Patterns, Pattern Avoidance, and Graphs on Words

Thesis submitted in partial fulfilment of the requirements
for the award of the degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Mrityunjay Singh

Under the supervision of

Dr. Benny George Kenkireth

Department of Computer Science and Engineering

Indian Institute of Technology Guwahati

Guwahati - 781039 Assam India

MAY, 2019

I dedicate this thesis to my late grand mother.

Acknowledgements

First and foremost, I thank God Almighty. I would like to take this opportunity to acknowledge the support and help of my guide Dr. Benny George Kenkireth. He has motivated, supported, and guided me throughout this work. This work has been a really nice learning experience for me. He has taught me how to communicate mathematics. I would also like to thank my doctoral committee members Prof G. Sajith, Prof. Kalpesh Kapoor, Dr. Deepanjan Kesh and Prof. K.V. Krishna for their guidance through out the duration of my doctoral studies.

I would like to thank to the professors P.A.S. Krishna, Shyamsree Upadhyay, Bhaba Kumar Sharma, Rupam Barman, Bikash Bhattacharya, Deepanjan Kesh, Sushant Karmakar, Pinaki Mitra, Purandar Bhaduri, V. Vijaya Saradhi, Sukumar Nandi, H. Ramesh and T. Venketesh for the various courses they taught me during my stay at IITG. I would also like to thank professors John Tsitsiklis, Vittal Rao, Fedrico Ardila, Arindama Singh, Somenath Biswas and Tim Roughgarden for the various online courses taught by them. I thank the whole IITG fraternity for all the different ways in which they have knowingly and unknowingly help me.

I would like to thank my friends Madhusudan Paul, Amit Kumar Srivastava, Pradeep Kumar Sharma, Rohit Tripathi, Laltendu Behra, Subhrendu Chattopadhyay, Samdarsi, Satyendra Kumar, Indrajeet Kumar, Praveen Malik, Ravikant, Arko Sobui, Kashish Babbar, Rishikant Kashyap, Ameya Daigwane, Kalyan Ranuwa and Deepak Yadav. I would also like to thank Dr. Parth Sarathi Deb, Director of the Department of Neurology, GNRC Guwahati.

I am grateful to my family for their love and support. I would like to thank infinite patience of my father, mother, wife, uncles, aunts, brothers, sisters, in laws and my whole extended family.

May 2019 Mrityunjay Singh

Declaration

I certify that

- The work contained in this thesis is original and has been done by myself and under the general supervision of my supervisors.
- The work reported herein has not been submitted to any other Institute for any degree or diploma.
- Whenever I have used materials (concepts, ideas, text, expressions, data, graphs, diagrams, theoretical analysis, results, etc.) from other sources, I have given due credit by citing them in the text of the thesis and giving their details in the references. Elaborate sentences used verbatim from published work have been clearly identified and quoted.
- I also affirm that no part of this thesis can be considered plagiarism to the best of my knowledge and understanding and take complete responsibility if any complaint arises.
- I am fully aware that my thesis supervisors are not in a position to check for any possible instance of plagiarism within this submitted work.

May 2019 Mrityunjay Singh

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati - 781039 Assam India

Certificate

This is to certify that this thesis entitled “**Patterns, Pattern Avoidance, and Graphs on Words**" submitted by **Mrityunjay Singh**, in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy, to the Indian Institute of Technology Guwahati, Assam, India, is a record of the bonafide research work carried out by him under our guidance and supervision at the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Assam, India. To the best of my knowledge, no part of the work reported in this thesis has been presented for the award of any degree at any other institution.

Date: May 2019
Place: Guwahati

Dr. Benny George Kenkireth
Assistant Professor
Email : ben@iitg.ac.in

Abstract

In this thesis, we look at various notions of patterns and pattern avoidance in words. The three themes we have looked at are pattern avoidance on two dimensional words, pattern based word representability of graph and quasiperiodicity patterns and their allied properties in Tribonacci words.

A mapping $f : \mathbb{Z}\times\mathbb{Z} \to \Sigma$ is called a two dimensional word. For each discrete line of a two dimensional word, we can get a one dimensional word by concatenating letters present at the lattice points of the line. If each of these one dimensional words are squarefree then we say that two dimensional word is squarefree. We prove that there are no two dimensional squarefree words on 8 letters.

For a given word w, G_w stands for alternating letter graph corresponding to w. Formally, $G_w = (V_w, E_w)$ where V_w is the set of letters in w and $(a, b) \in E_w$ if the letters a and b are alternating in w. We say that a word w represents a graph G if $G_w = G$. We give a fast algorithm to check if a two uniform word w represents G. We study the problem of counting the number of two uniform representants of the cycle graph and show that the number of two uniform representants of the cycle graph on n vertices is $4n$. We looked at the notion of uniform permutation representability of graphs and found graphs which are (k, p)-representable for some particular k and p.

A word is quasiperiodic if a finite length factor *covers* each of its indices. The Tribonacci words are a family of words generated using the Tribonacci-Rauzy morphisms. We find various parameters related to the quasiperiodicity of the Tribonacci words.

Contents

List of Symbols

Symbols	Description
$\mathbb{N}$	The set of natural numbers
$\mathbb{Z}$	The set of integers
$\mathbb{Z}_n$	The set of non negative integers which are less than n
[n]	The set $\{1, 2, 3 \cdots n\}$
Σ	A finite alphabet containing at least two symbols
$\|\Sigma\|$	The number of letters present in Σ
ϵ	The empty word
Σ^*	The set of finite words on Σ
Σ^ω	The set of infinite words on Σ
$w_{\{a,b\}}$	The word obtained by removing all letters except a and b from w
S_n	The set of all permutations on alphabet [n]
$Pref(w)$	The set of all prefixes of word w
$Suf(w)$	The set of all suffixes of word w
$\|w\|$	The length of the word w
w_i	The letter at the position i in w
w^r	The reverse of the word w
$\mathit{\Sigma}(w)$	The set of all letters present in the word w
$\sigma(w)$	The size of $\Sigma(w)$
$Factor(w)$	The set of all factors of the word w.
$n_x(w)$	The number of times the letter x has appeared in w.
$\delta(x)$	The number of edges incident on the vertex x
C_n	The cycle graph on n vertices
W_n	The wheel graph on $n+1$ vertices

Chapter 1

Introduction

In this chapter, we give a brief introduction to the study of combinatorics of words. We describe the basic mathematical preliminaries which will be used in each chapter, provide the basic definitions which are specifically related with our problems, and describe the organization of the thesis.

1.1 History of combinatorics on words

The earliest work on word combinatorics is traced to the result of Axel Thue[1, 2] in the year 1906. Thue's work was mainly on avoiding repetition in words. Thue created an infinite length word which avoids $xyxyx$ on two letters wherein x and y are any word on a two letter alphabet. Such a word was created by iterating a carefully constructed morphism. This morphism is now known as *Thue-Morse morphism* or *Prouhet-Thue-Morse sequence*. With the help of Thue Morse morphism, it is possible to construct a long word on three letter which avoids xx where x is a word on three letters. There are various problems in word combinatorics related to patterns and pattern avoidance. A famous conjecture on pattern avoidance named Dejean's conjecture was resolved in a series of papers [3, 4, 5, 6, 7, 8, 9, 10]. There are many interesting open problems in this area. Interested readers are encouraged to refer these articles [11, 12]. A recent topic of research in combinatorics on words was pioneered by Blanchet-Sadri in [13]. She has used a particular kind of word called *partial words*. In a partial word, some positions named as *holes*, on which any letter can appear. Note that the letters appears in holes can be different. These kind of words can be used to model words formed by partial information. Kitaev et.al., have studied the interaction between words and graphs and have introduced the notion of word representable graphs[14].

The results in combinatorics on word have application in cryptography, number theory, and bioinformatics. The interested reader may see [15, 16, 17] for a more comprehensive treatment of the subject.

1.2 Basic terminology

A *word* is a finite sequence of elements from a finite set Σ. The set Σ is called the *alphabet*. An element of the alphabet is called a *letter*. The symbol $|\Sigma|$ denotes cardinality of the alphabet Σ. The letters of the alphabet are denoted by small letters like a,b,$c \ldots$. The *length* of a word w, denoted by $|w|$, is defined as the number of letters, counting multiplicities, in w. The letter at index i in the word w is denoted by w_i. The empty word, i.e., the word of length zero, is denoted by ϵ. The set Σ^* denotes set of all finite words. Σ^+ denotes the set of all non-empty words. An *infinite word* is an infinite sequence of elements from a finite set Σ. Infinite words are represented by small bold letters like **x**,**y**, **z**, $\ldots$. The set Σ^ω denotes set of all infinite words. A finite non empty word can be viewed as a function from [n] to Σ. An infinite word can be viewed as a function from $\mathbf{N}$ to Σ. Finite words and one way infinite words can be viewed as members of the sets Σ^* and Σ^ω respectively. The set of letters occurring in a word w is denoted by $\Sigma(w)$ and the size of $\Sigma(w)$ is denoted by $\sigma(w)$.

Example 1.1. *$w = abccabbb$ is a word of length 8 on the alphabet $\Sigma = \{a, b, c\}$. For the word $s = abcdabc$, $\Sigma(s) = \{a, b, c, d\}$ and $\sigma(s) = 4$. The word $\mathbf{x} = abcabcabc\ldots$ is an infinite word on the alphabet $\{a, b, c\}$.*

Given words x and y, let xy denote the word corresponding to the sequence obtained by appending the sequence corresponding to the word y to the sequence corresponding to the word x. The *concatenation* of the words x and y is defined to be the word xy. Notice that concatenation is an associative operation. A word y is a *factor or subword* of a word w if w can be written as xyz. We denote by $Factor(w)$ the set of all factors of the word w. For $k \in \mathbb{N}$ and $w \in \Sigma^*$, w^k denotes the word obtained by concatenation of k copies of w.

Example 1.2. *Let $s = abb$ and $t = ba$. Then $st = abbba$, $Factor(s) = \{\epsilon, a, b, ab, bb, abb\}$ and $s^3 = abbabbabb$.*

A word w is called *primitive* if it can not be written as $w = u^k$ for $k \in \mathbb{N}$ and $k > 1$. Otherwise, it is called a *non primitive* word. A word u is a *prefix* of a word w if w can be written as uv. A word u is a *suffix* of a word w if w can be written as vu. The set of all prefixes of a word w is denoted by $Pref(w)$. The set of all suffixes of a word w is denoted by $Suf(w)$. A word u is called *border* of a word w if u is both a prefix as well as a suffix of w. Consider the word $w = a_1a_2\ldots a_n$ where $a_i \in \Sigma$ and $1 \leq i \leq n$. The *reverse* of w, denoted by w^r is the word $a_na_{n-1}\ldots a_1$.

Example 1.3. *$w = abccabbb$ is a word of length $|w| = 8$ on the alphabet $\Sigma = \{a, b, c\}$. The word cca is the factor of the word w. The word w is a primitive word whereas word $w' = abab = (ab)^2$ is not a primitive word. The prefix set $Pref(w)$ is equal to the set $\{a, ab, abc, abcc, abcca, abccab, abccabb, abccabbb\}$ and the suffix set $Suf(w)$ is equal to the set $\{b, bb, bbb, bbba, bbbac, bbbacc, bbbaccb, bbbaccba\}$*

1.3 Morphism

In this section, we describe the notion of a *morphism*. Morphisms can be used to transform a string into another. Morphisms satisfying certain properties can be used repeatedly to generate infinite words.

Definition 1.1. *Let Σ and Δ be two alphabets. A morphism h is a function from Σ^* to Δ^* such that $h(xy) = h(x)h(y)$ where x,$y \in \Sigma^*$.*

A morphism is uniquely specified upon specifying the images on all elements in Σ. A morphism from Σ^* to itself is called an *endomorphism*. A morphism is *non-erasing* if $h(a)$ is non empty for every letter in Σ. A morphism is called *k-uniform* if the word $h(a)$ is of length k for every a in Σ. A morphism is called *growing* if it is non empty and for at least one letter a in Σ, $|h(a)|$ is greater than one. For an endomorphism, let $h^i(a)$ be defined as the application of h, i times to a. Suppose h is growing endomorphism such that $h^i(a)$ is the prefix of $h^{i+1}(a)$ for all $i \in \mathbb{N}$, then $\lim_{i\to\infty} h^i(a)$ generates a unique infinite length word. The infinite length word obtained by repeated application of growing endomorphism h on the letter a is denoted by $h^\omega(a)$.

Example 1.4. *Let h be the morphism given by $h(0) = 011$,$h(1) = 10$. The infinite word h^ω obtained by repeated applications of h is the sequence given by*

$$h^\omega(0) = 01110101001110011\cdots$$

1.4 Pattern avoidance

The two main notions of pattern avoidances studied in this thesis are repetition avoidance or square avoidance and permutation avoidance. The words with avoid a pattern will be referred to as a pattern free word. While studying squarefree words, we shall assume that letters in the alphabet are unordered, generally denoted by $a, b, \cdots$ whereas in permutation avoiding words the letter of the alphabet are ordered, generally denoted by $1, 2, \cdots n$ where $n \in \mathbb{N}$. The square pattern is said to be avoidable if it is possible to construct an arbitrary long word on Σ which does not have factor $\alpha\alpha$ for any $\alpha \in \Sigma^*$. A word w on alphabet **[n]** $= \{1, 2, \cdots n\}$ avoids permutation p (permutation is a word on alphabet **[k]** where each letter of **[k]** appear exactly once in p) if there exist no indices (these indices must be in increasing order) in w such that, the order of the letter present in the indices mimics the order of the letters in permutation p. A more formal description shall be given the relevant chapters.

Example 1.5. *The morphism given by $t(0) = 01$ and $t(1) = 10$ on the binary alphabet $\{0, 1\}$ is called the Thue Morse morphism. By iterating morphism t on 0 we get the infinite word $t^\omega(0) = 0110100110010110\cdots$. This word is called the Thue Morse word. Thue Morse word does not contain pattern $xyxyx$ where x and y are the words on an alphabet $\{0, 1\}$[18].*

Example 1.6. *The word 4321 does not contain permutation 132 whereas the word 2413 contains permutation 132 because the word obtained from the letters at the indices 1^{st},2^{nd} and 4^{th} in 2413 is 243 and order of letters in the obtained word is same as the order of letters in 132 [19].*

1.4.1 Fibonacci words and its variants

The Fibonacci sequence is given by the following recurrence relation

$$f(0) = 1, f(1) = 1 \text{ and } f(n) = f(n-1) + f(n-2) \text{ for } n \geq 2.$$

The n th Fibonacci word w_n is obtained by following a similar process given below;

$$w_0 = 0, w_1 = 1 \text{ and } w_n = w_{n-1}w_{n-2} \text{ for } n \geq 2$$

. Note that the Fibonacci words can be generated using morphisms as well. Let h be the morphism given below;

$$h(0) = 1, h(1) = 10$$

Then, w_n is equal to $h^n(0)$. The Tribonacci and k-bonacci words are defined using a very similar process. These words are well studied and we explored the quasi periodicity properties of Tribonacci words. Interested reader may find more about these words here [19].

1.5 Multidimensional words

A word can be seen as function $f : \mathbb{N} \to \Sigma$ where Σ is an alphabet. This notion can be extended to the case of an n *dimensional word* which is defined as $f : \mathbb{N}^n \to \Sigma$ where $n \in \mathbb{N}$.

Finite two dimensional words can be viewed as matrices with entries from the underlying alphabet.

Example 1.7. *The following figure is an example of a finite two dimensional word on alphabet* $\{a, b, c\}$.

b	a	c	b	a
a	c	b	a	c
c	b	a	c	b
b	a	c	b	a
a	c	b	a	c

Figure 1.1: A 5×5 word on $\{a, b, c\}$

While studying multi dimensional words, we often look at the patterns present in the one dimensional words present in them. For example, in Figure 1.1, we may look at the columns and rows of the word and obtain many one dimensional words. We can additionally look at the discrete lines in the multidimensional words and extract words corresponding to them and them study the patterns present in them. For example, the diagonal starting at $(0, 0)$ *in Figure 1.1 has the word* $aaaaa$.

1.6 Word representable graphs

We say that a graph $G = (V, E)$ is *word representable* if there exists a word $w \in V^*$ such that $(a, b) \in E$ iff the word obtained by removing the letters from w other than a and b is of the form either $abab\cdots$ or $baba\cdots$ [14].

Example 1.8. *Graph given in Figure 1.2 is word representable.*

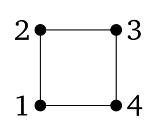


Figure 1.2: Representant word $w = 1241342$

1.7 Problems addressed in this thesis

The problems studied in this thesis are;

1. Minimum alphabet size required to construct two dimensional square free words.

Any word of length four or more on a two letter alphabet will contain a square. Thue constructed an infinite length word on three letter alphabet which avoids a square. It was known that two dimensional words on an alphabet of size six will invariably contain a square and that there exist an infinite squarefree two dimensional word on a sixteen letter alphabet. We improve the lower bound on this result.

2. Uniform word representability and permutation word representability of graphs.

   Kitaev introduced the notion of word representability of graphs. We further explore this notion and study the graphs which can be represented by special words like uniform words and permutation avoiding words. We also study some counting problems involving the number of representant words for certain special classes of graphs.

3. Quasiperiodic properties of Tribonacci words.

   Periodic properties of words are a well studied theme. In this problem we look at a quasiperiodity properties, i.e, a relaxation to the periodicity, for a special word known as the Tribonacci word. In particular, we characterize the covers and seeds of these words.

1.8 Organization of the thesis

In Chapter 2, we address the problem of finding the minimum alphabet size which is required to avoid a square in a two dimensional word. This question was posed in [20]. We improve the lower bound for the minimum alphabet size from 7 to 9. This result helps in improving lower bound of minimum alphabet size in an n dimensional squarefree word.

In Chapter 3, we study problems associated with word representability of graphs. Given a graph $G = (V, E)$ and a two uniform word $w \in V^*$, we would like to know if $G = G_w$. We propose an $\mathcal{O}(V \log V + E)$-time algorithm to solve this problem. We also study the problem of counting two uniform representant of the cycle graph and show that the number of two uniform words which represents the n vertex cycle graph is $4n$. We explore various permutation avoidance patterns in this chapter. The patterns we have studied are

- 2 uniform 132 representability
- 1342 representability
- 2 uniform 1342 representability without being 132 representable

In Chapter 4, we study quasiperiodicity properties of Tribonacci words. This is an extension of the work by Christou et.al [21]. We characterize the borders, covers and seed on the Tribonacci words. In the concluding chapter we describe some open problems related to our work.

Chapter 2

Square Free Words

In this chapter, we study two dimensional squarefree words. We are interested in the minimum alphabet size required to construct such a word. This question was consider by Carpi [20] and he showed that sixteen letters are sufficient and six letters are necessary to construct such a word. We tighten this gap and show that at least nine letters are necessary.

A squarefree two dimensional word avoids squares on all the one dimensional words corresponding to the discrete lines. For every lattice point p, there are many discrete lines passing through that point and therefore many other lattice points will be adjacent to p on some discrete line. Each of these points must necessarily have a different letter from the letter at p. We capture this idea in terms of “parity” of a letter. Section 2.1 introduces the basic definitions and results. The problem definition is provided in Section 2.2. We introduce the concept of “parity of a letter” in Section 2.4. Using these, in section 2.5, we derive some conditions that axis parallel words of a two dimensional squarefree words must satisfy. In Section 2.6, we obtain further conditions on two dimensional squarefree words by looking at the modulo two parities. In Section 2.7, we show that it is impossible to construct a two dimensional squarefree words which satisfies the necessary conditions derived in the earlier sections.

2.1 Mathematical preliminaries

In this section we define the concept of pattern avoidance in words. The words studied in this chapter are over an unordered alphabet.

2.1.1 Pattern and pattern avoidance

Let Σ be the alphabet and consider a set X such that $X \cap \Sigma = \emptyset$. We shall refer to the elements of X as *variables*. A *pattern* is a word on the alphabet $(\Sigma \cup X)$. For $x, y \in X$, the pattern xx is called a *square* and the pattern $xyxyx$ is called an *overlap*. (See Figure 2.1)

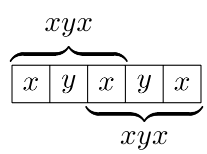


Figure 2.1: The overlap pattern

Definition 2.1. *The language defined by a pattern p, denoted by $L(p)$ is the set obtained by substituting the variables in the pattern p with elements of Σ^+. Formally,*

$$L(p) \triangleq \{f(p) \mid f : (\Sigma \cup X)^* \to \Sigma^*, \textit{ where } f \textit{ is a non-erasing morphism.}\}$$

A word w on alphabet Σ avoids a pattern p if $Factor(w) \cap L(p) = \emptyset$.

Example 2.1. *Consider the pattern $p = 0\alpha 0\alpha 0$ where $X = \{\alpha, \beta\}$ and $\Sigma = \{0, 1\}$. The pattern language would be $L(p) = \{0u0u0 : u \in \{0,1\}^+\}$. The word* 101101100 *does not avoid p, whereas the word* 011010 *avoids p.*

Definition 2.2. *We say that a pattern p is avoidable, if there exists an alphabet Σ such that there are infinitely many words in Σ^* which avoids pattern p. A pattern which is not avoidable is called an unavoidable pattern.*

For $k \in \mathbb{N}$, a word w is called *k-free* if it avoids pattern u^k. A word is called *squarefree* it it avoids the square pattern. A word is called *overlap free* if it avoids the overlap pattern. Thue showed that there exists an overlap free word on a two letter alphabet and a squarefree word on a three letter alphabet. Clearly such words cannot be constructed on a smaller sized alphabet. The Thue Morse word (ref. Example 1.5) can be used to construct a square free word. Note that the Thue Morse word $\mathbf{T} = \lim_{n\to\infty} T_n$ is the limit word as the sequence described by the equations below.

$$T_0 = 0$$

$$T_{n+1} = T_n \overline{T_n}$$

where $\overline{T_n}$ is the bitwise complement of T_n. We can inductively prove that every T_n and therefore limit word $\mathbf{T}$ will be cube free. In particular, the number of ones in between any two occurrences of zeros will be less than three. Let a_i denote the number of ones in between i^{th} and $(i+1)^{st}$ zeros in $\mathbf{T}$. The infinite word $\mathbf{S}$ obtained by concatenating the a_is will be a ternary word on the alphabet $\{0, 1, 2\}$. The word $\mathbf{S}$ will be squarefree as any square in $\mathbf{S}$ will force $\mathbf{T}$ to contain cube. The details of proof can be found in [18].

