CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Nuclear Sciences and Physical Engineering

Department of Mathematics
Theoretical Informatics Group

MASTER'S THESIS

Symmetries in Factor Languages and Palindromic Richness

Symetrie faktorových jazyků a bohatost na palindromy

Author: **BSc. Viola Hadjikyriacou**
Supervisor: **Prof. Ing. Edita Pelantová, CSc.**
Academic year: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hadjikyriacou** Jméno: **Viola** Osobní číslo: **503752**

Fakulta/ústav: **Fakulta jaderná a fyzikálně inženýrská**

Zadávající katedra/ústav: **Katedra matematiky**

Studijní program: **Matematické inženýrství**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Symetrie faktorových jazyků a bohatost na palindromy

Název diplomové práce anglicky:

Symmetries in factor languages and palindromic richness

Pokyny pro vypracování:

1. Nastudujte vlastnosti zobecněných Thueových-Morseových posloupností.
2. Pokuste se definovat co nejširší třídu morfizmů, jejichž pevné body mají jazyk uzavřený na zadaný involutivní antimorfizmus T. Věnujte se také otázce T-palindromicity pevných bodů.
3. Předchozí úlohu rozšiřte na invariantnost vzhledem ke konečné grupě G generované involutivními antimorfizmy.
4. Seznamte se s definicí klasického pojmu „slovo bohaté na palindromy" i s jeho zobecněním na G-bohatá slova pro zadanou grupu G.
5. Pokuste se nalézt příklady G-bohatých posloupností nad abecedou DNA-řetězců v případě, kdy grupa G je generovaná zrcadlovým obrazem a DNA-antimorfizmem.

Seznam doporučené literatury:

1. Š. Starosta, Generalized Thue–Morse words and palindromic richness. Kybernetika 48 no. 3, 2012, 361-370.
2. L. Kari, K. Mahalingam, Watson–Crick palindromes in DNA computing. Natural Computing 9, 2010, 297-316.
3. A. Glen, J. Justin, S. Widmer, L. Q. Zamboni, Palindromic richness. European Journal of Combinatorics 30, 2009, 510-531.
4. E. Pelantová, Š. Starosta, Palindromic richness for languages invariant under more symmetries. Theoretical Computer Science 518, 2014, 42-63.

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. Ing. Edita Pelantová, CSc. katedra matematiky FJFI

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **06.04.2023** Termín odevzdání diplomové práce: **08.01.2024**

Platnost zadání diplomové práce: **06.04.2025**

prof. Ing. Edita Pelantová, CSc.	prof. Ing. Zuzana Masáková, Ph.D.	doc. Ing. Václav Čuba, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomantka bere na vědomí, že je povinna vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

19.4.2023

Datum převzetí zadání

Podpis studentky

Acknowledgement

I would like to express my profound gratitude to my supervisor Prof. Pelantová for her enthusiasm, countless discussions and invaluable advice. Thanks to her kind and attentive attitude it was a great joy to be working on this topic. I also wish to thank to my family for their support throughout my studies.

Author's declaration

I declare that this work has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgement, the work presented is entirely my own.

27.4.2023	Hadjikyriacou
Date	BSc. Viola Hadjikyriacou

Název práce:
Symetrie faktorových jazyků a bohatost na palindromy

Autor: BSc. Viola Hadjikyriacou

Studijní program: Matematické inženýrství

Druh práce: Diplomová práce

Vedoucí práce: Prof. Ing. Edita Pelantová, CSc., Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, ČVUT

Abstrakt: Cílem této práce je studium nekonečných palindromických slov. Často uvažujeme abecedu $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ a tato volba je motivována strukturou DNA. Nejprve uvedeme potřebnou teorii, v níž je pro nás klíčový pojem obecného palindromu. Také odvodíme nové výsledky týkající se rovnic na slovech, kde se vyskytují palindromy. Jedním z předmětů našeho zkoumání je generování palindromických slov. Shrneme známé výsledky ohledně tohoto problému a následně zformulujeme obecnou teorii pro H-palindromická a G-palindromická slova. Dále představíme bohatost na palindromy jak v klasickém, tak zobecněném smyslu. Naší druhou úlohou je hledání G-bohatých slov. Popíšeme algoritmus, který využíváme na testování G-bohatosti slov a uvedeme několik tříd morfizmů, které pravděpodobně generují G-bohatá slova.

Klíčová slova: HKS domněnka, kombinatorika na slovech, pevný bod morfizmu, Watson-Crick palindrom, zobecněná bohatost na palindromy, zobecněný palindrom

Title:
Symmetries in Factor Languages and Palindromic Richness

Author: BSc. Viola Hadjikyriacou

Abstract: The aim of this work is to study infinite palindromic words. We frequently consider the alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and this choice is motivated by the structure of DNA. We summarize some essential theory, where the central concept for us is a general palindrome. We also derive some additional results concerning word equations with palindromes. Our first objective is to generate palindromic words. We present known results about this problem and then develop a more general theory for H-palindromic and G-palindromic words. Next, we introduce palindromic richness in the classical sense as well as its generalizations. Our second objective is to find examples of infinite G-rich words. We describe an algorithm that we use to test G-richness and give several classes of morphisms that likely generate G-rich words.

Key words: combinatorics on words, fixed point of morphism, generalized palindrome, generalized palindromic richness, HKS conjecture, Watson-Crick palindrome

Contents

Chapter 1

Introduction

In this work, we investigate some properties of infinite words concerning their symmetric factors. Several results are linked to a specific choice of the alphabet and we often consider infinite sequences of letters from the alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. This choice is motivated by the structure and significance of DNA molecules and we discuss this in more details in Chapter 2.

In Chapter 3, we set up the notation and terminology and we also give an overview of some important known results of combinatorics on words. We cover finite and infinite words and their properties, such as factor complexity, for which we prove some known results using the so-called Rauzy graphs. Then we move to morphisms, where we introduce the concept of conjugation, showing that it is an equivalence relation on the set of morphisms, and present definition of the incidence matrix of a morphism. Next, we show how to generate infinite words by morphisms and summarize results regarding such infinite words. Subsequently, we employ antimorphisms to give general definition of an H-palindrome and an H-palindromic word, where H denotes an arbitrary involutive antimorphism. In the case of the alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ we have a D-palindrome and a D-palindromic word. By an R-palindrome and an E-palindrome we denote what is commonly known as a palindrome and an antipalindrome, respectively. Lastly, we discuss finite groups of morphisms and antimorphisms G and consider another generalization of a palindrome called a G-palindrome.

Chapter 4 consists of our derivations of several results concerning equations on words that contain R-palindromes and D-palindromes. We use one of these results in Chapter 5.

First part of our investigation focuses on palindromic words. The goal is to generate H-palindromic and G-palindromic words and our method of doing so is described in Chapter 5. After reviewing some known results about the class of morphisms $\mathcal{P}$, which is used to generate R-palindromic words, and briefly summarizing results regarding E-palindromic words, we investigate D-palindromic words, defining new class of morphisms $\mathcal{D}$. We generalize this approach to any involutive antimorphism H by defining a corresponding class $\mathcal{H}$. We show that morphisms from class $\mathcal{H}$ that satisfy certain graph condition generate H-palindromic words. Next, we address the question of generating G-palindromic words. We define a class of morphisms $\mathcal{G}$ by specifying certain relations such morphisms have to satisfy. Then we derive the form of such morphisms for several concrete groups G. Again, we show that under certain conditions, fixed points of morphisms from class $\mathcal{G}$ are G-palindromic.

Second problem we pursue is finding examples of G-rich words. The topic of palindromic richness, both in the classical and a general sense, is covered in Chapter 6. The generalization we are interested in is with respect to a group of morphisms and antimorphisms G. We consider two specific groups and for each one design an algorithm for deciding whether a given finite word is G-rich or not. Then we employ this algorithm to test various fixed points of morphisms from class $\mathcal{G}$. We present several classes of morphisms that seem to generate G-rich words. It still requires formal proof to confirm that these words are indeed G-rich, since we can be certain about the result of the computer test only when the outcome is that given fixed point is not G-rich.

Chapter 2

DNA computing

One part of this work investigates words over the alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. We are especially interested in D-palindromes, words that are invariant under the so-called DNA map D, which is motivated by the structure of DNA. In literature, they are known as Watson-Crick palindromes. In this chapter, we look more closely at DNA, summarizing important facts about this remarkable molecule, and introducing the field of DNA computing, where results about D-palindromes could potentially be used.

2.1 Structure of DNA

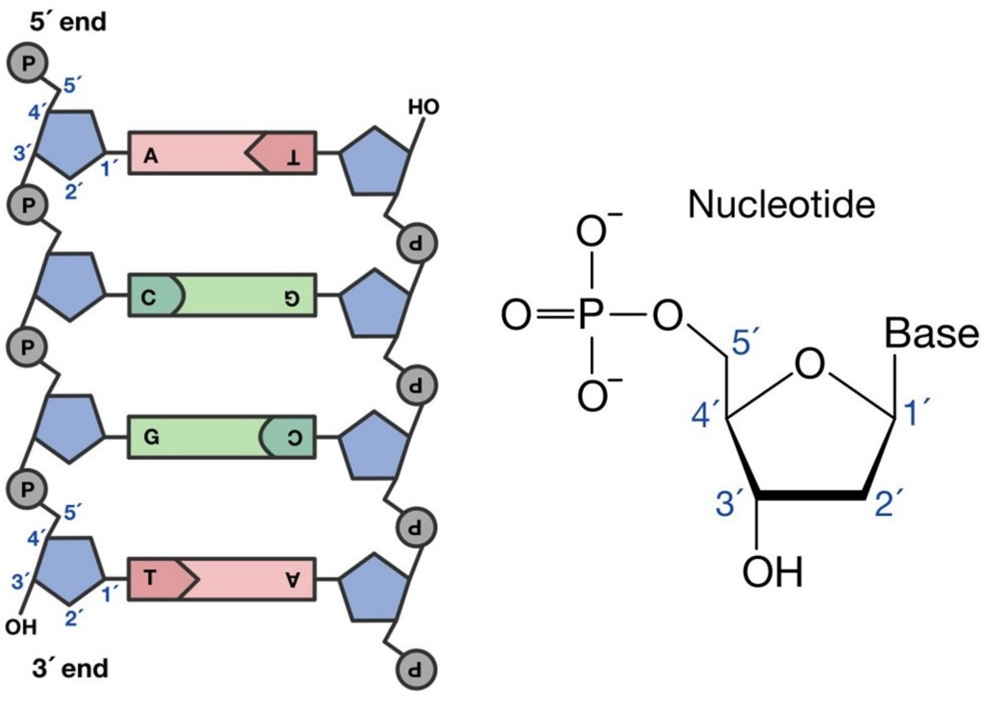


Figure 2.1: Diagram of the structure of a DNA molecule.

DNA (deoxyribonucleic acid) is found in cells of living organisms and its function is to encode information about structure of proteins that are produced. This information, called the genetic code, determines characteristic features of the organism and is passed on from generation to generation [37, 6]. One DNA molecule typically consists of two strands, polymer chains of nucleotides, forming a double helix, however, it can also be in the form of a single strand. One nucleotide contains a deoxyribose sugar, a phosphate group and one of 4 different bases, adenine (A), cytosine (C), guanine (G) or thymine (T).

Hence, a DNA strand can be represented as a sequence of letters from the set $\{A, C, G, T\}$ corresponding to the sequence of nucleotide bases in the molecule. Each strand has two chemically distinct ends, a $3'$ end and a $5'$ end, which gives it a natural orientation. This notation comes from the ordering of carbons in the sugar, the $3'$ end has a hydroxyl group on the $3'$ carbon and the $5'$ end has a phosphate group attached to the $5'$ carbon. This is shown in Figure 2.1. Standard convention is to write DNA sequences in a $5'$ to $3'$ direction. The two strands of one molecule have opposite direction and are held together by bonds between nucleotides, which are paired up in a specific way, A with T and C with G. Two bases that can bind are called Watson-Crick complementary. This complementarity enables DNA to replicate [41, 26].

In this work, we use the so-called DNA map, which is formally defined later. Let us now illustrate the connection between this map and DNA. The DNA map sends a string of letters from the set $\{A, C, G, T\}$ to the string that is created by reversing the original string and replacing each letter by its Watson-Crick complement. So, for example, the DNA map sends CTTGA to TCAAG. This corresponds to sending a DNA sequence to its Watson-Crick complementary sequence which would bind with it, as, for example, $5' - \mathrm{CTTGA} - 3'$ binds with $3' - \mathrm{GAACT} - 5'$, which also writes as $5' - \mathrm{TCAAG} - 3'$. If some string is sent by the DNA map to itself, we say that it is a *D*-palindrome. In literature, it is known as a Watson-Crick palindrome. An example of a *D*-palindrome is the sequence ACCGGT.

2.1.1 Hairpin structure in DNA

In DNA computing, where DNA molecules are used to perform calculations, the notion of Watson-Crick complementarity plays an important role, as it allows scientists to design and use DNA molecules in such a way that desired bindings occur. This can be observed below in section 2.4, where two specific DNA algorithms are described. However, care must be taken to ensure that the DNA molecules will not interact in undesirable ways. An example of such situation is when one part of a DNA strand is complementary to a different part of the same DNA strand and hence the DNA strand binds to itself. This creates a secondary structure called a hairpin [26].

```
                              G
                            C   C
  G C G A A C T G A G G T A       A
    | | | | | | | | | | | |         A
C A G C T T G A C T C C A T A     C
                            G T
```

Figure 2.2: Illustration of a hairpin structure in a DNA molecule.

Some DNA algorithms make use of the hairpin formation, for example in [39], an algorithm based on hairpin formation was proposed to solve the SAT problem. This problem, together with a different algorithm solving it, is described in section 2.4.2. However, in most DNA algorithms, hairpin formation is not desirable and it would interfere with the computation. Therefore, the information-encoding DNA molecules should be designed such that hairpin formation and other unwanted bindings are avoided. There are several approaches to optimal information encoding for DNA computing and their overview can be found in [26]. One of those approaches is formal language theoretical approach introduced in [24], where languages (i.e., sets of sequences of letters) with desired properties are studied. This approach involves, among other things, study of the DNA map. The concept of a D-palindrome is also relevant for DNA encoding and some aspects of this, as well as theoretical properties of D-palindromes, are discussed in [26].

2.2 Operations with DNA

Study and manipulation of DNA molecules has many various applications. Advances in molecular biotechnology enable scientists to perform a variety of operations on DNA. Here, following [37], we describe some operations that can be carried out in a lab and that are also used in the field of DNA computing.

- Annealing and melting: Annealing is the process of pairing two complementary single strands into one double-stranded molecule, whereas melting is the reverse operation of separating the two strands of one molecule. This can be done by setting suitable conditions of temperature, pH, etc.
- Synthesis: It is possible to synthesise a desired DNA strand up to a certain length.
- Extraction: Having a test-tube containing many DNA strands, this operation extracts all the strands that contain a specific subsequence s and separates them from the rest of the strands.
- Amplification: DNA strands in a test-tube can be amplified by duplicating all the strands.
- Polymerase Chain Reaction (PCR): This technique amplifies DNA strands rapidly and also allows to extract all the strands that begin (or end) with a given subsequence.
- Separation by length: A specific technique called gel electrophoresis is used to sort and separate DNA strands by length.
- Cutting: DNA molecules can be manipulated by other molecules called enzymes. The so called restriction enzymes can be employed to cut double-stranded DNA molecules at locations where specific subsequences appear.
- Ligation: Two strands of DNA can be joined end to end to form a single strand by an enzyme called ligase.

2.3 Introduction to DNA computing

In the last decades, there has been great effort to use DNA molecules to perform computations. The idea of computations at molecular level was first introduced by Richard Feynman in 1959 in his talk 'There's Plenty of Room at the Bottom' [19], but it was not realized in practice until 1994, when Leonard Adleman [1] performed the first molecular-level computation using DNA to solve an instance of the Hamiltonian path problem. We describe this experiment in section 2.4.1. Since then, several other mathematical problems have been solved experimentally and progress in theoretical design of DNA computers has been made [44].

Scientist see enormous potential in DNA computing, as the operations can be done all in parallel and DNA molecules offer excellent information density of approximately 1 bit per nm^3, which is several orders of magnitude more dense than the memory currently used in computers. This means that if a DNA-based computer is build, it could potentially utilize more processors than all silicon-based computers in the world combined and significantly outperform existing supercomputers in speed, energy efficiency and economic information storing [17, 44].

However, for practical success of DNA computers a lot of research still needs to be done. For the time being, the difficulties of building and using a DNA computer outweigh its benefits. Most designs of DNA computers solve only a specific problem, and hence cannot be used universally. Therefore, rather than replacing conventional computers, it seems more feasible to employ DNA computers for other types of problems that are impossible to solve effectively on conventional computers, but DNA computers are naturally good at solving [44]. Examples of such problems can be found in the NP class of problems, more specifically in NP-complete problems, which we focus on in the next section.

One discipline where DNA computers have huge advantage over conventional computers is interacting with biochemical environment, even within a living organism, by the means of other biological molecules. This ability of DNA could potentially be used for example to construct a molecular computer that operates as an autonomous "doctor" within a cell. It would register disease indicators, process them according to preprogrammed medical knowledge, and in the case of a positive diagnosis output a signal or an appropriate drug. Based on this vision, in 2004 Benenson et al. [11] programmed a DNA-based finite automaton, which was designed and implemented by Benenson et al. [10] in 2003, to identify and analyse certain molecular indicators of two specific types of cancer and to release a short DNA molecule functioning as a drug on positive diagnosis. They demonstrated that this molecular computer can operate in a test tube, however, applying such a device inside cells, let alone living organisms, comes with many new challenges [40, 25].

Other applications of DNA computing can be found in cryptography. Many DNA-based security schemes have been proposed, however, they all have some disadvantages. DNA cryptography is still in an early stage, nevertheless, DNA computing is a popular approach to improving data security. It is also used in fields such as big data, cloud computing and data storage [34].

2.4 NP-complete problems

There are several established classes of problems based on their time complexity. This section describing them is based on [7], where formal definitions can be found. Firstly, the class P consists of all decision problems that can be solved by a deterministic Turing machine (or by a conventional computer) in polynomial time. Turing machine is an abstract computing machine that enables us to formalize the intuitive notion of a calculation. Informally described, it consists of one input tape with read-only head; a finite set of work tapes and one output tape, where symbols from a given alphabet are written or read by tape heads one cell at a time; finite set of states with one starting state and one halting state; and set of rules that for each combination of a state and symbols read by tape heads give an instruction on what the tape heads do and to which state the machine transitions. A calculation starts in the starting state with the input on the input tape and all tape heads pointing at the first cell of each tape, and it ends when the halting state is reached with output on the output tape. Given a function f on strings and a function $T\colon \mathbb{N} \to \mathbb{N}$, we say that a Turing machine M computes f in T-time if for every possible input x the Turing machine halts with output $f(x)$ after at most $T(|x|)$ steps, where $|x|$ denotes the length of x.

Another variant of the Turing machine is the so-called non-deterministic Turing machine (NDTM), which differs from the Turing machine described above by having two sets of rules to choose from in every step of the computation. Hence the computation itself is non-deterministic, as it depends on the sequence of choices that are made. In addition, the machine has a special accepting state. We say that a NDTM M accepts an input x if there exists a sequence of choices leading to the accepting state. Given a function $T\colon \mathbb{N} \to \mathbb{N}$, M is said to run in T-time if for every possible input and every sequence of choices, M reaches either the halting state or the accepting state within $T(|x|)$ steps. Given a language L, i.e., a subset of all finite strings over some alphabet, we say that M decides L in T-time if it runs in T-time and M accepts an input x if and only if $x \in L$. Analogously to the class P, class NP consists of languages that can be decided by a non-deterministic Turing machine in polynomial time.

It is clear that P is a subset of NP. However, the exact relation between these two classes is not known. The question whether or not P=NP is still a major open question of theoretical computer science. There is a particular type of NP problems, namely NP-complete problems, that are especially useful for the study of the NP class. A problem is called NP-complete if it belongs to the class NP and it has the property that any other problem in the NP class can be reduced to it in polynomial time. So if such a problem has a polynomial-time deterministic algorithm solving it, than all NP problems are solvable in polynomial time. However, it is mostly believed that P≠NP, as all efforts to find a polynomial-time deterministic algorithm for an NP complete problem have failed so far.

Therefore, it seems likely that no efficient algorithm exists for solving NP-complete problems on a conventional computer. But what about a DNA computer? In fact, a DNA computer models a non-deterministic computer, since operations on DNA strings can be done all in parallel, and hence DNA computer can theoretically implement a polynomial-time non-deterministic algorithm for solving an NP-complete problem. This was also demonstrated in practice by Adleman in 1994 on the Hamiltonian path problem. Later, other models of DNA computation to solve NP-complete problems were suggested. We describe below one of those models solving the SAT problem.

Apart from DNA algorithms solving specific problems, DNA models of a Turing machine have also been suggested. Some early models of DNA-based Turing machines were proposed by Beaver in 1995 [9] and by Rothemund in 1996 [38]. More recently, in 2017 [17], a non-deterministic universal Turing machine (NUTM) was designed using DNA and functionality of the design was demonstrated both computationally and experimentally. However, further research is needed to construct a fully working physical NUTM [17].

2.4.1 Hamiltonian path problem

Let us now describe Adleman's experiment, which was published in [1]. In this experiment, an instance of the Hamiltonian path problem in a directed graph was solved using DNA. A path in a directed graph is a sequence of vertices $(v_i)_{i=1}^{k}$ such that there is an edge starting in v_j and ending in v_{j+1} for all $j \in \{1, \ldots, k-1\}$. A Hamiltonian path is a path where every vertex of the graph appears exactly once. The problem of deciding whether a given directed graph has a Hamiltonian path or not is known to be an NP-complete problem [7].

In this specific case, a particular graph with seven vertices was considered, which is shown in Figure 2.3. The vertices were represented by numbers from 0 to 6. There were two designated vertices $v_{in} = 0$ and $v_{out} = 6$, and the problem of the Hamiltonian path was restricted only to the paths beginning in the vertex v_{in} and ending in the vertex v_{out}, which is still an NP-complete problem. For this instance of the problem, there exists exactly one suitable Hamiltonian path, and that is the path $(0, 1, 2, 3, 4, 5, 6)$.

Adleman solved this problem by implementing the following algorithm:

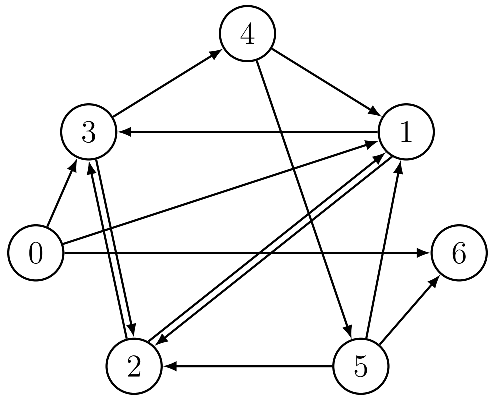


Figure 2.3: The directed graph used in Adleman's experiment.

Input: directed graph G with n vertices, vertex v_{in}, vertex v_{out}.

1. Generate random paths in G.
2. Keep only the paths that start in v_{in} and end in v_{out}.
3. Keep only the paths consisting of exactly n vertices.
4. Keep only the paths that contain each vertex at least once.
5. If at least one path remains, the output is "yes", otherwise the output is "no".

The output states whether G has a Hamiltonian path starting in v_{in} and ending in v_{out} or not.

The implementation was based on suitable encoding of the graph. Each vertex i of the graph was associated with a random strand of DNA of length 20, denoted by O_i. The DNA strand which is complementary to O_i was denoted by $\overline{O}_i$. Then an edge $i \to j$ of the graph was represented by a concatenation of the $3'$ half of O_i (unless $i = 0$, in which case it was the whole O_i) with the $5'$ half of O_j (unless $j = 6$, in which case it was the whole O_j), denoted by $O_{i \to j}$. This is demonstrated below:

$$
\begin{array}{ll}
O_2 : & 5' - \texttt{TATCGGATCGGTATATCCGA} - 3' \\
O_3 : & 5' - \texttt{GCTATTCGAGCTTAAAGCTA} - 3' \\
O_4 : & 5' - \texttt{GGCTAGGTACCAGCATGCTT} - 3' \\
O_{2\to 3} : & 5' - \texttt{GTATATCCGAGCTATTCGAG} - 3' \\
O_{3\to 4} : & 5' - \texttt{CTTAAAGCTAGGCTAGGTAC} - 3' \\
\overline{O}_3 : & 3' - \texttt{CGATAAGCTCGAATTTCGAT} - 5'
\end{array}
$$

This encoding was designed in such a way that the DNA strands $O_{i\to j}, O_{j\to k}$ and $\overline{O}_j$ can anneal like in the case below:

$$
\begin{array}{c}
5' - \overbrace{\texttt{GTATATCCGAGCTATTCGAG}}^{O_{2\to 3}}\,\overbrace{\texttt{CTTAAAGCTAGGCTAGGTAC}}^{O_{3\to 4}} - 3' \\
3' - \underbrace{\texttt{CGATAAGCTC}\,\texttt{GAATTTCGAT}}_{\overline{O}_3} - 5'
\end{array}
$$

With this encoding, the algorithm was carried out in the following way:

1. For each vertex i of the graph G, except for $i = 0$ and $i = 6$, and for each edge $j \to k$ in G, a sufficient amount of molecules $\overline{O}_i$ and $O_{j\to k}$ were mixed together and a ligation reaction was performed. Due to the annealing process this reaction resulted in double-stranded DNA molecules encoding random paths in G.

2. The next step was performed by polymerase chain reaction, which allowed to extract only molecules that encode paths that start in vertex 0 and end in vertex 6.

3. The molecules were separated by length using an agarose gel, and only those that were 140 base pair long were kept. These molecules correspond to paths containing exactly seven vertices.

4. All double-stranded molecules obtained in the previous step were melted into single strands, and for each vertex i, except for $i = 0$ and $i = 6$, strands containing the sequence O_i were successively extracted.

5. Finally, it was detected whether there are some DNA molecules left.

In between some of those steps described above, the amplification operation was used to increase the number of copies of the DNA molecules.

This calculation performed by Adleman took him a week of lab work. In theory, the number of procedures required grows linearly with the number of vertices of the graph. However, scaling up the computation comes with some difficulties. Great care needs to be taken with controlling errors and improvements in the method would have to be made to make it efficient. The weight of DNA molecules needed for such an experiment grows exponentially with the size of the problem, so this still places a barrier to the number of vertices the graph can have. Nevertheless, this experiment was a great breakthrough that demonstrated the possibility of employing DNA molecules to perform calculations [37].

2.4.2 SAT problem

In 1995, Richard J. Lipton [30] extended Adleman's idea and proposed a DNA model for solving another NP-complete problem, the so-called Boolean satisfiability (SAT) problem.

This problem can be formulated in the following way:

Given a set of Boolean variables $\{x_1, x_2, \ldots, x_n\}$, we define a literal to be some variable x_j or its negation, denoted by $\overline{x}_j$. A Boolean formula consists of literals and the logical operators "and" ($\wedge$) and "or" ($\vee$). We say that a Boolean formula is satisfiable if there exists some assignment of values (0 or 1) to the variables such that the value of the whole formula is 1. A clause is a Boolean formula of the form $u_1 \vee u_2 \vee \ldots \vee u_k$ for some $k \in \mathbb{N}$, where u_i is a literal for each $i \in \{1, \ldots, k\}$. The SAT problem is the problem of deciding whether a given Boolean formula $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$, where C_i is a clause for each $i \in \{1, \ldots, m\}$, is satisfiable or not.

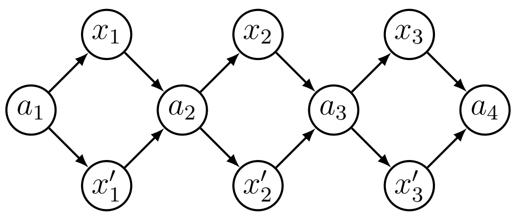


Figure 2.4: The directed graph G_3.

In Lipton's model, for a given instance of the SAT problem with m clauses $C_1, C_2, \ldots, C_m$ and n variables $x_1, x_2, \ldots, x_n$, a directed graph G_n is created. The graph G_n has $3n+1$ vertices $a_1, x_1, x'_1, \ldots, a_n, x_n, x'_n, a_{n+1}$ and $4n$ edges, for each $i \in \{1, \ldots, n\}$, there are edges $a_i \to x_i, a_i \to x'_i, x_i \to a_{i+1}, x'_i \to a_{i+1}$. For illustration, the graph G_3 is shown in Figure 2.4. The graph G_n is constructed in such a way that every possible assignment of values to the variables corresponds to one path in the graph G_n starting in vertex a_1 and ending in vertex a_{n+1}. If vertex x_i appears on the path, it means that the variable x_i has value 1. On the contrary, if vertex x'_i appears on the path, it means that the variable x_i has value 0.

Lipton used the same method as Adleman to encode paths in the graph G_n, and therefore all possible assignments of values to the set of variables, as DNA molecules. His molecular computation solving the SAT problem for a formula F with m clauses and n variables in linear time is the following:

1. In a test tube t_0, prepare DNA molecules encoding paths in the graph G_n starting in a_1 and ending in a_{n+1}. This is done analogously to the method of Adleman described in the previous section.

2. For each $k \in \{1, \ldots, m\}$, prepare a test tube t_k, which contains DNA molecules encoding assignment of values to the variables such that all clauses $C_1, C_2, \ldots, C_k$ have value 1, as follows:

    - Consider the clause $C_k = u_1 \vee u_2 \vee \ldots \vee u_l$. If $u_1 = x_j$, then from t_{k-1} extract all DNA molecules that encode paths containing x_j and place them in a test tube p_1, remainder of molecules place in a test tube p'_1. If $u_1 = \overline{x}_j$, then from t_{k-1} extract all DNA molecules that encode paths containing x'_j and place them in a test tube p_1, remainder of molecules place in a test tube p'_1.
    - For each $i \in \{2, \ldots, l\}$ do the following: if $u_i = x_j$, then from p'_{i-1} extract all DNA molecules that encode paths containing x_j and place them in a test tube p_i, remainder of molecules place in a test tube p'_i. If $u_i = \overline{x}_j$, then from p'_{i-1} extract all DNA molecules that encode paths containing x'_j and place them in a test tube p_i, remainder of molecules place in a test tube p'_i.
    - Pour all the test tubes $p_1, p_2, \ldots, p_l$ together to form t_k.

3. Detect whether the test tube t_m contains at least one DNA molecule or not. If yes, the output is "F is satisfiable", otherwise the output is "F is not satisfiable".

Chapter 3

Elements of combinatorics on words

We introduce some basic definitions and concepts from combinatorics on words that are relevant for our topic, mostly based on [31, 32]. However, the notation may differ and we present our examples and proofs.

3.1 Words

3.1.1 Finite words

Firstly, by an *alphabet*, usually denoted $\mathcal{A}$, we mean a non-empty set of finitely many elements called *letters*. Typically, letters are chosen to be some symbols. We can have for example the alphabet $\mathcal{A} = \{0, 1\}$ or $\mathcal{A} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$. A *word* w over $\mathcal{A}$ is a finite sequence of letters from $\mathcal{A}$, i.e., $w = w_1 w_2 \dots w_n$, where $w_i \in \mathcal{A}$ for all $i \in \{1, \dots, n\}$. Its *length* is $|w| = n$. By $|w|_a$ we denote the number of occurrences of letter $a \in \mathcal{A}$ in w. There is also the *empty word*, denoted by ε, which has length zero. Taking the set of all finite words over $\mathcal{A}$ and equipping it with the operation of concatenation, we get the free monoid $\mathcal{A}^*$. For all words $w \in \mathcal{A}$ we have $w\varepsilon = \varepsilon w = w$, and so ε is neutral element of $\mathcal{A}^*$.

Definition 3.1. *A word $y \in \mathcal{A}^*$ is called a factor of a word $w \in \mathcal{A}^*$, we also say that w contains y, if there exist words $x, z \in \mathcal{A}^*$ such that $w = xyz$. Moreover, if $x = \varepsilon$, resp. $z = \varepsilon$, then y is called a prefix, resp. a suffix, of the word w. A factor, prefix or suffix x of w is called proper if $x \neq w$.*

Note that the empty word is a prefix, a suffix and a factor of every word.

Definition 3.2. *Let $\mathcal{A} = \{a_1, \dots, a_d\}$ be an alphabet, where $d \in \mathbb{N}$, and let $w \in \mathcal{A}^*$. Then the Parikh vector of w, denoted $\vec{V}(w)$, is defined by*

$$\vec{V}(w) = \begin{pmatrix} |w|_{a_1} \\ \vdots \\ |w|_{a_d} \end{pmatrix}.$$

Proposition 3.3. *Let $\mathcal{A}$ be an alphabet and let $w \in \mathcal{A}^*$. Then*

$$|w| = \mathbf{1} \cdot \vec{V}(w),$$

where $\mathbf{1}$ is the row vector $\mathbf{1} = (1 \cdots 1)$ with the dimension $\mathrm{card}(\mathcal{A})$*.*

Proof. Let $\mathcal{A} = \{a_1, \dots, a_d\}$, where $d \in \mathbb{N}$. Then

$$\mathbf{1} \cdot \vec{V}(w) = (1 \cdots 1) \cdot \begin{pmatrix} |w|_{a_1} \\ \vdots \\ |w|_{a_d} \end{pmatrix} = |w|_{a_1} + \ldots + |w|_{a_d} = |w|. \qquad \square$$

Example 3.4. *Let $A = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$ and take $w = \mathrm{BBABCA}$. Letter* A *occurs in w twice, letter* B *three times, letter* C *once and letter* D *does not occur in w. The Parikh vector of w is therefore*

$$\vec{V}(w) = \begin{pmatrix} 2 \\ 3 \\ 1 \\ 0 \end{pmatrix}$$

and we have

$$|w| = \mathbf{1} \cdot \vec{V}(w) = 2 + 3 + 1 + 0 = 6.$$

3.1.2 Infinite words

So far we have seen finite words, but the definition of a word can be extended to infinite words. An *infinite word* $\boldsymbol{u}$ over the alphabet $\mathcal{A}$ is an infinite sequence $\boldsymbol{u} = u_1 u_2 \ldots$, where $u_i \in \mathcal{A}$ for all $i \in \mathbb{N}$. We denote the set of all infinite words over $\mathcal{A}$ by $\mathcal{A}^{\mathbb{N}}$.

Definition 3.5. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ and $w \in \mathcal{A}^*$. Then w is called a factor of $\boldsymbol{u}$ if $w = \varepsilon$ or there exist $i, j \in \mathbb{N}$, $i \leq j$ such that $w = u_i \ldots u_j$. In the latter case, the index i is called an occurrence of w in $\boldsymbol{u}$. The set of all factors of $\boldsymbol{u}$ is called the language of $\boldsymbol{u}$ and is denoted $\mathcal{L}(\boldsymbol{u})$. By $\mathcal{L}_n(\boldsymbol{u})$ we denote the set of factors of length n in $\boldsymbol{u}$.*

It is worth defining some properties of infinite words.

