8-31-2015 12:00 AM

A Study of Pseudo-Periodic and Pseudo-Bordered Words for Functions Beyond Identity and Involution

Manasi Kulkarni, *The University of Western Ontario*

Supervisor: Dr. Lila Kari, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Computer Science

A STUDY OF PSEUDO-PERIODIC AND PSEUDO-BORDERED WORDS FOR FUNCTIONS BEYOND IDENTITY AND INVOLUTION

(Thesis format: Integrated Article)

by

Manasi Kulkarni

Graduate Program in Computer Science

A thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

The School of Graduate and Postdoctoral Studies

The University of Western Ontario

London, Ontario, Canada

Abstract

Periodicity, primitivity and borderedness are some of the fundamental notions in combinatorics on words. Motivated by the Watson-Crick complementarity of DNA strands wherein a word (strand) over the DNA alphabet {A, G, C, T} and its Watson-Crick complement are informationally equivalent, these notions have been extended to consider pseudo-periodicity and pseudo-borderedness obtained by replacing the "identity" function with "pseudo-identity" functions (antimorphic involution in case of Watson-Crick complementarity). For a given alphabet Σ, an antimorphic involution θ is an antimorphism, i.e., $\theta(uv) = \theta(v)\theta(u)$ for all $u, v \in \Sigma^*$ and an involution, i.e., $\theta(\theta(u)) = u$ for all $u \in \Sigma^*$. In this thesis, we continue the study of pseudo-periodic and pseudo-bordered words for pseudo-identity functions including involutions.

To start with, we propose a binary word operation, θ-catenation, that generates θ-powers (pseudo-powers) of a word for any morphic or antimorphic involution θ. We investigate various properties of this operation including closure properties of various classes of languages under it, and its connection with the previously defined notion of θ-primitive words.

A non-empty word u is said to be θ-bordered if there exists a non-empty word v which is a prefix of u while $\theta(v)$ is a suffix of u. We investigate the properties of θ-bordered (pseudo-bordered) and θ-unbordered (pseudo-unbordered) words for pseudo-identity functions θ with the property that θ is either a morphism or an antimorphism with $\theta^n = I$, for a given $n \geq 2$, or θ is a literal morphism or an antimorphism.

Lastly, we initiate a new line of study by exploring the disjunctivity properties of sets of pseudo-bordered and pseudo-unbordered words and some other related languages for various pseudo-identity functions. In particular, we consider such properties for morphic involutions θ and prove that, for any $i \geq 2$, the set of all words with exactly i θ-borders is disjunctive (under certain conditions).

Keywords: morphic and antimorphic involutions, pseudo-bordered words, pseudo-unbordered words, disjunctivity, pseudo-identity, pseudo-power, pseudo-periodicity, pseudo-primitivity.

Co-authorship statement

This thesis consists of three research articles out of which the articles presented in Chapter 3 and 4 are published while the article in Chapter 5 is submitted for publication to the journal *Acta Informatica* and is undergoing a review process. All of them are co-authored with my supervisor Prof. Lila Kari. Note that, as customary in computer science, the author order is alphabetical.

The major individual contributions are listed below. However, the results of this collaboration cannot be decomposed into discrete sets of individual contributions, as some key results arose during discussions, and would not have existed without this interaction.

Chapter 3, “Generating pseudo powers of a word”,

L.K. - topics, research ideas, manuscript writing and editing

M.K. - research ideas and proofs, results, manuscript draft and manuscript editing

Chapter 4, “Pseudo-identities and bordered words”,

L.K. - topics, research ideas, manuscript writing and editing

M.K. - research ideas and proofs, results, manuscript draft and manuscript editing

Chapter 5, “Disjunctivity and other properties of sets of pseudo bordered words”,

L.K. - topics, research ideas, manuscript writing and editing

M.K. - research ideas and proofs, results, manuscript draft and manuscript editing

Acknowledgements

First and foremost I would like to express my immense gratitude to my supervisor Dr. Lila Kari for her support and guidance throughout my Ph.D. studies. She is an excellent mentor. Not only did she guide me in my studies but she extended her warm support in my tough times. I have hardly seen anybody who is always so energetic, enthusiastic and motivating. My doctoral program has been partially supported financially by the Natural Science and Engineering Research Council of Canada Discovery grant to Dr. Lila Kari.

I would like to thank Prof. Lucian Ilie and Prof. Kaizhong Zhang for reading my thesis proposal and for their constructive comments and suggestions.

I would like to acknowledge the University of Western Ontario for Western Graduate Research Scholarship, and also for providing excellent teaching related and other workshops. I would like to take this opportunity to thank the faculty and staff in the Department of Computer Science for providing me with a supportive environment to carry out my research, and for giving me an opportunity to demonstrate my leadership abilities.

Thanks to my colleagues in Biocomputing lab: Amirhossein Simjour, Rallis Karamichalis, Srujan Kumar Enaganti and Dr. Steffen Kopecki for maintaining a friendly atmosphere in the lab. Special thanks to Rallis Karamichalis for fruitful discussions and for reading drafts of my work whenever I asked him to do so. I am very grateful to all my friends and roommates for supporting me. Thanks to Amita, Ansh, Nikhil and London Marathi Mandal for providing me with a home away from home.

Thanks to Dr. Kalpana Mahalingam and Mr. Shriprasad Tambe for their motivation and encouragement to pursue Ph.D. studies.

Special thanks to my husband Amit, who supported me in all possible ways to pursue my career and dreams. Without his support it would have been really hard to achieve this milestone of my life. Last but not least, it would have been impossible to complete this journey without the support of my parents and my brother, Saurabh. In spite of being thousands of miles away they provided me with emotional support and constant encouragement. I never got a chance to

say thanks to my late grandfather, Eknath Shankpal who taught me to love mathematics. His contribution to my success is enormous.

To my loving parents, Shrikant and Prachi, and to the memory of my late grandfather,

Eknath Shankpal

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The study on repetitions and square-free words by Axel Thue [2] initiated the further exploration of word properties and formal languages, an area of discrete mathematics known as combinatorics on words. This exploration includes studies about word operations and properties of words ([6]) such as periodicity, primitivity, conjugacy, commutativity, palindromes, etc. It has also opened the door to the study of infinite words and sequences, with Thue-Morse words, Sturmian words, Fibonacci words ([5]) being some of the representatives.

Due to the close connection of this field with mathematics, the question that arose in this context was to generalize word properties by replacing the identity mapping with pseudo-identity functions. The experiment that used Deoxyribonucleic Acid (DNA) as a medium to encode information and solve computational problems, performed by Adleman ([1]) triggered the study of word and language properties using the pseudo-identity functions such as the antimorphic involution which is the mathematical formalization of the DNA Watson-Crick (WK) complementarity (see Chapter 2).

A DNA strand can be viewed as a word over the DNA alphabet {A, G, C, T} wherein A is a WK-complement of T and G is aWK-complement of C and vice versa. Two single DNA strands, that are WK complements of each other and have opposite orientations, bind to each other to form a DNA double strand via a process called base-pairing, or hybridization. Thus,

DNA WK-complementarity can be modelled as an antimorphic involution, a function that is an antimorphism, i.e., for all $u, v \in \Sigma^*$ $\theta(uv) = \theta(v)\theta(u)$ and an involution, i.e., $\theta(\theta(u)) = u$ for all $u \in \Sigma^*$.

In this thesis, we continue the exploration of word operations and properties for various pseudo-identity functions including morphic and antimorphic involutions.

In Chapter 2, we give an overview of research related to theoretical aspects of DNA encoded information along with some DNA memory models proposed in the literature.

Chapter 3 contains the article, "Generating pseudo-powers of a word", in which we propose binary word operations that produce θ-powers (θ-catenation) and Abelian-powers (Abelian-catenation) for any morphic or an antimorphic involution θ, thereby generalizing the notion of identity to pseudo-identity in terms of the binary word operation of catenation. We mainly focus on the properties of the operation of θ-catenation.

Chapter 4 contains the article, "Pseudo-identities and bordered words" where we continue the study of θ-bordered words initiated in [4] for θ being not just a morphic or an antimorphic involution, but any literal morphism or an antimorphism.

In Chapter 5, which contains the article, "Disjunctivity and other properties of sets of pseudo-bordered words", we continue to explore disjunctivity properties of some languages related to the set of θ-bordered words for (anti)morphic involutions θ which was initiated in [3]. In particular, we prove that, for all $i \geq 1$, the set of all words with exactly i θ-borders, $D_\theta(i)$, is disjunctive (under certain conditions).

We conclude the thesis with Chapter 6, a discussion of the main results in the thesis and future work.

Bibliography

[1] L. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994.

[2] J. Berstel and A. Thue. *Axel Thue's papers on repetitions in words: a translation*. Départements de mathématiques et d'informatique, Université du Québec à Montréal, 1995.

[3] L. Kari and M. S. Kulkarni. Pseudo-identities and bordered words. In G. Păun, G. Rozenberg, and A. Salomaa, editors, *Discrete Mathematics and Computer Science*, pages 207–222. Editura Academiei Române, 2014.

[4] L. Kari and K. Mahalingam. Involutively bordered words. *International Journal of Foundations of Computer Science*, 18(05):1089–1106, 2007.

[5] M. Lothaire. *Algebraic Combinatorics on Words*, volume 90. Cambridge University Press, 2002.

[6] S.-S. Yu. *Languages and Codes*. Tsang Hai Book Publishing Co., 2005.

Chapter 2

DNA Encoded Information: A Literature Review

2.1 Introduction

The demand for reliable media to store information safely is growing rapidly with the vastly increasing amount of data. Present day technologies are catering to the need to the best of their abilities, but many of these commonly used technologies such as hard drives and flash drives suffer from a lack of being sustainable through extreme environmental and some other conditions such as drought, earthquake, radiation, etc. Hence, scientists are in search of reliable media as an alternative to present-day silicon-based computers and hard drives. DNA is believed to be one of the very strong candidates, due to its natural ability to store the information about the genetic make-up of an organism, and the information storage density. According to [58], DNA can store up to 4.2×10^{21} bits per gram whereas conventional technologies can store maximum of 10^9 bits per gram. The experiment performed by Leonard Adleman ([1]) confirmed the fact that DNA indeed can be used for computation and data storage purposes. Adleman conducted an experiment to solve a 7-node instance of the NP-complete Hamiltonian Path Problem. The basic idea was to encode the vertices and edges of the graph into DNA

molecules and then to use sets of bio-operations to find the solution to the problem. This experiment initiated the field of DNA or bio-molecular computing which studies the arithmetic and logic operations that can be performed using molecular biology processes.

Recall that DNA (Deoxyribonucleic Acid) is the genetic information storage unit of every cell in all living organisms (and many viruses). Each single-stranded DNA molecule consists of a sequence of nucleotides. Each nucleotide is composed of: a cyclic five-carbon sugar ring (the carbon atoms are numbered 1' through 5'), a phosphate group, and a nitrogenous base (Adenine, Guanine, Cytosine, or Thymine abbreviated as A, G, C, or T respectively). The phosphate group is linked to the 5' carbon of the sugar, and the nitrogenous base attaches to the 1' carbon of the sugar. The 5'-phosphate group of one nucleotide binds to the 3'-hydroxyl group (the hydroxyl group attaches to the 3' carbon of the sugar) of other nucleotide by covalent bonds. This chain of alternating sugar and phosphate molecules forms the so called sugar-phosphate backbone of a DNA strand. A single-stranded DNA molecule has an orientation with one end being called the 5' end (since the free phosphate group attaches to the 5' carbon of the sugar) and the other end the 3' end (since the free hydroxyl group attaches to the 3' carbon of the sugar). A double-stranded DNA molecule consists of two single-stranded DNA molecules with opposite orientation which bind to each other with hydrogen bonds between nucleotides in the process of hybridization: the nucleotide A binds to the nucleotide T and vice versa, with double hydrogen bonds, whereas the nucleotide G binds to the nucleotide C and vice versa, with triple hydrogen bonds. The bases A, T and G, C are said to be Watson-Crick (WK)-complements of each other [67].

To bring the idea of storing information on DNA into practice, there are two issues that need to be taken into consideration. First, we need to find a suitable encoding method considering various constraints, so that the information can be stored and retrieved unambiguously without losing it. Second, we need to find a suitable host to store this encoded information. The first issue is called the *codeword design problem* and it is usually defined as finding short words over the DNA alphabet, which are usually equi-length, [44], that satisfy certain combinatorial

constraints. This chapter gives an overview of the solutions that deal with the above mentioned issues. Note that rather than focussing on algorithmic aspects more emphasis will be given to formal-language theoretical solutions that overcome the codeword design problem.

While synthesizing artificial DNA strands for the purpose of computation and storage, it is important to design these strands in such a way that only the desired computations and interactions will take place (*positive design problem*) while all other undesired computations and interactions are avoided (*negative design problem*). Briefly, the positive design problem is to design a set of DNA strands so that they will interact with each other in a programmable way so as to produce the desired results, whereas the negative design problem is to design a set of DNA strands so that these strands will not interact with each other in an unprogrammed way and will not produce undesired outputs [17, 52, 59]. Here undesired outputs mean strands that are the result of undesired self- or cross-hybridization.

The chapter is organized as follows. In Section 2.2 we discuss formal-language theoretical solutions to the codeword design problem, followed by other solutions such as, software simulation, and test tube experiments in Section 2.3. In Section 2.4 we discuss some *in vitro* and *in vivo* DNA memory models. In Section 2.5 we discuss generalizations of many classical concepts from the combinatorics on words inspired by the WK-complementarity of DNA strands, with concluding remarks in Section 2.6.

2.2 DNA codeword design and formal languages

In this section we discuss mainly formal language theoretic and combinatorial solutions to the negative design problem. According to [59] the input DNA strands involved in the computation which need to avoid unwanted hybridizations should satisfy certain conditions such as: (i) there should not be any strands with undesired secondary structures such as hairpin loops, (ii) no two strings in the library should hybridize with each other, and (iii) no string and a complement of another string in the library should hybridize with each other.

The hybridization in which two parts of the same DNA strand are WK-complements of each other, forming hairpin-like structures, is known as *intra-molecular hybridization*. When two DNA strands that are complete or partial WK-complements of each other hybridize then such a hybridization is called *inter-molecular hybridization*. Let us discuss some notations and definitions used in this section and in Section 2.5.

Basic definitions and notations

An alphabet Σ is a finite non-empty set of symbols. Σ^* denotes the set of all words over Σ including the empty word λ and $\Sigma^+ = \Sigma^* \backslash \lambda$. We will denote the DNA alphabet by $\Delta = \{A, C, G, T\}$. In this context, a set of codewords is a set of equi-length words over the DNA alphabet. The length of a word $u \in \Sigma^+$ is denoted by $|u|$, whereas the length of an empty word $|\lambda| = 0$, and Σ^i denotes the set of all words of length i over Σ for $i \geq 1$. An involution θ is a map $\theta : \Sigma^* \to \Sigma^*$ with the property that θ^2 is the identity function. A mapping θ is called a morphism if for any words $u, v \in \Sigma^*$, $\theta(uv) = \theta(u)\theta(v)$ and an antimorphism if $\theta(uv) = \theta(v)\theta(u)$. Recall that DNA WK-complementarity can be formalized as an antimorphic involution, a function which is an antimorphism and an involution.

A word $x \in \Sigma^+$ is a *prefix* (proper prefix) of the word u if $u = xy$ for $y \in \Sigma^*$ ($y \in \Sigma^+$), and this is denoted by $x \leq_{\mathrm{p}} u$ ($x <_{\mathrm{p}} u$). Similarly, for $u = xy$, y is a *suffix* (proper suffix) of u if $x \in \Sigma^*$ ($x \in \Sigma^+$) and this is denoted by $y \leq_{\mathrm{s}} u$ ($y <_{\mathrm{s}} u$). Let us denote the set of all prefixes (proper prefixes, suffixes, proper suffixes, respectively) of u by $\mathrm{Pref}(u)$ ($\mathrm{PPref}(u)$, $\mathrm{Suff}(u)$, $\mathrm{PSuff}(u)$, respectively). A word $v \in \Sigma^+$ is a *subword* of a word u if $u = xvy$ for $x, y \in \Sigma^*$. By $\mathrm{Sub}(u)$ and $\mathrm{Sub}_k(u)$, we denote the set of all subwords of u, and the set of all subwords of length k of u, respectively. For a word u such that $u = xy$ for $x, y \in \Sigma^*$, yx is called a *cyclic permutation* of u. The set of all cyclic permutations of u is denoted by $\mathrm{cp}(u)$. For all other concepts in the combinatorics on words and formal languages, the reader is referred to [30, 47, 61, 70].

In the following subsection we discuss some secondary structures such as hairpins and pseudo-knots which are result of an intra-molecular hybridization of a DNA or an RNA molecule,

respectively. Note that the terms "words" and "strands" will be used interchangeably in the following sections.

2.2.1 Intra-molecular hybridizations: hairpins and pseudoknots

The hairpin loop is one of the primary secondary structures formed due to the intra-molecular hybridization of a DNA molecule. Although the hairpin loop formation by a single-stranded DNA molecule is useful in solving some combinatorial problems (see, e.g., [1, 60]) and in improving the data transmission between two logic gates in DNA-based logic circuits [32], it is undesirable for the purpose of encoding information, since such molecules become unavailable for further computations. Hence, they should be avoided in most DNA computing experiments.

In this section we first discuss the formalization of hairpin loops as words over the DNA alphabet, and present some combinatorial and formal language-theoretic properties of such (sets of) words, and then we discuss languages which avoid DNA hairpin loop formation.

Mathematical formalization

A word u is called bordered if there exists a word $v \in \Sigma^+$ such that $u = vx = yv$ for $x, y \in \Sigma^+$ and this is denoted by $v <_d u$. Similarly,

Definition 2.1 *[39] For any (anti)morphism[1] θ, a word $u \in \Sigma^+$ is said to be θ-bordered if there exists $v \in \Sigma^+$ such that $u = vx = y\theta(v)$ for some $x, y \in \Sigma^+$ and this is denoted by $v <_d^\theta u$. A non-empty word which is not θ-bordered is called θ-unbordered.*

Let us denote by $L_d^\theta(u) = \{v \mid v \in \Sigma^*, v <_d^\theta u\}$ the set of all θ-borders of a word $u \in \Sigma^*$; by $\nu_d^\theta(u) = |L_d^\theta(u)|$ the number of θ-borders of a word $u \in \Sigma^*$, and by $D_\theta(i) = \{u \mid u \in \Sigma^+, \nu_d^\theta(u) = i\}$ the set of all words with exactly i θ-borders, for $i \geq 1$. When θ is an antimorphic involution and a θ-bordered word over the DNA alphabet has non-overlapping θ-borders, it forms a hairpin-like structure as shown in Figure 2.1.

[1] By (anti)morphism we mean either a morphism or an antimorphism.

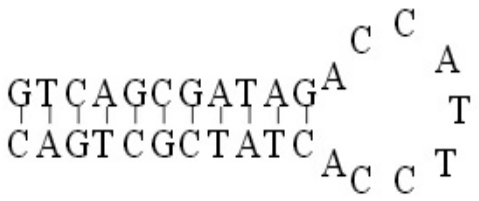

Figure 2.1: DNA hairpin structure formed by a θ-bordered word over the DNA alphabet with non-overlapping θ-border, GTCAGCGATAG (θ is an antimorphic involution over $\{A, G, C, T\}$) [39]

Example 2.1 *Let $\Sigma = \{A, C, G, T\}$ and θ be an antimorphic involution such that $\theta(A) = T$, $\theta(C) = G$ and vice versa. Then, $w = TCGTCTTACGA = (TCGT)CTT\theta(TCGT)$ is θ-bordered whereas $w' = TGCT$ is θ-unbordered.*

The following result provides a necessary and sufficient condition for a word to be θ-bordered for an antimorphic involution θ.

Lemma 2.2.1 *[39] Let θ be an antimorphic involution. Then $x \in \Sigma^+$ is θ-bordered iff $x = ay\theta(a)$ for some $a \in \Sigma$ and $y \in \Sigma^*$.*

For a morphic involution θ, the set of all θ-bordered words over Σ is not context-free but it is context-sensitive, whereas for an antimorphic involution θ, the set of all θ-bordered words over Σ^* is a regular and dense language, [39].

Proposition 2.2.2 *[39] Let $u \in \Sigma^+$. Then*

1. *For a morphic involution θ, $L_d^\theta(u)$ is a totally ordered set with $<_d$.*
2. *For an antimorphic involution θ, $L_d^\theta(u)$ is a totally ordered set with $<_p$ and $\theta(L_d^\theta(u))$ is a totally ordered set with $<_s$.*

Similar to a DNA strand, an RNA strand, which is a strand over the alphabet {A, G, C, U} (U-uracil) such that $\theta(A) = U$ and $\theta(G) = C$ and vice versa, can interact with itself to form pseudoknot like intra-molecular structures. An example of such a structure can be found in

E. Coli transfer-messenger-RNA as shown in Figure 2.2 and has been formalized in [43] as a string $v_1xv_2yv_3\theta(x)v_4\theta(y)v_5$. However the authors of [43] consider a special case of the general model of pseudoknot where $v_1 = v_2 = v_4 = v_5 = \lambda$ and call such words θ-pseudoknot-bordered words. Formally, a non-empty word u is θ-pseudoknot-bordered if $u = xy\alpha = \beta\theta(yx)$ for some words $x, y, \alpha, \beta \in \Sigma^+$. An example of a pseudoknot, that is, a word of the form $xy\gamma\theta(x)\theta(y)$, is given in Figure 2.3.

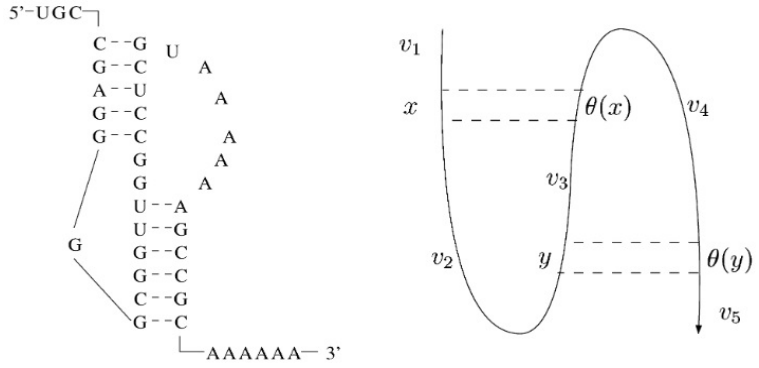

Figure 2.2: Left: A pseudoknot found in *E.coli* transfer-messenger-RNA. Right: Mathematical formalization in terms of a string $v_1xv_2yv_3\theta(x)v_4\theta(y)v_5$ [43]

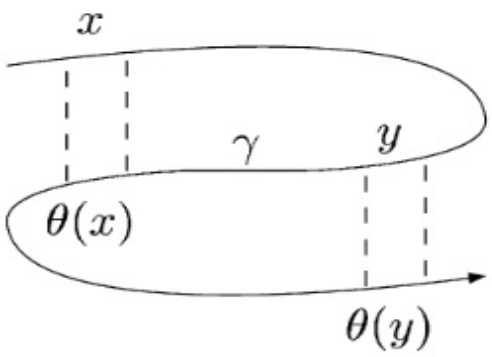

Figure 2.3: Pseudoknot structure formed due to intra-molecular hybridization modelled as a θ-pseudoknot-bordered word $xy\gamma\theta(x)\theta(y)$ [43]

A word u is said to be θ-pseudoknot-border (or θ-pk-border) of a word $v \in \Sigma^*$ if there exists a cyclic permutation w of u such that $v = u\alpha = \beta\theta(w)$ for some $\alpha, \beta \in \Sigma^*$. Furthermore, a non-empty word is said to be θ-pseudoknot-unbordered (or θ-pk-unbordered) if it does not have any non-empty θ-pk-borders. Let $L^{\theta}_{cd}(u)$ denote the set of all θ-pk-borders of a non-empty word u

and $K_\theta(i) = \{u \in \Sigma^+ \mid |L^\theta_{cd}(u)| = i\}$, the set of all words with exactly i θ-pk-borders.

Example 2.2 *Let θ be an antimorphic involution and let $\{a, b\} \in \Sigma$ be such that $\theta(a) = b$ and $\theta(b) = a$. Then for $x = baa$ and $y = b$, we have $w = baabbaa = xybaa = baa\theta(yx)$ which is a θ-pk-bordered word, whereas $w = aab$ is θ-pk-unbordered.*

We state the following results from [43] regarding some properties of θ-pk-borders of a word.

Proposition 2.2.3 *[43] Let θ be an (anti)morphic involution on Σ^*. The following hold:*

1. *If a word $w \in \Sigma^+$ has a θ-pk-border of length n then, for every $a \in \Sigma$, the number of occurrences of a letter a in the prefix of length n of w is equal to the number of occurrences of the letter $\theta(a)$ in the suffix of length n of w.*

2. *For all $a \in \Sigma$ such that $\theta(a) \neq a$, a^k is θ-pk-unbordered, for all $k \geq 1$.*

3. *For words $v, w \in \Sigma^+$ and $n \geq 1$, if $v \in L^\theta_{cd}(w^n)$ and $|w^{m-1}| < |v| \leq |w^m|$ for some $m \geq 1$, then $v \in L^\theta_{cd}(w^k)$, for all k with $m \leq k \leq n$.*

Now, we will discuss properties of languages that prevent the words of the language from interacting with themselves in an undesirable manner.

Hairpin and pseudoknot avoidance

It is believed that the hairpins with smaller stem length are less stable than those with bigger stem length, [44]. Hence, a strand which does not satisfy the stem length condition (of the stem length being smaller than a given value) is free from such hairpin structures. Formally,

Definition 2.2 *1. [38] Let θ be an (anti)morphic involution of Σ^* and k be a positive integer. A word $u \in \Sigma^*$ is said to be θ-k-hairpin-free or simply $hp(\theta, k)$-free if $u = xvy\theta(v)z$ for some $x, v, y, z \in \Sigma^*$ implies $|v| < k$.*

2. *[38] Denote by hpf(θ, k), the set of all hp(θ, k)-free words in Σ^*. The complement of hpf(θ, k) is hp(θ, k)=$\Sigma^* \backslash hpf(\theta, k)$.*

3. *[38] A language L is called θ-k-hairpin-free or simply hp(θ, k)-free if $L \subseteq hpf(\theta, k)$.*

It is obvious that an empty word and single letter words are hp(θ,1)-free whereas words of length less than $2k$ are hp(θ,k)-free.

Proposition 2.2.4 *[38] The languages $hp(\theta, k)$ and $hpf(\theta, k)$ are regular for $k \geq 1$.*

Since the languages given in Proposition 2.2.4 are regular, it is interesting from a theoretical point of view to see whether or not for a given automaton we can decide if the language accepted by this automaton is free from hairpin structures. Kari et al. ([38]) formalizes these problems as *Hairpin-Freedom Problem* and *Maximal Hairpin-Freedom Problem* as follows:

Hairpin-Freedom Problem

Input: A non-deterministic automaton M.

Output: Yes/No depending on whether $L(M)$ is $hp(\theta, k)$-free.

Maximal Hairpin-Freedom Problem

Input: A deterministic automaton M_1 accepting a hairpin-free language, and a NFA M_2.

Output: Yes/No depending on whether there is a word $w \in L(M_2) \backslash L(M_1)$ such that $L(M_1) \cup \{w\}$ is $hp(\theta, k)$-free.

The following theorem provides an answer to the question of decidability of hairpin-freedom problems.

Theorem 2.2.5 *[38]*

1. *The hairpin-freeness problem for regular (respectively context-free) languages is decidable in linear (respectively cubic) time with respect to $|M|$.*

2. *The maximal hairpin-freeness problem for regular (respectively deterministic context-free) languages is decidable in time proportional to $|M_1| \cdot |M_2|$ (respectively $O((|M_1| \cdot |M_2|)^3)$).*

Kari et al., [38], also discusses other variants of hairpins such as scattered hairpin and hairpin frames and also the decidability problems for hairpin-freeness of these other variants.

Paun et al., [56], denotes by uH_k the language $hp(\theta, k)$ which contains words of the form $xvy\theta(v)z$ where $|v| \geq k$ and calls it *unrestricted hairpin language*. The authors of [56] also define restricted hairpin languages in which the restriction is put on the annealing site[2] of a strand. We will mention these restricted languages and some of the properties of complements of these languages, i.e., languages which are hairpin-free.

Definition 2.3 *[56] For an antimorphic involution θ,*

$uH_k = \{zvwxy \mid z, v, w, x, y \in \Sigma^*, x = \theta(v)$ *and* $|v| \geq k\}$,

$bH_k = \{vwxy \mid v, w, x, y \in \Sigma^*, x = \theta(v)$ *and* $|v| \geq k\}$,

$cH_k = \{zvxy \mid z, v, x, y \in \Sigma^*, x = \theta(v)$ *and* $|v| \geq k\}$,

$fH_k = \{zvwx \mid z, v, w, x \in \Sigma^*, x = \theta(v)$ *and* $|v| \geq k\}$,

$bcH_k = \{vxy \mid v, x, y \in \Sigma^*, x = \theta(v)$ *and* $|v| \geq k\}$,

$bfH_k = \{vwx \mid v, w, x \in \Sigma^*, x = \theta(v)$ *and* $|v| \geq k\}$,

$cfH_k = \{zvx \mid z, v, x \in \Sigma^*, x = \theta(v)$ *and* $|v| \geq k\}$,

$bcfH_k = \{vx \mid v, x \in \Sigma^*, x = \theta(v)$ *and* $|v| \geq k\}$

The hairpin constructions corresponding to languages in Definition 2.3 are illustrated in Figure 2.4.

Along with Proposition 2.2.4 which states that the languages $hp(\theta, k)$ and $hpf(\theta, k)$ are regular, results from [56] state that the languages bH_k, cH_k, fH_k and bfH_k in Definition 2.3 are regular as well. We mention some of the results from [56] which are related to the complements of languages defined above.

Theorem 2.2.6 *1. The complement of the language $bcfH_k$, for $k \geq 1$, is linear.*

2. The complements of languages bcH_k and cfH_k, for $k \geq 1$, are not context-free.

[2] A site where two WK-complementary single-stranded DNA sequences hybridize to form double-stranded DNA molecule

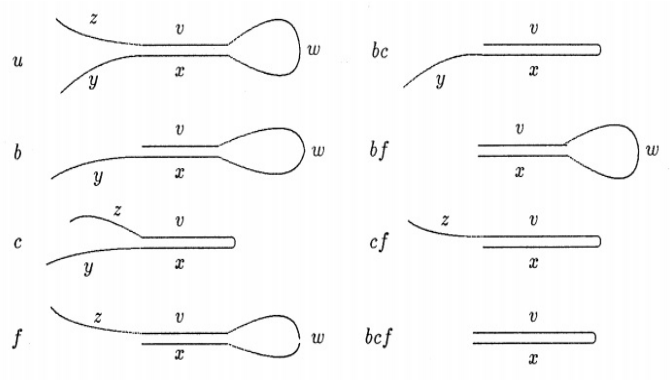

Figure 2.4: Hairpin constructions corresponding to the languages αH_k where $\alpha \in \{u, b, c, f, bc, bf, cf, bcf\}$, [56]

Kari et al., [40], calls the words of the language $bcfH_k$, θ-palindromes for (anti)morphic involutions θ. Formally, a word $x \in \Sigma^*$ is called a θ-palindrome if $x = \theta(x)$. Let P_θ denote the set of all θ-palindromes. The notion of θ-palindromes was independently introduced in [15]. We discuss the properties of P_θ in Section 2.5.

As discussed in the previous subsection, hairpins can be modelled as θ-bordered words with non-overlapping θ-borders. Hence, θ-unbordered words are the strings which will be free from hairpin structures. Let us look at the necessary and sufficient condition for a word to be θ-unbordered.

Proposition 2.2.7 *[39] Let θ be either a morphic or an antimorphic involution. Then for $u \in \Sigma^+$ such that $|u| \geq 2$, u is θ-unbordered iff $\theta(\text{PPref(u)}) \cap \text{PSuff(u)} = \emptyset$.*

The following proposition provides a necessary and sufficient condition for the catenation of θ-unbordered words to be θ-unbordered.

Proposition 2.2.8 *[39] Let θ be either a morphic or an antimorphic involution and let $u, v \in \Sigma^+$ be θ-unbordered. Then uv is θ-unbordered iff $\theta(\text{Pref(u)}) \cap \text{Suff(v)} = \emptyset$.*

Also, for an antimorphic involution θ on Σ^*, the set of all θ-unbordered words $D_\theta(1)$ is regular. A result from [39] states that it is decidable for a given non-empty word whether or not it belongs to $D_\theta(1)$.

We have seen another type of structure formed due to intra-molecular hybridization of RNA strands, pseudoknots. It is clear that θ-pseudoknot-unbordered (θ-pk-unbordered) words will be free from pseudoknot-like secondary structures. The following result gives a necessary and sufficient condition for a word to be θ-pk-unbordered.

Proposition 2.2.9 *[43] Let θ be an antimorphic involution on Σ^*. Then for $u \in \Sigma^+$, u is θ-pk-unbordered iff $\theta(\text{cp}(\text{Pref}(u)) \cap \text{Suff}(u) = \emptyset$.*

Also, for an (anti)morphic involution θ on Σ^*, the set of all θ-pk-unbordered words over Σ^*, $K_\theta(1)$, is a subset of set of all θ-unbordered words, $D_\theta(1)$, and a dense set, [43].

2.2.2 Inter-molecular hybridizations

In this section, we explore some secondary structures that result from inter-molecular hybridizations of DNA strands, and solutions preventing the formation of such structures using formal languages as a tool.

Let us begin with the notion of θ-bordered and θ-pseudoknot-bordered words defined in the earlier section. Recall that a θ-bordered words with non-overlapping θ-borders can form a hairpin-like structure. However, if such a word has overlapping θ-borders then it can interact with another copy of itself forming a secondary structure as shown in Figure 2.5. Similarly, RNA strands can form pseudoknot-like inter-molecular structures such as those depicted in Figure 2.6. We have already seen some properties of the set of all θ-unbordered and θ-pseudoknot-unbordered words in the previous subsection.

If a DNA strand involved in the computation, say v, is WK-complementary to part of some other strand, say u, in the computation, then this results in a secondary structure as shown in Figure 2.7.

Similarly, if subwords of two words are WK-complements of each other, then this results into a secondary structure as shown in Figure 2.7(a). The structure shown in Figure 2.7(b) is a result of the hybridization between a word and the catenation of two other words.

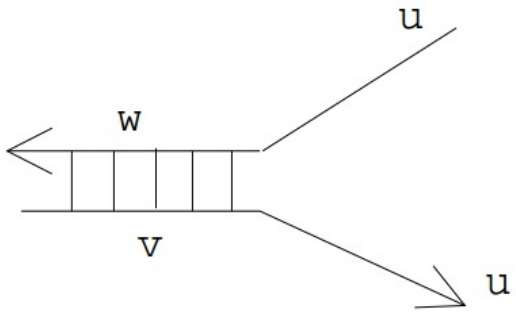

Figure 2.5: A θ-bordered word u over the DNA alphabet, with overlapping θ-border v where $w = \theta(v)$ [39]

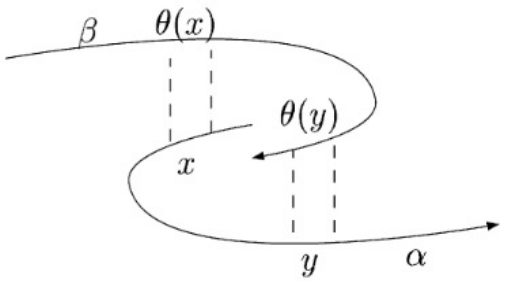

Figure 2.6: Pseudoknot-like structure formed due to inter-molecular hybridization between words $\beta\theta(x)\theta(y)$ and $xy\alpha$ [43]

The following definition gives the properties of languages that need to be satisfied for the words of the language to avoid the above-mentioned undesired inter-molecular hybridizations.

Definition 2.4 *For an (anti)morphic involution θ, the language L is called:*

1. *[34] θ-nonoverlapping if $L \cap \theta(L) = \emptyset$;*
2. *[34] θ-compliant if $\forall w \in L, x, y \in \Sigma^*, w, x\theta(w)y \in L \Rightarrow xy = \lambda$;*
3. *[34] θ-p-compliant if $\forall w \in L, y \in \Sigma^*, w, \theta(w)y \in L \Rightarrow y = \lambda$;*
4. *[34] θ-s-compliant if $\forall w \in L, x \in \Sigma^*, w, x\theta(w) \in L \Rightarrow x = \lambda$;*
5. *[34] strictly θ-compliant if $\forall w \in L, x, y \in \Sigma^*, w, x\theta(w)y \in L \Rightarrow xy = \lambda$ and $w \neq \theta(w)$;*
6. *[31] θ-free if $L^2 \cap \Sigma^+\theta(L)\Sigma^+ = \emptyset$;*

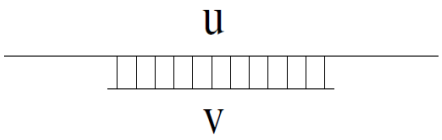

Figure 2.7: Inter-molecular hybridization between two strands u and v, $\theta(v)$ being a subword of u [34]

Figure 2.8: Undesired inter-molecular hybridizations, (a): two words that have WK-complementary subwords, (b): a word that is WK-complementary to the catenation of two other words [37]

7. *[35] θ-sticky-free if* $\forall w \in \Sigma^+, x, y \in \Sigma^*, wx, y\theta(w) \in L \Rightarrow xy = \lambda$;

8. *[35] θ-3'-overhang-free if* $\forall w \in L, x, y \in \Sigma^* wx, \theta(w)y \in L \Rightarrow xy = \lambda$;

9. *[35] θ-5'-overhang-free if* $\forall w \in L, x, y \in \Sigma^* xw, y\theta(w) \in L \Rightarrow xy = \lambda$;

10. *[35] θ-overhang-free if L is both θ-3'-overhang-free and θ-5'-overhang-free;*

11. *[33] $\theta(k, m_1, m_2)$-subword compliant if* $\forall u \in \Sigma^*$ *such that* $\forall u \in \Sigma^k$ *we have* $\Sigma^* u \Sigma^m \theta(u) \Sigma^* \cap L = \emptyset$ *for* $m_1 \leq m \leq m_2$;

12. *[33] θ-k-code if* $Sub_k(L) \cap Sub_k(\theta(L)) = \emptyset$ *for some* $k > 0$.

