^R bu_1^R = u_2au_3$, and so $bu_1^R = au_3$; a contradiction since $a \neq b$. Since aua and bub do not overlap and z is chosen to be minimal, z is unbordered, and thus Lemma 93 applies to z.

Let the palindromic generators of z be $(1, p)$ (s, n) and (i, j). Then $z[1, p]$ is a prefix of the maximal palindromic prefix aua of z, $z[s, n]$ is a suffix of the maximal palindromic suffix bub of z, and $z = xz[i, j]y$, where x and y are non-empty. By Lemma 93, $|x| + |y| \leq 4$, and $|x|, |y| \geq 1$ since two of the leaves (one a and one b) reside at the ends of z. We have $|x| \neq |y|$, since the last letter of aua and the first letter of bub cannot be reflected to each other by the generator (i, j). Therefore, either $|x| = 1$ or $|y| = 1$. We assume that $|y| = 1$ (i.e., $y = b$) and $|x| > 1$, the proof for $|x| = 1$ and $|y| > 1$ being similar. We now have found three leaves, two in the prefix of z and one at the end. Hence, we have $z = z[1, p]vz[s, n]$ where $|v| \leq 1$ since the positions in v are also leaves.

Since bub is a maximal palindromic suffix of z and $y = b$, $z[i, j]$ is preceded by the letter a. It follows that both leaves for the letter a occur in the prefix x, and the remaining fourth leaf resides at a position for the letter b. In particular, if $v \neq \varepsilon$ then $v = b$.

Suppose first that $|u|$ is odd. If $|v| = 1$, and thus $v = b$, then the position of v is the fourth leaf, and $z[1, p] = aua$ or $z[s, n] = bub$. Here $z[s, n] \neq bub$ since otherwise the midpoint of bub would be a fifth leaf. Hence $z[1, p] = aua$

and the midpoint of aua is a leaf. To avoid a fifth leave, it must be at position $i = 2$, i.e., $|u| = 1$, and consequently $z = acabcb$ where $c = a$ because b^3 does not occur in z. However, $\mu(aaabab) = 4$. Therefore, by Lemma 93, $v = \varepsilon$, in which case, $z[1, p] = aua$ and $z[s, n] = bub$, and they both have odd length. Again, necessarily $|u| = 1$ to avoid a fifth leaf, and as above $z = aaabab$ with $\mu(z) = 4$.

Consequently, $|u|$ must be even. Let $u = u_1u_2$ be such that $z[i, j] = u_2atbu_1u_2$ for some t and hence $z = au_1u_2atbu_1u_2b$. By the above, we have $|t| \le 1$, and if $|t| = 1$, then $z[1, p] = aua$ and $z[s, n] = bub$. Here u_2 is a palindrome since $z[i, j]$ is one. Also, in all cases both u_1 and t consist of leaves only, and hence $1 \le |u_1| + |t| \le 2$. Since u is a palindrome, we have $u = u_1u_2 = u_2^R u_1^R = u_2u_1^R$, and, by Lemma 100, $|u_1|$ is a period of u.

If $t \neq \varepsilon$, i.e., $t \in \{a, b\}$, then $|u_1| = 1$, and thus $u = a^k$ for some k. Assume then that $t = \varepsilon$. Now, $|u_1| = 1$ or $|u_1| = 2$. If $|u_1| = 1$, then again $u = a^k$ for some k. Finally, if $|u_1| = 2$, then $z = z[1, p] \cdot z[s, n]$ with $z[1, p] = aua$ and $z[s, n] = bub$ by Lemma 93, since the three positions of the prefix au_1 and the last position of z are leaves. Also, $u_1 = ba$ since $z[i, j]$ and thus also its middle factor abu_1 is a palindrome. Since $z[i, j]$ is a palindrome, and $|u_1|$ is a period of u, we have $z[1, p] = aua = a(ba)^{i+1}ba$ for some i. However, now $|u|$ is odd; a contradiction. This proves the claim. □

For $w \in \{0, 1\}^+ \cup \{0, 1\}^\omega$, we define $A(w) \subseteq \{0, 1\}$ by the rule $a \in A(w)$ if and only if w has no factors of the form $ba^{2k+1}b$ for any $k \ge 0$ with $a \neq b$. If w is a finite word, we define the *lean word* of w to be the shortest word u such that $d_{A(w)}(u)$ contains w as a factor. We extend this notion to the infinite case as follows: If $w \in \{0, 1\}^\omega$, we say $u \in \{0, 1\}^\omega$ is the *lean word* of w if $d_{A(w)}(u)$ contains w as a suffix and for all $v \in \{0, 1\}^\omega$, if w is a suffix of $d_{A(w)}(v)$, then u is a suffix of v. Clearly, if w is not periodic then in each case the lean word of w is uniquely determined by w.

For instance, if $w = 0010011$, then $A(w) = \{0\}$ and the lean word of w is $u = 01011$. Similarly if $w = 011001$, then $A(w) = \{0, 1\}$ and the lean word $u = 0101$. For $w = 0010110$, we have $A(w) = \emptyset$ and hence w is its own lean word.