2.1.2 Multidimensional words and pattern avoidance

A two dimensional word is a function from [n]$\times$ [m] to Σ. A two dimensional infinite word is a function from $\mathbb{Z} \times \mathbb{Z}$ to Σ. We may restrict two dimensional words to be functions from $\mathbb{N} \times \mathbb{N}$ to Σ. We shall not be overly concerned with whether we are looking at finite two dimensional words or functions from $\mathbb{Z} \times \mathbb{Z}$ or $\mathbb{N} \times \mathbb{N}$ as our main focus is on avoidability. The claims we make can be further carefully analyzed to provide an upper bound on the size of the largest two dimensional squarefree word on eight letters. We note in passing that this size is about 20×20 and hence the problem attempted herein is not amenable to a computer aided brute force search. The definitions provided here for the two dimensional case can be naturally extended to multidimensional words. The dimension of a word is usually clear from the context.

The notions of *factor*, *size* (in place of the one dimensional length), $\Sigma(\cdot), \sigma(\cdot)$ etc can be naturally defined in case of multidimensional words. The notion of pattern avoidance needs to be clearly articulated as there are many competing definitions. In this chapter, although words considered are multidimensional, the pattern that is to be avoided is a simple one dimensional word.

Let w be an n-dimensional word. A *line word* in w is obtained by looking at a *discrete line* in w and extracting the letters in w to form a one dimensional word.

Definition 2.3. *Let $i = (i_1, i_2, \ldots, i_n)$ and $j = (j_1, j_2, \ldots, j_n)$ be n-tuples of integers such that $gcd(j_1, j_2, \ldots, j_n) = 1$. Let w_x denote the letter at position x in w, i.e., $w(x)$. The line word corresponding to i (starting point) and j (slope) is given by sequence of letters $w_{i+\alpha j}$ where α takes integer values such that all the indices are within the range specified by the word w.*

Example 2.2. *In Figure 2.2, the words $cbacb$ and $ccccc$ are the line words corresponding to the lines l_1 and l_2 respectively.*

An n-dimensional word w is *squarefree* if all the line words in w are squarefree. In case of a two dimensional word w, every position in w can be represented by a tuple (i, j) where $i, j \in \mathbb{Z}$. Certain types of line words occur frequently in our analysis and so we name them. The words corresponding to the lines parallel to x-axis and y-axis will be referred to as *row word* and *column word* respectively. The words arising out of lines having slopes

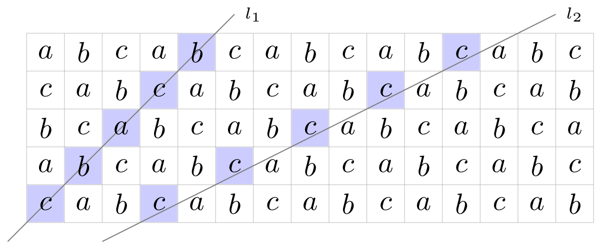


Figure 2.2: Discrete lines in a two dimensional word

$45°$ and $-45°$ will be referred to as *diagonal words*. A word which is either a row word or a column word will be called an *axis parallel word*. Given a two dimensional word w and points s and t, the two dimensional word w_{st} is defined as the two dimensional word formed from all the letters in the axis parallel rectangle with s and t as opposite corners. In a two dimensional word, the words corresponding to two consecutive rows (or columns) are called as *adjacent* words. Let u and v be the subwords of a two dimensional word w such that the discrete lines corresponding to u and v are parallel to an axis and the distance between these lines is a natural number say k. We call such words as k*-separated words*. In particular adjacent axis parallel words at are 1-separated words.

Example 2.3. *In Figure 2.3, the words $abcabcabcabcabc$ and $abcabcabcabcabc$ are seperated by distance three.*

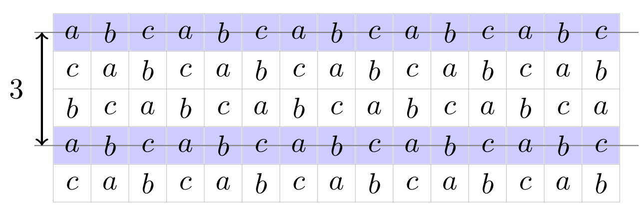


Figure 2.3: 3-seperated words

2.2 Problem statement and known results

Let $f(n)$ be the minimum size of alphabet on which an infinite n dimensional square free word can be constructed. Carpi proved that $2 \times 3^{n-1} \leq f(n) \leq 4^n$ [20]. Improving the lower bound of this inequality is the main result of this chapter. Thue [1] proved that $f(1) = 3$. In case of $n = 2$, $6 \leq f(2) \leq 16$.

2.3 Preliminary Observations

Lemma 2.1. *In a two dimensional word w, if there exist $p, q, i, k \in \mathbb{N}$ such that $w_{(p,i)} = w_{(q,i+k)}$ or $w_{(i,p)} = w_{(i+k,q)}$ where $gcd(p-q, k) = 1$, then the word w contains square.*

Proof. The lattice points (p, i) and $(q, i+k)$ are consecutive points in the line passing through these points when $gcd(p-q, k) = 1$. If the letters appearing at these points are same, they together constitute a square in w. Similar reasoning applies for the lattice points (i, p) and $(i+k, q)$.

□

A trivial but useful observation is that if two adjacent axis parallel words have a common letter, the discrete line passing through the positions of this common letter contains a square. We state this observation as a corollary.

Corollary 2.1. *If u and v are adjacent axis parallel words in w such that $\Sigma(u) \cap \Sigma(v) \neq \emptyset$, then w contains a square.*

2.3.1 Parity of a letter

For $n \in \mathbb{N}$, we use the notation $\mathbb{Z}_n$ to denote the set $\{0, 1, 2 \cdots, n-1\}$. For $a \in \Sigma, w \in \Sigma^*$ and $n \in \mathbb{N}$, the *parity* of a letter a in the word w with respect to n, denoted by $\pi_n(a, w)$ is given by

$$\pi_n(a, w) \triangleq \{j \in \mathbb{Z}_n \mid w_i = a \text{ and } j \equiv i \mod n\}$$

Given the set of positions where a letter a appears in the word w, the function π_n computes the *residues* of these positions modulo n. If a letter a appears in both even and odd positions in a word w, then $\pi_2(a, w)$ will be equal to $\{0, 1\}$. If a letter a appears only in even positions in a word w, then $\pi_2(a, w)$ will be equal to $\{0\}$.

For $n = (n_1, n_2, \ldots, n_d) \in \mathbb{N}^d$, we use the notion $\mathbb{Z}_n$ to denote the cross product $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_d}$. We generalize the above definition to the multidimensional case. Let w be an d-dimensional word. For $a \in \Sigma$ and $n \in \mathbb{N}^d$

$$\pi_n(a, w) \triangleq \{j \in \mathbb{Z}_n \mid w_i = a \text{ and } j \equiv i \mod n\}$$

where $j \equiv i \mod n$ means that the equation holds good coordinate wise.

Example 2.4. *Consider the two dimensional word w shown in Figure 2.4. The parity of various letters w.r.t $n = (2, 2$ are as follows.*

1. $\pi_{(2,2)}(a, w) = \{(0, 0)\}$

2. $\pi_{(2,2)}(b, w) = \{(1, 0)\}$

3. $\pi_{(2,2)}(c, w) = \{(0, 1), (1, 1)\}$

4. $\pi_{(2,2)}(d, w) = \{(1, 1)\}$

a	b	a
c	c	c
a	b	a

Figure 2.4: Two dimensional word on the letters a, b, c and d denoted by w

We define a pseudo inverse of the function π as follows. Given a d-dimensional word $w, n \in \mathbb{N}^d, j \in \mathbb{Z}_n$,

$$\pi_n^{-1}(j, w) \triangleq \{a \in \Sigma \mid w_i = a \text{ and } j \equiv i \mod n\}$$

In other words $\pi_n^{-1}(j, w)$ denotes the set of letters which can appear in a position whose residue modulo n is j.

Example 2.5. *In Figure 2.4 the pseudo inverse of the function π for each of the letters are as follows.*

1. $\pi^{-1}((0, 1), w') = \{c\}$

2. $\pi^{-1}((1, 1), w') = \{c\}$

3. $\pi^{-1}((0, 0), w') = \{a\}$

4. $\pi^{-1}((1, 0), w') = \{b\}$

A letter a is called a *fixed n-parity letter* in a word w if $|\pi_n(a, w)| = 1$. A word w is a *fixed n-parity word* if every letter $a \in \Sigma(w)$ is a fixed n-parity letter in w. In Figure 2.4 the letters a and b have fixed parity and the letter c does not have fixed parity. When n is clear from the context, we shall simply call the word as a *fixed parity word*.

2.4 Some results on parity of squarefree words

Consider a squarefree word on three letters. We examine how many of these letters can occur at both even and odd positions. We show that at least two letters must occur in both even and odd positions.

Lemma 2.2. *Let w be a squarefree word on a three letter alphabet Σ such that $|w| \geq 20,$ and let S be the set of fixed 2-parity letters. Then $|S| \leq 1$.*

Proof. Let Σ be $\{a, b, c\}$. Note that every factor of w of length four must contain every letter of Σ as there are no squarefree word of length four on a two letter alphabet.

For the sake of contradiction, let us assume that $|S| \geq 2$. W.l.o.g, let $a, b \in S$. We need to consider two cases namely

1. $\pi_2(a, w) = \pi_2(b, w)$
2. $\pi_2(a, w) \neq \pi_2(b, w)$

In the first case let i be the smallest number such that $w_i = a$. Note that $i \leq 4$. If $w_{i+2j+1} = a$ where $0 < i + 2j + 1 < |w|$, then $a \notin S$. Similarly w_{i+2j+1} cannot equal to b either. Thus , $w_{i+2j+1} = c$. One can easily verify that, under these conditions, the word $w_i w_{i+1} \ldots w_{i+7}$ contains a square.

For the remaining case, we can w.l.o.g. assume that $\pi_2(a, w) = \{0\}, \pi_2(b, w) = \{1\}$ and $\pi_2(c, w) = \{0, 1\}$.

Assume that $w_i = a$ and $w_{i+2} \neq a$. Based on the parity restrictions and squarefreeness, we can infer the possible letters at other locations. This information is summarized in Table 1. The justification for the inferences are provided just below the word under consideration

The last line of the table above shows that a square starting at position i of length 12 is forced by the assumptions. Therefore we conclude that once an a appears in w, all the subsequent even positions will contain a. By symmetry, once a b appears, all the subsequent odd positions will contain b. Clearly such a word cannot be squarefree.

□

The next couple of lemmas tells us about the maximum number of letters that can be shared by axis parallel words.

Lemma 2.3. *If u and v are 2-separated words in w such that and $|\Sigma(u) \cap \Sigma(v)| > 1$ and $\sigma(u) = \sigma(v) = 3$, then w contains a square.*

Letters of the word w starting at the i^{th} letter												
i	i+1	i+2	i+3	i+4	i+5	i+6	i+7	i+8	i+9	i+10	i+11	i+12
a		c										
Initial assumptions and parity constraints.												
a		c		a								
Every 4 length factor should contain an a.												
a	b	c	b	a								
Squarefreeness at $i+1$ and $i+3$												
a	b	c	b	a		a						
If $w_{i+6} \neq a$, then the $bcba$ starting at $i+1$ has to repeat.												
a	b	c	b	a	c	a						
Squarefreeness of the 4 length factor starting at $i+3$.												
a	b	c	b	a	c	a	b					
Squarefreeness starting at $i+4$												
a	b	c	b	a	c	a	b		b			
If $w_{i+9} \neq b$ then the string starting at $i+3$ repeats.												
a	b	c	b	a	c	a	b	c	b			
Squarefreeness of the 4 length factor starting at $i+6$												
a	b	c	b	a	c	a	b	c	b	a		
Squarefreeness of the 4 length factor starting at $i+7$												
a	b	c	b	a	c	a	b	c	b	a		a
If $w_{i+12} \neq a$ then $bcba$ starting at $i+7$ has to repeat.												
a	b	c	b	a	c	a	b	c	b	a	c	a
Parity constraints and squarefreeness at $i+9$												

Table 2.1: Construction of squarefree word w when $w_i = w_{i+4} = a$ and $\pi_2(a,w) = \{0\}, \pi_2(b,w) = \{1\}$ and $\pi_2(c,w) = \{0,1\}$

Proof. Let a and b be the letters which are contained in both u and v. As u is squarefree, by Lemma 2.2, at most one letter among a and b have a fixed parity(i.e., $|\pi_2((u),\cdot)| = 1$). Therefore, we may assume w.l.o.g that the letter a appears at points (r,i) and (s,i) where r is an even number and s is an odd number. Note that for any integer x, either $gcd(x-r,2)$ or $gcd(x-s,2)$ is equal to one. Let x to be the position in v where a appears. By Lemma 2.1 w contains a square.

□

Lemma 2.4. *If u and v are 3-separated words in w such that and $\Sigma(u) \cap \Sigma(v) \neq \emptyset$ and $\sigma(u) = \sigma(v) = 3$, then w contains a square.*

Proof. Let a and b be the letters which are contained in both u and v. We may assume that both these words are row words squarefree. If a is a fixed 3-parity word in u(or v) with parity p, then a must appear at all positions of the form $p \pm 3k$. Clearly we cannot construct a squarefree word under these requirements. So we can assume that $|\pi_3(u,a)|$ and $|\pi_3(v,a)|$

are greater than 1. This means that we can find positions (r, i) and $(s, i+3)$ in u and v such that $w_{r,i} = w_{s,i+3} = a$ and $gcd(r - s, 3) = 1$. By Lemma 2.1 this implies that w contains a square.

□

2.5 Axis parallel words

For a two dimensional squarefree word w, the number of letters used in the axis parallel words helps us lower bound the number of letters in w. These bounds are summarized in the two following lemmas.

Lemma 2.5. *Let w be a two dimensional squarefree word. If w contains an axis parallel word u such that $\sigma(u) = 3$, then $\sigma(w) \geq 9$.*

Proof. Let us assume that w is a two dimensional squarefree word on $\Sigma = \{a, b, c, d, e, f, g, h\}$. Further let u_1, u_2 and u_3 be immediately adjacent to u (Refer Figure 2.5).

u_3	$\Sigma(u) \cap \Sigma(u_3) = \emptyset$	
u_2	$\lvert\Sigma(u) \cap \Sigma(u_2)\rvert \leq 1$	
u_1	$\Sigma(u) \cap \Sigma(u_1) = \emptyset$	$\Sigma(u_1) = \{d, e, f\}$
u		$\Sigma(u) = \{a, b, c\}$

Figure 2.5: Consequence of having a three letter axis parallel word in w

Let $\Sigma(u)$ be $\{a, b, c\}$. Note that none of the letters from u_1 and at least two letters from u cannot appear in u_2(Refer. Corollary 2.1 and Lemma 2.3). Thus if $\sigma(u_1) \geq 4$, six letters out of the above mentioned letters cannot appear in u_2 leaving behind only two letters for constructing u_2. Clearly, there are no such squarefree words. We may therefore assume that u_1 (u_2 and u_3 as well inductively) contains 3 letters each.

We may assume that d, e and f are the letters appearing in u_1. Corollary 2.1 and Lemma 2.3 forces u_2 to contain letters g and h. By Lemma 2.4 u_3 cannot contain the letters a, b or c. By Corollary 2.1 u_3 cannot contain g or h. Thus u_3 must contain letters d, e and f. Apply Lemma 2.3 on words u_1 and u_3 to get the required contradiction.

□

Lemma 2.6. *Let w be a two dimensional squarefree word on Σ. If w contains an axis parallel word u such that $\sigma(u) \geq 5$, then $\sigma(w) \geq 9$.*

Proof. If an axis parallel word u in the word w uses 5 letters then the adjacent axis parallel words u_1 and u_2 must not use any of these five letters. If $\sigma(w) < 9$, then both u_1 and u_2 must use the letters left out by u. Therefore, By Lemma 2.3, w must contain a square. $\square$

2.6 Modulo two parity and Squarefreeness

Let w be a two dimensional word on an alphabet of size eight. We now show that if w is not a fixed parity word, then w contains a square. Additionally, we will show that the set of letters appearing at positions having a given parity is of cardinality two.

Lemma 2.7. *Let w be a word on an eight letter alphabet and let $n = (2, 2)$. If w fails to satisfy any of the two conditions mentioned below, then w will contain a square.*

1. *For every letter $a \in \Sigma$, $|\pi_n(a, w)| = 1$*
2. *For every $j \in \mathbb{Z}_n$, $|\pi_n^{-1}(j, w)| = 2$*

Proof. By Lemma 2.5 and Lemma 2.6, without any loss of generality, we assume that $\sigma(u) = 4$ for every axis parallel word u in w.

Assume $|\pi_n(a, w)| \geq 2$ for some letter a. Without loss of generality, we can assume that the letter a appears at an even numbered row and an odd numbered row. Note that $\sigma(w) = 8$ and every axis parallel word in w contains exactly four letters. Thus by Corollary 2.1, the set of letters forming adjacent axis parallel words must alternate. Therefore even numbered rows and odd numbered rows cannot contain any common letter contradicting the assumption about a. Thus every letter $a \in \Sigma$, $|\pi_n(a, w)| = 1$.

We define the following sets corresponding to the pseudo inverse parity functions:

$$\begin{aligned} S_0 &\triangleq \pi^{-1}((0,0), w) \\ S_1 &\triangleq \pi^{-1}((0,1), w) \\ S_2 &\triangleq \pi^{-1}((1,0), w) \\ S_3 &\triangleq \pi^{-1}((1,1), w) \end{aligned}$$

Let s_i be defined as $|S_i|$. Condition 2 asserts that each s_i is equal to two.

If any S_i and S_j, where $i \neq j$, have an overlap, then there exists some letter a which doesn't satisfy condition 1. Hence the S_is must form a partition of $\Sigma(w)$ satisfying the following conditions:

$$\begin{aligned} s_0 + s_1 &= 4; \\ s_0 + s_2 &= 4; \\ s_1 + s_3 &= 4; \\ s_2 + s_3 &= 4; \end{aligned}$$

Every s_i is at least 1. Suppose any s_i, say s_0 w.l.o.g, is equal to 1, then s_3 will also be 1. Thus the line word corresponding to $x = y$ will contain a square. Thus every s_i is greater than 1 and thus each s_i is equal to two.

□

2.7 Unavoidability of a square on an 8 letter alphabet in two dimensional words

Lemma 2.7 implies that set of possible letters that could appear at at position i must be equal to S_k for some $k \in 0, \ldots, 3$. Furthermore k depends only on the "parity" of position i. Consider a 3×3 subword u of w. The possibilities of letters that can appear at various positions in u is shown pictorially in Figure 2.6. The value in each "cell" is the set of possible letters that could appear at that position.

A	B	A
C	D	C
A	B	A

Figure 2.6: A, B, C and D must all be distinct and each should be equal one of S_0, S_1, S_2 or S_3

Theorem 2.1. *Every two dimensional word w such that $\sigma(w) < 9$ contains a square.*

Proof. If possible let w be a squarefree word on an eight letter alphabet.Let $A = \{a, \alpha\}$, $B = \{b, \beta\}$, $C = \{g, \gamma\}$ and $D = \{d, \delta\}$. Consider the four corners of 3×3 subword u of w. The letters appearing at these positions is either a or α. There are two possibilities, namely:

1. One of the letters appears at least three times.

2. Both letters appear twice.

We will look at a 3×3 subword of w under the assumptions made and work out the consequences. In particular, we will show that in each case, there will exist a position in the word where none of the letters can occur without producing a square in w.

Consider a word $\omega = x\rho x$ where x is any word and ρ is a letter. Suppose ω has to be extended to a squarefree word, then the letter used to extend must be different from ρ. In the figures that follows, the subwords will be represented using a grid and in each grid position, we will place letters that are already determined. The not yet determined letters will be indicated by blank positions. The oval boxes indicates the word which helps in determining the unique extensions possible. The red letters are the newly determined letters.

In each successive grid, we will place additional letters that gets determined. You may view this process as similar to completing a Sudoku[22].