Definition 3.6. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$. Then $\boldsymbol{u}$ is called eventually periodic if there exist $v, w \in \mathcal{A}^*$, $w \neq \varepsilon$, such that $\boldsymbol{u} = vw^{\infty}$. Here w^{∞} means concatenation of infinitely many w. In the case that $v = \varepsilon$, $\boldsymbol{u}$ is called purely periodic. If $\boldsymbol{u}$ is not eventually periodic, we say it is aperiodic.*

Definition 3.7. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$. Then $\boldsymbol{u}$ is called recurrent if each of its factors has at least two occurrences in $\boldsymbol{u}$. Moreover, $\boldsymbol{u}$ is called uniformly recurrent if for each $n \in \mathbb{N}$ there is $r \in \mathbb{N}$, $r \geq n$, such that all factors of $\boldsymbol{u}$ of length n have at least one occurrence in the set $\{i, i+1, \ldots, i+r-n\}$ for all $i \in \mathbb{N}$, or equivalently any factor of $\boldsymbol{u}$ of length r contains all factors of $\boldsymbol{u}$ of length n.*

Remark 3.8. *In fact, $\boldsymbol{u}$ is recurrent if and only if all factors $w \in \mathcal{L}(\boldsymbol{u})$ have infinitely many occurrences. Hence, this statement gives an alternative way of defining a recurrent word. It is clear that if all factors have infinitely many occurrences, then $\boldsymbol{u}$ is recurrent. To see the other implication, we consider a recurrent word $\boldsymbol{u}$ and some factor $w \in \mathcal{L}(\boldsymbol{u})$. It is given that w occurs at least twice in $\boldsymbol{u}$, and so we can find a factor $v \in \mathcal{L}(\boldsymbol{u})$, which contains both occurrences of w. It follows that v has at least two occurrences, which means w has at least one additional occurrence, and again we could find a larger factor that contains all occurrences of w and it appears at least twice. In this way, we can always increase the number of occurrences of w, and therefore there are infinitely many of them.*

3.1.3 Factor complexity

When we work with an infinite word, we need some information about its factors. Here we introduce the so-called complexity function, which gives the number of all existing factors of certain length. It is then shown how this function relates to some other properties of infinite words.

Definition 3.9. *Let $\mathcal{A}$ be an alphabet and let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$. Then we define the factor complexity function of $\boldsymbol{u}$, $c_{\boldsymbol{u}} \colon \mathbb{N} \to \mathbb{N}$, by $c_{\boldsymbol{u}}(n) = \mathrm{card}(\mathcal{L}_n(\boldsymbol{u}))$, that is the number of factors in $\boldsymbol{u}$ of length n.*

Proposition 3.10. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be eventually periodic. Then there exists $m \in \mathbb{N}$ such that $c_{\boldsymbol{u}}(n) \leq m$ for all $n \in \mathbb{N}$.*

Proof. Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be eventually periodic, so $\boldsymbol{u} = vw^{\infty}$ for some $v, w \in \mathcal{A}^*$, $w \neq 0$. Denote $p = |v|$ and $q = |w|$. We take $n \in \mathbb{N}$ and consider factors of $\boldsymbol{u}$ of length n. We see that if we choose a factor $x = u_{p+q+i}u_{p+q+i+1} \dots u_{p+q+i+n-1}$, where $i \geq 1$, it is equal to the factor $y = u_{p+i}u_{p+i+1} \dots u_{p+i+n-1}$. Therefore to find the set $\mathcal{L}_n(\boldsymbol{u})$, it is sufficient to only consider factors with occurrence between 1 and $p+q$. This implies that $c_{\boldsymbol{u}}(n) = \operatorname{card}(\mathcal{L}_n(\boldsymbol{u})) \leq p + q$ for all $n \in \mathbb{N}$, which concludes the proof. □

The following definitions are helpful in studying factor complexity and can be found in [35].

Definition 3.11. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ and $w \in \mathcal{L}(\boldsymbol{u})$. A word $r \in \mathcal{L}(\boldsymbol{u})$ is said to be a complete return word to w in $\boldsymbol{u}$ if w has exactly two occurrences in r, one as a prefix of r and one as a suffix of r. If r is a complete return word to w, then it can be written as $r = qw$ for some non-empty $q \in \mathcal{L}(\boldsymbol{u})$, which is called a return word to w.*

Definition 3.12. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ and $w \in \mathcal{L}(\boldsymbol{u})$. A letter $a \in \mathcal{A}$ is called a left extension of w in $\boldsymbol{u}$ if $aw \in \mathcal{L}(\boldsymbol{u})$. $\mathrm{Lext}_{\boldsymbol{u}}(w)$ denotes the set of all left extensions of w in $\boldsymbol{u}$. We say that w is left-special if $\operatorname{card}(\mathrm{Lext}_{\boldsymbol{u}}(w)) \geq 2$. Analogously, a letter $a \in \mathcal{A}$ is called a right extension of w in $\boldsymbol{u}$ if $wa \in \mathcal{L}(\boldsymbol{u})$, which leads to the analogous definition of $\mathrm{Rext}_{\boldsymbol{u}}(w)$ and right-special word. We say that w is special if it is left-special or right-special. Moreover, w is called bispecial if it is both left-special and right-special. We also define $\mathrm{Bext}_{\boldsymbol{u}}(w) = \{awb \mid a, b \in \mathcal{A},\, awb \in \mathcal{L}(\boldsymbol{u})\}$.*

Definition 3.13. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ and $w \in \mathcal{L}(\boldsymbol{u})$. The bilateral order of w, denoted by $b_{\boldsymbol{u}}(w)$, is defined as*

$$b_{\boldsymbol{u}}(w) = \operatorname{card}(\mathrm{Bext}_{\boldsymbol{u}}(w)) - \operatorname{card}(\mathrm{Lext}_{\boldsymbol{u}}(w)) - \operatorname{card}(\mathrm{Rext}_{\boldsymbol{u}}(w)) + 1.$$

We can relate these concepts to factor complexity [16].

Proposition 3.14. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be a recurrent word. Then the first difference $\Delta c_{\boldsymbol{u}}(n)$ and the second difference $\Delta^2 c_{\boldsymbol{u}}(n)$ of the factor complexity satisfy*

$$\Delta c_{\boldsymbol{u}}(n) = c_{\boldsymbol{u}}(n+1) - c_{\boldsymbol{u}}(n) = \sum_{w \in \mathcal{L}_n(\boldsymbol{u})} (\operatorname{card}(\mathrm{Lext}_{\boldsymbol{u}}(w)) - 1) = \sum_{w \in \mathcal{L}_n(\boldsymbol{u})} (\operatorname{card}(\mathrm{Rext}_{\boldsymbol{u}}(w)) - 1)$$

and

$$\Delta^2 c_{\boldsymbol{u}}(n) = \Delta c_{\boldsymbol{u}}(n+1) - \Delta c_{\boldsymbol{u}}(n) = \sum_{w \in \mathcal{L}_n(\boldsymbol{u})} b_{\boldsymbol{u}}(w).$$

Another useful tool that uses factors of an infinite word is the Rauzy graph.

Definition 3.15. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ and $n \in \mathbb{N}$. Then the Rauzy graph Γ_n of $\boldsymbol{u}$ is a directed graph $\Gamma_n = (V, E)$, where the set of vertices is $V = \mathcal{L}_n(\boldsymbol{u})$ and the set of edges is $E = \mathcal{L}_{n+1}(\boldsymbol{u})$. The vertices and edges are arranged so that an edge $e = w_1 w_2 \dots w_{n+1}$ starts in the vertex $v_1 = w_1 \dots w_n$ and ends in the vertex $v_2 = w_2 \dots w_{n+1}$.*

Remark 3.16. *We understand the Rauzy graph as an illustration of the structure of the infinite word $\boldsymbol{u}$. Moving along an edge $e = u_i \dots u_{i+n}$ in Γ_n from one vertex to another vertex represents shifting from a factor of length n with occurrence i to a factor of the same length with occurrence $i+1$. If we consider a factor $w = u_j \dots u_{j+m-1}$ of $\boldsymbol{u}$ of length $m > n$, it is represented in the Rauzy graph Γ_n by a directed path from the vertex $v_1 = u_j \dots u_{j+n-1}$ to the vertex $v_2 = u_{j+m-n} \dots u_{j+m-1}$. And so as we read the word $\boldsymbol{u}$, we move on a path in the Rauzy graph.*

It follows that Γ_n is always connected graph, which means that for each pair of vertices in Γ_n, there is a sequence of edges and vertices (not necessarily directed path) connecting them. Next proposition gives condition on $\boldsymbol{u}$ that insures Γ_n is strongly connected, meaning that for any vertex chosen as a starting point and any vertex chosen as an end, there is a directed path between them.

Proposition 3.17. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$. Then the Rauzy graph Γ_n of $\boldsymbol{u}$ is strongly connected for all $n \in \mathbb{N}$ if and only if $\boldsymbol{u}$ is recurrent.*

Proof. Firstly, assume that $\boldsymbol{u}$ is recurrent. As was discussed in Remark 3.8, then all factors of $\boldsymbol{u}$ have infinitely many occurrences. For some fixed $n \in \mathbb{N}$, we choose vertices v_1 and v_2 in Γ_n and we want to show that there is a directed path between them. Using the approach described in Remark 3.16, it is sufficient to find a factor of $\boldsymbol{u}$ with prefix v_1 and with suffix v_2. As both factors have infinitely many occurrences, we can take an occurrence i of v_1 and occurrence j of v_2 satisfying $i < j$. Then $w = u_i \dots u_{j+n-1}$ is the desired factor. Therefore Γ_n is strongly connected for all $n \in \mathbb{N}$.

Secondly, we start with the assumption that $\boldsymbol{u}$ is not recurrent and we want to show that there exists some $n \in \mathbb{N}$ such that Γ_n is not strongly connected. In order to satisfy the assumption, $\boldsymbol{u}$ has to have a factor that has exactly one occurrence. Moreover, $\boldsymbol{u}$ has to have a prefix that has exactly one occurrence. To see this, we assume that every prefix of $\boldsymbol{u}$ has at least two occurrences. Hence, also a prefix containing the factor with only one occurrence must appear in $\boldsymbol{u}$ at least twice, which is a contradiction. So we have a prefix v that appears in $\boldsymbol{u}$ only once and we choose n to be its length. Then this prefix is represented in Γ_n by a vertex v. It follows that there is no edge in Γ_n ending in v and so there cannot be a directed path ending in v. Therefore Γ_n is not strongly connected. □

Lemma 3.18. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be aperiodic. Then for all $n \in \mathbb{N}$ the Rauzy graph $\Gamma_n = (V, E)$ of $\boldsymbol{u}$ satisfies $\operatorname{card}(E) \geq 1 + \operatorname{card}(V)$.*

Proof. We prove the contrapositive statement. Assume that there is $n \in \mathbb{N}$ such that for $\Gamma_n = (V, E)$, $\operatorname{card}(E) \leq \operatorname{card}(V)$. V represents all factors of length n and from the construction of Γ_n we know that Γ_n is connected and also for every vertex there is at least one edge starting there. Therefore $\operatorname{card}(E) \geq \operatorname{card}(V)$. Hence, to satisfy our assumption, we need $\operatorname{card}(E) = \operatorname{card}(V)$, and so there is exactly one edge starting in every vertex. There are two possibilities. One is that for every vertex there is exactly one edge ending there, in which case the graph Γ_n is a cycle and $\boldsymbol{u}$ is purely periodic. Second possibility is that there is exactly one vertex which is not an ending point of any edge and exactly one vertex, which has exactly two edges ending there. This means that Γ_n contains a cycle and $\boldsymbol{u}$ is eventually periodic. To conclude, $\boldsymbol{u}$ is not aperiodic, which proves the statement. □

Next proposition relates aperiodicity and factor complexity.

Proposition 3.19. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be aperiodic. Then for all $n \in \mathbb{N}$, $c_{\boldsymbol{u}}(n) \geq n + 1$.*

Proof. Using the definition of a Rauzy graph, Lemma 3.18 gives us that for all $m \in \mathbb{N}$, $\operatorname{card}(\mathcal{L}_{m+1}(\boldsymbol{u})) \geq 1 + \operatorname{card}(\mathcal{L}_m(\boldsymbol{u}))$. If for a fixed $n \in \mathbb{N}$ we sum those inequalities from $m = 1$ to $m = n - 1$, we get

$$\operatorname{card}(\mathcal{L}_2(\boldsymbol{u})) + \cdots + \operatorname{card}(\mathcal{L}_n(\boldsymbol{u})) \geq \operatorname{card}(\mathcal{L}_1(\boldsymbol{u})) + \cdots + \operatorname{card}(\mathcal{L}_{n-1}(\boldsymbol{u})) + n - 1.$$

Subtracting terms from both sides and using that $\operatorname{card}(\mathcal{L}_1(\boldsymbol{u})) = \operatorname{card}(A) \geq 2$, as $\boldsymbol{u}$ is aperiodic, results in

$$c_{\boldsymbol{u}}(n) = \operatorname{card}(\mathcal{L}_n(\boldsymbol{u})) \geq n + 1.$$

□

Note that the opposite implication also holds, as we have shown in Proposition 3.10 that if a word is periodic, its factor complexity function is bounded. So if $c_{\boldsymbol{u}}(n) \geq n+1$, $\boldsymbol{u}$ is aperiodic word.

There is a special class of aperiodic words, called Sturmian words, for which the factor complexity function attains the minimal possible values, i.e., $c_{\boldsymbol{u}}(n) = n+1$ for all $n \in \mathbb{N}$. This condition implies that Sturmian words are over a binary alphabet, because $c_{\boldsymbol{u}}(1) = \operatorname{card}(\mathcal{A}) = 2$. One example of a Sturmian word is the Fibonacci word $\boldsymbol{f} = 01001010010010100101\ldots$ [12, p. 183]. We show how to generate this word in example 3.40. In the following example, we use the Fibonacci word to construct one of its Rauzy graphs.

Example 3.20. *Consider the Fibonacci word* $\boldsymbol{f} = 01001010010010100101\ldots$ *We want to construct the Rauzy graph* $\Gamma_4 = (V, E)$ *of* $\boldsymbol{f}$. *We know that it is a Sturmian word, and therefore* $c_{\boldsymbol{f}}(n) = n+1$ *for all* $n \in \mathbb{N}$. *We have* $\operatorname{card}(V) = \operatorname{card}(\mathcal{L}_4(\boldsymbol{f})) = c_{\boldsymbol{f}}(4) = 5$ *and* $\operatorname{card}(E) = \operatorname{card}(\mathcal{L}_5(\boldsymbol{f})) = c_{\boldsymbol{f}}(5) = 6$. *Hence, to construct the graph* Γ_4, *it is sufficient to consider only such a prefix of* $\boldsymbol{f}$ *that contains six distinct factors of length* 5. *The shortest prefix of* $\boldsymbol{f}$ *satisfying this is the prefix* $w = 010010100100$. *Using this word, we obtain the graph* Γ_4, *which is shown in Figure 3.1.*

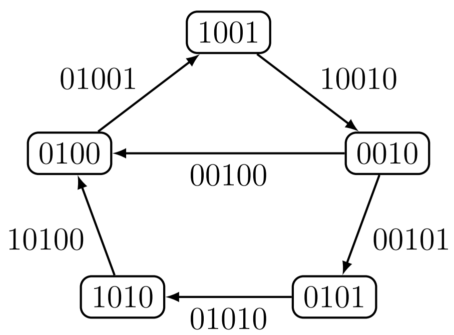


Figure 3.1: The Rauzy graph Γ_4 of the Fibonacci word $\boldsymbol{f}$ discussed in Example 3.20.

3.2 Morphisms

In this section, we define a type of mapping called morphism and introduce some related concepts.

Definition 3.21. *Let* $\mathcal{A}, \mathcal{B}$ *be alphabets and let* $\varphi\colon \mathcal{A}^* \to \mathcal{B}^*$ *be a mapping satisfying* $\varphi(uv) = \varphi(u)\varphi(v)$ *for all* $u, v \in \mathcal{A}^*$. *Then* φ *is called a morphism.*

In general, there are two alphabets in the definition of morphism, however, in most cases we consider $\mathcal{A} = \mathcal{B}$. It follows from the definition that to specify a morphism, it is sufficient to assign it only on letters. To find the image of a word $w = w_1 w_2 \ldots w_n$ under the morphism φ, we concatenate the images of the letters in w, so $\varphi(w_1 w_2 \ldots w_n) = \varphi(w_1)\varphi(w_2)\ldots\varphi(w_n)$. Note that the image of the empty word is the empty word. In the same way we can find the image of an infinite word $\boldsymbol{u} = u_1 u_2 \ldots$ simply as $\varphi(\boldsymbol{u}) = \varphi(u_1)\varphi(u_2)\ldots$. Hence, the domain of a morphism can be naturally extended to infinite words.

Definition 3.22. *Let* $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ *be a morphism. We say that it is a uniform morphism, if there exists* $l \in \mathbb{N}$ *such that* $|\varphi(a)| = l$ *for all* $a \in A$.

Definition 3.23. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism. We say that φ is erasing if there exists $a \in \mathcal{A}$ such that $\varphi(a) = \varepsilon$. Otherwise, we say that φ is non-erasing. If φ is a non-erasing morphism, we define functions $\operatorname{fst}_\varphi, \operatorname{lst}_\varphi\colon \mathcal{A} \to \mathcal{A}$ by setting $\operatorname{fst}_\varphi(a)$ to be the first letter of $\varphi(a)$ and $\operatorname{lst}_\varphi(a)$ to be the last letter of $\varphi(a)$ for all $a \in \mathcal{A}$.*

Definition 3.24. *Let $\varphi, \psi\colon \mathcal{A}^* \to \mathcal{A}^*$ be morphisms. If there exist $q \in \mathcal{A}^*$ such that $q\varphi(w) = \psi(w)q$ for every word $w \in \mathcal{A}^*$, then φ is called a left conjugate to ψ, ψ is called a right conjugate to φ and q is called the conjugacy word of this pair of morphisms. We also say that φ and ψ are conjugated and we denote it by $\varphi \sim \psi$ or equivalently $\psi \sim \varphi$. In addition, if φ is the only left conjugate to itself, then it is called the leftmost conjugate to ψ, and it is denoted by ψ_L. Analogously, if ψ is the only right conjugate to itself, then it is called the rightmost conjugate to φ, and it is denoted by φ_R.*

Remark 3.25. *Note that in the definition above, it is sufficient to consider only images of letters instead of words, i.e., if $q\varphi(a) = \psi(a)q$ for all $a \in \mathcal{A}$, then $q\varphi(w) = \psi(w)q$ for all $w \in \mathcal{A}^*$. To see this, consider a word $w = w_1 \dots w_n$. If $q\varphi(a) = \psi(a)q$ for all $a \in \mathcal{A}$, then*

$$\begin{aligned} q\varphi(w) &= q\varphi(w_1)\varphi(w_2)\dots\varphi(w_n) = \psi(w_1)q\varphi(w_2)\dots\varphi(w_n) \\ &= \psi(w_1)\psi(w_2)q\varphi(w_3)\dots\varphi(w_n) = \dots = \psi(w_1)\psi(w_2)\dots\psi(w_n)q = \psi(w)q. \end{aligned}$$

Proposition 3.26. *The relation of conjugation is an equivalence relation on the set of morphisms on $\mathcal{A}^*$.*

Proof. Let $\varphi, \psi, \theta\colon \mathcal{A}^* \to \mathcal{A}^*$ be morphisms. Then we have $\varphi \sim \varphi$ with the conjugacy word $q = \epsilon$. By definition, $\varphi \sim \psi$ if and only if $\psi \sim \varphi$. Finally, assume that $\varphi \sim \psi$ and $\psi \sim \theta$. Without loss of generality, we can assume that there exists $q \in \mathcal{A}^*$ such that $q\varphi(w) = \psi(w)q$ for every word $w \in \mathcal{A}^*$. Now we have two possibilities.

In the first case, there is a word $p \in \mathcal{A}^*$ such that $p\psi(w) = \theta(w)p$ for every word $w \in \mathcal{A}^*$. This implies that, for every $w \in \mathcal{A}^*$,

$$pq\varphi(w) = p\psi(w)q = \theta(w)pq,$$

which means that $\varphi \sim \theta$.

In the second case, there is a word $p \in \mathcal{A}^*$ such that $p\theta(w) = \psi(w)p$ for every word $w \in \mathcal{A}^*$. Hence, $p\theta(a) = \psi(a)p$ for all $a \in \mathcal{A}$. From above, we have $q\varphi(a) = \psi(a)q$ for all $a \in \mathcal{A}$. This means that for each $a \in \mathcal{A}$, there are words v_a and w_a such that

$$\varphi(a) = v_a q, \quad \psi(a) = qv_a, \quad \theta(a) = w_a p, \quad \psi(a) = pw_a.$$

This implies that $qv_a = pw_a$. Without loss of generality, assume $|q| \geq |p|$. Hence, there is a word $s \in \mathcal{A}^*$ such that

$$q = ps, \quad w_a = sv_a.$$

Combining this together, we get

$$s\varphi(a) = sv_a q = sv_a ps = w_a ps = \theta(a)s,$$

which means, by Remark 3.25, that $\varphi \sim \theta$. □

Example 3.27. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ and let $\varphi, \psi\colon \mathcal{A}^* \to \mathcal{A}^*$ be morphisms given by*

$$\begin{aligned} \varphi(\mathrm{A}) &= \mathrm{CBBBA} & \psi(\mathrm{A}) &= \mathrm{ACBBB} \\ \varphi(\mathrm{B}) &= \mathrm{CACBA} & \psi(\mathrm{B}) &= \mathrm{ACACB} \\ \varphi(\mathrm{C}) &= \mathrm{CCBA} & \psi(\mathrm{C}) &= \mathrm{ACCB}. \end{aligned}$$

In this example, ψ is a right conjugate to φ and the conjugacy word is $q = \mathrm{A}$. However, it is not the rightmost conjugate to φ. The rightmost and leftmost conjugates to φ are given by

$$\begin{aligned} \varphi_R(\mathrm{A}) &= \mathrm{BACBB} & \qquad \varphi_L(\mathrm{A}) &= \mathrm{BBBAC} \\ \varphi_R(\mathrm{B}) &= \mathrm{BACAC} & \qquad \varphi_L(\mathrm{B}) &= \mathrm{ACBAC} \\ \varphi_R(\mathrm{C}) &= \mathrm{BACC} & \qquad \varphi_L(\mathrm{C}) &= \mathrm{CBAC}. \end{aligned}$$

We see that φ_R and φ_L are also the rightmost and leftmost conjugates to ψ, respectively. This follows from Proposition 3.26.

Next, we introduce a useful tool for understanding some properties of a morphism.

Definition 3.28. *Let $\mathcal{A} = \{a_1, \dots, a_d\}$ be an alphabet, where $d \in \mathbb{N}$, and let $\varphi \colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism. Then the incidence matrix of φ, denoted M_φ, is the $d \times d$ matrix defined by*

$$[M_\varphi]_{ij} = |\varphi(a_j)|_{a_i}.$$

Example 3.29. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ and let $\varphi \colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism given by*

$$\begin{aligned} \varphi(\mathrm{A}) &= \mathrm{CA} \\ \varphi(\mathrm{B}) &= \mathrm{BAB} \\ \varphi(\mathrm{C}) &= \mathrm{AABAC}. \end{aligned}$$

Then the incidence matrix of φ is

$$M_\varphi = \begin{pmatrix} 1 & 1 & 3 \\ 0 & 2 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

Proposition 3.30. *Let $\mathcal{A}$ be an alphabet and let $\varphi, \psi \colon \mathcal{A}^* \to \mathcal{A}^*$ be morphisms. Then the incidence matrix of the composition $\varphi \circ \psi$ satisfies*

$$M_{\varphi \circ \psi} = M_\varphi \cdot M_\psi.$$

Proof. Let $\mathcal{A} = \{a_1, \dots, a_d\}$, where $d \in \mathbb{N}$. Then, for all $i, j \in \{1, \dots, d\}$, we have

$$[M_{\varphi\circ\psi}]_{ij} = |\varphi(\psi(a_j))|_{a_i} = \sum_{k=1}^{d} |\varphi(a_k)|_{a_i} \cdot |\psi(a_j)|_{a_k} = \sum_{k=1}^{d} [M_\varphi]_{ik} \cdot [M_\psi]_{kj} = [M_\varphi \cdot M_\psi]_{ij},$$

where we used that the number of letters a_i in $\varphi(\psi(a_j))$ is equal to the sum of the number of letters a_k in $\psi(a_j)$ times the number of letters a_i in $\varphi(a_k)$ over all letters $a_k \in \mathcal{A}$. □

Proposition 3.31. *Let $\mathcal{A}$ be an alphabet, $w \in \mathcal{A}^*$ and let $\varphi \colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism. Then the Parikh vector of $\varphi(w)$ satisfies*

$$\vec{V}(\varphi(w)) = M_\varphi \cdot \vec{V}(w).$$

Proof. Let $\mathcal{A} = \{a_1, \dots, a_d\}$, where $d \in \mathbb{N}$. Then, for all $i \in \{1, \dots, d\}$, we have

$$\left[\vec{V}(\varphi(w))\right]_i = |\varphi(w)|_{a_i} = \sum_{k=1}^{d} |\varphi(a_k)|_{a_i} \cdot |w|_{a_k} = \sum_{k=1}^{d} [M_\varphi]_{ik} \cdot \left[\vec{V}(w)\right]_k = \left[M_\varphi \cdot \vec{V}(w)\right]_i,$$

where we used that the number of letters a_i in $\varphi(w)$ is equal to the sum of the number of letters a_k in w times the number of letters a_i in $\varphi(a_k)$ over all letters $a_k \in \mathcal{A}$. □

3.2.1 Generating infinite words

Now we describe a way to employ morphisms with some specific properties to generate infinite words.

Definition 3.32. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism. It is called a primitive morphism if there exists $k \in \mathbb{N}$ such that for all $a, b \in \mathcal{A}$, a is a factor of $\varphi^k(b)$.*

Proposition 3.33. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism. Then it is primitive if and only if there exists $k \in \mathbb{N}$ such that the k-th power of the incidence matrix, M_φ^k, has all its elements positive.*

Proof. A morphism φ is primitive if and only if there exists $k \in \mathbb{N}$ such that for all $a, b \in \mathcal{A}$, a is a factor of $\varphi^k(b)$. This is equivalent to the statement that all elements of the incidence matrix M_{φ^k} are positive. By Proposition 3.30, $M_{\varphi^k} = M_\varphi^k$. Therefore φ is primitive if and only if there exists $k \in \mathbb{N}$ such that M_φ^k has all its elements positive. $\square$

Example 3.34. *Taking $\mathcal{A} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$, we define a morphism φ by*

$$\begin{aligned}
\varphi(\mathrm{A}) &= \mathrm{AB}\\
\varphi(\mathrm{B}) &= \mathrm{BA}\\
\varphi(\mathrm{C}) &= \mathrm{ABC}.
\end{aligned}$$

Then we see that for all $n \in \mathbb{N}$ the iteration $\varphi^n(\mathrm{A})$ contains only the letters A and B. Therefore φ cannot be a primitive morphism, because C is not a factor of $\varphi^n(\mathrm{A})$ for all $n \in \mathbb{N}$.

Example 3.35. *Again $\mathcal{A} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$. Now we define a morphism φ by*

$$\begin{aligned}
\varphi(\mathrm{A}) &= \mathrm{B}\\
\varphi(\mathrm{B}) &= \mathrm{AC}\\
\varphi(\mathrm{C}) &= \mathrm{A}.
\end{aligned}$$

Then if we look at the images of all the letters under iterations of φ, we get:

$$\begin{array}{ccccccccccc}
\mathrm{A} & \xrightarrow{\varphi} & \mathrm{B} & \xrightarrow{\varphi} & \mathrm{AC} & \xrightarrow{\varphi} & \mathrm{BA} & \xrightarrow{\varphi} & \mathrm{ACB} & \xrightarrow{\varphi} & \mathrm{BAAC}\\
\mathrm{B} & \xrightarrow{\varphi} & \mathrm{AC} & \xrightarrow{\varphi} & \mathrm{BA} & \xrightarrow{\varphi} & \mathrm{ACB} & \xrightarrow{\varphi} & \mathrm{BAAC} & \xrightarrow{\varphi} & \mathrm{ACBBA}\\
\mathrm{C} & \xrightarrow{\varphi} & \mathrm{A} & \xrightarrow{\varphi} & \mathrm{B} & \xrightarrow{\varphi} & \mathrm{AC} & \xrightarrow{\varphi} & \mathrm{BA} & \xrightarrow{\varphi} & \mathrm{ACB}
\end{array}$$

It is enough to consider iterations up to the fifth one, as for all $a \in \mathcal{A}$ every letter in $\mathcal{A}$ is a factor of the word $\varphi^5(a)$. Therefore we conclude that φ is a primitive morphism.

Definition 3.36. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism. It is called a substitution if there exist $a \in \mathcal{A}$ and $w \in \mathcal{A}^* \setminus \{\varepsilon\}$ such that $\varphi(a) = aw$ and $\lim_{n\to\infty}|\varphi^n(a)| = \infty$.*

Consider again Example 3.35. In this case there is no $a \in \mathcal{A}$ satisfying $\varphi(a) = aw$ for some $w \in \mathcal{A}^*$. Therefore φ is not a substitution. However, $\lim_{n\to\infty}|\varphi^n(a)| = \infty$ holds for all $a \in \mathcal{A}$ and we notice that $\varphi^2(\mathrm{A}) = \mathrm{AC}$, so φ^2 is a substitution.

Example 3.37. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ and let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism defined by*

$$\begin{aligned}
\varphi(\mathrm{A}) &= \mathrm{AB}\\
\varphi(\mathrm{B}) &= \mathrm{C}\\
\varphi(\mathrm{C}) &= \mathrm{AA}.
\end{aligned}$$

Then we can take $a = \mathrm{A}$, $w = \mathrm{B}$ *and we see that* $\varphi(a) = aw$ *is satisfied. Also by applying* φ *to* A *iteratively we get the sequence of words*

$$\mathrm{A} \xrightarrow{\varphi} \mathrm{AB} \xrightarrow{\varphi} \mathrm{ABC} \xrightarrow{\varphi} \mathrm{ABCAA} \xrightarrow{\varphi} \mathrm{ABCAAABAB} \xrightarrow{\varphi} \cdots$$

and it is clear that $\lim_{n\to\infty}|\varphi^n(a)| = \infty$. *Hence,* φ *is a substitution.*

Definition 3.38. *Let* $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ *be a morphism and let* $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$. *Then* $\boldsymbol{u}$ *is called a fixed point of* φ *if* $\varphi(\boldsymbol{u}) = \boldsymbol{u}$.

Every substitution has at least one infinite fixed point. We can illustrate it by considering the iterations of φ applied to a. We have the following sequence of words, which is generalization of the sequence in Example 3.37:

$$a \xrightarrow{\varphi} aw \xrightarrow{\varphi} aw\varphi(w) \xrightarrow{\varphi} aw\varphi(w)\varphi^2(w) \xrightarrow{\varphi} aw\varphi(w)\varphi^2(w)\varphi^3(w) \xrightarrow{\varphi} \cdots$$

We see that for $i \leq j$, $\varphi^i(a)$ is a prefix of $\varphi^j(a)$. If we take the infinite word

$$\boldsymbol{u} = aw\varphi(w)\varphi^2(w)\varphi^3(w)\ldots,$$

we see that $\varphi(\boldsymbol{u}) = \boldsymbol{u}$, and so $\boldsymbol{u}$ is a fixed point of φ.

This gives us a way how to generate infinite words. First, we define a substitution, and then we apply it iteratively, starting with a from the definition of a substitution. We say that an infinite word obtained in this way is generated by φ.

Definition 3.39. *Let* $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ *be a morphism and let* $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ *be its fixed point. Then we say that* $\boldsymbol{u}$ *is generated by* φ *if there exist* $a \in \mathcal{A}$ *and* $w \in \mathcal{A}^* \setminus \{\varepsilon\}$ *such that* $\varphi(a) = aw$ *and* $\boldsymbol{u} = aw\varphi(w)\varphi^2(w)\varphi^3(w)\ldots$

Example 3.40. *As one example we can take the so-called Fibonacci morphism* $\varphi_f\colon \{0,1\}^* \to \{0,1\}^*$ *defined by*

$$\varphi_f(0) = 01, \quad \varphi_f(1) = 0,$$

which is a substitution.

We start with 0 and apply φ_f *iteratively:*

$$0 \xrightarrow{\varphi_f} 01 \xrightarrow{\varphi_f} 010 \xrightarrow{\varphi_f} 01001 \xrightarrow{\varphi_f} 01001010 \xrightarrow{\varphi_f} 0100101001001 \xrightarrow{\varphi_f} \cdots$$

This gives the Fibonacci word $\boldsymbol{f} = 0100101001001\ldots$

Example 3.41. *Another example uses the so-called Thue-Morse morphism* $\varphi_t\colon \{0,1\}^* \to \{0,1\}^*$ *defined by*

$$\varphi_t(0) = 01, \quad \varphi_t(1) = 10,$$

which is a substitution.

We observe that this morphism has two fixed points. If we choose the one starting with 0, we generate it as follows:

$$0 \xrightarrow{\varphi_t} 01 \xrightarrow{\varphi_t} 0110 \xrightarrow{\varphi_t} 01101001 \xrightarrow{\varphi_t} 0110100110010110 \xrightarrow{\varphi_t} \cdots$$

and we obtain the Thue-Morse word $\boldsymbol{t} = 0110100110010110\ldots$

It is important to note that not only substitutions can have infinite fixed points. One example is the identity map, given by $\varphi(a) = a$ for all $a \in \mathcal{A}$. All finite and infinite words are fixed points of this morphism, however, it is not a substitution. Another example is the morphism $\varphi\colon \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}^* \to \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}^*$ given by

$$\begin{aligned} \varphi(\mathrm{A}) &= \mathrm{A} \\ \varphi(\mathrm{B}) &= \mathrm{BC} \\ \varphi(\mathrm{C}) &= \varepsilon. \end{aligned}$$

It is not a substitution, yet the infinite word ABCABCABC... is its fixed point.

Now if we add the condition of primitivity, we get the following proposition.