Note that θ-compliant languages avoid the situation in Figure 2.7, θ-k-codes avoid the situation in Figure 2.8(a), and θ-free languages avoid situation in Figure 2.7(b). Also, θ-p-compliant and θ-s-compliant languages avoid some special case of situations in Figure 2.7 and 2.8(a). Languages which satisfy properties 11 ($\theta(k, m_1, m_2)$-subword compliant) and 12 (θ-k-code) from Definition 2.4 avoid hairpin-like structures with restriction on the length of a stem of

a hairpin. Figure 2.9 depicts the unwanted hybridizations that other languages in Definition 2.4 avoid.

The following proposition shows the relationship between θ-sticky-free and θ-compliant languages.

Proposition 2.2.10 *[35] For every language $L \subseteq \Sigma^+$ and for every given (anti)morphic involution $\theta : \Sigma^+ \to \Sigma^+$, the following are equivalent:*

1. *L is θ-sticky-free;*

2. *$\theta(L)$ is θ-sticky-free;*

3. $\text{PPref(L)} \cap \theta(\text{PSuff(L)}) = \emptyset$ *and L is both θ-p-compliant and θ-s-compliant.*

For an antimorphic involution θ, a language which is θ-compliant and either θ-3'-overhang-free or θ-5'-overhang free is θ-free, [35]. Figure 2.9 shows an overview of the relationships between the aforementioned DNA languages, where arrows indicate the inclusion relation among classes of languages that satisfy certain properties. For example, a language that is θ-p-compliant is θ-3'-overhang free. For further details about the properties of DNA languages defined in Definition 2.4, such as closure properties and relationship among these languages, the reader is referred to [31, 33, 34, 35].

Kari et al., [37], introduced a general framework of *bond-free property* to analyse if a given DNA language is free from certain type of undesirable bonds. A property $\mathcal{P}$ is a mapping $\mathcal{P} : 2^{\Sigma^*} \to \{\text{true}, \text{false}\}$. A language L satisfies the property $\mathcal{P}$ if $\mathcal{P}(L) = \text{true}$.

Definition 2.5 *[37] A language property $\mathcal{P}$ is called a bond-free property of degree 2 if there exists binary word operations $\diamond_{lo}, \diamond_{up}$ and an antimorphic involution θ such that for an arbitrary $L \subseteq \Sigma^*$, $\mathcal{P}(L) = true$ iff*

$$\forall w \in \Sigma^+, x, y \in \Sigma^*, (w \diamond_{lo} x \cap L \neq \emptyset, w \diamond_{up} y \cap L \neq \emptyset) \Rightarrow xy = \lambda,$$

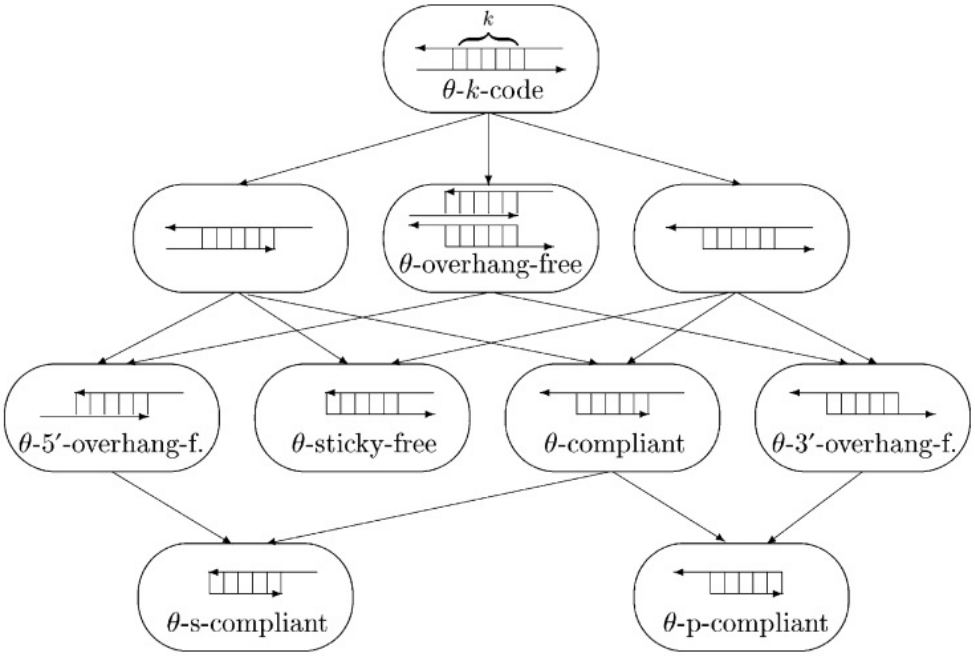

Figure 2.9: Classes of languages free from certain types of undesired hybridization [44]

where a binary word operation is a mapping $\diamond : \Sigma^* \times \Sigma^* \to 2^{\Sigma^*}$, *where* 2^{Σ^*} *is the set of all subsets of* Σ^*.

To study the bond-free properties of various languages, the tool that is used is word operation on trajectories. Consider a trajectory alphabet $V = \{0, 1\}$ and assume $V \cap \Sigma = \emptyset$. A trajectory is any string $t \in V^*$ which specifies the way in which an operation $\diamond$ is applied to the letters of its two operands. As an example, we mention the definition of the binary word operation *shuffle on trajectories*.

Definition 2.6 *[37] Let* $t \in V^*$ *be a trajectory and let* $\alpha, \beta \in \Sigma^*$. *Then, the shuffle of* α *with* β *on the trajectory t, denoted by* $\alpha \sqcup\!\sqcup_t \beta$ *is defined as follows:*

$$\alpha \sqcup\!\sqcup_t \beta = \{\alpha_1\beta_1 \dots \alpha_k\beta_k | \alpha = \alpha_1 \dots \alpha_k, \beta = \beta_1 \dots \beta_k, t = 0^{i_1}1^{j_1} \dots 0^{i_k}1^{j_k},$$
$$\textit{where } |\alpha_m| = i_m \textit{ and } |\beta_m| = j_m \textit{ for all } m, 1 \leq m \leq k\}.$$

We mention the following results from [37].

Theorem 2.2.11 *[37] The languages properties 2, 3, 4, 7, 8, 9 in Definition 2.4 are bond-free properties. Moreover, the associated sets of trajectories T_{lo}, T_{up} are regular where $\diamond_{lo} = \sqcup\sqcup_{T_{lo}}$, $\diamond_{up} = \sqcup\sqcup_{T_{up}}$.*

Theorem 2.2.12 *[37] Let $\mathcal{P}$ be a bond-free property associated with the regular set of trajectories T_{lo}, T_{up}. Then the following problem is decidable in quadratic time:*

Input: an NFA A.

Output: Yes/No depending on whether L(A) satisfies $\mathcal{P}$.

For more properties of bond-free languages the reader is referred to [36, 37].

2.3 DNA codeword design problem: other approaches

In Section 2.2, we have seen a theoretical approach to address the DNA codeword design problem. In this section, we explore a few other approaches including algorithmic, software simulation and the construction of *in vitro* DNA libraries.

In [5, 53, 58], authors considered the use of a restricted genetic alphabet in order to reduce the chances of undesirable secondary structure formations, in particular, the ones that are formed due to inter-molecular hybridization of DNA strands. Brenner et al., [5], observed that all the strands in the library that use all the four nitrogen bases, A, C, G, T increase the chance of secondary structure formation to a great extent. Motivated by this observation, [58] constructed DNA libraries using the restricted DNA alphabet $\{A, C, T\}$, with the help of a set of programs written in C++.

In order to avoid undesirable inter- and intra-molecular hybridizations it is clear that the strands involved should be as dissimilar as possible. Hence, it was intuitive to consider a restriction on the Hamming distance between the strands. The Hamming distance $H(w_1, w_2)$, for two equi-length words w_1 and w_2, is the number of positions in which the words w_1 and w_2 differ. The Hamming distance constraint poses the condition on any two words w_1 and w_2 to have $H(w_1, w_2) \geq d$. Also, the Hamming distance between a word w_1 and WK-complement of

w_2 needs to be $H(w_1, WK(w_2)) \geq d$. In addition, due to the parallelism in DNA computation, another condition that usually needs to be satisfied is that all the strands should have the similar melting temperatures[3]. Several software simulation packages and algorithms have been constructed that consider the Hamming distance, melting temperature and some other conditions to design DNA codewords and build DNA libraries that are free from undesirable secondary structures (see, e.g., [2, 21, 29, 51, 65, 66]).

For example, [51] proposed a dynamic programming algorithm to calculate the total number of words of some specified length n which satisfy the following four constraints and randomly output such a word. Firstly, the Hamming distance between all pairs of distinct words w_1 and w_2 should satisfy $H(w_1, w_2) \geq d$, secondly, the Hamming distance between the complement of one strand, w_1^C and reverse of another strand, w_2^R should be $H(w_1^C, w_2^R) \geq d$, thirdly, all the pairs of strands w_1, w_2 should satisfy $H(w_1, w_2^R) \geq d$ and lastly, all the strands should have a certain value of free energy $\triangle G$[4]. The time complexity of this algorithm, as reported by authors, is $O(n^2)$.

DNASequenceGenerator, a software tool created by [19], is capable of creating DNA sequences that meet user's requirements of melting temperature value, GC ratio values and uniqueness. The GC ratio is the percentage of nucleotides G or C present in each word, whereas uniqueness requests that any subsequence of certain length is allowed to occur at most once in the pool. Another tool, *DNASequenceCompiler*, [18], is very similar to DNASequenceGenerator. It translates formal grammars into DNA molecules representing the rules of the grammar. This compiler translates each rule of the grammar as a partially double-stranded DNA molecule where the double-stranded part represents a terminal letter, and the single-stranded "sticky-ends" represent variables. The parse module of the compiler reads the symbol sets and the rules of the grammar, and the physical and chemical requirements for the sequences. The generator module (DNASequenceGenerator) generates the DNA sequences (of the form men-

[3]The melting temperature is the temperature at which half of the strands of DNA are in double helical structure and the rest are in a dissociated state, i.e., they exists as two independent single strands.

[4]The (Gibbs) free energy G is usually given by the formula $G = H - TS$, where H is the enthalpy, S is the entropy (measure of disorder) and T is the temperature.

tioned above). In order to produce the final word, these DNA sequences, which represent the rules of the grammar, need to be concatenated and the concatenated DNA sequences need to be unique. The coordinating module takes care of the uniqueness requirement for sequences.

Garzon, [23], observed that the set of molecules/strands produced by *in silico* methods is relatively smaller in size compared to those produced by *in vitro* methods. Deaton et al., [16], proposed a Polymerase Chain Reaction[5] (PCR)-based protocol (Figure 2.10) to select a library of non-cross-hybridizing oligonucleotides[6] (shortly, oligos) *in vitro*. The key concept is to regulate the temperature of the reaction so as to amplify (multiply) the desired DNA molecules, i.e., the oligonucleotides which are maximally mismatched, with the help of PCR. The initial population of equi-length oligos consisted of random sequences with the specific primers P1 and WK-complement of P2, i.e., $P2^C$ attached to them at either ends (the same for all strands). In the subsequent computation, by regulating the temperature, only strands that are not perfect WK-complements of each other melted apart and were amplified using PCR and hence the test tube ultimately contained a non-cross(self)hybridizing set of oligos. It was observed that, at lower temperatures, maximally mismatched oligos were amplified over the other oligos which are perfectly matched or had lower degree of mismatches.

Nuser et al., [55], proposed a computer simulation of the above mentioned PCR-based protocol to gain insight about the behaviour of the protocol and to explore the computational capability of the same. Instead of representing DNA words by a DNA sequence, the simulation experiment rather represents such words by their concentration, i.e., the number of such words present in the test tube and a vector of pairwise hybridization energies[7] with all other words in the test tube. This facilitates the analysis of the result produced by the simulation in terms of the number of maximally mismatched words produced. Initially, all the words in the test tube had equal concentration. The simulation chose two random words according to their relative concentrations in the test tube and each such pair was analysed for their hybridization energies.

[5]PCR is a technique used in molecular biology to amplify DNA molecule(s),that is, to generate many copies of the particular DNA sequence with the help of DNA polymerase enzyme.

[6]Short, approximately 15-20bp DNA molecules

[7]Here hybridization energy represents the strength of interaction between two words

It was observed that if this (hybridization) energy value was less than the threshold of 0.5, the two words hybridized with each other and hence they were not selected. Thus, only those words with pairwise hybridization energy greater than 0.5 were selected and amplified, since they did not hybridize with each other. The authors suggest that a modification in the protocol can result into making the protocol useful for computations and, as an example, they generated the sequence of Fibonacci numbers. Furthermore [7] attempted to characterize the library of oligonucleotide that was generated by the PCR-based protocol and simulation experiment.

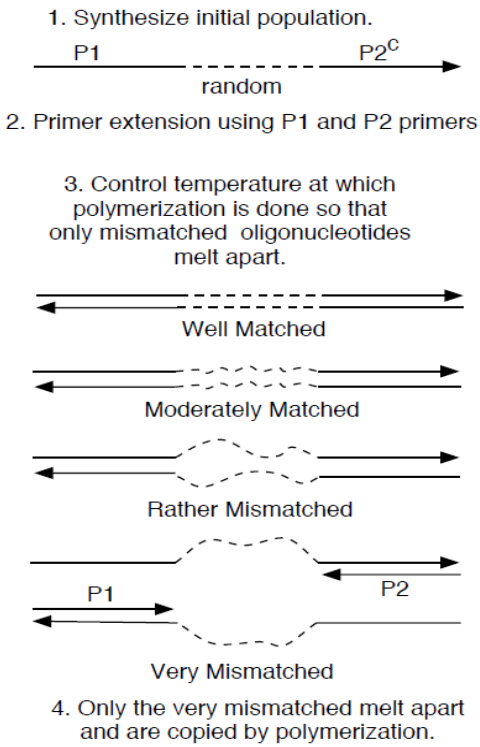

Figure 2.10: Protocol to select maximally mismatched oligonucleotides, starting with a population of strands with primer pair P1 and $P2^C$, which amplifies only very mismatched oligonucleotides [16]

For a more detailed and exhaustive review of solutions, non-theoretical, in particular, to the codeword design problem, we refer reader to [23, 52].

As we have seen in this and earlier sections, many attempts have been made to find the optimal solution to the codeword design problem for DNA computing experiments. Note that, according to [57], the solution to the general codeword design problem is NP-complete.

2.4 DNA memory

As seen in earlier sections, extensive work was done to find a good encodings of DNA strands considering various combinatorial and thermodynamic constrains. The next major step is to store these encoded data effectively in reliable media, resistant to external factors, as well as allowing easy and unambiguous retrieval.

Most of the work in the area of molecular memory is around the aim of building a content-addressable memory[8] using DNA. Eric Baum [3] was the first one to propose the idea of a content-addressable DNA memory. Out of many approaches that he suggested in his paper, one was to store binary words of a fixed length. He suggested to use two distinct single-stranded DNA molecules to encode the bit "1" and the bit "0" and that, in order to obtain a DNA molecule encoding for specific binary word, appropriate DNA sequences can be concatenated. To retrieve the required data from the memory, the technique to be used is to introduce complementary sequences to the address correspondent of the data to be searched, attached to magnetic beads. Thus, these complementary subsequences can then bind to the corresponding sequence in the memory and such molecules could be further extracted and sequenced in order to read the stored data.

Subsequently, several other attempts have been made to store data on DNA, including modelling of DNA memories with the help of computer simulation (*in silico*) [28, 64], hairpin DNA memories [63], and some *in vitro* and *in vivo* experiments. In this section, we particularly explore the Nested Primer Molecular Memory (*in vitro*) experiment and some organic (*in vivo*) DNA memory experiments.

2.4.1 *In vitro* DNA memory

The first use of *in vitro* memory was demonstrated by Adleman's experiment ([1]) to solve a 7-node instance of the Hamiltonian Path Problem wherein the vertices were encoded as suitable

[8] A memory where the data is located by the content of its address rather than by location.

DNA strands and some operations such as PCR and gel electrophoresis were performed to find the solution to the problem. Subsequently, several attempts have been made in order to attain Baum's dream of a content-addressable memory (e.g., see [8, 9, 22, 54, 69]). In [9], the authors stored a book with 53,426 words, 11 JPG images and 1 JavaScript program on a DNA microchip. The book was first converted to an html format and then encoded as DNA strands by using an encoding scheme where 0 was represented as the bases A and C and 1 was represented as the bases G and T. Also, a 19-bit binary sequence was used for addressing purposes and the data was read and retrieved using next-generation DNA sequencing[9].

Recently, Goldman et al., [27] successfully encoded and decoded 154 Shakespeare's sonnets (ASCII text), a scientific paper (PDF format), a medium-resolution coloured photograph (JPEG 2000 format) and 26-second excerpt from Martin Luther King's 1963 'I have a dream' speech (MP3 format). In the encoding scheme, they first replaced each byte of ASCII text with five or six base-3 digits (trit), using a Huffman code, and each trit was in turn converted to a DNA letter using an encoding scheme which ensures that no two nucleotides appear consecutively. The authors reported successful and unambiguous retrieval of all the files using DNA sequencing procedures.

Nested Primer Molecular Memory (NPMM)

The *in vitro* DNA memory model proposed by [45] is one of the best examples of implementation of the idea of content-addressable memory proposed by Baum. The model uses the simple operation of nested PCR. In the proposed model, each DNA strand consists of three types of blocks, a data block (the site for storing the encoded data over the DNA alphabet), the address block (the site for specifying the address of the data block, namely the A block, the B block and the C block) and the Re block (the site for the reverse primer to hybridize). The memory

[9] A term that is used to describe number of different modern, high-throughput DNA sequencing technologies.

capacity, M of NPMM is calculated as follows:

$$M(bit) = 2 \times Data(bp) \times Primer^{Block}$$

where *Data* is the length of the sequence in the data block, *Block* is the number of address blocks and *Primer* is the number of primers in each address block. In the initial NPMM experiment, [45], the authors could extract a single target DNA strand from the diluted solution of 27 strands (3 address blocks with 3 sequences in each address block). In the subsequent experiment, the authors of [46] could extract a single DNA strand from the diluted solution of 12,167 strands (3 address blocks with 23 sequences in each address block). Finally, the authors of [69] were able to retrieve the target DNA strand from the diluted solution of 16.8M strands (6 address blocks with 16 sequences in each block). We will briefly explore the NPMM model that was used to achieve this huge address space.

The NPMM consists of three layers of address space on each side of the data space (CL_i, BL_j, AL_k on the left and AR_l, BR_m, CR_n on the right) with sixteen 20-mer sequences in each layer, hence it is of the form $[CL_i, BL_j, AL_k, \text{DATA}, AR_l, BR_m, CR_n]$, where $i, j, k, l, m, n \in \{0, 1, \ldots, 15\}$ and DATA is either a 20-mer, a 40-mer or a 60-mer DNA sequence. As an example, we will describe the working of NPMM for the address

$$[CL_3, BL_0, AL_{10}, \text{DATA}, AR_{12}, BR_4, CR_1].$$

In the first step, PCR is performed using the primer pair CL_3 and WK(CR_1) and as a result the solution will have in large quantity molecules containing only CL_3 and CR_1. In the next step, considering only the amplified molecules obtained from first step, PCR is performed using the primer pair BL_0 and WK(BR_4). In the last step, PCR is performed using AL_{10} and WK(AR_{12}) and as a result, the solution will contain only those DNA molecules which have the above mentioned address, from which the data can then be retrieved by sequencing and decoding. The authors also proposed a solution to a combinatorial optimization problem demonstrating

the limitation for NPMM's capacity. One major disadvantage of NPMM is the occurrence of mutations that can happen during PCR.

2.4.2 Organic DNA memory

So far we have seen memory models which are either implemented using computer simulation experiments or test tube experiments. In this section, we briefly explore memory models implemented using living organisms as hosts to store the encoded data. One of the major obstacles in using living organisms as a host can be the lack of substantial knowledge about the cellular and molecular mechanism of the host organism, since inadequate knowledge about these mechanisms can lead to the misinterpretation of foreign DNA sequences by these organisms which subsequently can kill the host organism. Since the molecular mechanism of many bacteria is well known, bacteria are one of the widely used hosts .

The authors of [68] identified two such hosts in *Escherichia coli* (*E.coli*) and *Deinococcus radiodurans* (*Deinococcus*) as the cellular and molecular mechanism of these bacteria is well understood, and the latter can survive in extreme conditions such as cold, dehydration, vacuum, acid and radiation and hence can be an ideal host candidate. The encoding scheme that was used was to assign 3-mer sequences to numbers and various symbols in the English alphabet. For example, "1" was encoded as AAC, the letter "A" was encoded as AGG, etc. The next step was to identify two fixed size DNA sequences (20-base-pair long) with the condition that they should not occur in the bacterial genome yet they should satisfy all the genomic constraints so that the introduction of such sequences should not cause any mutation, or kill the bacteria. Another condition that had to be satisfied so as to preserve the integrity of the message without killing the bacterium, was the introduction of stop codons in these DNA sequences so that the bacterium would not misinterpret the embedded message as a protein-coding sequence. As a first step, two 46bp long complementary oligonucleotides consisting of two different 20bp long DNA sequences connected by a 6bp long restriction enzyme[10] site were created. In addition,

[10]DNA-cutting enzymes

this (restriction enzyme) site served as the location for the encoded message to be inserted in the subsequent computation. Then this double stranded DNA molecule was cloned into a recombinant plasmid[11] as shown in Figure 2.11.

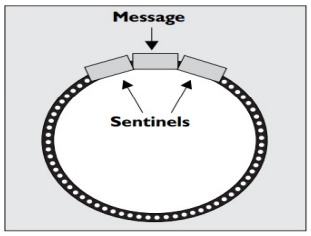

Figure 2.11: A recombinant plasmid with two DNA fragments as sentinels protecting the encoded message in between, [68]

This embedded DNA was then inserted into cloning vectors[12] which then were transferred into *E.coli* by high voltage shocks. As a next step, the cloning vectors and encoded DNA were incorporated into the genome of *Deinococcus* and retrieved by PCR. The authors reported the successful storage and retrieval of seven chemically synthesized DNA fragments with 57-99 base pairs of non-native information.

Other host organisms suggested (but not experimentally verified) by [11] were *Bacillus subtilis* (*B. subtilis*) and *Saccharomyces cerevisiae* (*S. cerevisiae*). The spore-forming capacity of these microorganisms is believed to be a protected medium, since it is a resistant structure that bacteria use for survival in unfavourable conditions. Also, the molecular genetics of these species is well-known, making them suitable hosts. Along with this, the authors also suggested a few possible encodings to encode the message, as shown in Figure 2.12. Three encodings were suggested: encode a complete word by a DNA sequence, encode each syllable of a word by a different DNA sequence, or encode each letter of a word by a different DNA sequence.

The authors of [62] proposed some codes that can be useful for encrypting data in DNA and name them *Huffman code*, *comma code* and *alternating code*. The Huffman code is constructed using Huffman's method which is based on the fact that some letters like a, e, s are used more frequently than letters q, z, and hence encodes the former by short k-mers such as AT, T, GT (for

[11] A union of foreign DNA molecules inserted into a circular DNA molecule
[12] A circular DNA molecule that can self-replicate within a bacterial host

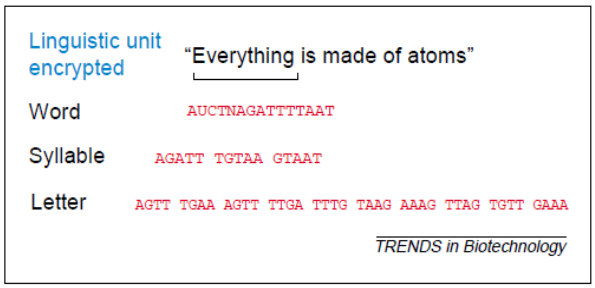

Figure 2.12: Encryption of the first word of Richard Feynman's suggested message to future civilizations [11]

a, e and s respectively) and the latter by comparatively longer k-mers such as CCCTA, CCCTG (for q and z respectively). In the comma code, consecutive 5-base codons are separated by a single uniform base which does not occur in 5-base codon, e.g., A_____A_____A_____A. This kind of design would help the user to orient the message even if the starting point is not mentioned, and it is effective in detecting insertion and deletion mutations. An alternating code consists of sixty four 6-base long alternating sequences of purines (A and G) and pyrimidines (C and T), e.g., PQPQPQ where P=A or G and Q=C or T. Even though the comma code and the alternating code are not economical to use (unlike the Huffman code), they are more suitable for encoding data for long-term storage.

2.5 DNA computing inspired combinatorics on words

The mathematical formalization of DNA WK-complementarity as an antimorphic involution has inspired generalizations of many classical and fundamental notions of formal languages and combinatorics on words including conjugacy, commutativity, borderedness, periodicity, palindromic property, etc. In this section, we discuss some generalizations of the above mentioned concepts, some of their properties, and generalization of two important results from combinatorics on words, namely Fine and Wilf's theorem and the Lyndon-Schützenberger equation. Note that, in this section, θ always denotes an (anti)morphic involution unless otherwise specified.

Recall that a word and its WK-complement encode the same information and that one can be obtained from the other by an application of an antimorphism that interchanges A with T and G with C and vice versa. Thus, it is natural to consider the notion of repetitions not being limited to just a finite concatenation of a word with itself, but rather a finite concatenation of a word and its WK-complement in some random order, thereby extending the notion of repetitions to pseudo-repetitions. [14] defines the θ(pseudo)-power of a word u as a word of the form $u_1u_2 \dots u_n$ where $u_1 = u$ and $u_i \in \{u, \theta(u)\}$ for $2 \leq i \leq n$. As the notion of repetition leads to the notion of primitivity, the notion of pseudo-repetitions can lead to the concept of a θ-primitive word. A word is said to be primitive if it cannot be expressed as a power of any other word and a word is said to be θ-primitive if it cannot be expressed as a θ-power of any other word, [14]. If $w \in \{u, \theta(u)\}^*$ such that u is a smallest such word, then u is said to be the θ-primitive root of w and is denoted by $\rho_\theta(w)$.

Example 2.3 *Let θ be an antimorphic involution on $\Sigma = \{A, C, G, T\}$ such that $\theta(A) = T$, $\theta(G) = C$ and vice versa. Then $w = GTCGTCGAC = (GTC)(GTC)\theta(GTC)$ is not θ-primitive, whereas $v = GTAG$ and $u = GTC$ are θ-primitive.*

The following result states the relationship between primitive and θ-primitive words.

Proposition 2.5.1 *[14] If a word $w \in \Sigma^+$ is θ-primitive then it is also primitive. The converse is not always true.*

Note that the θ-primitive root of a word is θ-primitive. The notion of θ-primitive words leads to a generalization of the classical result of Fine and Wilf. We will first state the classical result.

Theorem 2.5.2 *[20] Let $u, v, \in \Sigma^*$, $|u| = n, |v| = m$ and $d = gcd(n, m)$[13]. If two powers u^i and v^j of u and v have a common prefix of length at least $n + m - d$, then u and v are powers of a common word. Moreover, the bound $n + m - d$ is optimal.*

[13] $\gcd(n, m)$ denotes the greatest common divisor of integers n and m respectively.

Theorem 2.5.3 *[14] Let $\theta : \Sigma^* \to \Sigma^*$ be a morphic involution, $u, v \in \Sigma^+$ with $n = |u|, m = |v|$ and $d = gcd(n, m)$, $\alpha(u, \theta(u)) \in u\{u, \theta(u)\}^*$ and $\beta(v, \theta(v)) \in v\{v, \theta(v)\}^*$. If two θ-powers $\alpha(u, \theta(u))$ and $\beta(v, \theta(v))$ have a common prefix of length at least $n + m - d$, then there exists a word $t \in \Sigma^+$ such that $u, v \in t\{t, \theta(t)\}^*$, i.e., $\rho_\theta(u) = \rho_\theta(v)$. Moreover, the bound $n + m - d$ is optimal.*

Theorem 2.5.4 *[14] Let $\theta : \Sigma^* \to \Sigma^*$ be an antimorphic involution and $u, v \in \Sigma^+$ be such that $|u| > |v|$. If there exists two θ-powers $\alpha(u, \theta(u)) \in u\{u, \theta(u)\}^*$ and $\beta(v, \theta(v)) \in v\{v, \theta(v)\}^*$ sharing a common prefix of length $2|u| + 2|v| - gcd(|u|, |v|)$, then $\rho_\theta(u) = \rho_\theta(v)$. Furthermore, this bound is optimal.*

We know that every non-empty word is a unique power of a unique primitive word. We state the similar result concerning the uniqueness of θ-primitive words.

Theorem 2.5.5 *[14] For any word $w \in \Sigma^+$ there exists a unique θ-primitive word $t \in \Sigma^+$ such that $w \in t\{t, \theta(t)\}^*$, i.e., $\rho_\theta(w) = t$.*

The study of θ-periodicity has motivated researchers to consider further generalizations of the concept of θ-periodicity, replacing the (anti)morphic involution with some more general functions such as literal, erasing and uniform homomorphisms, [24, 25, 26]. Also, this notion was independently generalized to periodic-like words [6], pseudoperiodic words [4], weakly-periodic words [12] also known as Abelian periodic words [10].

The study of primitive words has inspired the study for a solution of a well-known equation

$$a^m = b^n c^p \text{ where } m, n, p \geq 2$$

known as the Lyndon-Schützenberger equation, [48]. The following result demonstrates the solution to the Lyndon-Schützenberger equation.

Theorem 2.5.6 *[48] If words u, v, w satisfy the relation $u^l = v^n w^m$ for some positive integers $l, m, n \geq 2$, then they are all powers of a common word, i.e., there exists a word t such that $u, v, w, \in \{t\}^*$.*

Czeizler et al., [13], initiated the study of a generalization of the Lyndon-Schützenberger equation to accommodate θ-powers of a word. The equation that authors have considered is

$$u_1 \dots u_l = v_1 \dots v_n w_1 \dots w_m$$

where $u_1, \dots, u_l \in \{u, \theta(u)\}$, $v_1, \dots, v_n \in \{v, \theta(v)\}$ and $w_1, \dots, w_m \in \{w, \theta(w)\}$ for $l, m, n \geq 2$. We mention the following result as a special case of the solution to the equation where $n, m \geq 3$ and $l \geq 6$, whereas the Table 2.1 summarizes the remaining results proved in [13, 42, 49, 50].

Theorem 2.5.7 *Let $u, v, w \in \Sigma^+$, $n, m \geq 3$, $l \geq 6$, $u_i \in \{u, \theta(u)\}$ for $1 \leq i \leq l$, $v_j \in \{v, \theta(v)\}$ for $1 \leq j \leq n$ and $w_k \in \{w, \theta(w)\}$ for $1 \leq k \leq m$. If $u_1 \dots u_l = v_1 \dots v_n w_1 \dots w_m$, then there exists a word $t \in \Sigma^+$ such that $u, v, w \in \{t, \theta(t)\}^+$.*

A word $u \in \Sigma^*$ is called a *conjugate* of a word $w \in \Sigma^*$ if there exists a word $v \in \Sigma^*$ such that $uv = vw$. The notion of conjugacy was extended to the notion of θ-conjugacy by [40]: if θ is either a morphic or an antimorphic involution then a word u is *θ-conjugate* of another word w if $uv = \theta(v)w$ for some $v \in \Sigma^*$.

Example 2.4 *Let θ be an antimorphic involution on $\Sigma = \{A, C, G, T\}$ such that $\theta(A) = T$, $\theta(G) = C$ and vice versa. Then $u = ACCT$ and $w = CTGT$ are θ-conjugates of each other for $v = GT$ since $(ACCT)(GT) = \theta(GT)(CTGT)$.*

Note that the θ-conjugacy relation for a morphic involution is transitive, whereas for an antimorphic involution it need not be transitive. The following proposition provides the characterization of θ-conjugate words.

Proposition 2.5.8 *[40] Let u be a θ-conjugate of w such that $uv = \theta(v)w$ for some $v \in \Sigma^*$. Then*

1. *For a morphic involution θ there exists $x, y \in \Sigma^*$ such that $u = xy$ and one of the following hold:*

Table 2.1: Solutions to the extended Lyndon-Schützenberger equation

l	m	n	θ-periodicity
≥ 4	≥ 3	≥ 3	YES ([13, 42])
3	≥ 12	≥ 12	YES ([49])
3	$5 \leq \min\{m, n\}$ m or n odd		YES ([49])
3	$5 \leq \min\{m, n\} < 12$ m or n even		YES ([50])
3	4	≥ 5 and odd	YES ([50])
3	4	≥ 4 and even	NO ([42])
3	3	≥ 3	NO ([42])
≥ 3	2	≥ 2	NO ([13])
		one of $\{l, m, n\}$ equals 2	NO ([13, 42])

(a) $w = y\theta(x)$ *and* $v = (\theta(x)\theta(y)xy)^i\theta(x)$ *for some* $i \geq 0$.

(b) $w = \theta(y)x$ *and* $v = (\theta(x)\theta(y)xy)^i\theta(x)\theta(y)x$ *for some* $i \geq 0$.

2. *For an antimorphic involution* θ*, there exists* $x, y \in \Sigma^*$ *such that either* $u = xy$ *and* $w = y\theta(x)$*, or* $w = \theta(u)$.

According to Proposition 2.5.8, for an antimorphic involution θ, if two words u and w are θ-conjugates of each other, then one of the possibilities is that $w = \theta(u)$, and hence the existence of a word and its θ-conjugate in the computation can lead to the formation of undesirable secondary structures.

Corollary 2.5.9 *[40] For a morphic involution* θ *on* Σ^**,* θ*-conjugacy on words is a symmetric relation.*

A word u is said to *commute* with the word v if $uv = vu$, [61]. Similarly, a word u is said to θ*-commute* with the word y if $xy = \theta(y)x$, [40]. Hence the existence of words that θ-commute with each other can lead to the formation of undesirable secondary structures.

Example 2.5 *Let* θ *be an antimorphic involution on* $\Sigma = \{A, C, G, T\}$ *such that* $\theta(A) = T$*,* $\theta(G) = C$ *and vice versa. Then the two words* $x = AT$ *and* $y = CGAT$ θ*-commute since* $(AT)(CGAT) = \theta(CGAT)AT$.

Let us denote the θ-commutativity order by $v \leq_c^\theta u$ iff $u = vx = \theta(x)v$ for some $x \in \Sigma^*$, $C_\theta(1) = \{u \in \Sigma^+ | v \leq_c^\theta u \Leftrightarrow v = u\}$, and by $L_c^\theta(u) = \{v | v \in \Sigma^*, v \leq_c^\theta u\}$ the set of all words that θ-commute with a word $u \in \Sigma^*$. We discuss some results related to θ-commutativity, important from a combinatorial perspective.

Lemma 2.5.10 *[40] For an antimorphic involution θ and $u \in \Sigma^+$, $L_c^\theta(u)$ is a totally ordered set with $\leq_c^\theta$.*

The following proposition characterizes the words that θ-commute with each other.

Proposition 2.5.11 *[40] Let $u, v \in \Sigma^+$ be such that u θ-commutes with v, i.e., $uv = \theta(v)u$.*

1. *If θ is an antimorphic involution then $u = x(yx)^i$, $v = yx$ where $i \geq 0$ and u, x, y are θ-palindromes where $x \in \Sigma^+$, $y \in \Sigma^*$.*

2. *If θ is a morphic involution then $u = x(yx)^i$ and $v = yx$ where $yx = \theta(x)\theta(y)$ and $i \geq 0$ with $x \in \Sigma^+$, $y \in \Sigma^*$.*

For an antimorphic involution θ, the set $L = \Sigma^* \backslash C_\theta(1)$ is context-free. Also, if $\theta(a) \neq a$ for any $a \in \Sigma$, the set of all θ-unbordered words, $D_\theta(1)$, is a subset of $C_\theta(1)$, i.e., $D_\theta(1) \subseteq C_\theta(1)$, [40].

Let us recall the definition of a *θ-palindrome* ([15, 40]) mentioned in the earlier section. A word u is said to be a θ-palindrome if $u = \theta(u)$ for the (anti)morphic involution θ, and P_θ denotes the set of all θ-palindromes. Kari et al., [41], has extended this line of research into a further exploration of the properties of P_θ as well as $\overline{P_\theta}$, the set of all non θ-palindromes. Note that, if a strand involved in the computation is a WK-palindrome, then it can hybridize to another copy of itself forming undesirable structures. One can easily observe that a non-empty θ-palindromic word always has length greater than or equal to 2, and that a power of a θ-palindromic word is again a θ-palindrome, [41]. The following proposition provides a necessary and sufficient condition for a word to be θ-palindrome.

Proposition 2.5.12 *Let θ be an antimorphic involution. Then $w \in P_\theta$ iff $w = \alpha(\beta\alpha)^i$ for $\alpha, \beta \in P_\theta$ and $i \geq 0$.*

The set P_θ is not regular, but is context-free, and the sets P_θ and $\overline{P_\theta}$ are dense. The following result establishes a connection between the primitive root of a non θ-palindrome and θ-palindromes.