Case 1: One of the letter appears at least three times

We need to show that when three corners of any 3×3 subword u of w are identical, w invariably contains a square. We may assume suitable letters without any loss of generality for six of the nine possible locations in u. The two (refer Figure 2.7) positions namely the upper middle and the right middle positions could be any letter from the sets B and C respectively. The right middle position in u can be either g or γ. We shall consider both these possibilities (refer Figure 2.8) and show that they lead to w containing a square. These two choices are analyzed in Figures 2.9 and 2.10

		a
g	d	
a	b	a

Figure 2.7: The subword u with vacant positions

		a
g	d	g
a	b	a

		a
g	d	γ
a	b	a

Figure 2.8: The two possible choices for the right column middle position in u

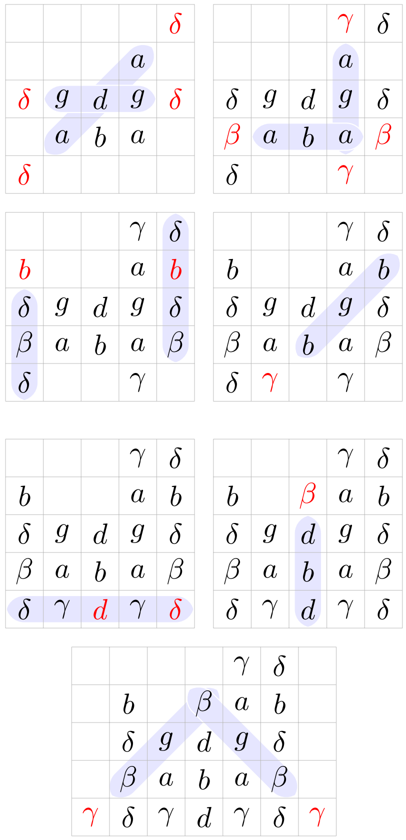


Figure 2.9: Three corners are a and right middle cell is g

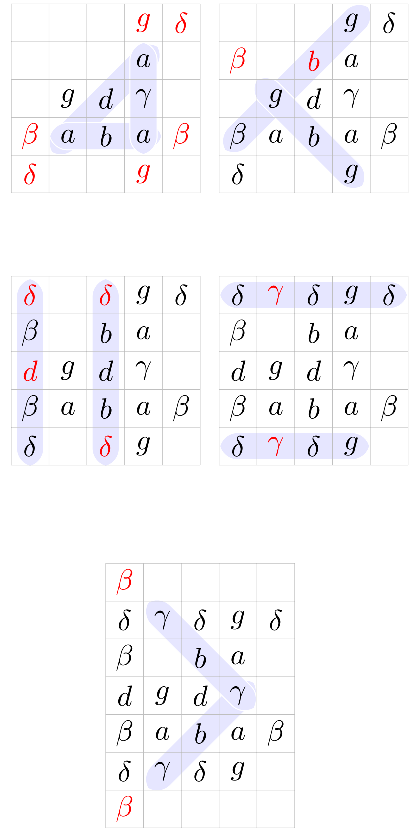


Figure 2.10: Three corners are a and right middle cell is γ

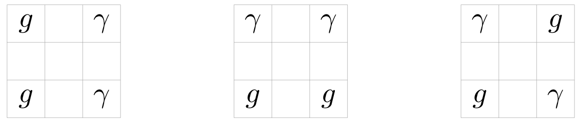
Case 2: Both letters appear twice.

As case 1 has been ruled out, we may now assume that every 3×3 subword u of w has exactly two letters and each of them appears twice. Thus every 3×3 subword is of one of the three types given in Figure 2.11. We will refer to the kind of subword shown in Figure 2.11(c) as a "diagonal subword". To complete the proof we will show the following:

(i) If w does not have a diagonal subword then w contains a square.

(ii) If w has a diagonal subword then w contains a square.

Note that if a 3×3 subword is not a diagonal subword, then letters in the alternate positions in the line perpendicular to the repetition gets fixed. This happens because corners of every 3×3 subword must contain each of the two possible letters exactly twice. This is illustrated in Figure 2.12. The blue oval indicates the repeated letter and the arrow points in the direction perpendicular to the repeated letter. The 3×3 subword indicated by the blue square must have γ in the corners on the right side. The vertical subwords appearing after the left vertical subword can all be inferred inductively.



(a) (b) (c)

Figure 2.11: Two letters appearing two times each

Consider any 3×3 subword u of w. If w does not contain a diagonal subword, we may assume without loss of generality that u is of the type shown as Figure 2.11(a). We may draw inferences about letters appearing in other positions using Figure 2.12. Consider the blue oval in Figure 2.12. Every letter in the column word corresponding to it must contain g at alternate positions. (If γ appears in the column word, then will be a diagonal word at the first such appearance). The remaining positions in the column word uses two letters. Clearly, there are no such long squarefree words.

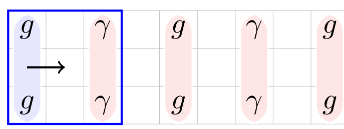


Figure 2.12: The repeated blue g determines the letters in the red positions

Now consider the case in which w contains a diagonal subword (Refer Figure 2.13). The diagonal subword is indicated by the blue square. Without loss of generality, the letter at the center of the square is chosen to be b. This forces the corners of the 5×5 subword to be β. The parity restrictions force the middle letter(indicated in red) in the top and bottom rows to be either b or β. Any of these choices result in w containing an axis parallel subword containing a letter (namely b or β) repeating in three alternate positions. We will complete the proof by showing that if a letter repeats in three alternate positions, then the word contains a square.

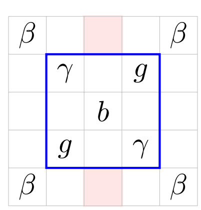
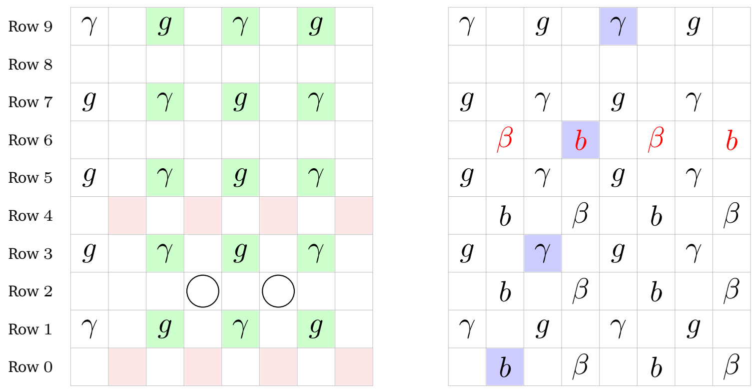


Figure 2.13: The diagonal squares are identical

Consider an axis parallel subword s of w. Clearly a letter cannot repeat in four alternate positions(ref. Figure 2.14) as that forces a square to appear(This is because parity restrictions ensure that the remaining positions are filled using two letters).

Figure 2.14: Letter g repeats in four alternate positions.

We may without loss of generally assume that the letter repeating is g and it repeats alternatively in a vertical word. This analysis is shown in Figure 2.15. As a letter cannot repeat at four alternate positions, we must have the letter γ appearing above and below the repetitions involving g. Moreover, these letters fixes the letters at all the positions indicated in green as every 3×3 subword must contain two letters appearing two times each.



(a) (b)

Figure 2.15: The letter g repeated thrice in alternate positions

Consider the circled positions in Figure 2.15. If the letters appearing at these positions are same, then we will surely have a case of identical letters appearing in the positions marked in red (Refer Figure 2.13). If the circled position is an even(odd) numbered column, then we can conclude that the letters b and β must alternate in even(odd) positions in row two. Fixing the letters at these positions in row two fixes the corresponding letters in rows 0 and 4.

The determined positions after all these inferences in shown in 2.15b. The triple repetition of b and β in the columns fixes the letter row 6 to be β and b (shown in red color) respectively. Notice that the word corresponding to the blue squares, i.e $b\gamma b\gamma$, is a square.

□

2.8 Conclusion and open problems

From Theorem 2.1, we know that $f(2)$, the minimum size of the alphabet on which an infinite two dimensional square free word can be constructed is at least 9. Following Carpi[20], we know that $f(n) \geq 2f(n-1)$. Since $f(2) \geq 9$, we can obtain a small improvement in the multidimensional case, namely $f(n) \geq 9 \times 2^{n-2}$.

The exact value of $f(2)$ is still an open problem. In case of overlap free word it is possible to construct a two dimensional overlap free word on 9 letters [23]. In case of one dimensional word, in order to construct an overlap free word, we need at least 2 letters and in order to, construct a square free word, we need at least 3 letters. The gap between the alphabet size is 1. In case of multidimensional words, this gap between the alphabet size can be explored further. The asymptotic growth of $f(n)$ too is an interesting open problem. Many questions from the usual one dimensional word combinatorics has analogues in multidimensional setting.

Chapter 3

Word Representable Graphs

This chapter contains some results on the theory of word representable graphs. It is an area of research which relates words and graph. The first section introduces the definitions and necessary results. In the next section, we give a brief overview of the various questions addressed in this area. Given a two uniform word w and a graph G, we want to know if $G = G_w$ i.e, does the two uniform word w "represent" G. In Section 3.3, , we give an efficient algorithm for this problem . We also give a formula to count the number of two uniform words which represent a cycle graph. Section 3.4 contains our results on permutation word representability of graphs (a word represents a graph and it also avoids a permutation). In this section we upper bound the length of representant word for "permutation representable graphs" with minimum degree greater than or equal to the permutation length . While studying permutation representable graphs, we have investigated the problem of counting the number of permutation avoiding "representants" of the complete graph for some specific permutations.

In the final Section 3.5 we study graph representability with additional constraints on the representant word. In particular, we place the restriction that the representant word must be uniform and must be permutation avoiding.

3.1 Mathematical preliminaries

Let w be a word on an alphabet Σ. The word $w_{\{a,b\}}$ where $a, b \in \Sigma$, is the word obtained by removing letters other than a and b from w. For example, if $w = abccbacab$ then $w_{\{a,b\}} = abbaab$. If $w_{\{a,b\}}$ is the factor of $(ba)^\omega$(the word obtained by concatenating infinite copies of the word ba), then we say that the letters a and b are alternating in the word w. Otherwise, we say that these letters are *non-alternating*.

Definition 3.1. *Let w be a word on an alphabet Σ. The alternating symbol graph of the word w, denoted by G_w is the graph whose vertex set V and edge set E are defined as below.*

$$V \triangleq \Sigma(w)$$
$$E \triangleq \{(a,b) \mid a,b \text{ are alternating in the word } w\}$$

Example 3.1. *The alternating symbol graph for the word $abdaedcbfc$ is shown in Figure 3.1.*

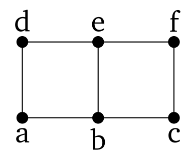


Figure 3.1: The alternating symbol graph for the word $abdaedcbfc$

Definition 3.2. *A graph G is called word representable graph if there exist a word w such that the alternating symbol graph of the word w is isomorphic to G, i.e., $G_w = G$. The word w is called a representant word of the graph G.*

Example 3.2. *The graph given in Figure 3.1 is a word representable graph. The graph given in Figure 3.2 is not a word representable graph. A proof of non representability of this graph can be found in [14].*

Sergey Kitaev introduced the notion of "semi-transitivity" and proved the following theorem which characterizes word representability in terms of semi-transitivity. Based on this theorem, it can be shown that word representability is decidable property. Theorem 3.1 characterizes word representability of a graph in terms of semi-transitivity.

Theorem 3.1 ([24])**.** *A graph is word representable iff it is semi-transitive.*

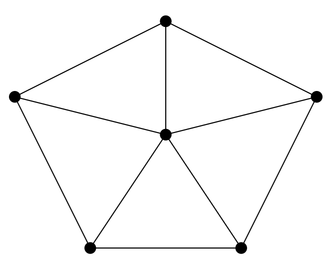


Figure 3.2: The wheel graphs W_5 is a non word representable graph on 6 vertices

Definition 3.3. *Let G be a directed acyclic graph. We say that G is semi-transitive, if for each path $v_1 \to v_2 \to \cdots \to v_k$ such that (v_1, v_k) is an edge, (v_i, v_j) is an edge for every $1 \le i < j \le k$.*

An undirected graph $G = (V, E)$ is *semi-transitive,* if it admits an orientation of the edges so that the resulting directed graph is semi-transitive.

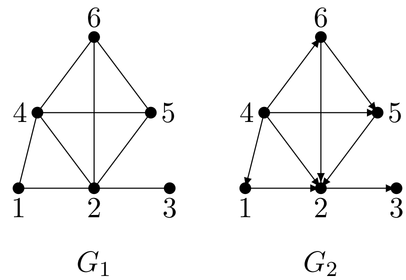


Figure 3.3: A semi-transitive orientation of a graph

Example 3.3. *The graph G_1 shown in Figure 3.3 is semi-transitive as the graph G_2 is a semi-transitive orientation of G_1.*

The number of paths in a directed acyclic graph is finite. Thus one can verify if a given graph is semi-transitive by examining all possible path in every possible directed acyclic orientation of G. It is shown in [14] that this problem is NP-Complete.

3.1.1 Uniform word representability

A word w is called *k-uniform* if every letter in w appears precisely k times in w. A graph G is called *uniform word representable* if there exist a uniform word representing G. A graph G is called *k-word-representable* if there exist a k-uniform word w which represents G.

Example 3.4. *The graph G given in Figure 3.4 is 2-word representable. The word $w = 41321423$ is a 2-uniform word which represents G.*

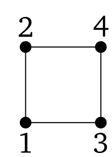


Figure 3.4: A 2-word-representable graph

A word w which represents a graph G can be converted into a word w' such that w' is a uniform word representing G. This can be done by appending a carefully created word to w. For example, the graph represented by the word $\mathrm{abdaedcbfc}$ is also represented by the 2 uniform word $\mathrm{efabdaedcbfc}$. The following theorem summarizes this fact.

Theorem 3.2 ([25]). *Every word representable graph G is uniform word representable.*

The minimum number k for which a given graph is k-word-representable is called the *representation number* of the graph and it is denoted by $\mathcal{R}(G)$. Any permutation $\sigma \in S_n$ represents K_n. Thus $\mathcal{R}(K_n) = 1$. The following theorem gives an upper bound on representation number for an arbitrary word representable graph.

Theorem 3.3 ([24]). *Let $G \neq K_n$ be a word representable graph on n vertices. Then $\mathcal{R}(G) \leq 2(n - \kappa(G))$ where $\kappa(G)$ is the size of the maximum clique in G.*

3.1.2 Ordered patterns

The alphabet we consider will be $\{1, 2, \cdots n\}$ where $n \in \mathbb{N}$. We shall denote this set by **[n]**. Note that there is an underlying natural order on the alphabet **[n]**. A *permutation* is a word on an alphabet **[n]** such that every letter occurs exactly once. The set S_n denotes the set of all permutation on the alphabet **[n]**. Any strict subsequence of permutation is called *subpermutation*. For example, the word 4132 is a permutation while the word 412 is a subpermutation. While studying permutation or subpermutation avoidance we shall always assume that the alphabet of the word under consideration is of the form **[n]** for some $n \in \mathbb{N}$ and the largest letter that appears in the permutation or subpermutation is less than or equal to n. Permutation avoidance in words is a well-explored research topic in the combinatorics of permutations [19, 26].

3.1.3 Ordered pattern avoidance

In the earlier chapters, we looked at the problem of avoiding patterns over an unordered alphabet. We introduce three type of pattern avoidance over an ordered alphabet in this section, namely permutation avoidance, avoiding a set of permutation and subpermutation avoidance. We will consider a word representability of graphs wherein we will stipulate that the representant word avoids permutation patterns of the kind introduced here.

We motivate the concept of permutation avoidance by providing a few examples. Consider the following sequence of 10 numbers $\{10, 30, 42, 20, 18, 36, 5, 38, 15, 49\}$. The sequence contains an increasing subsequnence of length 5 namely $\{11, 20, 36, 38, 49\}$. This sequence shares the property of having the same "order" as the permutation pattern $\{1, 2, 3, 4, 5\}$. In other word we will say that the sequence contains the permutation pattern $\{1, 2, 3, 4, 5\}$. An another example, the subsequence $\{42, 36, 5, 49\}$ mimics the permutation $\{3, 2, 1, 4\}$ and therefore we cansa that thesequence contains the permutation pattern $\{3, 2, 1, 4\}$. The formal definition is provided below.

Definition 3.4 (*Permutation avoidance*). *Let $w \in$ **[n]*** and $p = p_1 p_2 \ldots p_k$ be a permutation. We say that the word w contains the permutation p if there exist indices $1 \leq t_1 < t_2 < \ldots <$*

$t_k \leq n$ in the word w such that if $p_i > p_j$ then $w_{t_i} > w_{t_j}$ for all $i, j \in [k]$. In other words, we can find a subsequence t of length k in w such that the ordering of letters in t mimics the ordering given by p. The word $w_{t_1} w_{t_2} \dots w_{t_k}$ is called an instance of p. A word w avoids a permutation p, if w does not contain any instance of p.

Example 3.5. *Consider the word $w = 3721412$. The subsequence 374 is an instance of the permutation 132 because the ordering of the letters at any two indices i and j where $1 \leq i, j \leq 3$ in side the permutation and inside the subsequence are same. The subsequence 142 is also an instance of the permutation 132. The word 654321 does not contain an instance of the permutation 132 because it is in strictly decreasing order.*

Definition 3.5 (*Set permutation avoidance*)**.** *Let A be a set of permutations. We say that a word w avoids A iff w avoids every permutation in A.*

Example 3.6 ([27])**.** *Let A be the set containing the permutations $1234 \cdots n$ and $n(n-1) \cdots 321$ where $n \in \mathbb{N}$, i.e., the increasing and decreasing subsequences of length n. The longest word which avoids the set A is of length n^2.*

Definition 3.6 (*Subpermutation avoidance*)**.** *Let s be a subpermutation on $[n]$ containing n. Let A_s be the set as defined below.*

$$A_s \triangleq \{p \in S_n \mid s \text{ is a subsequence of } p\}$$

We say that a word w avoids the subpermutation s iff w avoids A_s.

Example 3.7. *The word $w = 78563412$ avoids the subpermutation 13. The word w is obtained by interleaving two decreasing sequences. There are no increasing sequences starting at an even number and the increasing sequences starting at the odd number is of length at most 2. None of these can be an instance of the subpermutation 13.*

Note that for a given subpermutation s, the word which avoids A_s need not avoid s if s itself a permutation. For example, if $s = 132$ and the set $A_{132} = \{4132, 1432, 1342, 1324\}$ then there exist words which avoids A_{132}, but does not avoid 132.

The different notions described above for pattern avoidance give rise to different notions of word representability of graphs. We shall explore more about these notions in the coming sections.

3.1.4 Representability and permutation patterns

For a property P of words, we can look at the problem of representing a given graph G using word having property P. We call such graph P-representable. The properties we have study in this thesis are

1. Uniformity

2. Ordered permutation avoidability

For a permutation p, we shall say that a graph G is *p-representable*, if there exist a word w which represents G and avoids permutation p. Given a $k \in \mathbb{N}$, a permutation p and a graph G, we say that G is *(k, p)-representable* if there exists a p avoiding k uniform word w which represents G.

Example 3.8. *The graph G shown below is 132 representable. The word $w = 43212341$ avoids the permutation 132 and represents G.*

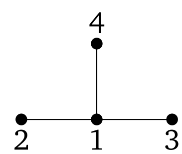


Figure 3.5: A 132-representable graph

All connected graphs on 5 vertices are 132-representable [28]. To find a word representable graph on 6 vertices which is not 132 representable is an open question[28].

Example 3.9. *The cycle graph C_4 in Figure 3.4 is not 12-representable. The graph given in Figure 3.5 is $(2, 132)$-representable because the word $w = 43212341$ represents the given graph and is a 2-uniform word avoiding the permutation 132.*

Example 3.10. *The graph given in Example 3.1 is not $(2, 132)$-representable. We shall provide a proof for this result in Section 3.5.*

3.2 Problem statement and known results

The problems studied in this chapter are related with the various types of graph representability. If a graph is word representable, then it has infinitely many representant words. Given a graph G, let $f_n(G)$ denote the number of representant word of G of length n. We study this function $f_n(G)$ and have computed $f_n(G)$ for some specific graphs. Note that $f_n(G) = m!$ for the complete graph K_m and $f_n(G) = 0$ for the graphs G which are not word representable.

We show that a cycle graph C_n is represented by a unique circular permutation of a 2-uniform word. This characterization helps us to count the number of 2-uniform words which represents C_n. We show that the number of 2-uniform words which represents C_n is $4n$.

Problem 1 (Word generated graph). *Given a word w, compute the graph G_w.*

The brute force algorithm can solve the problem in $\mathcal{O}(|\Sigma(w)|^2 + |\Sigma(w)| \times |w|)$-time and $\mathcal{O}(|\Sigma(w)|^2)$-space. Designing an efficient algorithm for this question is an open problem.

Problem 2 (Decidability of graph representability). *Given a graph G, is G word representable?*

This problem is known to be NP-Complete [14]. The associated counting problem is to compute the number of words of length n which represents G. There are no known complexity related literature corresponding to this counting problem. We look three variants of these problems.

Problem 2.1 (k-word representability). *Given a graph G, is G k-word representable?*

Problem 2.2 (p-representability). *Given a graph G and permutation p, is G p-representable?*

Problem 2.3 ((k, p)-representability). *Given a graph G, a number k and a permutation p, is G (k, p)-representable?*

For a given k where $3 \leq k \leq \lceil \frac{n}{2} \rceil$, Problem 2.1 is NP-Complete [14]. Problems 2.2 and 2.3 are decidable because the length of the representant word is bounded. For a given $k = 2$ uniform word w and a given $G = (V, E)$, we give an $\mathcal{O}(V + E)$-time algorithm to decide whether $G_w = G$.

It is known that cycle graphs, path graphs and trees are 132 representable[28]. Gao, Kitave and Zhang obtained a count for the number of 132 avoiding representants of the complete graph[28]. We show that the number of 1342 avoiding words which represent the complete graph K_n is 6 when the letter n appear three times in the representant word.

We give an example of a six vertex graph which is 132-representable but it is not $(2, 132)$-representable. Further, we show that grid graphs and ladder graphs are not $(2, 132)$-representable. The $(2, 132)$ representable graphs form subset of $(2, 1342)$ representable graphs because the word which avoids the permutation 132 avoids the permutation 1342. We show that the inclusion is strict by constructing a graph which is $(2, 1342)$-representable but not $(2, 132)$-representable.

3.3 Uniform word representability

For $3 \leq k \leq \lceil \frac{n}{2} \rceil$, deciding that whether the given graph is k-word representable, is NP-Complete. This section studies graphs with representation number 2. The representation number of cycle graphs is two. We obtain a count for the number of words which represents cycle graphs. We give a linear time algorithm to check whether a given two uniform word represents a given graph in Section 3.3.2.

3.3.1 Number of 2 uniform representant words for cycle graph

We look at the two uniform representants of the cycle graph. The following theorems tells us that we can restrict our attention to a single canonical two uniform representant word.