The proof of the next lemma is based on Lemmas 104, 105 and 106.

Lemma 107. *Let $w \in \{0, 1\}^*$ be a binary word with $\mu(w) \le 3$. Then the lean word u of w is balanced.*

Proof. Let w be a shortest counter example to the claim such that its lean word u is unbalanced. By appealing to symmetry and Lemma 104(i), we can assume that w contains no occurrences of 111.

Since u is not balanced, it has factors $0v0$ and $1v1$ for some palindrome v according to Lemma 105. Now, also $0d_{A(w)}(v)0$ and $1d_{A(w)}(v)1$ are palindromes, and they are factors of w. By Lemma 106, we have $d_{A(w)}(v) = 0^k$

for some even integer k. It follows that $v = 0^m$ for $m = k/2$ or $m = k$ depending on whether 0 is in $A(w)$ or not. Here u has factors 0^{m+2} and 10^m1.

Suppose first that $k > 0$. Since w has factors 0^{k+2} and 10^k1, it has no blocks of 0's of odd length by Lemma 104 (iii) and (iv). Hence $0 \in A(w)$ by the definition of $A(w)$, but then $d_{A(w)}(u)$ has factors 0^{2m+4} and $10^{2m}1$, and thus w has factors 0^{2m+3} and $10^{2m}1$ contradicting Lemma 104(iii).

Suppose then that $k = 0$. Since now 11 occurs in u but 111 does not occur in w, we have $1 \notin A(w)$. By the definition of $A(w)$, the word 010 must be a factor of u, and hence by the minimality of w, we have $u = 00(10)^r11$ for some positive $r \geq 1$ (the case $u = 11(01)^r00$ being similar). From this also $0 \notin A(w)$ follows, and hence $u = w$. However, the palindromic generators of w are now $(1, 2)$ that determines the prefix $w[1, 2] = 00$, and $(n - 1, n)$ that determines the suffix $w[n - 1, n] = 11$, and (i, j) that determines the factor $w[i, j]$. But the palindrome $w[i, j]$ cannot overlap with both $w[1, 2]$ and $w[n - 1, n]$ and thus the two 0's in $w[1, 2]$ and the latter 0 are not equivalent or the two 1's in $w[n - 1, n]$ are not equivalent to a preceding 1; a contradiction. □

Finally, we can complete the proof of the converse part.

Proof of Proposition 103. By Lemma 107, if $\mu(w) \leq 3$, then w is a factor of the word $d_{A(w)}(u)$ for a balanced word u. Since the factors of Sturmian words are exactly the balanced words, see [69], the claim follows. □

Thus now we have also finished the proof of the main result stating that for binary words $\mu(u) \leq 3$ if and only if u is a factor of a double Sturmian word. The proof of this main result, Theorem 96, is straightforwardly concluded from the propositions 101 and 103. We still have to prove the corollary of this theorem given in Corollary 97.

Proof of Corollary 97. Let $x \in \{0, 1\}^\omega$ be aperiodic. By Proposition 95, $\psi(x) \geq 3$.

If x is double Sturmian then, by Proposition 101, $\mu(w) \leq 3$ for all factors of x, and thus by definition, $\psi(x) = 3$.

For the converse, assume that $\psi(x) = 3$, and consider the set $A(x) \subseteq \{0, 1\}$ together with the lean word y of x. Then x is a suffix of $d_{A(x)}(y)$. We need to show that y is Sturmian. Let v be a factor of y, and let u be a factor of y such that u contains v and $d_{A(x)}(u)$ contains a factor of the form $ba^{2k+1}b$ with $k \geq 0$ for each $a \notin A(x)$. Hence $A(d_{A(x)}(u)) = A(x)$, and therefore u is the lean word of $d_{A(x)}(u)$; see Figure 5.4. By Lemma 89, $\mu(d_{A(x)}(u)) \leq 3$, and thus Lemma 107 yields that u, and hence also v, is balanced. It follows that y is Sturmian as required. □

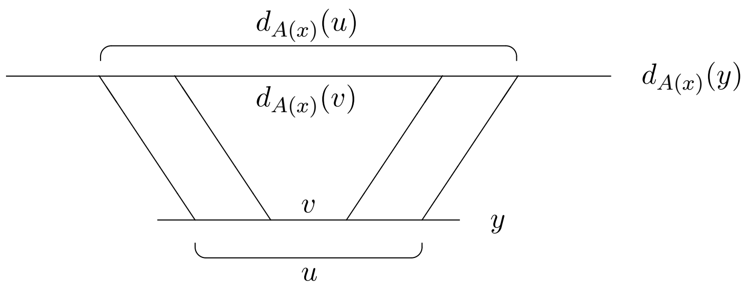


Figure 5.4: Proof of Corollary 97: The word v is extended to a lean factor u of $d_{A(x)}(u)$.