Proposition 2.5.13 *[41] Let θ be anitmorphic involution and let $v \in \Sigma^+ \setminus P_\theta$. Then the primitive root of v is the product of two non-empty Watson-Crick palindromes iff there exists a non-empty word $u \in P_\theta$ such that u θ-commutes with v.*

2.6 Conclusion

The idea of storing encoded data in DNA of micro-organisms and using DNA as a tool to solve problems in mathematics and computer science is undoubtedly a breakthrough, although there are several theoretical and practical constrains. In this chapter, we reviewed and reported some of the attempts that have been made to address the problem of encoding and storing encoded data on DNA. Also, we have discussed how the field of DNA computing has motivated the study of some meaningful generalizations of classical concepts in formal languages and combinatorics on words.

Bibliography

[1] L. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994.

[2] M. Arita and S. Kobayashi. DNA sequence design using templates. *New Generation Computing*, 20(3):263–277, 2002.

[3] E. B. Baum. Building an associative memory vastly larger than the brain. *Science*, 268(5210):583–585, 1995.

[4] A. Blondin Massé, S. Gaboury, and S. Hallé. Pseudoperiodic words. In H.-C. Yen and O. Ibarra, editors, *Developments in Language Theory*, volume 7410 of *Lecture Notes in Computer Science*, pages 308–319. Springer Berlin Heidelberg, 2012.

[5] S. Brenner, S. R. Williams, E. H. Vermaas, T. Storck, K. Moon, C. McCollum, J.-I. Mao, S. Luo, J. J. Kirchner, S. Eletr, R. B. DuBridge, T. Burcham, and G. Albrecht. In vitro cloning of complex mixtures of DNA on microbeads: Physical separation of differentially expressed cDNAs. *Proceedings of the National Academy of Sciences*, 97(4):1665–1670, 2000.

[6] A. Carpi and A. de Luca. Periodic-like words, periodicity, and boxes. *Acta Informatica*, 37(8):597–618, 2001.

[7] J. Chen, R. Deaton, M. Garzon, J. Kim, D. Wood, H. Bi, D. Carpenter, and Y.-Z. Wang. Characterization of non-crosshybridizing DNA oligonucleotides manufactured in vitro. In

C. Ferretti, G. Mauri, and C. Zandron, editors, *DNA Computing*, volume 3384 of *Lecture Notes in Computer Science*, pages 50–61. Springer Berlin Heidelberg, 2005.

[8] J. Chen, R. Deaton, and Y.-Z. Wang. A DNA-based memory with in vitro learning and associative recall. In J. Chen and J. Reif, editors, *DNA Computing*, volume 2943 of *Lecture Notes in Computer Science*, pages 145–156. Springer Berlin Heidelberg, 2004.

[9] G. M. Church, Y. Gao, and S. Kosuri. Next-generation digital information storage in DNA. *Science*, 337(6102):1628–1628, 2012.

[10] S. Constantinescu and L. Ilie. Fine and Wilf's theorem for abelian periods. *Bulletin of the EATCS*, 89:167–170, 2006.

[11] J. P. Cox. Long-term data storage in DNA. *Trends in Biotechnology*, 19(7):247–250, 2001.

[12] L. J. Cummings and W. F. Smyth. Weak repetitions in strings. *J. Combinatorial Mathematics and Combinatorial Computing*, 24:33–48, 1997.

[13] E. Czeizler, E. Czeizler, L. Kari, and S. Seki. An extension of the Lyndon Schützenberger result to pseudoperiodic words. In V. Diekert and D. Nowotka, editors, *Developments in Language Theory*, volume 5583 of *Lecture Notes in Computer Science*, pages 183–194. Springer Berlin Heidelberg, 2009.

[14] E. Czeizler, L. Kari, and S. Seki. On a special class of primitive words. *Theoretical Computer Science*, 411:617–630, 2010.

[15] A. de Luca and A. de Luca. Pseudopalindrome closure operators in free monoids. *Theoretical Computer Science*, 362(13):282–300, 2006.

[16] R. Deaton, J. Chen, H. Bi, M. Garzon, H. Rubin, and D. Harlan Wood. A PCR-based protocol for in vitro selection of non-crosshybridizing oligonucleotides. In M. Hagiya

and A. Ohuchi, editors, *DNA Computing*, volume 2568 of *Lecture Notes in Computer Science*, pages 196–204. Springer Berlin Heidelberg, 2003.

[17] R. M. Dirks, M. Lin, E. Winfree, and N. A. Pierce. Paradigms for computational nucleic acid design. *Nucleic Acids Research*, 32(4):1392–1403, 2004.

[18] U. Feldkamp, W. Banzhaf, H. Rauhe, et al. A DNA sequence compiler. In *6th DIMACS Workshop on DNA Based Computers*, page 253, 2000.

[19] U. Feldkamp, S. Saghafi, W. Banzhaf, and H. Rauhe. DNASequenceGenerator: A program for the construction of DNA sequences. In N. Jonoska and N. Seeman, editors, *DNA Computing*, volume 2340 of *Lecture Notes in Computer Science*, pages 23–32. Springer Berlin Heidelberg, 2002.

[20] N. Fine and H. Wilf. Uniqueness theorem for periodic functions. In *Proceedings of the Amercian Mathematical Society*, volume 16, pages 109–114, 1965.

[21] M. Garzon, D. Blain, and A. Neel. Virtual test tubes. *Natural Computing*, 3(4):461–477, 2004.

[22] M. Garzon, K. Bobba, and A. Neel. Efficiency and reliability of semantic retrieval in DNA-based memories. In J. Chen and J. Reif, editors, *DNA Computing*, volume 2943 of *Lecture Notes in Computer Science*, pages 157–169. Springer Berlin Heidelberg, 2004.

[23] M. H. Garzon. DNA codeword design: Theory and applications. *Parallel Processing Letters*, 24(02):1440001, 2014.

[24] P. Gawrychowski, F. Manea, R. Mercaş, D. Nowotka, and C. Tiseanu. Finding pseudo-repetitions. *Leibniz International Proceedings in Informatics*, 20:257–268, 2013.

[25] P. Gawrychowski, F. Manea, and D. Nowotka. Discovering hidden repetitions in words. In P. Bonizzoni, V. Brattka, and B. Löwe, editors, *The Nature of Computation. Logic,*

Algorithms, Applications, volume 7921 of *Lecture Notes in Computer Science*, pages 210–219. Springer Berlin Heidelberg, 2013.

[26] P. Gawrychowski, F. Manea, and D. Nowotka. Testing generalised freeness of words. In E. W. Mayr and N. Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, volume 25, pages 337–349, 2014.

[27] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature*, 494(7435):77–80, 2013.

[28] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark. Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angewandte Chemie International Edition*, 54(8):2552–2555, 2015.

[29] A. J. Hartemink, D. K. Gifford, and J. Khodor. Automated constraint-based nucleotide sequence selection for DNA computation. *Biosystems*, 52(13):227–235, 1999.

[30] J. E. Hopcroft and J. D. Ullman. *Formal Languages and their Relation to Automata*. Addison-Wesley Longman Inc., 1969.

[31] S. Hussini, L. Kari, and S. Konstantinidis. Coding properties of DNA languages. In N. Jonoska and N. Seeman, editors, *Proc. of DNA7*, volume 2340 of *Lecture Notes in Computer Science*, pages 57–69. Springer, 2002.

[32] P. Intaluck, R. Akkarawongsapat, P. Palittapongarnpim, and B. Yimwadsana. Reliable DNA signal relay by hairpin structure in DNA-based logic circuit. *Journal of Bionanoscience*, 9(1):47–54, 2015-02-01.

[33] N. Jonoska and K. Mahalingam. Languages of DNA based code words. In J. Chen and J. Reif, editors, *DNA Computing*, volume 2943 of *Lecture Notes in Computer Science*, pages 61–73. Springer Berlin Heidelberg, 2004.

[34] L. Kari, R. Kitto, and G. Thierrin. Codes, involutions, and DNA encodings. In W. Brauer, H. Ehrig, J. Karhumki, and A. Salomaa, editors, *Formal and Natural Computing*, volume 2300 of *Lecture Notes in Computer Science*, pages 376–393. Springer Berlin Heidelberg, 2002.

[35] L. Kari, S. Konstantinidis, E. Losseva, and G. Wozniak. Sticky-free and overhang-free DNA languages. *Acta Informatica*, 40(2):119–157, 2003.

[36] L. Kari, S. Konstantinidis, and P. Sosík. Bond-free languages: Formalizations, maximality and construction methods. *International Journal of Foundations of Computer Science*, 16:1039–1070, 2005.

[37] L. Kari, S. Konstantinidis, and P. Sosík. On properties of bond-free DNA languages. *Theoretical Computer Science*, 334(1):131–159, 2005.

[38] L. Kari, E. Losseva, S. Konstantinidis, P. Sosík, and G. Thierrin. A formal language analysis of DNA hairpin structures. *Fundamenta Informaticae*, 71:453–475, 2006.

[39] L. Kari and K. Mahalingam. Involutively bordered words. *International Journal of Foundations of Computer Science*, 18(05):1089–1106, 2007.

[40] L. Kari and K. Mahalingam. Watson-Crick conjugate and commutative words. In M. H. Garzon and H. Yan, editors, *Proc. of DNA13*, volume 4848 of *Lecture Notes in Computer Science*, pages 273–283. Springer-Verlag, 2008.

[41] L. Kari and K. Mahalingam. Watson-Crick palindromes in DNA computing. *Natural computing*, 9(2):297–316, June 2010.

[42] L. Kari, B. Masson, and S. Seki. Properties of pseudo-primitive words and their applications. *International Journal of Foundations of Computer Science*, 22(02):447–471, 2011.

[43] L. Kari and S. Seki. On pseudoknot-bordered words and their properties. *Journal of Computer and System Sciences*, 75:113–121, 2009.

[44] L. Kari, S. Seki, and P. Sosík. DNA computing: Foundations and implications. In G. Rozenberg, T. Bäck, and J. Kok, editors, *Handbook of Natural Computing*, pages 1073–1127. Springer Berlin Heidelberg, 2012.

[45] S. Kashiwamura, M. Yamamoto, A. Kameda, T. Shiba, and A. Ohuchi. Hierarchical DNA memory based on nested PCR. In M. Hagiya and A. Ohuchi, editors, *DNA Computing*, volume 2568 of *Lecture Notes in Computer Science*, pages 112–123. Springer Berlin Heidelberg, 2003.

[46] S. Kashiwamura, M. Yamamoto, A. Kameda, T. Shiba, and A. Ohuchi. Potential for enlarging DNA memory: the validity of experimental operations of scaled-up nested primer molecular memory. *Biosystems*, 80(1):99–112, 2005.

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

[48] R. C. Lyndon and M. P. Schutzenberger. The equation $a^M = b^N c^P$ in a free group. *Michigan Math. J.*, 9:289–298, 1962.

[49] F. Manea, M. Müller, and D. Nowotka. On the pseudoperiodic extension of $u^l = v^m w^n$. In *Foundations of Software Technology and Theoretical Computer Science*, volume 24, pages 475–486, 2013.

[50] F. Manea, M. Müller, D. Nowotka, and S. Seki. Generalised Lyndon-Schützenberger equations. In E. Csuhaj-Varjú, M. Dietzfelbinger, and Z. Ésik, editors, *Mathematical Foundations of Computer Science 2014*, volume 8634 of *Lecture Notes in Computer Science*, pages 402–413. Springer Berlin Heidelberg, 2014.

[51] A. Marathe, A. E. Condon, and R. M. Corn. On combinatorial DNA word design. *Journal of Computational Biology*, 8(3):201–219, 2001.

[52] G. Mauri and C. Ferretti. Word design for molecular computing: A survey. In J. Chen and J. Reif, editors, *DNA Computing*, volume 2943 of *Lecture Notes in Computer Science*, pages 37–47. Springer Berlin Heidelberg, 2004.

[53] K. U. Mir. A restricted genetic alphabet for DNA computing. *DNA Based Computers II*, pages 243–246, 1999.

[54] A. Neel and M. Garzon. Semantic retrieval in DNA-based memories with Gibbs energy models. *Biotechnology Progress*, 22(1):86–90, 2006.

[55] M. Nuser and R. Deaton. Simulations of DNA computing with in vitro selection. *Genetic Programming and Evolvable Machines*, 4(2):173–183, 2003.

[56] G. Paun, G. Rozenberg, and T. Yokomori. Hairpin languages. *Int. J. Found. Comput. Sci.*, 12:837–847, 2001.

[57] V. Phan and M. H. Garzon. On codeword design in metric DNA spaces. *Natural Computing*, 8(3):571–588, 2009.

[58] J. Reif, T. LaBean, M. Pirrung, V. Rana, B. Guo, C. Kingsford, and G. Wickham. Experimental construction of very large scale DNA databases with associative search capability. In N. Jonoska and N. Seeman, editors, *DNA Computing*, volume 2340 of *Lecture Notes in Computer Science*, pages 231–247. Springer Berlin Heidelberg, 2002.

[59] J. Sager and D. Stefanovic. Designing nucleotide sequences for computation: A survey of constraints. In A. Carbone and N. Pierce, editors, *DNA Computing*, volume 3892 of *Lecture Notes in Computer Science*, pages 275–289. Springer Berlin Heidelberg, 2006.

[60] K. Sakamoto, H. Gouzu, K. Komiya, D. Kiga, S. Yokoyama, T. Yokomori, and M. Hagiya. Molecular computation by DNA hairpin formation. *Science*, 288(5469):1223–1226, 2000.

[61] H. J. Shyr. *Free Monoids and Languages*. Department of Mathematics, Soochow University, Taipei, Taiwan, 1979.

[62] G. C. Smith, C. C. Fiddes, J. P. Hawkins, and J. Cox. Some possible codes for encrypting data in DNA. *Biotechnology Letters*, 25(14):1125–1130, 2003.

[63] M. Takinoue and A. Suyama. Molecular reactions for a molecular memory based on hairpin DNA. *Chem-Bio Informatics Journal*, 4(3):93–100, 2004.

[64] Y. Tsuboi, Z. Ibrahim, and O. Ono. DNA-based semantic memory with linear strands. *International Journal of Innovative Computing, Information and Control*, 1(4):755–766, 2005.

[65] D. Tulpan, M. Andronescu, S. B. Chang, M. R. Shortreed, A. Condon, H. H. Hoos, and L. M. Smith. Thermodynamically based DNA strand design. *Nucleic Acids Research*, 33(15):4951–4964, 2005.

[66] D. Tulpan, H. Hoos, and A. Condon. Stochastic local search algorithms for DNA word design. In M. Hagiya and A. Ohuchi, editors, *DNA Computing*, volume 2568 of *Lecture Notes in Computer Science*, pages 229–241. Springer Berlin Heidelberg, 2003.

[67] J. D. Watson and F. H. Crick. Molecular structure of nucleic acids. *Nature*, 171(4356):737–738, 1953.

[68] P. C. Wong, K.-k. Wong, and H. Foote. Organic data memory using the DNA approach. *Commun. ACM*, 46(1):95–98, 2003.

[69] M. Yamamoto, S. Kashiwamura, and A. Ohuchi. DNA memory with 16.8M addresses. In M. Garzon and H. Yan, editors, *DNA Computing*, volume 4848 of *Lecture Notes in Computer Science*, pages 99–108. Springer Berlin Heidelberg, 2008.

[70] S.-S. Yu. *Languages and Codes*. Tsang Hai Book Publishing Co., 2005.

Chapter 3

Generating Pseudo-Powers of A Word

3.1 Introduction

Periodicity and primitivity of words are fundamental properties in combinatorics on words and formal language theory. Their wide-ranging applications include pattern-matching algorithms (see e.g. [3], and [4]) and data-compression algorithms (see, e.g., [27]). Sometimes motivated by their applications, these classical notions have been modified or generalized in various ways. A representative example is the "weak periodicity" of [5] whereby a word is called *weakly periodic* if it consists of repetitions of words with the same Parikh vector. This type of period was also called *Abelian period* in [2]. Carpi and de Luca extended the notion of periodic words to that of periodic-like words, according to the extendability of factors of a word [1]. Czeizler, Kari, and Seki have proposed and investigated the notion of *pseudo-primitivity* (and pseudo-periodicity) of words in [6, 20], motivated by the properties of information encoded as DNA strands.

Indeed, one of the particularities of information encoded as DNA strands is that a word u over the DNA alphabet $\{A, C, G, T\}$ contains basically the same information as its Watson-Crick complement, denoted here by $\theta(u)$. This led to natural as well as theoretically interesting

[0]A version of this chapter has been published (L. Kari, M.S. Kulkarni. Generating the pseudo-powers of a word. *Journal of Automata, Languages and Combinatorics*, 19(2014), 1-4, 157-171)

extensions of various notions in combinatorics on words and formal language theory such as pseudo-palindrome [7], pseudo-commutativity [18], as well as hairpin-free and bond-free languages (e.g., [17, 19, 25, 13, 16]). In this context, Watson-Crick complementarity has been modeled mathematically by an antimorphic involution θ over an alphabet Σ, i.e., a function that is an antimorphism, $\theta(uv) = \theta(v)\theta(u)$, $\forall u, v \in \Sigma^*$, and an involution, $\theta(\theta(x)) = x$, $\forall x \in \Sigma^*$. In [6], a word w is called *θ-primitive*, or pseudo-primitive, if we cannot find any word u that is strictly shorter than w such that w can be written as repetitions of u and $\theta(u)$. A word w is called a θ-power or pseudo-power if $w \in \{u, \theta(u)\}^+$ for some $u \in \Sigma^+$, and is called θ-periodic or pseudo-periodic if it can be written as two or more repetitions of a word u and its image under θ.

The static notions of the power of a word, period of a word, and primitive word are intrinsically connected to the operation of catenation, that dynamically generates word repetitions. In the case of generalizations of the notion of power of a word (primitive word), other operations will be the ones that dynamically produce such generalized powers, [26, 21, 10, 14, 22, 9].

In this paper we define and investigate the operation of θ-catenation that gives rise to the notion of θ-power (pseudo-power) and θ-periodicity (pseudo-periodicity). We namely investigate the properties of θ-catenation (Section 3), its connection to the previously defined notion of θ-primitive word (Section 4), briefly explore closure properties of language families under θ-catenation and language operations involving this operation (Section 5), and conclude by proposing Abelian catenation as the operation that generates Abelian powers of words (Section 6).

3.2 Basic definitions and notations

An alphabet Σ is a finite non-empty set of symbols. Σ^* denotes the set of all words over Σ, including the empty word λ. Σ^+ is the set of all non-empty words over Σ. The length of a word $u \in \Sigma^*$ (i.e. number of symbols in the word) is denoted by $|u|$. A word $u \in L$ is said to be

minimal if for all $w \in L$, $|w| \geq |u|$. $|u|_a$ denotes the number of occurrences of a letter a in u. The complement of a language $L \subseteq \Sigma^*$ is $L^c = \Sigma^* \backslash L$.

An *involution* is a function $\theta : \Sigma^* \to \Sigma^*$ with the property that θ^2 is identity. θ is called a *morphism* if for all words $u, v \in \Sigma^*$ we have that $\theta(uv) = \theta(u)\theta(v)$, and an *antimorphism* if $\theta(uv) = \theta(v)\theta(u)$.

A word is called *primitive* if it cannot be expressed as a power of another word. Similarly, [6], a word is called as *θ-primitive* if it cannot be expressed as a non-trivial θ-power of another word. A *θ-power* of u is a word of the form $u_1 u_2 \cdots u_n$ for some $n \geq 1$, where $u_1 = u$ and for any $2 \leq i \leq n$, u_i is either u or $\theta(u)$. Also, *θ-primitive root* of w denoted by $\rho_\theta(w)$ is the shortest word t such that w is a θ-power of t.

The *left quotient* of a word u by a word v is defined by

$$v^{-1}u = w \text{ iff } u = vw,$$

and the *right quotient* of u by v,

$$uv^{-1} = w \text{ iff } u = wv.$$

A language $L \subseteq \Sigma^+$ is said to be a prefix code if $L \cap L\Sigma^+ = \emptyset$. For all other concepts related to formal language theory and combinatorics on words, the reader is referred to [11] and [23].

A *binary word operation with right identity*, [12, 26], (shortly *bw-operation)* is defined as a mapping $\circ : \Sigma^* \times \Sigma^* \longrightarrow 2^{\Sigma^*}$ with $u \circ \lambda = \{u\}$. Furthermore, $L_1 \circ L_2 = \bigcup_{u \in L_1, v \in L_2} (u \circ v)$ and $L_1 \circ \emptyset = \emptyset \circ L_2 = \emptyset$ for any two languages L_1 and L_2. The *iterated bw-operation* $\circ^i$ for $i \geq 1$ and languages L_1 and L_2 is defined as $L_1 \circ^0 L_2 = L_1$ and $L_1 \circ^i L_2 = (L_1 \circ^{i-1} L_2) \circ L_2$. The *$i$-th $\circ$-power* of a non-empty language L is defined as $L^{\circ(0)} = \{\lambda\}$ and $L^{\circ(i)} = L \circ^{i-1} L$ for $i \geq 1$. If $\circ$ is the operation of catenation, then $L^0 = \{\lambda\}$, $L^1 = L$ and $L^n = L^{n-1}L$, corresponding to the usual notions of power of a language.

A non-empty word w is called *$\circ$-primitive* if $w \in u^{\circ(i)}$ for some word $u \in \Sigma^+$ and $i \geq 1$ yields $i = 1$ and $w = u$.

The $+$-*closure* of a non-empty language L with respect to a bw-operation $\circ$, denoted by $L^{\circ(+)}$, is defined as $L^{\circ(+)} = \cup_{k\geq 1} L^{\circ(k)}$. A language L is $\circ$-*closed* if $u, v \in L$ imply $u \circ v \subseteq L$. A bw-operation is called *plus-closed* if for any non-empty language L, $L^{\circ(+)}$ is $\circ$-closed.

Given a non-empty language L, a word u is a *right* $\circ$-*residual* of L if $w \circ u \subseteq L$ for all $w \in L$, i.e., $L \circ u \subseteq L$. Let $\rho_\circ(L)$ denote the set of all right o-residuals of L, i.e., $\rho_\circ(L) = \{u \in \Sigma^* | \forall w \in L, (w \circ u) \subseteq L\}$. Note that $\rho_\circ(\emptyset) = \emptyset$ and $\lambda \in \rho_\circ(L)$ for any non-empty language L.

The $\circ$-*left-quotient*, denoted by $\triangleleft_\circ$, is defined as

$$L_1 \triangleleft_\circ L_2 = \{w \in \Sigma^* | (L_2 \circ w) \cap L_1 \neq \emptyset\}.$$

3.3 θ-catenation

We introduce a new bw-operation (binary word operation with right identity) called θ-*catenation* which generates pseudo-powers, that is, θ-powers where θ is a morphic or antimorphic involution. In this section we will give a formal definition of θ-*catenation* and discuss some of its properties. Note that, unless otherwise specified, θ is any morphic or antimorphic involution.

Definition 3.1 *Given a morphic or antimorphic involution θ on Σ^* and any two words $u, v \in \Sigma^*$, we define the binary operation θ-catenation as*

$$u \odot v = \{uv, u\theta(v)\}.$$

For example, consider the DNA alphabet $\Sigma = \{A, G, C, T\}$ and its associated antimorphic involution defined by $\theta(A) = T, \theta(T) = A, \theta(C) = G$ and $\theta(G) = C$. If $u = ATC$ and $v = GCTA$ then

$$u \odot v = \{ATCGCTA, ATCTAGC\}$$

The operation of θ-*catenation* can be generalized to languages in the usual way.

Note that for any (anti)morphic involution θ, the operation of *θ-catenation* has a right identity since $u \odot \lambda = \{u\}$ for all $u \in \Sigma^*$.

A bw-operation $\circ$ is called *length-increasing* if for any $u, v \in \Sigma^+$ and $w \in u \circ v$, $|w| > \max\{|u|, |v|\}$. The operation of *θ-catenation* is length-increasing since, if $w \in u \odot v = \{uv, u\theta(v)\}$ then $|w| = |u| + |v| > \max\{|u|, |v|\}$.

A bw-operation $\circ$ is called *propagating* if for any $u, v \in \Sigma^*$, $a \in \Sigma$ and $w \in u \circ v$, $|w|_a = |u|_a + |v|_a$. The operation of *θ-catenation* is clearly not propagating. However, a similar property does hold. We will namely call a bw-operation $\circ$ *θ-propagating* if for any $u, v \in \Sigma^*$, $a \in \Sigma$ and $w \in u \circ v$, $|w|_{a,\theta(a)} = |u|_{a,\theta(a)} + |v|_{a,\theta(a)}$. (The mapping which counts number of a's and $\theta(a)$'s together is the *characteristic function on the alphabet* Σ defined in [6].)

Proposition 3.3.1 *For a given (anti)morphic involution θ of Σ^*, the operation of θ-catenation is θ-propagating.*

Proof Let $u, v \in \Sigma^*$ and let $w \in u \odot v = \{uv, u\theta(v)\}$. If $w = uv$ then the required equality clearly holds.

If $w = u\theta(v)$, we have

$$\begin{aligned} |w|_{a,\theta(a)} &= |u|_{a,\theta(a)} + |\theta(v)|_{a,\theta(a)} \\ &= |u|_{a,\theta(a)} + (|\theta(v)|_a + |\theta(v)|_{\theta(a)}) \\ &= |u|_{a,\theta(a)} + (|v|_{\theta(a)} + |v|_a) \\ &= |u|_{a,\theta(a)} + |v|_{a,\theta(a)}. \end{aligned}$$

□

A bw-operation $\circ$ satisfies the *left-identity* condition if $\lambda \circ L = L$ for any language $L \subseteq \Sigma^*$. Note that, in general, the operation of *θ-catenation* does not satisfy the left-identity condition. However, there exists languages of Σ^* which satisfy this condition, such as the language of θ-palindromes $P_\theta = \{u \in \Sigma^* | u = \theta(u)\}$ for which $\lambda \odot P_\theta = P_\theta$.

A bw-operation $\circ$ is called *left-inclusive* if for any three words $u, v, w \in \Sigma^*$ we have

$$(u \circ v) \circ w \supseteq u \circ (v \circ w)$$

and is called *right-inclusive* if

$$(u \circ v) \circ w \subseteq u \circ (v \circ w).$$

If θ is a morphic involution then the *θ-catenation* is trivially associative. However, if θ is an antimorphic involution then θ-catenation is not associative in general, and not even right- or left-inclusive . The following proposition provides necessary and sufficient conditions for associativity to hold in the antimorphic case. To prove Proposition 3.3.4, we will make use of the following Lemmas from [24].

Lemma 3.3.2 *Let $u, v \in \Sigma^+$. Then $uv = vu$ implies that u and v are powers of a common word.*

Lemma 3.3.3 *If $u^m = v^n$ and $m, n \geq 1$, then u and v are powers of a common word.*

Proposition 3.3.4 *Let $\odot$ denote the operation of θ-catenation associated with an antimorphic involution θ of Σ^*. Given words $u, v, w \in \Sigma^*$ we have $(u \odot v) \odot w = u \odot (v \odot w)$ if and only if v and w are powers of the same θ-palindromic word.*

Proof For the direct implication, let us assume that $(u \odot v) \odot w = u \odot (v \odot w)$, i.e.,
$\{uvw, u\theta(v)w, uv\theta(w), u\theta(v)\theta(w)\} = \{uvw, uv\theta(w), u\theta(w)\theta(v), uw\theta(v)\}$, i.e.
$\{u\theta(v)w, u\theta(v)\theta(w)\} = \{u\theta(w)\theta(v), uw\theta(v)\}$.

Case 1 : $u\theta(v)\theta(w) = u\theta(w)\theta(v)$ and $u\theta(v)w = uw\theta(v)$ implies $\theta(wv) = \theta(vw)$ and $\theta(v)w = w\theta(v)$ which further implies $wv = vw$ and $\theta(v)w = w\theta(v)$, respectively. So, according to Lemma 3.3.2, v and w are powers of a common word, as well as w and $\theta(v)$ are powers of a common word. This means, v, w and $\theta(v)$ are all powers of a common word, say p. So, we have $v = p^i$, $w = p^j$ and $\theta(v) = p^k$ for some $i, j, k \geq 1$. It implies, $\theta(v) = \theta(p)^i = p^k$, which

further implies $i = k$ and $p = \theta(p)$. Hence v and w are powers of the same θ-palindromic word p.

Case 2 : $u\theta(v)w = u\theta(w)\theta(v)$ and $u\theta(v)\theta(w) = uw\theta(v)$ implies

$$\theta(v)w = \theta(w)\theta(v). \tag{3.1}$$

and

$$\theta(v)\theta(w) = w\theta(v). \tag{3.2}$$

Let us catenate $\theta(v)$ to the right of Equation (3.2). It will give, $\theta(v)\theta(w)\theta(v) = w\theta(v)\theta(v)$, which in turn along with Equation (3.1) implies

$$\theta(v)\theta(v)w = w\theta(v)\theta(v) \tag{3.3}$$

According to Lemma 3.3.2 w and $(\theta(v))^2$ are powers of a common word, say p. So, we will get $w = p^i$ and $(\theta(v))^2 = p^j$ for some $i, j \geq 1$. Now, according to Lemma 3.3.3 $\theta(v)$ and p are powers of a common word, say q. So, we get

$$p = q^l, \theta(v) = q^m \text{ and } w = q^n \text{ for } l, m, n \geq 1. \tag{3.4}$$

Substituting Equation (3.4) in the Equation (3.1) we get

$$q^m q^n = \theta(q^n)q^m \tag{3.5}$$

which implies that $q = \theta(q)$, i.e. q is a θ-palindromic word and v and w are powers of q.

Conversely, suppose v and w are powers of the same θ-palindromic word, say p. This implies, $v = p^i, w = p^j$ for $i, j \geq 1$ and $p = \theta(p)$, which further implies

$$\theta(v) = (\theta(p))^i = p^i \text{ and } \theta(w) = p^j. \tag{3.6}$$

Now, we know that, $(u \odot v) \odot w = \{uvw, u\theta(v)w, uv\theta(w), u\theta(v)\theta(w)\}$ and

$u \odot (v \odot w) = \{uvw, uv\theta(w), u\theta(w)\theta(v), uw\theta(v)\}$. If we compare these two expressions, we are left to show that $\{u\theta(v)w, u\theta(v)\theta(w)\} = \{u\theta(w)\theta(v), uw\theta(v)\}$, which is clear from Equation (3.6).

□

In the previous section, we have seen the definition of i-th $\circ$-power of a non-empty language L. The following Lemma and its Corollary clarify this definition in the case of any bw-operation.

Lemma 3.3.5 *Given a bw-operation $\circ$, we have*

$$L^{\circ(0)} = \{\lambda\},$$

$$L^{\circ(1)} = L,$$

$$L^{\circ(n)} = L^{\circ(n-1)} \circ L, \ \forall n \geq 2.$$

Proof Fistly, $L^{\circ(0)} = \{\lambda\}$ by definition. Secondly, $L^{\circ(1)} = L \circ^0 L = L$. Thirdly, for $n \geq 2$ we have $L^{\circ(n)} = L \circ^{n-1} L = (L \circ^{n-2} L) \circ L = L^{\circ(n-1)} \circ L$. □

Corollary 3.3.6 *Given a bw-operation $\circ$, we have*

$$u^{\circ(0)} = \lambda,$$

$$u^{\circ(1)} = u,$$

$$u^{\circ(n)} = u^{\circ(n-1)} \circ u, \ \forall n \geq 2.$$

The following lemma characterizes the form of the words in $L^{\odot(n)}$ when the operation that is applied iteratively is the *θ-catenation*.

Lemma 3.3.7 *If $\odot$ denotes the operation of θ-catenation associated to a morphic or antimorphic involution θ of Σ^* then for $n \geq 1$,*

$$L^{\odot(n)} = \{uv_1v_2 \cdots v_{n-1} | u \in L, v_i \in L \cup \theta(L), 0 \leq i \leq n-1\}.$$

In particular, when $n = 1$ we have $L^{\odot(1)} = L$.

Proof We will prove this by induction on n.

For $n = 1$, $L^{\odot(1)} = L \odot^0 L = L$.

For $n = 2$, $L^{\odot(2)} = LL \cup L\theta(L) = \{uv | u \in L, v \in L \cup \theta(L)\}$.

Assume that the result is true for an arbitrary $k \geq 2$, i.e.,

$$L^{\odot(k)} = \{uv_1v_2 \cdots v_{k-1} | u \in L, v_i \in L \cup \theta(L), 1 \leq i \leq k-1\}.$$

For $k+1 \geq 2$ the last equation of Lemma 3.3.5 holds and, together with the induction hypothesis we have

$$\begin{aligned} L^{\odot(k+1)} &= L^{\odot(k)} \odot L \\ &= \{uv_1v_2 \cdots v_{k-1} | u \in L, v_i \in L \cup \theta(L), 1 \leq i \leq k-1\} \odot L \\ &= \{uv_1v_2 \cdots v_k | u \in L, v_i \in L \cup \theta(L), 1 \leq i \leq k\}. \end{aligned}$$

□

The following Corollary demonstrates that, in the same way the operation of catenation dynamically generates regular powers of words, the operation of *θ-catenation* is the one that generates the θ-powers of a word.

Corollary 3.3.8 *If $\odot$ denotes the operation of θ-catenation associated to a morphic or anti-*

morphic involution θ of Σ^*, then every word $w \in u^{\odot(n)}$, $n \geq 1$, is of the form

$$w = uv_1v_2 \cdots v_{n-1}$$

where $v_i \in \{u, \theta(u)\}$ for $0 \leq i \leq n-1$. In particular, for $n = 1$ we have $w = u$.

The following Proposition relates the number of occurrences of a letter a and $\theta(a)$ in a word to the number of occurences of a and $\theta(a)$ of its $\circ$-power.

Proposition 3.3.9 *If $\circ$ is θ-propagating bw-operation, then for any $w \in u^{\circ(n)}$, $|w|_{a,\theta(a)} = n \cdot |u|_{a,\theta(a)}$, for $n \geq 1$.*

Lemma 3.3.10 *If $\circ$ is an associative bw-operation and $L \subseteq \Sigma^*$, $L \neq \emptyset$, we have*

$$L^{\circ(m)} \circ L^{\circ(n)} = L^{\circ(m+n)} \text{ for } m, n \geq 1.$$

Proof

$$\begin{aligned} L^{\circ(m+n)} &= L^{\circ(m+(n-1))} \circ L \\ &= (L^{\circ(m+(n-2))} \circ L) \circ L \\ &= L^{\circ(m+(n-2))} \circ (L \circ L) \\ &= L^{\circ(m+(n-2))} \circ L^{\circ(2)} \\ &= L^{\circ(m+(n-3))} \circ L^{\circ(3)} = ... \\ &= L^{\circ(m)} \circ L^{\circ(n)}. \end{aligned}$$

□

Lemma 3.3.10 does not hold in general for operations that are not associative. However, in the case of *θ-catenation*, when θ is an antimorphic involution, one of the inclusions in Lemma 3.3.10 holds, even though *θ-catenation* is not right- or left-inclusive. As a consequence, as seen in Corollary 3.3.12, *θ-catenation* is plus-closed.

Lemma 3.3.11 *If $\odot$ is the operation of θ-catenation associated with any morphic or antimorphic involution θ of Σ^* and $L \subseteq \Sigma^*$ is a nonempty language, then*

$$L^{\odot(m)} \odot L^{\odot(n)} \subseteq L^{\odot(m+n)}, \ \forall m, n \geq 1.$$

Proof If θ is a morphic involution then the operation of *θ-catenation* is associative and the inclusion holds by Lemma 3.3.10.

If θ is an antimorphic involution then, by Lemma 3.3.7, for every $n \geq 1$ we have

$$L^{\odot(n)} = \{uv_1v_2 \cdots v_{n-1} | u \in L, v_i \in L \cup \theta(L), 0 \leq i \leq n-1\}.$$

Let $x \in L^{\odot(m)}$ and $y \in L^{\odot(n)}$ for some $m, n \geq 1$. Then by Corollary 3.3.7 $x = uv_1v_2 \cdots v_{m-1}$ and $y = u'v'_1v'_2 \cdots v'_{n-1}$ for some $u, u' \in L$, $v_i, v'_i \in L \cup \theta(L)$, $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$. By the definition of *θ-catenation*,

$$x \odot y = \{uv_1v_2 \cdots v_{m-1}u'v'_1v'_2 \cdots v'_{n-1}, uv_1v_2 \cdots v_{m-1}\theta(v'_{n-1}) \ldots \theta(u')\},$$

which is a word in $L^{\odot(m+n)}$. □

Corollary 3.3.12 *The operation of θ-catenation is plus-closed for morphic as well as antimorphic involutions θ.*

A non-empty language $L \subseteq \Sigma^*$ is called *$\circ$-free* if $(L^{\circ(+)} \circ L) \cap L = \emptyset$. In the case of *$\theta$-catenation*, for example, if $L \subseteq \Sigma^*$ and

$$R = \{uv_1v_2 \ldots v_k | u \in L, v_i \in L \cup \theta(L), k \geq 1, 1 \leq i \leq k\}$$

then, if $L \cap R = \emptyset$, L is $\odot$-free. The following proposition provides more examples of $\odot$-free languages.

Proposition 3.3.13 *Given a morphic or antimorhic involution θ over Σ, and the operation $\odot$ (θ-catenation), any prefix code is $\odot$-free.*

Proof Let $L \subseteq \Sigma^*$ be a prefix code, and assume that L is not $\odot$-free. Then there exist $w \in L$, $u \in L^{\odot(+)}$ and $v \in L$ such that $w \in u \odot v = \{uv, u\theta(v)\}$. By the definition of *$\theta$-catenation* and Lemma 3.3.7, w is of the form $\alpha\beta_1\beta_2 \ldots \beta_{n-1}v$ or $\alpha\beta_1\beta_2 \ldots \beta_{n-1}\theta(v)$, where $\alpha \in L$ and $\beta_i \in L \cup \theta(L)$, $1 \leq i \leq n-1$, $n \geq 2$. This is a contradiction to the fact that L is a prefix code. □

The converse of the previous Proposition does not hold, as shown by the following example.