Theorem 3.4. *In any 2 uniform representable graph $G = (V, E)$, the representant word w and its circular shift, denoted by C_w both represent the same graph.[14]*

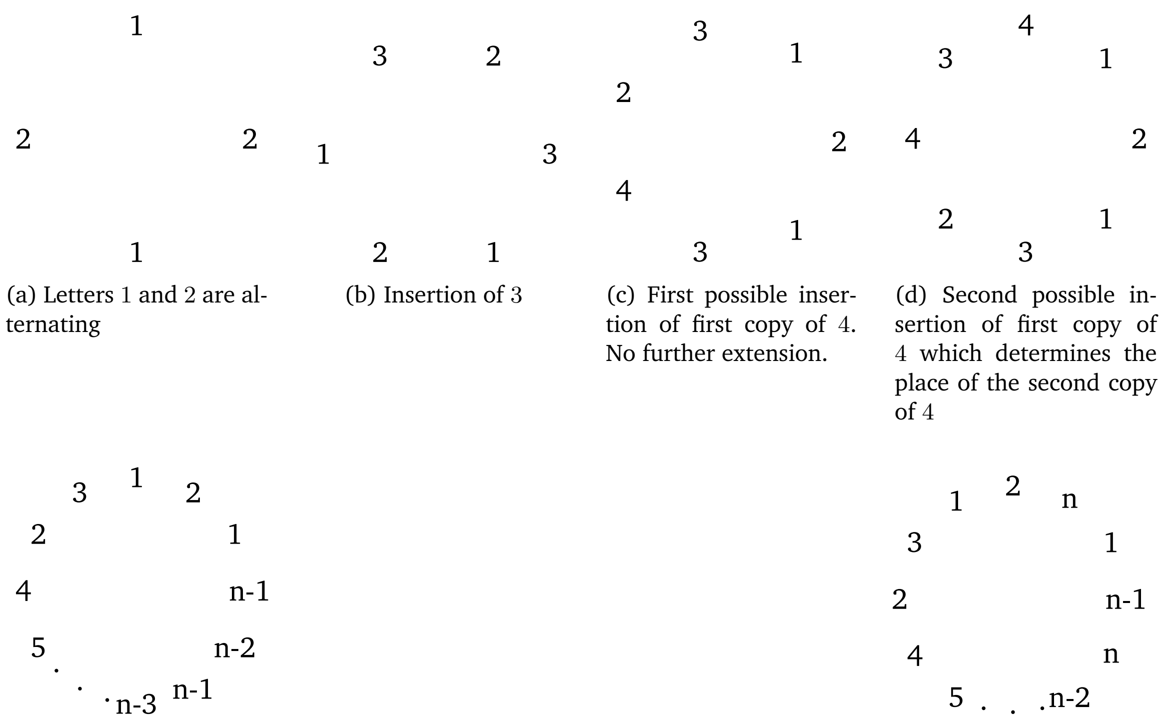
Given a representant word w, the above proposition guarantees any circular shift of w also represents the cycle graph. We show that there is precisely one circular permutation of a two uniform word which represents a cycle graph.

Theorem 3.5. *There is a unique circular permutation of n letters where each letter appears precisely twice and represents cycle graph C_n.*

Proof. Consider any two uniform word w representing C_n. Clearly all its cyclic shifts also represent C_n. We can thus consider the word w as being placed along the perimeter of a circle. Note that if the chord obtained by joining two copies of the letter i and the chord obtained by joining two copies of the letter j intersect in the circle then there is an edge in the graph which corresponds to the vertices i and j.

Since the vertex 1 and the vertex 2 are connected, the letter 1 and the letter 2 have to alternate in the representant word of the graph. These two letters must be put on the circle which is shown in Figure 3.6(a). The letter 3 is connected with the letter 2 and it is not connected with 1 hence 2 has to come in between the two copies of letter 3. This is shown in Figure 3.6(b). The letter 4 is connected with 3 hence precisely one copy of 4 must come in between two places within the arc determined by the word 323. If the first copy of the letter 4 occurs as 3423 then the second copy of 4 cannot occur anywhere in the arc determined by the word 31213. Otherwise, 4 is connected with either of letters 1 or 2. So 4 can not appear any place around the circle. Hence, the first copy of letter 4 must occur as 3243. The second copy of 4 cannot appear anywhere in the arc determined by 234121. Otherwise, 4 is connected with the letters either 1 or 2. Hence, the second copy of letter 4 has to appear as

324341. It is shown in Figure 3.6(c). Since we are getting a unique extension for each letter, this procedure can be extended up to the letter $n-1$. We get Figure 3.6(e). The letter n is connected with the letters $n-1$ and 1. The letter n must appear in the arcs determined by 121 and $(n-1)(n-2)(n-1)$. If the first copy of the letter n appear as $1n21$ and the second copy of the letter n appear as $(n-1)n(n-2)(n-1)$ or $(n-1)(n-2)n(n-1)$ then the vertex n is connected with the vertex 2, a contradiction. If the first copy of the letter n appear as $12n1$ and the second copy of the letter n appear as $(n-1)(n-2)n(n-1)$ then the vertex n is connected with the vertex $n-2$, a contradiction. Hence, n has to appear as $12n1$ and $(n-1)n(n-2)(n-1)$. The final figure is shown in Figure 3.6(f).



(a) Letters 1 and 2 are alternating

(b) Insertion of 3

(c) First possible insertion of first copy of 4. No further extension.

(d) Second possible insertion of first copy of 4 which determines the place of the second copy of 4

(e) Insertion of $n-1$

(f) Insertion of n

Figure 3.6: Representant word for C_n

□

The above characterization of the cycle graph by two uniform word helps us to count the number of two uniform words which represents a cycle graph.

Theorem 3.6. *For $n > 3$, the number of 2 uniform word which represents the cycle graph C_n is $4n$. For n equal to 3, 2 and 1 these numbers are 6, 4 and 1 respectively.*

Proof. The cases where $n \leq 3$ can be checked by a simple enumeration. We shall therefore assume that n is greater than 3. Let w (as shown in Figure 3.6(f)) be a word representing

C_n. No factor of w of length 3 can have a repeated letter as a repeated letter in a factor with length 3 will force the corresponding vertex to have degree less than 2. As there are more than 3 letters in w, if any 3 length factor abc has repeated in w, we can infer that G_w is not C_n, as G_w is either disconnected or there exist a vertex in G_w of degree 3. If the factors abc and cba have appeared in w then either there exist a vertex which has degree 3 or the graph is disconnected. Both possibilities give a contradiction. The number of words of size 3 on alphabet [n] is equal to $n(n-1)(n-2)$. The total number of circular shift of the word w is $2n$. Each circular shift gives two words (the word and its reflection) which represents the graph C_n. Since $n(n-1)(n-2) \geq 4n$ for $n > 3$, any circular shift of w and its reflection they are all unique because each word has the distinct prefix of length 3. Hence, the number of two uniform words which represents C_n is $4n$.

□

3.3.2 Algorithm for 2-word representability

Given a graph $G = (V, E)$ and a two uniform word $w \in V^*$, we want to check if $G = G_w$. The naive algorithm (Algorithm 1) to check if $G = G_w$ takes $\mathcal{O}(V^3)$ time. The naive algorithm checks for each pair (a, b) whether they are present in G and G_w. In this section we provide an optimal algorithm (Algorithm 2) which works in $\mathcal{O}(V + E)$ time.

Algorithm 1 The naive algorithm to check if $G = G_w$

Input: A graph G and a two uniform word w.
Output: TRUE if $G = G_w$ and FALSE if $G \neq G_w$.

procedure ALTERNATING GRAPH(G, w)
  $V \leftarrow$ the vertex set of G.
  $E \leftarrow$ the edge set of G.
  for all $a, b \in V \times V, a \neq b$ **do**
    if $(a, b) \in E$ and a, b doesn't alternate in w **then**
      return FALSE
    end if
    if $(a, b) \notin E$ and a, b alternate in w **then**
      return FALSE
    end if
  end for
  return TRUE
end procedure

For the optimal algorithm, we store the two uniform word as a doubly linked list. Note that every letter appears exactly twice.

In order to determine if $G = G_w$, we first check if every edge of G is an edge in G_w. Every

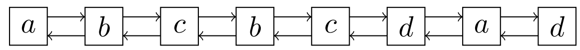


Figure 3.7: The word $abcbcdad$ stored as a linked list

letter appears twice in w. Let f_a and s_a be the indices corresponding to the first and the second occurrence respectively of a letter a. the By a linear scan of w, we can determine f_a and s_a, for every letter a. Note that (a, b) is an edge in G_w if and only if $f_a < f_b < s_a < s_b$ or $f_b < f_a < s_b < s_a$. Once the preprocessing of w is done to determine f_a and s_a, determining if an edge of G is an edge of G_w can be done in constant time per edge. Therefore, the total time taken will be linear in $|w|$ and $|E|$.

Note that $G = G_w$ if and only if every edge of G is an edge of G_w and both graphs have equal number of edges. The number of edges of G is readily known. In Algorithm 2, we compute the number of edges in G_w in linear time by computing the degree (The algorithm actually computes the residual degree) of each vertex.

The degree of vertex v in G_w is equal to the number of singleton occurrences of letters between f_v and s_v. The sum of the degrees is equal to twice the number of edges. Instead of counting the singleton occurrences, if the count only those single occurrences that corresponds to the first occurrence of a letter, we will get a number, we call it residual degree, that is less than or equal to the degree. Note that each singleton occurrence will now be counted exactly once and thus the sum of the residual degree will be equal to the number of edges in G_w.

Theorem 3.7. *Given a two uniform word w and a graph G, Algorithm 2 correctly solves the problem of checking if $G = G_w$ and runs in $\mathcal{O}(V + E)$ time.*

Proof. The algorithm verifies the following two conditions;

1. Every edge of G is an edge of G_w
2. Number of edges is G is equal to number of edges in G_w.

Clearly, these two conditions imply that $G = G_w$. The algorithm has three parts; the first part consisting of lines up to 6 is the initialization stage. The lines $7-11$ checks for condition 1 and the lines $12 - 23$ checks for condition 2.

If an edge (a, b) of G is not an edge of G_w, then line number 9 will return FALSE when the edge (a, b) is processed by the **for** loop. Thus, when the algorithm completes the execution of the **for** loop without returning, every edge of G has been verified to be an edge of G_w and thus verifies condition 1.

Algorithm 2 The linear time algorithm to check if $G = G_w$

Input: A graph G and a two uniform word w.
Output: TRUE if $G = G_w$ and FALSE if $G \neq G_w$.

```
procedure ALTERNATING GRAPH(G, w)
    V ← the vertex set of G.
    E ← the edge set of G.
    m ← the number of edges in E.
    m̂ ← 0
    D_w ← the doubly linked list corresponding to the word w.
    for all edge (a, b) ∈ E do
        if a and b do not alternate in w then
            return FALSE
        end if
    end for
    while D_w is not empty do
        x ← The first repeating letter in D_w.
        C_x ← The number of letters between the two occurrences of x.
        m̂ ← m̂ + C_x
        if m̂ > m then
            return FALSE
        end if
        Delete the occurrences of x from D_w.
    end while
    if m̂ < m then
        return FALSE
    end if
    return TRUE
end procedure
```

We will show that **while** loop computes the number of edges of G_w if it is less than or equal to the number of edges in G and returns FALSE otherwise. When the loop does not return FALSE, the variable $\widehat{m}$ will contain the number of edges in G_w.

Since w is a two uniform word, every letter appears exactly twice in w. Let the letters be ordered by the indices of their second appearances. Let us denote the r^{th} repeated letter by l_r. For example, for the word shown in Figure 3.7, $l_1 = b, l_2 = c, l_3 = a$ and $l_4 = a$. For $i \geq 1$, let G_w^i be the induced subgraph of G_w by restricting the vertex set to $V \setminus \{l_1, \ldots, l_i\}$ and let $G_w^0 = G_w$. Let D_w^0 be the doubly linked list corresponding to w and let D_w^i be the doubly linked list after i iterations of the **while** loop. Note that in the i^{th} iteration, the two copies of l_i are removed from D_w^{i-1} to obtained D_w^i and the number of letters between the two occurrences, which we shall denote by $\widehat{\delta_i}$, is added to $\widehat{m}$. Let δ_i denote the degree of the vertex l_i in the graph G_w^{i-1}. We claim that δ_i is equal to $\widehat{\delta_i}$.

Claim: $\delta_i = \widehat{\delta_i}$.

Proof of Claim:

Let $w^{(i)}$ be the word corresponding to the doubly linked list D_w^i. Clearly, the alternating word graph $G_{w^{(i)}}$ is equal to G_w^i. Thus the δ_i, the degree of l_i in G_w^{i-1}, is equal to the degree of l_i in the alternating word graph $G_{w^{(i-1)}}$. Since l_i is the first repeated letter in D_{i-1} and thus in $w^{(i-1)}$, every letter that appears in between the two appearances of l_i is a neighbor of l_i in $G_{w^{(i-1)}}$ and thus $\delta_i = \widehat{\delta_i}$.

End of Proof of Claim

For a vertex l_a, the edge $(l_a, l_b) \in G_w$ is in G_w^a if and only if $b > a$. Only the edges in G_w^a contribute towards δ_a. Therefore each each in $(l_a, l_b) \in G_w$ contributes a one to either to δ_a or to δ_b. Thus $\widehat{m}$ is always less than or equal to the number of edges in G_w. None of the steps in the algorithm decrements the value of $\widehat{m}$. Therefore, when $\widehat{m}$ becomes larger than the number of edges in G, we can rightly conclude that $G \neq G_w$. If the **while** loop exits without a returning FALSE in between then clearly $\widehat{m}$ contains the number of edges in G_w which is guaranteed to the less than or equal to the number of edges in G. Therefore the check in line number 21 correctly verifies condition 2.

Time Complexity: The initialization phase consisting of steps 1-6 clearly takes time proportional to $\mathcal{O}(V + E)$. The steps 7-11 takes $\mathcal{O}(E)$ as each check in line 8 can be done in constant time if we preprocess w and store the first and second occurrences of each letter $a \in \Sigma(w)$. The time taken in the while loop (steps 12-21) is proportional to the value of $\widehat{m}$. As the value of $\widehat{m}$ is bounded by $m + |V|$, we can bound the time taken by steps 12-21 by $\mathcal{O}(V + E)$. Therefore the algorithm runs in $\mathcal{O}(V + E)$ time. $\square$

3.4 Permutation representability of Graphs

The problem we address in this section is related to the word representability of K_n. It is easy to see that any representant word of K_n is a prefix of the word σ^ω where σ is a permutation on [n] where the prefix is of length at least n. We are interested in counting the number of such representants which avoid the permutation 1342. Given word w, let $n_x(w)$ denotes the number of occurrences of x in w. The following result connects $n_x(w)$ and the minimum degree of a permutation representable graph. Note that all 132-representable graphs are 1342-representable graphs because the word which avoids 132 cannot contain 1342.

Theorem 3.8. *Let p be a permutation of length at most $\delta + 1$ and let G be a p-representable graph such that the degree of each vertex is at least δ. For any p avoiding representant word w of G and any letter $x \in \Sigma$ we have $n_x(w) \leq \delta$.*

Proof. Let x be a vertex of G and w be any representant word of G. For the sake of contradiction, let assume that $n_x(w) > \delta$. Since the degree of x is greater than or equal to δ, the vertex x has at least δ adjacent vertices. Assume that a_1, a_2,... and a_δ are distinct neighbors of x in G. The word w can be written as $w_1xw_2xw_3xw_4\ldots w_{\delta+1}xw_{\delta+2}$. Since the letters a_1,a_2... a_δ are adjacent vertices of x, the letters a_1, a_2 ... a_δ must appear in words w_2,w_3 ... $w_{\delta+1}$. We shall show that p appears in w. Since, $T = \{x, a_1, a_2 \ldots a_\delta\} \subset \{x\} \cup \Sigma(w_i)$ where $1 \leq i \leq k$, each of these non overlapping substring of w contain every letter in T. Thus any permutation of length $\delta + 1$ can be obtained by picking the appropriate letter from T in xw_i. In particular, an instance of p can be obtained in w.

□

Corollary 3.1. *The maximum length of any word which avoids permutation p and it represents a graph in which each vertices has the degree at least $|p| - 1$ is $(|p| - 1)n$ where the letter n is the number of vertices in the graph.*

Proof. Let w be a word which avoids permutation p, and it represents a graph in which degree of each vertices is at least $|p| - 1$. By Theorem 3.8, each letter can appear at most $|p| - 1$ time in w. Hence, $|w| \leq (|p| - 1)n$. □

In the following section, we obtain a count for the number 1342 representant words of K_n.

3.4.1 Count of special 1342 representant words of K_n

To count the number of 1342 avoiding permutation which represents K_n, we first count the number of permutations avoiding subpermutations 13 and 342.

Proposition 3.1. *The number of permutations on [n] which avoids the subpermutation* 342 *is* nC_{n-1} *where* C_{n-1} *is Catalan number.*

Proof. If a word on the alphabet [n] $\setminus \{1\}$ where each of the letter appears precisely once, avoids subpermutation 342 then by reducing each letter of permutation by 1, we get a unique permutation on alphabet [n-1] which avoids 231. Now, if a permutation on alphabet [n-1] avoids permutation 231 then by increasing each element of permutation by 1, we get a unique 342 avoiding word on alphabet [n] $\setminus \{1\}$ where each letter appears precisely ones. Hence, the number of words on alphabet [n] $\setminus \{1\}$ where each letter appears precisely once and avoids set A_{342} is C_{n-1}. For each 342 avoiding word $x \in$ [n] $\setminus \{1\}$ where each letter appears exactly once, we can insert 1 at n positions. The obtained permutations still avoid set A_{342} because x avoids set A_{342}. Hence, for each such x, we get the permutation on [n] which avoids the set A_{342}. So the number of permutation on [n] which avoids set A_{342} is equal to nC_{n-1}. □

The following Proposition will be used to prove the main result of this section.

Proposition 3.2. *Let* $\Gamma_n(A) = \{x \in S_n : x$ *avoids the subpermutations* 13 *and* $342\}$. *Then , for all* $n \in \mathbb{N}$ *where* $n \geq 4$, $\Gamma_n(A)$ *has exactly three elements namely* $n(n-1)\dots 4231, n(n-1)\dots 4312$ *and* $n(n-1)\dots 4321\}$.

Proof. We have $A_{13} = \{123, 132, 213\}$ and $A_{342} = \{1342, 3142, 3412, 3421\}$. To avoid the set A_{13}, any permutation w, of the set $\Gamma_n(A)$ must be of the following forms.

1. All the letters which appear before n must be greater than the letters which appear after n. Otherwise, the permutation contains 132.

2. The letter n can appear either at the first or the second position. Otherwise, the permutation contains either 312 or 321.

Further, if the letter n appears in the second position in any permutation in $\Gamma_n(A)$ then there must be no increasing or decreasing sequence of length 2 after n. Otherwise, the permutation contains a permutation in the set $\{3412, 3421\}$. Thus, if the letter n where $n \geq 4$ appear at the second position in a permutation in $\Gamma_n(A)$ then the only one letter can appear after n. So we conclude that for $n \geq 4$, the letter n cannot appear at the second position in any permutation in $\Gamma_n(A)$. Since the letter n appears only at first position in every permutation in $\Gamma_n(A)$ for all $n \geq 4$, any permutation of the set $\{1342, 3142\}$ cannot occur in any permutation in $\Gamma_n(A)$. So for each element in $\Gamma_n(A)$ we get an exactly one element in $\Gamma_{n+1}(A)$ by appending $n+1$ at the first position. Thus, $|\Gamma_n(A)| = |\Gamma_{n+1}(A)|$. For $n = 3$, we have $\Gamma_3(A) = \{231, 312, 321\}$. Hence, $|\Gamma_n(A)| = 3$ for all $n \geq 3$. From the above analysis, the set $\Gamma_n(A)$ can be generated easily from the set $\Gamma_3(A) = \{231, 312, 321\}$. □

Theorem 3.9. *Let K_n be the n vertex complete graph and S be the set of words which represents K_n and avoids 1342. Let $T \subset S$ be the set of words in which n appear three times. Then $|T| = 6$.*

Proof. A word in which n appears precisely three times, and it represents complete graph must be of the form $w = w_1nw_2nw_3nw_4$ where $w_2, w_3 \in S_{n-1}$, and letters can not repeat inside w_1 and w_4. Not that to avoid 1342 in $w_1nw_2nw_3nw_4$, $\Sigma(w_1)$ must be a subset of $\{n-1, n-2\}$ and $\Sigma(w_4)$ must be the subset of $\{n-1, 1\}$.

Now the factor nw_2nw_3n of w, if the factor w_2 contains the subpermutation 13 or the factor w_3 contains the subpermutation 342 then the word w contains the permutation 1342. To ensure that each pair of the letters alternate in the factor nw_2nw_3n, in between two letter i where $1 \leq i \leq n-1$, all letters from the set [n] have to appear. Since every pair of the letter must alternate in the factor, we must have $w_2 = w_3$. By Proposition 3.2, the word w_2 must belong to the set $\Gamma(n-1)$. If $w_2 \in \Gamma(n-1)$ then $w_1 = \epsilon$ and $w_4 = n-1$ or $w_4 = \epsilon$. Hence, the total count becomes 6. □

3.5 Uniform permutation representability

In this section, we study the problem of representing graphs using uniform words which avoid certain permutation. In particular, we show that certain class of graphs can not be represented via uniform permutation avoiding words. The two permutation representability which we study are

(i) 132 representability

(ii) 1342 representability

3.5.1 Non $(2, 132)$-representability

Consider the graph G shown in Figure 3.8. The word $abdaedcbfcef$ represents this graph, is a two uniform word. The total number of 2-uniform words on [6] is $\frac{12!}{2^6}$. We have generated all the 132 avoiding words among these. For each of these words w, we have generated the graph G_w and verified that at least one of the following condition fails.

1 The degree sequence of the graph is $(3, 3, 2, 2, 2, 2)$

2 The vertices with degree 3 have distant neighbors.

3 The vertices with degree 3 are connected.

Since, the graph in Figure 3.8 satisfies all the above condition, we can conclude that none of the word generated can represent the graph in Figure 3.8. We summarize this as theorem.