5.4 Conclusions and perspectives

For an infinite word w the value of $\psi(w)$ means the minimal number of palindromes for which each factor of w can be generated up to word isomorphism. As a conclusion, we have shown that for all infinite binary aperiodic words w we need $\psi(w) \geq 3$. We have also shown that there exists an infinite binary word w that has $\psi(w) = +\infty$, namely, for example, the well-known Thue-Morse word. The main result is that we are able to characterize such infinite aperiodic binary words x that have $\psi(x) = 3$. Surprisingly, all these words are related to Sturmian words.

We finish the section of Conclusions and perspectives and at the same time the whole chapter by giving one more example. This shows how we can construct infinite binary words that can be generated, for example, by four palindromes. So there are many open problems for words having different values of $\psi()$ and for different sizes of alphabets.

Example 108. Let us construct a word which is palindromically generated by four palindromes. Let y' be a factor of a double Sturmian word $d_{\{0,1\}}(w) = w' \in \{0,1\}^\omega$ obtained from a Sturmian word w by doubling the letters. By Proposition 101, y' can be generated by three palindromes. Assume that y' contains both 0 and 1 and consider the word $d_{\{0\}}(y')$. Now, by Lemma 91, $d_{\{0\}}(y')$ can be generated by four palindromes, and three generators are not enough unless $d_{\{0\}}(y')$ is a factor of a double Sturmian word.

Take, e.g., φ to be the Fibonacci word, $\varphi = 0100101001001\ldots$ and thus $d_{\{0,1\}}(\varphi) = 00110000110011000011000011\ldots$. Let $y' = d_{\{0,1\}}(\varphi)[6,19] = 00011001100001$. Now $S = \{(1,12),(2,7),(9,14)\}$ is a palindromic generating set for y'. In addition, now the word $d_{\{0\}}(y') = 00000011000011000000001$ is generated by four palindromes, namely $\{(1,2),(1,20),(3,12),(14,23)\}$, where $(1,2)$ determines an added short palindrome 00.

Appendices

Appendix A

Cube-freeness

```
/**
 * To generate 2-abelian cube-free words over an alphabet {0,1}.
 */

public class Cube {

        //word is the list of letters and of size border

    private int[] word;
    private int border;

        //You give the maximal length of the word
        //and the three first letters to the constructor

    public Cube(int size, int fi, int se, int th) {
         word = new int[size];
         word[0] = fi;
         word[1] = se;
         word[2] = th;
         for(int t = 3; t < size; t++){ word[t] = 3; }
         border = size;
    }

        //generating() tries to generate a 2-abelian
        //cube-free word

    public void generating(){
         boolean tob = false;  //tob == true, to add letter 1
         int i = 3;
 while(i < border){
   boolean repetition = false;
   if(tob){ word[i] = 1; }
   else{ word[i] = 0; }
   int maxl = (i+1)/3;

//maxl is the maximal length of word to be repeated

   int j = 1;
   while(j < maxl+1 & !repetition){
     if(word[i] == word[i-j] &
        word[i] == word[i-(2*j)] &
        word[i-j+1] == word[i-(2*j)+1] &
        word[i-j+1] == word[i-(3*j)+1]){
       int[] u = new int[3];  //the first word
       int[] v = new int[3];  //the second word
       int[] w = new int[3];  //the third word
       u = this.factorcount(i-j+1, i);
       v = this.factorcount(i-(2*j)+1, i-j);
       w = this.factorcount(i-(3*j)+1, i-(2*j));
       if(u[0] == v[0] & v[0] == w[0] & u[1] == v[1] &
          v[1] == w[1] & u[2] == v[2] & v[2] == w[2]){
         System.out.println(i);
         repetition = true;
       }
       else{ j = j+1; }  //factors are different
     }
     else{ j = j+1; }  //pref&suf doesn't hold
   }
   if(repetition){
     if(word[i] == 0){
       tob = true;
       repetition = false;     //i doesn't change
     }
     else{ //word[i] == 1, have to go backwards
       int k = i;
       while(k > 2 & word[k] == 1){
         word[k] = 3;
         k = k-1;
       }
       if(k > 2){
         i = k;
         tob = true;
```

```
            }
            else {  //you reached the beginning
              System.out.println(i);
              System.out.println("Termination");
              i = border;
            }
          }
        }
        else{ //repetition==false
          tob = false;
          i = i+1;
        }
      }
      System.out.println(i);
      System.out.println("You reached the border");
}

     //factorcount() counts the number of factors 00,
     //01 and 10 together and 11

public int[] factorcount(int beg, int end){
      int[] result = new int[3];
      for(int i = beg; i < end; i++){
        if(word[i] + word[i+1] == 0){
          result[0] = result[0] + 1; }
        else{
          if(word[i] + word[i+1] == 1){
            result[1] = result[1] + 1; }
          else{ result[2] = result[2] + 1; }
        }
      }
      return result;
}

public String toString(){
      String asword = "[";
      for(int i = 0; i < border; i++){
        asword = asword + word[i]; }
      return asword + "]";
}
```

```
    public int[] aslist(){
         int[] copy = new int[border];
         for(int i = 0; i < border; i++){ copy[i] = word[i]; }
         return copy;
    }
}
```