Proposition 3.42. *Let $\mathcal{A}$ be an alphabet with $\operatorname{card}(\mathcal{A}) > 1$ and let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism. Then φ has an infinite fixed point $\boldsymbol{u}$ if and only if φ is a substitution. Moreover, $\boldsymbol{u}$ is generated by φ.*

Proof. We have already seen that every substitution has at least one infinite fixed point generated by this substitution. Now we want to show the other implication, that a primitive morphism φ with an infinite fixed point $\boldsymbol{u}$ is a substitution and $\boldsymbol{u}$ is generated by φ. Let us denote $\boldsymbol{u} = u_1 u_2 \ldots$, where $u_i \in \mathcal{A}$ for all $i \in \mathbb{N}$, and set $a = u_1$ and $d = \operatorname{card}(\mathcal{A}) > 1$. Because $\varphi(\boldsymbol{u}) = \boldsymbol{u}$, we must have $\varphi(a) = \varepsilon$ or $\varphi(a) = aw$, where $w \in \mathcal{A}^*$. $\varphi(a) = \varepsilon$ is not possible, because φ is primitive. Hence, $\varphi(a) = aw$. If $w = \varepsilon$, then $\varphi^n(a) = a$ for all $n \in \mathbb{N}$ and so φ could not be primitive. Therefore $w \neq \varepsilon$. As φ is primitive, there exists $k \in \mathbb{N}$ such that for all $b, c \in \mathcal{A}$, b is a factor of $\varphi^k(c)$. Therefore $\left|\varphi^k(c)\right| \geq d$ for all $c \in \mathcal{A}$. Hence $\left|\varphi^{nk}(a)\right| \geq d^n$ for all $n \in \mathbb{N}$ and so $\lim_{n\to\infty} |\varphi^n(a)| = \infty$. It follows that φ is a substitution. In addition, we have

$$a \xrightarrow{\varphi} aw \xrightarrow{\varphi} aw\varphi(w) \xrightarrow{\varphi} aw\varphi(w)\varphi^2(w) \xrightarrow{\varphi} aw\varphi(w)\varphi^2(w)\varphi^3(w) \xrightarrow{\varphi} \cdots,$$

which implies that $\boldsymbol{u} = aw\varphi(w)\varphi^2(w)\varphi^3(w)\ldots$, so $\boldsymbol{u}$ is generated by φ. □

Proposition 3.43. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism and let $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{A}^{\mathbb{N}}$ be its fixed points. Then $\mathcal{L}(\boldsymbol{u}) = \mathcal{L}(\boldsymbol{v})$.*

Proof. If $\mathcal{A} = \{a\}$, then clearly $\boldsymbol{u} = aaa\ldots = \boldsymbol{v}$, so the statement of the proposition is trivially valid.

Assume that $\operatorname{card}(\mathcal{A}) > 1$ and let $w \in \mathcal{L}(\boldsymbol{u})$. We want to show that $w \in \mathcal{L}(\boldsymbol{v})$. By Proposition 3.42, $\boldsymbol{u}$ is generated by φ, and this means that there is $a \in \mathcal{A}$ and $k \in \mathbb{N}$ such that w is a factor of $\varphi^k(a)$. As φ is primitive, $a \in \mathcal{L}(\boldsymbol{v})$. We know that $\boldsymbol{v}$ is a fixed point of φ, and therefore also $\varphi^k(a) \in \mathcal{L}(\boldsymbol{v})$. It follows that $w \in \mathcal{L}(\boldsymbol{v})$.

Analogously, if we take $z \in \mathcal{L}(\boldsymbol{v})$, then also $z \in \mathcal{L}(\boldsymbol{u})$. Therefore $\mathcal{L}(\boldsymbol{u}) = \mathcal{L}(\boldsymbol{v})$. □

This proposition motivates a definition of the language of a primitive morphism, which is independent of a fixed point. However, to be able to define it also for a primitive morphism that does not have an infinite fixed point, we still need to prove some more properties of primitive morphisms.

Proposition 3.44. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism. Then φ^n is primitive for all $n \in \mathbb{N}$.*

Proof. As φ is primitive, there exists $k \in \mathbb{N}$ such that all elements of the matrix M_φ^k are positive. It also implies that there is no row or column in M_φ with zeroes only. Hence for all $m \geq k$, M_φ^m has all its elements positive. Now if we consider φ^n for some $n \in \mathbb{N}$, by Proposition 3.30, its incidence matrix satisfies $M_{\varphi^n} = M_\varphi^n$. Then the k-th power of this matrix is M_φ^{nk}, and it has all its elements positive, as $nk \geq k$. Therefore φ^n is primitive. □

Proposition 3.45. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism. Then there is $k \in \mathbb{N}$ such that φ^k has an infinite fixed point.*

Proof. If there is $a \in \mathcal{A}$ such that $a = \mathrm{fst}_\varphi(a)$, then φ has an infinite fixed point starting with a. Now suppose that $a \neq \mathrm{fst}_\varphi(a)$ for all $a \in \mathcal{A}$. Let $d = \mathrm{card}(\mathcal{A})$ and consider the sequence $\mathrm{fst}_\varphi(a)$, $\mathrm{fst}^2_\varphi(a)$, $\dots$, $\mathrm{fst}^{d+1}_\varphi(a)$ for some $a \in \mathcal{A}$. It follows that there exists $b \in \mathcal{A}$ which appears in this sequence at least twice. Hence there is $i, j \in \mathbb{N}$, $i < j$, such that

$$b = \mathrm{fst}^i_\varphi(a) = \mathrm{fst}^j_\varphi(a) = \mathrm{fst}^{j-i}_\varphi(\mathrm{fst}^i_\varphi(a)) = \mathrm{fst}^{j-i}_\varphi(b).$$

This implies that φ^{j-i} has an infinite fixed point starting with b. □

Proposition 3.46. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism, let $i, j \in \mathbb{N}$, let $\boldsymbol{u} \in \mathcal{A}^\mathbb{N}$ be a fixed point of φ^i and let $\boldsymbol{v} \in \mathcal{A}^\mathbb{N}$ be a fixed point of φ^j. Then $\mathcal{L}(\boldsymbol{u}) = \mathcal{L}(\boldsymbol{v})$.*

Proof. Consider the morphism φ^{ij}. By Proposition 3.44 it is primitive. We have

$$\varphi^{ij}(\boldsymbol{u}) = \varphi^{i(j-1)}(\varphi^i(\boldsymbol{u})) = \varphi^{i(j-1)}(\boldsymbol{u}) = \ldots = \varphi^i(\boldsymbol{u}) = \boldsymbol{u},$$

and similarly $\varphi^{ij}(\boldsymbol{v}) = \boldsymbol{v}$. Therefore $\boldsymbol{u}$ and $\boldsymbol{v}$ are fixed points of φ^{ij} and by Proposition 3.43 $\mathcal{L}(\boldsymbol{u}) = \mathcal{L}(\boldsymbol{v})$. □

Now we are ready to give the definition of the language of a primitive morphism.

Definition 3.47. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism. Then the language of φ, denoted as $\mathcal{L}(\varphi)$ is defined by $\mathcal{L}(\varphi) = \mathcal{L}(\boldsymbol{u})$, where $\boldsymbol{u} \in \mathcal{A}^\mathbb{N}$ satisfies that there exists $k \in \mathbb{N}$ such that $\varphi^k(\boldsymbol{u}) = \boldsymbol{u}$.*

Now we state the following propositions, which can be found in [5, p. 5].

Proposition 3.48. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism and let $\boldsymbol{u} \in \mathcal{A}^\mathbb{N}$ be its fixed point. Then $\boldsymbol{u}$ is uniformly recurrent.*

Proposition 3.49. *Let $\varphi, \psi\colon \mathcal{A}^* \to \mathcal{A}^*$ be conjugated primitive morphisms, let $\boldsymbol{u} \in \mathcal{A}^\mathbb{N}$ be fixed point of φ and $\boldsymbol{v} \in \mathcal{A}^\mathbb{N}$ be fixed point of ψ. Then $\mathcal{L}(\boldsymbol{u}) = \mathcal{L}(\boldsymbol{v})$.*

More general formulation of Proposition 3.49 can be found in [29, p. 205]. We state it below.

Proposition 3.50. *Let $\varphi, \psi\colon \mathcal{A}^* \to \mathcal{A}^*$ be conjugated primitive morphisms. Then $\mathcal{L}(\varphi) = \mathcal{L}(\psi)$.*

In the following examples, we show that the Fibonacci and the Thue-Morse morphisms are primitive.

Example 3.51. *Example 3.40 uses the Fibonacci morphism $\varphi_f\colon \{0,1\}^* \to \{0,1\}^*$ defined by $\varphi_f(0) = 01$, $\varphi_f(1) = 0$ to generate the Fibonacci word $\boldsymbol{f}$. Now we consider the incidence matrix of this morphism,*

$$M_{\varphi_f} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

We see that for this matrix, not all its elements are positive. However, if we take the second power of this matrix,

$$M^2_{\varphi_f} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix},$$

all of its elements are positive. Therefore, by Proposition 3.33, the Fibonacci morphism φ_f is primitive, and hence, by Proposition 3.48, the Fibonacci word $\boldsymbol{f}$ is uniformly recurrent.

Example 3.52. *As shown in Example 3.41, the Thue-Morse word $\boldsymbol{t}$ is generated by the Thue-Morse morphism $\varphi_t\colon \{0,1\}^* \to \{0,1\}^*$ defined by $\varphi_t(0) = 01$, $\varphi_t(1) = 10$. We see that for all $a, b \in \{0,1\}^*$ a is a factor of $\varphi_t(b)$, and so, by Definition 3.32, φ_t is primitive. Hence, by Proposition 3.48, the Thue-Morse word $\boldsymbol{t}$ is uniformly recurrent.*

3.3 Antimorphisms

After defining morphisms, we introduce another type of mapping called antimorphism.

Definition 3.53. *Let $H\colon \mathcal{A}^* \to \mathcal{A}^*$ be a mapping satisfying $H(uv) = H(v)H(u)$ for all $u, v \in \mathcal{A}^*$. Then H is called an antimorphism.*

Similarly to a morphism, to define an antimorphism, we only need to define the images of letters. The image of a word is then constructed as $H(w_1w_2\dots w_n) = H(w_n)H(w_{n-1})\dots H(w_1)$.

We are interested in finite words that are fixed points of some specific antimorphism H. We call such words H-palindromes and we introduce them in the next section. However, in order to have finite fixed points and to be able to work with them, we place an additional property on H. This property is that the composition of H with itself is the identity. Then we say that H is an involution. Below we illustrate why is this condition necessary.

Example 3.54. *Let $H\colon \mathcal{A}^* \to \mathcal{A}^*$ be an antimorphism and $w = w_1w_2\dots w_n \in \mathcal{A}^*$ its fixed point, i.e., $H(w_1w_2\dots w_n) = H(w_n)\dots H(w_2)H(w_1) = w_1w_2\dots w_n$. Applying H again and using the property of an antimorphism and the previous equality gives*

$$H(H(w_n)\dots H(w_2)H(w_1)) = H^2(w_1)H^2(w_2)\dots H^2(w_n) = H(w_1w_2\dots w_n) = w_1w_2\dots w_n.$$

So we see that for all $i \in \{1,\dots,n\}$, $H^2(w_i) = w_i$. We want to be able to construct fixed points from all letters of the alphabet $\mathcal{A}$, and therefore H has to satisfy $H^2(a) = a$ for all $a \in \mathcal{A}$.

Below, we give examples of commonly used antimorphisms. It is easy to check that they are also involutions.

Definition 3.55. *Let $\mathcal{A}$ be an alphabet. Then the antimorphism $R\colon \mathcal{A}^* \to \mathcal{A}^*$ defined by $R(a) = a$ for all $a \in \mathcal{A}$ is called the mirror image map R.*

Then for a word $w = w_1w_2\dots w_n$ we have $R(w_1w_2\dots w_n) = w_nw_{n-1}\dots w_1$, e.g. $R(\mathrm{BBAC}) = \mathrm{CABB}$.

Definition 3.56. *Let $\mathcal{A} = \{0,1\}$. Then the antimorphism $E\colon \mathcal{A}^* \to \mathcal{A}^*$ defined by $E(0) = 1$ and $E(1) = 0$ is called the exchange map E.*

We can also write it as $E(a) = 1 - a$ and then $E(w_1w_2\dots w_n) = (1 - w_n)(1 - w_{n-1})\dots(1 - w_1)$, e.g. $E(10111) = 00010$.

Definition 3.57. *Let $\mathcal{A} = \{\mathrm{A},\mathrm{C},\mathrm{G},\mathrm{T}\}$. Then we call the antimorphism $D\colon \mathcal{A}^* \to \mathcal{A}^*$ defined by $D(\mathrm{A}) = \mathrm{T}$, $D(\mathrm{C}) = \mathrm{G}$, $D(\mathrm{G}) = \mathrm{C}$, $D(\mathrm{T}) = \mathrm{A}$ the DNA map D.*

This map is called the DNA map as it is motivated by the structure of DNA, see Chapter 2.

In [43], the following antimorphisms were defined:

Definition 3.58. *Let $\mathcal{A} = \mathbb{Z}_m$, i.e., integers modulo m, where $m \in \mathbb{N}$. Then for each $x \in \mathcal{A}$, $\Psi_x\colon \mathcal{A}^* \to \mathcal{A}^*$ is the antimorphism defined by $\Psi_x(k) = x - k \mod m$ for all $k \in \mathcal{A}$.*

Example 3.59. *We will often work with the antimorphisms Ψ_0, Ψ_1, Ψ_2 over the alphabet $\mathcal{A} = \mathbb{Z}_3$. They are of the form*

$$\begin{array}{lll}
\Psi_0(0) = 0 & \Psi_1(0) = 1 & \Psi_2(0) = 2\\
\Psi_0(1) = 2 & \Psi_1(1) = 0 & \Psi_2(1) = 1\\
\Psi_0(2) = 1 & \Psi_1(2) = 2 & \Psi_2(2) = 0.
\end{array}$$

3.3.1 Palindromes and palindromic words

Definition 3.60. *Let $\mathcal{A}$ be an alphabet, $H\colon \mathcal{A}^* \to \mathcal{A}^*$ an involutive antimorphism, $w \in \mathcal{A}^*$ and $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$. Then w is called an H-palindrome if $H(w) = w$ and $\boldsymbol{u}$ is called an H-palindromic word if $\mathcal{L}(\boldsymbol{u})$ contains an infinite number of H-palindromes.*

Note that the empty word ε is an H-palindrome for every H.

This is a general definition with any suitable H. The most common types of palindromes are derived from antimorphisms R, E and D. Some examples of R-palindromes are the words MADAM, NOON, 010, CGTTGC and also every word that consists of only one letter. In the case of the alphabet $\mathcal{A} = \{0, 1\}$, examples of E-palindromes are the words 01, 0011 or 110100. Finally, for $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ we have D-palindromes, and some of them are the words AT, GCGC or CGGTACCG.

Similarly to factor complexity, we define H-palindromic complexity of an infinite word and evaluate it using H-palindromic extensions, following [36].

Definition 3.61. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ and let $H\colon \mathcal{A}^* \to \mathcal{A}^*$ be an involutive antimorphism. Then we define the H-palindromic complexity of $\boldsymbol{u}$ as the function $p^H_{\boldsymbol{u}}\colon \mathbb{N} \to \mathbb{N}$ given by $p^H_{\boldsymbol{u}}(n) = \mathrm{card}(\{w \in \mathcal{L}_n(\boldsymbol{u}) \mid w = H(w)\})$, that is the number of H-palindromic factors in $\boldsymbol{u}$ of length n.*

Definition 3.62. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$, let $H\colon \mathcal{A}^* \to \mathcal{A}^*$ be an involutive antimorphism and $w \in \mathcal{L}(\boldsymbol{u})$ an H-palindrome. Then we define the set of H-palindromic extensions of w as $\mathrm{Pext}^H_{\boldsymbol{u}}(w) = \{a \in \mathcal{A} \mid awH(a) \in \mathcal{L}(\boldsymbol{u})\}$.*

We use these definitions in Chapter 6. When we consider R-palindromes, also simply called palindromes, we omit R in the notation and terminology and use $p_{\boldsymbol{u}}$ for palindromic complexity and $\mathrm{Pext}_{\boldsymbol{u}}(w)$ for the set of palindromic extensions of w.

Proposition 3.63. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ and let $H\colon \mathcal{A}^* \to \mathcal{A}^*$ be an involutive antimorphism. Then the H-palindromic complexity of $\boldsymbol{u}$ satisfies*

$$p^H_{\boldsymbol{u}}(n+2) = \sum_{\substack{w \in \mathcal{L}_n(\boldsymbol{u}) \\ w = H(w)}} \mathrm{card}(\mathrm{Pext}^H_{\boldsymbol{u}}(w)).$$

Later, we will need the following simple observation.

Proposition 3.64. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and $w \in \mathcal{A}^*$. Then w is both an R-palindrome and a D-palindrome if and only if $w = \varepsilon$.*

Proof. It is clear that the empty word is both an R-palindrome and a D-palindrome. Now suppose that $w \neq \varepsilon$ is both an R-palindrome and a D-palindrome. Let us denote the first letter of w as a and the last letter of w as b. w being an R-palindrome implies that $a = R(b) = b$. Moreover, w being a D-palindrome implies that $a = D(b) \neq b$, which is a contradiction. □

Proposition 3.65. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$. If $\boldsymbol{u}$ is uniformly recurrent and H-palindromic, then the language of $\boldsymbol{u}$ is closed under the antimorphism H, i.e., if $w \in \mathcal{L}(\boldsymbol{u})$ then $H(w) \in \mathcal{L}(\boldsymbol{u})$.*

Proof. Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be uniformly recurrent and H-palindromic. We take some $w \in \mathcal{L}(\boldsymbol{u})$ and we want to show that $H(w) \in \mathcal{L}(\boldsymbol{u})$. Set $n = |w|$ and as $\boldsymbol{u}$ is uniformly recurrent, by Definition 3.7 we have a number $r \in \mathbb{N}$ satisfying that every factor of $\boldsymbol{u}$ of length r contains all factors of length n, and therefore also the factor w. Using that $\boldsymbol{u}$ is H-palindromic, we can find an H-palindrome v of length greater of equal to n. Because of its length, v has to contain w, so there exist $x, y \in \mathcal{A}^*$ such that $v = xwy$. We know that $v = H(v) = H(y)H(w)H(x)$, so $H(w)$ is also a factor of v and therefore $H(w) \in \mathcal{L}(\boldsymbol{u})$. □

Now we state a result about an H-palindromic word, which is purely periodic.

Proposition 3.66. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be H-palindromic and purely periodic. Then there exist H-palindromes $u, v \in \mathcal{A}^*$ such that $\boldsymbol{u} = (uv)^\infty$.*

Proof. As $\boldsymbol{u}$ is purely periodic, there is a non-empty word w such that $\boldsymbol{u} = w^\infty$. Any factor of $\boldsymbol{u}$ is of the form $xw^k y$, where x is a proper suffix of w, y is a proper prefix of w and $k \in \mathbb{N}_0$. Now suppose that $xw^k y$, where $k \geq 2$, is an H-palindrome. This implies that

$$xw^k y = H(y)H(w)^k H(x).$$

Because $|x| < |wy|$, we have that $H(x)$ is a proper suffix of wy and this means that there exists a non-empty word $s = w^n z$, where $n \in \mathbb{N}$ and z is a proper prefix of w, such that

$$w^k y = sH(x).$$

It is clear that s is a prefix of $\boldsymbol{u}$. We know that $xw^k y = xsH(x)$ is an H-palindrome, therefore s is an H-palindrome.

Because $\boldsymbol{u}$ is H-palindromic, there are infinitely many factors of $\boldsymbol{u}$ that are H-palindromes, hence there are also infinitely many prefixes of $\boldsymbol{u}$ that are H-palindromes. So consider an H-palindrome $w^n p$, where $n \in \mathbb{N}$ and p is a proper prefix of w. It follows that there is a non-empty word q such that $w = pq$. Then $w^n p = (pq)^n p$ is an H-palindrome, hence

$$(pq)^n p = H(p)(H(q)H(p))^n,$$

which implies that $p = H(p)$ and $q = H(q)$, so p, q are H-palindromes. If we denote $u = p$ and $v = q$, then $\boldsymbol{u} = (uv)^\infty$, where u, v are H-palindromes. ☐

3.4 Groups of morphisms and antimorphisms

In this section we consider finite groups of morphisms and antimorphisms on $\mathcal{A}^*$. Following [36], we denote the set of all morphisms and antimorphisms on $\mathcal{A}^*$ as $AM(\mathcal{A}^*)$ and we consider a subset G of $AM(\mathcal{A}^*)$ satisfying that G is a finite group and G contains at least one antimorphism. The first condition implies that every element ψ of G is non-erasing (since otherwise ψ has no inverse in G) and for all $a \in \mathcal{A}$, $\psi(a) \in \mathcal{A}$ (since otherwise $\psi^n \neq \mathrm{Id}$ for all $n \in \mathbb{N}$). Also for all $b \in \mathcal{A}$ there is $a \in \mathcal{A}$ such that $b = \psi(a)$. Overall, ψ restricted to $\mathcal{A}$ must be a permutation of $\mathcal{A}$.

In section 5.2, we discuss a more general concept of palindromicity with respect to a group G, where only involutive antimorphisms in G are relevant. So for this purpose, we denote the set of all involutive antimorphisms in G by G^{inv}. Often, we consider a group G which is generated by a subset S of $AM(\mathcal{A}^*)$ and we write it as $G = \langle S \rangle$.

Example 3.67. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and consider the group $G = \langle\{R, D\}\rangle$. Then $G = \{\mathrm{Id}, R, D, R \circ D\}$ and $G^{\mathrm{inv}} = \{R, D\}$.*

Example 3.68. *Let $\mathcal{A} = \mathbb{Z}_m$. As given in [43], we can take the group*

$$G = \{\Psi_x \mid x \in \mathbb{Z}_m\} \cup \{\Pi_x \mid x \in \mathbb{Z}_m\},$$

where Ψ_x, $x \in \mathbb{Z}_m$, are the antimorphisms from Definition 3.58 given by $\Psi_x(k) = x - k \mod m$ for all $k \in \mathbb{Z}_m$, and Π_x, $x \in \mathbb{Z}_m$, are morphisms defined by $\Pi_x(k) = x + k \mod m$ for all $k \in \mathbb{Z}_m$. It can be shown that this group is isomorphic to the dihedral group of order $2m$.

In the case of $m = 3$, we have the group $G = \{\Psi_0, \Psi_1, \Psi_2, \Pi_0, \Pi_1, \Pi_2\}$, where the antimorphisms Ψ_0, Ψ_1, Ψ_2 are given by

$$\begin{array}{lll} \Psi_0(0) = 0 & \Psi_1(0) = 1 & \Psi_2(0) = 2 \\ \Psi_0(1) = 2 & \Psi_1(1) = 0 & \Psi_2(1) = 1 \\ \Psi_0(2) = 1 & \Psi_1(2) = 2 & \Psi_2(2) = 0 \end{array}$$

and the morphisms Π_0, Π_1, Π_2 are given by

$$\begin{array}{lll} \Pi_0(0) = 0 & \Pi_1(0) = 1 & \Pi_2(0) = 2 \\ \Pi_0(1) = 1 & \Pi_1(1) = 2 & \Pi_2(1) = 0 \\ \Pi_0(2) = 2 & \Pi_1(2) = 0 & \Pi_2(2) = 1. \end{array}$$

We see that Π_0 is the identity and $\Pi_2 = \Pi_1^{-1}$. So if we denote $\Pi_1 = \mu$, we can write $G = \{\mathrm{Id}, \mu, \mu^{-1}, \Psi_0, \Psi_1, \Psi_2\}$. Then $G^{\mathrm{inv}} = \{\Psi_0, \Psi_1, \Psi_2\}$ and the Cayley table of G is shown below, where for each $\theta \in G$ labelling a row and each $\sigma \in G$ labelling a column their composition $\theta \circ \sigma$ is written:

$\circ$	Id	μ	μ^{-1}	Ψ_0	Ψ_1	Ψ_2
Id	Id	μ	μ^{-1}	Ψ_0	Ψ_1	Ψ_2
μ	μ	μ^{-1}	Id	Ψ_1	Ψ_2	Ψ_0
μ^{-1}	μ^{-1}	Id	μ	Ψ_2	Ψ_0	Ψ_1
Ψ_0	Ψ_0	Ψ_2	Ψ_1	Id	μ^{-1}	μ
Ψ_1	Ψ_1	Ψ_0	Ψ_2	μ	Id	μ^{-1}
Ψ_2	Ψ_2	Ψ_1	Ψ_0	μ^{-1}	μ	Id

Definition 3.69. *A word $w \in \mathcal{A}^*$ is called a G-palindrome if there exists an antimorphism $\theta \in G$ such that $w = \theta(w)$.*

Definition 3.70. *Let $u, v \in \mathcal{A}^*$. We say that u and v are G-equivalent, denoted by $u \sim_G v$, if there exists $\sigma \in G$ such that $u = \sigma(v)$. This is an equivalence relation and the class of equivalence containing u is denoted $[u]_G$.*

Definition 3.71. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$. The language of $\boldsymbol{u}$ is said to be closed under G if for all $\sigma \in G$ we have $w \in \mathcal{L}(\boldsymbol{u}) \implies \sigma(w) \in \mathcal{L}(\boldsymbol{u})$.*

When we discuss palindromic richness of an infinite word $\boldsymbol{u}$ with respect to a group G in Chapter 6, we need the definition of graph of symmetries of the word $\boldsymbol{u}$, introduced in [35]. Firstly, we give the definition of the directed graph of symmetries, again based on [35]. These concepts are derived from the Rauzy graph defined in Definition 3.15.

Definition 3.72. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ such that $\mathcal{L}(\boldsymbol{u})$ us closed under G and let $n \in \mathbb{N}$. The directed graph of symmetries of $\boldsymbol{u}$ of order n, denoted by $\overrightarrow{\Gamma}_n(\boldsymbol{u})$, is the directed graph $\overrightarrow{\Gamma}_n(\boldsymbol{u}) = (V, \overrightarrow{E})$ with the set of vertices V and the set of edges $\overrightarrow{E}$ given by the following:*

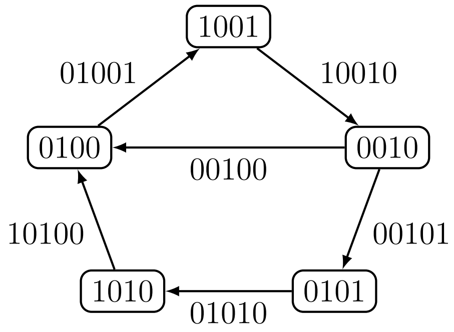
1. *$V = \{[w]_G \mid w \in \mathcal{L}_n(\boldsymbol{u}),\ w$ is special$\}$,*
2. *$e \in \mathcal{L}(\boldsymbol{u})$ is an edge from vertex $[u]_G$ to vertex $[v]_G$ if*
   - *the prefix of e of length n belongs to $[u]_G$,*
   - *the suffix of e of length n belongs to $[v]_G$,*

- *$|e| \geq n+1$ and all factors of e of length n except for its prefix and suffix are not special.*

Definition 3.73. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ such that $\mathcal{L}(\boldsymbol{u})$ us closed under G, let $n \in \mathbb{N}$ and let $\overrightarrow{\Gamma}_n(\boldsymbol{u}) = (V, \overrightarrow{E})$ be the directed graph of symmetries of $\boldsymbol{u}$ of order n. The graph of symmetries of $\boldsymbol{u}$ of order n, denoted by $\overline{\Gamma}_n(\boldsymbol{u})$, is the undirected graph $\overline{\Gamma}_n(\boldsymbol{u}) = (V, E)$ with the same set of vertices as $\overrightarrow{\Gamma}_n(\boldsymbol{u})$ and the set of edges E given by the following:*

- *for $e \in \mathcal{L}(\boldsymbol{u})$, $[e]_G$ is an edge in E joining vertices $[u]_G$ and $[v]_G$ if and only if e is an edge in $\overrightarrow{E}$ from $[u]_G$ to $[v]_G$ or vice versa.*

Example 3.74. *Let $G = \{\mathrm{Id}, R\}$ and consider the Fibonacci word $\boldsymbol{f}$. Its Rauzy graph Γ_4 was constructed in Example 3.20. It is shown again below:*



We see from the Rauzy graph that $\mathcal{L}_4(\boldsymbol{f}) = \{1001, 0100, 0010, 1010, 0101\}$. There are only two special words in $\mathcal{L}_4(\boldsymbol{f})$, namely 0100 and 0010. These correspond to vertices in Γ_4 that have at least two incoming edges or at least two outgoing edges. Moreover, $0100 \sim_G 0010$, and we denote their class of equivalence $[0100]$. Hence, $[0100]$ is the only vertex of the graphs $\overrightarrow{\Gamma}_4(\boldsymbol{f})$ and $\overline{\Gamma}_4(\boldsymbol{f})$. The edges in the graph $\overrightarrow{\Gamma}_4(\boldsymbol{f})$ are the words 010010, 00100 and 0010100 and they correspond to paths in Γ_4 that start and end in a special word and pass through non-special words only. In $\overline{\Gamma}_4(\boldsymbol{f})$, the edges are the classes of equivalence of the words 010010, 00100 and 0010100, i.e., $[010010]$, $[00100]$ and $[0010100]$. The graphs $\overrightarrow{\Gamma}_4(\boldsymbol{f})$ and $\overline{\Gamma}_4(\boldsymbol{f})$ are shown in Figures 3.2 and 3.3, respectively.

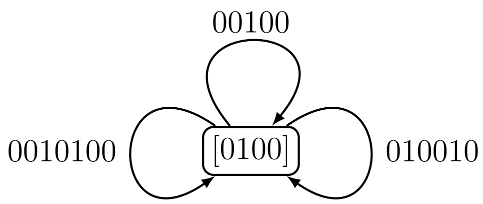


Figure 3.2: The directed graph of symmetries $\overrightarrow{\Gamma}_4(\boldsymbol{f})$.

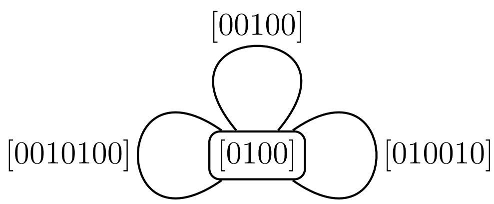


Figure 3.3: The graph of symmetries $\overline{\Gamma}_4(\boldsymbol{f})$.

Chapter 4

Word equations with palindromes

In this chapter, we derive several results concerning word equations with R-palindromes and D-palindromes. But first, we need to state two basic propositions regarding word equations, which can be found in [31, p. 8].

Proposition 4.1. *Let $x, y \in \mathcal{A}^*$ be non-empty words. Then $xy = yx$ if and only if there exists a non-empty word $w \in \mathcal{A}^*$ and $i, j \in \mathbb{N}$ such that*

$$x = w^i, \quad y = w^j.$$

Proposition 4.2. *Let $x, y, z \in \mathcal{A}^*$ be words, where x, y are non-empty. Then equality $xz = zy$ holds if and only if there exist $u, v \in \mathcal{A}^*$ and $i \in \mathbb{N}_0$ such that*

$$x = uv, \quad y = vu, \quad z = (uv)^i u.$$

Now, we present our proofs of three propositions. Proposition 4.3 is needed in Chapter 5, more specifically in the proof of Theorem 5.55. In fact, a more general version of Proposition 4.4 can be found in [26, Proposition 9].

Proposition 4.3. *Let $w, x, y, z \in \mathcal{A}^*$ be non-empty R-palindromes satisfying $wx = yz$. Then there exist R-palindromes u, v and $i, j, k, l \in \mathbb{N}_0$ such that*

$$w = (uv)^i u, \quad x = v(uv)^j, \quad y = (uv)^k u, \quad z = v(uv)^l.$$

Proof. The proof is by induction on the value $||w| - |y||$.

In the first step, we consider $||w| - |y|| = 0$. This means that $|w| = |y|$. Then equality $wx = yz$ implies that $w = y$ and $x = z$. Hence, if we take $i = j = k = l = 0$, $u = w$ and $v = x$ the statement holds.

In the second step, we consider $||w| - |y|| = N$ for some $N \in \mathbb{N}$ and we assume that the statement holds for all $K \in \mathbb{N}_0$ such that $K < N$. Without loss of generality, we assume that $|w| > |y|$. Then equality $wx = yz$ implies that there is some non-empty word s such that $w = ys$ and $z = sx$. Using that w, x, y, z are R-palindromes, we get

$$R(s)y = ys, \quad sx = xR(s).$$

All the words occurring in those equalities are non-empty, and so we can use Proposition 4.2 on each equality to get that there exist $n, m \in \mathbb{N}_0$ and $a, b, c, d \in \mathcal{A}^*$ such that

$$\begin{aligned} R(s) &= ab & s &= cd \\ s &= ba & R(s) &= dc \\ y &= (ab)^n a & x &= (cd)^m c. \end{aligned}$$

From the first two lines above we have

$$ab = R(s) = R(a)R(b), \quad dc = R(s) = R(d)R(c),$$

which implies that a, b, c, d are R-palindromes. Combining the results from above we obtain

$$\begin{aligned} y &= (ab)^n a & x &= (cd)^m c \\ w &= ys = (ab)^{n+1} a & z &= sx = (cd)^{m+1} c. \end{aligned}$$

Moreover, we know that $s = ba = cd$ and $|w| - |y| = |s|$.

Now assume that all the words a, b, c, d are non-empty. Then

$$N = |w| - |y| > \max\{|b|, |c|\} > ||b| - |c||,$$

and so from our assumption that the statement of the lemma holds for all $K < N$, we have that the equality $ba = cd$ implies that there exist R-palindromes p, q and $e, f, g, h \in \mathbb{N}_0$ such that

$$b = (pq)^e p, \quad a = q(pq)^f, \quad c = (pq)^g p, \quad d = q(pq)^h.$$

Substituting this into the expressions for w, x, y, z gives

$$\begin{aligned} w &= (ab)^{n+1} a = (qp)^{(e+f+1)(n+1)+f} q \\ x &= (cd)^m c = p(qp)^{(g+h+1)m+g} \\ y &= (ab)^n a = (qp)^{(e+f+1)n+f} q \\ z &= (cd)^{m+1} c = p(qp)^{(g+h+1)(m+1)+g}. \end{aligned}$$

Hence, if we take $u = q$, $v = p$ and

$$\begin{aligned} i &= (e + f + 1)(n + 1) + f \\ j &= (g + h + 1)m + g \\ k &= (e + f + 1)n + f \\ l &= (g + h + 1)(m + 1) + g \end{aligned}$$

the statement holds.

It remains to discuss the case when at least one of the words a, b, c, d is empty. We know that $ba = s = cd$ and $ab = R(s) = dc$. Without loss of generality, assume that $a = \varepsilon$. Then $b = cd = dc$.

If c, d are non-empty, we can use Proposition 4.1 to deduce that there exists a non-empty word r and $\alpha, \beta \in \mathbb{N}$ such that

$$c = r^\alpha, \quad d = r^\beta.$$

Because c and d are R-palindromes, r is also an R-palindrome. Substituting this result into the expressions for w, x, y, z gives

$$\begin{aligned} w &= (ab)^{n+1} a = b^{n+1} = r^{(\alpha+\beta)(n+1)} \\ x &= (cd)^m c = r^{(\alpha+\beta)m+\alpha} \\ y &= (ab)^n a = b^n = r^{(\alpha+\beta)n} \\ z &= (cd)^{m+1} c = r^{(\alpha+\beta)(m+1)+\alpha}. \end{aligned}$$

Hence, if we take $u = r$, $v = \varepsilon$ and

$$\begin{aligned} i &= (\alpha+\beta)(n+1) - 1 \\ j &= (\alpha+\beta)m + \alpha \\ k &= (\alpha+\beta)n - 1 \\ l &= (\alpha+\beta)(m+1) + \alpha \end{aligned}$$

the statement holds.