Theorem 3.10. *The graph given in Figure 3.8 is not* $(2, 132)$ *representable.*

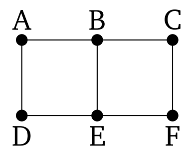


Figure 3.8: A non $(2, 132)$-representable graph

Corollary 3.2. *Ladder graphs and grid graphs are not* $(2, 132)$*-representable.*

Proof. These graphs contain Figure 3.8 as an induced sub graph, which is not $(2, 132)$-representable. Thus, the Ladder and the Grid graph are not $(2, 132)$-representable. □

Remark 3.1. *Graph given in Figure 3.8 is represented by the* 132 *avoiding word* 564534261*. If we remove any edge from the graph then we get a* $(2, 132)$*- representable graph. Their representant words are* 564534231261*,*564345236121 *and* 645342351261.

3.5.2 On $(2, 1342)$-representability

The graph which are $(2, 132)$-representable are surely $(2, 1342)$-representable. We show that the converse is not true. In this section, we shall construct a family of connected graphs that are $(2, 1342)$-representable graph. These graphs contain the graph in Figure 3.8 as a sub graph, and thus none of these graphs will be $(2, 132)$-representable graph.

Let $G_1 = (V_1, E_1)$ be the graph given in Figure 3.8 with the vertices labeled by $A = 4, B = 3, C = 1, D = 5, E = 6$ and $F = 2$. Let G_n where $n \in \mathbb{N}$ be the defined inductively as follows.

$G_{n+1} = (V_{n+1}, E_{n+1})$ and $A_n = \{6n+1, 6n+2, \cdots, 6n+6\}$ where

$$V_{n+1} = V_n \cup A_n$$

$$E_{n+1} = E_n \cup \{(a,b) \mid a, b \in A_n \text{ and } (a-6n, b-6n) \in G_1\} \cup S$$

and $S = \{(6n, 6n+1), (6n-1, 6n+3), (6n, 6n+3), (6n-1, 6n+1)\}$. We obtain the following theorem about the graph G_i where $i \in \mathbb{N}$. Th graph G_3 has shown in Figure 3.9(b).

Theorem 3.11. *For every $n \in \mathbb{N}$, the graphs G_n defined above are $(2, 1342)$-representable but not 132-representable.*

Proof. As each G_n where $n \in \mathbb{N}$ contain G_1 as a induced sub graph G_n is not 132 representable for any $n \in \mathbb{N}$. For any word w over $\mathbb{N}$ and for $n \in \mathbb{N}$, let $w \oplus n$ denote the word whose i^{th} letter is $w_i + n$ where w_i denotes the i^{th} letter of w.

Let $w_1 = 564534261231$. We define the word $w_n \triangleq \gamma_n\beta_1\beta_2\alpha_1\alpha_2\tau_n$ where γ_n, τ_n are unique words and α_1, α_2, β_1 and β_2 are the unique letters which satisfies the following conditions.

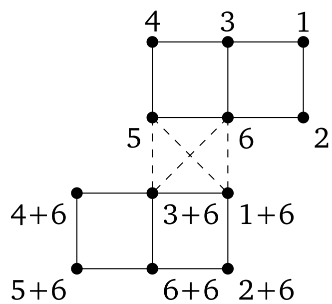
1. $\gamma_n\alpha_1\alpha_2 = w_1 \oplus 6(n-1)$
2. $\beta_1\beta_2\tau_n = w_{n-1}$

It is easy to check that w_n is a two uniform word which represents G_n. We shall verify that w_n avoids the permutation 1342. Clearly w_1 avoids 1342. Assume that w_n contain 1342 and is the last such n.

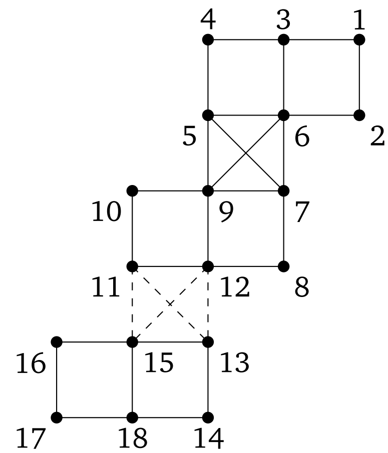
As $\gamma_n\alpha_1\alpha_2$ is a word obtained by adding a number to w_1, $\gamma_n\alpha_1\alpha_2$ clearly avoids 1342, $\beta_1\beta_2\tau_n$ being equal to w_{n-1}, we can inductively conclude that $\beta_1\beta_2\tau_n$ too avoids 1342. The word $\beta_1\beta_2\alpha_1\alpha_2$ is $(6n-1)(6n)(6n+3)(6n+1)$ and it clearly avoids 1342. Thus any occurrence of 1342 in w_n must span across the sub words γ_n, τ_n and $\beta_1\beta_2\alpha_1\alpha_2$.

Since, $\gamma_n\alpha_1\alpha_2$ contain letters which are greater than every letter in $\beta_1\beta_2\tau_n$, the smallest letter of an instance of the permutation 1342 in w_n must be a letter among $\beta_1\beta_2\tau_n$. Otherwise the instance of the permutation 1342 appears completely inside $\gamma_n\alpha_1\alpha_2$. Assume that the smallest letter of an instance of the permutation 1342 is β_1 or β_2, then it is not possible to get the third, fourth and the second smallest letter of the instance of the permutation 1342 in $\beta_1\beta_2\alpha_1\alpha_2\tau_n$ because all the letters which appear in side q_n are less than or equal to β_1 or β_2 and $\alpha_1 > \alpha_2$.

□



(a) The graph G_2



(b) The graph G_3

Figure 3.9: The graphs which is represented by a 2-uniform word which avoids permutation 1342, but it does not avoids permutation 132.

3.6 Conclusion and open problems

For a given two uniform word w and a graph G, we have given a fast algorithm to check that whether $G_w = G$. We studied the number of two uniform representant words for the cycle graph on n vertices and have shown that there are precisely $4n$ two uniform representant words. We have extended the study of 132 and 1342 permutational representable graphs and obtained the count of words which represents a complete graph of n vertices in a special case. We have also constructed a graph which $(2, 1342)$ representable but it is not 132 representable.

Finding a graph on 6 vertices other than complete graph which is not 132 representable is not known. Characterization of permutation p representable graphs which are not $(2, p)$ representable is an open question. There are various open question in this area have mentioned in [14].

Chapter 4

Quasiperiodicity in Tribonacci Word

Quasiperiodicity is the generalization of the well studied concept periodicity[17, 15]. The concept of quasiperiodicity is defined in [29]. A word w is "periodic" if it can be written as $w = u^k u'$ where k is a strictly positive integer and u' is prefix of u. The word u can "cover" every position of w in a non overlapping manner. A word w is "quasiperiodic" if there exist a word u such that w can be "covered" using u. Note that we do not insist that the "covering" is non overlapping. We call the length of the the covering words as period and quasiperiod as applicable. Note that every period is a quasiperiod and a word can have more than one period.

Example 4.1. *The word $w = abaababaabaaba$ is quasiperiodic as the word aba covers it. Note that w cannot be written as a prefix of $(aba)^k$.*

In this chapter we study the various "covering words" of the Tribonacci word. In particular we look of the following covering words

- Cover
- Seed
- Right Seed
- Left Seed

The notions are diagrammatically shown in Figure 4.1.

Given a word w, an algorithm to compute all the seeds and covers was given by Iliopoulos et al in [30, 31]. A linear time algorithm for seed computation was given by Kociumaka et. al.[32]. Li et. al, in [33, 34], computed all the covers of a given word in linear time.

We look at certain special words and characterize their covering words. We extend the work of Christou et al [21]. They studied the various covering words of the Fibonacci word.

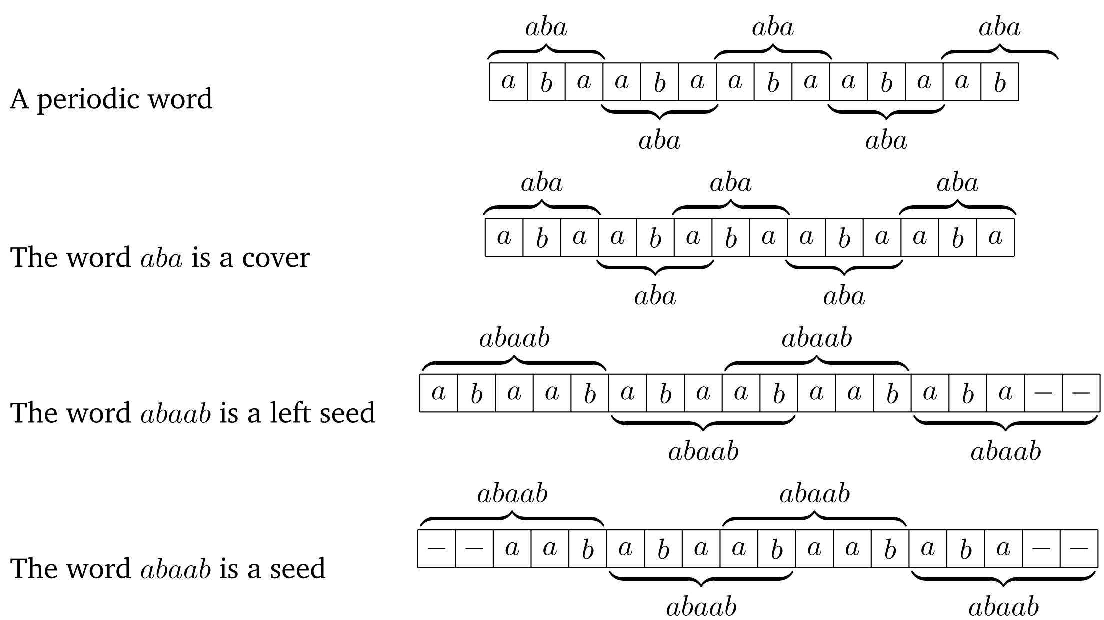


Figure 4.1: Word cover variants

Tribonacci words are generated using a process very similar to that of Fibonacci words. In this chapter we investigate the various covering words of the Tribonacci words.

The first section introduces the definitions and primary results. In the next section, we give a brief overview of the various problems involving Tribonacci words. In Section 4.4 we describe all the borders of Tribonacci word. This helps to determine the covers of Tribonacci word in Section 4.5. Then left/right seeds of Tribonacci word is described in Section 4.6.2. Finally, we characterizes the seed of Tribonacci word in Section 4.6.3

4.1 Mathematical preliminaries

In this section we define the morphism generated words and quasiperiodicity of a word. Consider a morphism given by $\sigma(a) = ab$ and $\sigma(b) = a$. If we denote $\sigma^n(0)$ by F_n then $F_n = F_{n-1}F_{n-2}$. The word F_n is called the n^{th} *Fibonacci word*. The limit word, which is well defined as each F_n contains F_{n-1} as a prefix, is called the *Fibonacci word*. If we denote the length F_n by f_n, then sequence given by f_n satisfies the *Fibonacci* recurrence $f_n = f_{n-1} + f_{n-2}$.

The k^{th} Fibonacci word F_k has many interesting properties. For example, it contains exactly $k + 1$ distinct k length factors for all $k \geq 1$. Words with this property is called as *Sturmian words*. Interested readers may see chapter on Sturmian words in [17] to know more about the Fibonacci words.

In this chapter we study a special word known as *Tribonacci* word. Tribonacci words are similar to Fibonacci words in the sense the recurrence relations used to define these words are very similar.

Consider the *Tribonacci* (also known as *Rauzy*) morphism [35] given by

$$\sigma(a) = ab$$

$$\sigma(b) = ac$$

$$\sigma(c) = a$$

The n^{th} Tribonacci word T_n, $n \in \mathbb{N}$ is obtained by applying n times the morphism σ on the letter a. For example, $T_3(a) = \sigma^3(a) = \sigma^2(ab) = \sigma(abac) = abacaba$. Note that T_n, for $n \geq 4$ is equal to $T_{n-1}T_{n-2}T_{n-3}$. As the word T_n contains T_{n-1} as a prefix, the limit word $\sigma^\omega(a)$ is well defined and is called the *Tribonacci word*. We denote this word by $\mathbf{T}$. If $\{t_n,\ n \in \mathbb{N}\}$ denotes the *Tribonacci* sequence given by the recurrence relation $t_n = t_{n-1} + t_{n-2} + t_{n-3}$ with $t_1 = 2$, $t_2 = 4$ and $t_3 = 7$, then T_n is the prefix of $\mathbf{T}$ of length t_n.

4.1.1 Quasiperiodicity

Periodicity properties of a word is a well studied topic. A periodic word can be *covered* using a single word in a non overlapping way. The notion of quasiperiodicity is obtained by allowing a covering which may overlap. The notion of a cover is described below.

Definition 4.1. *We say that a word u "covers" an index i in a word w if we can find an index j where $j \leq i < j + |u|$ and a subword of w starting at position j is u. A word w is quasiperiodic if there is a word u which covers all indices in w. The word u is said to be the cover of the word w.*

Example 4.2. *The word $w = abaababaabaaba$ is a quasiperiodic word and the word aba is a cover of w. It is described in Figure 4.2.*

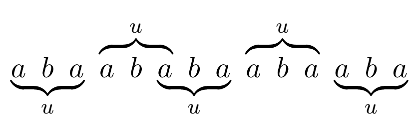


Figure 4.2

The notion of cover requires that the cover is both a prefix and suffix of the word. This would mean that a word which covers a word w may not cover a factor of w. The notion of *seed* is obtained by relaxing this requirement.

Definition 4.2. *A word u is a seed of a word w, if there exists words s and t such that the word u covers swt. A word u is a left seed of a word w, if there exists a word t such that the word u cover of the word wt. A word u is a right seed of a word w, if there exists a word s such that the word u cover of the word sw.*

Example 4.3. *The word $abaab$ is a left seed of the word $w = abaababaabaababa$. Note that $abaab$ does not cover w. It is shown in Figure 4.3.*

$$\underbrace{a\ b\ a\ a\ b}\ \underbrace{a\ b\ a\ a\ b}\ a\ a\ b\ \underbrace{a\ b\ a\ }$$

Figure 4.3

4.2 Problem statement and known results

The covers and seeds of Fibonacci and circular Fibonacci strings has been found in [21]. We extend this study to Tribonacci words. We study the following problems in this chapter.

Problem 3. *What are the borders of the Tribonacci word?*

Problem 4. *What are the covers of the Tribonacci word?*

Problem 5. *What are the seeds of the Tribonacci word?*

Problem 6. *What are the left seeds and right seeds of the Tribonacci word?*

4.3 Known results on border and primitivity of words

We state some basic facts about borders and primitivity of words. A more detailed description can be found in [18].

Lemma 4.1. *A border of a border of a word w is a border of the word w.*

Lemma 4.1 can be extended for covers as well; i.e, given a word w, a cover of a cover of w is another cover of w.

The following lemma can be used to check the primitivity of a word in linear time.

Lemma 4.2. *A word u is primitive if and only if u is not a non trivial factor of uu, i.e $uu = xuy$ implies that $x = \epsilon$ or $y = \epsilon$.*

If the concatenation of two words is commutative then either both words are equal or at least one of the word is not primitive.

Lemma 4.3. *Let $x, y \in \Sigma^*$ such that $xy = yx$. Then there exists a word z and two integers k, l such that $x = z^l$ and $y = z^k$ where $l, k > 1$.*

4.4 Borders of the Tribonacci words

Recall that if a word u which is both a prefix and a suffix of a word w, then we say that u is a border of w. We now show that the Tribonacci word is a primitive word. We shall then use the primitivity properties to determine the borders.

Lemma 4.4. *The word T_n is a primitive word for all $n \in \mathbb{N}$.*

Proof. It is easy to see that T_1 is a primitive. For the sake of contradiction, assume that k is the smallest number for which T_k is not a primitive word. Therefore $T_k = u^j$ where $j > 1$ and T_{k-1} is primitive. Note that $\sigma(T_{k-1}) = T_k$. Let $T_{k-1} = w_1 w_2 \ldots w_r$ where each w_i is a letter from $\{a, b, c\}$. We shall show that T_{k-1} is non primitive contradicting the minimality of k.

Clearly u must start with the letter a. Let $u = ax\alpha$ where α is the last letter of u. As $T_k = \sigma(T_{k-1})$, note that every letter in T_{k-1} "expands" to either ab or ac or a in T_k. In other words, every letter expands to a word of length either one or two. Therefore for every i, the prefix of T_k length i or $i + 1$ must be an image of a prefix of T_{k-1}. Hence, if u is not the image of a prefix of T_{k-1}, then ul where l is the letter immediately after u. But as T_k is equal to u^j, l is also the first letter of u, i.e l is a. Clearly this is impossible as the any word whose image under σ is ua must end in c forcing u to be the image of a prefix of T_{k-1}.

If u is the image of a prefix of T_{k-1} say u', clearly u^j equal to $\sigma(u')^j$. Thus T_{k-1} is not primitive.

□

The next lemma finds the maximum length border of $T_{n-2}T_{n-3}$. We shall use this lemma for calculating the covers of T_n.

Lemma 4.5. *For $n \geq 7$, the longest border of $T_{n-2}T_{n-3}$ is T_{n-3}*

Proof. The word $T_{n-2}T_{n-3}$ can be written as $T_{n-3}T_{n-5}T_{n-6}T_{n-7}T_{n-5}T_{n-3}$. Surely T_{n-3} is the border. Let assume that there is a prefix which is a border of $T_{n-2}T_{n-3}$ and it ends within T_{n-5}. The word $T_{n-2}T_{n-3}$ can be written as $T_{n-3}T_{n-4}T_{n-5}T_{n-3}$. It is clear from the expansion of T_{n-3} that T_{n-5} is a prefix of T_{n-3} and $T_{n-2}T_{n-3} = T_{n-5}xT_{n-3}y$ such that $T_{n-3} = T_{n-5}x$. Hence, if a prefix $T_{n-3}u$ of $T_{n-2}T_{n-3}$ equals a suffix $T_{n-3}v$ of $T_{n-2}T_{n-3}$ where u is a prefix of T_{n-5} then T_{n-3} appears as a factor of $T_{n-3}T_{n-3}$ other than prefix and suffix. By Lemma 4.2, this contradicts primitivity of T_{n-3}.

If the border is of the form $T_{n-3}x$ where $T_{n-5} \leq |x| \leq T_{n-4}$ then T_{n-3} appears as a factor of $T_{n-3}T_{n-3}$ because $T_{n-2}T_{n-3}$ can be written as $T_{n-3}T_{n-3}y$ and T_{n-4} is the prefix of T_{n-3}. $\square$

Given a word w, let x be a border of w, then a border of x must be a border of w. By using this idea, the following theorem characterizes the borders of T_n.

Theorem 4.1. *Every border of the word T_n where $n > 3$ is an element of the set B_n given below;*

$$B_n \triangleq \{T_{n-2}T_{n-3}, T_{n-3}, T_{n-5}T_{n-6}, T_{n-6} \dots T_{(n \bmod 3)+1}T_{(n \bmod 3)}, T_{(n \bmod 3)}\}$$

T_1 and T_2 do not have any borders and aba is the only border of T_3.

Proof. The cases where $n \leq 3$ can be easily checked. The word T_n can be expanded as $T_n = T_{n-2}T_{n-3}T_{n-4}T_{n-2}T_{n-3}$. Let a word x be a border of T_n. Let assume that $x = T_{n-2}T_{n-3}T_{n-4}T_{n-3}y$ where $|y| > 1$. We express T_n as follows.

$$\begin{aligned} T_n &= T_{n-2}T_{n-3}T_{n-4}T_{n-2}T_{n-3} && (4.1)\\ &= T_{n-2}T_{n-3}T_{n-4}T_{n-5}T_{n-6}T_{n-7}T_{n-5}T_{n-6}T_{n-4}T_{n-5}T_{n-3} && (4.2)\\ &= T_{n-2}T_{n-2}T_{n-6}T_{n-7}T_{n-5}T_{n-6}T_{n-4}T_{n-5}T_{n-3} && (4.3) \end{aligned}$$

Since the border x is a suffix of T_n, it must start in between T_{n-2} which has occurred first from left in T_n. By Lemma 4.2 it contradicts primitivity of T_{n-2}.

Let assume that the border $x = T_{n-2}T_{n-3}T_{n-4}y$ where the length of T_{n-3} follows following inequality.

$$|T_{n-3}| \leq |y| \leq |T_{n-2}T_{n-3}T_{n-4}T_{n-3}|$$

We express

$$\begin{aligned} T_n &= T_{n-2}T_{n-3}T_{n-4}T_{n-2}T_{n-3} && (4.4)\\ &= T_{n-2}T_{n-3}T_{n-4}T_{n-5}T_{n-6}T_{n-7}T_{n-5}T_{n-6}T_{n-4}T_{n-5}T_{n-3} && (4.5)\\ &= T_{n-2}T_{n-3}T_{n-3}T_{n-7}T_{n-5}T_{n-6}T_{n-4}T_{n-5}T_{n-3} && (4.6) \end{aligned}$$

The border x in T_n must start in between T_{n-3} which has occurred first from left in T_n.

Since T_{n-3} is a prefix of the word x, by Lemma 4.2 T_{n-3} cannot be primitive.

Let assume that $x = T_{n-2}T_{n-3}y$ where $1 \leq |y| \leq |T_{n-4}|$. Then the border x must start in between T_{n-4} in $T_{n-2}T_{n-3}T_{n-4}T_{n-2}T_{n-3}$. In $T_{n-2}T_{n-3}T_{n-4}T_{n-2}T_{n-3}$, the factor $T_{n-4}T_{n-2}$ can be expressed in such a way that T_{n-4} occurs as a prefix in T_{n-2}. Hence, T_{n-4} occur as a factor in $T_{n-4}T_{n-4}$ other than prefix and suffix. By Lemma 4.2 it contradicts the primitivity of T_{n-4}.

With the help of Lemma 4.1, it can be conclude that the borders of T_n are either $T_{n-2}T_{n-3}$ or border of $T_{n-2}T_{n-3}$. Hence, a border of T_n is an element from the set shown below.

$$\{T_{n-2}T_{n-3}, T_{n-3}, T_{n-5}T_{n-6}, T_{n-6} \ldots T_{1+(n \bmod 3)}T_{(n \bmod 3)}, T_{(n \bmod 3)}\}$$

□

4.5 Covers of Tribonacci word

Every cover is a border but all borders need not be covers. The following lemma describes a border that is not a cover of T_n.