Appendix B

Square-freeness

```
/**
 * To generate 3-abelian square-free words over
 * an alphabet {0,1,2}.
 */

public class Square {

        //word is the list of letters and of size border

    private int[] word;
    private int border;

        //You give the maximal length of the word
        //and the three first letters to the constructor

    public Square(int size, int fi, int se, int th) {
         word = new int[size];
         word[0] = fi;
         word[1] = se;
         word[2] = th;
         for(int t = 3; t < size; t++){ word[t] = 4; }
         border = size;
    }

        //generating() tries to generate a 3-abelian
        //square-free word

    public void generating(){
         boolean tob = false;  //tob == true, to add letter 1
 boolean toc = false;  //toc == true, to add letter 2
 int i = 3;
 while(i < border){
   boolean repetition = false;
   if(tob){ word[i] = 1; }
   else{
     if(toc){ word[i] = 2; }
     else{ word[i] = 0; }
   }
   int maxl = (i+1)/2;

//maxl is the maximal length of word to be repeated

   if(word[i] == word[i-1]){ repetition = true; }
   if(word[i] == word[i-2] & word[i-1] == word[i-3]){
     repetition = true;
   }
   int j = 3;
   while(j < maxl+1 & !repetition){
     if(word[i] == word[i-j] &
        word[i-j+1] == word[i-(2*j)+1] &
        word[i-1] == word[i-j-1] &
        word[i-j+2] == word[i-(2*j)+2]){
       int[] u = new int[12];  //the first word
       int[] v = new int[12];  //the second word
       u = this.factorcount(i-j+1, i);
       v = this.factorcount(i-(2*j)+1, i-j);
       if(u[0] == v[0] && u[1] == v[1] &&
          u[2] == v[2] && u[3] == v[3] &&
          u[4] == v[4] && u[5] == v[5] &&
          u[6] == v[6] && u[7] == v[7] &&
          u[8] == v[8] && u[9] == v[9] &&
          u[10] == v[10] && u[11] == v[11]){
         System.out.println(i);
         repetition = true;
       }
       else{ j = j+1; }  //factors are different
     }
     else{ j = j+1; }  //pref&suf doesn't hold
   }
   if(repetition){
     if(word[i] == 0){
       tob = true;
```

```
          toc = false;
          repetition = false;     //i doesn't change
        }
        else{
          if(word[i] == 1){
            tob = false;
            toc = true;
            repetition = false;   //i doesn't change
          }
          else{  //word[i] == 2, have to go backwards
            int k = i;
            while(k > 2 & word[k] == 2){
              word[k] = 4;
              k = k-1;
            }
            if(k > 2){
              i = k;
              if(word[i] == 0){
                tob = true;
                toc = false;
              }
              else{
                tob = false;
                toc = true;
              }
            }
            else{  //you reached the beginning
              System.out.println(i);
              System.out.println("Termination");
              i = border;
            }
          }
        }
      }
      else{ //repetition==false
        tob = false;
        toc = false;
        i = i+1;
      }
    }
    System.out.println(i);
    System.out.println("You reached the border");
}
```

```
    //factorcount() counts the number of factors 010, 012,
    //020, 021, 101, 102, 120, 121, 201, 202, 210 and 212

public int[] factorcount(int beg, int end){
     int[] result = new int[12];
     for(int i = beg; i < end-1; i++){
       if(word[i] == 0){
         if(word[i+1] == 1){
           if(word[i+2] == 0){ result[0] = result[0]+1; }
           else{ result[1] = result[1]+1; }
         }
         else{
           if(word[i+2] == 0){ result[2] = result[2]+1; }
           else{ result[3] = result[3]+1; }
         }
       }
       if(word[i] == 1){
         if(word[i+1] == 0){
           if(word[i+2] == 1){ result[4] = result[4]+1; }
           else{ result[5] = result[5]+1; }
         }
         else{
           if(word[i+2] == 0){ result[6] = result[6]+1; }
           else{ result[7] = result[7]+1; }
         }
       }
       else{
         if(word[i+1] == 0){
           if(word[i+2] == 1){ result[8] = result[8]+1; }
           else{ result[9] = result[9]+1; }
         }
         else{
           if(word[i+2] == 0){ result[10] = result[10]+1; }
           else{ result[11] = result[11]+1; }
         }
       }
     }
     return result;
}
```

```
    public String toString(){
         String asword = "[";
         for(int i = 0; i < border; i++){
           asword = asword + word[i]; }
         return asword + "]";
    }

    public int[] aslist(){
         int[] copy = new int[border];
         for(int i = 0; i < border; i++){ copy[i] = word[i]; }
         return copy;
    }
}
```

Bibliography

[1] B. Adamczewski and Y. Bugeaud. On complexity of algebraic numbers, II. Continued fractions. *Acta Math.*, 195:1–20, 2005.