In the case that also one of c, d is empty, we assume, without loss of generality, that $c = \varepsilon$. Then $b = d$. Therefore

$$\begin{aligned} w &= (ab)^{n+1}a = b^{n+1} \\ x &= (cd)^m c = b^m \\ y &= (ab)^n a = b^n \\ z &= (cd)^{m+1}c = b^{m+1}. \end{aligned}$$

Hence, if we take $u = b$, $v = \varepsilon$, $i = n$, $j = m$, $k = n - 1$ and $l = m + 1$ the statement holds. □

Proposition 4.4. *Let* $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ *and let* $x, y \in \mathcal{A}^*$ *be non-empty words satisfying that* xy *and* yx *are* D*-palindromes. Then there is a non-empty word* $w \in \mathcal{A}^*$, $i, j \in \mathbb{N}_0$ *and* $m \in \{0, 1\}$ *such that*

$$x = (wD(w))^i w^m, \quad y = D(w)^m (wD(w))^j.$$

Proof. The proof is by induction on the value $||x| - |y||$.

In the first step, we consider $||x| - |y|| = 0$. This means that $|x| = |y|$. As xy is a D-palindrome, $xy = D(y)D(x)$. Because $|x| = |D(y)|$, we have $x = D(y)$. This is equivalent to $y = D(x)$. Hence, if we take $w = x$, $i = j = 0$ and $m = 1$ the statement holds.

In the second step, we consider $||x| - |y|| = N$ for some $N \in \mathbb{N}$ and we assume that the statement holds for all $K \in \mathbb{N}_0$ such that $K < N$. Without loss of generality, we assume that $|x| > |y|$. From the equality $xy = D(y)D(x)$ we have that there is a non-empty word u such that $|u| = N$ and $x = D(y)u$. It also holds that $D(x) = uy$. Hence, $uy = D(x) = D(u)y$, and so u is a D-palindrome. As yx is a D-palindrome, $yx = D(x)D(y)$. This implies that there is a non-empty word v such that $|v| = N$ and $x = vD(y)$. It also holds that $D(x) = yv$. Hence, $yv = D(x) = yD(v)$, and so v is a D-palindrome. Moreover, we got two different expressions for x, and so we have

$$vD(y) = D(y)u.$$

Applying Proposition 4.2 gives that there exist words $p, q \in \mathcal{A}^*$ and $k \in \mathbb{N}_0$ such that

$$v = pq, \quad u = qp, \quad D(y) = (pq)^k p.$$

We see that pq and qp are D-palindromes and $N = |v| = |pq| = |p| + |q|$.

Assume that both p and q are non-empty. Then

$$N = |p| + |q| > \max\{|p|, |q|\} > ||p| - |q||,$$

and so from our assumption that the statement of the lemma holds for all $K < N$, the fact that p, q are non-empty and pq and qp are D-palindromes implies that there exist a non-empty word $r \in \mathcal{A}^*$, $l, s \in \mathbb{N}_0$ and $n \in \{0, 1\}$ such that

$$p = (rD(r))^l r^n, \quad q = D(r)^n (rD(r))^s.$$

Substituting this into the expression for $D(y)$ gives

$$D(y) = (pq)^k p = (rD(r))^{(l+n+s)k+l} r^n.$$

Hence,

$$x = vD(y) = pqD(y) = (rD(r))^{(l+n+s)(k+1)+l} r^n y = D(r)^n (rD(r))^{(l+n+s)k+l}.$$

Therefore, if we take $w = r$, $m = n$ and

$$i = (l + n + s)(k + 1) + l$$
$$j = (l + n + s)k + l$$

the statement holds.

Now we consider the case when one of the words p and q is empty. Without loss of generality, we assume that $p = \varepsilon$. Then q is a D-palindrome, therefore there is a non-empty word d such that $q = dD(d)$ and

$$D(y) = (pq)^k p = (dD(d))^k.$$

Hence,

$$x = vD(y) = pqD(y) = (dD(d))^{k+1}$$
$$y = (dD(d))^k.$$

Therefore, if we take $w = d$, $i = k + 1$, $j = k$ and $m = 0$ the statement holds. □

Proposition 4.5. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and let $x, y \in \mathcal{A}^*$ be non-empty R-palindromes satisfying that*

$$(D(x)x)^i = (D(y)y)^j$$

for some $i, j \in \mathbb{N}$. Then there is a non-empty R-palindrome $w \in \mathcal{A}^*$ and $k, l \in \mathbb{N}_0$ such that

$$x = w(D(w)w)^k, \quad y = w(D(w)w)^l.$$

Proof. Without loss of generality, assume $|x| \geq |y|$. The proof is by induction on the value $|y| \geq 1$.

In the first step, we consider $|y| = 1$. Then $(D(x)x)^i = (D(y)y)^j$ implies that $D(x)$ is a non-empty prefix of $(D(y)y)^j$. Because both y and $D(y)$ contain only one letter, $D(x)$ is either of the form $D(x) = (D(y)y)^\alpha$, where $\alpha \in \mathbb{N}$, or $D(x) = (D(y)y)^\beta D(y)$, where $\beta \in \mathbb{N}_0$. Consider the first case. Then $x = (D(y)y)^\alpha = D(x)$ and therefore x is a D-palindrome. However, x is also an R-palindrome and by Proposition 3.64 this is possible if and only if x is empty. But that is in contradiction with the assumption that x is non-empty. Hence, x cannot be of such a form and the second case holds, so we have $D(x) = (D(y)y)^\beta D(y)$. Then $x = y(D(y)y)^\beta$. So if we take $w = y$, which is an R-palindrome, $k = \beta$ and $l = 0$ the statement of the lemma holds.

In the second step, we consider $|y| = N \in \mathbb{N}$, $N \geq 2$, and we assume that the statement holds for all $K \in \mathbb{N}$ such that $K < N$. Then $|x| \geq |y| = N$ and $(D(x)x)^i = (D(y)y)^j$ implies that

$$D(x)x = (D(y)y)^m d,$$

where $m \in \mathbb{N}$ and $d \in \mathcal{A}^*$ with $0 \leq |d| < 2\,|y|$.

Consider the case when $d = \varepsilon$. Then

$$D(x)x = (D(y)y)^m.$$

If m is even, then $D(x) = x = (D(y)y)^{m/2}$ and x is a D-palindrome. This again gives contradiction, as x is also a non-empty R-palindrome. Therefore m is odd and

$$x = y(D(y)y)^{(m-1)/2}.$$

So if we take $w = y$, which is an R-palindrome, $k = (m-1)/2$ and $l = 0$ the statement holds.

Now consider the case when $d \neq \varepsilon$, so $D(x)x = (D(y)y)^m d$, where $0 < |d| < 2|y|$. This implies that

$$2|x| = 2|y|m + |d|,$$

and so

$$\frac{|x|}{|y|} = m + r, \tag{4.1}$$

where

$$r = \frac{|d|}{2|y|} < 1.$$

The equality $(D(x)x)^i = (D(y)y)^j$ implies that

$$2|x|i = 2|y|j,$$

and so

$$\frac{|x|}{|y|} = \frac{j}{i}. \tag{4.2}$$

Hence, from equations (4.1) and (4.2), we have

$$j = mi + ri. \tag{4.3}$$

Here, $ri \in \mathbb{N}$, and because $r < 1$ it implies that $i \geq 2$. It also means that $j > i \geq 2$. Then we have

$$((D(y)y)^m d)^i = (D(x)x)^i = (D(y)y)^j = ((D(y)y)^m)^i (D(y)y)^{ri}.$$

Let us denote $z = (D(y)y)^m$. Then we have equality

$$\underbrace{(zd)(zd)\dots(zd)}_{i\text{-times}} = \underbrace{zzzz\dots z}_{i\text{-times}}(D(y)y)^{ri}.$$

This implies that

$$d\,\underbrace{(zd)\dots(zd)}_{(i-1)\text{-times}} = \underbrace{zzz\dots z}_{(i-1)\text{-times}}(D(y)y)^{ri}. \tag{4.4}$$

We see from equality (4.4) that d is a prefix of $z = (D(y)y)^m$, and hence d is also a prefix of $(D(y)y)^{ri}$. In the case of $i = 2$, it follows that dz is a prefix of $z(D(y)y)^{ri}$ and at the same time zd is a prefix of $z(D(y)y)^{ri}$. In the case of $i > 2$ it follows that dz is a prefix of zz and at the same time zd is a prefix of zz. In both cases, zd and dz are prefixes of the same word and they also have the same length. This implies that $zd = dz$. Using this relation, we can rewrite equality (4.4) as

$$z^{i-1}d^i = z^{i-1}(D(y)y)^{ri}.$$

Therefore
$$d^i = (D(y)y)^{ri} \tag{4.5}$$
for all possible $i \in \mathbb{N}$.

Returning to the equality
$$D(x)x = (D(y)y)^m d, \tag{4.6}$$
we define words $a, b, c \in \mathcal{A}^*$ in the following way. We take a to be the prefix of the word $(D(y)y)^m$ of half of its length, so $|a| = \frac{1}{2}|(D(y)y)^m| = |y|\, m$ and a is non-empty. Then
$$|a| = \frac{1}{2}|(D(y)y)^m| < \frac{1}{2}|(D(y)y)^m d| = |D(x)|,$$
and so a is also a proper prefix of $D(x)$. Therefore there is a non-empty word b such that
$$D(x) = ab. \tag{4.7}$$
We know that $0 < |d| < 2|y| \leq |(D(y)y)^m|$, and hence
$$|d| < \frac{1}{2}(|(D(y)y)^m| + |d|) = \frac{1}{2}|(D(y)y)^m d| = \frac{1}{2}|D(x)x| = |x|.$$
This implies that d is a proper suffix of x, and therefore there is a non-empty word c such that
$$x = cd. \tag{4.8}$$
Substituting this into equality (4.6), we get
$$abcd = (D(y)y)^m d,$$
and hence
$$(D(y)y)^m = abc.$$

If m is even, we have
$$a = (D(y)y)^{m/2}, \quad bc = (D(y)y)^{m/2},$$
which implies that $a = D(bc) = D(c)D(b)$.

If m is odd, we have
$$a = (D(y)y)^{(m-1)/2}D(y), \quad bc = y(D(y)y)^{(m-1)/2},$$
which again implies that $a = D(bc) = D(c)D(b)$.

Therefore for all possible m we have
$$a = D(c)D(b). \tag{4.9}$$

Combining equations (4.7) and (4.8) gives
$$ab = D(cd) = D(d)D(c). \tag{4.10}$$

Substituting expression for a from equation (4.9) into equation (4.10) gives
$$D(d)D(c) = D(c)D(b)b.$$
We know that $D(d)$ and $D(b)b$ are non-empty words, and therefore Proposition 4.2 implies that there exist words $u, v \in \mathcal{A}^*$ and $n \in \mathbb{N}$ such that
$$D(d) = uv, \quad D(b)b = vu, \quad D(c) = (uv)^n u. \tag{4.11}$$

We see that $vu = D(b)b$ is a D-palindrome. From equations (4.11) and (4.10) we have

$$d = D(v)D(u), \quad ab = D(d)D(c) = (uv)^{n+1}u. \tag{4.12}$$

From above, we know that d is a prefix of $(D(y)y)^m$ and therefore it is a prefix of ab. We see from equation (4.12) that the prefix of ab of length $|d|$ is uv. Therefore

$$d = D(v)D(u) = uv,$$

which implies that uv is a D-palindrome. Substituting equations (4.11) into equation (4.8) gives

$$x = cd = D(u)(D(v)D(u))^{n+1}. \tag{4.13}$$

Now suppose that one of the words u and v is empty. Without loss of generality, assume $u = \varepsilon$. Then, as vu is a D-palindrome, v is a D-palindrome and equation (4.13) becomes

$$x = v^{n+1}.$$

We know that x is an R-palindrome, which means that v is also an R-palindrome. By Proposition 3.64, $v = \varepsilon$, which is a contradiction. Therefore, both u and v are non-empty.

Now, as uv and vu are both D-palindromes and u, v are non-empty, we can use Proposition 4.4 to deduce that there is a non-empty word $p \in \mathcal{A}^*$, $e, f \in \mathbb{N}$ and $g \in \{0, 1\}$ such that

$$u = (pD(p))^e p^g, \quad v = D(p)^g (pD(p))^f. \tag{4.14}$$

If we substitute this into equation (4.13), we get

$$x = D(u)(D(v)D(u))^{n+1} = D(p)^g (pD(p))^{(e+f+g)(n+1)+e}. \tag{4.15}$$

If $g = 0$, then x being an R-palindrome implies that $pD(p)$ is an R-palindrome. However, it as also a D-palindrome, and so by Proposition 3.64 it is empty, which is a contradiction.

Therefore $g = 1$. Then $x = R(x)$ if and only if $D(p)$ and p are R-palindromes, which happens if and only if p is an R-palindrome. Hence, p is an R-palindrome.

Next, we combine equation (4.5) with equations (4.12) and (4.14) to get

$$(D(y)y)^{ri} = d^i = (D(v)D(u))^i = (pD(p))^{(e+f+g)i}.$$

Let $q = D(p)$, so q is also an R-palindrome. Then substituting this into the equation above gives

$$(D(y)y)^{ri} = (D(q)q)^{(e+f+g)i}. \tag{4.16}$$

We know that $r < 1 \leq e+f+g$. If we consider the lengths of the words in equation (4.16), we have

$$2\,|y|\,ri = 2\,|q|\,(e + f + g)i.$$

This implies that

$$\frac{|y|}{|q|} = \frac{e + f + g}{r} > 1,$$

and hence $|q| < |y| = N$. Therefore, from our assumption that the statement of the lemma holds for all $K < N$, the equation (4.16) implies that there is a non-empty R-palindrome $s \in \mathcal{A}^*$ and $h, t \in \mathbb{N}_0$ such that

$$y = s(D(s)s)^h$$
$$q = s(D(s)s)^t.$$

Then substituting this and $q = D(p)$ into equation (4.15) gives

$$x = q(D(q)q)^{(e+f+g)(n+1)+e} = s(D(s)s)^{(2t+1)((e+f+g)(n+1)+e)+t}.$$

Therefore, if we take $w = s$, $k = (2t+1)((e+f+g)(n+1)+e)+t$ and $l = h$ the statement holds. $\square$

Chapter 5

General palindromicity

In this chapter, we examine in more details the topic of general palindromicity. In the classical sense, an infinite word is palindromic if its language contains infinitely many R-palindromes. This can be generalized to any involutive antimorphism H, and H-palindromic words were defined in Definition 3.60. Section 5.1 focuses on this concept of H-palindromicity, more specifically on classes of morphisms generating H-palindromic words. We summarize some known results regarding R-palindromic and E-palindromic words and then we study D-palindromic words and general H-palindromic words.

Another level of generalization is with respect to a group of morphisms and antimorphisms G and this is discussed in section 5.2.

5.1 Palindromicity with respect to an antimorphism

5.1.1 Mirror image map R

In [23], authors Hof, Knill and Simon employed R-palindromic words in the study of discrete Schrödinger operators. In order to generate R-palindromic words, they defined a class of morphisms called class $\mathcal{P}$. Following [23, p. 152], we define it below.

Definition 5.1. *Let $\mathcal{A}$ be an alphabet. A morphism $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ belongs to the class $\mathcal{P}$ if it is primitive and there is an R-palindrome $p \in \mathcal{A}^*$ such that for every $a \in \mathcal{A}$, $\varphi(a) = pq_a$, where $q_a \in \mathcal{A}^*$ is an R-palindrome.*

In this article, the authors showed that if an infinite word is generated by a morphism from class $\mathcal{P}$, then it is R-palindromic. We give this result in corollary of the following lemma, given in [4, p. 4].

Lemma 5.2. *Let φ be a morphism in class $\mathcal{P}$ of the form $\varphi(a) = pq_a$, where $q_a \in \mathcal{A}^*$ is an R-palindrome, for all $a \in \mathcal{A}$ and let $\boldsymbol{u}$ be its fixed point. Then*

1. $w \in \mathcal{L}(\boldsymbol{u}) \implies \varphi(w)p \in \mathcal{L}(\boldsymbol{u})$,

2. *w is an R-palindrome $\implies \varphi(w)p$ is an R-palindrome.*

Proof. 1. Let $w \in \mathcal{L}(\boldsymbol{u})$. w is a factor of $\boldsymbol{u}$ and it is followed by some letter $a \in \mathcal{A}$, so wa is a factor of $\boldsymbol{u}$. Because $\varphi(\boldsymbol{u}) = \boldsymbol{u}$, $\varphi(wa)$ is also a factor of $\boldsymbol{u}$. We have $\varphi(wa) = \varphi(w)\varphi(a) = \varphi(w)pq_a \in \mathcal{L}(\boldsymbol{u})$ and therefore $\varphi(w)p \in \mathcal{L}(\boldsymbol{u})$.

2. Let $w = w_1 \dots w_n$ be an R-palindrome, i.e., $R(w) = w$. We want to show that $R(\varphi(w)p) = \varphi(w)p$. Using that w, p and q_a are R-palindromes and $R(\varphi(a)) = q_a p$ we get

$$\begin{aligned} R(\varphi(w)p) &= R(p)R(\varphi(w)) = pR(\varphi(w_1)\dots\varphi(w_n)) = pR(\varphi(w_n))R(\varphi(w_{n-1}))\dots R(\varphi(w_1)) \\ &= pR(\varphi(w_1))R(\varphi(w_2))\dots R(\varphi(w_n)) = pq_{w_1}pq_{w_2}p\dots q_{w_n}p = \varphi(w_1)\dots\varphi(w_n)p \\ &= \varphi(w_1\dots w_n)p = \varphi(w)p. \end{aligned}$$

□

Corollary 5.3. *A fixed point of a morphism in class $\mathcal{P}$ is R-palindromic.*

This follows from the fact that there is at least one non-empty R-palindrome in $\mathcal{L}(\boldsymbol{u})$, p or q_a for some $a \in \mathcal{A}$, and then applying Lemma 5.2 repeatedly gives infinitely many R-palindromes in $\mathcal{L}(\boldsymbol{u})$.

In [23], Hof, Knill and Simon also posed the question of whether all R-palindromic words arise from morphisms in class $\mathcal{P}$. In response, several versions of the so-called HKS conjecture were formulated with the objective to answer this, more loosely worded, question.

Following [29, p. 201-202], we give a partial overview of different versions of the HKS conjecture and of some relevant results.

First result deals with periodic words, and it was proven in [2].

Theorem 5.4. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be periodic and R-palindromic. Then $\boldsymbol{u}$ is a fixed point of a morphism in class $\mathcal{P}$.*

Another result, proven by Tan in [45], solves the question of Hof, Knill and Simon for binary words.

Theorem 5.5. *Let $\mathcal{A}$ be an alphabet with $\mathrm{card}(\mathcal{A}) = 2$ and let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism and let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be a fixed point of φ. Then $\boldsymbol{u}$ is R-palindromic if and only if there exists a morphism ψ in class $\mathcal{P}$ such that $\mathcal{L}(\varphi) = \mathcal{L}(\psi)$.*

In fact, it follows from [45] that a stronger version of this theorem holds, which has the same assumptions but states that $\boldsymbol{u}$ is R-palindromic if and only if there exists a morphism ψ in class $\mathcal{P}$ such that $\psi \sim \varphi$ or $\psi \sim \varphi^2$.

After the results above were derived, the following version of the HKS conjecture was formulated in [27, 33].

HKS Conjecture (Version 1)**.** *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be a fixed point of a primitive morphism. Then $\boldsymbol{u}$ is R-palindromic if and only if there exists a morphism $\varphi \neq \mathrm{Id}$ such that $\varphi(\boldsymbol{u}) = \boldsymbol{u}$ and φ is conjugated to some morphism in class $\mathcal{P}$.*

However, this statement was proven to be false by Labbé in [28], as a counterexample was found. Nevertheless, in [29], it was shown that for a fixed point of a primitive morphism from a specific class called marked morphisms Version 1 of the conjecture still holds.

A different version of the HKS conjecture seems plausible, which was formulated by Tan in [45].

HKS Conjecture (Version 2)**.** *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be a fixed point of a primitive morphism φ. Then $\boldsymbol{u}$ is R-palindromic if and only if there exists a morphism ψ in class $\mathcal{P}$ such that $\mathcal{L}(\varphi) = \mathcal{L}(\psi)$.*

Yet another formulation of the HKS conjecture exists, motivated by [22], where it was proven for a certain class of words. This version uses the following definition.

Definition 5.6. *Let $\mathcal{A}, \mathcal{B}$ be alphabets and let $\boldsymbol{v} \in \mathcal{B}^{\mathbb{N}}$. We say that $\boldsymbol{v}$ is primitive morphic if there exist a morphism $f\colon \mathcal{A} \to \mathcal{B}$ and $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ such that $\boldsymbol{u}$ is a fixed point of some primitive morphism $\varphi\colon \mathcal{A} \to \mathcal{A}$ and $\boldsymbol{v} = f(\boldsymbol{u})$.*

HKS Conjecture (Version 3)**.** *If $\boldsymbol{u}$ is an R-palindromic primitive morphic word, then there exist morphisms φ, ψ with conjugates in class $\mathcal{P}$ and an infinite word $\boldsymbol{v}$ such that $\boldsymbol{v} = \varphi(\boldsymbol{v})$ and $\mathcal{L}(\boldsymbol{u}) = \mathcal{L}(\psi(\boldsymbol{v}))$.*

The question of Hof, Knill and Simon is still not fully answered, as only partial results exist.

Here, we are interested in morphisms in class $\mathcal{P}$, and for our purposes, the following proposition is useful.

Proposition 5.7. *Let φ be a morphism in class $\mathcal{P}$. Then φ is conjugated to some morphism ψ in class $\mathcal{P}$ satisfying, for all $a \in \mathcal{A}$, $\psi(a) = \hat{p}\hat{q}_a$, where $\hat{q}_a \in \mathcal{A}^*$ is an R-palindrome and $\hat{p} \in \mathcal{A}^*$ with $|\hat{p}| \leq 1$.*

Proof. The morphism φ is of the form $\varphi(a) = pq_a$, for all $a \in \mathcal{A}$, where $p, q_a \in \mathcal{A}^*$ are R-palindromes. Assume that $|p| \geq 2$, otherwise the proposition is trivially valid. Then p can be expressed as $p = wbR(w)$ for some non-empty word w and $b \in \mathcal{A}^*$ with $|b| \leq 1$. Hence, we can write

$$\varphi(a) = wbR(w)q_a \quad \text{for all } a \in \mathcal{A}.$$

Then we see that φ is conjugated to the morphism ψ given by

$$\psi(a) = bR(w)q_a w \quad \text{for all } a \in \mathcal{A}.$$

We denote $\hat{p} = b$ and $\hat{q}_a = R(w)q_a w$ for all $a \in \mathcal{A}$. Then clearly $|\hat{p}| \leq 1$ and $R(\hat{q}_a) = R(w)R(q_a)w = R(w)q_a w = \hat{q}_a$ and so $\hat{q}_a$ is an R-palindrome. Therefore ψ has the required form. □

5.1.2 Exchange map E

E-palindromic words have also been studied and some results have been derived. In [27], for binary alphabet Labbé defines an analogy to the class $\mathcal{P}$, the so-called class $\mathcal{E}$-$\mathcal{P}$.

Definition 5.8. *A morphism $\varphi\colon \{0,1\}^* \to \{0,1\}^*$ belongs to the class $\mathcal{E}$-$\mathcal{P}$ if there exist E-palindromes p, q_0, q_1 such that $\varphi(a) = pq_a$ for all $a \in \{0,1\}$.*

Fixed points of morphisms in class $\mathcal{E}$-$\mathcal{P}$ are not necessary E-palindromic, this is demonstrated by an example in [5, p. 15], and we do not state here more results regarding the class $\mathcal{E}$-$\mathcal{P}$.

Instead, we focus on [5], where two classes of morphisms, class $\mathcal{A}_1$ and $\mathcal{A}_2$, were defined and studied. Here, we summarize the main results of this paper.

Definition 5.9. *A morphism $\varphi\colon \{0,1\}^* \to \{0,1\}^*$ belongs to the class $\mathcal{A}_1$ if there exist words $p, s \in \{0,1\}^*$ such that $p \neq \varepsilon$, s is an E-palindrome, and $\varphi(0) = ps$, $\varphi(1) = E(p)s$.*

Proposition 5.10. *Let φ be a primitive morphism in class $\mathcal{A}_1$ and let $\boldsymbol{u} \in \{0,1\}^{\mathbb{N}}$ be its fixed point. Then $\boldsymbol{u}$ is E-palindromic.*

The definition of the class $\mathcal{A}_2$ uses the Thue-Morse morphism φ_t, defined by $\varphi_t(0) = 01$, $\varphi_t(1) = 10$.

Definition 5.11. *A morphism $\varphi\colon \{0,1\}^* \to \{0,1\}^*$ belongs to the class $\mathcal{A}_2$ if there exist a non-empty word $w \in \{0,1\}^*$ and $k, l \in \mathbb{N}$ such that $\varphi(0) = \varphi_t\big(w(R(w)w)^k\big)$, $\varphi(1) = \varphi_t\big((R(w)w)^l R(w)\big)$.*

Proposition 5.12. *Let φ be a primitive morphism in class $\mathcal{A}_2$ and let $\boldsymbol{u} \in \{0,1\}^{\mathbb{N}}$ be its fixed point. Then $\boldsymbol{u}$ is E-palindromic.*

Lemma 5.13. *Let $\boldsymbol{u}$ be an eventually periodic E-palindromic word. If $\boldsymbol{u}$ is recurrent, then it is a fixed point of a morphism in class $\mathcal{A}_1$.*

Theorem 5.14. *Let $\varphi\colon \{0,1\}^* \to \{0,1\}^*$ be a primitive uniform morphism with an aperiodic fixed point $\boldsymbol{u}$. If $\boldsymbol{u}$ is E-palindromic, then φ or φ^2 is conjugated to a morphism in class $\mathcal{A}_1$.*

Theorem 5.15. *Let $\varphi\colon \{0,1\}^* \to \{0,1\}^*$ be a primitive non-uniform morphism with an aperiodic fixed point $\boldsymbol{u}$. If $\boldsymbol{u}$ is E-palindromic and R-palindromic, then φ or φ^2 is in class $\mathcal{A}_2$ with w from the definition of class $\mathcal{A}_2$ being an E-palindrome.*

Based on these results, the authors of [5] formulated the following conjecture.

Conjecture 5.16. *Let $\varphi\colon \{0,1\}^* \to \{0,1\}^*$ be a primitive morphism with an E-palindromic fixed point $\boldsymbol{u}$. Then φ or φ^2 is conjugated to a morphism in class $\mathcal{A}_1 \cup \mathcal{A}_2$.*

By Proposition 3.48, we know that a fixed point of a primitive morphism is uniformly recurrent, hence also recurrent. Thus, Lemma 5.13 implies that this conjecture holds for eventually periodic words. Then, by Theorems 5.14 and 5.15, the conjecture is valid for φ uniform or $\boldsymbol{u}$ R-palindromic.

5.1.3 DNA map D

Here we define a class of morphisms $\mathcal{D}$ as an analogy to the class $\mathcal{P}$ and $\mathcal{A}_1$ for the DNA map D. We show that it can be used to generate D-palindromic words.

Definition 5.17. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. A morphism $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ belongs to the class $\mathcal{D}$ if it is primitive and there is a D-palindrome $s \in \mathcal{A}^*$ and some non-empty $p, q \in \mathcal{A}^*$ such that $\varphi(\mathrm{A}) = ps$, $\varphi(\mathrm{C}) = qs$, $\varphi(\mathrm{G}) = D(q)s$, $\varphi(\mathrm{T}) = D(p)s$.*

Proposition 5.18. *Let φ be a morphism in class $\mathcal{D}$. Then φ is conjugated to some morphism ψ in class $\mathcal{D}$ satisfying $\psi(\mathrm{A}) = \hat{p}$, $\psi(\mathrm{C}) = \hat{q}$, $\psi(\mathrm{G}) = D(\hat{q})$, $\psi(\mathrm{T}) = D(\hat{p})$, where $\hat{p}, \hat{q}$ are non-empty.*

Proof. The morphism φ is of the form $\varphi(\mathrm{A}) = ps$, $\varphi(\mathrm{C}) = qs$, $\varphi(\mathrm{G}) = D(q)s$, $\varphi(\mathrm{T}) = D(p)s$, where s is a D-palindrome and p, q non-empty words. Assume that $s \neq \varepsilon$, otherwise the proposition is trivially valid. Then s can be expressed as $s = wD(w)$ for some non-empty word w. Hence, we can write

$$\begin{aligned}
\varphi(\mathrm{A}) &= pwD(w)\\
\varphi(\mathrm{C}) &= qwD(w)\\
\varphi(\mathrm{G}) &= D(q)wD(w)\\
\varphi(\mathrm{T}) &= D(p)wD(w).
\end{aligned}$$

Then we see that φ is conjugated to the morphism ψ given by

$$\begin{aligned}
\psi(\mathrm{A}) &= D(w)pw\\
\psi(\mathrm{C}) &= D(w)qw\\
\psi(\mathrm{G}) &= D(w)D(q)w\\
\psi(\mathrm{T}) &= D(w)D(p)w.
\end{aligned}$$

We denote $\hat{p} = D(w)pw = \psi(\mathrm{A})$ and $\hat{q} = D(w)qw = \psi(\mathrm{C})$. Then clearly $D(\hat{p}) = D(w)D(p)w = \psi(\mathrm{T})$ and $D(\hat{q}) = D(w)D(q)w = \psi(\mathrm{G})$. Therefore ψ has the required form. $\square$

From Proposition 3.50, we know that primitive conjugate morphisms have the same language, and therefore when working with language of morphism φ in class $\mathcal{D}$, we can assume that $s = \varepsilon$.

Proposition 5.19. *Let φ be a morphism in class $\mathcal{D}$ and let $\boldsymbol{u}$ be its infinite fixed point. Then the language of $\boldsymbol{u}$ is closed under the antimorphism D, i.e., if $w \in \mathcal{L}(\boldsymbol{u})$ then $D(w) \in \mathcal{L}(\boldsymbol{u})$.*

Proof. Proposition 5.18 and Proposition 3.50 imply that there is a morphism ψ from class $\mathcal{D}$ conjugated to φ and with a fixed point $\boldsymbol{v}$ such that $\mathcal{L}(\boldsymbol{u}) = \mathcal{L}(\boldsymbol{v})$ and $\psi(\mathrm{A}) = p$, $\psi(\mathrm{C}) = q$, $\psi(\mathrm{G}) = D(q)$, $\psi(\mathrm{T}) = D(p)$, where p, q are non-empty. It is therefore sufficient to prove that if $w \in \mathcal{L}(\boldsymbol{v})$ then $D(w) \in \mathcal{L}(\boldsymbol{v})$.

Let $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and let $w \in \mathcal{A}^*$ be in the language of $\boldsymbol{v}$. Because ψ is primitive, we know from Proposition 3.42 that its fixed point $\boldsymbol{v}$ is generated by ψ. Therefore there is $a \in \mathcal{A}$ and $k \in \mathbb{N}$ such that w is a factor of $\psi^k(a) \in \mathcal{L}(\boldsymbol{v})$. Also $D(a) \in \mathcal{L}(\boldsymbol{v})$ and hence $\psi^k(D(a)) \in \mathcal{L}(\boldsymbol{v})$.

Now we prove that $\psi \circ D = D \circ \psi$. These compositions are antimorphisms, and so it is sufficient to check the equality on images of letters:

$$\begin{aligned}
\psi(D(\mathrm{A})) &= \psi(\mathrm{T}) = D(p) = D(\psi(\mathrm{A}))\\
\psi(D(\mathrm{C})) &= \psi(\mathrm{G}) = D(q) = D(\psi(\mathrm{C}))\\
\psi(D(\mathrm{G})) &= \psi(\mathrm{C}) = q = D(D(q)) = D(\psi(\mathrm{G}))\\
\psi(D(\mathrm{T})) &= \psi(\mathrm{A}) = p = D(D(p)) = D(\psi(\mathrm{T})).
\end{aligned}$$

By applying this relation repeatedly, we get $\psi^k(D(a)) = D(\psi^k(a))$. Because w is a factor of $\psi^k(a)$, $D(w)$ is a factor of $D(\psi^k(a)) = \psi^k(D(a)) \in \mathcal{L}(\boldsymbol{v})$ and so $D(w) \in \mathcal{L}(\boldsymbol{v})$. □

Lemma 5.20. *Let φ be a morphism in class $\mathcal{D}$ and let $\boldsymbol{u}$ be its fixed point. Then*

1. *$w \in \mathcal{L}(\boldsymbol{u}) \implies s\varphi(w) \in \mathcal{L}(\boldsymbol{u})$,*
2. *w is a D-palindrome $\implies s\varphi(w)$ is a D-palindrome.*

Proof. 1. Let $w \in \mathcal{L}(\boldsymbol{u})$. Firstly, we consider the case when w is not a prefix of $\boldsymbol{u}$. Then as a factor of $\boldsymbol{u}$ it is preceded by some letter, which we denote as a. Hence $aw \in \mathcal{L}(\boldsymbol{u})$. Because $\varphi(\boldsymbol{u}) = \boldsymbol{u}$, we have $\varphi(aw) = \varphi(a)\varphi(w) \in \mathcal{L}(\boldsymbol{u})$. As s is a suffix of $\varphi(a)$, then $s\varphi(w) \in \mathcal{L}(\boldsymbol{u})$.

Secondly, we consider w to be a prefix of $\boldsymbol{u}$. We denote the first letter of w by c. As the fixed point $\boldsymbol{u}$ is generated by applying φ to c repeatedly, there is $k \in \mathbb{N}$ such that w is a prefix of $\varphi^k(c)$. It follows from the fact that φ is primitive that the letter c has another occurrence in $\boldsymbol{u}$ and therefore w has another occurrence in $\boldsymbol{u}$. Then we can use the first case to conclude that $s\varphi(w) \in \mathcal{L}(\boldsymbol{u})$.

2. Let $w = w_1 \dots w_n$ be a D-palindrome, i.e., $D(w) = w$. We want to show $D(s\varphi(w)) = s\varphi(w)$. Let us denote $\varphi(w_j) = u_j s$, where $u_j \in \{p, q\}$. Then $\varphi(D(w_j)) = D(u_j)s$. Using this and the fact that s and w are D-palindromes, we get

$$\begin{aligned}
D(s\varphi(w)) &= D(\varphi(w))D(s) = D(\varphi(w_1)\dots\varphi(w_n))s = D(\varphi(w_n))\dots D(\varphi(w_1))s\\
&= D(u_n s)\dots D(u_1 s)s = sD(u_n)\dots sD(u_1)s\\
&= s\varphi(D(w_n))\dots\varphi(D(w_1)) = s\varphi(D(w)) = s\varphi(w).
\end{aligned}$$
□

Remark 5.21. *This lemma implies that if there is a non-empty D-palindrome in the language of $\boldsymbol{u}$, where $\boldsymbol{u}$ is a fixed point of a morphism φ from the class $\mathcal{D}$, then there are infinitely many D-palindromes in $\mathcal{L}(\boldsymbol{u})$, i.e., $\boldsymbol{u}$ is D-palindromic. Clearly, the D-palindrome s from the definition of class $\mathcal{D}$ belongs to $\mathcal{L}(\boldsymbol{u})$, and so if it is non-empty, then $\boldsymbol{u}$ is D-palindromic. The case when $s = \varepsilon$ is discussed in the following section.*

5.1.4 General involutive antimorphism H

Now we generalize the class $\mathcal{D}$ from above and we define a class of morphisms $\mathcal{H}$ for a general involutive antimorphism H. We derive analogous results to some results regarding class $\mathcal{D}$ and discuss the conditions under which a fixed point of a morphism from class $\mathcal{H}$ is H-palindromic.