Example 3.1 *Let $\Sigma = \{A, G, C, T\}$, $\theta(A) = T, \theta(G) = C$, $L = \{AG, TT, AGCA\}$. The language L is $\odot$-free, but not a prefix code.*

Another way of obtaining $\odot$-free languages is given by means of the left θ-quotient. The *left θ-quotient* of two languages $L_1, L_2 \subseteq \Sigma^*$ is defined as

$$L_1 \lhd_\odot L_2 = \{w \in \Sigma^* | \ (L_2 \odot w) \cap L_1 \neq \emptyset\}.$$

Lemma 3.3.14 *If θ is a morphic involution then the left θ-quotient is given by*

$$u \lhd_\odot v = \{v^{-1}u, \theta(v)^{-1}\theta(u)\}$$

and if θ is an antimorphic involution then the left θ-quotient is given by

$$u \lhd_\odot v = \{v^{-1}u, \theta(u)\theta(v)^{-1}\}.$$

Proof Let θ be a morphic involution and let $w \in (u \lhd_\odot v)$. This implies $(v \odot w) \cap \{u\} \neq \emptyset$, that is $u \in \{vw, v\theta(w)\}$, which further implies $w \in \{v^{-1}u, \theta(v)^{-1}\theta(u)\}$.

Let θ be an antimorphic involution and let $w \in (u \lhd_\odot v)$. This implies $(v \odot w) \cap \{u\} \neq \emptyset$, that is $u \in \{vw, v\theta(w)\}$, which further implies $w \in \{v^{-1}u, \theta(u)\theta(v)^{-1}\}$. □

Lemma 3.3.15 *Let θ be a morphic or antimorphic involution over Σ and let L be a language in Σ^*. If L closed under left θ-quotient then L is not $\odot$-free.*

Proof $\lhd_\odot(L, L) = \{w \in \Sigma^* | (L \odot w) \cap L \neq \emptyset\}$. As L is $\lhd_\odot$-closed, $\lhd_\odot(L, L) \subseteq L$, which implies that $(L \odot L) \cap L \neq \emptyset$ which, since $L \subseteq L^{\odot(+)}$, further implies that L is not $\odot$-free. □

3.4 θ-primitive words

In this section we show that if the operation under consideration is θ-catenation, denoted by $\odot$, then the $\odot$- primitive words coincide with the θ-primitive words defined in section 3.2. We study some properties of such θ-primitive words. Recall the following result from [12].

Proposition 3.4.1 *[12] Let $\circ$ be plus-closed and length-increasing. Then for every word $w \in \Sigma^+$ there exists a $\circ$-primitive word u and an integer $n \geq 1$ such that $w \in u^{\circ(n)}$.*

The following results (Proposition 3.4.2, Lemma 3.4.4, and Proposition 3.4.5) are similar to analogous results in [26], involving propagating bw-operations.

Proposition 3.4.2 *Let $\circ$ be plus-closed and θ-propagating. Then for every word $w \in \Sigma^+$ there exists a $\circ$-primitive word u and a unique integer $n \geq 1$ such that $w \in u^{\circ(n)}$.*

Proof Every θ-propagating bw-operation is length-increasing. Now, by Proposition 3.4.1, for every word $w \in \Sigma^+$ there exists a $\circ$-primitive word u and an integer $n \geq 1$ such that $w \in u^{\circ(n)}$. Consider $a \in \Sigma$ such that $|u|_{a,\theta(a)} \neq 0$. Since $\circ$ is θ-propagating, for any $w_1 \in u^{\circ(m)}$ with $m \neq n$, by Proposition 3.3.9, we get $|w_1|_{a,\theta(a)} = m|u|_{a,\theta(a)} \neq n|u|_{a,\theta(a)} = |w|_{a,\theta(a)}$. Thus $w \notin u^{\circ(m)}$ for any $m \neq n$. Hence n is such an unique integer. □

A $\circ$-primitive word $u \in \Sigma^+$ such that $w \in u^{\circ(n)}$ for some $n \geq 1$, is called a $\circ$-root of w. In general, a word may not have a unique $\circ$-root. However, if $\circ$ is the operation of θ-catenation, then every word $w \in \Sigma^+$ has an unique $\odot$-root, also called θ-root, denoted by $\rho_\theta(w)$. The uniqueness of the θ-root of a word was demonstrated by the following theorem (corollary of Theorems 13 and 14 from [6]).

Theorem 3.4.3 *If θ is a morphic or antimorphic involution on Σ^* then for any word $w \in \Sigma^+$ there exists a unique θ-primitive word $t \in \Sigma^+$ such that $w \in t\{t, \theta(t)\}^*$, i.e., $\rho_\theta(w) = t$.*

Lemma 3.4.4 *Let Σ be an alphabet with $|\Sigma| \geq 2$ and $\circ$ be plus-closed and θ-propagating bw-operation. If a word $w \in \Sigma^+$ is not $\circ$-primitive, then for any $a \neq b$, $a, b \in \Sigma$ we have that $|w|_{a,\theta(a)}$ and $|w|_{b,\theta(b)}$ have a common factor $n > 1$.*

Proof If w is not $\circ$-primitive, then according to Proposition 3.4.1, $w \in u^{\circ(n)}$ for some $\circ$-primitive word $u \in \Sigma^+$ and $n > 1$. Since $\circ$ is θ-propagating and Proposition 3.3.9 holds, $|w|_{a,\theta(a)} = n \cdot |u|_{a,\theta(a)}$ for all $a \in \Sigma$. Similarly, $|w|_{b,\theta(b)} = n \cdot |u|_{b,\theta(b)}$. Hence, for any $a, b \in \Sigma$, we have that $|w|_{a,\theta(a)}$ and $|w|_{b,\theta(b)}$ have the common factor $n > 1$. □

Proposition 3.4.5 *Let Σ be an alphabet with $|\Sigma| \geq 3$ and $\circ$ be plus-closed and θ-propagating bw-operation. If $w \in \Sigma^+$, $a \in \Sigma$, $w \notin \{a, \theta(a)\}^+$, then there is an integer $m \geq 1$ such that all the words $v_1 \in (w \circ w^{m-1}a)$, $v_2 \in (aw^{m-1} \circ w)$, $v_3 = w^m a$ and $v_4 = aw^m$ are $\circ$-primitive.*

Proof For $w \in \Sigma^+$, let $m = \prod_{b \in \Sigma, |w|_{b,\theta(b)} \neq 0} |w|_{b,\theta(b)}$. For any $a \in \Sigma$, suppose $w \notin \{a, \theta(a)\}^+$. Such a word exists since $|\Sigma| \geq 3$. Let $v_1 \in (w \circ w^{m-1}a)$, $v_2 \in (aw^{m-1} \circ w)$, $v_3 = w^m a$ and $v_4 = aw^m$. If $b \notin \{a, \theta(a)\}$ is a letter occurring in w, $|v_1|_{a,\theta(a)} = |v_2|_{a,\theta(a)} = |v_3|_{a,\theta(a)} = |v_4|_{a,\theta(a)} = m \cdot |w|_{a,\theta(a)} + 1$ whereas $|v_1|_{b,\theta(b)} = |v_2|_{b,\theta(b)} = |v_3|_{b,\theta(b)} = |v_4|_{b,\theta(b)} = m \cdot |w|_{b,\theta(b)}$. As the number of occurrences of a together with $\theta(a)$ respectively the number of occurrences of b together with $\theta(b)$ in each v_i, $i = 1, 2, 3, 4$, are relatively prime, by Lemma 3.4.4, v_1, v_2, v_3 and v_4 are $\circ$-primitive words. □

In the remainder of the section we will investigate some properties of θ-primitive words.

Definition 3.2 *[12] A language $L \subseteq \Sigma^*$ is called right-$\circ$-dense (resp. left-$\circ$-dense) if for each $w \in \Sigma^+$, there exists $u \in \Sigma^*$ such that $(w \circ u) \cap L \neq \emptyset$ (resp. $(u \circ w) \cap L \neq \emptyset$).*

If $\circ$ is the catenation of words, then the right and left $\circ$-dense languages are called right and left dense languages, respectively. Let $Q_\circ(\Sigma)$ denote the set of all $\circ$-primitive words over Σ.

Proposition 3.4.6 *If Σ is an alphabet with $|\Sigma| \geq 3$ and $\circ$ is plus-closed and θ-propagating bw-operation, then $Q_\circ(\Sigma)$ is right and left $\circ$-dense.*

Proof For each $w \in \Sigma^+$, since $|\Sigma| \geq 3$, there exists $a \in \Sigma$ such that $w \notin \{a, \theta(a)\}^+$. As $\circ$ is plus-closed and θ-propagating, by Proposition 3.4.5, there exists $m \geq 1$, such that $(w \circ w^{m-1}a) \in Q_\circ(\Sigma)$ and $(aw^{m-1} \circ w) \in Q_\circ(\Sigma)$. This proves that $Q_\circ(\Sigma)$ is right and left $\circ$-dense. □

Next, we show that the set of θ-primitive words $Q_\odot(\Sigma)$ is right and left dense.

Proposition 3.4.7 *Let the operation of θ-catenation $\odot$ associated to morphic or antimorphic involution θ be plus-closed and θ-propagating and let $|\Sigma| \geq 3$. Then $Q_\odot(\Sigma)$ is right and left dense.*

Proof Let $w \in \Sigma^+$. If $w \in \{a, \theta(a)\}^+$ and $b \in \Sigma$ such that $b \notin \{a, \theta(a)\}$, then, $|wb|_{a,\theta(a)} = |bw|_{a,\theta(a)} = m \geq 1$. Also, $|wb|_{b,\theta(b)} = |bw|_{b,\theta(b)} = 1$, hence by Lemma 3.4.4 $wb \in Q_\odot(\Sigma)$ and $bw \in Q_\odot(\Sigma)$. If $w \notin \{a, \theta(a)\}^+$, then by Proposition 3.4.5, $w^m a \in Q_\odot(\Sigma)$ and $aw^m \in Q_\odot(\Sigma)$ for some $m \geq 1$. This proves that $Q_\odot(\Sigma)$ is right and left dense. □

Proposition 3.4.8 *Let $\circ$ be a plus-closed and θ-propagating bw-operation and $L \subseteq \Sigma^+$ a non-empty $\circ$-closed language such that L^c is also $\circ$-closed. Let $F(L)$ be the set of length-minimal words of L and $P_\circ(L) = L \cap Q_\circ(\Sigma)$. Then*

1. *If $w \in L$ and if u is a $\circ$-root of w, then $u \in L$.*
2. *If L' is a $\circ$-closed language containing $P_\circ(L)$, then $L \subseteq L'$.*
3. *Every word $w \in F(L)$ is $\circ$-primitive.*

Proof 1. Since u is a $\circ$-root of w, $w \in u^{\circ(n)}$, for some $n \geq 1$. If $u \in L^c$, then, since L^c is $\circ$-closed, $u^{\circ(n)} = (u \circ^{n-1} u) \subseteq L^c$ and therefore, $w \in L^c$, which is a contradiction. Hence $u \in L$.

2. Let $w \in L$, then there are two possibilities, either $w \in P_\circ(L)$ or $w \notin P_\circ(L)$. If $w \in P_\circ(L)$, then $w \in L'$ as $P_\circ(L) \subseteq L'$. If $w \notin P_\circ(L)$ then w is not $\circ$-primitive. That means there exists a $\circ$-primitive word u and $n \in \mathbb{N}$ such that $w \in u^{\circ(n)}$. But as u is $\circ$-primitive, $u \in P_\circ(L) \subseteq L'$, so $w \in L'$. So, we have showed that in both cases $L \subseteq L'$.

3. Assume that $w \in F(L)$ is not $\circ$-primitive. Then by Proposition 3.4.1, $w \in u^{\circ(n)}$, for some $\circ$-primitive word u and $n > 1$. By (1), $u \in L$.

   Case (i): There is no $a \in \Sigma$ such that $\theta(a) = a$. Then, as Proposition 3.3.9 holds,

$$|w| = \frac{1}{2} \sum_{a \in \Sigma, a \neq \theta(a)} |w|_{a,\theta(a)} > \frac{1}{2} \sum_{a \in \Sigma, a \neq \theta(a)} |u|_{a,\theta(a)} = |u|$$

   which contradicts the fact that $w \in F(L)$.

   Case (ii): There exists $a \in \Sigma$ such that $\theta(a) = a$. Then as Proposition 3.3.9 holds true,

$$\begin{aligned} |w| &= \sum_{a \in \Sigma, a = \theta(a)} |w|_{a,\theta(a)} + \frac{1}{2} \sum_{a \in \Sigma, a \neq \theta(a)} |w|_{a,\theta(a)} \\ &> \sum_{a \in \Sigma, a = \theta(a)} |u|_{a,\theta(a)} + \frac{1}{2} \sum_{a \in \Sigma, a \neq \theta(a)} |u|_{a,\theta(a)} = |u| \end{aligned}$$

   which contradicts the fact that $w \in F(L)$. □

3.5 Closure properties and language equations

In this section we will briefly discuss the closure properties of families of languages under *θ-catenation* and explore language equations involving this operation.

Proposition 3.5.1 *The families of regular, context-free and context-sensitive languages are closed under the operation of θ-catenation.*

Binary word operations can be extended naturally to binary language operations by defining,

$$L_1 \diamond L_2 = \bigcup_{u \in L_1, v \in L_2} (u \diamond v)$$

Language equations of type $L \diamond Y = R$ and $X \diamond L = R$, where $\diamond$ is an invertible binary word operation and L and R are two given languages have been extensively studied, e.g., in [15]. Finding the solutions to such equations involves the concept of "right inverse" and "left inverse" of an operation.

Definition 3.3 *[15] Let $\circ$ and $\diamond$ be two binary word operations. The operation $\diamond$ is said to be the right-inverse of the operation $\circ$ if for all words u, v, w over the alphabet Σ the following relation holds:*

$$w \in (u \circ v) \text{ iff } v \in (u \diamond w).$$

Definition 3.4 *[15] Let $\circ$ and $\bullet$ be two binary word operations. The operation $\bullet$ is said to be the left-inverse of the operation $\circ$ if for all words u, v, w over the alphabet Σ, the following relation holds:*

$$w \in (u \circ v) \text{ iff } u \in (w \bullet v).$$

Proposition 3.5.2 and 3.5.3 find the right and left inverses of *θ-catenation* for θ morphic as well as antimorphic involution. Given a bw-operation $\circ$, the reverse of this operation, denoted by $\circ'$, is defined as

$$u \circ' v = v \circ u.$$

Proposition 3.5.2 *If θ is a morphic or antimorphic involution then the right-inverse of the operation of θ-catenation $\odot$ is the reverse left θ-quotient.*

Proof Let θ be a morphic involution, and let $w \in u \odot v$. Then either $w = uv$ or $w = u\theta(v)$. By the definition of left quotient, $w = uv$ implies that $v = u^{-1}w$. Also, $w = u\theta(v)$ which implies that $\theta(w) = \theta(u)v$ and thus that $v = \theta(u)^{-1}\theta(w)$. This shows that $v \in \{u^{-1}w, \theta(u)^{-1}\theta(w)\} = u \triangleleft'_{\odot} w$. The converse is similar.

Let θ be an antimorphic involution and let $w \in u \odot v$. Then either $w = uv$ or $w = u\theta(v)$. By the definition of left quotient, $w = uv$ implies that $v = u^{-1}w$. Also, $w = u\theta(v)$ implies that $\theta(w) = v\theta(u)$. Then, by the definition of right quotient, $\theta(w) = v\theta(u)$ which implies that $v = \theta(w)\theta(u)^{-1}$. This shows that

$$v \in \{u^{-1}w, \theta(w)\theta(u)^{-1}\} = u \triangleleft'_{\odot} w.$$

The converse is similar. □

Proposition 3.5.3 *Let θ be a morphic or antimorphic involution, and let the binary word operation $\bullet$ be defined as $w \bullet v = \{wv^{-1}, w\theta(v)^{-1}\}$. Then θ-catenation and $\bullet$ are left inverses of each other.*

Proof Let $w \in u \odot v$. Then either $w = uv$ or $w = u\theta(v)$. By definition of right quotient, $w = uv$ implies $u = wv^{-1}$. Also, $w = u\theta(v)$ implies $u = w\theta(v)^{-1}$. This shows that $u \in \{wv^{-1}, w\theta(v)^{-1}\} = w \bullet v$. The converse is similar. □

The preceding results provide tools to solve language equations involving the operation of θ-catenation. The following two propositions are consequences of more general results from [15].

Proposition 3.5.4 *Let L, R be languages over an alphabet Σ. If the equation $L \odot Y = R$ has a solution Y, then the language $R' = (L \triangleleft'_{\odot} R^c)^c$ is also a solution of the equation. Moreover, R' includes all the other solutions of the equation (set inclusion).*

Corollary 3.5.5 *Let L be a language in Σ^*. If the equation $L \odot Y = L$ has a solution, then $\rho_{\odot}(L)$, the set of all right $\odot$-residuals of L is a solution, which moreover includes all the other solutions to the equation.*

Proof By the previous proposition, if a solution to the equation $L \odot Y = L$ exists, then also $R' = (L \triangleleft'_{\odot} L^c)^c = (L^c \triangleleft_{\odot} L)^c$ is a solution. By a result in [12], for any language $L \subseteq \Sigma^*$ and

bw-operation $\circ$, the set of all right $\circ$ residuals of L, denoted by $\rho_\circ(L)$, equals $(\triangleleft_\circ(L^c, L))^c$, which proves the statement of the corollary. □

Proposition 3.5.6 *Let L,R be languages over an alphabet* Σ. *If the equation* $X \odot L = R$ *has a solution* $X \subseteq \Sigma^*$, *then also the language* $R' = (R^c \triangleleft'_\odot L)^c$ *is a solution of the equation. Moreover,* R' *includes all the other solutions of the equation (set inclusion).*

3.6 Conclusions and future work

This paper proposes and investigates the operation of θ-catenation, that generates the pseudo-powers (θ-powers) of a word. An avenue of further research is to determine and investigate operations that generate other types of generalized powers. One such type is the Abelian power, [8] defined as follows.

A word w is an *k-th Abelian power* if $w = u_1u_2\cdots u_k$ for some $u_1, u_2, \cdots u_k$, $u_i \in \Sigma^+$, $1 \leq i \leq k$, such that for all $1 \leq i, j \leq k$, $\pi(u_i) = \pi(u_j)$, where $\pi(u)$ denotes the set of all words obtained by permuting the letters of u. A word w is *Abelian primitive* if w fails to be a k-th Abelian power for every $k \geq 2$. A word u is an *Abelian root* of w if $w = uu_1u_2\cdots u_{k-1}$ for some $u_1\cdots u_{k-1} \in \Sigma^+$ with $\pi(u) = \pi(u_i)$ for all $1 \leq i \leq k-1$. Unlike words that are not primitive or not θ-primitive, a word that is not Abelian primitive may have several Abelian roots.

We can now define a bw-operation $\boxdot$, called *Abelian-catenation*, as $u \boxdot v = u\pi(v)$. For example, if we consider the alphabet $\Sigma = \{a, b, c\}$ and the words $u = acba$ and $v = bcc$, then

$$u \boxdot v = \{acbabcc, acbacbc, acbaccb\}.$$

The operation of *Abelian-catenation* is length-increasing as well as propagating, but its neither left-inclusive nor right-inclusive and therefore is not plus-closed.

Note that the operation of Abelian-catenation generates *Abelian-powers*. Indeed, if $w \in u^{\boxdot(k)}$, for $k \geq 1$, then $w = uv_1v_2\cdots v_{k-1}$, where $v_i \in \{\pi(u)\}$, for $1 \leq i \leq k-1$.

Bibliography

[1] A. Carpi and A. de Luca. Periodic-like words, periodicity, and boxes. *Acta Informatica*, 37(8):597–618, 2001.

[2] S. Constantinescu and L. Ilie. Fine and Wilf's theorem for abelian periods. *Bulletin of the EATCS*, 89:167–170, 2006.

[3] M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on Strings*. Cambridge University Press, 2007.

[4] M. Crochemore and W. Rytter. *Jewels of Stringology*. World Scientific, 2002.

[5] L. J. Cummings and W. F. Smyth. Weak repetitions in strings. *J. Combinatorial Mathematics and Combinatorial Computing*, 24:33–48, 1997.

[6] E. Czeizler, L. Kari, and S. Seki. On a special class of primitive words. *Theoretical Computer Science*, 411:617 – 630, 2010.

[7] A. de Luca and A. de Luca. Pseudopalindrome closure operators in free monoids. *Theoretical Computer Science*, 362(13):282 – 300, 2006.

[8] M. Domaratzki and N. Rampersad. Abelian primitive words. In G. Mauri and A. Leporati, editors, *Developments in Language Theory*, volume 6795 of *Lecture Notes in Computer Science*, pages 204–215. Springer Berlin Heidelberg, 2011.

[9] P. Dömösi, G. Horváth, M. Ito, and K. Shikishima-Tsuji. Some periodicity of words and marcus contextual grammars. *Vietnam Journal of Mathematics*, 34:381–387, 2006.

[10] P. Gawrychowski, F. Manea, R. Mercaş, D. Nowotka, and C. Tiseanu. Finding pseudo-repetitions. *Leibniz International Proceedings in Informatics*, 20:257–268, 2013.

[11] J. E. Hopcroft and J. D. Ullman. *Formal Languages and their Relation to Automata*. Addison-Wesley Longman Inc., 1969.

[12] H. K. Hsiao, C. C. Huang, and S. S. Yu. Word operation closure and primitivity of languages. *J.UCS*, 8(2):243–256, feb 2002.

[13] S. Hussini, L. Kari, and S. Konstantinidis. Coding properties of DNA languages. In N. Jonoska and N. Seeman, editors, *Proc. of DNA7*, volume 2340 of *Lecture Notes in Computer Science*, pages 57–69. Springer, 2002.

[14] M. Ito and G. Lischke. Generalized periodicity and primitivity for words. *Mathematical Logic Quarterly*, 53(1):91–106, 2007.

[15] L. Kari. On language equations with invertible operations. *Theoretical Computer Science*, 132:129–150, 1994.

[16] L. Kari, S. Konstantinidis, and P. Sosík. Bond-free languages: Formalizations, maximality and construction methods. *International Journal of Foundations of Computer Science*, 16:1039–1070, 2005.

[17] L. Kari, E. Losseva, S. Konstantinidis, P. Sosík, and G. Thierrin. A formal language analysis of DNA hairpin structures. *Fundamenta Informaticae*, 71:453–475, Mar. 2006.

[18] L. Kari and K. Mahalingam. Watson-Crick conjugate and commutative words. In M. H. Garzon and H. Yan, editors, *Proc. of DNA13*, volume 4848 of *Lecture Notes in Computer Science*, pages 273–283. Springer-Verlag, 2008.

[19] L. Kari and S. Seki. On pseudoknot-bordered words and their properties. *Journal of Computer and System Sciences*, 75:113 – 121, 2009.

[20] L. Kari and S. Seki. An improved bound for an extension of Fine and Wilf's theorem and its optimality. *Fundamenta Informaticae*, 101:215–236, 2010.

[21] L. Kari and G. Thierrin. Word insertions and primitivity. *Utilitas Mathematica*, 53:49–61, 1998.

[22] G. Lischke. Primitive words and roots of words. *Acta Universitatis Sapientiae*, 3:5–34, 2011.

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

[24] R. C. Lyndon and M. P. Schutzenberger. The equation $a^M = b^N c^P$ in a free group. *Michigan Math. J.*, 9:289–298, 1962.

[25] G. Paun, G. Rozenberg, and T. Yokomori. Hairpin languages. *Int. J. Found. Comput. Sci.*, 12:837–847, 2001.

[26] S.-S. Yu. *Languages and Codes*. Tsang Hai Book Publishing Co., 2005.

[27] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

Chapter 4

Pseudo-Identities and Bordered Words

4.1 Introduction

Periodicity, primitivity, and repetitions of words are fundamental properties in combinatorics on words and formal language theory. Their applications include pattern-matching algorithms (see *e.g.* [3], and [4]) and data-compression algorithms (see, *e.g.*, [23]). Sometimes motivated by their applications, these classical notions have been modified in various ways that, in essence, replace the identity function with a pseudo-identity, and the notion of repetition with the notion of pseudo-repetition. A representative example is the "weak periodicity" of [5] whereby a word is called *weakly periodic* if it consists of repetitions of words with the same Parikh vector. This type of period was also called *Abelian period* in [2]. Carpi and de Luca extended the notion of periodic words to that of periodic-like words, according to the extendability of factors of a word [1].

Czeizler, Kari, and Seki have proposed and investigated the notion of *pseudo-primitivity* (and pseudo-periodicity) of words in [6, 20], motivated by the properties of information encoded as DNA strands. One of the particularities of information encoded as DNA strands is

[0]A version of this chapter has been published (L. Kari, M.S. Kulkarni. Pseudo-identities and bordered words. In G. Păun, G. Rozenberg, A. Salomaa editors, *Discrete Mathematics and Computer Science*, Editura Academiei Române, 2014, 207-222)

that a word u over the DNA alphabet $\{A, C, G, T\}$ contains basically the same information as its Watson-Crick complement, denoted here by $\theta(u)$. This led to natural as well as theoretically interesting extensions of the notion of "identity", leading to several new notions in combinatorics on words and formal language theory such as pseudo-palindrome [7], pseudo-commutativity [18], as well as hairpin-free and bond-free languages (*e.g.*, [13, 14, 15, 19, 21]). In this context, Watson-Crick complementarity has been modeled mathematically by an antimorphic involution θ over an alphabet Σ, i.e., a function that is an antimorphism, $\theta(uv) = \theta(v)\theta(u)$, $\forall u, v \in \Sigma^*$, and an involution, $\theta(\theta(x)) = x$, $\forall x \in \Sigma^*$.

In [16], given a morphic or antimorphic involution θ, a nonempty word u was defined to be θ-bordered if there exists $v \in \Sigma^+$ that is a proper prefix of u, while $\theta(v)$ is a proper suffix of u. A nonempty word u was called θ-unbordered if it was not θ-bordered, and properties of θ-bordered and θ-unbordered words were investigated in [16], [17]. Other generalizations of the classical notions of bordered and unbordered words include pseudo-knot-bordered words, defined in [19] as nonempty words w with the property that $w = xy\alpha = \beta\theta(yx)$ for some words x, y, α, and β.

In [8, 9, 10], studies of θ-periodicity have been extended to consider the cases where the morphism or antimorphism θ is literal, non-erasing or uniform. We continue this line of study by extending the investigation of θ-bordered words from the case of morphic or antimorphic involutions θ to cases where θ^n is the identity function, for some $n \geq 2$, and the case where θ is a literal morphism or antimorphism. We study properties of θ-(un)bordered words in Section 4.3, some properties of the set θ-(un)bordered words where θ is a morphic involution in Section 4.4, and conclude with several directions of further research in Section 4.5.

4.2 Basic definitions and notations

An alphabet Σ is a finite non-empty set of symbols. Σ^* denotes the set of all words over Σ, including the empty word λ. Σ^+ is the set of all non-empty words over Σ. The length of a word

$u \in \Sigma^*$ (*i.e.* the number of symbols in a word) is denoted by $|u|$. By Σ^m we denote the set of all words of length $m > 0$ over Σ. The complement of a language $L \subseteq \Sigma^*$ is $L^c = \Sigma^* \setminus L$. A word is called *primitive* if it cannot be expressed as a power of another word. Let Q denote the set of all primitive words. A function $\theta : \Sigma^* \to \Sigma^*$ is said to be a *morphism* if for all words $u, v \in \Sigma^*$ we have that $\theta(uv) = \theta(u)\theta(v)$, an *antimorphism* if $\theta(uv) = \theta(v)\theta(u)$ and an *involution* if θ^2 is an identity on Σ^*. If for all $a \in \Sigma$, $|\theta(a)| = 1$, then θ is called *literal* (anti)morphism[1]. A θ-power of a word u is a word of the form $u_1 u_2 \cdots u_n$ for $n \geq 1$ where $u_1 = u$ and $u_i \in \{u, \theta(u)\}$ for $2 \leq i \leq n$. A word is called *θ-primitive* if it cannot be expressed as a θ-power of another word. Let Q_θ denote the set of all θ-primitive words.

For a language $L \subseteq \Sigma^*$, the *principal congruence* P_L determined by L is defined as follows: for any $x, y \in \Sigma^*$ such that $x \neq y$, $x \equiv y(P_L)$ if and only if $uxv \in L \Leftrightarrow uyv \in L$ for all $u, v \in \Sigma^*$. The index of P_L is the number of equivalence classes of P_L. L is said to be *disjunctive* if P_L is the identity, *i.e.*, for any $x \neq y \in \Sigma^*$ there exists $u, v \in \Sigma^*$ such that $uxv \in L$ and $uyv \notin L$ or vice versa.

A language $L \subseteq \Sigma^*$ is said to be *dense* if for all $u \in \Sigma^*$, $L \cap \Sigma^* u \Sigma^* \neq \emptyset$.

Definition 4.1 *1. For $v, w \in \Sigma^*$, $w \leq_p v$ iff $v \in w\Sigma^*$.*

2. For $v, w \in \Sigma^*$, $w \leq_s v$ iff $v \in \Sigma^* w$.

3. $\leq_d = \leq_p \cap \leq_s$.

4. For $u \in \Sigma^*$, $v \in \Sigma^*$ is said to be a border of u if $v \leq_d u$, i.e., $u = vx = yv$.

5. For $v, w \in \Sigma^*$, $w <_p v$ iff $v \in w\Sigma^+$.

6. For $v, w \in \Sigma^*$, $w <_s v$ iff $v \in \Sigma^+ w$.

7. $<_d = <_p \cap <_s$.

8. For $u \in \Sigma^*$, $v \in \Sigma^*$ is said to be a proper border of u if $v <_d u$.

[1] By (anti)morphism we mean either a morphism or an antimorphism.

9. *For* $u \in \Sigma^+$, $L_d(u) = \{v \in \Sigma^* | v <_d u\}$.

10. $\nu_d(u) = |L_d(u)|$.

11. $D(i) = \{u \in \Sigma^+ | \nu_d(u) = i\}$.

12. *A word* $u \in \Sigma^+$ *is said to be a bordered word if there exists* $v \in \Sigma^+$ *such that* $v <_d u$, *i.e.,* $u = vx = yv$ *for some* $x, y \in \Sigma^+$.

13. *A non-empty word which is not bordered is called unbordered.*

For a word w, $\text{Pref}(w) = \{u \in \Sigma^+ | \exists v \in \Sigma^*, w = uv\}$ and $\text{Suff}(w) = \{u \in \Sigma^+ | \exists v \in \Sigma^*, w = vu\}$ denotes the set of all prefixes and suffixes respectively. Similarly, the set of proper prefixes and proper suffixes of a word w can be defined as $\text{PPref}(w) = \{u \in \Sigma^+ | \exists v \in \Sigma^+, w = uv\}$ and $\text{PSuff}(w) = \{u \in \Sigma^+ | \exists v \in \Sigma^+, w = vu\}$ respectively.

Definition 4.2 *[16] Let* θ *be either a morphism or an antimorphism on* Σ^*.

1. *For* $v, w \in \Sigma^*$, $w \leq_p^\theta v$ *iff* $v \in \theta(w)\Sigma^*$.

2. *For* $v, w \in \Sigma^*$, $w \leq_s^\theta v$ *iff* $v \in \Sigma^*\theta(w)$.

3. $\leq_d^\theta = \leq_p \cap \leq_s^\theta$.

4. *For* $u \in \Sigma^*$, $v \in \Sigma^*$ *is said to be a* θ*-border of* u *if* $v \leq_d^\theta u$, *i.e.,* $u = vx = y\theta(v)$.

5. *For* $w, v \in \Sigma^*$, $w <_p^\theta v$ *iff* $v \in \theta(w)\Sigma^+$.

6. *For* $w, v \in \Sigma^*$, $w <_s^\theta v$ *iff* $v \in \Sigma^+\theta(w)$.

7. $<_d^\theta = <_p \cap <_s^\theta$.

8. *For* $u \in \Sigma^*$, $v \in \Sigma^*$ *is said to be a proper* θ*-border of* u *if* $v <_d^\theta u$.

9. *For* $u \in \Sigma^+$, *define* $L_d^\theta(u) = \{v \in \Sigma^* | v <_d^\theta u\}$.

10. $\nu_d^\theta(u) = |L_d^\theta(u)|$.

11. $D_\theta(i) = \{u \in \Sigma^+ | \nu_d^\theta(u) = i\}$.

12. *A word* $u \in \Sigma^+$ *is said to be* θ*-bordered if there exists* $v \in \Sigma^+$ *such that* $v <_d^\theta u$*, i.e.,* $u = vx = y\theta(v)$ *for some* $x, y \in \Sigma^+$.

13. *A nonempty word which is not* θ*-bordered is called* θ*-unbordered. Thus,* $D_\theta(1)$ *is the set of all* θ*-unbordered words over* Σ.

For $u, v \in \Sigma^*$, [11] calls $u <_d x_1 <_d x_2 <_d \cdots <_d v$ a $u - v$ chain. A $u - v$ chain, $u = x_1 <_d x_2 <_d \cdots <_d x_n = v$ is said to be *maximal* if for $u' \in \Sigma^*$, $u <_d u' <_d v$ implies $u' = x_i$ for some $1 < i < n$. Similarly, we can define $u -_\theta v$ chain as a sequence $u = x_1 <_d^\theta x_2 <_d^\theta \cdots <_d^\theta x_n = v$. The notion of maximal chain can be extended to that of θ-maximal chain in a similar fashion.

4.3 Properties of pseudo-(un)bordered words

In this section, we study some basic properties of θ-bordered and θ-unbordered words where θ is a (anti)morphism with the property that $\theta^n = I$ on Σ^* for $n \geq 2$ or any literal (anti)morphism. In the case where $\theta^n = I$ and θ is an antimorphism, it is clear that n has to be an even number.

The following result was proved in [11], and can be easily generalized to the case of morphic involutions.

Lemma 4.3.1 *[11] Let* $u \in \Sigma^+ \backslash D(1)$*. Then there exists* $v \in \Sigma^*$ *with* $|v| \leq \frac{|u|}{2}$ *such that* $v <_d u$.

Lemma 4.3.2 *Let* θ *be a morphic or an antimorphic involution and let* $u \in \Sigma^+ \backslash D_\theta(1)$*. Then there exists* $v \in \Sigma^*$ *with* $|v| \leq \frac{|u|}{2}$ *such that* $v <_d^\theta u$.

The next two results, Propositions 4.3.3 and 4.3.4, establish some relations between the set of θ-borders of a word u, namely $L_d^\theta(u)$, and the set of θ-borders of $\theta(u)$, namely $L_d^\theta(\theta(u))$.

Proposition 4.3.3 *Let* $u \in \Sigma^+$*. Then for a morphism* θ *on* Σ^* *such that* $\theta^n = I$ *for* $n > 2$*,* $L_d^\theta(\theta(u)) = \theta(L_d^\theta(u))$.

Proof Let $v \in L_d^\theta(\theta(u))$ which implies $\theta(u) = vx = y\theta(v)$ for some $x, y \in \Sigma^+$ which further implies $\theta^2(u) = \theta(v)\theta(x) = \theta(y)\theta^2(v)$. Continuing in this way, we will get $\theta^n(u) = \theta^{n-1}(v)\theta^{n-1}(x) = \theta^{n-1}(y)\theta^n(v)$ and thus $u = \theta^{n-1}(v)\theta^{n-1}(x) = \theta^{n-1}(y)\theta^n(v)$ which implies $\theta^{n-1}(v) \in L_d^\theta(u)$ and hence $v \in \theta(L_d^\theta(u))$. Thus, $L_d^\theta(\theta(u)) \subseteq \theta(L_d^\theta(u))$.
Conversely, let $v \in L_d^\theta(u)$ which implies $u = vx = y\theta(v)$ for $x, y \in \Sigma^+$ and hence $\theta(u) = \theta(v)\theta(x) = \theta(y)\theta^2(v)$ which further implies $\theta(v) \in L_d^\theta(\theta(u))$. Also, since $v \in L_d^\theta(u)$, $\theta(v) \in \theta(L_d^\theta(u))$. Thus, $L_d^\theta(\theta(u)) = \theta(L_d^\theta(u))$.

However, if θ is literal (anti)morphism that is not bijective, Proposition 4.3.3 does not necessarily hold, as demonstrated by Example 4.1.

Example 4.1 *Let $\Sigma = \{a, b\}$ and θ be (anti)morphism such that, $\theta(a) = a, \theta(b) = a$, $u = ababaa$. Then $\theta(u) = aaaaaa$, $L_d^\theta(u) = \{\lambda, a, ab\}$, $\theta(L_d^\theta(u)) = \{\lambda, a, aa\}$, $L_d^\theta(\theta(u)) = \{\lambda, a, aa, \cdots, aaaaa\}$. Clearly, $L_d^\theta(\theta(u)) \neq \theta(L_d^\theta(u))$.*

Note that the inclusion $\theta(L_d^\theta(u)) \subseteq L_d^\theta(\theta(u))$ holds in case of Example 4.1. Moreover, the inclusion holds in general for any literal morphism θ.

Proposition 4.3.4 *Let $u \in \Sigma^+$. Then for any literal morphism θ on Σ^*, $\theta(L_d^\theta(u)) \subseteq L_d^\theta(\theta(u))$.*

Proof Let $v \in L_d^\theta(u)$ which implies $u = vx = y\theta(v)$ for $x, y \in \Sigma^+$ and hence $\theta(u) = \theta(v)\theta(x) = \theta(y)\theta^2(v)$ which further implies $\theta(v) \in L_d^\theta(\theta(u))$. Also, since $v \in L_d^\theta(u)$, $\theta(v) \in \theta(L_d^\theta(u))$. Thus, $\theta(L_d^\theta(u)) \subseteq L_d^\theta(\theta(u))$.