Lemma 4.6. *Let $n \in \mathbb{N}$ and $n \geq 6$. The word $T_{n-2}T_{n-3}$ can not cover T_n.*

Proof. The word $T_n = T_{n-2}T_{n-3}T_{n-4}T_{n-2}T_{n-3}$. Clearly, $T_{n-2}T_{n-3}$ is the prefix and suffix of T_n. Since $|T_{n-4}| < |T_{n-2}T_{n-3}|$, the word $T_{n-2}T_{n-3}$ must start in between the first occurrence of $T_{n-2}T_{n-3}$ or it must start from T_{n-4}. If it starts within T_{n-2} then it contradicts the primitiveness of T_{n-2} and if it starts within T_{n-3} then it contradicts the primitiveness of T_{n-3}. To cover T_{n-4}, the word $T_{n-2}T_{n-3}$ must be of the form $T_{n-4}T_{n-2}T_{n-3} = T_{n-2}T_{n-3}x$ where $|x| = |T_{n-4}|$. We expand T_{n-3} as $T_{n-4}T_{n-5}T_{n-6}$ to get $T_{n-2}T_{n-4} = T_{n-4}T_{n-2}$. Since T_{n-2} and T_{n-4} differ in their lengths, by Lemma 4.3, the word T_{n-2} must be non primitive. It contradicts Lemma 4.4. □

Theorem 4.2. *For $n \geq 7$, the covers of T_n are elements of the set $\{T_{n-3}, T_{n-6}, T_{n-9}, \ldots, T_{n-3k}\}$ where $n - 3k \geq 7$ and $k \in \mathbf{N}$. The word $abacaba$ is the only cover of T_6. The remaining Tribonacci words T_n have no covers.*

Proof. A cover x of the word must appear as a border of T_n. By Lemma 4.6, the word $T_{n-2}T_{n-3}$ does not cover T_n. So, the next possible border of T_n is T_{n-3}. We show that it covers the word T_n. Because of the word

$$T_{n-2}T_{n-3} = T_{n-3}T_{n-4}T_{n-5}T_{n-3} = T_{n-3}T_{n-4}T_{n-5}T_{n-6}x$$

where $T_{n-3} = T_{n-6}x = T_{n-4}T_{n-5}T_{n-6}$, it is easy to see that T_{n-3} covers $T_{n-2}T_{n-3}$. The word $T_{n-4}T_{n-2}$ has a prefix T_{n-3}. Hence, the word T_{n-3} covers

$$T_n = T_{n-2}T_{n-3}T_{n-4}T_{n-2}T_{n-3}$$

The rest of the covers of T_n can be determined recursively. By Theorem 4.1, we can deduce that there does not exist any cover for T_n when $n \in \{1, 2, 3, \cdots 5\}$ and the word $abacaba$ is the only cover of T_6. □

We now move on to the study of the seeds of the Tribonacci words.

4.6 Seeds of Tribonacci words

To get every possible seed of a Tribonacci word we need a special expansion of the Tribonacci word such that we can able to uniquely identify the occurrences of a smaller Tribonacci word in the special expansion.

4.6.1 Results on the special expansion of Tribonacci word

The Tribonacci words can be described recursively. The following theorem gives us a handle on the shape of bigger Tribonacci words in terms of the smaller ones.

Theorem 4.3. *For $m, n \in \mathbb{N}$, there is a unique expansion of T_n in terms of the letters T_m, T_{m-1} and T_{m-2} where $m < n$.*

Proof. Straight forward using induction. □

This expansion has special value because with the help of expansion we will be able to define positions where the word T_m occurs in the expansion. We call such an expansion of the word T_n as (T_m, T_{m-1}, T_{m-2}) *expansion*.

Lemma 4.7. *Let $m, n \in \mathbb{N}$ where $3 < m < n$. None of the elements of the set S given below is a factor of the (T_m, T_{m-1}, T_{m-2}) expansion of the word T_n.*

$$S = \{T_mT_{m-2}, T_{m-2}T_{m-2}, T_{m-1}T_{m-1}, T_{m-2}T_{m-1}, T_mT_mT_m\}$$

Proof. We prove by induction on $i \in \mathbb{N}$ where $i = n - m$. The only $(T_mT_{m-1}T_{m-2}$ expansion of T_n express is $T_{n-1}T_{n-2}T_{n-3}$. It is clear that no element from the set S appears in this word. So, the base case of induction is verified. We may now assume that in the expansion of T_n where all factor come as T_{m+1}, T_m and T_{m-1}, no factors from the set S is present.

Induction hypothesis says that the words T_m or T_{m+1} follow T_{m+1} and the words T_m, T_{m+1} or T_{m-1} precede T_{m+1}. Now we expand T_{m+1} as $T_mT_{m-1}T_{m-2}$. If we combined the above possibilities for T_{m+1} then we introduce the factors of the following form from the set $\{T_{m-2}T_m, T_mT_m, T_{m-1}T_m, T_{m-1}T_{m-2}, T_mT_{m-1}\}$. It does not introduce any factor from the set S. □

Let the word obtained by concatenating the first $n-1$ Tribonacci words be denoted by D_n, i.e. $D_n = T_{n-1}T_{n-2}\cdots T_1$. The following properties of Tribonacci words are from [36].

Property 4.1. *Let $n \in \mathbb{N}$.*

1. *The longest common prefix of the words $T_{n-2}T_{n-3}T_{n-1}$ and T_n is the word $D_{n-2} = T_{n-3}T_{n-4}\cdots T_0 = T_{n-1}D_{n-5}$*
2. *The longest common prefix of the words $T_{n-1}T_n$ and T_nT_{n-1} is $D_n = T_nD_{n-3}$.*
3. *The longest common prefix of the words $T_{n-3}T_{n-1}T_{n-2}$ and T_n is the word $D_{n-3} = T_{n-4}T_{n-5}\ldots T_0$*
4. *The word T_n has prefix D_{n-1} and D_{n-2}.*
5. *The word T_n ends with the letter a for $n \equiv 0 \mod 3$, the letter b for $n \equiv 1 \mod 3$ and the letter c for $n \equiv 2 \mod 3$.*

The following lemma characterizes all the occurrences of T_m in (T_m, T_{m-1}, T_{m-2}) expansion of T_n.

Lemma 4.8. *In the (T_m, T_{m-1}, T_{m-2}) expansion of the word T_n where $4 < m < n$, the non trivial occurrences of T_m starts at a position from where a word T_{m-1} starts.*

Proof. In (T_m, T_{m-1}, T_{m-2}) expansion of T_n, it is easy to see that the word T_m occur as either T_m or starts at the position of T_{m-1}. Let us assume that T_m starts at the position from where T_{m-2} starts. By Lemma 4.7, T_{m-2} must be followed by T_m. So the word $T_{m-2}T_m$ expand as $T_{m-1}T_{m-5}T_{m-3}T_{m-4}T_{m-2}T_{m-3}$. By the third part of Property 4.1, $T_{m-5}T_{m-3}T_{m-4}$ is not equal to T_{m-2}. All the other possibilities are considered below.

Case:1 The word T_m starts in between the word T_m. By Lemma 4.7, we know that in the (T_m, T_{m-1}, T_{m-2}) expansion of T_n, T_m is followed by either T_m or by T_{m-1}. In the first case, T_m occur as a factor of T_mT_m other than prefix and suffix. Therefore, by Lemma 4.2, T_m is not a primitive word. This contradicts Lemma 4.4. For the second case, we know that T_m starts with the position of T_{m-1}. In this case, we get the same contradiction.

Case:2 The word T_m starts in between T_{m-1}. By Lemma 4.7 we know that in the (T_m, T_{m-1}, T_{m-2}) expansion of T_n, T_{m-1} is either followed by T_{m-2} or by T_m. If T_m occurs other than as prefix or suffix in $T_{m-1}T_m$ then by Lemma 4.2, T_{m-1} is not primitive. This contradicts Lemma 4.4. The next possibility is that $T_{m-1}T_{m-2}$ is followed by T_m in above expansion of T_n. In this case, T_m appears as a prefix of $xT_{m-2}T_m$ where the word x is a non empty suffix of T_{m-1}. The word T_m can be expanded as

$$T_m = T_{m-2}T_{m-3}T_{m-4}T_{m-2}T_{m-3} \tag{4.7}$$

$$= T_{m-3}T_{m-4}T_{m-5}T_{m-3}T_{m-4}T_{m-2}T_{m-3} \tag{4.8}$$

The word $xT_{m-2}T_m$ can be written as $xT_{m-1}T_{m-5}T_{m-3}T_{m-4}T_{m-2}T_{m-3}$. The word T_{m-1} is the prefix of $xT_{m-1}T_{m-5}T_{m-3}T_{m-4}T_{m-2}T_{m-3}$ and the word x is the suffix of T_{m-1}, which by Lemma 4.2 implies the non primitivity of T_{m-1}. This contradicts Lemma 4.4.

Case:3 The word T_m starts in between the word T_{m-2}. By Lemma 4.7 we know that in (T_m, T_{m-1}, T_{m-2}) expansion of T_n, T_m follows T_{m-2}. The word T_m has T_{m-2} as a prefix. If T_m starts from in between T_{m-2}, by Lemma 4.2, T_{m-2} is not primitive. This contradicts Lemma 4.4.

By Lemma 4.7, T_{m-2} or T_m follows T_{m-1}. The word $T_{m-1}T_m$ contains T_m as a prefix. By Lemma 4.7, $T_{m-1}T_{m-2}$ must be followed by T_m. Since T_m contains T_{m-3} as a prefix, $T_{m-1}T_{m-2}T_m$ contains T_m as a prefix.

AS we have ruled out all the other cases, T_m starts with either from the starting position of T_m or from the starting position of T_{m-1}. □

By Lemma 4.7, if we consider we consider the words T_m, T_{m-1} and T_{m-2} as letters then we know that in (T_m, T_{m-1}, T_{m-2}) expansion of T_n, the word $T_mT_mT_m$ does not occur as a factor in the expansion. But in (T_m, T_{m-1}, T_{m-2}) expansion of T_n, $T_mT_mT_m$ occurs as a prefix of the factor $T_mT_mT_{m-1}T_{m-2}T_m$ because T_{m-3} is a prefix of T_m. Based on these observations we define two types of factors or words. In the expansion of T_n, if the word $T_{x_1}T_{x_2}\cdots T_{x_r}$ where $x_i \in \{m, m-1, m-2\}$ occurs as a factor, then we call it an *existing word*. Any other occurrence will be referred to as a *derived word*. Lemma 4.8 states that every derived word in T_m starts at an occurrence of T_{m-1}.

4.6.2 Seeds of the one sided extensions of the Tribonacci word

In this section we characterize the left and right seeds of the Tribonacci words. It can be easily checked that the left seeds of T_3 are of the form $abacx$ where $x \in Pref(aba)$ and the left seeds of T_4 are of the form $abacabax$ where $x \in Pref(abacab)$. For the larger Tribonacci words, the following theorem characterizes all the left seeds of T_n.

Theorem 4.4. *Let s be a left seed the word T_n, $n \geq 5$. Then one of the following statements is true;*

- *s is a cover of T_n*
- *s is of the form $T_m x$ where x is a prefix of the word $T_{m-1}T_{m-2}$, $n > m > 4$ and $|x| \leq |D_{m-4}|$.*

Proof. Note that (T_m, T_{m-1}, T_{m-2}) expansion of T_n has T_m as prefix. It is enough to consider the left seeds whose length is greater than or equal to $|T_m|$ and less than $|T_{m+1}|$ because the left seeds of length less than $|T_m|$ is considered in the (T_p, T_{p-1}, T_{p-2}) expansion of T_n where $p < m$ and the length of the left seeds greater than or equal to $|T_{m+1}|$ is considered in the (T_p, T_{p-1}, T_{p-2}) expansion of the word T_n where $m < p$ for an appropriate choice of p.

In (T_m, T_{m-1}, T_{m-2}) expansion of T_n where $m < n$, by Lemma 4.7 T_m always occurs in the form of a word from the set

$$A = \{T_m T_m T_{m-1}, T_m T_{m-1} T_{m-2} T_m, T_m T_{m-1} T_m\}$$

Further by Lemma 4.7 the set A is extended as

$$A = \{T_m T_m T_{m-1} T_m, T_m T_m T_{m-1} T_{m-2} T_m, T_m T_{m-1} T_{m-2} T_m, T_m T_{m-1} T_m\}$$

By the Property 4.1.1 of the Tribonacci word, the longest common prefix of the words $T_m T_{m-1} T_{m-2}$ and $T_{m-1} T_{m-2} T_m$ is $T_m D_{m-4}$. Note that D_{m-4} is a prefix of $T_{m-1}T_{m-2}$. By Lemma 4.8, we conclude that there is no left seed whose length length lies between $|T_m D_{m-4}|$ and $|T_{m+1}|$. For any word $w = w_1 w_2 \ldots w_n$, let *suffixchop*$_i(w) = w_1 w_2 w_{n-i-1}$. It can be seen that for all the words $w \in A$, the word *suffixchop*$_i(w)$ where $i < |T_m|$ is covered by the word $T_m x$ where $|x| \leq |D_{m-4}|$.

Using Property 4.1 of the Tribonacci words, we can conclude that $T_m x$ always starts at the initial T_m and ends at the final T_m in the words from the set

$$A = \{T_m T_m T_{m-1}, T_m T_{m-1} T_{m-2} T_m, T_m T_{m-1} T_m\}$$

Therefore, we conclude that the word $T_m x$ is a left seed of T_n.

□

It can be easily checked that the right seeds of T_3 are of the form $ycaba$, $y \in Suf(aba)$ and the right seeds of T_4 are of the form $yaabacba$ where $y \in Suf(abacab)$. For the larger Tribonacci words, the following theorem characterizes all the right seeds of T_n.

Theorem 4.5. *Let s be a right seed the word T_n, $n \geq 5$. Then one of the following statements is true;*

- *s is a cover of T_n*
- *s is of the form $xT_{n-4}T_{n-2}T_{n-3}$ where x is a suffix of the word $T_{n-2}T_{n-3}$*

Proof. The word T_n is expressed as follows.

$$T_n = T_{n-3}T_{n-4}T_{n-5}T_{n-3}T_{n-4}T_{n-3}T_{n-4}T_{n-5}T_{n-3}$$

A right seed of T_n must be a suffix of T_n. The word T_{n-3} is a suffix of T_n. First we will try to characterize all right seeds whose length is greater than $|T_{n-3}|$ and then we will search right seeds whose length is less than $|T_{n-3}|$.

The word T_{n-5} precedes the suffix word T_{n-3} of T_n. By Lemma 4.8, T_{n-3} occurs at the starting position from T_{n-3} or the starting position of T_{n-4}. By using the Property 4.1.5, the only possibility where T_{n-3} or T_{n-4} is preceded by the length one suffix of T_{n-5}, is in the second occurrence (from left) of T_{n-3}. It implies that a right seed of T_n must be of the form $xT_{n-4}T_{n-3}T_{n-4}T_{n-5}T_{n-3}$ where x is a suffix of $T_{n-3}T_{n-4}T_{n-5}T_{n-3}$.

We have to ensure that no proper suffix of T_{n-3} is a right seed. A right seed which covers T_n must be a right seed of T_{n-3}. Note that we consider the right seeds of T_{n-3} which are not covers. We need to consider only these seeds as Theorem 4.2 and Lemma 4.1 implies that a word which covers the word T_{n-3} covers the word T_n as well. We can inductively conclude that T_{n-3} has right seed $x'T_{n-7}T_{n-5}T_{n-6}$ where x' is a suffix of $T_{n-5}T_{n-6}$. If a word is a right seed of T_n then it must be a right seed of the word:

$$T_{n-5}T_{n-3} = T_{n-5}T_{n-5}T_{n-6}T_{n-7}T_{n-5}T_{n-6}$$

By Property 4.1.5, the last letters of T_{n-7} and T_{n-5} differ. So the word $x'T_{n-7}T_{n-5}T_{n-6}$ can not be a right seed of $T_{n-5}T_{n-3}$. Hence, it can not be the right seed of T_n. Next we have to search for a right seed whose length less than $|T_{n-6}|$ and it is a suffix of T_{n-6}.

By repeated induction, we conclude that if $n \equiv i \mod 3$ where $i \in \{0, 1, 2\}$ then we do not have a right seed of T_{3+i} whose length is less than $|T_i|$ and it is a suffix of the word T_i. Hence there are no right seeds of T_n which is a proper suffix of T_{n-3}. □

4.6.3 Seeds of the two sided extensions Tribonacci word

Every cover, left seed as well as right seed is a seed. In this section, we study the seeds of the Fibonacci words which are not left seeds or right seeds or covers. The next couple of lemmas tries to get a handle on the form of factors inside the (T_m, T_{m-1}, T_{m-2}) expansion of T_n. The form of factors plays an important role in the shape of the seeds.

The (T_m, T_{m-1}, T_{m-2}) expansion of T_n where $m = n-3$, we get the following form of T_n.

$$T_{n-3}T_{n-4}T_{n-5}T_{n-3}T_{n-4}T_{n-3}T_{n-4}T_{n-5}T_{n-3}$$

It has the word $T_mT_{m-1}T_{m-2}T_mT_{m-1}T_mT_{m-1}T_{m-2}T_m$ where $m = n-3$ as a factor. If we expand further ($m = n-4$) then we get the following form of T_n.

$$T_{n-4}T_{n-5}T_{n-6}T_{n-4}T_{n-5}T_{n-4}T_{n-5}T_{n-6}T_{n-4}T_{n-4}T_{n-5}T_{n-6}T_{n-4}\cdots T_{n-5}T_{n-6}$$

The (T_m, T_{m-1}, T_{m-2}) expansion of T_n where $m = n-4$ has the following factors.

1. $T_mT_{m-1}T_{m-2}T_mT_mT_{m-1}T_{m-2}T_m$
2. $T_mT_{m-1}T_{m-2}T_mT_{m-1}T_mT_{m-1}T_{m-2}T_m$

If we expand further values of m then we get that the following words are factors of the expansion where $m \leq n-5$.

1. $T_mT_{m-1}T_{m-2}T_mT_{m-1}T_{m-2}T_m$
2. $T_mT_{m-1}T_{m-2}T_mT_mT_{m-1}T_{m-2}T_m$
3. $T_mT_{m-1}T_{m-2}T_mT_{m-1}T_mT_{m-1}T_{m-2}T_m$

Let $w_1 = T_mT_{m-1}T_{m-2}T_mT_{m-1}T_{m-2}T_m$, $w_2 = T_mT_{m-1}T_{m-2}T_mT_mT_{m-1}T_{m-2}T_m$ and $w_3 = T_mT_{m-1}T_{m-2}T_mT_{m-1}T_mT_{m-1}T_{m-2}T_m$ where T_i is a Tribonacci word. The Lemmas 4.9 and 4.10 prove some results regarding the words w_1, w_2 and w_3. In (T_m, T_{m-1}, T_{m-2}) expansion of T_n, a prefix with respect to the word w_3 has been identified in Lemma 4.9. Note that we are proving about existing word not derived word.

Lemma 4.9. *For $m \leq n-5$, the (T_m, T_{m-1}, T_{m-2}) expansion of the word T_n contains prefix w_3w_3.*

Proof. The $(T_{n-5}, T_{n-6}, T_{n-7})$ expansion of T_n contains prefix w_3w_3 for $m = n-5$. Assume that it is true for a (T_m, T_{m-1}, T_{m-2}) expansion of the word T_n.

$$T_mT_{m-1}T_{m-2}T_mT_{m-1}T_mT_{m-1}T_{m-2}T_mT_mT_{m-1}T_{m-2}T_mT_{m-1}T_mT_{m-1}T_{m-2}T_m$$

For $(T_{m-1}, T_{m-2}, T_{m-3})$ expansion of T_n, we expand T_m further then it contains w_3w_3 prefix for $m-1$. □

Remark 4.1. *The word obtained by appending T_{n-4} to the $(T_{n-4}, T_{n-5}, T_{n-6})$ expansion of T_n contains prefix w_3w_3.*

It is clear from Example 4.6.3 that in (T_m, T_{m-1}, T_{m-2}) expansion of T_n where $m \leq n-4$, we can find the factors w_1, w_2 or w_3 for any value of m. If we see the words w_1, w_2 and w_3 then we can find that the word $T_mT_{m-1}T_{m-2}T_m$ has appeared in an overlapping manner in w_1, has appeared in a concatenated manner in w_2 and is separated by T_{m-1} in w_3. In Lemma 4.10, we prove that these are the only possibilities for the two consecutive occurrence of the word $T_mT_{m-1}T_{m-2}T_m$ in the (T_m, T_{m-1}, T_{m-2}) expansion of T_n.

Lemma 4.10. *The two consecutive occurrences of $T_mT_{m-1}T_{m-2}T_m$ in the (T_m, T_{m-1}, T_{m-2}) expansion of the word T_n where $m \leq n-3$ appear as a factor w_1, w_2 or w_3.*

Proof. In (T_m, T_{m-1}, T_{m-2}) expansion of T_n, by Lemma 4.7, T_mT_{m-1} extend as the words $T_mT_{m-1}T_{m-2}T_m$ and $T_mT_{m-1}T_m$.

Assume that $T_mT_{m-1}T_{m-2}T_mT_mT_{m-1}T_m$ has appeared as a factor in (T_m, T_{m-1}, T_{m-2}) expansion of T_n. Now we move from (T_m, T_{m-1}, T_{m-2}) expansion to (T_{m+1}, T_m, T_{m-1}) for the factor $T_mT_{m-1}T_{m-2}T_mT_mT_{m-1}T_m$. Then we get the following possible words.

$$T_{m+1}T_mT_mT_{m-1}T_m \text{ or } T_{m+1}T_mT_mT_{m-1}T_{m+1}$$

By Lemma 4.7, the words $T_{m-1}T_m$ or T_mT_m can not appear as a factor in (T_{m+1}, T_m, T_{m-1}) expansion of T_n.