If an antimorphism $H\colon \mathcal{A}^* \to \mathcal{A}^*$ is an involution, it means that there are three disjoint sets of letters $\mathcal{A}_a = \{a_1, a_2, \ldots, a_k\}$, $\mathcal{A}_b = \{b_1, \ldots, b_l\}$ and $\mathcal{A}_c = \{c_1, \ldots, c_l\}$ such that

$$\mathcal{A} = \mathcal{A}_a \cup \mathcal{A}_b \cup \mathcal{A}_c$$

and

$$\begin{aligned} H(a_i) &= a_i \quad \text{for all } i \in \{1, \ldots, k\}, \\ H(b_i) &= c_i \quad \text{for all } i \in \{1, \ldots, l\}, \\ H(c_i) &= b_i \quad \text{for all } i \in \{1, \ldots, l\}, \end{aligned} \tag{5.1}$$

where $k, l \in \mathbb{N}_0$. In what follows, we keep this notation.

Definition 5.22. *Let $\mathcal{A} = \{a_1, a_2, \ldots, a_k, b_1, \ldots, b_l, c_1, \ldots, c_l\}$ be an alphabet and H an antimorphism of the form (5.1) above. Then we say that a morphism $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ belongs to the class $\mathcal{H}$ if it is primitive and there are words $p_1, p_2, \ldots, p_l \in \mathcal{A}^*$ and H-palindromes $q_1, q_2, \ldots, q_k, s \in \mathcal{A}^*$ such that*

$$\begin{aligned} \varphi(a_i) &= q_i s \quad && \textit{for all } i \in \{1, \ldots, k\}, \\ \varphi(b_i) &= p_i s && \textit{for all } i \in \{1, \ldots, l\}, \\ \varphi(c_i) &= H(p_i) s && \textit{for all } i \in \{1, \ldots, l\}. \end{aligned} \tag{5.2}$$

Example 5.23. *Consider the antimorphism R. Since $R(a) = a$ for all $a \in \mathcal{A}$, we have $\mathcal{A} = \mathcal{A}_a = \{a_1, a_2, \ldots, a_k\}$ for some $k \in \mathbb{N}$, so the sets $\mathcal{A}_b$ and $\mathcal{A}_c$ are empty. Hence, the class $\mathcal{H}$ for $H = R$ consists of primitive morphisms φ of the form*

$$\varphi(a_i) = q_i s \quad \textit{for all } i \in \{1, \ldots, k\},$$

where $q_1, \ldots, q_k, s \in \mathcal{A}^*$ are R-palindromes.

There is a clear correspondence between morphisms from this class and morphisms from the class $\mathcal{P}$, which are of the form

$$\psi(a_i) = p q_{a_i} \quad \textit{for all } i \in \{1, \ldots, k\},$$

where $q_{a_1}, \ldots, q_{a_k}, p \in \mathcal{A}^*$ are R-palindromes. We see that if we set $p = s$ and $q_{a_i} = q_i$, then $\varphi \sim \psi$. As conjugated morphisms have the same language, this class $\mathcal{H}$ and the class $\mathcal{P}$ are effectively the same.

Example 5.24. *Consider the antimorphism E, which is defined by $E(0) = 1$, $E(1) = 0$. In this case, $\mathcal{A}_a$ is empty, $\mathcal{A}_b = \{0\}$ and $\mathcal{A}_c = \{1\}$. Therefore, the class $\mathcal{H}$ for $H = E$ consists of primitive morphisms φ of the form*

$$\begin{aligned} \varphi(0) &= ps, \\ \varphi(1) &= E(p)s, \end{aligned}$$

where s is an E-palindrome and $p \in \mathcal{A}^*$. Note that the Thue-Morse morphism, which is defined by

$$\varphi_t(0) = 01, \quad \varphi_t(1) = 10,$$

belongs to this class.

Example 5.25. *Consider the antimorphism D. Here, $\mathcal{A}_a$ is empty, $\mathcal{A}_b = \{\mathrm{A}, \mathrm{C}\}$ and $\mathcal{A}_c = \{\mathrm{T}, \mathrm{G}\}$. It follows that the class $\mathcal{H}$ for $H = D$ consists of primitive morphisms φ of the form*

$$\begin{aligned}
\varphi(\mathrm{A}) &= p_1 s, \\
\varphi(\mathrm{C}) &= p_2 s, \\
\varphi(\mathrm{T}) &= D(p_1) s, \\
\varphi(\mathrm{G}) &= D(p_2) s,
\end{aligned}$$

where s is a D-palindrome and $p_1, p_2 \in \mathcal{A}^*$. If we denote $p_1 = p$ and $p_2 = q$, we see that this class $\mathcal{H}$ is exactly the class $\mathcal{D}$ defined earlier.

Lemma 5.26. *Let H be an antimorphism of the form (5.1) and let φ be a morphism from class $\mathcal{H}$. Then for all $w \in \mathcal{A}^*$ we have $s\varphi(H(w)) = H(\varphi(w))s$.*

Proof. Firstly, we show that the relation holds for all $a \in \mathcal{A}$.

If we consider the letters a_i, $i \in \{1, \dots, k\}$, we get

$$\begin{aligned}
s\varphi(H(a_i)) &= s\varphi(a_i) = s q_i s, \\
H(\varphi(a_i))s &= H(q_i s)s = s q_i s,
\end{aligned}$$

for all $i \in \{1, \dots, k\}$. Hence

$$s\varphi(H(a_i)) = H(\varphi(a_i)) \quad \text{for all } i \in \{1, \dots, k\}.$$

For letters b_i, $i \in \{1, \dots, l\}$, we have

$$\begin{aligned}
s\varphi(H(b_i)) &= s\varphi(c_i) = sH(p_i)s, \\
H(\varphi(b_i))s &= H(p_i s)s = sH(p_i)s,
\end{aligned}$$

for all $i \in \{1, \dots, l\}$. Hence

$$s\varphi(H(b_i)) = H(\varphi(b_i)) \quad \text{for all } i \in \{1, \dots, l\}.$$

If we take letters c_i, $i \in \{1, \dots, l\}$, we obtain

$$\begin{aligned}
s\varphi(H(c_i)) &= s\varphi(b_i) = s p_i s, \\
H(\varphi(c_i))s &= H(H(p_i)s)s = s p_i s,
\end{aligned}$$

for all $i \in \{1, \dots, l\}$. Hence

$$s\varphi(H(c_i)) = H(\varphi(c_i)) \quad \text{for all } i \in \{1, \dots, l\}.$$

Now let us take an arbitrary word $w \in \mathcal{A}^*$ and denote $|w| = n$, so $w = w_1 w_2 \dots w_n$, where $w_i \in \mathcal{A}$ for all $i \in \{1, \dots, n\}$. Then we have

$$\begin{aligned}
s\varphi(H(w)) &= s\varphi(H(w_1 w_2 \dots w_n)) = s\varphi(H(w_n)H(w_{n-1}) \dots H(w_1)) \\
&= s\varphi(H(w_n))\varphi(H(w_{n-1})) \dots \varphi(H(w_1)) = H(\varphi(w_n))H(\varphi(w_{n-1})) \dots H(\varphi(w_1))s \\
&= H(\varphi(w_1)\varphi(w_2) \dots \varphi(w_n))s = H(\varphi(w_1 w_2 \dots w_n))s = H(\varphi(w))s,
\end{aligned}$$

where we used that $s\varphi(H(a)) = H(\varphi(a))s$ for all $a \in \mathcal{A}$. ∎

Proposition 5.27. *Let $H \colon \mathcal{A}^* \to \mathcal{A}^*$ be an antimorphism of the form (5.1), let $\varphi \colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism and let s be an H-palindrome such that $|s| \leq |\varphi(a)|$ for all $a \in \mathcal{A}$. Then $s\varphi(H(w)) = H(\varphi(w))s$ for all $w \in \mathcal{A}^*$ if and only if φ belongs to the class $\mathcal{H}$ and is of the form (5.2) with the given s.*

Proof. We have already proved one of the two implications in Lemma 5.26. To prove the other implication, we assume that $s\varphi(H(w)) = H(\varphi(w))s$ for all $w \in \mathcal{A}^*$ and we want to show that φ is of the form (5.2). Since the relation holds for all words w, in particular, it holds for all letters $a \in \mathcal{A}$.

Firstly, we consider the letters a_i, $i \in \{1, \ldots, k\}$. We have

$$s\varphi(H(a_i)) = H(\varphi(a_i))s \quad \text{for all } i \in \{1, \ldots, k\}.$$

Hence,

$$s\varphi(a_i) = H(\varphi(a_i))s = H(\varphi(a_i))H(s) = H(s\varphi(a_i)) \quad \text{for all } i \in \{1, \ldots, k\},$$

which means that $s\varphi(a_i)$ is an H-palindrome for all $i \in \{1, \ldots, k\}$. Since $|s| \leq |\varphi(a_i)|$ for all $i \in \{1, \ldots, k\}$, this implies that for each $i \in \{1, \ldots, k\}$, there exists an H-palindrome $q_i \in \mathcal{A}^*$ such that $\varphi(a_i) = q_i s$.

Secondly, we consider the letters b_i, $i \in \{1, \ldots, l\}$. We have

$$s\varphi(H(b_i)) = H(\varphi(b_i))s \quad \text{for all } i \in \{1, \ldots, l\}.$$

It follows that

$$s\varphi(c_i) = H(\varphi(b_i))s \quad \text{for all } i \in \{1, \ldots, l\}.$$

If we apply the antimorphism H to both sides of this equality, we get

$$H(\varphi(c_i))s = s\varphi(b_i) \quad \text{for all } i \in \{1, \ldots, l\}.$$

This is equivalent to the relation for letters c_i, $i \in \{1, \ldots, l\}$. Since $|s| \leq |\varphi(b_i)|$ for all $i \in \{1, \ldots, l\}$, this implies that for each $i \in \{1, \ldots, l\}$, there exists a word $p_i \in \mathcal{A}^*$ such that $H(\varphi(c_i))s = s\varphi(b_i) = sp_i s$. Therefore, $\varphi(b_i) = p_i s$ and $H(\varphi(c_i)) = sp_i$, which is equivalent to $\varphi(c_i) = H(p_i)s$.

Overall, we have

$$\begin{aligned}
\varphi(a_i) &= q_i s && \text{for all } i \in \{1, \ldots, k\},\\
\varphi(b_i) &= p_i s && \text{for all } i \in \{1, \ldots, l\},\\
\varphi(c_i) &= H(p_i)s && \text{for all } i \in \{1, \ldots, l\},
\end{aligned}$$

where q_i, $i \in \{1, \ldots, k\}$, are H-palindromes. This is exactly the form (5.2). ☐

Corollary 5.28. *Let $H\colon \mathcal{A}^* \to \mathcal{A}^*$ be an antimorphism of the form (5.1) and let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism. Then $\varphi(H(w)) = H(\varphi(w))$ for all $w \in \mathcal{A}^*$ if and only if φ belongs to the class $\mathcal{H}$ and is of the form (5.2) with $s = \varepsilon$.*

Lemma 5.29. *Let H be an antimorphism of the form (5.1), let φ be a morphism from class $\mathcal{H}$ and let $\boldsymbol{u}$ be its fixed point. Then*

1. $w \in \mathcal{L}(\boldsymbol{u}) \implies s\varphi(w) \in \mathcal{L}(\boldsymbol{u})$,
2. *w is an H-palindrome $\implies$ $s\varphi(w)$ is an H-palindrome.*

Proof. 1. Let $w \in \mathcal{L}(\boldsymbol{u})$. Since $\boldsymbol{u}$ is a fixed point of a primitive morphism, it is uniformly recurrent (Prop. 2.44), and therefore w has infinitely many occurrences in $\boldsymbol{u}$. Hence there exists $a \in \mathcal{A}$ such that $aw \in \mathcal{L}(\boldsymbol{u})$. Because $\varphi(\boldsymbol{u}) = \boldsymbol{u}$, we have $\varphi(aw) = \varphi(a)\varphi(w) \in \mathcal{L}(\boldsymbol{u})$. As s is a suffix of $\varphi(a)$ for all $a \in \mathcal{A}$, we have $s\varphi(w) \in \mathcal{L}(\boldsymbol{u})$.

2. Let w be an H-palindrome, i.e., $H(w) = w$. We want to show $H(s\varphi(w)) = s\varphi(w)$. By using Lemma 5.26, we get

$$H(s\varphi(w)) = H(\varphi(w))s = s\varphi(H(w)) = s\varphi(w),$$

hence $s\varphi(w)$ is an H-palindrome. ☐

Lemma 5.30. *Let H be an antimorphism of the form (5.1) and let φ be a morphism from class $\mathcal{H}$ of the form (5.2). Then φ^2 also belongs to class $\mathcal{H}$ and it is of the form*

$$\begin{aligned}
\varphi^2(a_i) &= Q_i s\varphi(s) && \text{for all } i \in \{1,\dots,k\},\\
\varphi^2(b_i) &= P_i s\varphi(s) && \text{for all } i \in \{1,\dots,l\},\\
\varphi^2(c_i) &= H(P_i) s\varphi(s) && \text{for all } i \in \{1,\dots,l\},
\end{aligned}$$

where $P_1, P_2, \dots, P_l \in \mathcal{A}^*$ and $Q_1, Q_2, \dots, Q_k, s\varphi(s)$ are H-palindromes.

Proof. Let us examine images of letters under the morphism φ^2.

Firstly, we take letter a_i for some $i \in \{1,\dots,k\}$. Since $\varphi(a_i) = q_i s$, we have $\varphi^2(a_i) = \varphi(q_i)\varphi(s)$. It is clear that $\varphi(q_i)$ has s as its suffix, therefore we can write $\varphi(q_i) = Q_i s$, where $Q_i \in \mathcal{A}^*$. Hence,

$$\varphi^2(a_i) = Q_i s\varphi(s).$$

Now we need to show that Q_i and $s\varphi(s)$ are H-palindromes.

Let us consider the word sQ_is. Since q_i is an H-palindrome, we have

$$sQ_is = s\varphi(q_i) = s\varphi(H(q_i)).$$

Lemma 5.26 implies that $s\varphi(H(q_i)) = H(\varphi(q_i))s$. Hence,

$$sQ_is = H(\varphi(q_i))s = H(Q_is)s = sH(Q_i)s.$$

It follows that $Q_i = H(Q_i)$, i.e., Q_i is an H-palindrome.

To show that $s\varphi(s)$ is an H-palindrome, we consider $H(s\varphi(s))$. Since s is an H-palindrome, we have $H(s\varphi(s)) = H(\varphi(s))s$. By Lemma 5.26, $H(\varphi(s))s = s\varphi(H(s))$. Overall,

$$H(s\varphi(s)) = s\varphi(H(s)) = s\varphi(s),$$

which means that $s\varphi(s)$ is an H-palindrome.

Secondly, we take letters b_j and c_j for some $j \in \{1,\dots,l\}$. Since $\varphi(b_j) = p_j s$, we have $\varphi^2(b_j) = \varphi(p_j)\varphi(s)$. Again, we can write $\varphi(p_j) = P_j s$ for some $P_j \in \mathcal{A}^*$. Hence,

$$\varphi^2(b_j) = P_j s\varphi(s).$$

As $\varphi(c_j) = H(p_j)s$, we have $\varphi^2(c_j) = \varphi(H(p_j))\varphi(s)$. We denote $\varphi(H(p_j)) = R_j s$ for some $R_j \in \mathcal{A}^*$. It remains to be shown that $R_j = H(P_j)$, as this would give us

$$\varphi^2(c_j) = H(P_j)s\varphi(s).$$

Consider the word sR_js. Using Lemma 5.26, we get

$$sR_js = s\varphi(H(p_j)) = H(\varphi(p_j))s = H(P_js)s = sH(P_j)s.$$

This implies $R_j = H(P_j)$, which is what we wanted to show. □

Proposition 5.31. *Let H be an antimorphism of the form (5.1), let φ be a morphism from class $\mathcal{H}$ and let $\boldsymbol{u}$ be its fixed point. Then the language of $\boldsymbol{u}$ is closed under the antimorphism H, i.e., if $w \in \mathcal{L}(\boldsymbol{u})$ then $H(w) \in \mathcal{L}(\boldsymbol{u})$.*

Proof. Let $w \in \mathcal{L}(\boldsymbol{u})$ and denote $n = |w|$. Since φ is primitive, by Proposition 3.48, $\boldsymbol{u}$ is uniformly recurrent. This means that there exists $r \in \mathbb{N}$ such that any factor of $\boldsymbol{u}$ of length r contains all factors of $\boldsymbol{u}$ of length n, so in particular also w. Moreover, it follows from primitivity of φ that there exists $k \in \mathbb{N}$ such that $\left|\varphi^k(a)\right| \geq r$ for all $a \in \mathcal{A}$. Without loss of generality, we can assume that $k = 2^l$ for some $l \in \mathbb{N}$.

Let us denote $\psi = \varphi^k = \varphi^{2^l}$. Lemma 5.30 implies that ψ also belongs to class $\mathcal{H}$, so we can assume that it is of the form (5.2). Furthermore, $\boldsymbol{u}$ is a fixed point of ψ. Consider some letter $a \in \mathcal{A}$. Since φ is primitive, all letters of the alphabet are contained in $\mathcal{L}(\boldsymbol{u})$, so in particular $H(a) \in \mathcal{L}(\boldsymbol{u})$. If we apply Lemma 5.29 to the antimorphism ψ, we obtain $s\psi(H(a)) \in \mathcal{L}(\boldsymbol{u})$. We know that $|\psi(a)| \geq r$, hence w is a factor of $\psi(a)$ and clearly also of $s\psi(a)$. Then $H(w)$ is a factor of $H(s\psi(a)) = H(\psi(a))s$. By Lemma 5.26, $H(\psi(a))s = s\psi(H(a))$. Hence, $H(w)$ is a factor of $s\psi(H(a)) \in \mathcal{L}(\boldsymbol{u})$ and therefore $H(w) \in \mathcal{L}(\boldsymbol{u})$. □

Remark 5.32. *Again as in the case of the antimorphism D discussed in the previous section, Lemma 5.29 implies that if there is a non-empty H-palindrome in the language of $\boldsymbol{u}$, where $\boldsymbol{u}$ is a fixed point of a morphism φ from the class $\mathcal{H}$, then there are infinitely many H-palindromes in $\mathcal{L}(\boldsymbol{u})$, i.e., $\boldsymbol{u}$ is H-palindromic. Clearly, the H-palindrome s from the definition of class $\mathcal{H}$ belongs to $\mathcal{L}(\boldsymbol{u})$, and so if it is non-empty, then $\boldsymbol{u}$ is H-palindromic. Another case for which H-palindromicity is guaranteed is when $k \neq 0$, i.e., the set of letters fixed by H is non-empty. Then all letters a_i, $i \in \{1, \ldots, k\}$ are H-palindromes and by primitivity of φ they are in the language of $\boldsymbol{u}$.*

Hence, we have the following proposition.

Proposition 5.33. *Let $H\colon \mathcal{A}^* \to \mathcal{A}^*$ be an involutive antimorphism such that $H(a) = a$ for some $a \in \mathcal{A}$. Let φ be a morphism from the class $\mathcal{H}$ and let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be its fixed point. Then $\boldsymbol{u}$ is H-palindromic.*

Proof. Since φ is primitive, the language of $\boldsymbol{u}$ contains all letters, in particular $a \in \mathcal{A}$ such that $H(a) = a$. Hence, there is a non-empty H-palindrome in $\mathcal{L}(\boldsymbol{u})$. Lemma 5.29 tells us that for every H-palindrome w in $\mathcal{L}(\boldsymbol{u})$, $s\varphi(w)$ is an H-palindrome in $\mathcal{L}(\boldsymbol{u})$. Therefore, $\mathcal{L}(\boldsymbol{u})$ contains infinitely many H-palindromes. □

Now we want to address the question of whether $\boldsymbol{u}$ is H-palindromic when $H(a) \neq a$ for all $a \in \mathcal{A}$, i.e., $k = 0$. In other words, we want to be able to decide if there exists a non-empty H-palindrome in $\mathcal{L}(\boldsymbol{u})$. For this purpose, it is sufficient to restrict ourselves only to H-palindromes of length two, since if $k = 0$, any non-empty H-palindrome has even length and is of the form $waH(a)H(w)$, where $w \in \mathcal{A}^*$ and $a \in \mathcal{A}$. Hence it contains the factor $aH(a)$, which is a H-palindrome of length two. Therefore $\mathcal{L}(\boldsymbol{u})$ contains a non-empty H-palindrome if and only if it contains a H-palindrome of length two.

The antimorphism D from the previous section belongs to this category of antimorphism with $k = 0$, i.e., no letter is fixed by D. In the following example, we observe that there are fixed points of morphisms from the class $\mathcal{D}$ that do not contain any D-palindrome of length two.

Example 5.34. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism defined by*

$$\begin{aligned} \varphi(\mathrm{A}) &= \mathrm{ACA} \\ \varphi(\mathrm{C}) &= \mathrm{CTC} \\ \varphi(\mathrm{G}) &= \mathrm{GAG} \\ \varphi(\mathrm{T}) &= \mathrm{TGT}. \end{aligned}$$

This morphism is primitive and it belongs to the class $\mathcal{D}$ with $p = \mathrm{ACA}$, $q = \mathrm{CTC}$ and $s = \varepsilon$. It has four fixed points, but as was shown in Proposition 3.43, their languages coincide. Let us consider for example the fixed point

$$\boldsymbol{u} = \mathrm{ACACTCACACTCTGTCTCACACTCACA}\ldots$$

As $\boldsymbol{u}$ is generated by φ, a factor of $\boldsymbol{u}$ of length two is either a factor of $\varphi(a)$ for some $a \in \mathcal{A}$ or it is a concatenation of the last letter of $\varphi(b)$ and the first letter of $\varphi(c)$, where bc is a different factor of $\boldsymbol{u}$ of length two. In this case, for all $a \in \mathcal{A}$, $\varphi(a)$ does not contain any D-palindrome and it starts and ends with the letter a. Hence concatenating the last letter of $\varphi(b)$ with the first letter of $\varphi(c)$ gives again the factor bc, and therefore all the factors of $\boldsymbol{u}$ of length two are also factors of some $\varphi(a)$. This implies that there is no D-palindrome of length two in $\mathcal{L}(\boldsymbol{u})$.

We want to generalise the method used in this example to any morphism from the class $\mathcal{H}$. In order to describe how a factor of length two is created, we use the 2-factor graph defined below.

Definition 5.35. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a non-erasing morphism. Then the 2-factor graph Φ of φ is a directed graph $\Phi = (V, E)$, where the set of vertices is $V = \mathcal{A}^2$, i.e., the set of all words of length two over $\mathcal{A}$, and the set of edges is $E = \{(ab, \mathrm{lst}_\varphi(a)\,\mathrm{fst}_\varphi(b)) \mid a, b \in \mathcal{A}\}$.*

The 2-factor graph is defined for a general alphabet $\mathcal{A}$. The following example gives a 2-factor graph for the alphabet $\mathcal{A} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$.

Example 5.36. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ and let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism defined by*

$$\begin{aligned} \varphi(\mathrm{A}) &= \mathrm{AAB} \\ \varphi(\mathrm{B}) &= \mathrm{CB} \\ \varphi(\mathrm{C}) &= \mathrm{ACC}. \end{aligned}$$

Clearly it is a non-erasing morphism. In the 2-factor graph of φ, there is exactly one edge starting in each vertex. If we take for example the vertex AB, the edge starting here ends in the vertex $\mathrm{lst}_\varphi(\mathrm{A})\,\mathrm{fst}_\varphi(\mathrm{B})$, which is the vertex BC. This corresponds to the fact that the image of AB under φ contains apart from factors of length two of images of individual letter also the factor BC, as $\varphi(\mathrm{AB}) = \mathrm{AA}\underline{\mathrm{BC}}\mathrm{B}$. In the same way, we find all the edges of the 2-factor graph Φ of φ, which is shown in Figure 5.1.

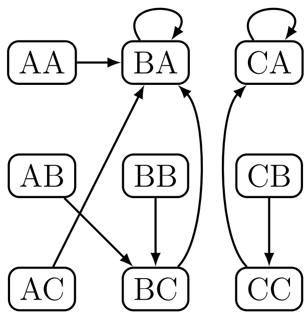


Figure 5.1: The 2-factor graph Φ of the morphism φ from Example 5.36.

Remark 5.37. *If we have a primitive morphism φ generating a fixed point $\boldsymbol{u}$, we can use the 2-factor graph Φ of φ to find the set $\mathcal{L}_2(\boldsymbol{u})$. It follows from the way how $\boldsymbol{u}$ is generated and from the definition of Φ that a word $w \in \mathcal{A}^2$ belongs to $\mathcal{L}_2(\boldsymbol{u})$ if and only if there is $u \in \mathcal{A}^2$ such that u is a factor of $\varphi(a)$ for some $a \in \mathcal{A}$ and there is a directed path in Φ starting in the vertex u and ending in the vertex w. This path can have length zero, i.e., $u = w$. If the length of such path is some $n \in \mathbb{N}_0$, it means that w is a factor of $\varphi^{n+1}(a)$.*

Therefore, for a primitive morphism φ with infinite fixed point $\boldsymbol{u}$, the method from Remark 5.37 gives the set $\mathcal{L}_2(\boldsymbol{u})$. This motivates the following definition.

Definition 5.38. *Let $\mathcal{A}$ be an arbitrary alphabet, let $H\colon \mathcal{A}^* \to \mathcal{A}^*$ be an involutive antimorphism, let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a primitive morphism and $\Phi = (V, E)$ be its 2-factor graph. Let V_A be the set of vertices v of Φ satisfying that v is a factor of $\varphi(a)$ for some $a \in \mathcal{A}$. Also let V_H be the set of vertices v of Φ satisfying that v is a H-palindrome. Then we say that the 2-factor graph Φ of φ allows H-palindromes if there is a directed path starting in a vertex from V_A and ending in a vertex from V_H.*

For an alphabet with n letters, the 2-factor graph of any morphism over this alphabet has n^2 vertices and n^2 edges. Even with a naive algorithm, it would take $\mathcal{O}(n^2)$ steps to decide whether a given 2-factor graph allows H-palindromes or not.

Remark 5.37 implies that if a primitive morphism φ with infinite fixed point $\boldsymbol{u}$ has a 2-factor graph that allows H-palindromes, then the set $\mathcal{L}_2(\boldsymbol{u})$ contains a H-palindrome. Therefore, we can formulate the following proposition.

Proposition 5.39. *Let $H\colon \mathcal{A}^* \to \mathcal{A}^*$ be an involutive antimorphism such that $H(a) \neq a$ for all $a \in \mathcal{A}$. Let φ be a morphism from the class $\mathcal{H}$ and let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be its fixed point. Then $\boldsymbol{u}$ is H-palindromic if and only if the 2-factor graph Φ of φ allows H-palindromes.*

Proof. By Remark 5.32, $\boldsymbol{u}$ is H-palindromic if and only if $\mathcal{L}(\boldsymbol{u})$ contains a non-empty H-palindrome. We have shown that this is equivalent to $\mathcal{L}_2(\boldsymbol{u})$ containing an H-palindrome and it follows from Remark 5.37 and Definition 5.38 that this is equivalent to Φ allowing H-palindromes. □

We illustrate this result on the example of antimorphism D.

Example 5.40. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism defined by*

$$\begin{aligned} \varphi(\mathrm{A}) &= \mathrm{ACC} \\ \varphi(\mathrm{C}) &= \mathrm{CTT} \\ \varphi(\mathrm{G}) &= \mathrm{AAG} \\ \varphi(\mathrm{T}) &= \mathrm{GGT}. \end{aligned}$$

This morphism is primitive and it belongs to the class $\mathcal{D}$ with $p = \mathrm{ACC}$, $q = \mathrm{CTT}$ and $s = \varepsilon$. One of its fixed points is $\boldsymbol{u} = \mathrm{ACCCTTCTTCTTGGT}\ldots$. φ is also a non-erasing morphism and we can find its 2-factor graph Φ, which is shown in Figure 5.2. The vertices from the set V_A are coloured yellow and the vertices from the set V_D are coloured red. There is a directed path from the vertex $\mathrm{AG} \in V_A$ to the vertex $\mathrm{TA} \in V_D$ and therefore Φ allows D-palindromes. This path has length two and as AG is a factor of $\varphi(\mathrm{G})$, by Remark 5.37, TA is a factor of $\varphi^3(\mathrm{G})$. We conclude that $\mathcal{L}_2(\boldsymbol{u})$ contains a D-palindrome.

Example 5.41. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism defined by*

$$\begin{aligned} \varphi(\mathrm{A}) &= \mathrm{AC} \\ \varphi(\mathrm{C}) &= \mathrm{TG} \\ \varphi(\mathrm{G}) &= \mathrm{CA} \\ \varphi(\mathrm{T}) &= \mathrm{GT}. \end{aligned}$$

This morphism is primitive and it belongs to the class $\mathcal{D}$ with $p = \mathrm{AC}$, $q = \mathrm{TG}$ and $s = \varepsilon$. It has a fixed point $\boldsymbol{u} = \mathrm{ACTGGTCACAGT}\ldots$. It is also non-erasing and we can find its 2-factor graph Φ, which is shown in Figure 5.3. We mark the vertices in the same way as in the previous example. We see that there is no directed path from a vertex in V_A to a vertex in V_D and therefore Φ does not allow D-palindromes. We conclude that $\mathcal{L}_2(\boldsymbol{u})$ does not contain a D-palindrome.

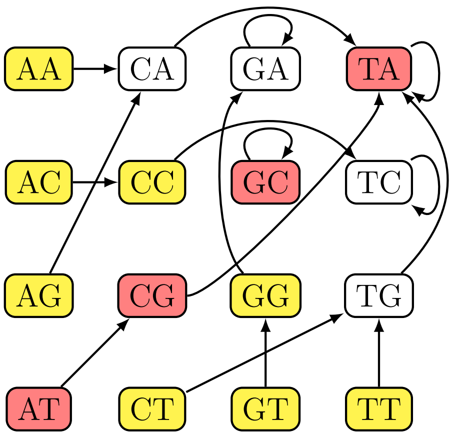


Figure 5.2: The 2-factor graph Φ of the morphism φ from Example 5.40. The set of vertices V_A is marked with yellow colour and the set V_D with red colour.

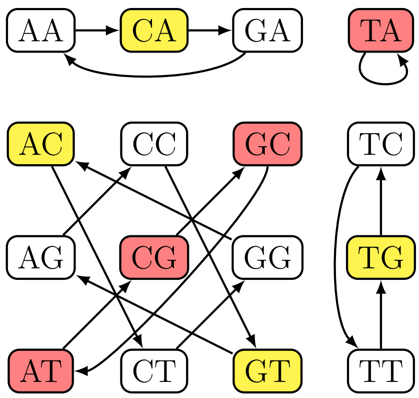


Figure 5.3: The 2-factor graph Φ of the morphism φ from Example 5.41. The set of vertices V_A is marked with yellow colour and the set V_D with red colour.

5.2 Palindromicity with respect to a group G

In section 3.4, we introduced groups of morphisms and antimorphisms. Now, we generalize the concept of palindromicity to such groups. As before, G denotes a finite group of morphisms and antimorphisms on $\mathcal{A}^*$ containing at least one antimorphism and G^{inv} is the set of all involutive antimorphisms in G.

Definition 5.42. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$. We say that $\boldsymbol{u}$ is G-palindromic if $\mathcal{L}(\boldsymbol{u})$ contains infinitely many H-palindromes for every $H \in G^{\text{inv}}$.*

Theorem 5.43. *Let φ be a primitive morphism on $\mathcal{A}^*$ and let $\boldsymbol{u}$ be its fixed point. If there exists a permutation π of G^{inv}, $\pi\colon G^{\text{inv}} \to G^{\text{inv}}$, such that $H \circ \varphi = \varphi \circ \pi(H)$ for all $H \in G^{\text{inv}}$, then the language of $\boldsymbol{u}$ is closed under each antimorphism $H \in G^{\text{inv}}$, i.e., $w \in \mathcal{L}(\boldsymbol{u}) \implies H(w) \in \mathcal{L}(\boldsymbol{u})$ for all $H \in G^{\text{inv}}$.*

Proof. Consider some antimorphism $H \in G^{\text{inv}}$ and an arbitrary word $w \in \mathcal{L}(\boldsymbol{u})$. We want to show that under the assumptions of the theorem we have $H(w) \in \mathcal{L}(\boldsymbol{u})$.

Since $\boldsymbol{u}$ is a fixed point of φ, there exist $a \in \mathcal{A}$ and $k \in \mathbb{N}$ such that w is a factor of $\varphi^k(a)$. Then $H(w)$ is a factor of $H(\varphi^k(a))$. Using the relation $H \circ \varphi = \varphi \circ \pi(H)$ for all

$H \in G^{\text{inv}}$, we get

$$H(\varphi^k(a)) = H \circ \varphi^k(a) = \varphi \circ \pi(H) \circ \varphi^{k-1}(a) = \varphi^2 \circ \pi^2(H) \circ \varphi^{k-2}(a) = \ldots = \varphi^k \circ \pi^k(H)(a).$$

Clearly, $\pi^k(H)(a) \in \mathcal{A}$, and let us denote $\pi^k(H)(a) = b$. Since φ is primitive, we have $b \in \mathcal{L}(\boldsymbol{u})$, and hence $\varphi^k(b) \in \mathcal{L}(\boldsymbol{u})$. As $H(w)$ is a factor of $\varphi^k(b)$, we obtain $H(w) \in \mathcal{L}(\boldsymbol{u})$. □

Remark 5.44. *In the statement of the theorem above, it would be sufficient to assume that there exists a mapping $\tau\colon G^{\text{inv}} \to G^{\text{inv}}$ such that for all $H \in G^{\text{inv}}$, $H \circ \varphi = \varphi \circ \tau(H)$. However, we will show that for $H, K \in G^{\text{inv}}$, $H \neq K \implies \tau(H) \neq \tau(K)$, and hence τ must be a permutation.*

Assume that there exist $H, K \in G^{\text{inv}}$, $H \neq K$ and $\tau(H) = \tau(K)$. Then we have $\varphi \circ \tau(H) = \varphi \circ \tau(K)$ and therefore $H \circ \varphi = K \circ \varphi$. Since φ is primitive, there exists a word w containing all letters $a \in \mathcal{A}$ such that $H(w) = K(w)$. But H and K restricted to letters are permutations of $\mathcal{A}$, and hence $H(w) = K(w)$ implies $H = K$. This is a contradiction.