It is known, [16], that, for an antimorphic involution θ, the relation $<_d^\theta$ is transitive.

Lemma 4.3.5 *[16] Let $u \in \Sigma^*$ and $v, w \in \Sigma^+$ such that $u <_d^\theta w$ and $w <_d^\theta v$. Then for a morphic involution θ, we have $u <_d v$ and for an antimorphic involution θ, we have $u <_d^\theta v$.*

The statement of Lemma 4.3.5 does not necessarily hold in the case when θ is a morphism which is literal and not bijective, as demonstrated by Example 4.2.

Example 4.2 *Let $\Sigma = \{a, b\}$ and θ be a morphism such that $\theta(a) = a$, $\theta(b) = a$, $u = ab$, $w = abaa$, $v = abaabbaaaa$. Then $u <_d^\theta w$ and $w <_d^\theta v$ but $u \not<_d v$.*

The following proposition demonstrates the transitivity of relation $<_d^\theta$ for literal antimorphisms θ.

Proposition 4.3.6 *If θ is any literal antimorphism on Σ^*, then the relation $<_d^\theta$ is transitive, i.e. for $u \in \Sigma^*$ and $v, w \in \Sigma^+$ such that $u <_d^\theta w$ and $w <_d^\theta v$, we have $u <_d^\theta v$.*

Proof Let θ be any literal antimorphism such that $u <_d^\theta w$ and $w <_d^\theta v$ which implies $w = ux = y\theta(u)$ and $v = w\alpha = \beta\theta(w)$ for some $x, y, \alpha, \beta \in \Sigma^+$, hence $v = ux\alpha = \beta\theta(ux)$ which further implies $v = ux\alpha = \beta\theta(x)\theta(u)$. Hence $u <_d^\theta v$.

Corollary 4.3.7 *Let $v \in L_d^\theta(u)$ and $w \in \Sigma^+$. Then for any literal antimorphism θ on Σ^*, if $w <_d^\theta v$ then $w \in L_d^\theta(u)$.*

The converse of the Corollary 4.3.7 does not hold in general. In fact, in the case of an antimorphism, Proposition 4.3.9 holds.

The next results describe relations between the θ-borders of a word u when θ is a morphism with $\theta^n = I, n > 2$, (Proposition 4.3.8) or literal (anti)morphisms (Proposition 4.3.9).

Proposition 4.3.8 *Let $u, v, w \in \Sigma^+$, $u \neq v$ and $u <_d^\theta w, v <_d^\theta w$. If θ is a morphism on Σ^* such that $\theta^n = I$ for $n > 2$, then either $v <_d u$ or $u <_d v$.*

Proof Let θ be a morphism such that $\theta^n = I$ and $u <_d^\theta w, v <_d^\theta w$ which implies $w = ux = y\theta(u)$ and $w = v\alpha = \beta\theta(v)$ for some $x, y, \alpha, \beta \in \Sigma^+$. If $|u| > |v|$, then $u = vp$ and $\theta(u) = q\theta(v)$ for some $p, q \in \Sigma^+$ which imply $\theta^n(u) = \theta^{n-1}(q)\theta^n(v) = \theta^{n-1}(q)v$. Thus, we get $u = vp = \theta^{n-1}(q)v$ which implies $v <_d u$. Similarly, if $|u| < |v|$ then $v = up'$ and $\theta(v) = q'\theta(u)$ for some $p', q' \in \Sigma^+$ which imply $\theta^n(v) = \theta^{n-1}(q')\theta^n(u) = \theta^{n-1}(q')u$. Thus, we get $v = up' = \theta^{n-1}(q')u$ which implies $u <_d v$.

Proposition 4.3.8 does not necessarily hold if θ is a literal (anti)morphism that is not bijective, as demonstrated by Example 4.3.

Example 4.3 *Let $\Sigma = \{a, b\}$, and θ be a morphism or antimorphism such that $\theta(a) = a, \theta(b) = a$, $u = ab$, $v = abaa$, and $w = abaabbaaaa$. Then $u <_d^\theta w, v <_d^\theta w$ but neither $v <_d u$ nor $u <_d v$.*

Proposition 4.3.9 *Let $u, v, w \in \Sigma^+$, $u \neq v$ and $u <_d^\theta w, v <_d^\theta w$. Then for any literal morphism θ on Σ^*, either $\theta(v) <_d \theta(u)$ or $\theta(u) <_d \theta(v)$. If θ is any literal antimorphism, then either $v <_p u$ or $u <_p v$.*

Proof Let θ be any literal morphism and $u <_d^\theta w, v <_d^\theta w$ which imply $w = ux = y\theta(u)$ and $w = v\alpha = \beta\theta(v)$ for some $x, y, \alpha, \beta \in \Sigma^+$. If $|u| > |v|$, then $u = vp$ and $\theta(u) = q\theta(v)$ for some $p, q \in \Sigma^+$ which imply $\theta(u) = \theta(v)\theta(p) = q\theta(v)$. Thus, we get $\theta(v) <_d \theta(u)$. Similarly, if $|u| < |v|$ then $v = up'$ and $\theta(v) = q'\theta(u)$ for some $p', q' \in \Sigma^+$ which imply $\theta(v) = \theta(u)\theta(p') = q'\theta(u)$. Thus, we get $\theta(u) <_d \theta(v)$.

Let θ be any literal antimorphism and $u <_d^\theta w, v <_d^\theta w$ which imply that $w = ux = y\theta(u)$ and $w = v\alpha = \beta\theta(v)$ for some $x, y, \alpha, \beta \in \Sigma^+$. Hence, we have, $ux = v\alpha$. If $|u| > |v|$, $v <_p u$ and if $|v| > |u|$ then $u <_p v$.

Corollary 4.3.10 *Let $u, v, w \in \Sigma^+$, $u \neq v$ and $u <_d^\theta w$, $v <_d^\theta w$. Then for any literal antimorphism θ on Σ^*, either $\theta(v) <_s \theta(u)$ or $\theta(u) <_s \theta(v)$.*

Corollary 4.3.11 *Let $u \in \Sigma^+$. Then*

1. *For any morphism θ on Σ^* such that $\theta^n = I$ for $n > 2$, $L_d^\theta(u)$ is a totally ordered set with $<_d$, i.e. $L_d^\theta(u) = \{\lambda <_d u_1 <_d u_2 <_d \cdots <_d u_{i-1}\}$.*

2. *For any literal morphism θ on Σ^*, $\theta(L_d^\theta(u))$ is a totally ordered set with $<_d$.*

3. *For any literal antimorphism θ on Σ^*, $L_d^\theta(u)$ is a totally ordered set with $<_p$, i.e. $L_d^\theta(u) = \{\lambda <_p u_1 <_p u_2 <_p \cdots <_p u_{i-1}\}$ and $\theta(L_d^\theta(u))$ is a totally ordered set with $<_s$.*

Proof Statement 1 follows from Proposition 4.3.8, statement 2 from Proposition 4.3.9 and statement 3 from Proposition 4.3.9 and Corollary 4.3.10, respectively.

The next two propositions (Proposition 4.3.12, 4.3.13) list some properties of θ-unbordered words for (anti)morphisms θ such that $\theta^n = I, n > 2$.

Proposition 4.3.12 *Let θ be a morphism on Σ^* such that $\theta^n = I$ for $n > 2$. Then for all $x, y \in D_\theta(1)$ such that $x \neq y$, we have that $xy \neq \theta^{n-1}(y)x$.*

Proof Let $x, y \in D_\theta(1)$. As $D_\theta(i) \subseteq \Sigma^+$ for $i \geq 1$, both x and y are non-empty. Suppose $xy = \theta^{n-1}(y)x$, then we have following three cases to consider.

Case 1: $|x| = |y|$. Then $x = \theta^{n-1}(y)$ and $y = x$, which is a contradiction since $x \neq y$.

Case 2: $|x| > |y|$. Then there exists $p \in \Sigma^+$ such that $x = \theta^{n-1}(y)p$ and $x = py$ which imply that $x = \theta^{n-1}(y)p = p\theta^n(y)$, which is a contradiction since $x \in D_\theta(1)$.

Case 3: $|y| > |x|$. Then there exists $q \in \Sigma^+$ such that $\theta^{n-1}(y) = xq$ and $y = qx$ which imply that $y = qx = \theta(x)\theta(q)$, which is a contradiction since $y \in D_\theta(1)$.

Since all the three cases leads to a contradiction $xy \neq \theta^{n-1}(y)x$.

Proposition 4.3.13 *Let θ be an antimorphism on Σ^* such that $\theta^n = I$ for $n > 2$. Then for $x \in D_\theta(1)$ and $y \in \Sigma^+$ such that $x \neq y$ and $\theta(x) \neq x$, we have that $xy \neq \theta^{n-1}(y)x$.*

Proof Let $x \in D_\theta(1)$. As $D_\theta(i) \subseteq \Sigma^+$ for $i \geq 1$, x is non-empty. Suppose $xy = \theta^{n-1}(y)x$, then we have following three cases to consider.

Case 1: $|x| = |y|$. Then $x = \theta^{n-1}(y)$ and $y = x$, which is a contradiction since $x \neq y$.

Case 2: $|x| > |y|$. Then there exists $p \in \Sigma^+$ such that $x = \theta^{n-1}(y)p$ and $x = py$ which imply that $x = \theta^{n-1}(y)p = p\theta^n(y)$, which is a contradiction since $x \in D_\theta(1)$.

Case 3: $|y| > |x|$. Then there exists $q \in \Sigma^+$ such that $\theta^{n-1}(y) = xq$ and $y = qx$ which imply that $y = qx = \theta(q)\theta(x)$, which further implies $\theta(q) = q$ and $\theta(x) = x$ which is a contradiction since $\theta(x) \neq x$.

Since all the three cases leads to a contradiction $xy \neq \theta^{n-1}(y)x$.

The following lemma provides a necessary and sufficient condition for a word to be θ-bordered, in the case when θ is a literal antimorphism.

Lemma 4.3.14 *Let θ be any literal antimorphism on Σ^*. Then $x \in \Sigma^+$ is θ-bordered iff $x = ay\theta(a)$ for some $a \in \Sigma$ and $y \in \Sigma^*$.*

The result below gives several properties of θ-unbordered words, for literal antimorphisms θ.

Proposition 4.3.15 *Let θ be any literal antimorphism on Σ^*, then*

1. *For all $u, v \in \Sigma^+$ and $w \in \Sigma^*$, we have $uwv \in D_\theta(1)$ iff $uv \in D_\theta(1)$.*
2. *If Σ is an alphabet such that there exist $a, b \in \Sigma$ with $\theta(a) \neq b$, then $D_\theta(1)$ is a dense set.*
3. *Let $a, b \in \Sigma$ such that $a \neq b$. Then for all $u \in \Sigma^+$, either ua or ub is θ-unbordered.*

Proof 1. Suppose $uwv \in D_\theta(1)$ and $uv \notin D_\theta(1)$ which imply that $uv = ay\theta(a)$ for some $a \in \Sigma$ and $y \in \Sigma^*$. If $w = \lambda$, then clearly $uwv \notin D_\theta(1)$, a contradiction. Now, if $w \neq \lambda$, then we have three possibilities.

Case a: $u = a, v = y\theta(a)$, hence $uwv = awy\theta(a) \notin D_\theta(1)$.

Case b: $u = ay, v = \theta(a)$, hence $uwv = aywy\theta(a) \notin D_\theta(1)$.

Case c: $u = ap, v = q\theta(a)$ where $y = pq$ for some $p, q \in \Sigma^*$, hence $uwv = apwq\theta(a) \notin D_\theta(1)$.

Since all the three cases leads to a contradiction, $uv \in D_\theta(1)$.

Conversely, suppose $uwv \notin D_\theta(1)$ which imply that $uwv = ay\theta(a)$ for some $a \in \Sigma$ and $y \in \Sigma^*$. Hence, $u = au_1$ and $v = v_1\theta(a)$ for some $u_1, v_1 \in \Sigma^*$ which further implies, $uv = au_1v_1\theta(a) \notin D_\theta(1)$, a contradiction. Hence $uwv \in D_\theta(1)$.

2. Choose $a, b \in \Sigma$ such that $\theta(a) \neq b$. Then for all $w \in \Sigma^*$, there exists $a, b \in \Sigma^*$ such that $awb \in D_\theta(1)$. Hence $D_\theta(1)$ is a dense set.

3. Let us assume that both ua and ub are θ-bordered. Then we have, $ua = a_1y_1\theta(a_1)$ and $ub = a_2y_2\theta(a_2)$ for some $a_1, a_2 \in \Sigma$ and $y_1, y_2 \in \Sigma^*$ which implies $u = a_1y_1 = a_2y_2$ and $a = \theta(a_1), b = \theta(a_2)$. This further implies that $a_1y_1 = a_2y_2$ which implies $a_1 = a_2$ and $y_1 = y_2$ which further implies $a = \theta(a_2) = b$, a contradiction. Hence, either ua or ub is θ-unbordered.

If θ is an antimorphism such that $\theta^n = I, n > 2$, the following result holds.

Proposition 4.3.16 *Let θ be an antimorphism on Σ^* such that $\theta^n = I$ for $n > 2$. Then $u \in D_\theta(1)$ iff $\theta^{n-2}(u) \in D_\theta(1)$.*

Proof Let $u \in D_\theta(1)$ and suppose $\theta^{n-2}(u) \notin D_\theta(1)$ then we have $\theta^{n-2}(u) = ay\theta(a)$ for some $a \in \Sigma$ and $y \in \Sigma^*$ which imply that $u = \theta^n(u) = \theta^2(a)\theta^2(y)\theta^3(a)$ and thus $u \notin D_\theta(1)$, a contradiction. Hence $\theta^{n-2}(u) \in D_\theta(1)$.

Conversely, suppose $\theta^{n-2}(u) \in D_\theta(1)$ and $u \notin D_\theta(1)$. Then $u = ay\theta(a)$ for some $a \in \Sigma$ and $y \in \Sigma^*$. Since n is even and $\theta^n = I$, $n-2$ is also even and thus $\theta^{n-2}(u) = \theta^{n-2}(a)\theta^{n-2}(y)\theta^{n-1}(a) \notin D_\theta(1)$, a contradiction. Hence $u \in D_\theta(1)$.

Lemma 4.3.17 *Let θ be a morphic involution on Σ^* and $u \in \Sigma^+$ such that $u \in D(1)$, then $\theta(u) \in D(1)$.*

Proof Let $u \in D(1)$. Suppose $\theta(u) \notin D(1)$. Then $\theta(u) = \alpha\beta_1 = \beta_2\alpha$ for $\alpha, \beta_1, \beta_2 \in \Sigma^+$. Thus, $u = \theta(\alpha)\theta(\beta_1) = \theta(\beta_2)\theta(\alpha) \notin D(1)$, a contradiction. Thus, $\theta(u) \in D(1)$.

Along similar lines, we can prove the following result concerning $D_\theta(1)$ for a morphism of the form $\theta^n = I, n \geq 2$.

Lemma 4.3.18 *Let θ be a morphism on Σ^* such that $\theta^n = I$, $n \geq 2$ and $u \in \Sigma^+$. Then the following are equivalent:*

1. $u \in D_\theta(1)$.

2. $\theta^{n-1}(u) \in D_\theta(1)$.

3. $\theta(u) \in D_\theta(1)$.

Proof (1) $\Rightarrow$ (2): Let $u \in D_\theta(1)$ and suppose $\theta^{n-1}(u) \notin D_\theta(1)$. Then $\theta^{n-1}(u) = vx = y\theta(v)$ for some $v, x, y \in \Sigma^+$. This implies $u = \theta(v)\theta(x) = \theta(y)\theta^2(v)$, a contradiction since $u \in D_\theta(1)$. Hence $\theta^{n-1}(u) \in D_\theta(1)$.

(2) $\Rightarrow$ (3): Let $\theta^{n-1}(u) \in D_\theta(1)$ and suppose $\theta(u) \notin D_\theta(1)$. Then $\theta(u) = vx = y\theta(v)$ for some $v, x, y \in \Sigma^+$. This implies $\theta^{n-1}(u) = \theta^{n-2}(v)\theta^{n-2}(x) = \theta^{n-2}(y)\theta^{n-1}(v)$, a contradiction since $\theta^{n-1}(u) \in D_\theta(1)$. Hence $\theta(u) \in D_\theta(1)$.

(3) $\Rightarrow$ (1): Let $\theta(u) \in D_\theta(1)$ and suppose $u \notin D_\theta(1)$. Then $u = vx = y\theta(v)$ for some $v, x, y \in \Sigma^+$. This implies $\theta(u) = \theta(v)\theta(x) = \theta(y)\theta^2(v)$, a contradiction since $\theta(u) \in D_\theta(1)$. Hence $u \in D_\theta(1)$.

In fact, the implication $\theta^{n-2}(u) \in D_\theta(1) \Rightarrow u \in D_\theta(1)$ of Proposition 4.3.16 and implications (2) $\Rightarrow$ (3) and (3) $\Rightarrow$ (1) in Lemma 4.3.18 hold if θ is a literal morphism, not necessarily bijective.

Proposition 4.3.19 *Let θ be a morphism on Σ^* such that $\theta^n = I$ and $u \in \Sigma^+$. If $u \in D_\theta(i)$ for some $i \geq 2$, then for all $1 \leq k < i$, $L_d^\theta(u) \cap D(k) \neq \emptyset$.*

Proof By Corollary 4.3.11 we have

$$L_d^\theta(u) = \{\lambda <_d u_1 <_d u_2 <_d \cdots <_d u_{i-1}\}.$$

Note that $u_k <_d^\theta u$ for all $1 \leq k \leq i-1$. Now, since $u_j \in L_d^\theta(u)$ and $|u_j| < |u_k|$ for all $1 \leq j < k$, by Proposition 4.3.8 we have that $u_j <_d u_k$. Hence,

$$L_d(u_k) = \{\lambda, u_1, \cdots u_{k-1}\}.$$

Thus $u_k \in D(k)$ and $L_d^\theta(u) \cap D(k) \neq \emptyset$.

Recall that, a $u -_\theta v$ chain, $u = x_1 <_d^\theta x_2 <_d^\theta \cdots <_d^\theta x_n = v$ is said to be θ-maximal if for $u' \in \Sigma^*$, $u <_d^\theta u' <_d^\theta v$ implies $u' = x_i$ for some $1 < i < n$.

Lemma 4.3.20 *[6] Let $u \in \Sigma^+$ be a primitive word. Then u cannot be a factor of u^2 in a nontrivial way, i.e., if $u^2 = xuy$, then necessarily either $x = \lambda$ or $y = \lambda$.*

Proposition 4.3.21 *Let θ be an antimorphic involution on Σ^* and $f \in Q$. If $f \leq_d^\theta u \leq_d^\theta f^2$, then $u = f$ or $u = f^2$, i.e., $f \leq_d^\theta f^2$ is a θ-maximal chain.*

Proof Suppose $f \leq_d^\theta f^2$ is not a θ-maximal chain, i.e., $u \neq f$ and $u \neq f^2$. Since $f \leq_d^\theta u \leq_d^\theta f^2$, we have $u = fx = y\theta(f)$ and $f^2 = u\alpha = \beta\theta(u)$ for $x, y, \alpha, \beta \in \Sigma^*$ with $|x| = |y|$ and $|\alpha| = |\beta|$. Then,

$$f^2 = fx\alpha = y\theta(f)\alpha = \beta\theta(x)\theta(f) = \beta f\theta(y).$$

Now, since $f^2 = \beta f\theta(y)$, by Lemma 4.3.20 either $\beta = \lambda$ or $\theta(y) = \lambda$.

Case 1: Suppose, $\beta = \lambda$. This implies $f = \theta(y)$. Since, $fx\alpha = f^2$, we get $x\alpha = f = \theta(y)$. But since, $|x| = |y|$, $x = \theta(y) = f$ and thus $u = fx = f^2$, a contradiction.

Case 2: Suppose, $\theta(y) = \lambda$. This implies $\beta = f$. Since, $fx\alpha = f^2$, we get $x\alpha = f = \beta$. But since, $|\alpha| = |\beta|$, $\alpha = \beta = f$ which implies $f^2 = u\alpha = uf$ and thus $u = f$, a contradiction.

Since both the cases leads to a contradiction, $f \leq_d^\theta f^2$ is a θ-maximal chain.

The θ-unbounded annihilator $\alpha_{ub}(u)$ of a word u is defined, [12], as

$$\alpha_{ub}(u) = \{v \in \Sigma^+ | uv \in D_\theta(1)\}.$$

The following results find a relationship between the θ-unbounded annihilator of a word u and the set of catenations of suffixes of u, for θ-unbordered words u, and morphisms θ with $\theta^n = I$, $n \geq 2$ (Proposition 4.3.22) or literal antimorphisms (Proposition 4.3.23).

Proposition 4.3.22 *Let θ be a morphism on Σ^* such that $\theta^n = I, n \geq 2$. If $u \in D_\theta(1)$, then $(PSuff(u))^+ \subseteq \alpha_{ub}(u)$.*

Proof Let $u \in D_\theta(1)$. Let $v = u_1u_2\cdots u_m$ for some $u_i \in \text{PSuff}(u)$ and $1 \leq i \leq m$. Suppose that $uv \notin D_\theta(1)$. Then there exists $\alpha, \alpha_1, \beta_1 \in \Sigma^+$ such that $uv = \alpha\alpha_1 = \beta_1\theta(\alpha)$. Then, we have following two cases:

Case 1: $|\alpha| > |v|$. Then, we have $\theta(\alpha) = u''v$ and $u = u'u''$ for some $u', u'' \in \Sigma^+$. This implies $u'' <_s u$. From $uv = \alpha\alpha_1$, we get $uv = \theta^{n-1}(u'')\theta^{n-1}(v)\alpha_1$. This implies $\theta^{n-1}(u'') <_p u$. This will further imply that $u \notin D_\theta(1)$, a contradiction.

Case 2: $|\alpha| \leq |v|$. Also, we have $v = u_1u_2\cdots u_m$ for some $u_i \in \text{PSuff}(u)$ for $1 \leq i \leq m$. Thus we have following two sub-cases:

Case 2(a): $|\alpha| < |u_m|$. Then, we have $\theta(\alpha) = u_{m''}$ and $u_m = u_{m'}u_{m''}$ for some $u_{m'}, u_{m''} \in \Sigma^+$. Since, $u_m \in \text{PSuff}(u)$, we have $u = u'_m u_m = u'_m u_{m'} u_{m''}$ for some $u'_m \in \Sigma^+$. Thus, we have $u_{m''} <_s u$. From $uv = \alpha\alpha_1$, we get $uv = \theta^{n-1}(u_{m''})\alpha_1$. This implies $\theta^{n-1}(u_{m''}) <_p u$. This will further imply that $u \notin D_\theta(1)$, a contradiction.

Case 2(b): $|\alpha| \geq |u_m|$. Then, we have $\theta(\alpha) = u''_i u_{i+1}\cdots u_m$ for $u_i = u'_i u''_i$, $u'_i \in \Sigma^*$, $u''_i \in \Sigma^+$ and $i = 1, 2, \cdots, m-1$. Since, $u_i \in \text{PSuff}(u)$, we have $u = u_{i'}u_i = u_{i'}u'_i u''_i$ for some $u_{i'} \in \Sigma^+$. Thus, we have $u''_i <_s u$. From $uv = \alpha\alpha_1$, we get $uv = \theta^{n-1}(u''_i)\theta^{n-1}(u_{i+1}\cdots u_m)\alpha_1$. This implies $\theta^{n-1}(u''_i) <_p u$. This will further imply that $u \notin D_\theta(1)$, a contradiction.

Since all the cases leads to a contradiction, $(\text{PSuff}(u))^+ \subseteq \alpha_{ub}(u)$.

Proposition 4.3.23 *Let θ be any literal antimorphism on Σ^*. If $u \in D_\theta(1)$, then $(PSuff(u))^+ \subseteq \alpha_{ub}(u)$.*

Proof Let $v = u_1u_2\cdots u_m$ for some $u_i \in \text{PSuff}(u)$ and $1 \leq i \leq m$. Suppose, $uv \notin D_\theta(1)$. Then $uv = ay\theta(a)$ for some $a \in \Sigma$ and $y \in \Sigma^*$. This further implies, $u = ay_1$, $v = y_2\theta(a)$ and $y = y_1y_2$ for some $y_1, y_2 \in \Sigma^*$. Clearly, $a <_p u$. But, since, $v = u_1u_2\cdots u_m = y_2\theta(a)$ where $u_m \in \text{PSuff}(u)$, we will have $u_m = u_{m'}\theta(a)$ for $u_{m'} \in \Sigma^*$. Also, $u = u'u_m = u'u_{m'}\theta(a)$ and thus $\theta(a) <_s u$. This imply $u \notin D_\theta(1)$, a contradiction.

4.4 Disjunctivity of the set of θ-(un)bordered words

In this section we study some properties of the set of θ-bordered and θ-unbordered words. In [11] it was shown that, for every $i \geq 1$, the set of all (un)bordered words $D(i)$ is disjunctive. Similarly, we will show that, under some conditions, if θ is a morphic involution then the set of all θ-unbordered words $D_\theta(1)$ is disjunctive, and the set of all words with exactly two θ-borders $D_\theta(2)$, are also disjunctive (Theorem 4.4.7). We also study the disjunctivity of some related languages (Theorem 4.4.13).

The following proposition provides a necessary and sufficient condition for a language to be disjunctive.

Proposition 4.4.1 *[22] Let $L \subseteq \Sigma^*$. Then the following two statements are equivalent:*

1. *L is a disjunctive language.*
2. *If $u, v \in \Sigma^+$, $u \neq v$, $|u| = |v|$, then $u \not\equiv v(P_L)$.*

The following auxiliary lemmas are needed for the main results of this section, Theorem 4.4.7 and Theorem 4.4.13.

Lemma 4.4.2 *Let θ be a morphic involution and $a, b \in \Sigma$, $a \neq b$. Let $x, y \in \Sigma^m$, $m > 0$. Then,*

1. *$a^m x\theta(b) \in D_\theta(1)$.*
2. *If $a \neq \theta(a)$, $x = \theta(b)x'$, $x' \in \Sigma^*$ and $k \geq m$, then $(a^k y\theta(b))(a^k x\theta(b)) \in D_\theta(1)$.*

Proof 1. Since there does not exist any word $u \in \Sigma^+$ with $|u| \leq m$ such that $u <_d^\theta a^m x\theta(b)$, by Lemma 4.3.2, $a^m x\theta(b) \in D_\theta(1)$.

2. Let $(a^k y\theta(b))(a^k x\theta(b)) \notin D_\theta(1)$. Then there exists $u \in \Sigma^+$ such that

$$u <_d^\theta (a^k y\theta(b))(a^k x\theta(b)).$$

By Lemma 4.3.2, it is enough to consider only the case $|u| \leq m + k + 1$.

Case (i): $|u| \leq k$. Then $u = a^n$ for some $n \leq k$ and $\theta(u) = \alpha''\theta(b)$ for $x = \alpha'\alpha'', \alpha' \in \Sigma^+, \alpha'' \in \Sigma^*$. Hence $a^n = \theta(\alpha'')b$ which implies $a = b$, a contradiction.

Case (ii): $k < |u| < m + k + 1$. Then $u = a^k y'$ for $y = y'y''$, $y' \in \Sigma^+$, $y'' \in \Sigma^*$ and $\theta(u) = a^n x\theta(b) = a^n\theta(b)x'\theta(b)$ for $0 \leq n < k$. Hence $a^k y' = \theta(a^n)b\theta(x')b$ which implies $a = b$, a contradiction.

Case (iii): $|u| = m + k + 1$. Then $u = a^k y\theta(b) = \theta(a^k)\theta(x)b$ which implies $a = \theta(a)$, a contradiction.

Since, all the three cases leads to a contradiction $(a^k y\theta(b))(a^k x\theta(b)) \in D_\theta(1)$.

Lemma 4.4.3 *Let θ be a morphic involution and let $a, b \in \Sigma$, $a \neq \theta(b)$. Let $x \neq y$, $x, y \in \Sigma^m$, $m > 0$. If $x = \theta(b)x'$, $x' \in \Sigma^*$ and $k \geq m$, then $(a^k y\theta(b))(\theta(a^k x\theta(b))) \in D_\theta(1)$.*

Proof Let $(a^k y\theta(b))(\theta(a^k x\theta(b))) \notin D_\theta(1)$. Then there exists $u \in \Sigma^+$ such that

$$u <_d^\theta (a^k y\theta(b))(\theta(a^k x\theta(b))).$$

By Lemma 4.3.2, it is enough to consider only the case $|u| \leq m + k + 1$.

Case (i): $|u| \leq k$. Then $u = a^n$ for some $n \leq k$ and $\theta(u) = \theta(\alpha'')b$ for $x = \alpha'\alpha'', \alpha' \in \Sigma^+, \alpha'' \in \Sigma^*$. Hence $a^n = \alpha''\theta(b)$ which implies $a = \theta(b)$, a contradiction.

Case (ii): $k < |u| < m + k + 1$. Then $u = a^k y'$ for $y = y'y''$, $y' \in \Sigma^+$, $y'' \in \Sigma^*$ and $\theta(u) = \theta(a^n)\theta(x)b = \theta(a^n)b\theta(x')b$ for $0 \leq n < k$. Hence $a^k y' = a^n\theta(b)x'\theta(b)$ which implies $a = \theta(b)$, a contradiction.

Case (iii): $|u| = m+k+1$. Then $u = a^k y\theta(b) = a^k x\theta(b)$ which implies $y = x$, a contradiction.

Since, all the three cases lead to a contradiction $(a^k y\theta(b))(\theta(a^k x\theta(b))) \in D_\theta(1)$.

Lemma 4.4.4 *Let θ be a literal (anti)morphism on Σ^* and $a, b \in \Sigma$ such that $a \neq \theta(b)$. Let $x \neq y$, $x, y \in \Sigma^m$, $m > 0$. Then:*

1. $a^m x\theta(b) \in D(1)$.

2. *If* $x = \theta(b)x'$, $x' \in \Sigma^*$ *and* $k \geq m$, *then* $(a^k y\theta(b))(a^k x\theta(b)) \in D(1)$.

Proof Let θ be a literal (anti)morphism.

1. Since there does not exist any word $u \in \Sigma^+$ with $|u| \leq m$ such that $u <_d a^m x\theta(b)$, by Lemma 4.3.1, $a^m x\theta(b) \in D(1)$.

2. Let $(a^k y\theta(b))(a^k x\theta(b)) \notin D(1)$. Then there exists $u \in \Sigma^+$ such that

$$u <_d (a^k y\theta(b))(a^k x\theta(b)).$$

   By Lemma 4.3.1, it is enough to consider only the case $|u| \leq m + k + 1$.

   Case (i): $|u| \leq k$. Then $u = a^n = \alpha''\theta(b)$ for some $n \leq k$ and $x = \alpha'\alpha'', \alpha' \in \Sigma^+, \alpha'' \in \Sigma^*$, which implies $a = \theta(b)$, a contradiction.

   Case (ii): $k < |u| < m + k + 1$. Then $u = a^k y' = a^n x\theta(b) = a^n\theta(b)x'\theta(b)$ for $y = y'y''$, $y' \in \Sigma^+$, $y'' \in \Sigma^*$ and $0 \leq n < k$, which implies $a = \theta(b)$, a contradiction.

   Case (iii): $|u| = m + k + 1$. Then $u = a^k y\theta(b) = a^k x\theta(b)$ which implies $x = y$, a contradiction.

   Since, all the three cases leads to a contradiction $(a^k y\theta(b))(a^k x\theta(b)) \in D(1)$.

Corollary 4.4.5 follows immediately from Lemma 4.4.2 and 4.4.4.

Corollary 4.4.5 *Let* θ *be a morphic involution on* Σ^*, *where* Σ *is an alphabet with* $|\Sigma| \geq 3$ *that contains letters* $a \neq b$ *such that* $a \notin \{\theta(b), \theta(a)\}$. *Let* $x \neq y$, $x, y \in \Sigma^m$, $m > 0$. *Then:*

1. $a^m x\theta(b) \in D_\theta(1) \cap D(1)$.

2. *If* $x = \theta(b)x'$, $x' \in \Sigma^*$ *and* $k \geq m$, *then* $(a^k y\theta(b))(a^k x\theta(b)) \in D_\theta(1) \cap D(1)$.

Lemma 4.4.6 *Let* θ *be a morphic involution and let* $a, b \in \Sigma$ *such that* $a \notin \{b, \theta(b)\}$. *Let* $x \in \Sigma^m$, $m > 0$. *If* $x = \theta(b)x'$, $x' \in \Sigma^*$, *then* $(a^m x\theta(b))(\theta(a^m x\theta(b))) \in D_\theta(2)$.

Proof Clearly $\lambda, a^m x\theta(b) \in L_d^\theta((a^m x\theta(b))(\theta(a^m x\theta(b))))$.

Let $(a^m x\theta(b))(\theta(a^m x\theta(b))) \notin D_\theta(2)$. Then there exists $u \in \Sigma^+$ such that

$$u <_d^\theta (a^m x\theta(b))(\theta(a^m x\theta(b)))$$

and $u \notin \{\lambda, a^m x\theta(b)\}$. Then, we have following cases to consider.

Case (i): $|u| \leq m$. Then, $u = a^n$ for some $n \leq m$ and $\theta(u) = \theta(\alpha'')b$ for $x = \alpha'\alpha''$, $\alpha' \in \Sigma^+$ and $\alpha'' \in \Sigma^*$. Hence $a^n = \alpha''\theta(b)$ which implies $a = \theta(b)$, a contradiction.

Case (ii): $m < |u| < 2m + 1$. Then, $u = a^m\alpha'$ for $x = \alpha'\alpha''$, $\alpha' \in \Sigma^+$, $\alpha'' \in \Sigma^*$ and $\theta(u) = \theta(a^n)\theta(x)b = \theta(a^n)b\theta(x')b$ for $0 \leq n < m$. Hence $a^m\alpha' = a^n\theta(b)x'\theta(b)$ which implies $a = \theta(b)$, a contradiction.

Case (iii): $2m + 1 < |u| \leq 3m + 1$. Then, $u = a^m x\theta(b)\theta(a^k)$ for some $0 < k \leq m$ and $\theta(u) = \alpha''\theta(b)\theta(a^m)\theta(x)b$ for $x = \alpha'\alpha''$, $\alpha' \in \Sigma^+, \alpha'' \in \Sigma^*$. Hence, $u = a^m x\theta(b)\theta(a^k) = \theta(\alpha'')ba^m x\theta(b)$ which implies $a = b$, a contradiction.

Case (iv): $3m + 1 < |u| \leq 4m + 1$. Then, $u = a^m x\theta(b)\theta(a^m)\theta(\alpha')$ for $x = \alpha'\alpha''$, $\alpha' \in \Sigma^+$, $\alpha'' \in \Sigma^*$ and $\theta(u) = a^k x\theta(b)\theta(a^m)\theta(x)b$ for $0 \leq k < m$. Hence, $u = a^m x\theta(b)\theta(a^m)\theta(\alpha') = \theta(a^k)b\theta(x')ba^m x\theta(b)$ which implies $a = b$, a contradiction.

Since all the cases leads to a contradiction $(a^m x\theta(b))(\theta(a^m x\theta(b))) \in D_\theta(2)$.

Theorem 4.4.7 *Let θ be a morphic involution on Σ^*, where Σ is an alphabet with $|\Sigma| \geq 2$ that contains letters $a \neq b$ such that $a \neq \theta(b)$. Then the set of θ-unbordered words, $D_\theta(1)$ and set of words with exactly two θ-borders $D_\theta(2)$ are disjunctive.*

Proof Let $x, y \in \Sigma^m$, $x \neq y$, $m > 0$. Without loss of generality let us assume that $x = \theta(b)x'$, $x' \in \Sigma^*$. Let $u = a^m$, $v = \theta(b)\theta(a^m x\theta(b))$. Since $a \neq b$, by Lemma 4.4.2(1), we have $a^m x\theta(b) \in D_\theta(1)$ and by Lemma 4.4.6,

$$uxv = a^m x\theta(b)\theta(a^m x\theta(b)) \in D_\theta(2).$$

Since $D_\theta(2) \cap D_\theta(1) = \emptyset$, it follows that $uxv \notin D_\theta(1)$. Further, by Lemma 4.3.18 $\theta(a^m x\theta(b)) \in D_\theta(1)$. Since $a \neq \theta(b)$, by Lemma 4.4.3,

$$uyv = a^m y\theta(b)(\theta(a^m x\theta(b))) \in D_\theta(1).$$

Since, for $x, y \in \Sigma^+$ $x \neq y$, $|x| = |y|$, we got $x \not\equiv y(P_L)$ where $L = D_\theta(1)$. Hence, by Proposition 4.4.1, we have that $D_\theta(1)$ is disjunctive. From the proof it follows that also $D_\theta(2)$ is disjunctive.

The following Lemmas are needed for the proof of Theorem 4.4.13.

Lemma 4.4.8 *Let $m \geq 1$, $x \in \Sigma^+$, $u', u'', y \in \Sigma^*$ and θ be a morphic involution on Σ^*. For any $u \in D_\theta(1) \cap D(1)$, if $(x_1y_1 \cdots x_my_m)x_{m+1} = u'uu''$, where $x_i = x$ and $y_j = y$ if i and j are odd, $x_i = \theta(x)$ and $y_j = \theta(y)$ if i and j are even for $1 \leq i \leq m+1$ and $1 \leq j \leq m$, then $|u| \leq |xy|$.*

Proof Suppose, $|u| > |xy|$. We will prove just 3 cases here, the other cases follow similarly.