[2] S. I. Adian. *The Burnside Problem and Identities in Groups.* Springer-Verlag, 1979.

[3] M. H. Albert and J. Lawrence. A proof of Ehrenfeucht's Conjecture. *Theoret. Comput. Sci.*, 41:121–123, 1985.

[4] J.-P. Allouche and J. Shallit. *Automatic Sequences: Theory, Applications, Generalizations.* Cambridge University Press, 2003.

[5] G. Badkobeh and M. Crochemore. Finite-Repetition threshold for infinite ternary words. In P. Ambroz, S. Holub, and Z. Masáková, editors, *8th International Conference WORDS 2011*, 63:37–43. EPTCS, 2011.

[6] J. Berstel. Axel Thue's papers on repetitions in words: a translation. *Publications du Laboratoire de Combinatoire et d'Informatique Mathématique, Université du Québec à Montréal*, 20, 1995.

[7] J. Berstel. Sturmian and episturmian words (a survey of some recent results). In S. Bozapalidis and G. Rahonis, editors, *CAI 2007*, LNCS 4728:23–47. Springer, 2007.

[8] J. Berstel and D. Perrin. *Theory of Codes.* Academic Press, 1985.

[9] J. Berstel, D. Perrin, J. F. Perrot, and A. Restivo. Sur le theor‘eme du defaut. *J. Algebra*, 60:169–180, 1979.

[10] F.-J. Brandenburg. Uniformly growing k-th power-free homomorphisms. *Theoret. Comput. Sci.*, 23(1):69–82, 1988.

[11] S. Brlek, S. Hamel, M. Nivat, and C. Reutenauer. On the Palindromic Complexity of Infinite Words. *Internat. J. Found. Comput. Sci.*, 15:293–306, 2004.

[12] J. R. Büchi and S. Senger. Coding in the existential theory of concatenation. *Arch. Math. Logik Grundlag.*, 26:101–106, 1986/87.

[13] A. Cărăuşu and G. Păun. String intersection and short concatenation. *Revue Roumaine de Mathématiques Pures et Appliquées*, 26:713–726, 1981.

[14] A. Carpi. On Abelian Power-free Morphisms. *Int. J. Algebra Comput.*, 3(2):151–167, 1993.

[15] A. Carpi. On Repeated Factors in C^{∞}-Words. *Inf. Process. Lett.*, 52(6):289–294, 1994.

[16] A. Carpi and A. de Luca. Square-free words on partially commutative free monoids. *Inf. Process. Lett.*, 22(3):125–131, 1986.

[17] A. Carpi and A. de Luca. Codes of central Sturmian words. *Theoret. Comput. Sci.*, 340:220–239, 2005.

[18] J. Cassaigne. Unavoidable Binary Patterns. *Acta Inf.*, 30:385–395, 1993.

[19] C. Choffrut, T. Harju, and J. Karhumäki. A note on decidability questions on presentations of word semigroups. *Theoret. Comput. Sci.*, 183:83–92, 1997.

[20] C. Choffrut and J. Karhumäki. Combinatorics of words. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, 1:329–438. Springer-Verlag, 1997.

[21] S. Col. Palindromes dans les progressions arithmétiques, [Palindromes in arithmetic progressions]. *Acta Arith.*, 137:1–41, 2009.

[22] E. Csuhaj-Varjú, I. Petre, and G. Vaszil. Self-assembly of strings and languages. *Theoret. Comput. Sci.*, 374:74–81, 2007.

[23] K. Culik, J. Karhumäki, and A. Lepistö. Alternating iteration of morphisms and the Kolakoski sequence. In G. Rozenberg and A. Salomaa, editors, *Lindenmayer Systems*, 93–106. Springer, 1992.

[24] J. D. Currie. Open Problems in Pattern Avoidance. *Amer. Math. Monthly*, 100:790–793, 1993.

[25] A. de Luca. Sturmian words: structure, combinatorics, and their arithmetics. *Theoret. Comput. Sci.*, 183(1):45–82, 1997.

[26] A. de Luca and F. Mignosi. Some combinatorial properties of Sturmian words. *Theoret. Comput. Sci.*, 136:361–385, 1994.

[27] F. M. Dekking. Strongly nonrepetitive sequences and progression-free sets. *J. Combin. Theory Ser. A*, 27(2):181–185, 1979.

[28] F. Deloup. Palindromes and orderings in Artin groups. *J. Knot Theory Ramifications*, 19:145–162, 2010.

[29] M. Domaratzki. Semantic Shuffle on and Deletion Along Trajectories. In C. Calude, E. Calude, and M. Dineen, editors, *Developments in Language Theory*, LNCS 3340:163–174. Springer, 2005.

[30] M. Domaratzki. Minimality in template-guided recombination. *Inf. Comput.*, 207:1209–1220, 2009.