Assume that $T_mT_{m-1}T_mT_mT_{m-1}T_{m-2}$ has appeared as a factor in (T_m, T_{m-1}, T_{m-2}) expansion of T_n. We move from (T_m, T_{m-1}, T_{m-2}) expansion to (T_{m+1}, T_m, T_{m-1}) for the factor $T_mT_{m-1}T_mT_mT_{m-1}T_{m-2}$. Then we get the word $T_mT_{m-1}T_mT_{m+1}$. By Lemma 4.7, the word $T_{m-1}T_m$ can not appear as a factor in the (T_{m+1}, T_m, T_{m-1}) expansion of T_n. By the similar arguments, the factors $T_mT_{m-1}T_mT_mT_{m-1}T_m$ or $T_mT_{m-1}T_mT_{m-1}T_m$ cannot appear as a factor in (T_m, T_{m-1}, T_{m-2}) expansion of T_n. The only remaining possibility is $T_mT_{m-1}T_{m-2}T_mT_{m-1}T_mT_{m-1}T_{m-2}T_m$. We claim that the two consecutive occurrences of the word $T_mT_{m-1}T_{m-2}T_m$ can not be separated by more than the word T_{m-1}. For the sake of contradiction assume that it is possible. Consider inserting the letters T_m, T_{m-1} and T_{m-2} inside the word $T_mT_{m-1}T_{m-2}T_mT_{m-1}T_mT_{m-1}T_{m-2}T_m$. There are many possible words that can be constructed. It can be checked that every possible word formed will either having a factor that is forbidden by Lemma 4.7, or have a factor which is already forbidden by this proof, or must contain one of the words among w_1, w_2 and w_3 as a factor. Therefore the consecutive occurrences of the word $T_mT_{m-1}T_{m-2}T_m$ can not be separated by more than the word T_{m-1}. The word $T_mT_{m-1}T_{m-2}T_m$ can also occur in a concatenated manner and can also occur in an overlapping manner in the expansion. □

We can infer from Theorem 4.10 that the words w_1, w_2 and w_3 occur throughout the expansion either in an overlapping (the maximum length of overlap is $|T_{m+1}T_m|$) fashion or in the concatenated manner.

Remark 4.2. *Note that in the (T_m, T_{m-1}, T_{m-2}) expansion of T_n, we can append (or prepend) a word such that the obtained word contains the suffix (or prefix) w_1, w_2 or w_3.*

We mention some properties of the (T_m, T_{m-1}, T_{m-2}) expansion of T_n which will be useful for characterizing the seeds.

Property 4.2. *Let $m, n \in \mathbb{N}$ and $m \le n$. The following properties in the (T_m, T_{m-1}, T_{m-2}) expansion of T_n is true.*

1. *The word T_m is preceded by different letters.*
2. *The longest common prefix of the words T_nT_{n-1} and $T_{n-1}T_{n-2}T_n$ is T_nD_{n-4}.*
3. *The word $T_nT_nT_{n-1}$ is not a prefix of the word $T_{n-1}T_nT_{n-1}T_{n-2}T_n$.*
4. *The word T_nT_n is not a prefix of the word $T_{n-1}T_{n-2}T_nT_{n-1}$.*
5. *The word $T_mT_{m-1}T_m$ is not preceded by T_m or T_{m-1}.*
6. *The word $T_mT_{m-1}T_mT_m$ occurs as a factor but it cannot occur as an existing word.*
7. *The word $T_mT_{m-1}T_mT_mT_{m-1}$ does not occur as a factor.*

Proof. 1. By Lemma 4.10, in the (T_m, T_{m-1}, T_{m-2}) expansion of T_n where $n \ge m$, T_m is preceded by T_m, T_{m-1} and T_{m-2}. By Property 4.1.5, T_m is preceded by different letters in the expansion.

2. By Property 4.1.1,the longest common prefix of $T_{n-3}T_{n-4}T_{n-5}$ and $T_{n-4}T_{n-5}T_{n-3}$ is D_{n-4}.

3. By Property 4.1.2, the words T_nT_{n-1} and $T_{n-1}T_n$ has a longest common prefix.

4. By Property 4.1.1, D_{n-4} is the longest common prefix of the words $T_{n-3}T_{n-4}T_{n-5}$ and $T_{n-4}T_{n-5}T_{n-3}$.

5. If $T_mT_{m-1}T_m$ is preceded by T_m or T_{m-1} then by Lemma 4.7 it must be the following words $T_mT_{m-1}T_mT_{m-1}T_m$, $T_mT_{m-1}T_{m-2}T_mT_mT_{m-1}T_m$ or $T_mT_{m-1}T_mT_mT_{m-1}T_m$. By Lemma 4.10, these words do not occur as a factor in the expansion.

6. The word $T_mT_{m-1}T_m$ can be extended as an existing word $T_mT_{m-1}T_mT_{m-1}$. By Lemma 4.4, in the (T_m, T_{m-1}, T_{m-2}) expansion of T_n, T_m starts with either T_m or T_{m-1}. Hence, in the expansion the derived word $T_mT_{m-1}T_mT_m$ appears as a factor.

7. By Property 4.1.2 and Property 4.2.3, the word $T_mT_{m-1}T_mT_mT_{m-1}$ cannot occur as a factor in (T_m, T_{m-1}, T_{m-2}) extension of T_n.

□

In the (T_m, T_{m-1}, T_{m-2}) expansion of T_n, we look at all the words of the form xFy where $F \in \{T_m, T_{m-1}, T_{m-2}\}^+$ and x, y are words such that xFy covers sw_3w_3t for some words s and t. If xFy covers w_3w_3 then we search that whether xFy covers w_1, w_2 and w_3. If a word xFy covers the words w_1, w_2, w_3 and w_3w_3 then it is a seed of T_n. We shall do the same in the following Lemma 4.11 and Theorem 4.6. In Appendix A, we have given all possible cases which are missed in the following Lemma 4.11 and Theorem 4.6.

Lemma 4.11. *In the (T_m, T_{m-1}, T_{m-2}) of T_n where $m \leq n-4$, there are no seeds of the form xFy where $F \in \{T_m, T_{m-1}, T_{m-2}\}^+$, y is a prefix of a word from the alphabet $\{T_{m-i} \mid i \in \mathbf{N}\}$, $x \in Suf(T_p)$ and $p \in \{m, m-1, m-2\}$ of the word T_n when $|F| \in \{2, 3, 5, 6, 7\}$.*

Proof. We will search seeds of the form xFy where $F \in \{T_m, T_{m-1}, T_{m-2}\}^+$ where we define the words x and y as follows. If the word F has the prefix T_{m-2} then by Lemma 4.7, $x \in Suf(T_{m-1})$. If the word F has the prefix T_{m-1} then by Lemma 4.7, $x \in Suf(T_m)$. If the word F has the prefix T_m then by Lemma 4.7, $x \in Suf(T_m)$ or $x \in Suf(T_{m-1})$ or $x \in Suf(T_{m-2})$. In all these cases the word y will decide during the proof. The word F must contain the factor T_m. If the word F does not contain the factor T_m then we have to consider $(T_{m-1}, T_{m-2}, T_{m-3})$ or $(T_{m-2}, T_{m-3}, T_{m-4})$ expansion of T_n. If for any words x and y the word xFy contain the factor T_{m+1} then instead of (T_m, T_{m-1}, T_{m-2}) expansion of T_n, we search our seed in (T_{m+1}, T_m, T_{m-1}) expansion of T_n. Hence we are not considering the cases when the word sFy contains the factor T_{m+1} and the word F does not contain factor T_m. We extend F from the length $|F|$ to $|F|+1$ by using Lemma 4.7, 4.10 and the 5^{th}, 6^{th} and 7^{th} properties of Property 4.2. These extensions in the word F occur either forward or backward. Then for each possibility of F and x we have searched that whether xFy is a seed of the word w_3w_3 or not. If we explain each and every case then the proof becomes longer in the length. Instead of explaining each and every case we have put only few cases such that an interested reader can infer the central idea of the proof.

Case:1.1.1 The word $F = T_mT_{m-1}$ and $x \in Suf(T_{m-1})$. There are precisely two places in the word w_3w_3 where T_mT_{m-1} is preceded by a suffix of T_{m-1}. To cover w_3w_3, the word xFy must contain factor T_{m+1}. Hence, we have to consider (T_{m+1}, T_m, T_{m-1}) expansion of word T_n.

Case:1.1.2 The word $F = T_mT_{m-1}$ and $x \in Suf(T_{m-2})$. A suffix of T_{m-2} precedes T_mT_{m-1} at the three positions (we are not considering a prefix word T_mT_{m-1}) in the word w_3w_3. So the word y must be a prefix of the words $T_mT_{m-1}T_{m-2}$ and $T_{m-2}T_{m-3}T_{m-1}T_{m-2}$. By Property 4.1.1, $|y| < |T_m|$. So the word xFy does not cover the word w_3w_3.

Case:1.1.3 The word $F = T_mT_{m-1}$ and $x \in Suf(T_m)$. By Property 4.1.2 and Property 4.2.2, 4.2.3 and 4.2.4, a suffix of T_m precedes T_mT_{m-1} at precisely one place in the word w_3w_3. Hence, the word xFy contains the factor T_{m+1}.

Case:1.2.1 The word $F = T_mT_m$ and $x \in Suf(T_{m-2})$. A suffix of T_{m-2} precedes T_mT_m(existing or derived) at three positions in the word w_3w_3. To cover the word w_3w_3, the word y must be a prefix of the words $T_{m-4}T_{m-2}T_{m-3}T_{m-1}T_{m-2}$ and $T_{m-2}T_{m-3}T_{m-4}T_{m-2}$ such that $|x|+|y| \geq |T_{m-4}T_{m-2}T_{m-3}T_{m-1}T_{m-2}|$. By Property 4.1.3, $|y| < |T_{m-1}|$. The word xFy does not cover the word w_3w_3.

The following strategy has followed through out this proof. We search the seed of form xFy which covers the word sw_3w_3t for the minimal length words s and t. If xFy does not cover sw_3w_3t then we reject xFy. For the full proof of this Lemma, an interested reader may see Appendix A.

□

The left seeds, the right seeds and the covers of a word are trivially form a seed. Theorem 4.6 characterizes all the non trivial seeds of the word T_n for $n \geq 4$. The seeds of T_3 and T_4 are of the form xcy where x is a suffix of aba. In case of T_3, y is a prefix of aba such that $|xy| \geq 3$. In case of T_4, y is a prefix of $abaabacab$ and $|xy| \geq 6$.

Theorem 4.6. *Let x be any suffix of T_m, y be any prefix of D_{m-4} and z be any prefix of T_{m-2}. The following words are non trivial seeds of T_n, where $n \geq 4$;*

- *xT_my where $m \leq n-4$ and $|xy| \geq |T_m|$*
- *$xT_{m-1}T_{m-2}T_mT_{m-1}z$ where $m \leq n-4$ and $|xz| \geq |T_m|$*

- *$xT_{m-1}T_{m-2}T_mT_my$ where $m \leq n-4$ and $|xy| \geq |T_m|$*

Proof. We will search seeds of the form xFy where $F \in \{T_m, T_{m-1}, T_{m-2}\}^+$.

Case:1.1 The word $F = T_m$ and $x \in Suf(T_{m-2})$. The word T_m is preceded by a suffix of T_{m-2} at exactly four places inside the word w_3w_3. To cover the word w_3w_3, the word y must be a prefix of the words $T_{m-1}T_mT_{m-1}T_{m-2}$, $T_mT_{m-1}T_{m-2}$ such that $|x| + |y| \geq |T_{m-1}T_mT_{m-1}T_{m-2}|$. By Property 4.1.2, $|y|$ must be less than $|T_mT_{m-1}|$. Hence, it does not cover w_3w_3. So it does not cover word T_n.

Case:1.2 The word $F = T_m$ and $x \in Suf(T_{m-1})$. The word $T_{m-1}T_m$ occurs only in the word w_3. To cover the word w_3w_3, the word xFy must contain the factor T_{m+1}.

Case:1.3 The word $F = T_m$ and $x \in Suf(T_m)$. If the word $y \in Pref(D_{m-4})$ where the word T_mD_{m-4} is equal to the longest common prefix of the words T_mT_{m-1},$T_{m-1}T_m$ and $T_{m-1}T_{m-2}T_m$ and $|x| + |y| \geq |T_m|$ then xFy covers the words w_1, w_2 and w_3 in such a way that the word xFy starts with the first word T_m and ends with the last word T_m. Hence, it covers sT_nt where s and t might be an empty word.

Case:2.1 The word $F = T_{m-2}T_mT_{m-1}T_m$ and $x \in Suf(T_{m-1})$. In the word w_3w_3 it has precisely two occurrences. To cover the word w_3w_3, the word y must contain prefix $T_{m-1}T_{m-2}$. Hence, the word xFy contains the factor T_{m+1}.

Case:2.2.1 The word $F = T_mT_{m-1}T_mT_{m-1}$ and $x \in Suf(T_{m-2})$. There are precisely two places where T_{m-1} precedes T_m in the word w_3w_3. To cover the word w_3w_3, the word y must contain the prefix T_{m-2}. Hence, the word xFy contains the factor T_{m+1}.

Case:2.2.2 The word $F = T_mT_{m-1}T_mT_{m-1}$ and $x \in Suf(T_{m-1})$. By Property 4.2.5, there are no place in (T_m, T_{m-1}, T_{m-2}) expansion of T_n where T_{m-1} precedes $T_mT_{m-1}T_m$.

Case:2.2.3 The word $F = T_mT_{m-1}T_mT_{m-1}$ and $x \in Suf(T_m)$. By Property 4.2.5, there are no places in the expansion where T_m precedes $T_mT_{m-1}T_m$.

Case:2.3 The word $F = T_{m-1}T_mT_mT_{m-1}$ and $x \in Suf(T_m)$. There are two places where T_{m-1} precedes T_m in the word w_3w_3. By Property 4.1.2 the word F does not occur at those places.

Case:2.4 The word $F = T_{m-2}T_mT_mT_{m-1}$ and $x \in Suf(T_{m-1})$. There are exactly three places in the word w_3w_3 where the word $T_{m-2}T_m$ occurs. Note that we are excluding suffix word $T_{m-2}T_m$. By Property 4.1.2, the word F does not occur at first and third place. So the word F has exactly one occurrence in the word w_3w_3. Hence, the word xFy contains factor T_{m+1} for all words x and y.

Case:2.5.1 The word $F = T_mT_{m-1}T_mT_m$ and $x \in Suf(T_{m-2})$. By Lemma 4.10, in the expansion, $T_{m-1}T_{m-2}T_m$ follows $T_mT_{m-1}T_m$. It gives our derived word F. The word F is occurring as a derived word. Hence, to cover w_3w_3, xFy contains factor T_{m+1}.

Case:2.5.2 The word $F = T_mT_{m-1}T_mT_m$ and $x \in Suf(T_{m-1})$. By Property 4.2.5, in (T_m, T_{m-1}, T_m) expansion of the word T_n, T_{m-2} always precedes $T_mT_{m-1}T_m$.

Case:2.5.3 The word $F = T_mT_{m-1}T_mT_m$ and $x \in Suf(T_m)$. By Property 4.2.5, in (T_m, T_{m-1}, T_m) expansion of T_n, T_{m-2} always precedes $T_mT_{m-1}T_m$.

Case:2.6 The word $F = T_{m-1}T_{m-2}T_mT_m$ and $x \in Suf(T_m)$. To cover the word w_3w_3, the word y must be a prefix of the words $T_{m-4}T_{m-2}T_{m-3}T_{m-1}$ and $T_{m-2}T_{m-3}T_{m-4}T_{m-1}$ such that $|x| + |y| \geq |T_m|$. By Property 4.1.1 the word $y \in Pref(D_{m-4})$. The word xFy covers words w_1,w_2 and w_3. Hence, xFy covers T_n.

Case:2.7 The word $F = T_{m-1}T_{m-2}T_mT_{m-1}$ and $x \in Suf(T_m)$. Then $x \in Suf(T_m)$ and $y \in Pref(T_{m-2})$ such that $|x|+|y| \geq |T_m|$. It covers the word w_3w_3 such that it starts within from the first word T_m and end in between at the last word T_m. Same happens to the words w_1 and w_2. Hence, it covers the word T_n.

□

The remaining seeds can be obtained by considering the (T_m, T_{m-1}, T_{m-2}) expansion of T_n when $m \in \{n-1, n-2, n-3\}$. We have not analyzed these cases.

4.7 Conclusion and open problems

We have characterized the cover, border and various kind of seeds of the Tribonacci word. We know that Fibonacci word is a Sturmian word [17]. The characterization of the cover, border and seeds can be studied for other Sturmian words.

Chapter 5
Conclusion and Future Work

5.1 Conclusion

In this thesis we studied three themes related to patterns. These are

- Repetitions
- Permutations
- Quasiperiodicity

The repetition pattern and quasiperiodicity patterns were studied on an unordered alphabet while the study of permutation patterns requires an ordering on the alphabet. We looked at the problem of avoiding repetitions in two dimensional words and showed that it is not possible to construct arbitrarily large two dimensional word on 8 letters such that every line words avoid squares. It raises the lower bound of $f(n)$ to $4.5 \times 2^{n-1}$. Based on the difficulties we faced, our guess is that using similar techniques to improve the bounds we have obtained may be very tedious. It will be interesting to compute the alphabet size required to avoid patterns like cubes and higher powers in two dimension as well as higher dimensions.

Permutation patterns have been studied for it own sake by many researchers. In this thesis, we have attempted to understand graphs in terms of representability by pattern avoiding words. Even for very specific graphs, this is combinatorially challenging. We looked at cycle graphs and have proved that there is a unique (up to circular permutation) two uniform representant word. We have similar results for complete graphs as well. We have gained some understanding on the topic of graphs being representable by uniform and permutation avoiding words.

There are many algorithmic issues related to word representability of graphs. In particular, we looked at the complexity of computing the alternating graphs and have obtained some initial results in this direction.

The last part of this thesis focuses on the quasiperiodicity property morphism generated words. We were interested in finding these properties for the Tribonacci word. We have managed to characterize the border, cover, left/right seed and seed of a Tribonacci words.

5.2 Future work

The exact value of $f(2)$ is still not known. There is a significant gap in the bounds for $f(n)$ where $n > 2$ as well. There are many open problems in the area of word generated graphs and permutation avoiding word generated graphs. The total number of 1342 avoiding words which represents a complete graph is unknown. Characterization of permutation p avoiding word generated graphs which cannot be generated by 2 uniform p avoiding word is open. Most of the questions we have looked at were characterization problems. There are very natural algorithmic problems associated with most of these works. Determining quasiperiodicity properties of arbitrary words is an interesting computational problem. Quasiperiodicity properties of Sturmian words, Quasiperiodicity properties of morphism generated words etc are very interesting research problems.

Appendix A

Proofs

We provide the complete proof of the cases that we have left unproven in Lemma 4.11.

Lemma A.1. *In the* (T_m, T_{m-1}, T_{m-2}) *of* T_n *where* $m \leq n-4$*, there are no seeds* xFy *where* $F \in \{T_m, T_{m-1}, T_{m-2}\}^+$*,* y *is a prefix of a word from the alphabet* $\{T_{m-i} \mid i \in \mathbf{N}\}$*,* $x \in Suf(T_p)$ *and* $p \in \{m, m-1, m-2\}$ *of the word* T_n *when* $|F| \in \{2, 3, 5, 6, 7\}$.

Proof. We will search seeds of the form xFy where $F \in \{T_m, T_{m-1}, T_{m-2}\}^+$ where we define the words x and y as follows. If the word F has the prefix T_{m-2} then by Lemma 4.7, $x \in Suf(T_{m-1})$. If the word F has the prefix T_{m-1} then by Lemma 4.7, $x \in Suf(T_m)$. If the word F has the prefix T_m then by Lemma 4.7, $x \in Suf(T_m)$ or $x \in Suf(T_{m-1})$ or $x \in Suf(T_{m-2})$. In all these cases the word y will decide during the proof. The word F must contain the factor T_m. If the word F does not contain the factor T_m then we have to consider $(T_{m-1}, T_{m-2}, T_{m-3})$ or $(T_{m-2}, T_{m-3}, T_{m-4})$ expansion of T_n. If for any words x and y the word xFy contain the factor T_{m+1} then instead of (T_m, T_{m-1}, T_{m-2}) expansion of T_n, we search our seed in (T_{m+1}, T_m, T_{m-1}) expansion of T_n. Hence we are not considering cases when the word F contains the factor T_{m+1} or does not contain factor T_m. We are extending F from the length $|F|$ to $|F|+1$ by using Lemma 4.7, 4.10 and the Property 4.2.5, 4.2.6 and 4.2.7. These extensions in the word F occur either forward or backward. Then for each possibility of F and x we have searched that whether xFy is a seed of the word w_3w_3 or not. Note that if we increase the length of F beyond 7 then it contains the factor T_{m+1}. It can be seen from the Table A.1 and A.2.

Case:1.1.1 The word $F = T_mT_{m-1}$ and $x \in Suf(T_{m-1})$. There are precisely two places in the word w_3w_3 where T_mT_{m-1} is preceded by a suffix of T_{m-1}. To cover w_3w_3, the word xFy must contain factor T_{m+1}. Hence, we have to consider (T_{m+1}, T_m, T_{m-1}) expansion of word T_n.

Case:1.1.2 The word $F = T_mT_{m-1}$ and $x \in Suf(T_{m-2})$. A suffix of T_{m-2} precedes

T_mT_{m-1} at the three positions (we are not considering a prefix word T_mT_{m-1}) in the word w_3w_3. So the word y must be a prefix of the words $T_mT_{m-1}T_{m-2}$ and $T_{m-2}T_{m-3}T_{m-1}T_{m-2}$. By Property 4.1.1, $|y| < |T_m|$. So the word xFy does not cover the word w_3w_3.