In section 5.2.1, we study more closely some specific examples of groups G. We also employ these groups in Chapter 6. All of them satisfy $G = \langle G^{\text{inv}} \rangle$, and for this reason, we state the following corollary of Theorem 5.43.

Corollary 5.45. *Let G be a group such that $G = \langle G^{\text{inv}} \rangle$, let φ be a primitive morphism on $\mathcal{A}^*$ and let $\boldsymbol{u}$ be its fixed point. If there exists a permutation π of G^{inv}, $\pi\colon G^{\text{inv}} \to G^{\text{inv}}$, such that $H \circ \varphi = \varphi \circ \pi(H)$ for all $H \in G^{\text{inv}}$, then the language of $\boldsymbol{u}$ is closed under G.*

Proof. We want to prove that for all $\sigma \in G$, $w \in \mathcal{L}(\boldsymbol{u}) \implies \sigma(w) \in \mathcal{L}(\boldsymbol{u})$. Since $G = \langle G^{\text{inv}} \rangle$, any $\sigma \in G$ can be expressed as a composition of antimorphisms from G^{inv}. Therefore the result immediately follows from Theorem 5.43, as the language of $\boldsymbol{u}$ is closed under each antimorphism $H \in G^{\text{inv}}$. □

Theorem 5.46. *Let φ be a primitive morphism on $\mathcal{A}^*$ and π be a permutation of G^{inv} such that for all $H \in G^{\text{inv}}$, $H \circ \varphi = \varphi \circ \pi(H)$. Let $\boldsymbol{u}$ be a fixed point of φ. If $\mathcal{L}(\boldsymbol{u})$ contains a non-empty H-palindrome for some $H \in G^{\text{inv}}$, then $\boldsymbol{u}$ is H-palindromic.*

Proof. Assume that $\mathcal{L}(\boldsymbol{u})$ contains a non-empty H-palindrome w for some $H \in G^{\text{inv}}$, i.e., $H(w) = w$. We want to show that $\mathcal{L}(\boldsymbol{u})$ contains infinitely many H-palindromes. Since π is a permutation of G^{inv}, there is some $k \in \mathbb{N}$ such that $\pi^k(H) = H$. Now consider the word $\varphi^k(w)$. Clearly, $\varphi^k(w) \in \mathcal{L}(\boldsymbol{u})$ and $\left|\varphi^k(w)\right| > |w|$. Moreover,

$$\begin{aligned} H \circ \varphi^k(w) &= \varphi \circ \pi(H) \circ \varphi^{k-1}(w) = \varphi^2 \circ \pi^2(H) \circ \varphi^{k-2}(w) = \ldots \\ &= \varphi^k \circ \pi^k(H)(w) = \varphi^k \circ H(w) = \varphi^k(w). \end{aligned}$$

Therefore $\varphi^k(w)$ is also a H-palindrome. By repeating this process, we can find infinitely many H-palindromes in $\mathcal{L}(\boldsymbol{u})$. □

Corollary 5.47. *Let φ be a primitive morphism on $\mathcal{A}^*$ and π be a permutation of G^{inv} such that for all $H \in G^{\text{inv}}$, $H \circ \varphi = \varphi \circ \pi(H)$. Let $\boldsymbol{u}$ be a fixed point of φ. If $\mathcal{L}(\boldsymbol{u})$ contains a non-empty H-palindrome for every $H \in G^{\text{inv}}$, then $\boldsymbol{u}$ is G-palindromic.*

This gives us a method for generating G-palindromic words. However, we need to pose some additional requirements on the morphism φ to ensure that its fixed point is G-palindromic. We have already discussed this for H-palindromic words in section 5.1.4 and the conditions for the case of G-palindromic words follow from this discussion. We state it in the following proposition.

Proposition 5.48. *Let φ be a primitive morphism on $\mathcal{A}^*$ and π be a permutation of G^{inv} such that for all $H \in G^{\text{inv}}$, $H \circ \varphi = \varphi \circ \pi(H)$. Let $\boldsymbol{u}$ be a fixed point of φ. Then $\boldsymbol{u}$ is G-palindromic if and only if the 2-factor graph Φ of φ allows H-palindromes for every $H \in G^{\text{inv}}$ such that $H(a) \neq a$ for all $a \in \mathcal{A}$.*

Proof. By Corollary 5.47, we know that $\boldsymbol{u}$ is G-palindromic if and only if $\mathcal{L}(\boldsymbol{u})$ contains a non-empty H-palindrome for every $H \in G^{\text{inv}}$. Let us take some $H \in G^{\text{inv}}$. If there exists $a \in \mathcal{A}$ such that $H(a) = a$, then $a \in \mathcal{L}(\boldsymbol{u})$ is a non-empty H-palindrome. If $H(a) \neq a$ for all $a \in \mathcal{L}(\boldsymbol{u})$, then $\mathcal{L}(\boldsymbol{u})$ contains a non-empty H-palindrome if and only if $\mathcal{L}_2(\boldsymbol{u})$ contains an H-palindrome and this is equivalent to Φ allowing H-palindromes. □

Proposition 5.49. *Let φ be a primitive morphism on $\mathcal{A}^*$ and π be a permutation of G^{inv} such that for all $H \in G^{\text{inv}}$, $H \circ \varphi = \varphi \circ \pi(H)$. Then there exists some $k \in \mathbb{N}$ such that the morphism φ^k satisfies that $H \circ \varphi^k = \varphi^k \circ H$ for all $H \in G^{\text{inv}}$.*

Proof. Since π is a permutation of G^{inv}, there exists some $k \in \mathbb{N}$ such that π^k is the identity permutation. Then for every $H \in G^{\text{inv}}$ we have

$$H \circ \varphi^k = \varphi \circ \pi(H) \circ \varphi^{k-1} = \ldots = \varphi^k \circ \pi^k(H) = \varphi^k \circ H.$$ □

Corollary 5.50. *Let φ be a primitive morphism on $\mathcal{A}^*$ and π be a permutation of G^{inv} such that for all $H \in G^{\text{inv}}$, $H \circ \varphi = \varphi \circ \pi(H)$. Then there exists a primitive morphism ψ satisfying $H \circ \psi = \psi \circ H$ for all $H \in G^{\text{inv}}$ such that $\mathcal{L}(\varphi) = \mathcal{L}(\psi)$.*

Proof. This is a direct consequence of Proposition 5.49 and Definition 3.47. □

As we are interested in morphisms generating G-palindromic words, this corollary tells us that we do not need to consider morphisms satisfying the condition $H \circ \varphi = \varphi \circ \pi(H)$ for all $H \in G^{\text{inv}}$ for a general permutation π of G^{inv}, but that we can restrict ourselves only to the identity permutation, and hence to morphisms satisfying $H \circ \varphi = \varphi \circ H$ for all $H \in G^{\text{inv}}$. This leads us to the following definition of a class of morphisms $\mathcal{G}$.

Definition 5.51. *Let G be a group as above. We say that a morphism $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ belongs to the class $\mathcal{G}$ if it is primitive and it satisfies $H \circ \varphi = \varphi \circ H$ for all $H \in G^{\text{inv}}$.*

We can reformulate Proposition 5.48 for morphisms from class $\mathcal{G}$.

Proposition 5.52. *Let φ be a morphism from the class $\mathcal{G}$ and let $\boldsymbol{u}$ be its fixed point. Then $\boldsymbol{u}$ is G-palindromic if and only if the 2-factor graph Φ of φ allows H-palindromes for every $H \in G^{\text{inv}}$ such that $H(a) \neq a$ for all $a \in \mathcal{A}$.*

There is a connection between the class $\mathcal{G}$ and the individual classes $\mathcal{H}$ for $H \in G^{\text{inv}}$, stated in the following proposition.

Proposition 5.53. *Let $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ be a morphism. Then φ belongs to the class $\mathcal{G}$ if and only if $\varphi \in \mathcal{H}_\varepsilon$ for each $H \in G^{\text{inv}}$, where $\mathcal{H}_\varepsilon$ denotes the set of morphisms belonging to the class $\mathcal{H}$ and being of the form (5.2) with $s = \varepsilon$.*

Proof. This is a direct consequence of Corollary 5.28 and Definition 5.51. □

Another, more general, way of generating G-palindromic words would be to consider morphisms that belong to the class $\mathcal{H}$ for each $H \in G^{\text{inv}}$. By Proposition 5.27, this is equivalent to considering morphisms φ satisfying for each $H \in G^{\text{inv}}$ the relation $s_H\varphi(H(w)) = H(\varphi(w))s_H$ for all $w \in \mathcal{A}^*$ and some H-palindrome s_H such that $|s_H| \leq |\varphi(a)|$ for all $a \in \mathcal{A}$.

Even more generally, we could consider fixed points of morphisms that are conjugated to a morphism from class $\mathcal{H}$ for each $H \in G^{\text{inv}}$.

However, both these approaches are much more complicated and it is not clear whether they would bring something new compared to the method with class $\mathcal{G}$. In the following section, we consider some specific examples of groups G and we show that under some condition, in the case of the group $G = \langle\{R, D\}\rangle$, these more general approaches do not produce different G-palindromic words than morphisms from class $\mathcal{G}$ (Theorem 5.55).

5.2.1 Groups generated by two antimorphisms

In what follows, we study some concrete examples of groups generated by two involutive antimorphisms. Given a group G, we want to determine whether there exist some morphisms belonging to the class $\mathcal{G}$ and of what form these morphisms are. By Proposition 5.52, such morphisms can generate G-palindromic words.

Firstly, we consider the group $G = \langle\{R, D\}\rangle$. Except for morphisms from class $\mathcal{G}$, in this case we also examine morphisms satisfying the condition $H \circ \varphi = \varphi \circ \pi(H)$ for all $H \in G^{\text{inv}}$ for a general permutation π of G^{inv} to demonstrate the results from above.

1. $G = \langle\{R, D\}\rangle$

Let us take $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and consider the group $G = \langle\{R, D\}\rangle$, i.e., $G = \{\mathrm{Id}, R, D, RD\}$. In this case, $G^{\text{inv}} = \{R, D\}$. There are two permutations of the set G^{inv}, the identity permutation and the permutation π given by $\pi(R) = D$, $\pi(D) = R$.

Firstly, let us consider a morphism φ belonging to the class $\mathcal{G}$, i.e., satisfying

$$\begin{aligned} R \circ \varphi &= \varphi \circ R, \\ D \circ \varphi &= \varphi \circ D. \end{aligned}$$

We can restrict these equalities to letters only, as two antimorphisms are equal if and only if they coincide on letters. The first equality is equivalent to

$$R(\varphi(a)) = \varphi(a), \quad \text{for all } a \in \mathcal{A},$$

i.e., $\varphi(a)$ is an R-palindrome for all $a \in \mathcal{A}$. The second equality is equivalent to

$$\begin{aligned} D(\varphi(\mathrm{A})) &= \varphi(\mathrm{T}), \\ D(\varphi(\mathrm{C})) &= \varphi(\mathrm{G}). \end{aligned}$$

We omit the other two equalities, as they are equivalent to the two equalities above. Overall, φ belongs to the class $\mathcal{G}$ if and only if it is primitive and is of the form

$$\varphi(\mathrm{A}) = p, \quad \varphi(\mathrm{C}) = q, \quad \varphi(\mathrm{G}) = D(q), \quad \varphi(\mathrm{T}) = D(p),$$

where p, q are R-palindromes. This is sufficient, since necessarily $D(p)$ and $D(q)$ are R-palindromes. Note that a morphism of this form indeed satisfies that it belongs to the class $\mathcal{P}$ with p from the definition of $\mathcal{P}$ being empty and it also belongs to the class $\mathcal{D}$ with s from the definition of $\mathcal{D}$ being empty, as stated in Proposition 5.53. An example of such morphism φ is

$$\begin{aligned} \varphi(\mathrm{A}) &= \mathrm{ACGTGCA}, \\ \varphi(\mathrm{C}) &= \mathrm{GAG}, \\ \varphi(\mathrm{G}) &= \mathrm{CTC}, \\ \varphi(\mathrm{T}) &= \mathrm{TGCACGT}. \end{aligned}$$

Let us denote its fixed point starting with A as $\boldsymbol{u}$. Clearly, $\mathcal{L}(\boldsymbol{u})$ contains a non-empty R-palindrome (e.g. GAG) and a non-empty D-palindrome (e.g. ACGT), hence by Corollary 5.47 $\boldsymbol{u}$ is G-palindromic.

Secondly, let us take the permutation π and consider a morphism φ satisfying

$$\begin{aligned} R \circ \varphi &= \varphi \circ D, \\ D \circ \varphi &= \varphi \circ R. \end{aligned} \tag{5.3}$$

The first equality is equivalent to

$$\begin{aligned} R(\varphi(\mathrm{A})) &= \varphi(\mathrm{T}), \\ R(\varphi(\mathrm{C})) &= \varphi(\mathrm{G}). \end{aligned}$$

The second equality is equivalent to

$$D(\varphi(a)) = \varphi(a), \quad \text{for all } a \in \mathcal{A},$$

i.e., $\varphi(a)$ is a D-palindrome for all $a \in \mathcal{A}$. Overall, φ satisfies equalities (5.3) if and only if it is of the form

$$\varphi(\mathrm{A}) = p, \quad \varphi(\mathrm{C}) = q, \quad \varphi(\mathrm{G}) = R(q), \quad \varphi(\mathrm{T}) = R(p), \tag{5.4}$$

where p, q are D-palindromes. An example of such morphism φ that is also primitive is

$$\begin{aligned} \varphi(\mathrm{A}) &= \mathrm{AGCT}, \\ \varphi(\mathrm{C}) &= \mathrm{TA}, \\ \varphi(\mathrm{G}) &= \mathrm{AT}, \\ \varphi(\mathrm{T}) &= \mathrm{TCGA}. \end{aligned} \tag{5.5}$$

Let us denote its fixed point starting with A as $\boldsymbol{u}$. Clearly, $\mathcal{L}(\boldsymbol{u})$ contains a non-empty R-palindrome (e.g. any letter) and a non-empty D-palindrome (e.g. AGCT), hence by Corollary 5.47 $\boldsymbol{u}$ is G-palindromic.

Since π satisfies that π^2 is the identity permutation, Proposition 5.49 implies that for every primitive morphism φ of the form (5.4), φ^2 belongs to the class $\mathcal{G}$. For the morphism φ given by (5.5), we have

$$\begin{aligned} \varphi^2(\mathrm{A}) &= \mathrm{AGCTATTATCGA}, \\ \varphi^2(\mathrm{C}) &= \mathrm{TCGAAGCT}, \\ \varphi^2(\mathrm{G}) &= \mathrm{AGCTTCGA}, \\ \varphi^2(\mathrm{T}) &= \mathrm{TCGATAATAGCT}, \end{aligned}$$

which indeed belongs to the class $\mathcal{G}$.

We can summarize the main result of this discussion by the following proposition.

Proposition 5.54. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and let $G = \langle\{R, D\}\rangle$. Then a primitive morphism $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ belongs to the class $\mathcal{G}$ if and only if it is of the form*

$$\varphi(\mathrm{A}) = p, \quad \varphi(\mathrm{C}) = q, \quad \varphi(\mathrm{G}) = D(q), \quad \varphi(\mathrm{T}) = D(p),$$

where p, q are R-palindromes.

As mentioned above, there are more general approaches for generating G-palindromic words. The following theorem suggests that in the case of $G = \langle\{R, D\}\rangle$, it is sufficient to consider only morphisms from class $\mathcal{G}$. The proof of this theorem is quite technical and it requires one result about word equations with palindromes derived in Chapter 4.

Theorem 5.55. *Let* $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ *and let* $\varphi \colon \mathcal{A}^* \to \mathcal{A}^*$ *be a morphism conjugated to a morphism from class* $\mathcal{D}$ *and also conjugated to a morphism from class* $\mathcal{P}$ *such that* $\varphi(\mathrm{A}) \neq \varphi(\mathrm{T})$ *or* $\varphi(\mathrm{C}) \neq \varphi(\mathrm{G})$*. Then* φ *is conjugated to a morphism* ψ *which is of the form*

$$\psi(\mathrm{A}) = p, \quad \psi(\mathrm{C}) = q, \quad \psi(\mathrm{G}) = D(q), \quad \psi(\mathrm{T}) = D(p),$$

where p, q *are non-empty* R*-palindromes, i.e.,* ψ *belongs to the class* $\mathcal{G}$ *for* $G = \langle\{R, D\}\rangle$*.*

Proof. Let $\tilde{\varphi}$ be a morphism from class $\mathcal{D}$ that is conjugated to φ. By Proposition 5.18, $\tilde{\varphi}$ is conjugated to some morphism θ satisfying

$$\theta(\mathrm{A}) = \hat{p}, \quad \theta(\mathrm{C}) = \hat{q}, \quad \theta(\mathrm{G}) = D(\hat{q}), \quad \theta(\mathrm{T}) = D(\hat{p}),$$

where $\hat{p}, \hat{q}$ are non-empty.

By Proposition 3.26, $\varphi \sim \tilde{\varphi}$ and $\tilde{\varphi} \sim \theta$ implies that $\varphi \sim \theta$.

Since φ is conjugated to a morphism in class $\mathcal{P}$, then also θ is conjugated to this morphism in class $\mathcal{P}$. Proposition 5.7 implies that such a morphism from class $\mathcal{P}$ is conjugated to a morphism σ of the form

$$\sigma(\mathrm{A}) = p_{\mathrm{A}}\gamma, \quad \sigma(\mathrm{C}) = p_{\mathrm{C}}\gamma, \quad \sigma(\mathrm{G}) = p_{\mathrm{G}}\gamma, \quad \sigma(\mathrm{T}) = p_{\mathrm{T}}\gamma,$$

where $p_{\mathrm{A}}, p_{\mathrm{C}}, p_{\mathrm{G}}, p_{\mathrm{T}}$ are R-palindromes and $|\gamma| \leq 1$.

Therefore, $\theta \sim \sigma$. This means that there exists $w \in \mathcal{A}^*$ such that $\theta(a)w = w\sigma(a)$ for all $a \in \mathcal{A}$ or $w\theta(a) = \sigma(a)w$ for all $a \in \mathcal{A}$. It is sufficient to consider only the first option, since in the other case when $w\theta(a) = \sigma(a)w$ for all $a \in \mathcal{A}$ we can take the morphism $\tilde{\sigma}$ given by $\tilde{\sigma}(a) = \gamma p_a$ for all $a \in \mathcal{A}$, which satisfies $\gamma\sigma(a) = \tilde{\sigma}(a)\gamma$ for all $a \in \mathcal{A}$. Then we have $\hat{w}\theta(a) = \tilde{\sigma}(a)\hat{w}$ for all $a \in \mathcal{A}$, where $\hat{w} = \gamma w$, and these equalities are symmetric to the equalities $\theta(a)w = w\sigma(a)$ for all $a \in \mathcal{A}$ and can be solved analogously.

Hence, using the form of θ and σ, we get the equalities

$$\begin{aligned} \hat{p}w &= wp_{\mathrm{A}}\gamma \\ \hat{q}w &= wp_{\mathrm{C}}\gamma \\ D(\hat{q})w &= wp_{\mathrm{G}}\gamma \\ D(\hat{p})w &= wp_{\mathrm{T}}\gamma. \end{aligned} \tag{5.6}$$

Now we distinguish two possibilities, either $|w| \leq |\gamma|$ or $|w| > |\gamma|$.

A. $|w| \leq |\gamma|$

Since $|\gamma| \leq 1$, we have either $w = \gamma$ or $w = \varepsilon$.

If $w = \gamma$, then equalities (5.6) imply

$$\begin{aligned} \hat{p} &= \gamma p_{\mathrm{A}} \\ \hat{q} &= \gamma p_{\mathrm{C}} \\ D(\hat{q}) &= \gamma p_{\mathrm{G}} \\ D(\hat{p}) &= \gamma p_{\mathrm{T}}. \end{aligned} \tag{5.7}$$

If $w = \varepsilon$, then equalities (5.6) become

$$\begin{aligned} \hat{p} &= p_{\mathrm{A}}\gamma \\ \hat{q} &= p_{\mathrm{C}}\gamma \\ D(\hat{q}) &= p_{\mathrm{G}}\gamma \\ D(\hat{p}) &= p_{\mathrm{T}}\gamma. \end{aligned} \tag{5.8}$$

Let us consider only equalities (5.8), as equalities (5.7) can be solved analogously. Our goal is to show that $\gamma = \varepsilon$, since then $\hat{p}$ and $\hat{q}$ are R-palindromes and if we take ψ to be the morphism θ, then it has the required form. Hence, to reach a contradiction, we assume that $\gamma \neq \varepsilon$, i.e., $|\gamma| = 1$.

From equalities (5.8), we obtain the following relations

$$D(\hat{p}) = p_{\mathrm{T}}\gamma = D(\gamma)D(p_{\mathrm{A}}) \tag{5.9}$$

$$D(\hat{q}) = p_{\mathrm{G}}\gamma = D(\gamma)D(p_{\mathrm{C}}). \tag{5.10}$$

Here, all the words γ, $D(\gamma)$, p_{T}, $D(p_{\mathrm{A}})$, p_{G}, $D(p_{\mathrm{C}})$ are non-empty R-palindromes, as the image of an R-palindrome under the antimorphism D is again an R-palindrome. Hence, we can use Proposition 4.3. Applying this proposition to equality (5.9) implies that there exist R-palindromes $\alpha, \beta \in \mathcal{A}^*$ and $i, j, k, l \in \mathbb{N}_0$ such that

$$p_{\mathrm{T}} = (\alpha\beta)^i\alpha, \quad \gamma = \beta(\alpha\beta)^j, \quad D(\gamma) = (\alpha\beta)^k\alpha, \quad D(p_{\mathrm{A}}) = \beta(\alpha\beta)^l. \tag{5.11}$$

Since $|\gamma| = 1$ and $\gamma \neq D(\gamma)$, it follows from equalities (5.11) that $\gamma = \beta$ and $D(\gamma) = \alpha$. Hence we get

$$p_{\mathrm{T}} = (D(\gamma)\gamma)^i D(\gamma)$$
$$D(p_{\mathrm{A}}) = \gamma(D(\gamma)\gamma)^l.$$

Since $|p_{\mathrm{T}}| = |p_{\mathrm{A}}|$, we have $i = l$ and

$$p_{\mathrm{T}} = (D(\gamma)\gamma)^i D(\gamma)$$
$$p_{\mathrm{A}} = (D(\gamma)\gamma)^i D(\gamma),$$

hence we obtained $p_{\mathrm{T}} = p_{\mathrm{A}}$. This implies that $\hat{p} = D(\hat{p})$, i.e., $\theta(\mathrm{A}) = \theta(\mathrm{T})$.

Analogously, applying Proposition 4.3 to equality (5.10) would result in the observation $p_{\mathrm{G}} = p_{\mathrm{C}}$. Hence, $\hat{q} = D(\hat{q})$, i.e., $\theta(\mathrm{C}) = \theta(\mathrm{G})$.

These results are in contradiction with the assumption that $\varphi(\mathrm{A}) \neq \varphi(\mathrm{T})$ or $\varphi(\mathrm{C}) \neq \varphi(\mathrm{G})$, as this implies that θ, which is conjugated to φ, also satisfies $\theta(\mathrm{A}) \neq \theta(\mathrm{T})$ or $\theta(\mathrm{C}) \neq \theta(\mathrm{G})$.

So we conclude that $\gamma = \varepsilon$, which is what we wanted to show.

B. $|w| > |\gamma|$

In this case, equalities (5.6) imply that $w = x\gamma$ for some $x \neq \varepsilon$, and we can write

$$\hat{p}x = x\gamma p_{\mathrm{A}} \tag{5.12}$$

$$\hat{q}x = x\gamma p_{\mathrm{C}} \tag{5.13}$$

$$D(\hat{q})x = x\gamma p_{\mathrm{G}} \tag{5.14}$$

$$D(\hat{p})x = x\gamma p_{\mathrm{T}}. \tag{5.15}$$

Let us take equalities (5.12) and (5.15). We can apply Proposition 4.2 to both these equalities. Then equality (5.12) implies that there exist $a, b \in \mathcal{A}^*$, $a \neq \varepsilon$, and $m \in \mathbb{N}_0$ such that

$$\hat{p} = ab, \quad \gamma p_{\mathrm{A}} = ba, \quad x = (ab)^m a. \tag{5.16}$$

Similarly, equality (5.15) implies that there exist $c, d \in \mathcal{A}^*$, $c \neq \varepsilon$, and $n \in \mathbb{N}_0$ such that

$$D(\hat{p}) = cd, \quad \gamma p_{\mathrm{T}} = dc, \quad x = (cd)^n c. \tag{5.17}$$

Since $|\hat{p}| = |D(\hat{p})|$, we have $|ab| = |cd|$. In addition, $|a| \leq |ab|$ and $|c| \leq |cd|$. Therefore, $x = (ab)^m a$ implies

$$m\,|ab| < |x| \leq (m+1)\,|ab|$$

and $x = (cd)^n c$ implies

$$n\,|ab| < |x| \leq (n+1)\,|ab|\,.$$

Combining this together, we get $m < n+1$ and $n < m+1$. Therefore, $m = n$. Hence, we have

$$x = (ab)^m a = (cd)^m c. \tag{5.18}$$

If $m \neq 0$, then clearly $a = c$ and $b = d$. This implies $\hat{p} = ab = cd = D(\hat{p})$, i.e., $\theta(\mathrm{A}) = \theta(\mathrm{T})$.

Now let us consider the case when $m = 0$. Then equality (5.18) becomes $x = a = c$ and from equalities (5.16) and (5.17) we have

$$\hat{p} = ab, \quad D(\hat{p}) = ad, \quad \gamma p_{\mathrm{A}} = ba, \quad \gamma p_{\mathrm{T}} = da. \tag{5.19}$$

Since $|\hat{p}| = |D(\hat{p})|$, we have $|b| = |d|$.

If $b = d = \varepsilon$, then $\hat{p} = a = D(\hat{p})$, i.e., $\theta(\mathrm{A}) = \theta(\mathrm{T})$.

So let us take $|b| > 0$. If $|a| \geq |b|$, then necessarily $\hat{p} = D(\hat{p})$, i.e., $\theta(\mathrm{A}) = \theta(\mathrm{T})$. Hence, consider $|a| < |b|$. Since $|\gamma| \leq 1$, we have $|b| > |\gamma|$. Then it follows from equalities (5.19) that there exist $y, z \in \mathcal{A}^*$, $|y| = |z| > 0$, such that

$$b = \gamma y, \quad d = \gamma z.$$

Using these relations, we obtain from equalities (5.19) the following:

$$\hat{p} = a\gamma y, \quad D(\hat{p}) = a\gamma z, \quad p_{\mathrm{A}} = ya, \quad p_{\mathrm{T}} = za. \tag{5.20}$$

Since $|a| < |b|$, we have $|a| \leq |y| = |z|$. In the case when $|a| = |y|$, we get $y = R(a) = z$, hence $\hat{p} = D(\hat{p})$, i.e., $\theta(\mathrm{A}) = \theta(\mathrm{T})$. Consider the other case when $|a| < |y|$. Since $p_{\mathrm{A}}, p_{\mathrm{T}}$ are R-palindromes, we have

$$ya = R(a)R(y), \quad za = R(a)R(z).$$

Then $|a| < |y|$ implies that there exist R-palindromes u, v such that $|u| = |v| > 0$ and

$$y = R(a)u, \quad z = R(a)v. \tag{5.21}$$

Hence, equalities (5.20) become

$$\hat{p} = a\gamma R(a)u, \quad D(\hat{p}) = a\gamma R(a)v, \quad p_{\mathrm{A}} = R(a)ua, \quad p_{\mathrm{T}} = R(a)va. \tag{5.22}$$

From here, we get

$$D(\hat{p}) = a\gamma R(a)v = D(u)D(a\gamma R(a)), \tag{5.23}$$

where all the words $a\gamma R(a)$, v, $D(u)$, $D(a\gamma R(a))$ are non-empty R-palindromes. Therefore, we can apply Proposition 4.3 to equality (5.23). Hence, there exist R-palindromes δ, η and $e, f, g, h \in \mathbb{N}_0$ such that

$$a\gamma R(a) = (\delta\eta)^e\delta, \quad v = \eta(\delta\eta)^f, \quad D(u) = (\delta\eta)^g\delta, \quad D(a\gamma R(a)) = \eta(\delta\eta)^h. \tag{5.24}$$

If one of the two R-palindromes δ, η is empty, we get $a\gamma R(a) = D(a\gamma R(a))$, which means that $a\gamma R(a)$ is a D-palindrome. But this is not possible, since $a\gamma R(a)$ is a non-empty

R-palindrome. Hence, we have $0 < |\delta| \le |\delta\eta|$ and $0 < \eta \le |\delta\eta|$. Since $|a\gamma R(a)| = |D(a\gamma R(a))|$, equalities (5.24) imply

$$\begin{aligned} e\,|\delta\eta| < |a\gamma R(a)| \le (e+1)\,|\delta\eta|\,, \\ h\,|\delta\eta| < |a\gamma R(a)| \le (h+1)\,|\delta\eta|\,. \end{aligned}$$

Combining this together, we get $e < h+1$ and $h < e+1$. Therefore, $e = h$.

Similarly, since $|v| = |D(u)|$, we get $f = g$.

Now, we have

$$D(a\gamma R(a)) = \eta(\delta\eta)^e = D((\delta\eta)^e\delta) = D(\delta)(D(\eta)D(\delta))^e.$$

This implies $\eta = D(\delta)$.

Overall, we obtain from equalities (5.24) that

$$v = D(\delta)(\delta D(\delta))^f, \quad D(u) = (\delta D(\delta))^f\delta.$$

It follows that

$$u = D(\delta)(\delta D(\delta))^f,$$

hence, $u = v$. Substituting this into equalities (5.22), we get $\hat{p} = a\gamma R(a)u = D(\hat{p})$, i.e., $\theta(\mathrm{A}) = \theta(\mathrm{T})$.

To summarize, we showed that equalities (5.12) and (5.15) imply $\theta(\mathrm{A}) = \theta(\mathrm{T})$. If we consider equalities (5.13) and (5.14) and proceed analogously, we obtain $\theta(\mathrm{C}) = \theta(\mathrm{G})$. These results are in contradiction with the assumption that $\varphi(\mathrm{A}) \neq \varphi(\mathrm{T})$ or $\varphi(\mathrm{C}) \neq \varphi(\mathrm{G})$, therefore, the case **B.** is not possible. □

2. $G = \langle\{\Psi_0, \Psi_1\}\rangle$

Let us take $\mathcal{A} = \mathbb{Z}_3$ and consider the group $G = \{\mathrm{Id}, \mu, \mu^{-1}, \Psi_0, \Psi_1, \Psi_2\}$ from Example 3.68. Here, $G^{\mathrm{inv}} = \{\Psi_0, \Psi_1, \Psi_2\}$. It can be observed from the Cayley table given in Example 3.68 that $G = \langle\{\Psi_0, \Psi_1\}\rangle$, so this group is also generated by two involutive antimorphisms. Recall that the involutive antimorphisms Ψ_0, Ψ_1 and Ψ_2 are of the form

$$\begin{array}{lll} \Psi_0(0) = 0 & \Psi_1(0) = 1 & \Psi_2(0) = 2 \\ \Psi_0(1) = 2 & \Psi_1(1) = 0 & \Psi_2(1) = 1 \\ \Psi_0(2) = 1 & \Psi_1(2) = 2 & \Psi_2(2) = 0. \end{array}$$

Now, let us consider a primitive morphism φ belonging to the class $\mathcal{G}$, i.e., satisfying

$$\Psi_0 \circ \varphi = \varphi \circ \Psi_0, \tag{5.25}$$
$$\Psi_1 \circ \varphi = \varphi \circ \Psi_1, \tag{5.26}$$
$$\Psi_2 \circ \varphi = \varphi \circ \Psi_2. \tag{5.27}$$

As $G = \langle\{\Psi_0, \Psi_1\}\rangle$, Ψ_2 can be expressed as some composition of Ψ_0 and Ψ_1, in particular, we have $\Psi_2 = \Psi_0 \circ \Psi_1 \circ \Psi_0$. Therefore, the equalities (5.25) and (5.26) imply equality (5.27), since

$$\Psi_2 \circ \varphi = \Psi_0 \circ \Psi_1 \circ \Psi_0 \circ \varphi = \Psi_0 \circ \Psi_1 \circ \varphi \circ \Psi_0 = \ldots = \varphi \circ \Psi_0 \circ \Psi_1 \circ \Psi_0 = \varphi \circ \Psi_2.$$

So it is sufficient to consider only the first two equalities.

Equality (5.25) is equivalent to

$$\Psi_0(\varphi(0)) = \varphi(\Psi_0(0)) = \varphi(0), \tag{5.28}$$
$$\Psi_0(\varphi(1)) = \varphi(\Psi_0(1)) = \varphi(2), \tag{5.29}$$
$$\Psi_0(\varphi(2)) = \varphi(\Psi_0(2)) = \varphi(1). \tag{5.30}$$

Equality (5.28) is equivalent to $\varphi(0)$ being a Ψ_0-palindrome, and since Ψ_0 is an involution, both (5.29) and (5.30) are equivalent to $\varphi(2) = \Psi_0(\varphi(1))$.

Equality (5.26) is equivalent to

$$\Psi_1(\varphi(0)) = \varphi(\Psi_1(0)) = \varphi(1), \tag{5.31}$$
$$\Psi_1(\varphi(1)) = \varphi(\Psi_1(1)) = \varphi(0), \tag{5.32}$$
$$\Psi_1(\varphi(2)) = \varphi(\Psi_1(2)) = \varphi(2). \tag{5.33}$$

Equality (5.33) is equivalent to $\varphi(2)$ being a Ψ_1-palindrome, and since Ψ_1 is an involution, both (5.31) and (5.32) are equivalent to $\varphi(1) = \Psi_1(\varphi(0))$.

Combining this together, a primitive morphism φ belongs to the class $\mathcal{G}$ if and only if

$$\varphi(0) \text{ is a } \Psi_0\text{-palindrome,} \tag{5.34}$$
$$\varphi(1) = \Psi_1(\varphi(0)), \tag{5.35}$$
$$\varphi(2) = \Psi_0(\varphi(1)) \text{ and} \tag{5.36}$$
$$\varphi(2) \text{ is a } \Psi_1\text{-palindrome.} \tag{5.37}$$

Using (5.34) and (5.35) as well as the relation $\Psi_2 = \Psi_0 \circ \Psi_1 \circ \Psi_0$, we can rewrite (5.36) as

$$\varphi(2) = \Psi_0(\varphi(1)) = \Psi_0 \circ \Psi_1(\varphi(0)) = \Psi_0 \circ \Psi_1 \circ \Psi_0(\varphi(0)) = \Psi_2(\varphi(0)).$$

Now we can deduce that (5.34), (5.35) and (5.36) imply (5.37), since

$$\Psi_1(\varphi(2)) = \Psi_1 \circ \Psi_2(\varphi(0)) = \Psi_1 \circ \Psi_2 \circ \Psi_0(\varphi(0)) = \Psi_2(\varphi(0)) = \varphi(2),$$

where we used $\Psi_1 \circ \Psi_2 \circ \Psi_0 = \Psi_2$, which can be seen form the Cayley table given in Example 3.68.