[31] X. Droubay, J. Justin, and G. Pirillo. Episturmian words and some constructions of de Luca and Rauzy. *Theoret. Comput. Sci.*, 255(12):539–553, 2001.

[32] X. Droubay and G. Pirillo. Palindromes and Sturmian words. *Theoret. Comput. Sci.*, 223(12):73–85, 1999.

[33] A. Ehrenfeucht and D. Silberger. Periodicity and unbordered segments of words. *Discrete Math.*, 26:101–109, 1979.

[34] A. A. Evdokimov. Strongly asymmetric sequences generated by a finite number of symbols. *Dokl. Akad. Nauk SSSR*, 179:1268–1271, 1968. English translation in Soviet Math. Dokl. 9:536–539, 1968.

[35] S. Ferenczi and L. Q. Zamboni. Eigenvalues and Simplicity of Interval Exchange Transformations. *Ann. Sci. École Norm. Sup. (4)*, 44:361–392, 2011.

[36] N. J. Fine and H. S. Wilf. Uniqueness Theorem for Periodic Functions. *Proc. Amer. Math. Soc.*, 16:109–114, 1965.

[37] A. Glen, J. Justin, S. Widmer, and L. Q. Zamboni. Palindromic richness. *European J. Combin.*, 30:510–531, 2009.

[38] V. S. Guba. Equivalence of infinite systems of equations in free groups and semi-groups to finite subsystems. *Mat. Zametki*, 40:321–324, 1986. In Russian.

[39] T. Harju, M. Huova, and L. Q. Zamboni. On Generating Binary Words Palindromically. Manuscript, arXiv:1309.1886 (submitted).

[40] T. Harju and J. Karhumäki. Morphisms. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, 1:439–510. Springer-Verlag, 1997.

[41] T. Harju and J. Karhumäki. Many aspects of defect theorems. *Theoret. Comput. Sci.*, 324:35–54, 2004.

[42] T. Harju, J. Karhumäki, and W. Plandowski. Independent Systems of Equations. In M. Lothaire, editor, *Algebraic Combinatorics on Words*, chapter 13:443–472. Cambridge University Press, 2002.

[43] A. Hof, O. Knill, and B. Simon. Singular Continuous Spectrum for Palindromic Schrödinger Operators. *Commun. Math. Phys.*, 174:149–159, 1995.

[44] M. Holzer and S. Jakobi. Chop Operations and Expressions: Descriptional Complexity Considerations. In G. Mauri and A. Leporati, editors, *Developments in Language Theory*, Proceedings of the 15th International Conference DLT 2011, 264–275. Springer, 2011.

[45] M. Huova. Existence of an Infinite Ternary 64-Abelian Square-free Word. In R. Jungers, V. Bruyère, M. Rigo, and R. Hollanders, editors, *RAIRO - Theoretical Informatics and Applications: Selected papers dedicated to Journees Montoises d'Informatique Theorique 2012.* (to appear).

[46] M. Huova. A Note on Defect Theorems for 2-Dimensional Words and Trees. *Journal of Automata, Languages and Combinatorics*, 14(3-4):203–209, 2009.

[47] M. Huova and J. Karhumäki. Equations in the Partial Semigroup of Words with Overlapping Products. In H. Bordihn, M. Kutrib, and B. Truthe, editors, *Dassow Festschrift*, LNCS 7300:99–110. Springer, 2012.

[48] M. Huova and J. Karhumäki. On k-Abelian Avoidability. In *Combinatorics and graph theory. Part IV, RuFiDiM'11*, Zap. Nauchn. Sem. POMI 402:170–182. POMI, 2012. Translation of the volume in: *J. Math. Sci. (N. Y.)*, 192(3):352–358, 2013.

[49] M. Huova and J. Karhumäki. On the Unavoidability of k-Abelian Squares in Pure Morphic Words. *Journal of Integer Sequences*, 16(2):13.2.2, 2013.

[50] M. Huova, J. Karhumäki, and A. Saarela. Problems in between words and abelian words: k-abelian avoidability. *Theoret. Comput. Sci.*, 454:172–177, 2012.

[51] M. Huova, J. Karhumäki, A. Saarela, and K. Saari. Local Squares, Periodicity and Finite Automata. In C. S. Calude, G. Rozenberg, and

A. Salomaa, editors, *Rainbow of Computer Science*, LNCS 6570:90–101. Springer, 2011.

[52] M. Huova and A. Saarela. Strongly k-Abelian Repetitions. In J. Karhumäki, A. Lepistö, and L. Q. Zamboni, editors, *Combinatorics on Words (Proceedings for the 9th International Conference WORDS 2013)*, LNCS 8079:161–168. Springer, 2013.

[53] M. Ito and G. Lischke. Generalized periodicity and primitivity. *Math. Log. Q.*, 53:91–106, 2007.

[54] J. Karhumäki. Generalized Parikh mappings and homomorphisms. *Information and Control*, 47:155–165, 1980.