Case (i): u occurs as a subword of $y\theta(x)\theta(y)$. Then there exists $\alpha_1, \alpha_2 \in \Sigma^+$ and $\beta_1, \beta_2, \beta_1', \beta_2' \in \Sigma^*$ such that $x = \alpha_1\alpha_2$, $y = \beta_1\beta_1' = \beta_2'\beta_2$, $|\beta_2| > |\beta_1'|$, then there exists $\alpha \in \Sigma^+$ such that $\beta_1 = \beta_2'\alpha$, $\beta_2 = \alpha\beta_1'$ and we have

$$u = \beta_2\theta(\alpha_1)\theta(\alpha_2)\theta(\beta_1) = \alpha\beta_1'\theta(\alpha_1\alpha_2)\theta(\beta_2')\theta(\alpha) \notin D_\theta(1)$$

Case (ii): u occurs as a subword of $y\theta(x)\theta(y)x$. Then there exists $\alpha_1, \alpha_2 \in \Sigma^+$ and $\beta_1, \beta_2 \in \Sigma^*$ such that $x = \alpha_1\alpha_2$, $y = \beta_1\beta_2$, then

$$u = \beta_2\theta(\alpha_1)\theta(\alpha_2)\theta(\beta_1)\theta(\beta_2)\alpha_1 \notin D_\theta(1)$$

a contradiction.

Case (iii): u occurs as a subword of $y\theta(x)\theta(y)xy\theta(x)$. Then $\alpha_1, \alpha_2 \in \Sigma^+$ and $\beta_1, \beta_2 \in \Sigma^*$ such

that $x = \alpha_1\alpha_2$, $y = \beta_1\beta_2$, then

$$u = \beta_2\theta(\alpha_1)\theta(\alpha_2)\theta(y)x\beta_1\beta_2\theta(\alpha_1) \notin D(1)$$

a contradiction.

All the other cases will lead to a similar contradiction, hence $|u| \leq |xy|$.

Lemma 4.4.9 *Let θ be a morphic involution on Σ^*. If $f_1 \cdots f_m = u_1u_2 \cdots u_k$ with $u_i \in D_\theta(1) \cap D(1)$, $i = 1, 2, \cdots, k$ such that $f_j = f$ if j is odd and $f_j = \theta(f)$ if j is even, $1 \leq j \leq m$, then $|u_i| \leq |f|$ for all $1 \leq i \leq k$.*

Proof Follows from the proof of Lemma 4.4.8 replacing y by an empty word λ.

Lemma 4.4.10 *Let $m \geq 2$, $m \geq n \geq 1$, θ be a morphic involution on Σ^*. Then for any $x \in \Sigma^+$, $y \in \Sigma^*$, $(x_1y_1 \cdots x_my_m)x_{m+1} \notin [D_\theta(1) \cap D(1)]^n$, where the conditions placed on x_i and y_j for $1 \leq i \leq m+1$ and $1 \leq j \leq m$ are the same as those in Lemma 4.4.8.*

Proof Suppose $(x_1y_1 \cdots x_my_m)x_{m+1} \in [D_\theta(1) \cap D(1)]^n$. Then there exists

$u_1, u_2, \cdots, u_n \in D_\theta(1) \cap D(1)$ such that $(x_1y_1 \cdots x_my_m)x_{m+1} = u_1u_2 \cdots u_n$. By Lemma 4.4.8, we will get $|u_i| \leq |xy|$ for $1 \leq i \leq n$. However, this would further imply,

$$|u_1u_2 \cdots u_n| \leq n|xy| \leq m|xy| < m|xy| + |x|$$

which is a contradiction. Hence $(x_1y_1 \cdots x_my_m)x_{m+1} \notin [D_\theta(1) \cap D(1)]^n$.

Lemma 4.4.11 *Let $m > n \geq 1$ and θ be a morphic involution on Σ^*. Then for any $f, \theta(f) \in \Sigma^+$, we have $f_1 \cdots f_m \notin [D_\theta(1) \cap D(1)]^n$, where the conditions placed on f_i for $1 \leq i \leq m$ are the same as those of Lemma 4.4.9.*

Proof Follows from the proof of Lemma 4.4.10 replacing y by an empty word λ.

Lemma 4.4.12 *Let θ be a morphic involution on Σ^*. For any $f, \theta(f) \in D_\theta(1) \cap D(1)$ and $n \geq 2$, $f_1 \cdots f_n \notin [D_\theta(1) \cap D(1)]^{n-1}$, where the conditions placed on f_i for $1 \leq i \leq n$ are the same as those of Lemma 4.4.9.*

Proof We will prove this result by induction on n. For $n = 2$ result holds trivially as $f\theta(f) \notin D_\theta(1) \cap D(1)$. Assume that the result holds for $n = k$, i.e., $f_1 \cdots f_k \notin [D_\theta(1) \cap D(1)]^{k-1}$. Suppose, $f_1 \cdots f_{k+1} \in [D_\theta(1) \cap D(1)]^k$, then there exists $u, v \in \Sigma^+$ such that $uv = f_1 \cdots f_{k+1}$, $u \in D_\theta(1) \cap D(1)$ and $v \in [D_\theta(1) \cap D(1)]^{k-1}$. By Lemma 4.4.9, $|u| \leq |f|$. If $|u| < |f|$, then $f = uu'$ for some $u' \in \Sigma^+$. Hence, we get

$$f_1 \cdots f_{k+1} = u_1 u_1' \cdots u_{k+1} u_{k+1}' = u_1 (u_1' u_2 \cdots u_k' u_{k+1}) u_{k+1}'$$

where $u_i u_i' = uu'$ if i is odd and $u_i u_i' = \theta(u)\theta(u')$ if i is even. But then $(u_1' u_2 \cdots u_k' u_{k+1}) u_{k+1}' \in [D_\theta(1) \cap D(1)]^{k-1}$ which is a contradiction to Lemma 4.4.10. If $|u| = |f|$, then $u = f$. Thus, $v = f_2 \cdots f_{k+1} \in [D_\theta(1) \cap D(1)]^{k-1}$, which is a contradiction to Lemma 4.4.11. Hence $f_1 \cdots f_n \notin [D_\theta(1) \cap D(1)]^{n-1}$.

Theorem 4.4.13 *Let θ be a morphic involution on Σ^*, where Σ is an alphabet with $|\Sigma| \geq 3$ that contains letters $a \neq b$ such that $a \notin \{\theta(b), \theta(a)\}$. Then the set $[D_\theta(1) \cap D(1)]^n$ is disjunctive for any even number $n \geq 2$.*

Proof Choose $x \neq y \in \Sigma^m$, $m > 0$ with $y = \theta(b)y'$ for some $y' \in \Sigma^*$. Let $L = [D_\theta(1) \cap D(1)]^n$. By Corollary 4.4.5(1), $a^m x\theta(b) \in D_\theta(1) \cap D(1)$ and thus by Lemma 4.3.17 and 4.3.18 $\theta(a^m x\theta(b)) \in D_\theta(1) \cap D(1)$. Since $x \neq y$ and $a \neq \theta(b)$, by Lemma 4.4.3 we have $a^m x\theta(b)\theta(a^m y\theta(b)) \in D_\theta(1) \cap D(1)$, which further by Lemma 4.3.17 and 4.3.18 implies $\theta(a^m x\theta(b))a^m y\theta(b) \in D_\theta(1) \cap D(1)$. Let

$$u = (u_1 \cdots u_n)a^m, v = \theta(b).$$

where $u_i = a^m x\theta(b)$ if i is odd and $u_i = \theta(a^m x\theta(b))$ if i is even.

Since n is even, we obtain

$$uyv = (u_1 \cdots u_n)a^m y\theta(b) = (u_1 \cdots u_{n-1})(\theta(a^m x\theta(b))a^m y\theta(b)) \in L.$$

On the other hand, by Lemma 4.4.12,

$$uxv = (u_1 \cdots u_n)a^m x\theta(b) = u_1 \cdots u_{n+1} \notin L.$$

Since, for $x, y \in \Sigma^+$, $x \neq y$, $|x| = |y|$, we got $x \not\equiv y(P_L)$, by Proposition 4.4.1, L is disjunctive.

In [11], it was shown that the language $D(i) \cap Q$ is disjunctive for $i \geq 1$. However, the following example shows that there exist morphic involutions θ for which the language $D_\theta(1) \cap Q_\theta$ is not disjunctive.

Example 4.4 *Let $\Sigma = \{A, C, G, T\}$ with θ being the morphic involution defined as $\theta(A) = T$, $\theta(T) = A$, $\theta(G) = C$ and $\theta(C) = G$. Let $u = ACT$, $v = CA$, $x = AGG$ and $y = TCA$. Then $uxv = ACTAGGCA \in D_\theta(1) \cap Q_\theta$ and $uyv = ACTTCACA \in D_\theta(1) \cap Q_\theta$, which shows that $D_\theta(1) \cap Q_\theta$ is not disjunctive.*

Proposition 4.4.14 *If θ is any literal antimorphism on Σ^*, $D_\theta(1)$ is a regular language.*

Proof We know that, for all $a \in \Sigma$, a is θ-unbordered and from Lemma 4.3.14, we have $D_\theta(1) = \Sigma \cup Y$ where $Y = \cup_{a,b\in\Sigma} a\Sigma^* b$ such that $\theta(a) \neq b$. Since Σ is finite, Y is regular and hence $D_\theta(1)$ is regular.

4.5 Conclusions

In this paper we investigate properties of θ-bordered words, where θ is not just the identity function or a morphic or antimorphic involution, but, more generally, a morphism or an antimorphism with the property that $\theta^n = I$, for $n \geq 2$, or a literal (anti)morphism θ. Results we

obtained include the transitivity of the relation $<_d^\theta$ for literal antimorphisms θ, and the disjunctivity of the set of all θ-unbordered words for morphic involutions θ.

Future directions of research includes exploring other properties of θ-bordered and θ-unbordered words, as well as the disjunctivity of other languages related to $D_\theta(i)$.

Bibliography

[1] A. Carpi and A. de Luca. Periodic-like words, periodicity, and boxes. *Acta Informatica*, 37(8):597–618, 2001.

[2] S. Constantinescu and L. Ilie. Fine and Wilf's theorem for abelian periods. *Bulletin of the EATCS*, 89:167–170, 2006.

[3] M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on Strings*. Cambridge University Press, 2007.

[4] M. Crochemore and W. Rytter. *Jewels of Stringology*. World Scientific, 2002.

[5] L. J. Cummings and W. F. Smyth. Weak repetitions in strings. *J. Combinatorial Mathematics and Combinatorial Computing*, 24:33–48, 1997.

[6] E. Czeizler, L. Kari, and S. Seki. On a special class of primitive words. *Theoretical Computer Science*, 411:617 – 630, 2010.

[7] A. de Luca and A. de Luca. Pseudopalindrome closure operators in free monoids. *Theoretical Computer Science*, 362(13):282 – 300, 2006.

[8] P. Gawrychowski, F. Manea, R. Mercaş, D. Nowotka, and C. Tiseanu. Finding pseudo-repetitions. *Leibniz International Proceedings in Informatics*, 20:257–268, 2013.

[9] P. Gawrychowski, F. Manea, and D. Nowotka. Discovering hidden repetitions in words. In P. Bonizzoni, V. Brattka, and B. Löwe, editors, *The Nature of Computation. Logic,*

Algorithms, Applications, volume 7921 of *Lecture Notes in Computer Science*, pages 210–219. Springer Berlin Heidelberg, 2013.

[10] P. Gawrychowski, F. Manea, and D. Nowotka. Testing generalised freeness of words. In E. W. Mayr and N. Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, volume 25, pages 337–349, 2014.

[11] S. Hsu, M. Ito, and H. Shyr. Some properties of overlapping order and related languages. *Soochow Journal of Mathematics*, 15(1):29–45, 1989.

[12] C. Huang, P.-C. Hsiao, and C. J. Liau. A note of involutively bordered words. *Journal of Information and Optimization Sciences*, 31(2):371–386, 2010.

[13] S. Hussini, L. Kari, and S. Konstantinidis. Coding properties of DNA languages. In N. Jonoska and N. Seeman, editors, *Proc. of DNA7*, volume 2340 of *Lecture Notes in Computer Science*, pages 57–69. Springer, 2002.

[14] L. Kari, S. Konstantinidis, and P. Sosík. Bond-free languages: Formalizations, maximality and construction methods. *International Journal of Foundations of Computer Science*, 16:1039–1070, 2005.

[15] L. Kari, E. Losseva, S. Konstantinidis, P. Sosík, and G. Thierrin. A formal language analysis of DNA hairpin structures. *Fundamenta Informaticae*, 71:453–475, Mar. 2006.

[16] L. Kari and K. Mahalingam. Involutively bordered words. *International Journal of Foundations of Computer Science*, 18(05):1089–1106, 2007.

[17] L. Kari and K. Mahalingam. Watson-crick bordered words and their syntactic monoid. *International Journal of Foundations of Computer Science*, 19(05):1163–1179, 2008.

[18] L. Kari and K. Mahalingam. Watson-Crick conjugate and commutative words. In M. H. Garzon and H. Yan, editors, *Proc. of DNA13*, volume 4848 of *Lecture Notes in Computer Science*, pages 273–283. Springer-Verlag, 2008.

[19] L. Kari and S. Seki. On pseudoknot-bordered words and their properties. *Journal of Computer and System Sciences*, 75:113 – 121, 2009.

[20] L. Kari and S. Seki. An improved bound for an extension of Fine and Wilf's theorem and its optimality. *Fundamenta Informaticae*, 101:215–236, 2010.

[21] G. Paun, G. Rozenberg, and T. Yokomori. Hairpin languages. *Int. J. Found. Comput. Sci.*, 12:837–847, 2001.

[22] H. J. Shyr. *Free Monoids and Languages*. Department of Mathematics, Soochow University, Taipei, Taiwan, 1979.

[23] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

Chapter 5

Disjunctivity and Other Properties of Sets of Pseudo-Bordered Words

5.1 Introduction

Combinatorics on words, coding theory, and formal language theory have had a wide range of applications ranging from bioinformatics, to cryptography, to DNA computing. For example, the concepts of periodicity and primitivity are at the root of pattern-matching and data compression algorithms, [5, 6, 33], and the study of codes is essential in determining the unique decipherability of encoded messages, [28]. Notably, the recent connection with DNA computing has motivated a new line of study wherein classical concepts are generalized to ones where the identity function is replaced with more general pseudo-identity functions. A representative example of such a generalization is the concept of antimorphic involution which models the DNA Watson-Crick complementarity, as described below.

DNA single strands can be viewed as strings over the DNA alphabet $\{A, C, G, T\}$. The Watson-Crick complementarity is the property whereby two DNA *single* strands of opposite orientation and with complementary "letters" at each position can bind together by hydro-

[0]A version of this chapter has been submitted for publication (*Acta Informatica)*

gen bonds to form a DNA *double* strand with its well-known double helical structure [29]. Given an alphabet Σ, an antimorphic involution θ is a function that is an antimorphism, that is, $\theta(uv) = \theta(v)\theta(u)$, $\forall u, v \in \Sigma^*$, and an involution, that is, $\theta(\theta(x)) = x$, $\forall x \in \Sigma^*$. Thus, the first property (antimorphism) models the fact that DNA single strands that bind to each other must have opposite orientations, and the second property (involution) models the letter-to-letter complementarity of the two single strands (whereby A binds to a T, and C binds to a G) that is necessary for the binding to occur.

Note that a DNA single strand and its Watson-Crick complement are informationally equivalent, since one uniquely determines the other and viceversa. Thus, a DNA strand and its Watson-Crick complement can be viewed in a sense as "identical", and this motivated the idea of generalizing the notion of identity function to pseudo-identity functions, such as antimorphic involutions. Some of the new concepts in combinatorics on words and coding theory that were thus obtained are: Pseudo-periodicity, [8, 23], pseudo-commutativity, pseudo-conjugacy, [20], pseudo-palindrome, [21, 9], involution codes, [3, 16, 17], etc. Some of these concepts were further generalized in [10, 11, 12] by replacing the morphic involution with length-preserving, erasing and uniform morphism functions. Also, independently, the notion of periodicity was extended to periodic-like words, [2], weakly periodic words, [7], also known as Abelian periodic words, [4], and pseudoperiodic words, [1].

A non-empty word w is said to be bordered if there exists a word that is a proper prefix and a proper suffix of w. A word which is not bordered is called unbordered. In [19] the notion bordered word was generalized to that of a θ-bordered word (also called pseudo-bordered word), where θ is (anti)morphic involution: A word w is said to be θ-bordered if there exists a word $v \in \Sigma^+$ that is a proper prefix of w, while $\theta(v)$ is a proper suffix of w. Naturally, a word which is not θ-bordered is θ-unbordered. Properties of θ-bordered and θ-unbordered words were explored in, e.g., [15, 19]. The classical notions of bordered and unbordered words have also been generalized to pseudo-knot-bordered words in [22], where a non-empty word w is said to be pseudo-knot-bordered if $w = xy\alpha = \beta\theta(yx)$ for $\alpha, \beta, x, y \in \Sigma^+$.

In this paper we continue to explore the properties of θ-bordered and θ-unbordered words, for *morphic involutions* θ. The main focus is on disjunctivity properties of sets of θ-bordered words and some other related languages. The paper is organized as follows. Section 5.2 includes basics definitions and notions used throughout the paper. In Section 5.3 we prove, e.g., that under some conditions, the set of all θ-bordered words with exactly i θ-borders, $D_\theta(i)$, is disjunctive for all $i \geq 1$ (Theorem 5.3.8). In Section 5.4 and 5.5, we discuss relationships between and among the sets $D_\theta(1)$, the set of all θ-unbordered words, and the set $D(i)$, of all bordered words with exactly i borders. In particular, we show that, under some conditions, the set $D^i_\theta(1) \setminus D(i)$ is disjunctive for all $i \geq 2$ (Theorem 5.4.4). In Section 5.6 we discuss some conditions for catenations of languages of θ-unbordered words to remain θ-unbordered, and offer a preview of further generalizations of these results by proving that the set of all θ-bordered words is not context-free for all morphisms θ over an alphabet Σ with $|\Sigma| \geq 3$ such that $\theta(a) \neq a$ for all $a \in \Sigma$ and θ^3 equals the identity function on Σ.

5.2 Basic definitions and notations

An alphabet Σ is a finite non-empty set of symbols. Σ^* denotes the set of all words over Σ, including the empty word λ. Σ^+ is the set of all non-empty words over Σ. The length of a word $u \in \Sigma^*$ (i.e. the number of symbols in a word) is denoted by $|u|$. By Σ^m we denote the set of all words of length $m > 0$ over Σ. The complement of a language $L \subseteq \Sigma^*$ is $L^c = \Sigma^* \setminus L$. For a language $L \subseteq \Sigma^*$ and $i \geq 2$, let $L^{(i)} = \{u^i | u \in L\}$ and $L^1 = L$ and $L^n = L^{n-1}L$ for $n \geq 2$. A word is called *primitive* if it cannot be expressed as a power of another word. Let Q denote the set of all primitive words. A function $\theta : \Sigma^* \to \Sigma^*$ is said to be a *morphism* if for all words $u, v \in \Sigma^*$ we have that $\theta(uv) = \theta(u)\theta(v)$, an *antimorphism* if $\theta(uv) = \theta(v)\theta(u)$, and an *involution* if θ^2 is an identity on Σ^*. If for all $a \in \Sigma$, $|\theta(a)| = 1$, then θ is called *literal* (anti)morphism[1]. A *θ-power* of a word u, [8] is a word of the form $u_1 u_2 \dots u_n$ for $n \geq 1$ where $u_1 = u$ and $u_i \in \{u, \theta(u)\}$ for

[1] By (anti)morphism we mean either a morphism or an antimorphism.

$2 \leq i \leq n$. A word is called θ-*primitive*, [8], if it cannot be expressed as a θ-power of another word. Let Q_θ denote the set of all θ-primitive words. For (anti)morphic involution θ, a word $u \in \Sigma^*$ is called a θ-*palindrome*, [21, 9], if $u = \theta(u)$. Let P_θ denote the set of all θ-palindromes.

For a language $L \subseteq \Sigma^*$, the *principal congruence* P_L determined by L is defined as follows: for any $x, y \in \Sigma^*$ such that $x \neq y$, $x \equiv y(P_L)$ if and only if $uxv \in L \Leftrightarrow uyv \in L$ for all $u, v \in \Sigma^*$. The index of P_L is the number of equivalence classes of P_L. L is said to be *disjunctive* if P_L is the identity, i.e., for any $x \neq y \in \Sigma^*$ there exists $u, v \in \Sigma^*$ such that $uxv \in L$ and $uyv \notin L$ or viceversa. A language $L \subseteq \Sigma^*$ is said to be *dense* if for all $u \in \Sigma^*$, $L \cap \Sigma^* u \Sigma^* \neq \emptyset$. Every disjunctive language is dense and every dense language contains a disjunctive subset, [27].

Definition 5.1 *1. For $v, w \in \Sigma^*$, w is a prefix of v ($w \leq_p v$) iff $v \in w\Sigma^*$.*

2. For $v, w \in \Sigma^*$, w is a suffix of v ($w \leq_s v$) iff $v \in \Sigma^* w$.

3. $\leq_d = \leq_p \cap \leq_s$.

4. For $u \in \Sigma^*$, $v \in \Sigma^*$ is said to be a border of u if $v \leq_d u$, i.e., $u = vx = yv$.

5. For $v, w \in \Sigma^*$, w is a proper prefix of v ($w <_p v$) iff $v \in w\Sigma^+$.

6. For $v, w \in \Sigma^*$, w is a proper suffix of v ($w <_s v$) iff $v \in \Sigma^+ w$.

7. $<_d = <_p \cap <_s$.

8. For $u \in \Sigma^*$, $v \in \Sigma^*$ is said to be a proper border of u if $v <_d u$.

9. For $u \in \Sigma^+$, denote by $L_d(u) = \{v \in \Sigma^* | v <_d u\}$, the set of all borders of a word $u \in \Sigma^*$.

10. $\nu_d(u) = |L_d(u)|$.

11. Denote by $D(i) = \{u \in \Sigma^+ | \nu_d(u) = i\}$, the set of all words with exactly i borders for $i \geq 1$.

12. A word $u \in \Sigma^+$ is said to be a bordered word if there exists $v \in \Sigma^+$ such that $v <_d u$, i.e., $u = vx = yv$ for some $x, y \in \Sigma^+$.

13. *A non-empty word which is not bordered is called unbordered. Thus, $D(1)$ is the set of all unbordered words over Σ.*

For a word w, $\text{Pref}(w) = \{u \in \Sigma^+ | \exists v \in \Sigma^*, w = uv\}$ and $\text{Suff}(w) = \{u \in \Sigma^+ | \exists v \in \Sigma^*, w = vu\}$ denotes the set of all prefixes and suffixes respectively. Similarly, the set of all proper prefixes and proper suffixes of a word w can be defined as $\text{PPref}(w) = \{u \in \Sigma^+ | \exists v \in \Sigma^+, w = uv\}$ and $\text{PSuff}(w) = \{u \in \Sigma^+ | \exists v \in \Sigma^+, w = vu\}$ respectively. For further notions in formal language theory and combinatorics on words the reader is referred to [13, 25, 27, 32].

The following definitions extend the notion of bordered and unbordered words to θ-bordered and θ-unbordered words and for any (anti)morphism on Σ^*.

Definition 5.2 *[19] Let θ be either a morphism or an antimorphism on Σ^*.*

1. *For $v, w \in \Sigma^*$, w is a θ-prefix of v ($w \leq_p^\theta v$) iff $v \in \theta(w)\Sigma^*$.*

2. *For $v, w \in \Sigma^*$, w is a θ-suffix of v ($w \leq_s^\theta v$) iff $v \in \Sigma^*\theta(w)$.*

3. $\leq_d^\theta = \leq_p \cap \leq_s^\theta$.

4. *For $u \in \Sigma^*$, $v \in \Sigma^*$ is said to be a θ-border of u if $v \leq_d^\theta u$, i.e., $u = vx = y\theta(v)$.*

5. *For $w, v \in \Sigma^*$, w is a proper θ-prefix of v ($w <_p^\theta v$) iff $v \in \theta(w)\Sigma^+$.*

6. *For $w, v \in \Sigma^*$, w is a proper θ-suffix of v ($w <_s^\theta v$) iff $v \in \Sigma^+\theta(w)$.*

7. $<_d^\theta = <_p \cap <_s^\theta$.

8. *For $u \in \Sigma^*$, $v \in \Sigma^*$ is said to be a proper θ-border of u if $v <_d^\theta u$.*

9. *For $u \in \Sigma^+$, define by $L_d^\theta(u) = \{v \in \Sigma^* | v <_d^\theta u\}$, the set of all θ-borders of a word $u \in \Sigma^*$.*

10. $\nu_d^\theta(u) = |L_d^\theta(u)|$.

11. *Denote by $D_\theta(i) = \{u \in \Sigma^+ | \nu_d^\theta(u) = i\}$, the set of all words with exactly i θ-borders for $i \geq 1$.*

12. *A word $u \in \Sigma^+$ is said to be θ-bordered if there exists $v \in \Sigma^+$ such that $v <_d^\theta u$, i.e., $u = vx = y\theta(v)$ for some $x, y \in \Sigma^+$.*

13. *A nonempty word which is not θ-bordered is called θ-unbordered. Thus, $D_\theta(1)$ is the set of all θ-unbordered words over Σ.*

Recall that every disjunctive language has infinitely many principle congruence classes whereas the number of principle congruence classes for regular language is finite. Hence, it is clear that disjunctive languages are not regular.

The following proposition provides a necessary and sufficient condition for a language to be disjunctive, and will be used throughout this paper.

Proposition 5.2.1 *[27] Let $L \subseteq \Sigma^*$. Then the following two statements are equivalent:*

1. *L is a disjunctive language.*

2. *If $u, v \in \Sigma^+$, $u \neq v$, $|u| = |v|$, then $u \not\equiv v(P_L)$.*

While proving disjunctivity or any other properties of the sets of words with exactly i borders or i θ-borders, $D(i)$ or $D_\theta(i)$ respectively, one of the important tools is the knowledge about the number of borders and θ-borders of a word. Proposition 5.2.2 characterizes the number of borders of a power of a primitive word.

Proposition 5.2.2 *[14] For any $f \in Q$ and $j \geq 1$, $\nu_d(f^j) = \nu_d(f) + j - 1$.*

Similarly, Lemma 5.2.3 provides a characterization for the number of θ-borders of a θ-palindrome, for morphic involutions.

Lemma 5.2.3 *[19] Let u be a θ-palindromic primitive word and j be an integer, $j > 1$. Then, for a morphic involution θ, $\nu_d^\theta(u^j) = \nu_d^\theta(u) + j - 1$.*

The following lemma provides a sufficient condition for a word to be bordered.

Lemma 5.2.4 *[14] Let $u \in \Sigma^+ \backslash D(1)$. Then there exists $v \in \Sigma^*$ with $|v| \leq \frac{|u|}{2}$ such that $v <_d u$.*

By the definition of an unbordered word, it is clear that the set of all unbordered words $D(1)$ is a subset of set of all primitive words Q, i.e., $D(1) \subseteq Q$. A similar inclusion does not hold in the case of set of all θ-unbordered words $D_\theta(1)$ and the set of all θ-primitive words Q_θ for a morphic involution θ, as demonstrated by following example. The example also demonstrates the fact that Q_θ is not a subset of $D_\theta(1)$.

Example 5.1 *Let $\Sigma = \{a, b, c\}$, θ be a morphic involution such that $\theta(a) = b$, $\theta(b) = a$ and $\theta(c) = c$. Let $u = abaa$, then $u \in D_\theta(1)$ but $u = abaa = a\theta(a)aa \notin Q_\theta$ and hence $D_\theta(1) \nsubseteq Q_\theta$. Now, let $v = acb$, then $u \in Q_\theta$ but $u = acb = ac\theta(a) \notin D_\theta(1)$ and hence $Q_\theta \nsubseteq D_\theta(1)$.*

However, for a morphic involution θ, the set $D_\theta(1) \cap Q_\theta \neq \emptyset$. For example, if $\Sigma = \{a, b, c\}$ such that $\theta(a) = b$, $\theta(b) = a$ and $\theta(c) = c$, then $abc \in D_\theta(1) \cap Q_\theta$. Moreover, the set $D_\theta(i) \cap Q_\theta \neq \emptyset$ for all $i \geq 1$.

5.3 Disjunctivity properties of $D_\theta(i)$

In [14] it was shown that the languages $D(i)$, $D(i) \cap Q$ and $D(i) \cap Q^{(j)}$ are disjunctive for $i \geq j \geq 1$. In this section, we will prove the disjunctivity of the set $D_\theta(i)$ for all $i \geq 1$ (Theorem 5.3.8). Also, we know from Example 5.1 that neither $D_\theta(1) \subseteq Q_\theta$ nor $Q_\theta \subseteq D_\theta(1)$ but $D_\theta(i) \cap Q_\theta \neq \emptyset$ for all $i \geq 1$. Furthermore, in this section we will prove that the set $D_\theta(i) \cap Q_\theta^{2i-2}$ is disjunctive for $i \geq 2$ (Corollary 5.3.9).

In the previous section, we have seen a sufficient condition for a word to be bordered. The following lemma provides a sufficient condition for a word to be θ-bordered in the case when θ is a morphic involution.

Lemma 5.3.1 *[18] Let θ be a morphic involution and let $u \in \Sigma^+ \backslash D_\theta(1)$. Then there exists $v \in \Sigma^*$ with $|v| \leq \frac{|u|}{2}$ such that $v <_d^\theta u$.*

Theorem 5.3.2 and 5.3.3 are mentioned for completeness.

Theorem 5.3.2 *[18] Let θ be a morphic involution on Σ^*, where Σ is an alphabet with $|\Sigma| \geq 2$ that contains letters $a \neq b$ such that $a \neq \theta(b)$. Then the set of all θ-unbordered words, $D_\theta(1)$ and set of words with exactly two θ-borders $D_\theta(2)$ are disjunctive.*

Theorem 5.3.3 *[18] Let θ be a morphic involution on Σ^*, where Σ is an alphabet with $|\Sigma| \geq 3$ that contains letters $a \neq b$ such that $a \notin \{\theta(b), \theta(a)\}$. Then the set $[D_\theta(1) \cap D(1)]^n$ is disjunctive for any even number $n \geq 2$.*

While Theorem 5.3.2 proves the disjunctivity of the set $D_\theta(i)$ for the cases $i = 1, 2$, we will prove (Theorem 5.3.8) that the set $D_\theta(i)$ is disjunctive for all $i \geq 3$ as well.

We first need several auxiliary results. In the previous section, we mentioned a characterization of the number of borders of a power of a primitive word. Now, we will provide a characterization of the number of θ-borders of a θ-power of a θ-unbordered word for morphic involution θ (Proposition 5.3.5). Note that here we consider a special case of a θ-power of a word $w = u_1u_2 \dots u_n$, where $u_i = u$ when i is odd and $u_i = \theta(u)$ when i is even for $1 \leq i \leq n$. The following lemma is needed for the proof of Proposition 5.3.5.

Lemma 5.3.4 *Let θ be morphic involution such that $\theta(a) \neq a$ for all $a \in \Sigma$. If $u \in D_\theta(1)$, then for $w = (u\theta(u))^k$, $u' <_p u$ we have that $(u\theta(u))^j u', (u\theta(u))^j u\theta(u') \notin L_d^\theta(w)$ for all $k > j \geq 1$.*

Proof We will prove the result by contradiction. Let $k > j \geq 1$ and $u' <_p u$.

First, assume that $(u\theta(u))^j u' <_d^\theta w$. Then, there exists $\alpha, \beta \in \Sigma^+$ such that $w = (u\theta(u))^k = (u\theta(u))^j u'\alpha = \beta(\theta(u)u)^j\theta(u')$. Since $|u'| < |u|$, we have that $\theta(u') <_s \theta(u)$ which implies $\theta(u) = u''\theta(u')$ for $u'' \in \Sigma^+$. This implies that $\theta(u'') <_p u$ since $u'' <_p \theta(u)$. But then, $(u\theta(u))^k = (u\theta(u))^{k-1}uu''\theta(u') = \beta(\theta(u)u)^{j-1}\theta(u)u\theta(u')$ which implies $u'' <_s u$ since $|u''| < |u|$ which further implies that $\theta(u'') <_d^\theta u$, i.e., $u \notin D_\theta(1)$, a contradiction. Hence, $(u\theta(u))^j u' \notin L_d^\theta(w)$.

Now, let $(u\theta(u))^j u\theta(u') <_d^\theta w$. Then there exists $\alpha', \beta' \in \Sigma^+$ such that $w = (u\theta(u))^k = (u\theta(u))^j u\theta(u')\alpha' = \beta'(\theta(u)u)^j\theta(u)u'$. Since $|u'| < |u|$, which implies $u' <_s \theta(u)$, i.e., $\theta(u') <_s u$ which further implies $u' <_d^\theta u$ and hence $u \notin D_\theta(1)$, a contradiction. □

Proposition 5.3.5 *Let θ be morphic involution such that $\theta(a) \neq a$ for all $a \in \Sigma$. If $u \in D_\theta(1)$, then $w = (u\theta(u))^n \in D_\theta(n+1)$ for all $n \geq 1$.*

Proof We will prove this statement by induction on n.

Let $n = 1$, then for $w = u\theta(u)$, since $u \in D_\theta(1)$, $L_d^\theta(w) = \{\lambda, u\}$. Hence $w = u\theta(u) \in D_\theta(2)$.

Let $n = 2$, then for $w = u\theta(u)u\theta(u)$, by Lemma 5.3.4 $u\theta(u)u', u\theta(u)u\theta(u') \notin L_d^\theta(w)$ where $u' \in \mathrm{PPref}(u)$ and hence $L_d^\theta(w) = \{\lambda, u, u\theta(u)u\}$. Thus $w \in D_\theta(3)$.

Let us assume that the result holds for $n = k$, i.e., $w = (u\theta(u))^k \in D_\theta(k+1)$.

Now, we will prove that the result holds for $n = k+1$. We have $w = (u\theta(u))^{k+1} = (u\theta(u))^k u\theta(u)$. By inductive hypothesis, we know that $(u\theta(u))^k \in D_\theta(k+1)$. Also, by Lemma 5.3.4, $(u\theta(u))^k u', (u\theta(u))^k u\theta(u') \notin L_d^\theta(w)$ for some $u' <_p u$. Thus, $L_d^\theta(w) = L_d^\theta((u\theta(u))^k) \cup \{(u\theta(u))^k u\}$ and hence $w \in D_\theta(k+2)$.

Hence, $w = (u\theta(u))^n \in D_\theta(n+1)$ for all $n \geq 1$. □

In the preceding two results we considered a special case of θ-powers, namely, words w consisting of alternations of u and $\theta(u)$. Under certain conditions, if in such words the first occurrence of u is replaced by $v \neq u$, then the word w becomes θ-unbordered, as showed by the following result.

Lemma 5.3.6 *Let θ be a morphic involution on Σ^*, where Σ is an alphabet with $|\Sigma| > 2$ that contains letters $a \neq b$ such that $a \neq \theta(b)$. Let $x \neq y$, $x, y \in \Sigma^m$, $m > 0$, $x, y \in \theta(b)\Sigma^*$. Then, for all $i \geq 2$,*

$$a^m y\theta(b)(\theta(a^m x\theta(b))a^m x\theta(b))^{i-2}\theta(a^m x\theta(b)) \in D_\theta(1).$$

Proof Let us assume that

$$w = a^m y\theta(b)(\theta(a^m x\theta(b))a^m x\theta(b))^{i-2}\theta(a^m x\theta(b)) \notin D_\theta(1).$$

Then there exists $v \in \Sigma^+$ such that $v <_d^\theta w$, i.e., $w = v\alpha = \beta\theta(v)$ for some $\alpha, \beta \in \Sigma^+$. Let $w = w'\theta(a^m x\theta(b))$ where $w' = a^m y\theta(b)(\theta(a^m x\theta(b))a^m x\theta(b))^{i-2}$. Then, by Lemma 5.3.1, it is enough to consider only the cases when $1 \leq |v| < (2m+1)(i-1)$.

Case 1: $v = a^k$ for $1 \leq k \leq m$. Then,

$$w = w'\theta(a^m x\theta(b)) = \beta\theta(a^k)$$

which implies $\theta(a) = b$, a contradiction.

Case 2: $v = a^m y'$ for $y = y'y''$ where $y' \in \Sigma^+$ and $y'' \in \Sigma^*$. Then,

$$w = w'\theta(a^m x\theta(b)) = \beta\theta(a^m y').$$

Now, since $|a^m y'| < |a^m x\theta(b)|$, $\theta(a^m y') <_s \theta(a^m x\theta(b))$, i.e., $\theta(a^m x\theta(b)) = \theta(a^m)b\theta(x')b = \beta'\theta(a^m y')$ for $x = \theta(b)x'$ where $x' \in \Sigma^*$. This implies $\theta(a) = b$, a contradiction.

Case 3: $v = a^m y\theta(b)$. Then,

$$w = w'\theta(a^m x\theta(b)) = \beta\theta(a^m y\theta(b))$$

which implies $x = y$, a contradiction.

Case 4: $v = a^m y\theta(b)\theta(a^k)$ for $1 \leq k \leq m$. Then,

$$w = w'\theta(a^m x\theta(b)) = \beta\theta(a^m y\theta(b))a^k$$

which implies $a = b$, a contradiction.