Therefore, we conclude that a primitive morphism φ belongs to the class $\mathcal{G}$ if and only if it is of the form

$$\varphi(0) = p, \quad \varphi(1) = \Psi_1(p), \quad \varphi(2) = \Psi_2(p),$$

where p is a Ψ_0-palindrome. An example of such morphism is φ given by

$$\varphi(0) = 02010,$$
$$\varphi(1) = 10121,$$
$$\varphi(2) = 21202.$$

Any fixed point of this morphism is G-palindromic, since each of the antimorphisms Ψ_0, Ψ_1 and Ψ_2 satisfies that there exists a letter $a \in \mathcal{A}$ being fixed by it.

Based on this analysis, we can state the following proposition.

Proposition 5.56. *Let $\mathcal{A} = \mathbb{Z}_3$ and let $G = \{\mathrm{Id}, \mu, \mu^{-1}, \Psi_0, \Psi_1, \Psi_2\}$. Then a primitive morphism $\varphi\colon \mathcal{A}^* \to \mathcal{A}^*$ belongs to the class $\mathcal{G}$ if and only if it is of the form*

$$\varphi(0) = p, \quad \varphi(1) = \Psi_1(p), \quad \varphi(2) = \Psi_2(p),$$

where p is a Ψ_0-palindrome.

3. $G = \langle\{\Psi_0, R\}\rangle$

Let us take $\mathcal{A} = \mathbb{Z}_3$ and consider the group $G = \langle\{\Psi_0, R\}\rangle$, i.e., $G = \{\mathrm{Id}, \Psi_0, R, \Psi_0 \circ R\}$. Here, $G^{\mathrm{inv}} = \{\Psi_0, R\}$.

We consider a morphism φ satisfying the relations

$$\Psi_0 \circ \varphi = \varphi \circ \Psi_0, \tag{5.38}$$

$$R \circ \varphi = \varphi \circ R. \tag{5.39}$$

Equality (5.38) is the same as equality (5.25) from above. We have seen that it is equivalent to $\varphi(0)$ being a Ψ_0-palindrome and $\varphi(2) = \Psi_0(\varphi(1))$.

On the other hand, equality (5.39) is equivalent to

$$R(\varphi(a)) = \varphi(a), \quad \text{for all } a \in \mathcal{A},$$

i.e., $\varphi(a)$ is an R-palindrome for all $a \in \mathcal{A}$.

Therefore, $\varphi(0)$ has to be a Ψ_0-palindrome and an R-palindrome at the same time. This can be satisfied if and only if $\varphi(0)$ contains only the letter 0. In the case of $\varphi(1)$ and $\varphi(2)$, it is sufficient to take $\varphi(1) = p$, where p is an R-palindrome, and $\varphi(2) = \Psi_0(p)$. Then $\varphi(2)$ is also an R-palindrome, since

$$R(\varphi(2)) = R \circ \Psi_0(p) = \Psi_0 \circ R(p) = \Psi_0(p) = \varphi(2),$$

and hence both equalities (5.38) and (5.39) are satisfied.

Overall, we obtain that φ has to be of the form

$$\varphi(0) = 0^i, \quad \varphi(1) = p, \quad \varphi(2) = \Psi_0(p),$$

where $i \in \mathbb{N}$ and p is an R-palindrome. However, in this case φ cannot be primitive. Hence no primitive morphism satisfies the relations (5.38) and (5.39), so we conclude that the class $\mathcal{G}$ is empty.

Chapter 6

Palindromic richness

This chapter is devoted to the concept of palindromic richness. Firstly, we summarize some results about classical palindromic richness and secondly, we discuss its generalizations. We focus on the generalization with respect to a group of morphisms and antimorphisms G and describe an algorithm for deciding whether a given finite word is G-rich or not. We also present results of our tests, which lead us to formulate some conjectures about classes of morphisms generating G-rich words.

6.1 Classical palindromic richness

In this section, we describe the concept of palindromic richness in the classical sense, where only classical palindromes, i.e., R-palindromes, are considered. Here, we also use the word palindrome for an R-palindrome.

Definition 6.1. *Let $w \in \mathcal{A}^*$. By $\mathbb{P}(w)$, we denote the set of all palindromic factors of w. This includes the empty word ε.*

Example 6.2. *Let $\mathcal{A} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ and $w = \mathrm{ABBCBBC}$. Then*

$$\mathbb{P}(w) = \{\varepsilon, \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{BB}, \mathrm{BCB}, \mathrm{CBBC}, \mathrm{BBCBB}\}.$$

Note that $|w| = 7$ and $\operatorname{card}(\mathbb{P}(w)) = 8$.

In [18], the following proposition was given.

Proposition 6.3. *Let $w \in \mathcal{A}^*$. Then w has at most $|w| + 1$ distinct palindromic factors, i.e., $\operatorname{card}(\mathbb{P}(w)) \leq |w| + 1$.*

This upper bound on the number of palindromic factors in a word motivates the definition of a rich word, stated in [20].

Definition 6.4. *A word $w \in \mathcal{A}^*$ is called rich if $\operatorname{card}(\mathbb{P}(w)) = |w| + 1$. An infinite word $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ is called rich if all its factors are rich.*

We see that in Example 6.2, $\operatorname{card}(\mathbb{P}(w)) = |w| + 1$, so the upper bound on the number of palindromic factors is attained and w is rich.

Palindromic richness can be reformulated in terms of the notion of defect of a word, introduced in [14].

Definition 6.5. *The defect of a finite word $w \in \mathcal{A}^*$, denoted by $d(w)$, is defined as*

$$d(w) = |w| + 1 - \operatorname{card}(\mathbb{P}(w)).$$

The defect of an infinite word $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$, denoted by $d(\boldsymbol{u})$, is defined as

$$d(\boldsymbol{u}) = \sup\{d(w) \mid w \in \mathcal{L}(\boldsymbol{u})\}.$$

Clearly, a finite or an infinite word is rich if and only if its defect is zero. If an infinite word has finite defect, it is called almost rich [20].

There is another equivalent characterisation of rich words, which uses the definition below [18, 20].

Definition 6.6. *Let $w \in \mathcal{A}^*$. A factor v of w is called unioccurrent in w if v has exactly one occurrence in w.*

Proposition 6.7. *Let $w \in \mathcal{A}^*$. Then w is rich if and only if all its prefixes have a unioccurrent palindromic suffix (UPS).*

Example 6.8. *Let us take again the word $w = \mathrm{ABBCBBC}$ from Example 6.2. We will show that w satisfies the condition from Proposition 6.7. The prefixes of w are ε, A, AB, ABB, ABBC, ABBCB, ABBCBB, ABBCBBC and their UPS are ε, A, B, BB, C, BCB, BBCBB, CBBC, respectively.*

In fact, if a word w has a UPS, then this UPS is unique and it is the longest palindromic suffix (LPS) of w [20]. This follows from the observation that if w has two different palindromic suffixes, then the shorter one cannot be unioccurrent, since it is both proper suffix and proper prefix of the longer palindrome.

Following [36], we summarize results characterizing rich infinite words with language closed under reversal, which can be found in [20, 18, 15, 8]. As defined before, $c_{\boldsymbol{u}}$ is the factor complexity of $\boldsymbol{u}$, $p_{\boldsymbol{u}}$ is the palindromic complexity of $\boldsymbol{u}$, $\overline{\Gamma}_n(\boldsymbol{u})$ is the graph of symmetries, in this case for $G = \{\mathrm{Id}, R\}$, $b_{\boldsymbol{u}}(w)$ is the bilateral order of $w \in \mathcal{L}(\boldsymbol{u})$ and $\mathrm{Pext}_{\boldsymbol{u}}(w)$ is the set of palindromic extensions of a palindrome $w \in \mathcal{L}(\boldsymbol{u})$.

Theorem 6.9. *Let $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ be an infinite word with language closed under the mirror image map R. Then the following statements are equivalent:*

1. *$\boldsymbol{u}$ is rich,*
2. *all complete return words to any palindromic factor of $\boldsymbol{u}$ are palindromes [20],*
3. *for any $w \in \mathcal{L}(\boldsymbol{u})$, every $v \in \mathcal{L}(\boldsymbol{u})$ that contains w only as its prefix and $R(w)$ only as its suffix is a palindrome [20],*
4. *for any $w \in \mathcal{L}(\boldsymbol{u})$, the longest palindromic suffix of w is unioccurrent in w [18, 20],*
5. *for all $n \in \mathbb{N}$, $c_{\boldsymbol{u}}(n+1) - c_{\boldsymbol{u}}(n) + 2 = p_{\boldsymbol{u}}(n) + p_{\boldsymbol{u}}(n+1)$ [15],*
6. *for all $n \in \mathbb{N}$, the graph of symmetries $\overline{\Gamma}_n(\boldsymbol{u})$ satisfies that all its loops are palindromes and by removing loops from $\overline{\Gamma}_n(\boldsymbol{u})$ a tree is obtained [15],*
7. *for any bispecial factor w of $\boldsymbol{u}$, $b_{\boldsymbol{u}}(w) = \mathrm{card}(\mathrm{Pext}_{\boldsymbol{u}}(w)) - 1$ if w is a palindrome and $b_{\boldsymbol{u}}(w) = 0$ otherwise [8].*

We will illustrate these properties of rich words on the Fibonacci word $\boldsymbol{f}$. The Fibonacci word is a fixed point of the Fibonacci morphism $\varphi_f \colon \{0,1\}^* \to \{0,1\}^*$ defined by

$$\varphi_f(0) = 01, \quad \varphi_f(1) = 0,$$

which belongs to the class $\mathcal{P}$, hence, by Proposition 5.31, the language of $\boldsymbol{f}$ is closed under R. Moreover, $\boldsymbol{f}$ is known to be rich [18].

Example 6.10. *Consider the Fibonacci word*

$$\boldsymbol{f} = 01001010010010100101001001010010\ldots$$

We know that this word is rich, hence also all the other statements in Theorem 6.9 must hold. We illustrate each property on an example.

2. *Let us take the palindromic factor* 101 *of* $\boldsymbol{f}$. *Below, we underline the first four occurrences of* 101 *in* $\boldsymbol{f}$:

$$\boldsymbol{f} = 0100\underline{101}00100\underline{101}00\underline{101}00100\underline{101}0010\ldots$$

From these occurrences, we can identify its complete return words 10100100101 *and* 10100101 *and we see that both are indeed palindromes.*

3. *Consider* $w = 1010 \in \mathcal{L}(\boldsymbol{f})$. *Then* $R(w) = 0101$ *and for example* $v = 10100100101 \in \mathcal{L}(\boldsymbol{f})$ *contains* w *only as its prefix and* $R(w)$ *only as its suffix. As expected,* v *is a palindrome.*

4. *If we take for example* $w = 01001010$, *then the longest palindromic suffix is* 01010, *which is truly unioccurrent in* w.

5. *Here, we choose* $n = 3$ *and we want to verify the relation* $c_{\boldsymbol{f}}(4) - c_{\boldsymbol{f}}(3) + 2 = p_{\boldsymbol{f}}(3) + p_{\boldsymbol{f}}(4)$. *It can be determined that* $\mathcal{L}_3(\boldsymbol{f}) = \{010, 100, 001, 101\}$ *and* $\mathcal{L}_4(\boldsymbol{f}) = \{0100, 1001, 0010, 1010, 0101\}$. *Then* $c_{\boldsymbol{f}}(3) = \mathrm{card}(\mathcal{L}_3(\boldsymbol{f})) = 4$, $c_{\boldsymbol{f}}(4) = \mathrm{card}(\mathcal{L}_4(\boldsymbol{f})) = 5$, *and as the number of palindromes in* $\mathcal{L}_3(\boldsymbol{f})$ *and* $\mathcal{L}_4(\boldsymbol{f})$ *is* 2 *and* 1, *respectively, we have* $p_{\boldsymbol{f}}(3) = 2$ *and* $p_{\boldsymbol{f}}(4) = 1$. *Overall,*

$$c_{\boldsymbol{f}}(4) - c_{\boldsymbol{f}}(3) + 2 = 5 - 4 + 2 = 3 = 2 + 1 = p_{\boldsymbol{f}}(3) + p_{\boldsymbol{f}}(4).$$

6. *For* $n = 4$, *the graph of symmetries* $\overline{\Gamma}_4(\boldsymbol{f})$ *was constructed in Example 3.74 and it is shown again in Figure 6.1. Clearly, all its loops are palindromes and if they are removed, we get a tree.*

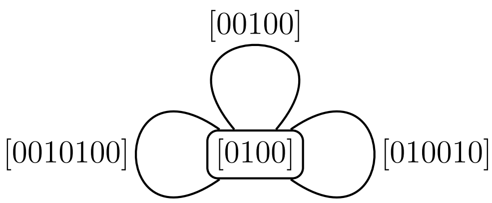


Figure 6.1: The graph of symmetries $\overline{\Gamma}_4(\boldsymbol{f})$ for the Fibonacci word.

7. *An example of a bispecial factor of* $\boldsymbol{f}$ *is* $w = 010$. *It is a palindrome, so it should hold that* $b_{\boldsymbol{f}}(w) = \mathrm{card}(\mathrm{Pext}_{\boldsymbol{f}}(w)) - 1$. *By definition,*

$$\begin{aligned} b_{\boldsymbol{f}}(w) &= \mathrm{card}(\mathrm{Bext}_{\boldsymbol{f}}(w)) - \mathrm{card}(\mathrm{Lext}_{\boldsymbol{f}}(w)) - \mathrm{card}(\mathrm{Rext}_{\boldsymbol{f}}(w)) + 1 \\ &= \mathrm{card}(\{00101, 00100, 10100\}) - \mathrm{card}(\{0,1\}) - \mathrm{card}(\{0,1\}) + 1 \\ &= 3 - 2 - 2 + 1 = 0. \end{aligned}$$

$\mathrm{Pext}_{\boldsymbol{f}}(w)$ *is the set of all palindromic extensions of* w, *and in this case* 0 *is the only palindromic extension of* w, *as* 00100 *is the only palindrome in* $\mathrm{Bext}_{\boldsymbol{f}}(w)$. *Hence* $\mathrm{card}(\mathrm{Pext}_{\boldsymbol{f}}(w)) - 1 = 0$ *and the relation is satisfied.*

Example 6.11. *Let us take the Thue-Morse word*

$$\boldsymbol{t} = 0110100110010110100101100110100 1\dots$$

Unlike the Fibonacci word, the Thue-Morse word is not rich. We can see this by considering the palindromic factor 11 *of* $\boldsymbol{t}$*. Below is underlined one of its complete return words:*

$$\boldsymbol{t} = 0110100\underline{110010110}100101100110100 1\dots$$

This word is not a palindrome, and so property 2. from Theorem 6.9 is not satisfied.

6.2 Generalized palindromic richness

There are some generalizations of palindromic richness, which we will discuss in this section. Firstly, instead of the antimorphism R, we can take any involutive antimorphism H and define H-richness analogously. Second method, which is a lot more interesting, is to consider a group of morphisms and antimorphisms G.

6.2.1 Richness with respect to an antimorphism

Here, by H we mean a general involutive antimorphism $H\colon \mathcal{A}^* \to \mathcal{A}^*$, and we discuss the concept of H-richness.

Definition 6.12. *Let* $w \in \mathcal{A}^*$*. By* $\mathbb{P}_H(w)$*, we denote the set of all* H*-palindromic factors of* w*. This includes the empty word* ε*.*

In [42], the following proposition was given.

Proposition 6.13. *Let* $w \in \mathcal{A}^*$*. Then*

$$\operatorname{card}(\mathbb{P}_H(w)) \leq |w| + 1 - \gamma_H(w),$$

where $\gamma_H(w) = \operatorname{card}\big(\{\{a, H(a)\} \mid a \in \mathcal{A},\ a \text{ occurs in } w,\ a \neq H(a)\}\big)$*.*

Definition 6.14. *A word* $w \in \mathcal{A}^*$ *is called* H*-rich if* $\operatorname{card}(\mathbb{P}_H(w)) = |w| + 1 - \gamma_H(w)$*. An infinite word* $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ *is called* H*-rich if all its factors are* H*-rich.*

Note that for the antimorphism R, we have $R(a) = a$ for all $a \in \mathcal{A}$, and therefore $\gamma_R(w) = 0$ for all $w \in \mathcal{A}^*$. So Proposition 6.13 gives the same upper bound on the number of R-palindromic factors in a word as Proposition 6.3.

Example 6.15. *Consider the antimorphism* $E\colon \{0,1\}^* \to \{0,1\}^*$ *given by* $E(0) = 1$ *and* $E(1) = 0$*, and let* $w = 10101$*. Then*

$$\operatorname{card}(\mathbb{P}_E(w)) = \operatorname{card}(\{\varepsilon, 10, 01, 1010, 0101\}) = 5.$$

Also we have

$$\gamma_E(w) = \operatorname{card}\big(\{\{0,1\}\}\big) = 1,$$

therefore

$$|w| + 1 - \gamma_E(w) = 5 + 1 - 1 = 5$$

and we conclude that w *is* E*-rich.*

Example 6.16. *Consider the antimorphism D and let $w = \mathrm{TTAA}$. Then*

$$\gamma_D(w) = \operatorname{card}\big(\{\{\mathrm{A}, \mathrm{T}\}\}\big) = 1,$$

so

$$|w| + 1 - \gamma_D(w) = 4 + 1 - 1 = 4.$$

However,

$$\operatorname{card}(\mathbb{P}_D(w)) = \operatorname{card}(\{\varepsilon, \mathrm{TA}, \mathrm{TTAA}\}) = 3,$$

so w is not D-rich.

On the other hand, if we take $v = \mathrm{ATCGAT}$, we have

$$\gamma_D(v) = \operatorname{card}\big(\{\{\mathrm{A}, \mathrm{T}\}, \{\mathrm{C}, \mathrm{G}\}\}\big) = 2,$$

and

$$|v| + 1 - \gamma_D(v) = 6 + 1 - 2 = 5.$$

Since

$$\operatorname{card}(\mathbb{P}_D(v)) = \operatorname{card}(\{\varepsilon, \mathrm{AT}, \mathrm{CG}, \mathrm{TCGA}, \mathrm{ATCGAT}\}) = 5,$$

we conclude that v is D-rich.

Definition 6.17. *The H-defect of a finite word $w \in \mathcal{A}^*$, denoted by $d_H(w)$, is defined as*

$$d_H(w) = |w| + 1 - \gamma_H(w) - \operatorname{card}(\mathbb{P}_H(w)).$$

The H-defect of an infinite word $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$, denoted by $d_H(\boldsymbol{u})$, is defined as

$$d_H(\boldsymbol{u}) = \sup\{d_H(w) \mid w \in \mathcal{L}(\boldsymbol{u})\}.$$

Again, it immediately follows that a finite or an infinite word is H-rich if and only if its H-defect is zero. If an infinite word has finite H-defect, it is called almost H-rich.

Example 6.18. *Let us consider the Thue-Morse word $\boldsymbol{t} = 0110100110010110 10\ldots$, which is a fixed point of the Thue-Morse morphism given by*

$$\varphi_t(0) = 01 \quad \varphi_t(1) = 10.$$

Firstly, note that φ_t belongs to the class $\mathcal{H}$ for $H = E$ and hence $\mathcal{L}(\boldsymbol{t})$ is closed under E and E-palindromic. Moreover, φ_t^2 belongs to the class $\mathcal{P}$, since

$$\varphi_t^2(0) = 0110 \quad \varphi_t^2(1) = 1001,$$

and so $\mathcal{L}(\boldsymbol{t})$ is also closed under R and R-palindromic.

We have already seen in Example 6.11 that the Thue-Morse word is not rich. In fact, it is not E-rich either. We can see this by considering the word $w = 0110 \in \mathcal{L}(\boldsymbol{t})$. We have

$$\begin{aligned} d_E(w) &= |w| + 1 - \gamma_E(w) - \operatorname{card}(\mathbb{P}_E(w)) = 4 + 1 - \operatorname{card}\big(\{\{0, 1\}\}\big) - \operatorname{card}(\{\varepsilon, 01, 10\}) \\ &= 4 + 1 - 1 - 3 = 1 \neq 0, \end{aligned}$$

so w is not E-rich.

We have shown that the Thue-Morse word is neither R-rich nor E-rich. In fact, a stronger result, stated in [35, 13], holds.

Theorem 6.19. *The Thue-Morse word is not almost H-rich for any involutive antimorphism H.*

Furthermore, the concept of E-richness does not prove to be interesting, since it was shown in [21] that the only finite words that are E-rich are $(01)^k$, $(01)^k\,0$, $(10)^k$ and $(10)^k\,1$ for some $k \in \mathbb{N}_0$ and hence there are only two infinite E-rich words, $010101\ldots$ and $101010\ldots$

However, there is another generalization of palindromic richness that takes into account more types of palindromes at the same time and we describe this in the next section. We will see that in the case of the Thue-Morse word, if we consider both R-palindromes and E-palindromes simultaneously, then in some sense the Thue-Morse word is rich.

6.2.2 Richness with respect to a group

We have already discussed the concept of G-palindromicity in section 5.2. Now, we focus on G-richness, following [35, 36]. Again, G denotes here a finite group of morphisms and antimorphisms on $\mathcal{A}^*$ containing at least one antimorphism and G^{inv} is the set of all involutive antimorphisms in G.

There are more possible ways how to approach generalization of palindromic richness to a group G, since there are several equivalent characterisation of palindromic richness, see Theorem 6.9. In [35], G-richness of an infinite word $\boldsymbol{u}$ was defined using the graph of symmetries of $\boldsymbol{u}$. But later, it was shown by the same authors in [36] that there are again equivalent characterizations of G-rich words analogous to the characterizations given in Theorem 6.9 using analogous concepts, including the so-called G-defect of a word. Here, we choose to define G-rich word by their property of having zero G-defect.

Definition 6.20. *Let $w \in \mathcal{A}^*$. By $\mathbb{P}_G(w)$, we denote the set of all G-palindromic classes of equivalence in w, i.e.,*

$$\mathbb{P}_G(w) = \{[v]_G \mid v \textit{ is a factor of } w \textit{ and a } G\textit{-palindrome}\}.$$

Example 6.21. *Let $G = \langle\{R, E\}\rangle$ and consider $w = 011010$. For this group, a word is a G-palindrome if it is an R-palindrome or an E-palindrome. The set of R-palindromic factors of w is*

$$\mathbb{P}_R(w) = \{\varepsilon, 0, 1, 11, 101, 0110\}$$

and the set of E-palindromic factors of w is

$$\mathbb{P}_E(w) = \{\varepsilon, 01, 10, 1010\}.$$

Note that $[0]_G = [1]_G$ and $[01]_G = [10]_G$. Then we have

$$\mathbb{P}_G(w) = \{[\varepsilon]_G, [0]_G, [01]_G, [11]_G, [101]_G, [0110]_G, [1010]_G\},$$

i.e.,

$$\mathbb{P}_G(w) = \big\{\{\varepsilon\}, \{0, 1\}, \{01, 10\}, \{11, 00\}, \{101, 010\}, \{0110, 1001\}, \{1010, 0101\}\big\}.$$

In [36], it was shown that there is again an upper bound on the number of elements in the set $\mathbb{P}_G(w)$ as there was for the set $\mathbb{P}_H(w)$, and hence an analogy to the notion of the defect of a word can be defined in this case as well.

Proposition 6.22. *Let $w \in \mathcal{A}^*$. Then*

$$\operatorname{card}(\mathbb{P}_G(w)) \le |w| + 1 - \gamma_G(w),$$

where

$$\gamma_G(w) = \operatorname{card}\big(\{[a]_G \mid a \in \mathcal{A},\ a \textit{ occurs in } w,\ a \neq \theta(a) \textit{ for every antimorphism } \theta \in G\}\big).$$

Definition 6.23. *The G-defect of a finite word $w \in \mathcal{A}^*$, denoted by $d_G(w)$, is defined as*

$$d_G(w) = |w| + 1 - \gamma_G(w) - \operatorname{card}(\mathbb{P}_G(w)).$$

The G-defect of an infinite word $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$, denoted by $d_G(\boldsymbol{u})$, is defined as

$$d_G(\boldsymbol{u}) = \sup\{d_G(w) \mid w \in \mathcal{L}(\boldsymbol{u})\}.$$

Definition 6.24. *A word $w \in \mathcal{A}^*$ is called G-rich if $d_G(w) = 0$. An infinite word $\boldsymbol{u} \in \mathcal{A}^{\mathbb{N}}$ is called G-rich if $d_G(\boldsymbol{u}) = 0$ and it is called almost G-rich if $d_G(\boldsymbol{u})$ is finite.*

Remark 6.25. *Note that if we take the group $G = \{\mathrm{Id}, H\}$ for some involutive antimorphism H, then the concept of G-richness coincides with the concept of H-richness. We can see this by considering the G-defect of a finite word w. We have*

$$d_G(w) = |w| + 1 - \gamma_G(w) - \operatorname{card}(\mathbb{P}_G(w)),$$

where

$$\gamma_G(w) = \operatorname{card}\big(\{[a]_G \mid a \in \mathcal{A},\ a \text{ occurs in } w,\ a \neq \theta(a) \text{ for every antimorphism } \theta \in G\}\big).$$

Since $G = \{\mathrm{Id}, H\}$ and H is the only antimorphism in G, we have

$$\gamma_G(w) = \operatorname{card}\big(\{\{a, H(a)\} \mid a \in \mathcal{A},\ a \text{ occurs in } w,\ a \neq H(a)\}\big) = \gamma_H(w).$$

Also a word v is a G-palindrome if and only if it is an H-palindrome, hence $\mathbb{P}_G(w) = \mathbb{P}_H(w)$.

Overall

$$d_G(w) = |w| + 1 - \gamma_G(w) - \operatorname{card}(\mathbb{P}_G(w)) = |w| + 1 - \gamma_H(w) - \operatorname{card}(\mathbb{P}_H(w)) = d_H(w)$$

and so a finite or an infinite word is G-rich if and only if it is H-rich.

Example 6.26. *Let $G = \langle\{R, E\}\rangle$ and consider $w = 011010$. The set $\mathbb{P}_G(w)$ was given in Example 6.21 and we observe that $\operatorname{card}(\mathbb{P}_G(w)) = 7$. Since $a = R(a)$ for all letters a, we have $\gamma_G(w) = 0$. Hence*

$$d_G(w) = |w| + 1 - \gamma_G(w) - \operatorname{card}(\mathbb{P}_G(w)) = 6 + 1 - 0 - 7 = 0,$$

and we can conclude that w is G-rich.

The word w from Example 6.26 is a prefix of the Thue-Morse word $\boldsymbol{t}$. As discussed above, despite its language being closed under R and R-palindromic, the Thue-Morse word is not rich, and in spite of its language being closed under E and E-palindromic, it is not E-rich either. However, if we consider the group generated by both antimorphisms that the Thue-Morse word is closed under, i.e., $G = \langle\{R, E\}\rangle$, we find that the Thue-Morse word is in fact G-rich, as was shown in [35]. We state it in the following theorem.

Theorem 6.27. *The Thue-Morse word is G-rich for $G = \langle\{R, E\}\rangle$.*

Except for the Thue-Morse word, there are other known examples of G-rich words. It was shown in [43] that the so-called generalized Thue-Morse words are G-rich, and for each such word the corresponding group G was explicitly given. We describe these words below.

Definition 6.28. *Let $b, m \in \mathbb{N}$, $b \geq 2$ and $m \geq 1$. Let $s_b(n)$ denote the sum of digits of an integer n in the base b representation and let $\mathcal{A} = \mathbb{Z}_m$. Then the infinite word $\boldsymbol{t}_{b,m} \in \mathcal{A}^{\mathbb{N}}$ defined by*

$$\boldsymbol{t}_{b,m} = (s_b(n) \mod m)_{n \in \mathbb{N}_0}$$

is called a generalized Thue-Morse word.

The generalized Thue-Morse words include the classical Thue-Morse word, which is equal to the word $\boldsymbol{t}_{2,2}$. As stated in [3], $\boldsymbol{t}_{b,m}$ is a fixed point of a primitive morphism given by the following lemma.

Lemma 6.29. *Let $b, m \in \mathbb{N}$, $b \geq 2$ and $m \geq 1$. Let $\mathcal{A} = \mathbb{Z}_m$. Then the word $\boldsymbol{t}_{b,m}$ is a fixed point of the morphism $\varphi_{b,m} \colon \mathcal{A}^* \to \mathcal{A}^*$ defined by*

$$\varphi_{b,m}(k) = k\,(k+1)\,(k+2)\,\ldots\,(k+b-1) \quad \text{for } k \in \mathcal{A},$$

where the letters are expressed as sums modulo m.

Finally, we state the theorem about G-richness of $\boldsymbol{t}_{b,m}$ from [43].

Theorem 6.30. *Let $b, m \in \mathbb{N}$, $b \geq 2$ and $m \geq 1$. Let $\mathcal{A} = \mathbb{Z}_m$ and let G be the group*

$$G = \{\Psi_x \mid x \in \mathbb{Z}_m\} \cup \{\Pi_x \mid x \in \mathbb{Z}_m\}$$

from Example 3.68, where Ψ_x, $x \in \mathbb{Z}_m$, are antimorphisms defined by $\Psi_x(k) = x - k \mod m$ for all $k \in \mathbb{Z}_m$, and Π_x, $x \in \mathbb{Z}_m$, are morphisms defined by $\Pi_x(k) = x + k \mod m$ for all $k \in \mathbb{Z}_m$. Then the word $\boldsymbol{t}_{b,m}$ is G-rich.

Example 6.31. *Consider the case $m = 3$. Then the group G from Theorem 6.30 is equal to $G = \{\mathrm{Id}, \mu, \mu^{-1}, \Psi_0, \Psi_1, \Psi_2\}$, using the notation from Example 3.68. By Theorem 6.30, all words $\boldsymbol{t}_{b,3}$ for $b \geq 2$ are G-rich.*

First, let us take $b = 2$. The sequence of non-negative integers in the base 2 representation is

$$0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, \ldots$$

and the corresponding sequence of sums of digits modulo 3 is

$$0, 1, 1, 2, 1, 2, 2, 0, 1, 2, 2, 0, 2, 0, 0, 1 \ldots$$

Hence $\boldsymbol{t}_{2,3} = 0112122012202001\ldots$ By Lemma 6.29, it is a fixed point of the morphism given by

$$\varphi_{2,3}(0) = 01,$$
$$\varphi_{2,3}(1) = 12,$$
$$\varphi_{2,3}(2) = 20.$$

Note that $\varphi_{2,3}^3$ is given by

$$\varphi_{2,3}^3(0) = 01121220,$$
$$\varphi_{2,3}^3(1) = 12202001,$$
$$\varphi_{2,3}^3(2) = 20010112$$

and we observe that $\varphi_{2,3}^3(0)$ is a Ψ_0-palindrome, let us denote it by p, $\varphi_{2,3}^3(1) = \Psi_1(p)$ and $\varphi_{2,3}^3(2) = \Psi_2(p)$. Therefore, $\varphi_{2,3}^3$ belongs to the class $\mathcal{G}$ given in Proposition 5.56.

Second, let us take $b = 3$. The sequence of non-negative integers in the base 3 representation is

$$0, 1, 2, 10, 11, 12, 20, 21, 22, 100, 101, 102, 110, 111, 112, 120, \ldots$$

and the corresponding sequence of sums of digits modulo 3 *is*

$$0, 1, 2, 1, 2, 0, 2, 0, 1, 1, 2, 0, 2, 0, 1, 0, \ldots$$

Hence $\boldsymbol{t}_{3,3} = 0121202011202010\ldots$ *By Lemma 6.29, it is a fixed point of the morphism given by*

$$\begin{aligned}\varphi_{3,3}(0) &= 012,\\ \varphi_{3,3}(1) &= 120,\\ \varphi_{3,3}(2) &= 201.\end{aligned}$$

Note that $\varphi_{3,3}^3$ *is given by*

$$\begin{aligned}\varphi_{3,3}^3(0) &= 012120201120201012201012120,\\ \varphi_{3,3}^3(1) &= 120201012201012120012120201,\\ \varphi_{3,3}^3(2) &= 201012120012120201120201012\end{aligned}$$

and it again belongs to the class $\mathcal{G}$ *given in Proposition 5.56.*

We want to find some other examples of morphisms generating G-rich words for the group $G = \langle\{\Psi_0, \Psi_1, \Psi_2\}\rangle$ and also for the group $G = \langle\{R, D\}\rangle$. A necessary condition for an infinite word to be G-rich is that it is G-palindromic. Hence, we consider some examples of morphisms belonging to the class $\mathcal{G}$ and test whether their fixed points could be G-rich or not using a computer program. This is described in section 6.2.3, but first we need some more theory introduced in [36].

Definition 6.32. *Let* $v, w \in \mathcal{A}^*$ *and* $w = w_1 \ldots w_n$, *where* $w_k \in \mathcal{A}$ *for all* $k \in \{1, \ldots, n\}$.

An index $i \in \{1, \ldots, n+1\}$ *is called an occurrence of* v *in* w *if* $v = w_i \ldots w_j$ *for some* $j \in \{i, \ldots, n\}$ *or* $v = \varepsilon$.

An index $i \in \{1, \ldots, n+1\}$ *is called a* G*-occurrence of* v *in* w *if there exists* $u \in [v]_G$ *that has occurrence* i *in* w.

We say that v *is* G*-unioccurrent in* w *if* v *has an occurrence in* w *and no other index is a* G*-occurrence of* v *in* w.

Definition 6.33. *A suffix* v *of a word* $w \in \mathcal{A}^*$ *is called* G*-longest palindromic suffix (*G*-LPS) of* w, *if it is a* G*-palindrome and it satisfies* $|v| \geq |u|$ *for any* G*-palindromic suffix* u *of* w.

Note that it can happen that the G-LPS of some word w is the empty word ε.

Definition 6.34. *Let* $w = w_1 \ldots w_n \in \mathcal{A}^*$, *where* $w_k \in \mathcal{A}$ *for all* $k \in \{1, \ldots, n\}$. *An index* $i \in \{1, \ldots, n\}$ *is called a* G*-lacuna in* w *if both* w_i *and* G*-LPS of* $w_1 \ldots w_i$ *are not* G*-unioccurrent in* $w_1 \ldots w_i$.

The G-defect of a finite word w can be expressed using G-lacunas in w, as was shown in [36].

Proposition 6.35. *Let* $w \in \mathcal{A}^*$. *Then*

$$d_G(w) = \mathrm{card}(\{i \in \mathbb{N} \mid i \text{ is a } G\text{-lacuna in } w\}).$$

Example 6.36. *Let* $G = \langle\{R, E\}\rangle$ *and consider* $w = 01101111001$. *The words* $w_1 \ldots w_i$ *for* $i \in \{1, \ldots, 11\}$ *are* 0, 01, 011, 0110, 01101, 011011, 0110111, 01101111, 011011110, 0110111100 *and* 01101111001 *with their* G*-LPS being* 0, 01, 11, 0110, 101, 11011, 111, 1111, 011110, 1100 *and* 1001, *respectively.*

For each $i \in \{1, \ldots, 10\}$, G-LPS of $w_1 \ldots w_i$ is G-unioccurrent in $w_1 \ldots w_i$ and hence i is not a G-lacuna in w.