[55] J. Karhumäki and S. Mantaci. Defect Theorems for Trees. *Fund. Inform.*, 38:119–133, 1999.

[56] J. Karhumäki, F. Mignosi, and W. Plandowski. The Expressibility of Languages and Relations By Word Equations. *J. ACM*, 47:483–505, 2000.

[57] J. Karhumäki, S. Puzynina, and A. Saarela. Fine and Wilf's Theorem for k-Abelian Periods. In H.-C. Yen and O. H. Ibarra, editors, *Developments in Language Theory - 16th International Conference, DLT 2012*, LNCS 7410:296–307. Springer, 2012.

[58] J. Karhumäki, A. Saarela, and L. Q. Zamboni. Variations of the Morse-Hedlund Theorem for k-Abelian Equivalence. Manuscript, arXiv:1302.3783.

[59] J. Karhumäki, A. Saarela, and L. Q. Zamboni. On a generalization of Abelian equivalence and complexity of infinite words. *J. Combin. Theory Ser. A*, 120(8):2189–2206, 2013.

[60] L. Kari and K. Mahalingam. Watson-Crick palindromes in DNA computing. *Nat. Comput.*, 9:297–316, 2010.

[61] C. Kassel and C. Reutenauer. A palindromization map for the free group. *Theoret. Comput. Sci.*, 409:461–470, 2008.

[62] V. Keränen. Abelian squares are avoidable on 4 letters. In W. Kuich, editor, *Automata, Languages and Programming*, LNCS 623:41–52. Springer, 1992.

[63] D. A. Klarner, J.-C. Birget, and W. Satterfield. On the undecidability of the freeness of integer matrix semigroups. *Int. J. Algebra Comput.*, 1(2):223–226, 1991.

[64] M. Kolarz and W. Moczurad. Directed figure codes are decidable. *Discrete Mathematics and Theoretical Computer Science*, 11(2):1–14, 2009.

[65] J. Leech. A problem on strings of beads. *Math. Gazette*, 41:277–278, 1957.

[66] A. Lepistö. Repetitions in Kolakoski Sequence. In G. Rozenberg and A. Salomaa, editors, *Developments in Language Theory*, 130–143, 1993.

[67] F. W. Levi. On semigroups. *Bull. Calcutta Math. Soc.*, 36:141–146, 1944.

[68] M. Lothaire. *Combinatorics on Words.* Addison-Wesley, 1983.

[69] M. Lothaire. *Algebraic Combinatorics on Words.* Cambridge University Press, 2002.

[70] M. Lothaire. *Applied Combinatorics on Words.* Cambridge University Press, 2005.

[71] R. C. Lyndon and M. P. Schützenberger. The equation $a^M = b^N c^P$ in a free group. *Michigan Math. J.*, 9:289–298, 1962.

[72] G. S. Makanin. The problem of the solvability of equations in a free semigroup. *Mat. Sb. (N.S.)*, 103:147–236, 1977. In Russian and a English translation in Math. USSR-Sb. 32:129–198, 1997.

[73] S. Mantaci and A. Restivo. Codes and Equations on Trees. *Theoret. Comput. Sci.*, 255:483–509, 1998.

[74] S. Mantaci and A. Restivo. On the defect theorem for trees. In A. Ádám and P. Dömösi, editors, *Proc. 8th Int. Conf. Automata and Formal Languages, VIII*, 54:923–932. Publ. Math. Debrecen, 1999.

[75] A. Mateescu, G. Păun, G. Rozenberg, and A. Salomaa. Simple splicing systems. *Discrete Applied Mathematics*, 84:145–162, 1998.

[76] A. Mateescu and A. Salomaa. Parallel Composition of Words with Re-entrant Symbols. *Analele Universtăţii Bucureşti Mathematică-Informatică*, 45:71–80, 1996.

[77] R. Mercas and A. Saarela. 5-Abelian Cubes Are Avoidable on Binary Alphabets. In *Proceedings of the 14th Mons Days of Theoretical Computer Science (2012).* 2012. http://users.utu.fi/amsaar/en/mesa12jm.pdf.

[78] R. Mercas and A. Saarela. 3-Abelian Cubes Are Avoidable on Binary Alphabets. In M.-P. Béal and O. Carton, editors, *Developments in Language Theory - 17th International Conference, DLT 2013*, LNCS 7907:374–383. Springer, 2013.

[79] W. Moczurad. Defect theorem in the plane. *Theor. Inform. Appl.*, 41:403–409, 2007.

[80] M. Morse and G. Hedlund. Unending chess, symbolic dynamics and a problem in semigroups. *Duke Math. J.*, 11(1):1–7, 1944.

[81] P. Ochem, N. Rampersad, and J. Shallit. Avoiding Approximate Squares. *Int. J. Found. Comput. Sci.*, 19(3):633–648, 2008.

[82] The On-Line Encyclopedia of Integer Sequences. http://oeis.org/.