Case 5: $v = a^m y\theta(b)\theta(a^m x_1)$ for $x = x_1x_2$ where $x_1 \in \Sigma^+$ and $x_2 \in \Sigma^*$. Then,

$$w = w'\theta(a^m x\theta(b)) = \beta\theta(a^m y\theta(b))a^m x_1.$$

Now, since $2m \geq |a^m x_1| \geq |\theta(x)b| = m + 1$, $\theta(x)b \leq_s a^m x_1$, i.e., $a^m x_1 = \alpha'\theta(x)b = \alpha' b\theta(x')b$ with $|\alpha'| < m$ and $x = \theta(b)x'$ for $x' \in \Sigma^*$. This implies $a = b$, a contradiction.

Case 6: $v = a^m y\theta(b)\theta(a^m x\theta(b))$. Then,

$$w = w'\theta(a^m x\theta(b)) = \beta\theta(a^m y\theta(b))a^m x\theta(b)$$

which implies $\theta(a) = a$, a contradiction.

Case 7: $v = a^m y\theta(b)\theta(a^m x\theta(b))a^k$ for $1 \leq k \leq m$. Then,

$$w = w'\theta(a^m x\theta(b)) = \beta\theta(a^m y\theta(b))a^m x\theta(b)\theta(a^k).$$

which implies $\theta(a) = b$, a contradiction

Case 8: $v = a^m y\theta(b)\theta(a^m x\theta(b))a^m x'_1$ for $x = x'_1x'_2$ where $x'_1 \in \Sigma^+$ and $x'_2 \in \Sigma^*$. Then,

$$w = w'\theta(a^m x\theta(b)) = \beta\theta(a^m y\theta(b))a^m x\theta(b)\theta(a^m x'_1).$$

Now, since $|ba^m x'_1| \leq |a^m x\theta(b)| = 2m+1$, $\theta(ba^m x'_1) \leq_s \theta(a^m x\theta(b))$, i.e., $\theta(a^m x\theta(b)) = \alpha_2\theta(ba^m x'_1)$ where $|\alpha_2| < m$. This implies $\theta(a) = \theta(b)$, i.e., $a = b$, a contradiction.

Case 9: $v = a^m y\theta(b)(\theta(a^m x\theta(b))a^m x\theta(b))^k$ where $1 \leq k < \frac{i-2}{2}$. Then,

$$\begin{aligned} w &= a^m y\theta(b)(\theta(a^m x\theta(b))a^m x\theta(b))^{i-2}\theta(a^m x\theta(b)) \\ &= a^m y\theta(b)(\theta(a^m x\theta(b))a^m x\theta(b))^{i-2-k}\theta(a^m x\theta(b))(a^m x\theta(b)\theta(a^m x\theta(b)))^k \\ &= \beta\theta(a^m y\theta(b))(a^m x\theta(b)\theta(a^m x\theta(b)))^k. \end{aligned}$$

which implies $\theta(x) = \theta(y)$, i.e., $x = y$, a contradiction.

Since all the cases lead to a contradiction, $w \in D_\theta(1)$. □

The next lemma is used for proving the main result of this section.

Lemma 5.3.7 *[18] Let θ be a morphic involution and $a, b \in \Sigma$, $a \neq b$. Let $x, y \in \Sigma^m$, $m > 0$. Then*

1. $a^m x\theta(b) \in D_\theta(1)$.

2. *If* $a \neq \theta(a)$, $x = \theta(b)x'$, $x' \in \Sigma^*$ *and* $k > m$, *then* $(a^k y\theta(b))(a^k x\theta(b)) \in D_\theta(1)$.

Now, we will prove the main result of the section which shows that, under certain conditions, the set of words with exactly i θ-borders, $D_\theta(i)$, is disjunctive for all $i \geq 1$ and morphic involutions θ.

Theorem 5.3.8 *Let* θ *be a morphic involution on* Σ^*, *where* Σ *is an alphabet with* $|\Sigma| > 2$ *that contains letters* $a \neq b$ *such that* $a \neq \theta(b)$ *and* $\theta(a) \neq a$ *for all* $a \in \Sigma$. *Then the set of all* θ*-bordered words with exactly* i θ*-borders,* $D_\theta(i)$, *is disjunctive for all* $i \geq 1$.

Proof By Theorem 5.3.2, $D_\theta(i)$ is disjunctive for $i = 1, 2$. Now, we will prove the result for $i \geq 3$. Let $x, y \in \Sigma^n$, $x \neq y$, $m = n + 1$, $n > 0$. Let $u = a^m\theta(b)$,

$$v = \theta(b)(\theta(a^m\theta(b)x\theta(b))a^m\theta(b)x\theta(b))^{i-2}\theta(a^m\theta(b)x\theta(b))$$

. Since $a \neq b$, by Lemma 5.3.7, we have $a^m\theta(b)x\theta(b) \in D_\theta(1)$ and by Proposition 5.3.5,

$$uxv = [a^m\theta(b)x\theta(b)\theta(a^m\theta(b)x\theta(b))]^{i-1} \in D_\theta(i).$$

Further by Lemma 5.3.6,

$$uyv = a^m\theta(b)y\theta(b)[\theta(a^m\theta(b)x\theta(b))a^m\theta(b)x\theta(b)]^{i-2}\theta(a^m\theta(b)x\theta(b)) \in D_\theta(1).$$

Therefore, $x \not\equiv y(P_{D_\theta(i)})$ for every $x, y \in \Sigma^+$, $x \neq y$, $|x| = |y|$ and $i \geq 3$. Hence, by Proposition 5.2.1, $D_\theta(i)$ is disjunctive for $i \geq 1$. □

Let $\{a, b\} \subseteq \Sigma$ be such that $a \notin \{b, \theta(b)\}$ and θ be a morphic involution. Then for $x \in \Sigma^n$, $n > 0$ and $m = n + 1$, it is clear that $a^m\theta(b)x\theta(b), \theta(a^m\theta(b)x\theta(b)) \in Q_\theta$. Thus, we have following result as a consequence of Theorem 5.3.8.

Corollary 5.3.9 *Let θ be a morphic involution on Σ^*, where Σ is an alphabet with $|\Sigma| > 2$ that contains letters $a \neq b$ such that $a \neq \theta(b)$ and $\theta(a) \neq a$ for all $a \in \Sigma$. Then the set $D_\theta(i) \cap Q_\theta^{2i-2}$ is disjunctive for all $i \geq 2$.*

5.4 Disjunctivity of the set $D_\theta^i(1)\backslash D(i)$

Let us consider the relationship between the set of all words with exactly i θ-borders, $D_\theta(i)$, and the set of all words with exactly i borders, $D(i)$, for $i \geq 1$, an alphabet Σ, and morphic involutions θ such that $\theta(a) \neq a$ for all $a \in \Sigma$. It is clear that in general neither $D_\theta(i) \subseteq D(i)$ nor $D(i) \subseteq D_\theta(i)$. However, the set $D_\theta(i) \cap D(i) \neq \emptyset$ for all $i \geq 1$. For example, if $\Sigma = \{a, b, c\}$ such that $\theta(a) = b$, $\theta(b) = a$ and $\theta(c) = c$, then $abc \in D_\theta(1) \cap D(1)$ and $abba \in D_\theta(2) \cap D(2)$. Moreover, Theorem 5.3.3 proved that the set $(D_\theta(1) \cap D(1))^n$ is disjunctive for any even number $n \geq 2$. In this section, we will show that, under certain conditions, the set $D_\theta^i(1)\backslash D(i)$ is disjunctive for $i \geq 2$ (Theorem 5.4.4).

In order to show that the language $D_\theta^i(1)\backslash D(i)$ is disjunctive, we need to characterize some catenations of unbordered and θ-unbordered words. The following proposition shows such a relationship.

Proposition 5.4.1 *[31] Let $\{a, b\} \in \Sigma$ and let $x, y \in b\Sigma^*$ with $x \neq y$. If $|x| = |y|$ or $|x| > |y|$ and $x \in ya\Sigma^*$, then for $k \geq |x| \geq |y|$, $(a^k xb)^i (a^k yb)^j \in D(1)$ and $(a^k yb)^j (a^k xb)^i \in D(1)$ for all $i, j \geq 1$.*

Similarly, in Proposition 5.4.2 we show the relationship between some catenations of powers of two θ-unbordered words and the set of all θ-unbordered words.

Proposition 5.4.2 *Let θ be a morphic involution on Σ^* and let $a, b \in \Sigma$ such that $a \notin \{\theta(a), b\}$. Let $x, y \in \theta(b)\Sigma^*$ with $x \neq y$. Then for all $k > |x| \geq |y|$, $i, j \geq 1$, $(a^k x\theta(b))^i (a^k y\theta(b))^j \in D_\theta(1)$ and $(a^k y\theta(b))^j (a^k x\theta(b))^i \in D_\theta(1)$.*

Proof To prove the result, we will use Lemma 11 of [19] which states that $\theta(\text{Pref}(u)) \cap \text{Suff}(v) = \emptyset$ and the set of all words in u^+v^+ are θ-unbordered are equivalent statements. Hence we need

to show that,

$$\theta(\text{Pref}(a^k y\theta(b))) \cap \text{Suff}(a^k x\theta(b)) = \emptyset \text{ and } \theta(\text{Pref}(a^k x\theta(b))) \cap \text{Suff}(a^k y\theta(b)) = \emptyset.$$

First, let $|x| = |y|$. Then, from Lemma 5.3.7 and since $x, y \in \theta(b)\Sigma^*$, we have that, $(a^k y\theta(b))(a^k x\theta(b)) \in D_\theta(1)$ and $(a^k x\theta(b))(a^k y\theta(b)) \in D_\theta(1)$. Therefore, if $|x| = |y|$, then $\theta(\text{Pref}(a^k x\theta(b))) \cap \text{Suff}(a^k y\theta(b)) = \emptyset$ and $\theta(\text{Pref}(a^k y\theta(b))) \cap \text{Suff}(a^k x\theta(b)) = \emptyset$.

Now, let $|x| > |y|$. We will only prove that $\theta(\text{Pref}(a^k x\theta(b))) \cap \text{Suff}(a^k y\theta(b)) = \emptyset$, since the other equality can be proved similarly. Let us assume that $\theta(\text{Pref}(a^k x\theta(b))) \cap \text{Suff}(a^k y\theta(b)) \neq \emptyset$, i.e., there exists $w \in \Sigma^+$ such that $w \in \theta(\text{Pref}(a^k x\theta(b))) \cap \text{Suff}(a^k y\theta(b))$, i.e. $\theta(w) <^\theta_d (a^k x\theta(b))(a^k y\theta(b))$. By Lemma 5.3.1, it is enough to consider only the cases when $1 \leq |w| \leq k + |x|$.

Case 1: $|w| < k$. Then $w = \theta(a^n) = y''\theta(b)$ for some $1 \leq n < k$ and $y = y'y''$, for $y' \in \Sigma^+$ and $y'' \in \Sigma^*$ which implies $\theta(a) = \theta(b)$, i.e., $a = b$, a contradiction.

Case 2: $k \leq |w| \leq k + |x|$. Then $w = \theta(a^k)\theta(x') = a^n y\theta(b)$ for some $1 \leq n \leq k$ and $x = x'x''$, $x', x'' \in \Sigma^*$ which implies $\theta(a) = a$, a contradiction.

Since both the cases lead to a contradiction, we have that

$$\theta(\text{Pref}(a^k x\theta(b))) \cap \text{Suff}(a^k y\theta(b)) = \emptyset$$

Similarly, we can prove that $\theta(\text{Pref}(a^k y\theta(b))) \cap \text{Suff}(a^k x\theta(b)) = \emptyset$. Hence,

$$(a^k y\theta(b))^i (a^k x\theta(b))^j, (a^k x\theta(b))^j (a^k y\theta(b))^i \in D_\theta(1).$$

□

We will illustrate Proposition 5.4.2 with the following example.

Example 5.2 *Let $\Sigma = \{A, C, G, T\}$ and θ be a morphic involution such that $\theta(A) = T$, $\theta(G) = C$ and viceversa. Let $k = 3$, $i = 2$, $j = 1$ and let $x = TAG, y = TC$. Since $x \neq y$, $x, y \in T\Sigma^*$,*

$\theta(a) \neq a$ for all $a \in \Sigma$ and $k > i > j$, we have that,

$$(GGGTAGT)^2(GGGTCT) = GGGTAGTGGGTAGTGGGTCT \in D_\theta(1) \text{ and}$$

$$(GGGTCT)(GGGTAGT)^2 = GGGTCTGGGTAGTGGGTAGT \in D_\theta(1).$$

The following lemma is needed for proving the main result of this section.

Lemma 5.4.3 *[18] Let θ be a literal (anti)morphism on Σ^* and let $a, b \in \Sigma$ such that $a \neq \theta(b)$. Let $x \neq y$, $x, y \in \Sigma^m$, $m > 0$. Then:*

1. *$a^m x\theta(b) \in D(1)$.*
2. *If $x = \theta(b)x'$, $x' \in \Sigma^*$ and $k > m$, then $(a^k y\theta(b))(a^k x\theta(b)) \in D(1)$.*

Now, we will prove one of the main results of the section which shows that, under certain conditions, the set $D^i_\theta(1)\backslash D(i)$ is disjunctive for all $i \geq 2$, alphabet Σ, and morphic involutions θ.

Theorem 5.4.4 *Let $|\Sigma| \geq 3$ and θ be a morphic involution on Σ^* such that $\theta(a) \neq a$ for some $a \in \Sigma$. Then $D^i_\theta(1)\backslash D(i)$ is disjunctive for all $i \geq 2$.*

Proof Since $|\Sigma| \geq 3$ and $\theta(a) \neq a$ for all $a \in \Sigma$ there exists $c \neq a$ such that $\theta(a) = c$ and $\theta(c) = a$. Also, since $|\Sigma| \geq 3$, there exists $b \in \Sigma$ such that $b \neq a$, $b \neq c = \theta(a)$. Let $x, y \in \Sigma^n$, $x \neq y$, $m = n + 1$, $n > 0$. Choose $u = (a^m\theta(b)x\theta(b))^{i-1}a^m\theta(b)$ and $v = \theta(b)$. Since $a \neq b$, by Lemma 5.3.7, $a^m\theta(b)x\theta(b) \in D_\theta(1)$. Also, since $a \neq \theta(b)$, $a^m\theta(b)x\theta(b) \in Q$ and by Lemma 5.4.3, $a^m\theta(b)x\theta(b) \in D(1)$. Hence by Proposition 5.2.2, $\nu_d((a^m\theta(b)x\theta(b))^i) = i$. Thus,

$$uxv = (a^m\theta(b)x\theta(b))^i \in D^i_\theta(1) \cap D(i).$$

On the other hand, by Proposition 5.4.1, since $|x| = |y|$, $(a^m\theta(b)x\theta(b))^{i-1}(a^m\theta(b)y\theta(b)) \in D(1)$ and hence

$$uyv = (a^m\theta(b)x\theta(b))^{i-1}(a^m\theta(b)y\theta(b)) \in D^i_\theta(1)\backslash D(i) \text{ for } i \geq 2.$$

Thus, $x \not\equiv y(P_{D^i_\theta(1)\backslash D(i)})$ for every $x \neq y$, $|x| = |y|$ and $i \geq 2$. Hence, by Proposition 5.2.1, $D^i_\theta(1)\backslash D(i)$ is disjunctive for all $i \geq 2$. □

We know from [30] that $D(1)\backslash\Sigma \subseteq D^2(1)$. Moreover, $D(i)\backslash\Sigma^i \not\subseteq D^{i+1}(1)$ for $i \geq 2$, see [31]. However, for a morphic involution θ, we have that $D_\theta(1)\backslash\Sigma \not\subseteq D^2_\theta(1)$. For example, let $\{a, b\} \in \Sigma$ be such that $\theta(a) = b$ and $\theta(b) = a$. Then $w = abaa \in D_\theta(1)$ but there does not exist any $u, v \in D_\theta(1)$ such that $w = uv$ and $u, v \neq \lambda$. Theorem 5.4.6 establishes, under certain conditions, the relationship between $D_\theta(i)$ and $D_\theta(1)$ for $i \geq 2$. The following is a known result, here with a different proof.

Lemma 5.4.5 *[15] Let θ be a morphic involution on Σ^* and $u \in \Sigma^+$. Then $u \in D_\theta(1)$ if and only if $\theta(u) \in D_\theta(1)$.*

Proof Let $u \in D_\theta(1)$. Assume that $\theta(u) \notin D_\theta(1)$. Then there exists $v, \alpha, \beta \in \Sigma^+$ such that $\theta(u) = v\alpha = \beta\theta(v)$ which implies $u = \theta(v)\theta(\alpha) = \theta(\beta)v$ which further implies $u \notin D_\theta(1)$, a contradiction. Hence $\theta(u) \in D_\theta(1)$. Similarly, we can prove that if $\theta(u) \in D_\theta(1)$ then $u \in D_\theta(1)$. □

Theorem 5.4.6 *Let θ be a morphic involution on Σ^*, where Σ is an alphabet with $|\Sigma| > 2$ that contains letters $a \neq b$ such that $a \neq \theta(b)$ and $\theta(a) \neq a$ for all $a \in \Sigma$. Then the set $D^{2i}_\theta(1)\backslash D_\theta(i+1)$ is disjunctive for all $i \geq 1$.*

Proof Let $x, y \in \Sigma^n$, $x \neq y$, $m = n + 1$, $n > 0$. Let $u = a^m\theta(b)$,

$$v = \theta(b)(\theta(a^m\theta(b)x\theta(b))a^m\theta(b)x\theta(b))^{i-1}\theta(a^m\theta(b)x\theta(b)).$$

Since $a \neq b$, by Lemma 5.3.7, we have $a^m\theta(b)x\theta(b) \in D_\theta(1)$. Hence by Lemma 5.4.5, $\theta(a^m\theta(b)x\theta(b)) \in D_\theta(1)$. Thus, by Proposition 5.3.5,

$$uxv = (a^m\theta(b)x\theta(b)\theta(a^m\theta(b)x\theta(b)))^i \in D_\theta(i + 1) \cap D^{2i}_\theta(1).$$

Further by Lemma 5.3.6,

$$uyv = a^m\theta(b)y\theta(b)(\theta(a^m\theta(b)x\theta(b))a^m\theta(b)x\theta(b))^{i-1}\theta(a^m\theta(b)x\theta(b)) \in D_\theta(1)$$

and hence $uyv \in D_\theta^{2i}(1)\backslash D_\theta(i+1)$ for $i \geq 1$. Therefore, $x \not\equiv y(P_{D_\theta^{2i}(1)\backslash D_\theta(i+1)})$ for every $x, y \in \Sigma^+$, $x \neq y$, $|x| = |y|$ and $i \geq 1$. By Proposition 5.2.1, $D_\theta^{2i}(1)\backslash D_\theta(i+1)$ is disjunctive for $i \geq 1$. □

5.5 Disjunctivity of the set $(D_\theta(2) \cap D(2))\backslash(D_\theta(1) \cap D(1))^k$ for $k = 1, 2$

We have already discussed some relationships between the sets $D_\theta(i)$ and $D(i)$. In particular, the intersection of these two sets is a non-empty set and that, under certain conditions, the sets $D_\theta^i(1)\backslash D(i)$ are disjunctive for all $i \geq 2$. A natural question that arises in this context is what are the relationships between the sets $D_\theta(i) \cap D(i)$ for different values of $i \geq 1$. In this section, we will show that under certain conditions the set $(D_\theta(2) \cap D(2))\backslash(D_\theta(1) \cap D(1))^k$ is disjunctive for $k = 1, 2$ (Theorem 5.5.6).

In order to prove the disjunctivity of the set $(D_\theta(2) \cap D(2))\backslash(D_\theta(1) \cap D(1))^k$, $k = 1, 2$, we need to characterize a word or set of words that have exactly two borders and two θ-borders. The following proposition provides such characterization.

Proposition 5.5.1 *Let θ be a morphic involution such that $\theta(a) \neq a$ for all $a \in \Sigma$, let $u \in D_\theta(1) \cap D(1)$ and let $u' \in Pref(u) \cup Suff(u) \cup \theta(Suff(u))$ with $u' \neq u$. Then $w = u\theta(u)u'\theta(u)u \in D_\theta(2) \cap D(2)$.*

Proof Let $w = u\theta(u)u'\theta(u)u$. Let us assume that $u' \in \text{Pref}(u)$. Clearly, $\{\lambda, u\} \subseteq L_d(w)$ and $\{\lambda, u\theta(u)\} \subseteq L_d^\theta(u)$. Now, we need to show that $L_d(w) \subseteq \{\lambda, u\}$ and $L_d^\theta(u) \subseteq \{\lambda, u\theta(u)\}$. Since $a \neq \theta(a)$ for all $a \in \Sigma$, $u\theta(u) \notin L_d(w)$ and $u \notin L_d^\theta(u)$. Let us assume that $x \in L_d(w)$ or $y \in L_d^\theta(w)$, i.e. $x <_d w$ or $y <_d^\theta w$ such that $x, y \notin \{\lambda, u, u\theta(u)\}$ and let, $|u| = m, |u'| = n$ where $m > n > 0$.

Since $x, y \notin \{u, u\theta(u)\}$, the cases $|x| = |y| = m$ and $|x| = |y| = 2m$ are not possible. Thus we have following 5 cases to consider.

Case 1: If $0 < |x| < m$, then $x \in \text{Pref}(u) \cap \text{Suff}(u)$ which implies $u \notin D(1)$, a contradiction.

If $0 < |y| < m$, then $y \in \text{Pref}(u) \cap \theta(\text{Suff}(u))$ which implies $u \notin D_\theta(1)$, a contradiction.

Case 2: If $m < |x| < 2m$, then for $u = u_1u_2 = u_1'u_2'$ and $|u_1| = |u_2'|$, we will get $x = u\theta(u_1) = \theta(u_2')u = u_1'u_2'\theta(u_1) = \theta(u_2')u_1'u_2'$ where $u_1, u_2, u_1', u_2' \in \Sigma^+$ which implies $\theta(u_1) = u_2'$ which further implies $u \notin D_\theta(1)$, a contradiction.

If $m < |y| < 2m$, then for $u = u_3u_4 = u_3'u_4'$ and $|u_3| = |u_4'|$, we will get $y = u\theta(u_3) = u_4'\theta(u) = u_3'u_4'\theta(u_3) = u_4'\theta(u_3')\theta(u_4')$ where $u_3, u_4, u_3', u_4' \in \Sigma^+$ which implies $\theta(u_3) = \theta(u_4')$, i.e., $u_3 = u_4'$ which further implies $u \notin D(1)$, a contradiction.

Case 3: If $2m < |x| \leq 2m + n$, then $x = u\theta(u)u_1' = u_2'\theta(u)u$ where $u_1' \leq_p u' <_p u$ and $u_2' \leq_s u'$ for $u_1', u_2' \in \Sigma^+$. Since, $|u_1'| \leq |u'| < |u|$, $u_1' <_s u$ which implies $u \notin D(1)$, a contradiction.

If $2m < |y| \leq 2m + n$, then $y = u\theta(u)u_3' = \theta(u_4')u\theta(u)$ where $u_3' \leq_p u' <_p u$ and $u_4' \leq_s u'$ for $u_3', u_4' \in \Sigma^+$. Since $|u_3'| \leq |u'| < |u|$, $u_3' <_s \theta(u)$, i.e., $\theta(u_3') <_s u$ which implies $u \notin D_\theta(1)$, a contradiction.

Case 4: If $2m + n < |x| \leq 3m + n$. Then $x = u\theta(u)u'\theta(u_1) = \theta(u_2)u'\theta(u)u$ where $u_1 \leq_p u$ and $u_2 \leq_s u$ for $u_1, u_2 \in \Sigma^+$. Since, $|u_1| \leq |u|$, $\theta(u_1) \leq_s u$ which implies $u \notin D_\theta(1)$, a contradiction.

If $2m + n < |y| \leq 3m + n$. Then $y = u\theta(u)u'\theta(u_3) = u_4\theta(u')u\theta(u)$ where $u_3 \leq_p u$ and $u_4 \leq_s^\theta u$ for $u_3, u_4 \in \Sigma^+$. Since, $|u_3| \leq |u|$, $\theta(u_3) \leq_s \theta(u)$, i.e., $u_3 \leq_s u$ which implies $u \notin D(1)$, a contradiction.

Case 5: If $3m + n < |x| < 4m + n$. Then $x = u\theta(u)u'\theta(u)u_1 = u_2\theta(u)u'\theta(u)u$ where $u_1 <_p u$, $u_2 <_s u$ for $u_1, u_2 \in \Sigma^+$. Since, $|u_1| < |u|$, $u_1 <_s u$, which implies $u \notin D(1)$, a contradiction.

If $3m + n < |y| < 4m + n$. Then $y = u\theta(u)u'\theta(u)u_3 = \theta(u_4)u\theta(u')u\theta(u)$ where $u_3 <_p u$, $\theta(u_4) <_s u$ for $u_3, u_4 \in \Sigma^+$. Since, $|u_3| < |u|$, $u_3 <_s \theta(u)$, i.e., $\theta(u_3) <_s u$ which implies $u \notin D_\theta(1)$, a contradiction.

If we assume that $u' \in \text{Suff}(u)$ or $u' \in \theta(\text{Suff}(u))$, we will reach a similar contradiction.

Since all the cases lead to a contradiction, $w \in D_\theta(2) \cap D(2)$. □

We illustrate Proposition 5.5.1 with the following example.

Example 5.3 *Let $\Sigma = \{A, C, G, T\}$ and θ be a morphic involution such that $\theta(A) = T$, $\theta(G) = C$ and vice versa. Let $u = GTA$ and $u' = GT$. Then for $w = u\theta(u)u'\theta(u)u = GTACATGTCATGTA$, $L_d^\theta(w) = \{\lambda, GTACAT\}$ and $L_d(w) = \{\lambda, GTA\}$. Hence $w \in D_\theta(2) \cap D(2)$.*

Similarly, we need to prove that the words of the form mentioned in Proposition 5.5.1 cannot be decomposed as a catenation of less than three words which are unbordered as well as θ-unbordered.

Proposition 5.5.2 *Let θ be a morphic involution such that $\theta(a) \neq a$ for all $a \in \Sigma$ and let $u \in D_\theta(1) \cap D(1)$, $u' \in Pref(u) \cup Suff(u) \cup \theta(Suff(u))$ such that $u' \neq u$, then $u\theta(u)u'\theta(u)u \notin (D_\theta(1) \cap D(1))^n$ for $1 \leq n \leq 2$.*

Proof Let us assume that $w = u\theta(u)u'\theta(u)u \in (D_\theta(1) \cap D(1))^n$ for $1 \leq n \leq 2$. Let us assume that $u' \in \text{Pref}(u)$. The case $n = 1$ is not possible since by Proposition 5.5.1, $w \in D_\theta(2) \cap D(2)$. Hence, let $n = 2$, i.e., $u\theta(u)u'\theta(u)u = v_1v_2$ where $v_1, v_2 \in D_\theta(1) \cap D(1)$. Then we have following cases to consider.

Case 1: $v_1 = u$, $v_2 = \theta(u)u'\theta(u)u$. Then $v_2 \in D_\theta(2)$, a contradiction.

Case 2: $v_1 = u\theta(u)$, $v_2 = u'\theta(u)u$. Then, $v_1 \in D_\theta(2)$, a contradiction.

Case 3: $v_1 = u\theta(u)u'$, $v_2 = \theta(u)u$. Then $v_2 \in D_\theta(2)$, a contradiction.

Case 4: $v_1 = u\theta(u)u'\theta(u)$, $v_2 = u$. Then $v_1 \in D_\theta(2)$, a contradiction.

Case 5: $v_1 = u_1$, $v_2 = u_2\theta(u)u'\theta(u)u$ where $u = u_1u_2$ and $u_1, u_2 \in \Sigma^+$. This implies $v_2 = u_2\theta(u)u'\theta(u)u_1u_2 \in D(2)$, a contradiction.

Case 6: $v_1 = u\theta(x_1)$, $v_2 = \theta(x_2)u'\theta(u)u$ where $u = x_1x_2$ and $x_1, x_2 \in \Sigma^+$. This implies, $v_2 = \theta(x_2)u'\theta(u)x_1x_2 \in D_\theta(2)$, a contradiction.

Case 7: $v_1 = u\theta(u)u'_1$, $v_2 = u'_2\theta(u)u$ where $u' = u'_1u'_2$ and $u'_1, u'_2 \in \Sigma^+$. Also, since $u' <_p u$, $u = u'u'_3 = u'_1u'_2u'_3$ where $u'_3 \in \Sigma^+$. This implies, $v_1 = u'_1u'_2u'_3\theta(u)u'_1, \in D(2)$, a contradiction.

Case 8: $v_1 = u\theta(u)u'\theta(u_1)$, $v_2 = \theta(u_2)u$ where $u = u_1u_2$ and $u_1, u_2 \in \Sigma^+$ which implies, $v_1 = u_1u_2\theta(u)u'\theta(u_1) \in D_\theta(2)$, a contradiction.

Case 9: $v_1 = u\theta(u)u'\theta(u)u_1$, $v_2 = u_2$ where $u = u_1u_2$ and $u_1, u_2 \in \Sigma^+$, which implies, $v_1 = u_1u_2\theta(u)u'\theta(u)u_1 \in D(2)$, a contradiction.

If we assume that $u' \in \text{Suff}(u)$ or $u' \in \theta(\text{Suff}(u))$, we will reach a similar contradiction.

Since all cases led to contradictions, $w \notin (D_\theta(1) \cap D(1))^n$ for $1 \leq n \leq 2$. □

As a consequence of Proposition 5.5.1 and 5.5.2, we have the following result.

Corollary 5.5.3 *Let θ be a morphic involution such that $\theta(a) \neq a$ for all $a \in \Sigma$ and let $u \in D_\theta(1) \cap D(1)$, $u' \in Pref(u) \cup Suff(u) \cup \theta(Suff(u))$ such that $u' \neq u$. Then $u\theta(u)u'\theta(u)u \in (D_\theta(2) \cap D(2))\backslash(D_\theta(1) \cap D(1))^n$ for $1 \leq n \leq 2$.*

The following result is needed for the proof of Lemma 5.5.5.

Lemma 5.5.4 *[18] Let θ be a morphic involution on Σ^*, where Σ is an alphabet with $|\Sigma| \geq 3$ that contains letters $a \neq b$ such that $a \notin \{\theta(b), \theta(a)\}$. Let $x \neq y$, $x, y \in \Sigma^m$, $m > 0$. Then:*

1. $a^m x\theta(b) \in D_\theta(1) \cap D(1)$.

2. *If $x = \theta(b)x'$, $x' \in \Sigma^*$ and $k \geq m$, then $(a^k y\theta(b))(a^k x\theta(b)) \in D_\theta(1) \cap D(1)$.*

In the following lemma we prove that, for a morphic involution θ, certain words of the form $u\theta(u)u'\theta(u)v$, where $u' <_p u$ and $u \neq v$, are unbordered as well as θ-unbordered.

Lemma 5.5.5 *Let $|\Sigma| \geq 3$, θ be a morphic involution with the property that there exists $a \in \Sigma$ such that $a \notin \{\theta(a), b, \theta(b)\}$. Then for $u, v \in \Sigma^n$ such that $u \neq v$, $n > 0$ and $m = n + 2$,*

$$a^m\theta(b)u\theta(b)\theta(a^m\theta(b)u\theta(b))a^m\theta(a^m\theta(b)u\theta(b))a^m\theta(b)v\theta(b) \in D_\theta(1) \cap D(1).$$

Proof Let us assume that

$$w = a^m\theta(b)u\theta(b)\theta(a^m\theta(b)u\theta(b))a^m\theta(a^m\theta(b)u\theta(b))a^m\theta(b)v\theta(b) \notin D_\theta(1) \cap D(1),$$

i.e., there exists $w_1, w_2 \in \Sigma^+$ such that $w_1 <_d w$ or $w_2 <_d^\theta w$. By Lemma 5.2.4 and Lemma 5.3.1, it is enough consider the case $|w_1| < 5m$ or $|w_2| < 5m$. Further by Lemma 5.5.4, taking $y = \theta(b)u$ and $x = \theta(b)v$ we know that $(a^m\theta(b)u\theta(b))(a^m\theta(b)v\theta(b)) \in D_\theta(1) \cap D(1)$, hence none of the prefixes of $a^m\theta(b)u\theta(b)$ can be a border or a θ-border of w and hence the cases $1 \leq |w_1| \leq 2m$ or $1 \leq |w_2| \leq 2m$ are not possible. So, we only need to consider the cases when $2m < |w_1| < 5m$ or $2m < |w_2| < 5m$.

Case 1: $2m < |w_1| < 3m$ or $2m < |w_2| < 3m$. Then,

$$w_1 = a^m\theta(b)u\theta(b)\theta(a^k) = \theta(u_2)ba^m\theta(b)v\theta(b) \text{ or}$$

$$w_2 = a^m\theta(b)u\theta(b)\theta(a^{k'}) = u_2'\theta(b)\theta(a^m)b\theta(v)b$$

where $1 \leq k, k' < m$, $u = u_1u_2 = u_1'u_2'$, $u_1, u_1' \in \Sigma^+$ and $u_2, u_2' \in \Sigma^*$ which implies $a = b$ or $a = \theta(b)$, a contradiction.

Case 2: $|w_1| = 3m$ or $|w_2| = 3m$. Then,

$$w_1 = a^m\theta(b)u\theta(b)\theta(a^m) = b\theta(u)ba^m\theta(b)v\theta(b) \text{ or}$$

$$w_2 = a^m\theta(b)u\theta(b)\theta(a^m) = \theta(b)u\theta(b)\theta(a^m)b\theta(v)b$$

which implies $a = b$ or $a = \theta(b)$, a contradiction.

Case 3: $|w_1| = 3m + 1$ or $|w_2| = 3m + 1$. Then,

$$w_1 = a^m\theta(b)u\theta(b)\theta(a^m)b = \theta(a)b\theta(u)ba^m\theta(b)v\theta(b) \text{ or}$$

$$w_2 = a^m\theta(b)u\theta(b)\theta(a^m)b = a\theta(b)u\theta(b)\theta(a^m)b\theta(v)b$$

which implies $a = \theta(a)$ (and $a = b$) or $a = \theta(b)$, a contradiction.

Case 4: $3m + 1 < |w_1| \leq 4m - 1$ or $3m + 1 < |w_2| \leq 4m - 1$. Then,

$$w_1 = a^m\theta(b)u\theta(b)\theta(a^m)b\theta(u_1) = \theta(a^k)b\theta(u)ba^m\theta(b)v\theta(b) \text{ or}$$

$$w_2 = a^m\theta(b)u\theta(b)\theta(a^m)b\theta(u_1') = a^{k'}\theta(b)u\theta(b)\theta(a^m)b\theta(v)b$$

where $2 \leq k, k' \leq n + 1$, $u = u_1u_2 = u_1'u_2'$, $u_1, u_1' \in \Sigma^+$ and $u_2, u_2' \in \Sigma^*$ which implies $a = b$ (and $\theta(a) = a$) or $a = \theta(b)$, a contradiction.

Case 5: $|w_1| = 4m$ or $|w_2| = 4m$. Then,

$$w_1 = a^m\theta(b)u\theta(b)\theta(a^m)b\theta(u)b = \theta(a^m)b\theta(u)ba^m\theta(b)v\theta(b) \text{ or}$$

$$w_2 = a^m\theta(b)u\theta(b)\theta(a^m)b\theta(u)b = a^m\theta(b)u\theta(b)\theta(a^m)b\theta(v)b$$

which implies $a = \theta(a)$ or $\theta(u) = \theta(v)$, i.e., $u = v$, a contradiction.

Case 6: $4m < |w_1| < 5m$ or $4m < |w_2| < 5m$. Then,

$$w_1 = a^m\theta(b)u\theta(b)\theta(a^m)b\theta(u)ba^k = a^{k'}\theta(a^m)b\theta(u)ba^m\theta(b)v\theta(b) \text{ or}$$

$$w_2 = a^m\theta(b)u\theta(b)\theta(a^m)b\theta(u)ba^{k_1} = \theta(a^{k_2})a^m\theta(b)u\theta(b)\theta(a^m)b\theta(v)b$$

where $1 < k, k', k_1, k_2 < m$ which implies $a = \theta(b)$ (and $a = \theta(a)$) or $a = b$ (and $a = \theta(a)$), a contradiction.

Since all the cases lead to a contradiction $w \in D_\theta(1) \cap D(1)$. □

The following theorem uses Lemma 5.5.5, along with certain conditions on the alphabet Σ, to show the disjunctivity of the languages $(D_\theta(2) \cap D(2))\backslash(D_\theta(1) \cap D(1))^k$ for $k = 1, 2$.

Theorem 5.5.6 *Let $|\Sigma| \geq 3$ and θ be a morphic involution such that $\theta(a) \neq a$ for all $a \in \Sigma$. Then $(D_\theta(2) \cap D(2))\backslash(D_\theta(1) \cap D(1))^k$ is disjunctive for $k = 1, 2$.*

Proof Let $\{a, b\} \in \Sigma$ such that $a \notin \{b, \theta(b)\}$. Let $u, v \in \Sigma^n$ for $n > 0$ be such that $u \neq v$. Let

$m = |bub| = n + 2$. Now let,

$$x = a^m\theta(b)u\theta(b)\theta(a^m\theta(b)u\theta(b))a^m\theta(a^m\theta(b)u\theta(b))a^m\theta(b)$$

and $y = \theta(b)$. Then

$$xuy = a^m\theta(b)u\theta(b)\theta(a^m\theta(b)u\theta(b))a^m\theta(a^m\theta(b)u\theta(b))a^m\theta(b)u\theta(b) \text{ and}$$

$$xvy = a^m\theta(b)u\theta(b)\theta(a^m\theta(b)u\theta(b))a^m\theta(a^m\theta(b)u\theta(b))a^m\theta(b)v\theta(b).$$

By Corollary 5.5.3, $xuy \in (D_\theta(2) \cap D(2))\backslash(D_\theta(1) \cap D(1))^k$ for $k = 1, 2$. Now, by Lemma 5.5.5, we know that $xvy \in D_\theta(1) \cap D(1)$. Therefore, $u \not\equiv v(P_L)$ for every $u, v \in \Sigma^+$, $u \neq v$, $|u| = |v|$ and $L = (D_\theta(2) \cap D(2))\backslash(D_\theta(1) \cap D(1))^k$ for $k = 1, 2$ is, by Proposition 5.2.1, disjunctive. □

We conclude this section with some observations on the disjunctivity of some other languages related to $D_\theta(i)$, $i \geq 1$. Let us recall the definition of a singular language from [26]. For any language $L \subseteq \Sigma^+$, [26] defines,

$$l(L) = \{g \in L | gx \notin L \text{ for all } x \in \Sigma^+ \text{ and } g = yz, z \in \Sigma^+, \text{ implies } y \notin L\}.$$

Each element of $l(L)$ is called a *singular word in* L and L is said to be a *singular language* if $l(L) \neq \emptyset$.

Theorem 5.5.7 *[26] Let L' be a disjunctive language and let L be a singular language. Then LL' is a disjunctive language.*

Corollary 5.5.8 *If Σ is such that $|\Sigma| > 2$ and $a \neq \theta(a)$ for all $a \in \Sigma$, θ is a morphic involution on Σ^*, and L is a singular language over Σ, then the following hold:*

1. *If there exist $\{a, b\} \in \Sigma$ such that $a \notin \{b, \theta(b)\}$ then $LD_\theta(i)$ is disjunctive for all $i \geq 1$.*
2. *If there exist $\{a, b\} \in \Sigma$ such that $a \notin \{b, \theta(b)\}$ then $L(D_\theta(i) \cap Q_\theta^{2i-2})$ is disjunctive for all $i \geq 2$.*

3. *The language* $L(D^i_\theta(1)\backslash D(i))$ *is disjunctive for all* $i \geq 2$.

4. *If there exist* $\{a, b\} \in \Sigma$ *such that* $a \notin \{b, \theta(b)\}$ *then* $L(D^{2i}_\theta(1)\backslash D_\theta(i+1))$ *is disjunctive for all* $i \geq 1$.

5. *The language* $L((D_\theta(2) \cap D(2))\backslash(D_\theta(1) \cap D(1))^k)$ *is disjunctive for* $k = 1, 2$.

Proof 1. By Theorems 5.3.8 and 5.5.7, $LD_\theta(i)$ is disjunctive for $i \geq 1$.

2. By Corollary 5.3.9 and Theorem 5.5.7, $L(D_\theta(i) \cap Q^{2i-2}_\theta)$ is disjunctive for all $i \geq 2$.

3. By Theorems 5.4.4 and 5.5.7, $L(D^i_\theta\backslash D(i))$ is disjunctive for all $i \geq 2$.

4. By Theorems 5.4.6 and 5.5.7, $L(D^{2i}_\theta(1)\backslash D_\theta(i+1))$ is disjunctive for all $i \geq 1$.

5. By Theorems 5.5.6 and 5.5.7, $L((D_\theta(2)\cap D(2))\backslash(D_\theta(1)\cap D(1))^k)$ is disjunctive for $k = 1, 2$.

□

5.6 Further remarks on $D_\theta(i)$ and related languages

As seen in Section 4, for a word $u \in D_\theta(1)$ there might not exist a decomposition $u = u_1u_2$ such that $u_1, u_2 \in D_\theta(1)$. If, however, such a decomposition exists for a non-empty word, then that word is said to be $D_\theta(1)$-*concatenate*. The word u is said to be *completely* $D_\theta(1)$-*concatenate*, if $u = xy$ for $x, y \in \Sigma^+$, imply that $x, y \in D_\theta(1)$. These notions generalize concepts related to $D(1)$-*concatenate* words, defined in [14].

Example 5.4 *Let* $\Sigma = \{a, b\}$*, and* θ *be (anti)morphic involution such that* $\theta(a) = b$ *and vice versa. Then* $u = ab$ *is* $D_\theta(1)$*-concatenate. Also,* $v = a^i$*,* $i \geq 1$ *is completely* $D_\theta(1)$*-concatenate, but* $w = aba = (a\theta(a))(a)$ *is not* $D_\theta(1)$*-concatenate and hence not completely* $D_\theta(1)$*-concatenate.*

The following proposition shows that the set of all completely $D_\theta(1)$-concatenate words is regular for an (anti)morphic involution θ.

Proposition 5.6.1 *Let Σ be an alphabet, L be the set of all completely $D_\theta(1)$-concatenate words over Σ, and let θ be an (anti)morphic involution. Then L is regular.*

Proof Let $\Sigma = \{a_1, a_2, \ldots, a_n\}$. Let $u \in L$ be such that $u = a_i w a_j$, $1 \leq i, j \leq n$, $w \in \Sigma^*$. If w does not contain $\theta(a_i)$ and $\theta(a_j)$, then for $w = w'w''$, $w', w'' \in \Sigma^*$, $a_i w', w'' a_j \in D_\theta(1)$. On the other hand, if w contains $\theta(a_i)$ or $\theta(a_j)$, then for some $w', w'' \in \Sigma^*$, $w = w'\theta(a_i)w''$ or $w = w'\theta(a_j)w''$. Thus we have, $u = (a_i w'\theta(a_i))w'' a_j$ or $a_i w'(\theta(a_j)w'' a_j)$, which contradicts to the fact that u is completely $D_\theta(1)$-concatenate. Thus,

$$L = \bigcup_{i=1,j=1}^{n} a_i(\Sigma \backslash \{\theta(a_i), \theta(a_j)\})^* a_j.$$

Since $a_i(\Sigma\backslash\{\theta(a_i), \theta(a_j)\})^* a_j$ is regular, L is regular. □

The catenation of θ-unbordered words is not necessarily θ-unbordered. Additional conditions, such as the one below, are needed to guarantee that the catenation of θ-unbordered words is θ-unbordered.

Proposition 5.6.2 *[19] Let θ be either a morphic or an antimorphic involution and let $u, v \in \Sigma^+$ be θ-unbordered. Then uv is θ-unbordered iff $\theta(Pref(u)) \cap Suff(v)) = \emptyset$.*

Based on above proposition and the notion of non-overlapped languages defined in [31], we now introduce a new class of languages, called θ-non-overlapped languages. θ-non-overlapped languages are a special class of θ-unbordered words, whose additional properties imply that the catenation between any two words in the language remains θ-unbordered.

A pair of words $u, v \in \Sigma^+$, $u \neq v$, is said to be *θ-non-overlapped* iff $\theta(\text{Pref}(u)) \cap \text{Suff}(v) = \emptyset$ and $\theta(\text{Pref}(v)) \cap \text{Suff}(u) = \emptyset$. A language $L \subseteq \Sigma^+$ is said to be *θ-non-overlapped* if $L \subseteq D_\theta(1)$ and $u, v \in L$, $\theta(u) \neq v$, implies u and v are θ-non-overlapped.

For a language L, let us denote $L_\theta^{(2)} = \{u\theta(u) | u \in L\}$. The following results describe some properties of θ-non-overlapped languages.

Lemma 5.6.3 *Let θ be morphic involution and L be θ-non-overlapped. Then $\theta(L)$ is also θ-non-overlapped.*

The following proposition shows the necessary and sufficient condition for a language to be θ-non-overlapped.

Proposition 5.6.4 *Let $L \subseteq \Sigma^+$ and θ be a morphic involution. Then L is θ-non-overlapped language if and only if $L \subseteq D_\theta(1)$ and $L^2 \backslash L_\theta^{(2)} \subseteq D_\theta(1)$.*

Proof Let $L \subseteq \Sigma^+$. Assume that L is a θ-non-overlapped language. Then $L \subseteq D_\theta(1)$. Now, let, $u, v \in L$ such that $v \neq \theta(u)$, i.e. $uv \in L^2 \backslash L_\theta^{(2)}$. Suppose $uv \notin D_\theta(1)$, then there exists $w \in \Sigma^+$ such that $w <_d^\theta uv$. If $|w| > |u|$, then there exists $w' \in \Sigma^+$ such that $w = uw'$ and $uv = uw'\alpha = \beta\theta(u)\theta(w')$ for $\alpha, \beta \in \Sigma^+$. Thus $w' <_d^\theta v$ and $L \nsubseteq D_\theta(1)$, a contradiction. We will reach a similar contradiction if we assume that $|w| > |v|$. If $|w| \leq |u|$ and $|w| \leq |v|$, then $\theta(w) \in \theta(\text{Pref}(u)) \cap \text{Suff}(v)$, a contradiction. Hence $uv \in D_\theta(1)$ and $L^2 \backslash L_\theta^{(2)} \subseteq D_\theta(1)$.

Conversely, assume that $L \subseteq D_\theta(1)$ and $L^2 \backslash L_\theta^{(2)} \subseteq D_\theta(1)$. Consider $u, v \in L$ such that $v \neq \theta(u)$. Then, clearly $uv \in L^2 \backslash L_\theta^{(2)}$. Suppose u, v are not θ-non-overlapped, then $\theta(\text{Pref}(u)) \cap \text{Suff}(v) \neq \emptyset$ or $\theta(\text{Pref}(v)) \cap \text{Suff}(u) \neq \emptyset$. Let $w \in \theta(\text{Pref}(u)) \cap \text{Suff}(v)$ which implies $\theta(w) <_d^\theta uv$. Thus $uv \notin D_\theta(1)$, which is a contradiction to the assumption that $L^2 \backslash L_\theta^{(2)} \subseteq D_\theta(1)$. We will reach a similar contradiction if we assume that $w' \in \theta(\text{Pref}(v)) \cap \text{Suff}(u)$. Thus, $\theta(\text{Pref}(u)) \cap \text{Suff}(v) = \emptyset$ and $\theta(\text{Pref}(v)) \cap \text{Suff}(u) = \emptyset$ for every $u, v \in L$, $\theta(u) \neq v$ and $L \subseteq D_\theta(1)$, i.e. L is θ-non-overlapped. □

We will illustrate Proposition 5.6.4 with the following example.

Example 5.5 *Let $\Sigma = \{A, C, G, T\}$ and θ be a morphic involution such that $\theta(A) = T$, $\theta(G) = C$ and viceversa. Let $L = \{AG, GACG\}$, which is a θ-non-overlapped language. Then $L \subseteq D_\theta(1)$ and*

$$L^2 \backslash L_\theta^{(2)} = \{AGAG, AGGAGC, GAGCGAGC, GAGCAG\} \in D_\theta(1).$$

Proposition 5.6.5 *Let $L \subseteq \Sigma^+$ be a θ-non-overlapped language and let $w \in L^m$ for some $m \geq 1$. If there exists $u \in \Sigma^+$ such that $u \leq_d^\theta w$, then $u \in L^i(\theta(L))^j$ for some $1 \leq i, j, \leq m$.*

Proof Let $w \in L^m$, i.e., $w = w_1 w_2 \dots w_m$ for some $w_1, w_2, \dots w_m \in L$. Let $u \in \Sigma^+$ be such that $u \leq_d^\theta w$. Then there exist $1 \leq l \leq m$ such that $u = w_1 \dots w_{l-1} u_1$, where $u_1 \in \text{Pref}(w_l)$. Thus $u_1 \leq_d^\theta w_l \dots w_m$. Similarly, there exist $l \leq k \leq m$ such that $\theta(u_1) = u_2 w_{k+1} \dots w_m$ where $u_2 \in \text{Suff}(w_k)$ which implies $u_1 = \theta(u_2)\theta(w_{k+1}) \dots \theta(w_m)$. Now, since $\theta(u_2) \leq_p u_1 \leq_p w_l$ and $u_2 \leq_s w_k$, we will get $\theta(u_2) \leq_d^\theta w_l w_k$. Since by Proposition 5.6.4, $L^2 \backslash L_\theta^{(2)} \subseteq D_\theta(1)$, $w_k = \theta(w_l)$. Also, since $L, \theta(L) \subseteq D_\theta(1)$, $w_k = \theta(w_l) = u_2$. Thus,

$$\begin{aligned} u &= w_1 \dots w_{l-1} u_1 \\ &= w_1 \dots w_{l-1} \theta(u_2)\theta(w_{k+1}) \dots \theta(w_m) \\ &= w_1 \dots w_{l-1} w_l \theta(w_{k+1}) \dots \theta(w_m) \in L^l(\theta(L))^{m-k-1}. \end{aligned}$$

□

For a word $u \in \Sigma^+$,

$$\text{IN}(u) = \{v \in \Sigma^+ | u = xvy \text{ for some } x, y \in \Sigma^*\}.$$

The following result shows the relationship between the length of an infix of a θ-periodic word and the number of borders as well as θ-borders of such infix, for morphic involutions θ.

Proposition 5.6.6 *Let $u, v \in \Sigma^+$ and θ be a morphic involution. If $v \in D_\theta(i_1) \cap D(i_2)$ with $i = i_1 + i_2$ for $i_1, i_2 \geq 1$ and $v \in IN(u_1 u_2 \dots u_m)$ where $u_k = u$ if k is odd and $u_k = \theta(u)$ if k is even for $1 \leq k \leq m$ and $m \geq i$, then $|v| \leq |u^i|$.*

Proof Let us assume that $|v| > |u^i|$. Then there is an integer $i \leq j < m$ such that $|u^j| < |v| \leq |u^{j+1}|$. Hence, v is of the form $v = (v_1 v_2 \dots v_j)v'$ where $|v_l| = |u|$, $1 \leq l \leq j$, and $v' \in \Sigma^+$, $|v'| \leq |u|$. Since, v is an infix of a word of the form $u_1 u_2 \dots u_m$ where $u_k = u$ if k is odd and $u_k = \theta(u)$ if

k is even for $1 \leq k \leq m$, there exists a word $w \in \Sigma^+$, $|w| = |u|$, such that $v_l = w$ if l is odd and $v_l = \theta(w)$ if l is even, $1 \leq l \leq j$. Furthermore, if j is odd, then $v' \leq_p \theta(w)$ and if j is even, then $v' \leq_p w$. We have the following two cases to consider.

Case 1: j is odd. Then $v = (w\theta(w)w\theta(w)\dots w)\theta(w')$ for $w = w'\alpha$ where $w' \in \Sigma^+$ and $\alpha \in \Sigma^*$. This implies, $v = (w'\alpha\theta(w')\theta(\alpha)\dots w'\alpha)\theta(w')$.

Thus, $\lambda, w', (v_1v_2)w', \dots, (v_1v_2\dots v_{j-1})w' \in L_d^\theta(v)$ which implies $\nu_d^\theta(v) = \frac{j+3}{2}$.

Also, $\lambda, v_1\theta(w'), (v_1v_2v_3)\theta(w'), \dots, (v_1v_2\dots v_{j-2})\theta(w') \in L_d(v)$. This implies $\nu_d(v) = \frac{j+1}{2}$.

Hence, $\nu_d^\theta(v) + \nu_d(v) = \frac{j+3}{2} + \frac{j+1}{2} = j + 2 \geq i + 2 > i$, which is a contradiction.

Case 2: j is even. Then $v = (w\theta(w)w\theta(w)\dots\theta(w))w''$ for $w = w''\beta$ where $w'' \in \Sigma^+$ and $\beta \in \Sigma^*$. This implies, $v = (w''\beta\theta(w'')\theta(\beta)\dots\theta(w'')\theta(\beta))w''$.

Thus, $\lambda, w'', (v_1v_2)w'', \dots, (v_1v_2\dots v_{j-2})w'' \in L_d(v)$ which implies $\nu_d(v) = \frac{j+2}{2}$.

Also, $\lambda, v_1\theta(w''), (v_1v_2v_3)\theta(w''), \dots, (v_1v_2\dots v_{j-1})\theta(w'') \in L_d\theta(v)$. This implies $\nu_d(v) = \frac{j+2}{2}$.

Hence, $\nu_d^\theta(v) + \nu_d(v) = \frac{j+2}{2} + \frac{j+2}{2} = j + 2 \geq i + 2 > i$, which is a contradiction.

Since both the cases lead to a contradiction, $|v| \leq |u^i|$. □

We conclude with a preview of possible generalizations of this research to cases where $\theta^3 = I$ over Σ or, more generally, where $\theta^n = I$ over Σ. In [19] it was shown that, for a morphic involution θ, the set of all θ-bordered words over Σ is not context-free. The following results shows that this holds also for the case of a morphism θ with the property that $\theta(a) \neq a$ for all $a \in \Sigma$ and θ^3 equals the identity on Σ with $|\Sigma| \geq 3$.

Proposition 5.6.7 *If $|\Sigma| \geq 3$, θ is a morphism such that $\theta^3 = I$ on Σ and $\theta(a) \neq a$ for all $a \in \Sigma$, then the set of all θ-bordered words over Σ is not context-free.*

Proof Let $a \in \Sigma$. Now, since $a \neq \theta(a)$ there exists $c \in \Sigma$ such that $\theta(a) = c$. By the same argument there exists $b \in \Sigma$ such that $\theta(c) = b$. Since, $\theta^3 = I$, $\theta(b) = a$.

Assume that L is context-free. Let n be the constant defined by pumping lemma for context-free languages. Let $w_1 = c^{n+1}a^{n+1}b^{n+1}c^{n+1}$ which is clearly a θ-bordered word. By the pumping lemma, there is a decomposition $w_1 = \alpha xvy\beta$ such that $|xvy| \leq n$, $|xy| \geq 1$ and for all $i \geq 0$,

$w_i = \alpha x^i v y^i \beta \in L$. Since w_i begins with c for any $i \geq 0$, every θ-border z of w_i has the property $z = cu$ for some $u \in \Sigma^+$.

Case 1: xvy is a subword of $c^{n+1}a^{n+1}$ of w_1. In this case, since w_i has the suffix c^{n+1}, $\theta(z) \in b\Sigma^* c^{n+1}$. ($\theta(z)$ cannot begin with c or a because in those cases z would begin with a or b respectively, which is not possible.) Hence, $z \in c\Sigma^* a^{n+1}$. If neither x nor y contains any as which means xvy is a subword of c^{n+1} of w_1, we get $w_i = c^m a^{n+1} b^{n+1} c^{n+1}$, for $i \geq 2$ and $m > n+1$. But then, $z = c^m a^{n+1}$ which further imply that $\theta(z_i) = b^m c^{n+1}$, which is a contradiction since w_i does not contain m consecutive bs. Hence, either x or y must include at least one letter a. But this would imply that w_0 has at most n letters a which is a contradiction since it has $z = cua^{n+1}$ for some $u \in \Sigma^*$ as its θ-border.

Case 2: xvy is a subword of $a^{n+1}b^{n+1}$ of w_1. In this case, since w_i has the suffix c^{n+1}, $\theta(z) \in b\Sigma^* c^{n+1}$. Hence, $z \in c\Sigma^* a^{n+1}$. If neither x nor y contains any bs which means xvy is a subword of a^{n+1} of w_1, we get $w_0 = c^{n+1}a^k b^{n+1} c^{n+1}$ for $k \leq n$, which means that w_0 has at most n letters a which contradicts the fact that w_0 has $z_0 = cua^{n+1}$ for some $u \in \Sigma^*$ as its θ-border. Hence, either x or y must include at least one letter b. But then, $w_0 = c^{n+1}a^l b^k c^{n+1} \notin L$ for $k < n+1$ and $l \leq n+1$ since $k < n+1$, $c^{n+1}a^l$ cannot be a θ-border of w_0. Hence we have reached a contradiction

Case 3: xvy is a subword of $b^{n+1}c^{n+1}$ of w_1. In this case, since w_i has prefix c^{n+1}, $z \in c^{n+1}\Sigma^* a$. (z cannot end with c or b because in those cases $\theta(z)$ would end with b or a which is not possible.) Hence, $\theta(z) \in b^{n+1}\Sigma^* c$. If neither x nor y contains any cs which means xvy is a subword of b^{n+1} of w_1, we get $w_0 = c^{n+1}a^{n+1}b^{k'}c^{n+1}$ for $k' \leq n$, which means $w_0 \notin L$ which is a contradiction, because, $c^{n+1}a^{n+1}$ cannot be a θ-border of w_i due to the fact that $k' \leq n$. Hence, either x or y must include at least one letter c. But then, $w_i = c^{n+1}a^{n+1}b^j c^{j'} \notin L$ for $j \geq n+1$, $j' > n+1$ and $i \geq 2$ since $c^{n+1}a^{n+1}$ cannot be a θ-border of w_i because $j' > n+1$. Hence, we have reached a contradiction.

Lastly, since $|xvy| \leq n$, we have that xvy can also not be a subword of $c^{n+1}a^{n+1}b^{n+1}$ or $a^{n+1}b^{n+1}c^{n+1}$.

Since all the cases lead to a contradiction, our assumption was incorrect and hence L is not context-free. □

In general, the set of all θ-bordered words, B_θ, is not context-free for any morphism θ such that there exists $n \geq 2$ with $\theta^n(a) = a$ for all $a \in \Sigma$. The idea of the proof, [24], is to consider such a morphism and a letter $a \in \Sigma$ such that $\theta^n(a) = a$ for $n > 1$ and $\theta^i(a) \neq a$ for all $0 < i < n$. Now, consider the set $S = B_\theta \cap a^+\theta(a)^+\theta^2(a)^+ \dots \theta^{n-1}(a)^+a^+$. If $w \in S$, then $w = a^{i_0}(\theta(a))^{i_1}(\theta^2(a))^{i_2} \dots (\theta^{n-1}(a))^{i_{n-1}}(\theta^n(a))^{i_n}$ where $i_m \geq 1$ for $1 \leq m \leq n$. Let $v \in \Sigma^+$ be such that $v <_d^\theta w$. Thus,

$$\theta(v) = (\theta(a))^j(\theta^2(a))^{i_2} \dots (\theta^{n-1}(a))^{i_{n-1}}(\theta^n(a))^{i_n}$$

for $j \leq i_1$. Also, $v = a^{i_0}(\theta(a))^{i_1}(\theta^2(a))^{i_2} \dots (\theta^{n-1}(a))^k$ for $k \leq i_{n-1}$. This implies

$$\theta(v) = (\theta(a))^{i_0}(\theta^2(a))^{i_1} \dots (\theta^{n-1}(a))^{i_{n-2}}(\theta^n(a))^k.$$

Thus, the comparison of expressions for $\theta(v)$ yields, $i_0 = j \leq i_1$, $i_1 = i_2 = i_3 = \dots = i_{n-2} = i_{n-1}$ and $i_n = k \leq i_{n-1}$. Hence,

$$S = \{a^{i_0}(\theta(a))^l(\theta^2(a))^l \dots (\theta^{n-1}(a))^l a^{i_n} | i_0, i_n \leq l\}$$

which is clearly not a context-free language. Thus, if we consider any word from the set S, it will clearly be a θ-bordered word and hence the set B_θ is not context-free.

5.7 Conclusions

This paper continues the exploration of properties of θ-bordered (pseudo-bordered) words and θ-unbordered words for the case where θ is a morphic involution. We prove, under certain conditions, the disjunctivity of the language of words with exactly i θ-borders, for all $i \geq 1$, and also that the set $D_\theta^i(1) \setminus D(i)$ of the language of words which consist of catenations of

i θ-unbordered words, but which do not have exactly i borders, is disjunctive for all $i \geq 2$. Further directions of research include generalizations of these and similar results for morphism or antimorphisms θ with the property that θ^n equals the identity function on Σ for an arbitrary $n \geq 3$.

Acknowledgements We thank Steffen Kopecki for valuable suggestions, in particular those that lead to shortening the proof of Proposition 5.6.6, and Sepinoud Azimi for discussions.

Bibliography

[1] A. Blondin Massé, S. Gaboury, and S. Hallé. Pseudoperiodic words. In H.-C. Yen and O. Ibarra, editors, *Developments in Language Theory*, volume 7410 of *Lecture Notes in Computer Science*, pages 308–319. Springer Berlin Heidelberg, 2012.

[2] A. Carpi and A. de Luca. Periodic-like words, periodicity, and boxes. *Acta Informatica*, 37(8):597–618, 2001.

[3] D.-J. Cho, Y.-S. Han, and S.-K. Ko. Decidability of involution hypercodes. *Theoretical Computer Science*, 550(0):90–99, 2014.

[4] S. Constantinescu and L. Ilie. Fine and Wilf's theorem for abelian periods. *Bulletin of the EATCS*, 89:167–170, 2006.

[5] M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on Strings*. Cambridge University Press, 2007.

[6] M. Crochemore and W. Rytter. *Jewels of Stringology*. World Scientific, 2002.

[7] L. J. Cummings and W. F. Smyth. Weak repetitions in strings. *J. Combinatorial Mathematics and Combinatorial Computing*, 24:33–48, 1997.

[8] E. Czeizler, L. Kari, and S. Seki. On a special class of primitive words. *Theoretical Computer Science*, 411:617–630, 2010.

[9] A. de Luca and A. de Luca. Pseudopalindrome closure operators in free monoids. *Theoretical Computer Science*, 362(13):282–300, 2006.

[10] P. Gawrychowski, F. Manea, R. Mercaş, D. Nowotka, and C. Tiseanu. Finding pseudo-repetitions. *Leibniz International Proceedings in Informatics*, 20:257–268, 2013.

[11] P. Gawrychowski, F. Manea, and D. Nowotka. Discovering hidden repetitions in words. In P. Bonizzoni, V. Brattka, and B. Löwe, editors, *The Nature of Computation. Logic, Algorithms, Applications*, volume 7921 of *Lecture Notes in Computer Science*, pages 210–219. Springer Berlin Heidelberg, 2013.

[12] P. Gawrychowski, F. Manea, and D. Nowotka. Testing generalised freeness of words. In E. W. Mayr and N. Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, volume 25, pages 337–349, 2014.

[13] J. E. Hopcroft and J. D. Ullman. *Formal Languages and their Relation to Automata*. Addison-Wesley Longman Inc., 1969.

[14] S. Hsu, M. Ito, and H. Shyr. Some properties of overlapping order and related languages. *Soochow Journal of Mathematics*, 15(1):29–45, 1989.

[15] C. Huang, P.-C. Hsiao, and C. J. Liau. A note of involutively bordered words. *Journal of Information and Optimization Sciences*, 31(2):371–386, 2010.

[16] S. Hussini, L. Kari, and S. Konstantinidis. Coding properties of DNA languages. In N. Jonoska and N. Seeman, editors, *Proc. of DNA7*, volume 2340 of *Lecture Notes in Computer Science*, pages 57–69. Springer, 2002.

[17] N. Jonoska, D. Kephart, and K. Mahalingam. Generating DNA code words. In *GECCO Late Breaking Papers*, pages 240–246, 2002.

[18] L. Kari and M. S. Kulkarni. Pseudo-identities and bordered words. In G. Păun, G. Rozenberg, and A. Salomaa, editors, *Discrete Mathematics and Computer Science*, pages 207–222. Editura Academiei Române, 2014.

[19] L. Kari and K. Mahalingam. Involutively bordered words. *International Journal of Foundations of Computer Science*, 18(05):1089–1106, 2007.

[20] L. Kari and K. Mahalingam. Watson-Crick conjugate and commutative words. In M. H. Garzon and H. Yan, editors, *Proc. of DNA13*, volume 4848 of *Lecture Notes in Computer Science*, pages 273–283. Springer-Verlag, 2008.

[21] L. Kari and K. Mahalingam. Watson-Crick palindromes in DNA computing. *Natural computing*, 9(2):297–316, June 2010.

[22] L. Kari and S. Seki. On pseudoknot-bordered words and their properties. *Journal of Computer and System Sciences*, 75:113–121, 2009.

[23] L. Kari and S. Seki. An improved bound for an extension of Fine and Wilf's theorem and its optimality. *Fundamenta Informaticae*, 101:215–236, 2010.

[24] S. Kopecki. Personal communication.

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

[26] H. Shyr and G. Thierrin. Disjunctive languages and codes. In M. Karpiński, editor, *Fundamentals of Computation Theory*, volume 56 of *Lecture Notes in Computer Science*, pages 171–176. Springer Berlin Heidelberg, 1977.

[27] H. J. Shyr. *Free Monoids and Languages*. Department of Mathematics, Soochow University, Taipei, Taiwan, 1979.

[28] W. Trappe and L. C. Washington. *Introduction to Cryptography with Coding Theory*. Pearson Education India, 2006.

[29] J. D. Watson and F. H. Crick. Molecular structure of nucleic acids. *Nature*, 171(4356):737–738, 1953.

[30] F.-R. Wong. Algebraic Properties of d-Primitive Words. Master's thesis, Chuang-Yuan Christian University, Chuang Li, Taiwan, 1994.

[31] S. Yu. d-minimal languages. *Discrete Applied Mathematics*, 89(13):243–262, 1998.

[32] S.-S. Yu. *Languages and Codes*. Tsang Hai Book Publishing Co., 2005.

[33] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

Chapter 6

Conclusion and Discussion

In this thesis, we continue the study initiated in [1] and [5] on the generalization of two fundamental notions in combinatorics on words, namely periodicity and borderedness, for various pseudo-identity functions.

The operation of catenation is known to generate the power (repetition) of a word. However, the catenation operation cannot generate pseudo-powers (pseudo-repetition) of a word, where the identity function is replaced by a pseudo-identity function. In Chapter 3, [2], we propose and investigate the binary word operation of θ-catenation that generates θ-powers (pseudo-powers) of a word, for morphic or antimorphic involutions θ. We study the connection of the operation of θ-catenation with the previously defined notions of θ-primitive and θ-periodic words, and explore closure properties of various language families under the operation of θ-catenation. In particular, we find the right and left inverses of θ-catenation, characterize, under certain conditions, some θ-primitive words which result by an application of θ-catenation between two words, and show that the families of regular, context-free and context-sensitive languages are closed under the operation of θ-catenation.

In related research project, which attempts to generalize the notion of identity, we extend the notion of pseudo-bordered words for functions beyond identity and involution (Chapter 4, [3]). In particular, we study properties of θ-bordered (pseudo-bordered) and θ-unbordered

(pseudo-unbordered) words for functions θ such that either θ is an (anti)morphism with the property that $\theta^n = I$, for $n \geq 2$, or θ is any literal (anti)morphism. Some of the obtained properties include necessary and sufficient condition for a word to be θ-bordered, and the transitivity of the relation $<_d^\theta$ for literal (anti)morphisms θ. We also proved that the set of all θ-bordered words is not context-free for morphisms θ such that θ^3 is an identity function on Σ, and with the property that $\theta(a) \neq a$ for all $a \in \Sigma$.

The relation between disjunctive and regular languages is that disjunctive languages are not regular. This and the fact that the set of all words with exactly i borders, $D(i)$, is disjunctive for all $i \geq 1$, motivated the study of disjunctivity of the set of all θ-(un)bordered words and some other related languages for morphic involutions θ (Chapter 5, [4]). We show that the set of all words with exactly i θ-borders, $D_\theta(i)$, is disjunctive under certain conditions for all $i \geq 1$. In an attempt to establish the relationship between $D_\theta(i)$ and $D(i)$, we prove that the set $D_\theta^i(1) \backslash D(i)$, is disjunctive, under certain conditions, for all $i \geq 2$.

As future work, we are interested primarily in investigating the disjunctivity properties of the set of all pseudo-bordered words for functions which are further generalizations of involution functions, as well as studying binary word operations which generate other pseudo-powers. Another notion that can be generalized by considering pseudo-identities is the notion of Fibonacci words and languages, which is of great mathematical interest. Also, it would be interesting to model other secondary and complex structures that are formed in DNA as well as RNA molecules.

Bibliography

[1] E. Czeizler, L. Kari, and S. Seki. On a special class of primitive words. *Theoretical Computer Science*, 411:617 – 630, 2010.

[2] L. Kari and M. S. Kulkarni. Generating the pseudo-powers of a word. *Journal of Automata, Languages and Combinatorics*, 19(1–4):157–171, 2014.

[3] L. Kari and M. S. Kulkarni. Pseudo-identities and bordered words. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Discrete Mathematics and Computer Science*, pages 207–222. Editura Academiei Române, 2014.

[4] L. Kari and M. S. Kulkarni. Disjunctivity and other properties of sets of pseudo-bordered words. Submitted, 2015.

[5] L. Kari and K. Mahalingam. Involutively bordered words. *International Journal of Foundations of Computer Science*, 18(05):1089–1106, 2007.

Chapter 7

Addendum

Since this thesis is formated as integrated-article, the content of all the technical chapters should be exactly the same as those of published article and no change is allowed. Hence, we list all the modifications according to the comments provided by the thesis examiners in this chapter.

Implementation of the comments

page 44, line 1: "properties in combinatorics on words" → "properties in the combinatorics on words".

page 49, line 7: "left-inclusive ." → "left-inclusive."

page 51, line 6: "The following Lemma and its Corollary" → "The following lemma and its corollary".

page 51, line 12: "Firstly" → "First", "Secondly" → "Second", "Thirdly" → "Third".

page 52, line 5: "We will prove this by induction on n." → "We prove this by induction on n."

page 52, line 16: "in the same way the operation of catenation" → "in the same was as the operation of catenation".

page 52, line 17: "the operation of θ-*catenation* is the one that generates the θ-powers of a word." → "the operation of θ-*catenation* generates the θ-powers of a word.".

page 54, line 15: If $\circ$ is the operation of catenation then any prefix-free language will be

$\circ$-free.

page 62, line 20: "its" → "it's".

page 66, line 1: "properties in combinatorics on words and formal language theory" → "properties in the combinatorics on words and in formal language theory".

page 67, line 3: "several new notion in combinatorics on words" → "several notions in the combinatorics on words".

page 67, line 6: "modeled" → "modelled".

page 74, last line: "leads" → "lead".

page 75, line 18: "leads" → "lead".

page 79, line 17: "leads" → "lead".

page 82, line 8: "leads" → "lead".

page 82, line 14: "leads" → "lead".

page 83, line 16: "leads" → "lead".

Appendix A

Appendices

Copyright releases

- The contents of Chapter 3 were published by Otto-von-Guericke University, Magdeburg (Germany) in "Journal of Automata, Languages and Combinatorics". I requested them to grant me a permission to reuse the article for my thesis and I got the following kind permission from them.

  From: jalc <jalc@iws.cs.uni-magdeburg.de>
  Sent: Friday, July 17, 2015 1:05 PM
  To: Manasi Kulkarni
  Subject: Re: Permission to resue my paper for my thesis

  Dear Mr. Kulkarni,

  I ask you to excuse my delay. By some illness I did not look into this e-mail-account since some days. I hope that the information comes in time.

  Herewith I give the permission to reuse the paper in your thesis.

Enclosed I send you published version as pdf and LaTeX file.

Best regards

J"urgen Dassow
Editor-in-Chief

---Original message---

Am 2015-07-06 19:17, schrieb Manasi Kulkarni:
Dear Sir/Ma'am,

I am a PhD student at the Department of Computer Science at the University of Western Ontario.

My research article on "Generating pseudo-powers of a word" co-authored with Professor Lila Kari appeared in the volume 19 of 2014. I would like to include the article in my thesis which needs to be submitted by July 20, 2015. Hence, I would like to request you to grant the permission to reuse the article in my PhD thesis.

Can I get a copy of the published version in order to compare with the version that I have? As I read from your web site, "Authors of published papers will be provided with 20 free reprints".

I appreciate your time and help.

Thank you.

--

Regards,

Manasi Kulkarni

http://www.csd.uwo.ca/~mkulkar3/

- The contents of Chapter 4 were published by Editura Academiei Române in "Discrete Mathematics and Computer Science". I requested the editor of the journal to grant me a permission to reuse the article for my thesis and I got the following kind permission from him.

From: gpaun@us.es <gpaun@us.es>

Sent: Tuesday, July 21, 2015 2:57 PM

To: Manasi Kulkarni

Subject: Re: Permission to resuse my paper for my thesis

Dear Manasi,

There is no problem to include the paper you mention in your PhD Thesis. If you need a more formal permission letter, please let me know.

Best regards,

Gh. Paun, editor of the mentioned volume

---Original message---

El 20/07/2015 17:45, Manasi Kulkarni escribi:

Dear Dr. Gheorghe Paun,

I am a Ph.D. student in the Department of Computer Science at the University of Western Ontario.

My research article on "Pseudo-identities and bordered words" co-authored with Dr. Lila Kari appeared in a special volume of "Discrete Mathematics and Computer Science" published by Editura Academiei Romane. I would like to include the article in my thesis. Hence, I would like to request you to grant the permission to reuse the article in my PhD thesis.

I appreciate your time and help.

Thank you.

--Regards,
Manasi Kulkarni
http://www.csd.uwo.ca/~mkulkar3/

Curriculum Vitae

Name:	Manasi Kulkarni
Post-Secondary Education and Degrees:	University of Mumbai Mumbai, India 2005-2008 B.Sc. (Mathematics) Indian Institute of Technology Madras Chennai, India 2008 - 2010 M.Sc. (Mathematics) University of Western Ontario London, ON, Canada 2011 - 2015 Ph.D. (Computer Science)
Honours and Awards:	Graduate Teaching and Research Assistant 2011-2015 Western Graduate Research Scholarship 2011-2015
Related Work Experience:	Lecturer Rajiv Gandhi Institute of Knowledge Technologies Kadapa, India 2010 - 2011 Graduate Teaching Assistant University of Western Ontario London, ON, Canada 2011-2015

Publications:

1. L. Kari, M.S. Kulkarni. Generating the pseudo-powers of a word. *Journal of Automata, Languages and Combinatorics*, 19(2014), 1-4, 157-171.

2. L. Kari, M.S. Kulkarni. Pseudo-identities and bordered words. In G. Păun, G. Rozenberg, A. Salomaa editors, *Discrete Mathematics and Computer Science*, Editura Academiei Romane, 2014, 207-222.

3. L. Kari, M.S. Kulkarni. Disjunctivity and other properties of sets of pseudo-bordered words. Submitted, *Acta Informatica*.