For $i = 11$, $w_{11} = 1$ is not G-unioccurrent in $w_1 \ldots w_{11} = w$ and also G-LPS of w, which is the word 1001, is not G-unioccurrent in w, since $0110 \in [1001]_G$ has occurrence $j = 1$ in w. Therefore, $i = 11$ is a G-lacuna in w.

By Proposition 6.35, we have

$$d_G(w) = \mathrm{card}(\{i \in \mathbb{N} \mid i \textit{ is a } G\textit{-lacuna in } w\}) = \mathrm{card}(\{11\}) = 1 \neq 0,$$

so w is not G-rich.

We can check that this is indeed correct by considering the set $\mathbb{P}_G(w)$, which is equal to $\{[\varepsilon]_G, [0]_G, [01]_G, [11]_G, [101]_G, [111]_G, [0110]_G, [1111]_G, [1100]_G, [11011]_G, [011110]_G\}$. Then, by definition,

$$d_G(w) = |w| + 1 - \gamma_G(w) - \mathrm{card}(\mathbb{P}_G(w)) = 11 + 1 - 0 - 11 = 1.$$

Example 6.37. *Let $G = \{\mathrm{Id}, D\}$ and consider $w = \mathrm{TTAA}$. By Remark 6.25, $d_G(w) = d_H(w)$, and from Example 6.16, we know that*

$$d_H(w) = |w| + 1 - \gamma_H(w) - \mathrm{card}(\mathbb{P}_H(w)) = 4 + 1 - 1 - 3 = 1.$$

Let us now find the G-lacunas of w. The words $w_1 \ldots w_i$ for $i \in \{1, \ldots, 4\}$ are T, TT, TTA *and* TTAA *with their G-LPS being ε, ε,* TA *and* TTAA, *respectively.*

For $i = 1$, $w_1 = \mathrm{T}$ is G-unioccurrent in $w_1 = \mathrm{T}$ and hence $i = 1$ is not a G-lacuna in w.

For $i = 2$, $w_2 = \mathrm{T}$ is not G-unioccurrent in $w_1 w_2 = \mathrm{TT}$ and also G-LPS of TT, *which is the empty word ε, is not G-unioccurrent in* TT. *Therefore, $i = 2$ is a G-lacuna in w.*

For $i = 3$ and $i = 4$, G-LPS of $w_1 \ldots w_i$ is G-unioccurrent in $w_1 \ldots w_i$ and hence i is not a G-lacuna in w.

Hence,

$$d_G(w) = \mathrm{card}(\{i \in \mathbb{N} \mid i \textit{ is a } G\textit{-lacuna in } w\}) = \mathrm{card}(\{2\}) = 1,$$

as expected.

6.2.3 Calculation of G-defect

In this section, we describe algorithms that we use to determine whether a given finite word $w = w_1 \ldots w_n$ is G-rich or not for two specific groups G, namely $G = \langle\{R, D\}\rangle$ and $G = \langle\{\Psi_0, \Psi_1, \Psi_2\}\rangle$. The algorithms calculate the G-defect of w using the relation from Proposition 6.35. In the algorithms, we use the notation $w[i : j] \coloneqq w_i \ldots w_j$ for $i, j \in \{1, \ldots, n\}$, $i \leq j$. We also make use of the following lemmas.

Lemma 6.38. *Let $G = \langle\{R, D\}\rangle$ and let p be a G-palindrome. Then*

$$[p]_G = \{R(p), D(p)\}.$$

Proof. Since $G = \{\mathrm{Id}, R, D, R \circ D\}$, we have

$$[p]_G = \{p, R(p), D(p), R \circ D(p)\}. \tag{6.1}$$

It is easy to see that $R \circ D = D \circ R$. We know about p that it is either an R-palindrome or a D-palindrome.

First, consider the case when p is an R-palindrome, so $p = R(p)$. Then

$$R \circ D(p) = D \circ R(p) = D(p),$$

and hence it follows from equality (6.1) that $[p]_G = \{R(p), D(p)\}$.

Second, assume that p is a D-palindrome, so $p = D(p)$. Then

$$R \circ D(p) = R(p),$$

and hence equality (6.1) implies that $[p]_G = \{R(p), D(p)\}$. □

Lemma 6.39. *Let $G = \langle\{\Psi_0, \Psi_1, \Psi_2\}\rangle$ and let p be a G-palindrome. Then*

$$[p]_G = \{\Psi_0(p), \Psi_1(p), \Psi_2(p)\}.$$

Proof. We can write $G = \{\mathrm{Id}, \mu, \mu^{-1}, \Psi_0, \Psi_1, \Psi_2\}$ using the notation from Example 3.68. Hence,

$$[p]_G = \{p, \mu(p), \mu^{-1}(p), \Psi_0(p), \Psi_1(p), \Psi_2(p)\}. \tag{6.2}$$

In Example 3.68, the Cayley table of G was given, and we use the relations from this table below. We have three possibilities here.

First, if $p = \Psi_0(p)$, then

$$\begin{aligned}\mu(p) &= \mu \circ \Psi_0(p) = \Psi_1(p)\\ \mu^{-1}(p) &= \mu^{-1} \circ \Psi_0(p) = \Psi_2(p).\end{aligned}$$

Second, if $p = \Psi_1(p)$, then

$$\begin{aligned}\mu(p) &= \mu \circ \Psi_1(p) = \Psi_2(p)\\ \mu^{-1}(p) &= \mu^{-1} \circ \Psi_1(p) = \Psi_0(p).\end{aligned}$$

Third, if $p = \Psi_2(p)$, then

$$\begin{aligned}\mu(p) &= \mu \circ \Psi_2(p) = \Psi_0(p)\\ \mu^{-1}(p) &= \mu^{-1} \circ \Psi_2(p) = \Psi_1(p).\end{aligned}$$

Overall, we get from equality (6.2) that $[p]_G = \{\Psi_0(p), \Psi_1(p), \Psi_2(p)\}$. □

Lemma 6.40. *Let v be a G-palindrome. Then every $u \in [v]_G$ is a G-palindrome.*

Proof. Since v is a G-palindrome, there exists an antimorphism $\theta \in G$ such that $v = \theta(v)$. Let us take some $u \in [v]_G$. This means that $u = \sigma(v)$ for some $\sigma \in G$. Then $\sigma \circ \theta \circ \sigma^{-1} \in G$ is an antimorphism and we have

$$\sigma \circ \theta \circ \sigma^{-1}(u) = \sigma \circ \theta(v) = \sigma(v) = u,$$

hence u is a G-palindrome. □

Lemma 6.41. *Let $n \in \mathbb{N}$ and let $w = w_1 \dots w_n \in \mathcal{A}^*$. Then we have*

$$\mathbb{P}_G(w) = \{[\varepsilon]_G\} \cup \{[v]_G \mid v \textit{ is a } G\textit{-LPS of } w_1 \dots w_i \textit{ for some } i \in \{1, \dots, n\}\}.$$

Proof. Clearly, $\mathbb{P}_G(w)$ contains the union of the two sets on the right-hand side of the equality. Hence, it remains to show that $\mathbb{P}_G(w)$ is also contained in this union. By definition,

$$\mathbb{P}_G(w) = \{[v]_G \mid v \text{ is a factor of } w \text{ and a } G\text{-palindrome}\}.$$

So let us consider some G-palindromic factor u of w. We want to show that $[u]_G$ is an element of the union $\{[\varepsilon]_G\} \cup \{[v]_G \mid v \text{ is a } G\text{-LPS of } w_1 \dots w_i \text{ for some } i \in \{1, \dots, n\}\}$. If $u = \varepsilon$, then this is trivial. Hence, suppose that $u \neq \varepsilon$. Let $m = |u|$ and let j be the first G-occurrence of u in w. It follows that there exists $s \in [u]_G$ such that s is a suffix of $w_1 \dots w_{j+m-1}$ and s is G-unioccurrent in $w_1 \dots w_{j+m-1}$. Moreover, by Lemma 6.40, s is a G-palindrome. We will show that s is a G-LPS of $w_1 \dots w_{j+m-1}$.

Assume, for the sake of contradiction, that there is a G-palindromic suffix p of $w_1 \dots w_{j+m-1}$ such that $|p| > |s|$. Let us denote the occurrence of p in $w_1 \dots w_{j+m-1}$ as k. Then $k < j$ and it follows that s is a proper suffix of p, i.e., $p = rs$ for some $r \neq \varepsilon$. Since p is a G-palindrome, there exists an antimorphism $\theta \in G$ such that $rs = p = \theta(p) = \theta(s)\theta(r)$. This implies that k is a G-occurrence of s in $w_1 \dots w_{j+m-1}$, but since $j > k$ is also a G-occurrence of s in $w_1 \dots w_{j+m-1}$, it is in contradiction with the fact that s is G-unioccurrent in $w_1 \dots w_{j+m-1}$.

Hence, s is a G-LPS of $w_1 \dots w_{j+m-1}$ and so $[u]_G = [s]_G$ is an element of the set $\{[v]_G \mid v \text{ is a } G\text{-LPS of } w_1 \dots w_i \text{ for some } i \in \{1, \dots, n\}\}$. □

1. $G = \langle\{R, D\}\rangle$

Firstly, we take $G = \langle\{R, D\}\rangle$ and describe the algorithm calculating G-defect of a finite word $w \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}^*$ with $|w| = n$, see Algorithm 1 below. At the beginning, the variable called *defect* is set to zero (line 2). For each index $j \in \{1, \dots, n\}$ the algorithm decides whether j is a G-lacuna or not. This happens in the for loop starting at line 7. If j is a G-lacuna, the *defect* is increased by 1 (lines 30 and 32). At the end of the calculation, the value of the *defect* corresponds to the G-defect of w.

An index j is a G-lacuna if both w_j and G-LPS of $w_1 \dots w_j$ are not G-unioccurrent in $w_1 \dots w_j$. To check the first condition, we create a variable called *letterClasses* which is initialized to be the set $\{[a]_G \mid a \in \mathcal{A}\} = \{\{\mathrm{A}, \mathrm{T}\}, \{\mathrm{C}, \mathrm{G}\}\}$ (line 3). This variable corresponds to classes of letters that do not yet occur in the prefix of w that has been processed up to that point. For each index $j \in \{1, \dots, n\}$, it is first checked whether the equivalence class of w_j, which is equal to $\{w_j, D(w_j)\}$, is in the set *letterClasses* or not (line 8). If it belongs to this set, it means that w_j is G-unioccurrent in $w_1 \dots w_j$ and hence j cannot be a G-lacuna. Then, $\{w_j, D(w_j)\}$ is removed from *letterClasses* (line 9). Note that this is always the case for $j = 1$. If it does not belong to the set *letterClasses*, w_j is not G-unioccurrent in $w_1 \dots w_j$ and so j can be a G-lacuna, but then the second condition has to be verified.

In order to check the second condition, we create a variable called *GpalClasses*, which corresponds to the set of equivalence classes of G-palindromes of length greater than 1 that have already occurred in w (line 4). By Lemma 6.41, this set can be obtained by adding only the equivalence classes of words v such that $|v| > 1$ and v is a G-LPS of some prefix of w. For the given index $j > 1$, the algorithm finds the G-LPS of $w_1 \dots w_j$ (we describe this part of the algorithm below), and the G-LPS is stored in variable *Glps* (line 28). If *Glps* is only one letter, than it follows from the first condition that it is not G-unioccurrent in $w_1 \dots w_j$ and hence j is a G-lacuna (lines 29 and 30). If $|Glps| > 1$, the algorithm checks whether the equivalence class of *Glps*, which, by Lemma 6.38, is equal to $\{R(Glps), D(Glps)\}$, belongs to *GpalClasses* (line 31). If it does, then *Glps* is not G-unioccurrent in $w_1 \dots w_j$ and therefore j is a G-lacuna (line 32). If it does not, j cannot be a G-lacuna, since *Glps* is G-unioccurrent in $w_1 \dots w_j$. In this case, the equivalence class of *Glps* is added in the set *GpalClasses* (line 34). After that, the algorithm moves to the next index, unless $j = n$, in which case the value of *defect* is outputted.

Algorithm 1 RD-defect(w)

```
1: n ← |w|
2: defect ← 0
3: letterClasses ← {{A, T}, {C, G}}
4: GpalClasses ← {}
5: prevR-index, R-index ← {}, {}
6: prevD-index, D-index ← {}, {}
7: for j ∈ {1, ..., n} do
8:    if {w_j, D(w_j)} ∈ letterClasses then
9:       remove {w_j, D(w_j)} from letterClasses
10:   else
11:      G-index ← {}
12:      prevR-index ← R-index
13:      add j − 1, j in prevR-index
14:      R-index ← {}
15:      for i ∈ prevR-index do
16:         if w_i = w_j then
17:            add i in G-index
18:            if i ≠ 1 then
19:               add i − 1 in R-index
20:      prevD-index ← D-index
21:      D-index ← {}
22:      add j − 1 in prevD-index
23:      for i ∈ prevD-index do
24:         if  w_i = D(w_j) then
25:            add i in G-index
26:            if i ≠ 1 then
27:               add i − 1 in D-index
28:      Glps ← w[min(G-index) : j]
29:      if |Glps| = 1 then
30:         defect ← defect + 1
31:      else if {R(Glps), D(Glps)} ∈ GpalClasses then
32:         defect ← defect + 1
33:      else
34:         add {R(Glps), D(Glps)} in GpalClasses
35: return  defect
```

The G-LPS of $w_1 \dots w_j$ for $j > 1$ is obtained by finding all G-palindromic suffixes of $w_1 \dots w_j$ while storing their occurrences in the set *G-index* and then taking the one that is longest, i.e., with the lowest index of occurrence (lines 11-28). In the process of finding all G-palindromic suffixes, we use the fact that for $i \leq j$, $w_i \dots w_j$ is an H-palindrome if and only if $w_i = H(w_j)$ and $w_{i+1} \dots w_{j-1}$, which we consider to be empty if $i > j - 2$, is an H-palindrome.

In order to find all R-palindromic suffixes of $w_1 \dots w_j$, we use the sets *prevR-index* and *R-index*. The set *prevR-index* contains all the indices i for which $w_i \dots w_j$ is an R-palindrome if and only if $w_i = R(w_j) = w_j$. This means that either $i \in \{j-1, j\}$ or $w_{i+1} \dots w_{j-1}$ is an R-palindrome. The indices i such that $w_{i+1} \dots w_{j-1}$ is an R-palindrome are taken from the previous step of the calculation when we considered the index $j-1$, and they correspond exactly to the indices in the set *R-index* at the end of that step. Hence, *prevR-index* gets all the elements of the set *R-index* from the previous step of the calculation (line 12) plus the elements $j-1$ and j (line 13). The set *R-index* is then emptied (line 14). To find the occurrences of all R-palindromic suffixes of $w_1 \dots w_j$, it is then sufficient to take only the indices i from the set *prevR-index* that satisfy $w_i = w_j$ (lines 15-17). In addition, for each such index $i \neq 1$ we add the index $i-1$ in the set *R-index* (lines 18-19).

Finding all D-palindromic suffixes of $w_1 \dots w_j$ is analogous (lines 20-27). The only difference is that now we know that no D-palindrome of length 1 exists, so the suffix w_j cannot be a D-palindrome. Hence, we do not add the index j in the set *prevD-index* as we did in the case of *prevR-index* (line 22).

Having this algorithm, we conducted some computer experiments to test whether some morphisms from the class $\mathcal{G}$ could generate G-rich words.

Example 6.42. *Let $G = \langle\{R, D\}\rangle$ and consider morphisms from the class $\mathcal{G}$ of the form*

$$\begin{aligned}
\varphi(\mathrm{A}) &= (\mathrm{AT})^i\,(\mathrm{CG})^j\,\mathrm{C}\,(\mathrm{TA})^i,\\
\varphi(\mathrm{C}) &= (\mathrm{AT})^k\,\mathrm{A},\\
\varphi(\mathrm{G}) &= (\mathrm{TA})^k\,\mathrm{T},\\
\varphi(\mathrm{T}) &= (\mathrm{TA})^i\,(\mathrm{GC})^j\,\mathrm{G}\,(\mathrm{AT})^i,
\end{aligned}$$

where $i, j, k \in \mathbb{N}$. We tested whether the fixed points of these morphisms could be G-rich or not by considering a prefix of such a fixed point starting with A and calculating its G-defect using Algorithm 1. The results we obtained are that for all combination of $i, j, k \in \{1, \dots, 5\}$ the corresponding prefix of length 10000 has G-defect equal to 0. Therefore, it seems plausible that fixed points of the morphisms of this form are G-rich.

These results lead us to formulating the following conjecture.

Conjecture 6.43. *Let $G = \langle\{R, D\}\rangle$. Then fixed points of morphisms of the form*

$$\begin{aligned}
\varphi(\mathrm{A}) &= (\mathrm{AT})^i\,(\mathrm{CG})^j\,\mathrm{C}\,(\mathrm{TA})^i,\\
\varphi(\mathrm{C}) &= (\mathrm{AT})^k\,\mathrm{A},\\
\varphi(\mathrm{G}) &= (\mathrm{TA})^k\,\mathrm{T},\\
\varphi(\mathrm{T}) &= (\mathrm{TA})^i\,(\mathrm{GC})^j\,\mathrm{G}\,(\mathrm{AT})^i,
\end{aligned}$$

where $i, j, k \in \mathbb{N}$, are G-rich.

2. $G = \langle\{\Psi_0, \Psi_1, \Psi_2\}\rangle$

Let us now take the group $G = \langle\{\Psi_0, \Psi_1, \Psi_2\}\rangle$ and again consider the algorithm calculating G-defect of a finite word $w \in \{0, 1, 2\}^*$ with $|w| = n$, see Algorithm 2. It is analogous to Algorithm 1 described above. However, in this case we can determine whether an index $j \in \{1, \dots, n\}$ is a G-lacuna without checking if w_j is G-unioccurrent in $w_1 \dots w_j$ or not. This is because there is only one class of equivalence of letters, i.e., $\{[a]_G \mid a \in \mathcal{A}\} = \{\{0, 1, 2\}\}$, and hence w_j is G-unioccurrent in $w_1 \dots w_j$ only for $j = 1$. So $j = 1$ cannot be a G-lacuna and for $j \in \{2, \dots, n\}$ we only check whether the G-LPS of $w_1 \dots w_j$ is G-unioccurrent in $w_1 \dots w_j$ or not. This is done in the same way as in Algorithm 1. Here we use Lemma 6.39 as an analogy of Lemma 6.38.

Algorithm 2 Psi012-defect(w)

```
n ← |w|
defect ← 0
GpalClasses ← {}
prevPsi0-index, Psi0-index ← {}, {}
prevPsi1-index, Psi1-index ← {}, {}
prevPsi2-index, Psi2-index ← {}, {}
for j ∈ {2, . . . , n} do
  G-index ← {}
  prevPsi0-index ← Psi0-index
  add j − 1, j in prevPsi0-index
  Psi0-index ← {}
  for i ∈ prevPsi0-index do
    if w_i = Ψ_0(w_j) then
      add i in G-index
      if i ≠ 1 then
        add i − 1 in Psi0-index
  prevPsi1-index ← Psi1-index
  add j − 1, j in prevPsi1-index
  Psi1-index ← {}
  for i ∈ prevPsi1-index do
    if w_i = Ψ_1(w_j) then
      add i in G-index
      if i ≠ 1 then
        add i − 1 in Psi1-index
  prevPsi2-index ← Psi2-index
  add j − 1, j in prevPsi2-index
  Psi2-index ← {}
  for i ∈ prevPsi2-index do
    if w_i = Ψ_2(w_j) then
      add i in G-index
      if i ≠ 1 then
        add i − 1 in Psi2-index
  Glps ← w[min(G-index) : j]
  if |Glps| = 1 then
    defect ← defect + 1
  else if {Ψ_0(Glps), Ψ_1(Glps), Ψ_2(Glps)} ∈ GpalClasses then
    defect ← defect + 1
  else
    add {Ψ_0(Glps), Ψ_1(Glps), Ψ_2(Glps)} in GpalClasses
return defect
```

Example 6.44. *Let $G = \langle\{\Psi_0, \Psi_1, \Psi_2\}\rangle$ and consider morphisms from the class $\mathcal{G}$ of the form*

$$\varphi(0) = (0102)^i\,0,$$
$$\varphi(1) = (1210)^i\,1,$$
$$\varphi(2) = (2021)^i\,2,$$

where $i \in \mathbb{N}$. Using Algorithm 2, we found out that for each $i \in \{1, \ldots, 50\}$ the prefix of length 10000 of the fixed point of φ starting with the letter 0 has G-defect equal to 0.

Hence, the results of our experiments suggest that fixed points of these morphisms are G-rich.

Example 6.45. *Let* $G = \langle\{\Psi_0, \Psi_1, \Psi_2\}\rangle$ *and consider morphisms* φ *and* ψ *from the class* $\mathcal{G}$ *of the form*

$$\begin{aligned}\varphi(0) &= (0)^i\, 12\, (0)^i,\\ \varphi(1) &= (1)^i\, 20\, (1)^i,\\ \varphi(2) &= (2)^i\, 01\, (2)^i,\end{aligned}$$

and

$$\begin{aligned}\psi(0) &= 0\, (12)^j\, 0,\\ \psi(1) &= 1\, (20)^j\, 1,\\ \psi(2) &= 2\, (01)^j\, 2,\end{aligned}$$

where $i, j \in \mathbb{N}$. *For all* $i, j \in \{1, \ldots, 50\}$, *we tested the fixed points of these morphisms starting with the letter 0. We obtained that the G-defect of the prefix of length 10000 of each of these fixed points has G-defect equal to 0. This leads us to the belief that fixed points of these morphisms* φ *and* ψ *are G-rich.*

However, if we take morphisms σ *from the class* $\mathcal{G}$ *of the form*

$$\begin{aligned}\sigma(0) &= (0)^i\, (12)^j\, (0)^i,\\ \sigma(1) &= (1)^i\, (20)^j\, (1)^i,\\ \sigma(2) &= (2)^i\, (01)^j\, (2)^i,\end{aligned}$$

where $i, j \in \{2, 3, \ldots\}$, *we do not get the same results. Our calculations show that for all* $i, j \in \{2, \ldots, 5\}$ *the fixed points of* σ *are not G-rich.*

Again, based on our results we formulate the following conjectures.

Conjecture 6.46. *Let* $G = \langle\{\Psi_0, \Psi_1, \Psi_2\}\rangle$. *Then fixed points of morphisms of the form*

$$\begin{aligned}\varphi(0) &= (0102)^i\, 0,\\ \varphi(1) &= (1210)^i\, 1,\\ \varphi(2) &= (2021)^i\, 2,\end{aligned}$$

where $i \in \mathbb{N}$, *are G-rich.*

Conjecture 6.47. *Let* $G = \langle\{\Psi_0, \Psi_1, \Psi_2\}\rangle$. *Then fixed points of morphisms* φ *and* ψ *of the form*

$$\begin{aligned}\varphi(0) &= (0)^i\, 12\, (0)^i,\\ \varphi(1) &= (1)^i\, 20\, (1)^i,\\ \varphi(2) &= (2)^i\, 01\, (2)^i,\end{aligned}$$

and

$$\begin{aligned}\psi(0) &= 0\, (12)^j\, 0,\\ \psi(1) &= 1\, (20)^j\, 1,\\ \psi(2) &= 2\, (01)^j\, 2,\end{aligned}$$

where $i, j \in \mathbb{N}$, *are G-rich.*

Chapter 7

Conclusion

To conclude, we have investigated some areas of combinatorics on words, working with general alphabets as well as more specifically with the alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$, which is motivated by the molecule of DNA, as was discussed in Chapter 2. In this chapter, we also reviewed the field of DNA computing, where study of symmetries in words over the alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ could potentially be used.

In Chapter 3, we summarized some essential concepts from combinatorics on words while including our proofs for a number of the known results. We focused mainly on two types of functions on words, namely morphisms and antimorphisms. In addition, we discussed finite groups composed of these mappings.

In Chapter 4, we derived new results about equations on words with palindromes. One of these results is used in the proof of Theorem 5.55, which specifies the form of a morphism φ that is conjugated both to a morphism in class $\mathcal{P}$ and to a morphism in class $\mathcal{D}$ and satisfies $\varphi(\mathrm{A}) \neq \varphi(\mathrm{T})$ or $\varphi(\mathrm{C}) \neq \varphi(\mathrm{G})$.

At the centre of our focus were fixed points of primitive morphisms. One goal of our work was to find a way how to generate uniformly recurrent H-palindromic and more generally G-palindromic words. This was discussed in Chapter 5. Firstly, we gave an overview of known results about class $\mathcal{P}$ and of different versions of the HKS conjecture relating this class to R-palindromic words. We also summarized results of [5] regarding classes $\mathcal{A}_1$ and $\mathcal{A}_2$ for E-palindromic words. Then, we suggested a new class of morphisms $\mathcal{H}$ as an analogy to the previously mentioned classes for H-palindromic words. We showed, however, that not all morphisms from this class generate H-palindromic words, so we developed a way how to distinguish morphisms that generate H-palindromic words using directed graphs. We demonstrated our results on the example of antimorphism D. Our hypothesis is that an analogy of the HKS conjecture holds for the class $\mathcal{H}$ and all H-palindromic fixed points of primitive morphisms are related to morphisms from class $\mathcal{H}$.

In the case of G-palindromic words, we defined a class of morphisms $\mathcal{G}$ by a set of relations such morphisms have to satisfy. We proved that under some conditions morphisms from class $\mathcal{G}$ generate G-palindromic words. We discussed concrete examples of groups $\mathcal{G}$, which are generated by two involutive antimorphisms, and for those cases derived the specific form of a morphism belonging to class $\mathcal{G}$. We also showed that class $\mathcal{G}$ is closely related to classes $\mathcal{H}$ for involutive antimorphisms H in G and this suggests that there is a more general approach to generating G-palindromic words. However, this approach seems overly complicated and it is not clear whether it would bring something new. In fact, Theorem 5.55 suggests that in the case of the group $G = \langle\{R, D\}\rangle$ the approach with class $\mathcal{G}$ is sufficient. However, investigating the more general approach for an arbitrary group G still remains to be an open problem. This would provide further details about characterising G-palindromic fixed points of primitive morphisms.

Chapter 6 was devoted to the concept of palindromic richness. We started by introducing palindromic richness in the classical sense and then reviewed different generalizations of this notion. The most interesting generalization is with respect to a group of morphisms and antimorphisms and it is called G-richness. Our objective was to find some examples of G-rich infinite words. An infinite word can be G-rich only if it is G-palindromic. Hence, we examined G-palindromic fixed points of morphisms from class $\mathcal{G}$, and using a computer program, we tested whether they could be G-rich or not. By these experiments, we discovered several classes of morphisms that are likely generating G-rich words. It is an incentive for further research to prove that fixed points of these morphisms are indeed G-rich.

Bibliography

[1] Leonard M Adleman. Molecular Computation of Solutions to Combinatorial Problems. *Science*, 266(5187):1021–1024, 1994.

[2] Jean-Paul Allouche, Michael Baake, Julien Cassaigne, and David Damanik. Palindrome Complexity. *Theoretical Computer Science*, 292(1):9–31, 2003.

[3] Jean-Paul Allouche and Jeffrey Shallit. Sums of Digits, Overlaps, and Palindromes. *Discrete Mathematics and Theoretical Computer Science*, 4(1):1–10, 2000.

[4] Petr Ambrož, Ondřej Kadlec, Zuzana Masáková, and Edita Pelantová. Palindromic Length of Words and Morphisms in Class $\mathcal{P}$. *Theoretical Computer Science*, 780:74–83, 2019.

[5] Petr Ambrož, Zuzana Masáková, and Edita Pelantová. Morphisms Generating Antipalindromic Words. *European Journal of Combinatorics*, 89:103160, 2020.

[6] Martyn Amos. *DNA Computation*. PhD thesis, University of Warwick, 1997.

[7] Sanjeev Arora and Boaz Barak. *Computational Complexity: a Modern Approach*. Cambridge University Press, 2009.

[8] L'ubomíra Balková, Edita Pelantová, and Štěpán Starosta. Sturmian Jungle (or Garden?) on Multiliteral Alphabets. *RAIRO-Theoretical Informatics and Applications*, 44(4):443–470, 2010.

[9] Donald Beaver. *Molecular Computing*. Pennsylvania State University, Department of Computer Science and Engineering, College of Engineering, 1995.

[10] Yaakov Benenson, Rivka Adar, Tamar Paz-Elizur, Zvi Livneh, and Ehud Shapiro. DNA Molecule Provides a Computing Machine with Both Data and Fuel. *Proceedings of the National Academy of Sciences*, 100(5):2191–2196, 2003.

[11] Yaakov Benenson, Binyamin Gil, Uri Ben-Dor, Rivka Adar, and Ehud Shapiro. An Autonomous Molecular Computer for Logical Control of Gene Expression. *Nature*, 429(6990):423–429, 2004.

[12] Valérie Berthé and Michel Rigo. *Combinatorics, Automata and Number Theory*. Number 135. Cambridge University Press, 2010.

[13] Alexandre Blondin-Massé, Srecko Brlek, Ariane Garon, and Sébastien Labbé. Combinatorial Properties of f-palindromes in the Thue-Morse Sequence. *Pure Mathematics and Applications*, 19(2-3):39–52, 2008.

[14] Srecko Brlek, Sylvie Hamel, Maurice Nivat, and Christophe Reutenauer. On the Palindromic Complexity of Infinite Words. *International Journal of Foundations of Computer Science*, 15(02):293–306, 2004.

[15] Michelangelo Bucci, Alessandro De Luca, Amy Glen, and Luca Q Zamboni. A Connection Between Palindromic and Factor Complexity Using Return Words. *Advances in Applied Mathematics*, 42(1):60–74, 2009.

[16] Julien Cassaigne. Complexité et facteurs spéciaux. *Bulletin of the Belgian Mathematical Society-Simon Stevin*, 4(1):67–88, 1997.

[17] Andrew Currin, Konstantin Korovin, Maria Ababi, Katherine Roper, Douglas B Kell, Philip J Day, and Ross D King. Computing Exponentially Faster: Implementing a Non-Deterministic Universal Turing Machine Using DNA. *Journal of the Royal Society Interface*, 14(128):20160990, 2017.

[18] Xavier Droubay, Jacques Justin, and Giuseppe Pirillo. Episturmian Words and Some Constructions of de Luca and Rauzy. *Theoretical Computer Science*, 255(1-2):539–553, 2001.

[19] Richard P Feynman. There's Plenty of Room at the Bottom. *Engineering and Science*, 23(5):22–36, 1960.

[20] Amy Glen, Jacques Justin, Steve Widmer, and Luca Q Zamboni. Palindromic Richness. *European Journal of Combinatorics*, 30(2):510–531, 2009.

[21] Chuan Guo, Jeffrey Shallit, and Arseny M Shur. On the Combinatorics of Palindromes and Antipalindromes. *arXiv preprint arXiv:1503.09112*, 2015.

[22] Tero Harju, Jetro Vesti, and Luca Q Zamboni. On a Question of Hof, Knill and Simon on Palindromic Substitutive Systems. *Monatshefte für Mathematik*, 179(3):379–388, 2016.

[23] Albertus Hof, Oliver Knill, and Barry Simon. Singular Continuous Spectrum for Palindromic Schrödinger Operators. *Communications in Mathematical Physics*, 174(1):149–159, 1995.

[24] Salah Hussini, Lila Kari, and Stavros Konstantinidis. Coding Properties of DNA Languages. *Theoretical Computer Science*, 290(3):1557–1579, 2003.

[25] Maya Kahan, Binyamin Gil, Rivka Adar, and Ehud Shapiro. Towards Molecular Computers That Operate in a Biological Environment. *Physica D: Nonlinear Phenomena*, 237(9):1165–1172, 2008.

[26] Lila Kari and Kalpana Mahalingam. Watson–Crick Palindromes in DNA Computing. *Natural Computing*, 9(2):297–316, 2010.

[27] Sébastien Labbé. Propriétés combinatoires des f-palindromes. Master's thesis, Université du Québec à Montréal, 2008.

[28] Sébastien Labbé. A Counterexample to a Question of Hof, Knill and Simon. *The Electronic Journal of Combinatorics*, 21(P3.11), 2014.

[29] Sébastien Labbé and Edita Pelantová. Palindromic Sequences Generated from Marked Morphisms. *European Journal of Combinatorics*, 51:200–214, 2016.

[30] Richard J Lipton. DNA Solution of Hard Computational Problems. *Science*, 268(5210):542–545, 1995.

[31] M. Lothaire. *Combinatorics on Words*, volume 17. Cambridge University Press, 1997.

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

[33] Alexandre Blondin Massé. *Sur le défaut palindromique des mots infinis*. PhD thesis, Université du Québec à Montréal, 2008.

[34] Suyel Namasudra and Ganesh Chandra Deka. *Advances of DNA Computing in Cryptography*. Chapman & Hall/CRC, 1st edition, 2018.

[35] Edita Pelantová and Štěpán Starosta. Languages Invariant Under More Symmetries: Overlapping Factors Versus Palindromic Richness. *Discrete Mathematics*, 313(21):2432–2445, 2013.

[36] Edita Pelantová and Štěpán Starosta. Palindromic Richness for Languages Invariant Under More Symmetries. *Theoretical Computer Science*, 518:42–63, 2014.

[37] Nadia Pisanti. DNA Computing: a Survey. *Bull. EATCS*, 64:188–216, 1998.

[38] Paul WK Rothemund. *A DNA and Restriction Enzyme Implementation of Turing Machines*, volume 27 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. DIMACS/AMS, 1996.

[39] Kensaku Sakamoto, Hidetaka Gouzu, Ken Komiya, Daisuke Kiga, Shigeyuki Yokoyama, Takashi Yokomori, and Masami Hagiya. Molecular Computation by DNA Hairpin Formation. *Science*, 288(5469):1223–1226, 2000.

[40] Ehud Shapiro and Yaakov Benenson. Bringing DNA Computers to Life. *Scientific American*, 294(5):44–51, 2006.

[41] Richard R Sinden. *DNA Structure and Function*. Gulf Professional Publishing, 1994.

[42] Štěpán Starosta. On Theta-Palindromic Richness. *Theoretical Computer Science*, 412(12-14):1111–1121, 2011.

[43] Štěpán Starosta. Generalized Thue-Morse Words and Palindromic Richness. *Kybernetika*, 48(3):361–370, 2012.

[44] Somnath Tagore, Saurav Bhattacharya, M Islam, and Md Lutful Islam. DNA Computation: Application and Perspectives. *Journal of Proteomics and Bioinformatics*, 3(07), 2010.

[45] Bo Tan. Mirror Substitutions and Palindromic Sequences. *Theoretical Computer Science*, 389(1-2):118–124, 2007.