Case:1.1.3 The word $F = T_mT_{m-1}$ and $x \in Suf(T_m)$. By Property 4.1.2 and Property 4.2.2, 4.2.3 and 4.2.4 a suffix of T_m precedes T_mT_{m-1} at precisely one place in the word w_3w_3. Hence, the word xFy contains the factor T_{m+1}.

Case:1.2.1 The word $F = T_mT_m$ and $x \in Suf(T_{m-2})$. A suffix of T_{m-2} precedes T_mT_m(existing or derived) at three positions in the word w_3w_3. To cover the word w_3w_3, the word y must be a prefix of the words $T_{m-4}T_{m-2}T_{m-3}T_{m-1}T_{m-2}$ and $T_{m-2}T_{m-3}T_{m-4}T_{m-2}$ such that $|x|+|y| \geq |T_{m-4}T_{m-2}T_{m-3}T_{m-1}T_{m-2}|$. By Property 4.1.3, $|y| < |T_{m-1}|$. The word xFy does not cover the word w_3w_3.

Case:1.2.2 The word $F = T_mT_m$ and $x \in Suf(T_{m-1})$. The word T_{m-1} precedes T_mT_m at two positions in the word w_3w_3. To cover the word w_3w_3, the word y must contain the factor T_{m+1}.

Case:1.2.3 The word $F = T_mT_m$ and $x \in Suf(T_m)$. The word T_m precedes T_mT_m at precisely one place in the word w_3w_3. To cover the word w_3w_3, the word xFy must contain the factor T_{m+1}.

Case:1.3 Let $F = T_{m-2}T_m$ and $x \in Suf(T_{m-1})$. The word F appears in three places in the word w_3w_3 excluding suffix. To cover the word w_3w_3, the word y must be a prefix of the words $T_{m-1}T_mT_{m-1}$ and T_mT_{m-1} such that $|x| + |y| \geq |T_{m-1}T_mT_{m-1}|$. By Property 4.1.2, $|y| < |T_mT_{m-1}|$. The word xFy does not cover the word w_3w_3.

Case:1.4 The word $F = T_{m-1}T_m$ and $x \in Suf(T_m)$. The word T_m preceded by the word T_{m-1} at the two positions in the word w_3w_3. To cover the word w_3w_3, the word xFy must contain factor T_{m+1}.

Case:2.1 The word $F = T_{m-2}T_mT_{m-1}$ and $x \in Suf(T_{m-1})$. The word F precede by a suffix of T_{m-1} at precisely three positions in the word w_3w_3. To cover the word w_3w_3, the word y must be a prefix of the words T_mT_{m-1} and $T_{m-2}T_{m-3}T_{m-1}$ such that $|x| + |y| \geq |T_mT_{m-1}|$. By Property 4.1.1, $|y| < |T_m|$. Hence, the word xFy does not cover the word w_3w_3.

Case:2.2 The word $F = T_{m-1}T_mT_{m-1}$ and $x \in Suf(T_m)$. There are two places in the word w_3w_3 where T_{m-1} precedes T_m. To cover the word w_3w_3, the word y must contain the word T_{m-2}. Hence, the word xFy contains the factor T_{m+1}.

Case:2.3.1 The word $F = T_mT_mT_{m-1}$ and $x \in Suf(T_{m-2})$. By Property 4.1.2, the word $T_mT_mT_{m-1}$ occurs at exactly one position in the word w_3w_3. Hence, to cover w_3w_3 the word xFy contains factor T_{m+1}.

Case:2.3.2 The word $F = T_m T_m T_{m-1}$ and $x \in Suf(T_{m-1})$. The word T_{m-1} precedes T_m at precisely two positions in the word $w_3 w_3$. By Property 4.2.2 the word xF where $|x| > 0$ does not occur at those places.

Case:2.3.3 The word $F = T_m T_m T_{m-1}$ and $x \in Suf(T_m)$. By Lemma 4.7, we know that the existing word $T_m T_m T_m$ does not occur in the expansion of T_n. However, the derived word $T_m T_m T_m$ occurs as a prefix of $T_m T_m T_{m-1} T_{m-2} T_m$ which occurs at precisely one place in word $w_3 w_3$. By Property 4.2.2, the word xF where $|x| > 0$ does not occur at that place in $w_3 w_3$.

Case:2.4 The word $F = T_{m-1} T_m T_m$ and $x \in Suf(T_m)$. There are precisely two places where T_{m-1} precedes T_m in the word $w_3 w_3$. To cover the word $w_3 w_3$, the word xFy must contain the factor T_{m+1}.

Case:2.5 The word $F = T_{m-2} T_m T_m$ and $x \in Suf(T_{m-1})$. Since, the word $T_{m-2} T_m T_m$ has precisely three occurrences in the word $w_3 w_3$, to cover the word $w_3 w_3$ the word y must be a prefix of the words $T_{m-4} T_{m-2} T_{m-3} T_{m-1}$ and T_{m-1} such that $|x| + |y| \geq |T_{m-4} T_{m-2} T_{m-3} T_{m-1}|$. By Property 4.1.3, $|y| < |T_{m-1}|$. The word xFy does not cover the word $w_3 w_3$.

Case:2.6 The word $F = T_{m-1} T_{m-2} T_m$ and $x \in Suf(T_m)$. The word F appears precisely four places in the word $w_3 w_3$. To cover $w_3 w_3$, the word y must be a prefix of the words $T_{m-1} T_m$ and $T_m T_{m-1}$ such that $|x| + |y| \geq |T_{m-1} T_m|$. So xFy covers the word $w_3 w_3$. Since the word w_2 is a factor of the word $w_3 w_3$, xFy covers the word w_2 too. The word w_1 must be followed by $T_{m-1} T_m$ or $T_m T_{m-1}$, but it cannot be followed by $T_{m-1} T_{m-2} T_m$ because it creates $T_{m+1} T_{m+1} T_{m+1}$(existing) factor in (T_{m+1}, T_m, T_{m-1}) expansion of the word T_n which is a contradiction of Lemma 4.7. To cover the word w_1, the word y must be a prefix of the words $T_{m-1} T_m$ and $T_m T_{m-1}$ such that $|x| + |y| \geq |T_{m-1} T_m|$. By Property 4.1.2, we conclude that $y \in Pref(D_m)$. Since it covers the words w_1, w_2 and w_3, it covers the word T_n. We know that $x \in Suf(T_m)$ from the inequality $|x| + |y| \geq |T_{m-1} T_m|$, we conclude that $|y| > |T_{m-1}|$. Since the word y is a prefix of the words $T_{m-1} T_m$ and $T_m T_{m-1}$, we conclude that the word y must contain T_{m-1} as a prefix. So this case has converted into the case when $F = T_{m-1} T_{m-2} T_m T_{m-1}$ (or $|F| = 4$) and $x \in Suf(T_m)$. This case we will see in our final Theorem.

Case:2.7.1 The word $F = T_m T_{m-1} T_m$ and $x \in Suf(T_{m-2})$. There are no places where T_{m-1} precedes T_m in words w_1 and w_2. There are precisely two places in the word $w_3 w_3$ which contain the factor $T_m T_{m-1} T_m$. To cover the word $w_3 w_3$, whatever the possible words x and y we take, the word xFy has the factor T_{m+1}.

Case:2.7.2 The word $F = T_m T_{m-1} T_m$ and $x \in Suf(T_{m-1})$. By Property 4.2.5, the word T_{m-2} always precedes F in the (T_m, T_{m-1}, T_{m-2}) expansion of the word T_n. So this case is not possible.

Case:2.7.3 The word $F = T_m T_{m-1} T_m$ and $x \in Suf(T_m)$. Same reasoning as in Case:2.7.2.

Case:3.1 The word $F = T_{m-2} T_m T_{m-1} T_m T_{m-1}$ and $x \in Suf(T_{m-1})$. The word xF occurs precisely two places in $w_3 w_3$. To cover $w_3 w_3$ the word $y \in Pref(T_{m-2} T_m T_m)$ such that $|x| + |y| \geq |T_{m-2} T_m T_m)|$. For all possible y the word xFy contains the factor T_{m+1}.

Case:3.2 The word $F = T_{m-2} T_m T_{m-1} T_m T_m$. By Lemma 4.10, in the expansion, the word $T_{m-1} T_{m-2} T_m$ always follows $T_m T_{m-1} T_m$ which gives the word F; it is the derived word. Hence, the word F contains the factor T_{m+1}.

Case:3.3 The word $F = T_m T_{m-1} T_m T_m T_{m-1}$. By Property 4.2.7, in the expansion, the word F does not occur.

Case:3.4 The word $F = T_{m-1} T_{m-2} T_m T_m T_{m-1}$ and $x \in Suf(T_m)$. The word $T_{m-2} T_m$ occur at three positions (excluding suffix word $T_{m-2} T_m$) in the word $w_3 w_3$. By Property 4.1.2, it can not occur at first and third position in the word $w_3 w_3$. To cover the word $w_3 w_3$, the word y must contain the word T_{m-2} as a prefix. Hence, the word xFy contains factor T_{m+1}.

Case:3.5 The word $F = T_{m-1} T_{m-2} T_m T_{m-1} T_m$ and $x \in Suf(T_m)$. The word F has exactly two occurrences in the word $w_3 w_3$. To cover the word $w_3 w_3$, the word y must have a prefix $T_{m-1} T_{m-2}$. So every possible word xFy contains the factor T_{m+1}.

Case:4.1 The word $F = T_{m-1} T_{m-2} T_m T_{m-1} T_m T_{m-1}$ and $x \in Suf(T_m)$. The word F occurs precisely two places in $w_3 w_3$. To cover $w_3 w_3$ the word xFy must contain the factor T_{m+1} for all words x and y.

Case:4.2 The word $F = T_{m-2} T_m T_{m-1} T_m T_m T_{m-1}$. The word $T_{m-2} T_m T_{m-1} T_m T_m T_{m-1}$ does not occur in $w_3 w_3$ because of Property 4.2.7.

Case:4.3 The word $F = T_{m-1} T_{m-2} T_m T_{m-1} T_m T_m$ and $x \in Suf(T_m)$. In this case the word F occurs at precisely two positions in the word $w_3 w_3$. To cover the word $w_3 w_3$, the word y must contain a prefix T_{m-2}. Hence, the word xFy contains the factor T_{m+1} for all words x and y.

Case:5 The word $F = T_{m-1} T_{m-2} T_m T_{m-1} T_m T_m T_{m-1}$. By Property 4.2.7, it does not occur in $w_3 w_3$.

□

We have missed some cases in Lemma A.1 and Theorem 4.6 which requires one line argument. These cases are described in Table A.1 and A.2. We extend F forward with help of Lemma 4.7 and Property 4.2.5.

$\|F\|$	F	Remark
1	T_m	**Theorem 4.6 Case 1**
	T_{m-1}	See $(T_{m-1}, T_{m-2}, T_{m-3})$ expansion of T_n
	T_{m-2}	See $(T_{m-1}, T_{m-2}, T_{m-3})$ expansion of T_n
2	T_mT_m	**Lemma A.1 Case 1.2**
	T_mT_{m-1}	**Lemma A.1 Case 1.1**
	$T_{m-1}T_m$	**Lemma A.1 Case 1.4**
	$T_{m-1}T_{m-2}$	See $(T_{m-1}, T_{m-2}, T_{m-3})$ expansion of T_n
	$T_{m-2}T_m$	**Lemma A.1 Case 1.3**
3	$T_mT_mT_{m-1}$	**Lemma A.1 Case 2.3**
	$T_mT_{m-1}T_m$	**Lemma A.1 Case 2.7**
	$T_mT_{m-1}T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_{m-1}T_mT_m$	**Lemma A.1 Case 2.4**
	$T_{m-1}T_mT_{m-1}$	**Lemma A.1 Case 2.2**
	$T_{m-1}T_{m-2}T_m$	**Lemma A.1 Case 2.6**
	$T_{m-2}T_mT_m$	**Lemma A.1 Case 2.5**
	$T_{m-2}T_mT_{m-1}$	**Lemma A.1 Case 2.1**
4	$T_mT_mT_{m-1}T_m$	Contradicts Property 4.2.5
	$T_mT_mT_{m-1}T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_mT_{m-1}T_mT_m$	**Theorem 4.6 Case 2.5**
	$T_mT_{m-1}T_mT_{m-1}$	**Theorem 4.6 Case 2.2**
	$T_{m-1}T_mT_mT_{m-1}$	**Theorem 4.6 Case 2.3**
	$T_{m-1}T_mT_{m-1}T_m$	Contradicts Property 4.2.5
	$T_{m-1}T_mT_{m-1}T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_{m-1}T_{m-2}T_mT_m$	**Theorem 4.6 Case 2.6**
	$T_{m-1}T_{m-2}T_mT_{m-1}$	**Theorem 4.6 Case 2.7**
	$T_{m-2}T_mT_mT_{m-1}$	**Theorem 4.6 Case 2.4**
	$T_{m-2}T_mT_{m-1}T_m$	**Theorem 4.6 Case 2.1**
	$T_{m-2}T_mT_{m-1}T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
5	$T_mT_mT_{m-1}T_mT_m$	Contradicts Property 4.2.5
	$T_mT_mT_{m-1}T_mT_{m-1}$	Contradicts Property 4.2.5
	$T_mT_mT_{m-1}T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_mT_{m-1}T_mT_mT_{m-1}$	**Lemma A.1 Case 3.3**

Table A.1: Table for each possible cases present in Lemma A.1 and Theorem 4.6

$\lvert F \rvert$	F	Remark
5	$T_m T_{m-1} T_m T_{m-1} T_m$	Contradicts Property 4.2.5
	$T_m T_{m-1} T_m T_{m-1} T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_{m-1} T_m T_m T_{m-1} T_m$	Contradicts Property 4.2.5
	$T_{m-1} T_m T_m T_{m-1} T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_{m-1} T_m T_{m-1} T_m T_m$	Contradicts Property 4.2.5
	$T_{m-1} T_m T_{m-1} T_m T_{m-1}$	Contradicts Property 4.2.5
	$T_{m-1} T_m T_{m-1} T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_{m-1} T_{m-2} T_m T_m T_{m-1}$	**Lemma A.1 Case 3.4**
	$T_{m-1} T_{m-2} T_m T_{m-1} T_m$	**Lemma A.1 Case 3.5**
	$T_{m-1} T_{m-2} T_m T_{m-1} T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_{m-2} T_m T_m T_{m-1} T_m$	Contradicts Property 4.2.5
	$T_{m-2} T_m T_m T_{m-1} T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_{m-2} T_m T_{m-1} T_m T_m$	**Lemma A.1 Case 3.2**
	$T_{m-2} T_m T_{m-1} T_m T_{m-1}$	**Lemma A.1 Case 3.1**
	$T_{m-2} T_m T_{m-1} T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
6	$T_m T_{m-1} T_m T_m T_{m-1} T_m$	Contradicts Property 4.2.5
	$T_m T_{m-1} T_m T_m T_{m-1} T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_{m-1} T_{m-2} T_m T_m T_{m-1} T_m$	Contradicts Property 4.2.5
	$T_{m-1} T_{m-2} T_m T_m T_{m-1} T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
	$T_{m-1} T_{m-2} T_m T_{m-1} T_m T_m$	**Lemma A.1 Case 4.3**
	$T_{m-1} T_{m-2} T_m T_{m-1} T_m T_{m-1}$	**Lemma A.1 Case 4.1**
	$T_{m-2} T_m T_{m-1} T_m T_m T_{m-1}$	**Lemma A.1 Case 4.2**
	$T_{m-2} T_m T_{m-1} T_m T_{m-1} T_m$	Contradicts Property 4.2.5
	$T_{m-2} T_m T_{m-1} T_m T_{m-1} T_{m-2}$	See (T_{m+1}, T_m, T_{m-1}) expansion of T_n
7	$T_{m-1} T_{m-2} T_m T_{m-1} T_m T_m T_{m-1}$	**Lemma A.1 Case 5**

Table A.2: Table for each possible cases present in Lemma A.1 and Theorem 4.6

Bibliography

[1] Axel Thue. Über unendliche Zeichenreihen. *Norske Vid. Selsk. Skr. Mat. Nat. Kl. Christiania*, 7:1–22, 1906. Reprinted in *Selected Mathematical Papers of Axel Thue*, T. Nagell, editor, Universitetsforlaget, Oslo, 1977, 139–158.

[2] Axel Thue. Über die gegenseitige lage gleicher teile gewisser zeichenreihen. *Kra. Vidensk. Selsk. Skrifter, I. Mat. Nat. Kl.*, 46:1–67, 1912.

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

[4] J. Currie and N. Rampersad. Dejean's conjecture holds for n greater than or equal to 27. *arXiv preprint arXiv:0901.3188*, 2009.

[5] J. Currie and N. Rampersad. Dejean's conjecture holds for n greater than or equal to 30. *Theoretical Computer Science*, 410(30):2885–2888, 2009.

[6] J. Currie and N. Rampersad. A proof of Dejean's conjecture. *Mathematics of Computation*, 80(274):1063, 2010.

[7] M. Mohammad-Noori and J.D. Currie. Dejean's conjecture and Sturmian words. *European Journal of Combinatorics*, 28(3):876–890, 2007.

[8] J.M. Ollagnier. Proof of Dejean's conjecture for alphabets with 5, 6, 7, 8, 9, 10 and 11 letters. *Theoretical Computer Science*, 95(2):187–205, 1992.

[9] Jean-Jacques Pansiot. A propos d'une conjecture de F. Dejean sur les répétitions dans les mots. *Discrete Applied Mathematics*, 7(3):297 – 311, 1984.

[10] Michaël Rao. Last cases of Dejean's conjecture. *Theoretical Computer Science*, 412(27):3010–3018, 2011.

[11] Jean Berstel and Juhani Karhumäki. Combinatorics on words: a tutorial. *Bulletin of the EATCS*, 79:178, 2003.

[12] Christian Choffrut and Juhani Karhumäki. Combinatorics of words. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages (1)*, chapter Combinatorics of Words, pages 329–438. Springer-Verlag, Berlin, Heidelberg, 1997.

[13] Francine Blanchet-Sadri. *Algorithmic combinatorics on partial words*. CRC Press, 2007.

[14] Sergey Kitaev and Vadim Lozin. *Words and graphs*. Springer, 2016.

[15] M. Lothaire. *Applied combinatorics on words*, volume 105. Cambridge University Press, 2005.

[16] M. Lothaire. *Combinatorics on Words*. Cambridge Mathematical Library. Cambridge University Press, 2 edition, 1997.

[17] M. Lothaire. *Algebraic combinatorics on words*. Cambridge Univ Press, 2002.

[18] J. Shallit. *A second course in formal languages and automata theory*. Cambridge University Press, 2008.

[19] Miklós Bóna. *Combinatorics of permutations*. Chapman and Hall/CRC, 2016.

[20] Arturo Carpi. Multidimensional unrepetitive configurations. *Theoretical Computer Science*, 56:233–241, 1988.

[21] Michalis Christou, Maxime Crochemore, and Costas S. Iliopoulos. Quasiperiodicities in Fibonacci strings. *Ars Combinatoria*, 129:211–225, 2016.

[22] Tom Davis. The mathematics of Sudoku, 2010.

[23] Jui-Yi Kao, Narad Rampersad, Jeffrey Shallit, and Manuel Silva. Words avoiding repetitions in arithmetic progressions. *Theoretical Computer Science*, 391(1-2):126–137, 2008.

[24] Magnús M. Halldórsson, Sergey Kitaev, and Artem V. Pyatkin. Semi-transitive orientations and word-representable graphs. *Discrete Applied Mathematics*, 201:164–171, 2016.

[25] Sergey Kitaev and Artem V. Pyatkin. On representable graphs. *Journal of Automata, Languages and Combinatorics*, 13(1):45–54, 2008.

[26] Jacob Fox. Combinatorics of permutations. In *Symposium on Discrete Mathematics*, 2014.

[27] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.

[28] Alice L. L. Gao, Sergey Kitaev, and Philip B. Zhang. On 132-representable graphs. *Australasian Journal of Combinatorics*, 69:105–118, 2017.

[29] Alberto Apostolico and Andrzej Ehrenfeucht. Efficient detection of quasiperiodicities in strings. *Theoretical Computer Science*, 119(2):247–265, 1993.

[30] Costas S. Iliopoulos, D. W. G. Moore, and Kunsoo Park. Covering a string. *Algorithmica*, 16(3):288–297, 1996.

[31] Michalis Christou, Maxime Crochemore, Costas S Iliopoulos, Marcin Kubica, Solon P Pissis, Jakub Radoszewski, Wojciech Rytter, Bartosz Szreder, and Tomasz Waleń. Efficient seeds computation revisited. In *Annual Symposium on Combinatorial Pattern Matching*, pages 350–363. Springer, 2011.

[32] Tomasz Kociumaka, Marcin Kubica, Jakub Radoszewski, Wojciech Rytter, and Tomasz Waleń. A linear time algorithm for seeds computation. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1095–1112. Society for Industrial and Applied Mathematics, 2012.

[33] Yin Li and William F Smyth. Computing the cover array in linear time. *Algorithmica*, 32(1):95–106, 2002.

[34] Dennis Moore and William F Smyth. An optimal algorithm to compute all the covers of a string. *Information Processing Letters*, 50(5):239–246, 1994.

[35] Gérard. Rauzy. Nombres algébriques et substitutions. *Bulletin de la Société Mathématique de France*, 110(2):147–178, 1982.

[36] Bo Tan and Zhi-Ying Wen. Some properties of the Tribonacci sequence. *European Journal of Combinatorics*, 28(6):1703–1719, 2007.

Publications

Journals

- Benny George Kenkireth, Mrityunjay Singh. “On the Minimal Alphabet Size in Multidimensional Unrepetitive Configurations”, *Discrete Applied Mathematics*. accepted.

Workshop

- Ameya Daigwane, Benny George Kenkireth, Mrityunjay Singh. “2-uniform words: cycle graphs, and an algorithm to verify specific word-representations of graphs", *Words and Complexity*, Villeurbanne, France in February 2018.

Submitted

- Mrityunjay Singh. “Quasiperiodicity in Tribonacci Words", in *Discrete Mathematics and Theoretical Computer Science*.