[83] G. Păun, G. Rozenberg, and A. Salomaa. *DNA Computing, New Computing Paradigms.* Springer, 1998.

[84] W. Plandowski. Satisfiability of word equations with constants is in PSPACE. *J. ACM*, 51:483–496, 2004.

[85] P. A. B. Pleasant. Non-repetitive sequences. *Proc. Cambridge Philos. Soc.*, 68:267–274, 1970.

[86] E. L. Post. A Variant of a Recursively Unsolvable Problem. *Bull. Amer. Math. Soc.*, 52:264–268, 1946.

[87] M. Rao. On some generalizations of abelian power avoidability. 2013. Manuscript, perso.ens-lyon.fr/michael.rao/publi/kab.pdf.

[88] G. Richomme, K. Saari, and L. Q. Zamboni. Abelian complexity of minimal subshifts. *J. Lond. Math. Soc.*, 83(1):79–95, 2011.

[89] K. Saari. Lyndon words and Fibonacci numbers. *J. Combin. Theory Ser. A*, 121:34–44, 2013. ArXiv:1207.4233.

[90] D. Skordev and B. Sendov. On equations in words. *Z. Math. Logic Grundlagen Math.*, 7:289–297, 1961. In russian.

[91] A. Thue. Über unendliche Zeichenreihen. *Videnskapsselskapets Skrifter. I. Mat.-naturv. Klasse*, pages 1–22, 1906.

[92] A. Thue. Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen. *Videnskapsselskapets Skrifter. I. Mat.-naturv. Klasse*, pages 1–67, 1912.

[93] T. Zech. Wiederholungsfreie Folgen. *Z. Angew. Math. Mech.*, 38:206–209, 1958.

Errata

A correction for Lemma 89

The Lemma 89 is incorrect as stated in the dissertation on page 83. Here we give a corrected version of Lemma 89. We will state the lemma for a binary alphabet because we will use only that case afterwards in Chapter 5. Fortunately, the new version of Lemma 89 is still efficient enough for the conclusions made from it in Chapter 5.

Lemma 89. *Let $u \in \{0,1\}^+$. Then $\mu(v) \leq \max\{3, \mu(u)\}$ for all factors v of u.*

Proof. The result is clear in case $\mu(u) = +\infty$. So suppose $S \subseteq \mathcal{S}(|u|)$ palindromically generates u and set $k = \#S$. It suffices to show that if $u = ax = yb$, where $a, b \in \{0,1\}$, then $\max\{\mu(x), \mu(y)\} \leq \max\{3, k\}$. We prove only that $\mu(x) \leq \max\{3, k\}$ as the proof that $\mu(y) \leq \max\{3, k\}$ is completely symmetric.

Suppose $S = \{I_1, I_2, \ldots, I_k\}$ palindromically generates u and let $m \in \mathbb{N}$ be the largest integer such that $I = (1, m) \in S$. Let $D = \{r \in \{1, 2, \ldots, k\} \mid I_r = (1, q) \text{ with } q < m\}$. Let

$$S' = S \cup \{I'_r \,|\, r \in D\} \setminus \{I_r \,|\, r \in D\}$$

where for each $r \in D$ we set $I'_r = \rho_I(I_r) = (m - q + 1, m)$ (see Fig. 5.1).

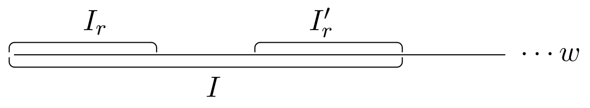


Figure 5.1: Reflecting the generators in S.

It follows that S' also palindromically generates u and I is the only generator in S' containing the initial position 1. Whence 1 is a leaf w.r.t. S', and hence putting

$$S'' = S' \cup \{(2, m - 1)\} \setminus \{I\}$$

it follows that either $S_1 = \{(i-1, j-1) | (i,j) \in S''\}$ or $S_2 = S_1 \cup \{(|x|, |x|)\}$ palindromically generates the suffix x of u. The set S_2 is for the case where $I = (1, |u|)$ and $|u|$ is a leaf w.r.t. S', and otherwise S_1 will palindromically generate x. If $I = (1, |u|)$ and $|u|$ is a leaf w.r.t. S' then $u = ab^n a$, for some integer $n \geq 0$. Clearly, $\mu(ab^n a) \leq 2$ and $x = b^n a$ is palindromically generated by a set $\{(1,1)\}$, $\{(1,1),(2,2)\}$ or $\{(1,n),(2,n),(n+1,n+1)\}$ respectively in cases $n = 0$, $n = 1$ or $n > 1$. This proves the claim. $\square$

Acknowledgement: I want to thank Dr. Michelangelo Bucci for pointing out the inaccuracy in Lemma 89.

1 April 2014

Turku Centre for Computer Science
TUCS Dissertations

ISBN 978-952-12-3034-9
ISSN 1239-1883

Mari Huova

Combinatorics on Words

Mari Huova

Combinatorics on Words

Mari Huova

Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes