Automatic Sequences

Fatlonder Cakolli

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Applied Mathematics and Computer Science

Eastern Mediterranean University
July 2023
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Applied Mathematics and Computer Science.

Prof. Dr. Nazım Mahmudov
Chair, Department of Mathematics

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Applied Mathematics and Computer Science.

Prof. Dr. Benedek Nagy
Supervisor

Examining Committee

1. Prof. Dr. Benedek Nagy
2. Asst. Prof. Dr. Sedef Emin
3. Asst. Prof. Dr. Müge Saadetoğlu

ABSTRACT

Finite automaton is a well known and utilized computational model. Automatic sequences' definition is bootstraped using the notion of finite automaton. More specifically for the definition we use DFA (Deterministic Finite Automaton) with an output function τ and call it DFAO (Deterministic Finite Automaton with Output). Looking from the Chomsky's hierarchy of languages it's exactly the regular type ones that the DFA model recognizes. Using the notion of finite automaton we can show properties such as cross product of automatic sequences and composition of output functions. Relation between morphisms and finite automaton is established for automaticity of a sequence. Using morphisms we can have an alternative way of treating the automatic sequences. Additionally the notion of $k-$Kernels is introduced and the relation is established with automatic sequences. The interest of finding the algebraicity of formal power series will lead to Christol's theorem which establishes the relation with automatic sequences, proving another way of representing automatic sequences by the means of formal power series, a notion from the broad field of algebra.

Keywords: automatic-sequence, formal-power-series, morphisms, finite-automaton

ÖZ

Sonlu otomat, iyi bilinen ve yaygın olarak kullanılan bir hesaplama modelidir. Otomatik dizilerin tanımı, sonlu otomat kavramı kullanılarak elde edilir. Daha spesifik olarak, tezimizde tanım icin çıktı fonksiyonu τ'ya sahip bir DFA (Deterministik Sonlu Otomat) kullanıp ve bunu DFAO (Çikti fonksiyonuna sahip deterministik sonlu otomat) olarak adlandırıyoruz. Chomsky'nin dil hiyerarşisine bakıldığında, DFA modelinin tanıdığı diller tam olarak düzenli tipteki dillerdir. Sonlu otomat kavramını kullanarak, otomatik dizilerin direkt çarpımı ve çıktı fonksiyonlarının birleşimi gibi özellikleri gösterebiliriz. Dizinin otomatikliği icin morfizmalarla sonlu otomatlar arasındaki bağlantıyı kurarız. Morfizmaları kullanarak otomatik dizileri incelemenin alternatif bir yolunu elde ederiz. Ayrıca, $k-$Çekirdek kavramı tanıtılır ve otomatik dizilerle ilişkisi kurulur. Formal kuvvet serilerinin cebirsel özelliklerini bulma ilgisi, Christol teoremi ile sonuçlanır. Bu teorem, otomatik dizilerle bağlantı kurarak, otomatik dizileri formal kuvvet seriler aracılığıyla, başka bir şekilde temsil etmenin mümkün olduğunu kanıtlar. Bu, cebirin geniş bir alanından gelen bir kavramdır.

Anahtar Kelimeler: otomatik-dizi, formal kuvvet serisi, morfizmalar, sonlu-otomat

Those that seek the light and shine it in the vacuum of the darkness

ACKNOWLEDGMENTS

The completion of this journey, representing the cyclical nature of every human endeavor, has been made possible thanks to the invaluable guidance and mentorship provided by professors Müge Saadetoğlu and Arran Fernandez, as well as the unwavering support of my supervisor Benedek Nagy. I am deeply grateful to them for their instrumental role in shaping my journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

τ	Output function
δ	State Transition Function
Σ	Set of symbols (Alphabet)
$\mathbb{P}$	Set of Prime Numbers
DFA	Deterministic Finite Automaton
DFAO	Deterministic Finite Automaton with Output

Chapter 1

INTRODUCTION

A common way of defining sequences over non-negative integers $\mathbb{N}$, which are denoted as $(a_n)_{n\geq 0}$, is to expressing it as a relation with indeterminates $n \in \mathbb{N}$ and possibly some constants $c \in \mathbb{R}$. Instead of expressing relations explicitly between variables and constants, another way is to have this relation using a finite automaton. Since, when evaluated, the deterministic finite automaton model doesn't behave as we expect for the sequences we extend the model of deterministic finite automaton to add an output function τ. The function τ corresponds to a deterministic finite automaton and operates on the input of the final accepted state $q \in Q$ for a given input $[n]_k$, here $[n]_k$ denotes the base k representation of the natural number $n \in \mathbb{N}$. This relationship is represented by $\tau(\delta(q_0, [n]_k))$. Then the output of the function τ is in some finite set Δ. Sequences that are defined using deterministic finite automaton with output function are called automatic sequences. An important notion that is borrowed from finite automaton is languages, which we use to treat these automatic sequences as languages over some alphabet. This enables to clasify a sequence as automatic or not using the tools of formal languages and finite automaton. A sequences is automatic sequences if and only if it forms a regular language, that is recognized by some deterministic finite automaton.

Morphisms provide an alternative approach to defining automatic sequences, offering a different view on their construction. In this context, one commonly used type of morphism is the k-uniform morphism. Such morphisms, denoted by φ, map elements

$a \in \Sigma$ to elements of fixed length k. By applying k-uniform morphisms to the elements of an automatic sequence, we obtain a sequence of words, each of which has a fixed length determined by k. These word sequences, generated by the morphism, provide a symbolic representation of the automatic sequence.

k-kernels are introduced as an important concept in the study of automatic sequences. A k-kernel of a sequence $\mathbf{a}$ is a finite set of subsequences that capture the essential patterns of the sequence modulo k. In other words, a k-kernel consists of representative subsequences that provide information about the repetition of elements in the original sequence. [9, 11]

Formal power series play a crucial role in the study of automatic sequences, and they are also fundamental objects in algebraic structures such as groups, rings, and fields [5, 10]. Each term of a formal power series corresponds to a coefficient of the sequence, with the power of the indeterminate representing the position of the coefficient in the sequence. These power series enable us to express automatic sequences algebraically, providing a compact representation that facilitates various mathematical operations and analyses. In the context of algebraic structures, formal power series possess algebraic properties similar to polynomials. They can be added, multiplied, and composed with each other, allowing for the manipulation and combination of automatic sequences. The addition of formal power series is carried out by adding the corresponding coefficients term by term, while multiplication involves the convolution of coefficients, akin to polynomial multiplication. Composition, on the other hand, entails substituting one power series into another, leading to the generation of new power series. By leveraging the algebraic properties of formal power series, we can explore the relationships between sequences and investigate various algebraic structures associated with automatic sequences. For instance, the set of formal power series with coefficients in a field forms a ring, denoted as the ring of formal power series over that field. The study of formal power series within the realm of automatic sequences provides a uniquee framework for investigating their algebraic properties and exploring connections to other areas of mathematics. Automatic sequences and the algebraicity of formal power series are closely intertwined concepts. An automatic sequence is a sequence that can be defined using a deterministic finite automaton with an output function. On the other hand, the algebraicity of a formal power series refers to the property of the series being a solution to a polynomial equation with coefficients in a given field. There is a strong connection between automatic sequences and the algebraicity of formal power series. Specifically, a sequence is automatic if and only if its corresponding generating formal power series is algebraic. This means that the properties of being automatic and algebraic are equivalent for sequences and their corresponding formal power series. The algebraicity of a formal power series implies that it satisfies a polynomial equation, which can provide insights into the underlying structure and behavior of the automatic sequence it represents. Conversely, if a sequence is known to be automatic, then we can infer the existence of a polynomial equation that governs its behavior [13]. The connection between automatic sequences and the algebraicity of formal power series provides a uniquee approach for studying and understanding these fundamental mathematical objects. The objective of this work is to provide a succinct overview of automatic sequences [7, 13].

Chapter 2

AUTOMATIC SEQUENCES

On the simplest way, automatic sequences are sequences of numbers on a given base that are generated by deterministic finite automaton with output (DFAO). Among the ways to defining automatic sequences is using a DFAO and the chapter's 2.1 focus is to introduce the notion of finite automaton with the extended model with output known as DFAO and languages [19]. On chapter 2.2 we formally define automatic sequences and explore its properties. Chapter 2.3 is focused on morphisims between sets, and the establishing the equivalent representation of automatic sequences. On chapter 2.4 we will introduce an equivalent way to DFAO that is called k-Kernel and explore its properties. The goal of Chapter 2.5 is to show an alternative representation, by Christol's theorem, of automatic sequences using formal power series.

2.1 Languages and Finite Automaton

2.1.1 Languages

Symbols, and words—together with languages which are defined over these symbols—are central pieces to the conceptualization of a computational model explored on the next section. We are very well familiar with these notions, of symbols that are used to make words and collection of words that make up a language, from the daily use a language. An alphabet is a non-empty finite set of symbols which has been denoted by the symbol Σ. Generally for any $k \in \mathbb{N},$ where, $k \geq 2$ then the set

$$\Sigma_k = \{0, 1, ..., k-2, k-1\}.$$

is defined as an alphabet of order k. A commonly, non-exhaustive, list of alphabets include:

$$\Sigma_2 = \{0,1\}, \text{binary alpabet},$$

$$\Sigma_3 = \{0,1,2\}, \text{ternary alphabet},$$

$$\Sigma_{10} = \{0,1,2,3,4,5,6,7,8,9\}, \text{decimal alphabet}.$$

The juxtapostion of symbols of a given alphabet is defined as word. For instance, we let the word 10010 be defined over the alphabet $\Sigma = \{0,1\}$. We let the symbol ε denote the empty word. An intrinsic property of words is their length, which is the number of positions with symbols over a given alphabet that are used to make that word. For the empty word ε, it is natural to call that it has the property of having the length equal to zero, $|\varepsilon| = 0$. For $a \in \Sigma$ and $x \in \Sigma^*$, the $|x|_a$ denotes the number of symbols a that occur on the word x. Powers of the alphabet Σ is to be understood as the set of words of a given length k. Irrespective of the alphabet Σ for $k = 0$ we have $\Sigma^0 = \{\varepsilon\}$. Σ^* denotes the set of all words of finite length over the alphabet Σ. Using the set notation we have the equality

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots.$$

Similarly the meaning of notation Σ^+ is defined as

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots.$$

A useful operation on the set Σ^k is the concatenation operator. For any $z \in \Sigma^k$, it can be expressed as a concatination of words $x, y \in \Sigma^k$ which is denoted $z = xy$. Importantly, for $x, y \in \Sigma^k$, $xy = x$ if and only if $y = \varepsilon$. And for $x, y \in \Sigma^k$, $xy = yx$ if and only if $x = y$. That is, concatenation operator is not cummutative. For a word $x = spq$ we say p, with $|p| \geq 1$, is a subword or factor of x, then for s, with $|s| \geq 1$, it is said to be a prefix of x, s is said to be a proper prefix of x if $pq \neq \varepsilon$ where $|s| \geq 1$, and we say q, with $|q| \geq 1$, is a suffix of x. Let $x = a_1 a_2 ... a_n$ be a word and $1 \leq i \leq n$, the i^{th} symbol is defined as $x[i] = a_i$, and if we have i, j such that $1 \leq i \leq n$ and $i - 1 \leq j \leq n$, then a finite subword of the word x starting from i to j is expressed as follows: $x[i, j] = a_i a_{i+1} ... a_j$.

For an alphabet Σ and the set Σ^*, as denoted above, a subset L of Σ^* is defined as a language over Σ.

Example 2.1: Some languages over specified alphabet:
Set of prime numbers over the binary alphabet:

$$\{10, 11, 101, 111, ...\}$$

Set of words with equal number of $0's$ and $1's$:

$$\{\varepsilon, 10, 01, 0011, 1100, ...\}$$

Given that the notion of languages is already in place, introduction of common operations on them naturally follows [2]. For given languages $L, L_1, L_2 \subseteq \Sigma^*$, the product of languages is defined as the set

$$L_1 L_2 = \{wx : w \in L_1, x \in L_2\}.$$

For languages, similar to the alphabets, we define $L^0 = \{\varepsilon\}$. For $i \geq 1$ we define L^i as LL^{i-1}. The Kleene closure operator is defined as $L^* = \bigcup_{i \geq 0} L^i$. The quotient of languages is defined as the set

$$L_1 / L_2 = \{x \in \Sigma^* : \exists y \in L_2 \text{ such that } xy \in L_1\}.$$

An important class of languages that bears a central role to automatic sequences is the class of regular languages. Regular expressions are defined over the alphabet Σ in conjunction with the special symbols $\{\varepsilon, \emptyset, (,), +, *\}$, which are not elements of the alphabet Σ [2]. When evaluation regular expressions the $*$ operator represents the

Kleene's closure and is evaluated first, followed by the concatenation, and then by $+$ which represents union. Additionally parentheses within the regular expression x are used to represent grouping of a regular expression. We denote $\emptyset$ as a basic regular expression that refers to empty language, ε as a basic regular expression that refers to $\{\varepsilon\}$ language, and for every $a \in \Sigma$ as a basic regular expression that refers to the language $\{a\}$. By utilizing the aforementioned operators and adhering to the precedence order, we combine these elementary regular expressions to construct another regular expressions. These expressions, denoted as x, define a set of regular language $L(x)$. On the Example 2.2 is shown a regular language specified using regular expression.

Example 2.2: Regular language specified by the regular expression $x = (10)^*$:

$$L(x) = \{\varepsilon, 10, 1010, 101010, ...\}$$

Example 2.3: Regular language specified by the regular expression $x = 1(10)^*1$:

$$L(x) = \{11, 1101, 11010101, ...\}$$

Following lemma shows a trivial property of finite languages.

Lemma 2.1: Finite languages are regular.

Proof. Using the $+$ operator we construct regular expression, for the finite language $L = \{w_1, w_2, w_3, \ldots, w_i\}$, as the concatenation of all the words $w \in L$. □

An infinite sequence $\mathbf{a} = a_0 a_1 a_2 \ldots$ where $a_i \in \Sigma$, $\forall i \in \mathbb{N}$ is called right-sided sequence, and can be see as a map from $\mathbb{N}$ to Σ.

Example 2.4: Consider the right-infinite word (or sequence) denoted as $(q_n)_{n \geq 0} =$

11001000010000001..., known as the characteristic sequence of perfect squares. In this sequence, q_n is equal to 1 if n is a perfect square, and 0 otherwise.

Maps from infinite set $\mathbb{N}$ to the finite set Σ is the set [1]

$$\Sigma^{\omega} = \{(a_n) : a_0 a_1 \cdots \in \Sigma\},$$

which we call the set of right-sided infinite sequences. Similary we define left-sided infinite sequence $...a_{-3}a_{-2}a_{-1}a_0$ as a map from $\mathbb{Z}^-$ to Σ. And the set of all left-sided infinite sequences over the alphabet Σ is denoted by ${}^{\omega}\Sigma$. Two-sided infinite sequence $...a_{-3}a_{-2}a_{-1}a_0.a_1a_2...$ over the alphabet Σ is defined as a map from $\mathbb{Z}$ to Σ. Similarly the set of all two-sided infinite sequences over the alphabet Σ is denoted by $\Sigma^{\mathbb{Z}}$. For a two-sided infinite sequence $\mathbf{w} = \ldots a_{-3}a_{-2}a_{-1}a_0.a_1a_2\ldots$ we define

$$L(\mathbf{w}) = \ldots a_{-3}a_{-2}a_{-1}a_0,$$

$$R(\mathbf{w}) = a_1a_2a_3\ldots$$

as left-sided and right-sided infinite sequences, correspondingly.

A non-empty finite word x is considered to be a right-infinite word denoted as x^{ω}, and it is referred to as a purely periodic word. An infinite word $\mathbf{w} = xy^{\omega}$, where y is a non-empty word, is known as an ultimately periodic word. If $\mathbf{w}$ is ultimately periodic, it can be expressed in the form xy^{ω}, where x and y are finite words with y not being the empty word. Then, x is referred to as a preperiod of $\mathbf{w}$ [1].

2.1.2 Finite Automaton

A finite automaton is a computation model that is a restricted version of more universal computation model called Turing machines. Since, the reason that we're considering finite automaton is for defining automatic sequences, we are containing this chapter to only discussing the DFAO model. For some input word (or sequence)—whose representation is related to the finite automaton—that is read with each state transition and if finite automaton reaches the terminal state and there are no more symbols of the input sequence to be read then the input word (or sequence) is said to have been accepted by the deterministic finite automaton (DFA). Unlike its non-deterministic version—where the transition function that depends on the current state and current symbol of the input takes values on a set of states—DFA's transition function that similary depends only on the current state and the current symbol of the input determines the only next state that it can move on. The set of words that are accepted by DFA is called the language of automaton. A graph representation of a DFA is given on the Figure 2.1, where the arrow with no label pointing to the cicrcle indicates the initial state, each arrow with a label(symbol) indicates the next state given the current state and if that symbols is read. A special accepting state is marked with double circles. Each state as represented by circles is named, conventionally, by the symbols $q_i, i \geq 0$. Formally we proceed with the definitions of DFA model and it's

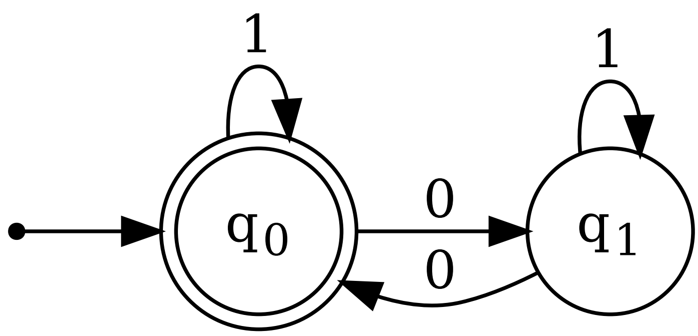

Figure 2.1: Accepts the words w, in the base 2 representation, that contain even number of 0's.

version with output DFAO.

Definition 2.1: A DFA is defined by a 5-tuple denoted as:

$$M = (Q, \Sigma, \delta, q_0, F) \tag{2.1}$$

where,

Q is a finite set of states,

Σ is a finite set of symbols (alphabet),

δ is a transition function defined as: $\delta : Q \times \Sigma \rightarrow Q$,

q_0 is an initial state, where $q_0 \in Q$

F is a set of final states, where $F \subseteq Q$.

For $w = w_0 ... w_n$, with $w \in \Sigma^*$, starting with $\delta(q_0, w_0)$ then the transition function $\delta(\delta(q_i), w_j)$, where $0 \leq j \leq n$, is applied for each symbol and where q_i is one of the finite states from the previous iteration of transition function. In the case where the word w is the empty word ε, the transition function $\delta(q, w)$ returns the current state q for any state q in the set of states Q. Then the language $L(M)$ accepted by a given DFA M is,

$$L(M) = \{w \in \Sigma^* : \delta(q_0, w) \in F\}.$$

Having these definitions of DFA and it's corresponding language the following properties are derived.

Theorem 2.1: For a given M, with m states, that generates $L = L(M)$, the language $\overline{L} = \Sigma^* \backslash L$ can be generated by some DFA with exactly the same number states.

Proof. Given that L is generated by $M = (Q, \Sigma, \delta, q_0, F)$, we can define another DFA $M' = (Q, \Sigma, \delta, q_0, Q \setminus F)$. It is evident that the language $\overline{L}$, which is the complement of L, can be generated by M'. Therefore, M' generates $\overline{L}$.

□

While it has already been shown, on the previous section on languages, that regular expressions are a way to specify languages and we also have a way of generating languages using DFA, the Kleene's theorem [1], tells that the only languages that DFA accepts are the ones that are specified with regular expressions. The following lemma is a commonly used tool to show a language is non-regular.

Lemma 2.2 (Pumping Lemma): Let $L \subseteq \Sigma^*$ be a regular language. Then, there exists an integer $n \geq 1$, depending on DFA of L, such that $\forall w \in L$ with $|w| \geq n$ it can be decomposed on $w = xyz$, where $x, y, z \in \Sigma^*$ and $|xy| \leq n, |y| \geq 1$, such that the words $xy^iz \in L, \forall i \geq 0$.

Proof. Let L be regular language as given and a corresponding DFA with designated number of states n. We take a word $w \in L$ such that $|w| \geq n$. Then it must be the case that for some $i \geq 1$ and $j > i$, the transition function $\delta(q_0, w[0...i]) = \delta(q_0, w[0...j])$. Therfore we split $w = xyz$ where $x = w[0...i], y = w[j, j+c]$, for some $c \geq 1$, and z is the rest of the sequence of the word w starting at the $j+c+1^{th}$ position. □

Importantly, following we have the definition of DFAO model, which is a DFA model extended with an additional output function τ.

Definition 2.2: We define a DFAO M as a 6-tuple:

$$M = (Q, \Sigma, \delta, q_0, \Delta, \tau). \tag{2.2}$$

Here Q, Σ, δ, q_0 are as defined on Definition 2.1. With the Δ being finite set of output symbols. And essentially, $\tau : \Sigma^* \to \Delta$, being the output function. In this case M defines the $M_k : \Sigma^* \to \Delta$, which is called finite-state function also denoted as $f_M(w) = \tau(\delta(q_0, w))$.

Similar to DFA, the DFAO is represented on the Figure 2.2, where states are lebeled by the q/a indicating the output $a = \tau(q)$ for the current state q.

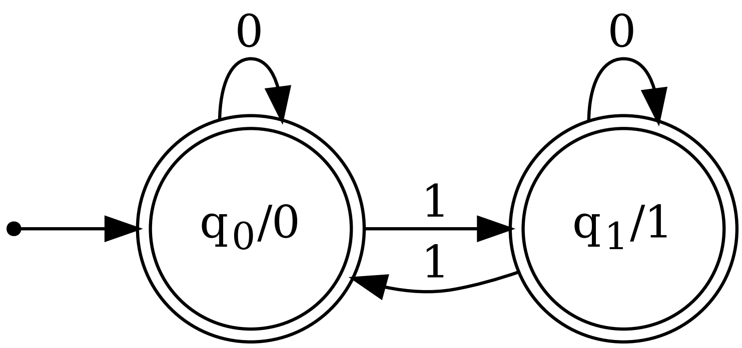

Figure 2.2: Sequence of the $mod(2)$ sum of the digits for the non-negative integers w, in the base 2 representation.

It would be expected that the DFAO is very related, in terms of languages that it generates, with DFA and the next theorem stated below makes that relation explicit [1].

Theorem 2.2: For any DFAO $M = (Q, \Sigma, \delta, q_0, \Delta, \tau)$, where $Q, \Sigma, \delta, q_0, \Delta, \tau$ are given then for every $d \in \Delta$ the set

$$I_d(M) = \{w \in \Sigma^* : \tau(\delta(q_0, w)) = d\}$$

forms a regular language.

2.2 Automatic Sequences

As stated at the start of the chapter, automatic sequences is the focal point of this work that is achieved by DFAO and other alternative means explored on the upcoming sections. In a non-formal way a sequence $(a_n)_{n \geq 0}$ where each of its terms a_n are values of the output function τ, corresponding to some *DFAO* M, that is applied on the final state for the reading of the input w is said to be automatic sequence [1]. Since the sequences $(a_n)_{n\geq 0}$ are defined for $n \in \mathbb{N}$ but the input alphabet Σ of *DFAO* M can be of any order $k \geq 2$ we explicitly note it by $k - DFAO$ M to indicate the input bases of *DFAO* M. That is, for the sequence $(a_n)_{n\geq 0}$ each $n \in \mathbb{N}$ is feed to *DFAO* M in the base k representation.

Definition 2.3: k-Automatic Sequences. An infinite sequence $(a_n)_{n\geq 0}$ over the finite alphabet Δ is said to be k-automatic if there exists a corresponding k-DFAO M $= (Q, \Sigma_k, \delta, q_0, \Delta, \tau)$ such that for every $n \in \mathbb{N}$, we have $a_n = \tau(\delta(q_0, w))$, where $w = [n]_k$.

The definition can be further clarified by considering the example of the Thue-Morse sequence.

Example 2.5: Thue-Morse Sequence. As can be observed from the Figure 2.2 the sequence $\mathbf{t} = (t_n)_{n\geq 0}$ is 2-automatic. Some of the first few terms are presented below:

$n =$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...
$t_n =$	0	1	1	0	1	0	0	1	1	0	0	0	1	1	0	1	0	...

Using the definition above we can observe that for some input that is read to $k - DFAO$ M it may be possible to have another similar input i.e. by having more leading zeros, and yet the $k - DFAO$ M would produce a different output in each case, which leads to inconsistency. For instance the number $[2]_{10}$ is exactly the same as $[02]_{10}$ but by the definitions they can produce different outputs. The following theorem eliminates the ambiguity [1].

Theorem 2.3: A sequence $(a_n)_{n\geq 0}$ is $k - automatic$ if and only if there exists a $k - DFAO$ M such that $a_n = \tau(\delta(q_0, [n]_k))$ for all $n \in \mathbb{N}$.

Proof. $\Longrightarrow$: Follows by the Definition 2.3

$\Longleftarrow$: Given a k-DFAO M = (Q, Σ_k, δ, q_0, Δ, τ), we construct the $k-DFAO\ M^{'}$ = $(Q', \Sigma_k, \delta^{'}, q_0^{'}, \Delta, \tau^{'})$ as follows:

$$Q' = Q \cup q_0',$$

$$\delta'(q,a) = \delta(q,a) \text{ for all } q \in Q \text{ and } a \in \Sigma_k,$$

$$\delta'(q_0',a) = \begin{cases} \delta(q_0,a) & \text{if } a \neq 0, \\ q_0' & \text{if } a = 0, \end{cases}$$

$$\tau'(q) = \tau(q) \text{ for all } q \in Q,$$

$$\tau'(q_0') = \tau(q_0).$$

It is evident that $\tau'(\delta'(q_0', 0^i[n]_k)) = \tau(\delta(q_0, [n]_k))$ for all $i, n \in \mathbb{N}$. □

Lemma 2.3: If L is a regular language then languages that are created be removing leading and trailing zeros of each $w \in L$ are regular.

Proof. Consider the set $C_k := \{\varepsilon\} \cup (\Sigma_k \setminus \{0\})\Sigma_k^*$ which correspond to some DFA M. Then the obtained regular language with leading zeros removed is $rlz(L) = (L^R/0^*)^R \cap C_k$, by the properties of regular languages under intersection, reversal, and quotient. Similarly the regular language with trailing zeros removed is $rtz(L) = (L/0^*) \cap (C_k)^R$.

□

The following is presented an alternative definition of automatic sequences, which will be further explored in the section on k-kernels. This definition usese the concept of k-fibers, denoted as $I_k(\mathbf{a}, d)$, for corresponding sequence $\mathbf{a} = (a_n)_{n \geq 0}$ which is defined as the set of all residue classes $[n]_k$ such that $a_n = d$, where d is an element of Δ and $k \geq 2$ [18].

Lemma 2.4: The sequence $\mathbf{a} = (a_n)_{n\geq 0}$ over Δ is k-automatic if and only if each k-fiber $I_k(\mathbf{a}, d)$ forms a regular language for every $d \in \Delta$.

Theorem 2.4: If a sequence $(b_n)_{n\geq 0}$ differs from a k-automatic sequence $(a_n)_{n\geq 0}$ only in a finite number of terms, then $(b_n)_{n\geq 0}$ is also a k-automatic sequence.

Proof. Let $M = (Q, \Sigma_k, \delta, q_0, \Delta, \tau)$ be a k-DFAO that generates the sequence $(a_n)_{n\geq 0}$. When the two sequences $(a_n)_{n\geq 0}$ and $(b_n)_{n\geq 0}$ differ for a finite set of input values n, the modification occurs in the functions τ, δ. Thus from M, we can define the DFAO $M' = (\{p_0, p_1, \ldots, p_n\} \cup Q, \Sigma_k, \delta', p_0, \Gamma, \tau')$ that generates the sequence $(b_n)_{n\geq 0}$. □

2.3 Morphisms

With some of the context already given on the previous sections the title of this section indicates the initiation of the journey to morphisms way of defining automatic sequences. The notion of homomorphisim is already familiar concept in the feld of analysis and topology, which we will shortly call it a morphism, with the only requirement that will be imposed is that for any map h from Σ^* to Δ^* and for all $x, y \in \Sigma^*$ the map h is linear with respect to the operator of concatenation i.e. $h(xy) = h(x)h(y)$. Based on the given condition, it is evident that once the function h is established for all individual elements x in Σ, it can be naturally extended to encompass the entire set Σ^*. Consequently, it follows that $h(\varepsilon) = \varepsilon$.

In the case where Σ is equivalent to Δ and for a given morphism h, we introduce the concept of an implicit morphism as follows. Define $h^0(a)$ as simply equal to a, and for any $a \in \Sigma$, then, for all $i \geq 1$ we recursively define $h^i(a) = h(h^{i-1}(a))$ [1]. The following example uses the notion of morphisms to define Thue-Morse sequence, which is already define using DFAO on the previous section.

Example 2.6: When Σ is equal to Δ, we define a mapping h such that $h(0) = 01$ and $h(1) = 10$. By repeatedly applying h, we find that $h^2(0) = 0110$ and $h^3(0) = 0110101$.

For any given morphism $h : \Sigma^* \to \Delta^*$ we have the following properties:

Width h is defined as the maximum length among all images of symbols in Σ under h: $Width(h) = \max_{a\in\Sigma} |h(a)|$. Depth h refers to the cardinality of the symbol set Σ: $Depth(h) = |\Sigma|$. Size h represents the sum of the lengths of all images of symbols in Σ under h: $Size(h) = \sum_{a\in\Sigma} |h(a)|$. We introduce the notion of uniform morphism which allows us to classify the morphisms in finer classes. A morphism $h : \Sigma^* \to \Delta^*$ is called a k-uniform morphism if, for some constant k, the length of the image of any symbol $a \in \Sigma$ under h is always equal to k, denoted as $|h(a)| = k$. A morphism $h : \Sigma^* \to \Delta^*$ is called expanding if the length of the image of every symbol $a \in \Sigma$ under h is greater than or equal to 2, i.e., $|h(a)| \geq 2$. h is non-erasing if $\nexists a \in \Sigma$ such that $h(a) = \varepsilon$. For $a \in \Sigma$ if there exists some integer $i \geq 1$ such that $h^i(a) = \varepsilon$ then we say that a is mortal. We associate the set of mortal letters M_h to each morphism h. We denote as $exp(h)$ the mortality exponent of h that is the least integer $i \geq 0$ such that $h^i(a) = \varepsilon$ for all $a \in M_h$. For a morphism $h : \Sigma^* \to \Sigma^*$, we have the following definitions: If there exists a sequence $a \in \Sigma^*$ such that $h(a) = a$, then we say that a is a fixed point of h. Let $a \in \Sigma$ be a sequence such that for some morphisim h we have that $h(a) = ax$ with $x \notin M_h^*$. In this case, we say that the morphisim h is prolongable on a. Now, we define an infinite sequence $\mathbf{w}$ as follows:

$$\mathbf{w} = h^{\omega}(a) := axh(x)h^2(x)\dots$$

Here, $h^{\omega}(a)$ represents the infinite concatenation of a, followed by x, followed by $h(x)$, followed by $h^2(x)$, and so on.

As shown on the Example 2.6, Thue-Morse sequence is defined using uniform morphisims. In the next lemma, which will be utilized in Cobham's theorem, we will establish a distinctive relationship between automatic sequences and k-uniform morphisms.

Lemma 2.5: Let $\mathbf{w} = h(w) = a_0a_1a_2...$ be an infinite sequence of some k-uniform morphism h. Then $h(a_i) = a_{ki}a_{ki+1}...a_{ki+k-1}$.

Proof. For a finite word $a_0a_1...a_i$, we have

$$h(a_0a_1...a_i) = a_0a_1...a_{ki+k-1}.$$

Then $h(a_0a_1...a_{i-1})h(a_i) = (a_0a_1...a_{ki-1})(a_{ki}a_{ki+1}...a_{ki+k-1})$. □

Theorem 2.5: For an integer $k \geq 2$, the sequence $\mathbf{a} = (a_n)_{n\geq 0}$ is k-automatic if and only if there exists a k-uniform morphism h and an element $a \in \Sigma$ such that $h^{\omega}(a)$ is a fixed point of h and $\mathbf{a}$ is the image of the coding function τ applied to $h^{\omega}(a)$.

Proof. $\Longleftarrow$: Suppose that $\mathbf{a} = \tau(\mathbf{w})$, where $\tau : \Delta \to \Delta'$ is the coding function and $\mathbf{w} = \varphi(\mathbf{w})$ for a k-uniform morphism $\varphi : \Delta^* \to \Delta^*$. Let $\mathbf{w} = w_0w_1\ldots$ where $\mathbf{w} \in \Delta^*$. We can construct a k-DFAO M as follows: Set the initial state q_0 of M to be w_0. For any state q in M and input $b \in \Sigma_k$, define the transition function $\delta(q,b)$ as the bth letter of $\varphi(q)$. By induction, we will show that for all $n \geq 0$ it is true that $w_n = \delta(q_0,[n]_k)$. First, the base case $n = 0$ is trivially true since $\delta(q_0,[0]_k) = \delta(q_0,\varepsilon) = q_0 = w_0$. Now, lets assume that the statement holds for all $i < n$ and prove it for n. Let $[n]_k = n_1n_2\ldots n_t$, where $0 \leq n_t < k$ and $n = kn' + n_t$. We can deduce the following:

$$\begin{aligned}
\delta(q_0, [n]_k) &= \delta(q_0, n_1 n_2 \dots n_t) \\
&= \delta(\delta(q_0, [n']_k), n_t) \\
&= \delta(w_{n'}, n_t) \quad \text{(by the induction hypothesis)} \\
&= \text{the } n_t\text{th symbol of } \varphi(w_{n'}) \quad \text{(by the definition of } \delta\text{)} \\
&= w_{kn'+n_t} \quad \text{(using the previous lemma)} \\
&= w_n.
\end{aligned}$$

Now, let τ be a coding function, and consider the sequence $\mathbf{a} = (a_n)_{n\geq 0}$. We can express a_n as follows:

$$a_n = \tau(w_n) = \tau(\delta(q_0, [n]_k)).$$

$\Longrightarrow$: Assuming that $(a_n)_{n\geq 0}$ is a k-automatic sequence generated by some k-DFAO M, we want to show that it corresponds to the image under τ of a fixed point $\mathbf{w} = w_0 w_1 w_2 \dots$ of the morphism φ.

We define the morphism φ for each $q \in Q$ as:

$$\varphi(q) := \delta(q,0)\delta(q,1)\dots\delta(q,k-1).$$

Let $\mathbf{w}$ be a fixed point of the morphism φ, starting with q_0. Then, we will prove by induction that $\delta(q_0, y) = w_{[y]_k}$ for all $y \in \Sigma^*$. For the base case where $|y| = 0$, we have $\delta(q_0, \varepsilon) = q_0 = w_0$, which satisfies the condition. Now, assuming that $\delta(q_0, y) = w_{[y]_k}$ holds for all $|y| < i$, we will show it holds true for $|y| = i$. Let $y = xa$ where $a \in \Sigma_k$. We can deduce the following:

$$
\begin{aligned}
\delta(q_0,y) &= \delta(q_0,xa) \\
&= \delta(\delta(q_0,x),a) \quad \text{(using the definition of } \delta) \\
&= \delta(w_{[x]_k},a) \quad \text{(by the induction hypothesis)} \\
&= \varphi(w_{[x]_k})_a \quad \text{(using the definition of } \varphi) \\
&= w_{k[x]_k+a} \quad \text{(by the definition of } \varphi) \\
&= w_{[xa]k} \\
&= w_{[y]_k}.
\end{aligned}
$$

Consequently, we can conclude that $a_n = \tau(\delta(q_0,[n]_k)) = \tau(w_n)$, which means that $(a_n)_{n\geq 0}$ is the image under coding τ of a fixed point $\mathbf{w}$ of the morphism φ □

2.4 *k*-Kernel

Already on the section of automatic sequences we have introduced the notion of *k*-fibers, which will play a central role in this section. In this section the relation between automatic sequences and *k*-kernels will be established [14].

Definition 2.4: For an infinite sequence $\mathbf{a} = (a_n)_{n\geq 0}$, the *k*-kernel of $\mathbf{a}$ is defined to be the set of subsequences

$$K_k(\mathbf{a}) = \{(a_{k^i n+j})_{n\geq 0} : i \geq 0 \text{ and } 0 \leq j < k^i\}.$$

The following is an example *k*-kernels for the commonly used Thue-Morse sequence.

Example 2.7: Consider the Thue-Morse sequence $\mathbf{t} = (t_n)_{n\geq 0}$ as defined above. We can determine the 2-kernel of the sequence $\mathbf{t}$ as follows:

$$K_2(\mathbf{t}) = \{\mathbf{t}, \bar{\mathbf{t}}\}.$$

where $\bar{\mathbf{t}}$ represents the complement of $\mathbf{t}$. The elements of the 2-kernel satisfy the recursive relationships: $t_{2n} = t_n$ and $t_{2n+1} = (t_n + 1) \mod 2$.

The next theorem points to the relation between automatic sequences and k-kernels.

Theorem 2.6: Let $k \geq 2$. The sequence $\mathbf{a} = (a_n)_{n\geq 0}$ is k-automatic if and only if the set $K_k(\mathbf{u})$ is finite.

Proof. $\Longrightarrow$: Let $\mathbf{a} = (a_n)_{n\geq 0}$ be the given sequence, and let $M = (Q, \Sigma_k, \delta, q_0, \Delta, \tau)$ be the corresponding k-DFAO. We know that for any $t \geq 0$:

$$a_n = \tau(\delta(q_0, (n)_k^R 0^t)), \quad \text{for all } n \geq 0.$$

Now, consider a word $w = w_0 w_1 \dots w_i$, where $|w| = i$. Let $q = \delta(q_0, w^R)$, and let $[w]_k = \sum_{1\leq i\leq r} w_i k^{r-i} = j$. We can observe that $(k^i n + j)_k = (n)_k w$ for all $n \geq 0$. In particular, when $n = 0$, we have $(k^i n + j)_k = (j)_k$. Furthermore, $w = 0^t (j)_k$ for some $t \geq 0$. We can now proceed as follows:

$$\delta(q_0, (k^i n + j)_k^R) = \delta(q_0, (j)_k^R) \quad = \delta(q_0, (j)_k^R 0^t) \ = \delta(q_0, w^R) \quad = q \ = \delta(q, (0)_k^R).$$

Thus, we have shown that the sequence $(a_{k^i n+j})_{n\geq 0}$ is generated by the k-DFAO M. Since there are only a finite number of states q, it follows that $K_k(\mathbf{a})$ is a finite set.

$\Longleftarrow$: Let us assume that $K_k(\mathbf{a})$ is a finite set. In this case, we can partition Σ_k^* into a finite number of disjoint equivalence classes. Where two words w and x in Σ_k^* are equivalent, denoted as $w \equiv x$, if and only if $a_{k^{|w|}n+[w]k} = a_{k^{|x|}n+[x]_k}$ for all $n \geq 0$ [1]. To construct a corresponding k-DFAO based on this equivalence relation, we define: The set of states Q as $Q = \{[x] : x \in \Sigma_k^*\}$, where $[x]$ represents the equivalence class containing x. The transition function δ as $\delta([x], b) = [xb]$ for all $[x] \in Q$ and $b \in \Sigma_k$. The output function τ as $\tau([w]) = a_{[w]_k}$ for all $[w] \in Q$. The initial state q_0 as $[\varepsilon]$, which is the equivalence class containing the empty word. Using the equivalence relation defined above, we can observe that for any $[x],[w] \in Q$ and $b \in \Sigma_k$, we have $\delta([x],b) = \delta([w],b)$ and $\tau(\delta([x],b)) = \tau(\delta([w],b))$. □

2.5 Formal Power Series

We begin by presenting the foundational concepts of algebra, including groups, rings, and fields. Then we continue the section with the introduction of the notion of formal Laurent series and its subset, formal power series. We examine the properties of the elements within the ring of formal Laurent series [8]. Crucially, we investigate how a formal Laurent series can be characterized as either algebraic or transcendental, drawing insights from their connection to k-automatic sequences. In conclusion, we highlight Christol's theorem, which asserts that a formal Laurent series is algebraic if and only if there exists a relationship between its coefficients and a k-automatic sequence [6, 8, 16, 17].

2.5.1 Group

A group is a fundamental algebraic structure that captures the essence of symmetry and transformation [12].

Definition 2.5: A group G is a set equipped with a binary operation that combines any two elements $a, b \in G$ to produce another element in G. For G to be considered a group, it must satisfy following axioms:

Associativity: that is, $(ab)c = a(bc), \ \forall a,b,c \in G$.

Identity: there is an element (called the identity) $e \in G$ such that $ae = ea = a, \ \forall a \in G$.

Inverses: for each element $a \in G$, there is an element $a^{-1} \in G$, called an inverse of a, such that

$aa^{-1} = a^{-1}a = e$.

Example 2.8: The group of integers under addition: The set of all integers, denoted by $\mathbb{Z}$, forms a group under the operation of addition. The closure property is satisfied since adding any two integers results in another integer. The identity element is 0, and every integer has an inverse (the negative of that integer) that when added yields the identity element.

Example 2.9: The group of 2×2 invertible matrices over $\mathbb{C}$: The set of all 2×2 matrices with non-zero determinants forms a group under matrix multiplication. The identity element is the 2×2 identity matrix, and every matrix in this set has an inverse (its inverse matrix) that when multiplied yields the identity matrix. That is,

$$\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \times \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} = \begin{bmatrix} a_1a_2 + b_1c_2 & a_1b_2 + b_1d_2 \\ c_1a_2 + d_1c_2 & c_1b_2 + d_1d_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Example 2.10: The set $\mathbb{R}^n$, consisting of all n-tuples $(a_1, a_2, ..., a_n)$ where each component belongs to $\mathbb{R}$, forms a group under componentwise addition. In other words, the operation of adding two n-tuples $(a_1, a_2, ..., a_n)$ and $(b_1, b_2, ..., b_n)$ is defined as $(a_1 + b_1, a_2 + b_2, ..., a_n + b_n)$. The closure property is satisfied since the sum of any two n-tuples in $\mathbb{R}^n$ is still an n-tuple in $\mathbb{R}^n$. The identity element is the n-tuple $(0, 0, ..., 0)$, where each component is zero. Finally, for any n-tuple $(a_1, a_2, ..., a_n)$, its inverse is the n-tuple $(-a_1, -a_2, ..., -a_n)$, as the sum of an n-tuple with its componentwise additive inverse yields the identity element. Thus, $\mathbb{R}^n$ under componentwise addition forms a group.

The order of a group, whether it is finite or infinite, refers to the number of elements it contains. It is denoted by $|G|$, where G represents the group. The order represents the cardinality or size of the group, indicating the total count of distinct elements within it. For a finite group, the order is a positive integer, while for an infinite group, the order is considered to be infinite. In a group G, the order of an element g refers to the smallest positive integer n such that raising g to the power of n results in the identity element e of the group. This concept is denoted as $g^n = e$. If there is no such positive integer n that satisfies this condition, we say that g has infinite order.

Example 2.11: Consider the group $U(15) = \{1,2,4,7,8,11,13,14\}$. This group has an order of 8, meaning it contains 8 elements. To determine the order of a specific element, such as 13, we compute the sequence of powers: $13^1 \equiv 13 \pmod{15}$, $13^2 \equiv 4 \pmod{15}$, $13^3 \equiv 7 \pmod{15}$, and $13^4 \equiv 1 \pmod{15}$. Thus, the order of the element 13, denoted as $|13|$, is 4.

2.5.2 Ring

Certain sets possess inherent properties that make them suitable for two fundamental binary operations: addition and multiplication. Well-known examples include the integers, integers modulo n, real numbers, matrices, and polynomials. When these sets are treated as groups, the primary focus is typically on addition, with multiplication often overlooked. However, there are numerous scenarios where it becomes essential to consider both addition and multiplication simultaneously. To address this need, the abstract concept of a ring arises as a powerful mathematical tool.

Definition 2.6: A ring R is defined as a set equipped with two binary operations: addition (denoted by a + b) and multiplication (denoted by ab). These operations satisfy several properties for all elements a, b, and c belonging to the ring R. These properties include:

1. Commutativity of addition: $a+b=b+a$.

2. Associativity of addition: $(a+b)+c=a+(b+c)$.

3. Additive identity: there is an element $0 \in R$ such that $a+0=a$ for all $a \in R$.

4. Additive inverse: For every element $a \in R$, there exists an element $(-a)$ in R such that $a+(-a)=0$.

5. Distributive law: $a(b+c)=ab+ac$, and $(b+c)a=ba+ca$.

Example 2.12: Consider the set $\mathbb{Z}_n=\{0,1,\ldots,n-1\}$ equipped with the operations of addition and multiplication modulo n. It can be observed that $\mathbb{Z}_n$ forms a ring with unity 1, where the unity element is the residue class 1 modulo n. Furthermore, within $\mathbb{Z}_n$, there exists a subset of elements called units, denoted by $U(n)$, which consists of the elements that possess multiplicative inverses.

Example 2.13: Consider the set $\mathbb{Z}[x]$ consisting of all polynomials in the variable x with integer coefficients. With the operations of ordinary addition and multiplication, $\mathbb{Z}[x]$ forms a commutative ring with unity, where the unity element is the polynomial $f(x)=1$.

2.5.3 Field

By extending the concept of a ring, we introduce a structure known as a field. An additional property is introduced that allows for the existence of multiplicative inverses for nonzero elements in the field.

Definition 2.7: A field F is defined as a set equipped with two binary operations: addition (denoted by a + b) and multiplication (denoted by ab). These operations satisfy several properties for all elements a, b, and c belonging to the field F. These properties include:

1. Commutativity of addition and multiplication: $a+b=b+a,$ and $a \cdot b = b \cdot a$
2. Associativity of addition and multiplication: $(a+b)+c = a+(b+c)$ and $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
3. Additive and multiplicative identity: there is an element $0,1 \in F$ such that $a+0=a,\ a \cdot 1 = a$ for all $a \in F$.
4. Additive inverse: For every element $a \in R$, there exists an element $(-a)$ in R such that $a+(-a)=0$.
5. Multiplicative inverse: For every element $a \neq 0 \in F$, there exists an element a^{-1} in F such that $a \cdot a^{-1} = 1$.
6. Distributive law: $a \cdot (b+c) = a \cdot b + a \cdot c$.

Example 2.14: Consider the finite field $GF(7) = \{0,1,2,3,4,5,6\}$ where addition and multiplication are performend through modular arithmetic with respect to prime number 7. For the number 2 the multiplicative inverse is: $2 \cdot 4 \equiv 1 \pmod 7$.

2.5.4 Formal Power Series and Christol's Theorem

To start, initialy the concept of ring of polynomials over the commutative ring R is defined.

Definition 2.8: Let R be a commutative ring, then the set

$$R[X] = \{a_n X^n + a_{n-1} X^{n-1} + \cdots + a_1 X + a_0 \mid a_i \in R \text{ and } n \geq 0\}$$

is the ring of polynomials over R with X being the indeterminate, and the non-zero term $a_i X^i$ with highest exponent i determines the degree of a specific polynomial.

The addition and multiplication are naturally defined for the elements of $R[X]$ [15]. For $p, q \in R[X]$ which are given as $p(X) = a_nX^n + a_{n-1}X^{n-1} + \cdots + a_1X + a_0$ and $q(X) = b_nX^n + b_{n-1}X^{n-1} + \cdots + b_1X + b_0$, addition is: $(p+q)(X) = (a_n + b_n)X^n + (a_{n-1} + b_{n-1})X^{n-1} \cdots + (a_1 + b_1)X + (a_0 + b_0)$. And multiplication is: $(p \times q)(X) = a_0b_0 + (a_0b_0 + a_1b_0)X + (a_0b_2 + a_1b_1 + a_2b_0)X^2 + \ldots$

For F being a field, then the field of fractions of $F[X]$ consists of elements f/g where $f, g \in F[X]$ and $g \neq 0$. This field of fractions over F is denoted by $F(X)$. Any other property of the algebraic structures, whenever used in some other theorems, is assumed and not proven here.

Definition 2.9: For a ring R, the set

$$R[[X]] = \{a_0 + a_1X + a_2X^2 + \ldots \mid a_i \in R \text{ and } i \geq 0\}$$

forms a ring, where each element represents a formal power series.

In the context of formal power series, the concept of convergence is not applicable or relevant [11]. Addition and multiplication is performed by treating the series similarly to polynomials. For $p, q \in R[[X]]$ where $p(X) = \Sigma_{i\in\mathbb{N}}^{\infty} a_iX^i$ and $q(X) = \Sigma_{i\in\mathbb{N}}^{\infty} b_iX^i$ we have, $(p+q)(X) = \Sigma_{i\in\mathbb{N}}^{\infty}(a_i + b_i)X^i$ and $(p \times q)(X) = \Sigma_{n\in\mathbb{N}}^{\infty}(\Sigma_{k=0}^{n} a_kb_{n-k})X^n$. Following we let F be a field and have the definition of formal Laurent series.

Definition 2.10: Let F be a field, the set

$$F((X)) = \{\Sigma_{i\geq N}^{\infty} a_iX^i \mid a_i \in F \text{ and } N \in \mathbb{Z}\}$$

is defined as the field of formal Laurent series.

Similar to the fields $\mathbb{C}$, or $\mathbb{R}$ that we define algebraicity of elements of thereof, the definition of algebraicity for formal Laurent series follows

Definition 2.11: Let $p \in R((X))$ with coefficients over $R(X)$. Then p is said to be algebraic over $R(X)$ if there exists elements $f_0(X), f_1(X) \dots f_n(X) \in R(X)$ not all zero, such that $f_0 + f_1 p^1 + \dots + f_n p^n = 0$

The following we have examples of algebraic formal power series.

Example 2.15: Let f be the formal power series over $\mathbb{F}_2$ given by

$$f(X) = X + X^2 + X^4 + \ldots = \sum_{i \geq 0} X^{2^i}.$$

We can observe that this series is algebraic since,

$$f(X^2) = f(X) - X,$$

therefore,

$$f(X)^2 + f(X) + X = 0.$$

Example 2.16: Let $T(X) = \sum_{n \geq 0} t_n X^n$, where $(t_n)_{n \geq 0}$ represents the sequence already defined on the Example 2.5. Now, lets analyze the expression of $T(X)$:

$$\begin{aligned} T(X) &= \sum_{n \geq 0} t_n X^n \\ &= \sum_{n \geq 0} t_{2n} X^{2n} + \sum_{n \geq 0} t_{2n+1} X^{2n+1} \\ &= \sum_{n \geq 0} t_n X^{2n} + X \sum_{n \geq 0} (t_n + 1) X^{2n} \\ &= T(X^2) + XT(X^2) + X \frac{1}{1 - X^2}. \end{aligned}$$

From this expression, we can deduce, over the field $\mathbb{F}_2$, that the series is algebraic.

Example 2.17: Let $F(X)$ over the non-negative integers $\mathbb{N}$ and be defined by

$$F(X) = 1 + X + 2X^2 + 3X^3 + 5X^4 + \dots$$

It can be seen that the coefficients of the series are fibonaci series, then we have

$$\begin{aligned}
F(X) &= \sum_{n\geq 0} F_n X^n \\
&= F_0 + F_1 + \sum_{n\geq 2} F_n X^n \\
&= 1 + X + \sum_{n\geq 2} (F_{n-1} + F_{n-2}) X^n \\
&= 1 + X + X \sum_{n\geq 2} F_{n-1} X^{n-1} + X^2 \sum_{n\geq 2} F_{n-2} X^{n-2} \\
&= 1 + X + X(F(X) - 1) + X^2 F(X) \\
&= 1 + XF(X) + X^2 F(X) \\
&= \frac{1}{1 - X - X^2}
\end{aligned}$$

We arrange the equality to get

$$(1 - X - X^2)F(X) - 1 = 0.$$

Hence the series $F(X)$ is algebraic over $\mathbb{N}$.

The former two examples are instances of formal power series defined over finite fields, for which the Christol's theorem is used to show the algebraicity of it in terms of its relation with p-automatic sequences.

We represent formal Laurent series $\sum_{n\geq N} a_n X^n$ by the equivalance class

$$\left[\sum_{n\geq 0} a_{k+n} X^n, k \right]$$

The equivalence relation is established by shifting the terms of a formal Laurent series [4]. Specifically, for any given formal Laurent series S, $j, k \in \mathbb{N}$, and $n \in \mathbb{Z}$, the equivalence relation is defined as:

$$[X^j S, n-j] = [X^k S, n-k],$$

where S belongs to the ring of formal Laurent series $F[[X]]$.

Definition 2.12: Operations on equivalance classes of formal Laurent Series are defined as follows:

For addition:

$$[S,n] + [T,n] = [S+T,n].$$

For multiplication:

$$[S,n] + [T,m] = [ST, n+m].$$

Definition 2.13: For $0 \leq i < q$ we define a linear transformation

$$\Gamma_{i,q} : \mathbb{F}_q[[X]] \to \mathbb{F}_q[[X]]$$

with

$$\Gamma_{i,q}\left(\sum_{n\geq 0} a_n X^n\right) = \sum_{n\geq 0} a_{qn+i} X^n.$$

Lemma 2.6: For a formal power series $F \in \mathbb{F}_q[[X]]$ with $i, q \in \mathbb{N}$, $q \geq 1$ and $0 \leq i < q$, the following property holds

$$F(X) = \sum_{n\geq 0} a_n X^n = \sum_{0\leq i<q} X^i \Gamma_{i,q}(F(X))^q.$$

Proof. For a given $F \in \mathbb{F}_q[[X]]$ we have

$$
\begin{aligned}
F(X) &= \sum_{n\geq 0} a_n X^n \\
&= \sum_{0\leq i<q} \sum_{n\geq 0} a_{qn+i} X^{qn+i} \\
&= \sum_{0\leq i<q} X^i \sum_{n\geq 0} a_{qn+i} X^{qn} \\
&= \sum_{0\leq i<q} X^i \left(\sum_{n\geq 0} a_{qn+i} X^n \right)^q \\
&= \sum_{0\leq i<q} X^i \Gamma_{i,q}(F(X))^q.
\end{aligned}
$$

□

Lemma 2.7: For formal power series $F, G \in \mathbb{F}_q[[X]]$ with $i, q \in \mathbb{N}$, $q \geq 1$ and $0 \leq i < q$, the following property holds

$$
\Gamma_{i,q}(F^q \cdot G) = F \cdot \Gamma_{i,q}(G).
$$

Proof. For F, G as give, we have

$$
\begin{aligned}
\Gamma_{i,q}(F^q \cdot G) &= \Gamma_{i,q}\left(\left(\sum_{k\geq 0} a_k X^k \right)^q \left(\sum_{j\geq 0} b_j X^j \right) \right) \\
&= \Gamma_{i,q}\left(\left(\sum_{k\geq 0} a_k X^{qk} \right) \left(\sum_{j\geq 0} b_j X^j \right) \right) \\
&= \Gamma_{i,q}\left(\sum_{n\geq 0} X^n \left(\sum_{\substack{k,j\geq 0,\\ qk+j=n}} a_k b_j \right) \right) \\
&= \sum_{n\geq 0} X^n \left(\sum_{\substack{k,j\geq 0,\\ qk+j=qn+i}} a_k b_j \right) \\
&= \sum_{n\geq 0} X^n \left(\sum_{0\leq k\leq n} a_k b_{q(n-k)+i} \right) \\
&= \sum_{k\geq 0} a_k X^k \left(\sum_{n\geq k} b_{q(n-k)+i} X^{n-k} \right) \\
&= \sum_{k\geq 0} a_k X^k \left(\sum_{n\geq 0} b_{qn+i} X^n \right) \\
&= F \cdot \Gamma_{i,q}(G).
\end{aligned}
$$

□

We extend the definition of the $\Gamma_{i,q}(S)$ operator to $\mathbb{B}_{i,q}([S,aq]) = [\Gamma_{i,q}(S),a]$ which is defined over the equivalance classes of formal Larent series.

Then, for $A,B \in F((X))$, all $q,i \in \mathbb{N}$ and $0 \leq i < q$ similarly to Lemma 2.7 we have

$$\mathbb{B}_{i,q}(A \cdot B^q) = \mathbb{B}_{i,q}(A) \cdot B \tag{2.3}$$

Lemma 2.8: Let $F \in F_q[[X]]$ be formal power series, where $q = p^n$. F is algebraic over $F_q(X)$ if and only if there exists non-zero polynomials $A_0(X),\ldots,A_t(X)$, such that

$$A_0F + A_1F^q + \cdots + A_tA^{q^t} = 0.$$

Lemma 2.9: The sequence $\mathbf{a} = (a_n)_{n\geq 0}$ defined over the finite field $\mathbb{F}_q$ is q-automatic if and only if there exists a finite set of formal power series $\mathfrak{F}$ satisfying the following conditions: For every formal power series $F(X) = \sum_{n\geq 0} a_nX^n$ associated with $\mathbf{a}$, we have $F \in \mathfrak{F}$. For every $G \in \mathfrak{F}$ and every $0 \leq i < q$, $\Gamma_{i,q}(G)$ belongs to $\mathfrak{F}$.

Proof. Consider the set $\mathbb{K}_q(\mathbf{a}) = \{\mathbf{a}^1,\mathbf{a}^2,\ldots,\mathbf{a}^i\}$, where $\mathbf{a}^j$ represents the j-th element of the q-kernel of the sequence $\mathbf{a}$.

$\Longrightarrow$: We construct the set for $\mathfrak{F}$ as follows

$$\mathfrak{F} = \left\{ \sum_{n\geq 0} a_n^i X^n : 1 \leq i \leq r \right\},$$

and we see that if we let $\mathbf{a} = \mathbf{a}^1$, then $F \in \mathfrak{F}$.

$\Longleftarrow$: By observing that $\sum_{n\geq 0} a_n^i X^n \in \mathfrak{F}$, we can deduce that the cardinality of $\mathbb{K}_q(\mathbf{a})$ is less than or equal to the cardinality of $\mathfrak{F}$. Therefore, we can conclude that the q-kernel is a finite set. □

Following is the Christol's theorem, which as stated at the beginning, establishes a relation between algebraicity of elements of $F((X))$ and a corresponding p-automatic sequence.

Theorem 2.7: Let $\mathbf{a} = (a_i)_{i\geq 0}$ be a sequence over Δ, with Δ being a non-empty finite set. Let $q = p^n$ for some prime $p \in \mathbb{P}$ and $n \in \mathbb{N}$. Then the sequence $\mathbf{a}$ is a p-automatic if and only if there exists an injective map $\beta : \Delta \to \mathbb{F}_q$ and $n \geq 1$ such that the formal power series $\sum_{i\geq 0} \beta(a_i)X^i$ is algebraic over the $\mathbb{F}_q(X)$.

Proof. $\Longrightarrow$: To make β an injective map we pick n such that $|\Delta| \leq p^n$. By the [3][p.109, proposition 3.5] we know that the sequence $\mathbf{a} = (a_n)_{n\geq 0}$ is also q-automatic. Then by Theorem 2.6 we let the $\mathbb{K}_q(\mathbf{a}) = \{\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^d\}$, where $\mathbf{a} = \mathbf{a}^1$. Define the formal power series as follows

$$F_j(X) = \sum_{n\geq 0} a_n^j X^n, \text{ for } 1 \leq j \leq d.$$

Then,

$$F_j(X) = \sum_{0\leq r\leq (q-1)} \sum_{0\leq m} a_{qm+r}^j X^{qm+r} \tag{2.4}$$

$$= \sum_{0\leq r\leq (q-1)} X^r \sum_{0\leq m} a_{qm+r}^j X^{qm}. \tag{2.5}$$

By 2.4 we see that the $F_j(X)$ is on a span of vector space generated by the bases vectors $F_1(X^q), F_2(X^q), \ldots F_d(X^q)$. But similarly, we have that $F_i(X^q)$ is linear combination of the bases vectors $F_1(X^{q^2}), F_2(X^{q^2}), \ldots F_d(X^{q^2})$. Applying the same reasoning, we can conclude that for any j in the range $1 \leq j \leq d$ and any k in the range $0 \leq k \leq d$, the power series $F_j(X^{q^k})$ can be expressed as linear combinations of the base vectors $F_1(X^{q^{d+1}}), F_2(X^{q^{d+1}}), \ldots, F_d(X^{q^{d+1}})$. However, the dimension of the vector space spanned by the base vectors $F_1(X^{q^{d+1}}), F_2(X^{q^{d+1}}), \ldots, F_d(X^{q^{d+1}})$ over the field $\mathbb{F}_q(X)$ is at most d. Consequently, the power series $F_j(X), F_j(X^q), \ldots, F_j(X^{q^d})$ are linearly dependent over $\mathbb{F}_q(X)$. Specifically, when $j = 1$, we can deduce that $F_1(X)$ is algebraic over $\mathbb{F}_q(X)$.

$\Longleftarrow$: Let $F(X) = \sum_{i \geq 0} a_i X^i$ be algebraic over $\mathbb{F}_q(X)$. Then by Lemma 2.8 we know that there exists polynomials $A_0(X), \ldots, A_t(X)$ with $A_0(X) \neq 0$ such that

$$\sum_{0 \leq i \leq t} A_i(X) F(X)^{q^i} = 0.$$

Let $G = \frac{F(X)}{A_0(X)}$, with $G = \sum_{1 \leq i \leq t} C_i G^{q^i}$, where $C_i = -A_i A_0^{q^i - 2}$. For $N = \max(\deg A_0, \max_i \deg C_i)$, define

$$\mathfrak{F} = \left\{ H \in \mathbb{F}_q((X)) : H = \sum_{0 \leq i \leq t} D_i G^{q^i}, D_i \in \mathbb{F}_q[X],\ \deg D_i \leq N \right\}.$$

Let $H \in \mathfrak{F}$, then

$$\begin{aligned} \mathbb{B}_{i,q}(H) &= \mathbb{B}_{i,q}\left(D_0 G + \sum_{1 \leq i \leq t} D_i G^{q^i} \right) \\ &= \mathbb{B}_{i,q}\left(\sum_{1 \leq i \leq t} (D_0 C_i + D_i) G^{q^i} \right) \\ &= \sum_{1 \leq i \leq t} \mathbb{B}_{i,q}(D_0 C_i + D_i) G^{q^i}. \end{aligned}$$

Here the $\deg D_0$, $\deg D_i$, $\deg C_i \leq N$, then $\deg(D_0 C_i + D_i) \leq 2N$ therefore

$$\deg(\mathbb{B}_{i,q}(D_0 C_i + D_i)) \leq \frac{2N}{q} \leq N.$$

By the Lemma 2.9 we have that the sequence **a** is q-automatic. □

REFERENCES

[1] Jean-Paul Allouche, Jeffrey Shallit (2003) *Automatic Sequences Theory, Applications, Generalizations*, Cambridge University Press.

[2] Géza Horváth, Benedek Nagy (2014) *Formal Languages and Automata Theory*, Typotex.

[3] Samuel Eilenberg (1974) *Automata, languages, and machines. Volume A*, Academic Press.

[4] Joost Winter (2015) *Erratum to various proofs of Christol's theorem*, Citeseer.

[5] David S. Dummit, Richard M. Foote (2004) *Abstract Algebra*, John Wiley and Sons, Inc.

[6] N. Pytheas Fogg (2002) *Automatic sequences and transcendence, Lecture Notes in Mathematics, 2002* Springer Berlin Heidelberg.

[7] Hu, Yining (2018) *Transcendence of* $L(1,\chi s)/\Pi$ *and automata. Journal of Number Theory, 187, 215-232.*

[8] Xiao-Xiong Gan, Dariusz Bugajewski (2011) *On formal Laurent series. Bull. Brazilian Math. Soc, 42(3), 415-437.*

[9] Adamczewski, B., & Bell, J. P. (2012). *On vanishing coefficients of algebraic power series over fields of positive characteristic.* Inventiones mathematicae, 187(2), 343-393.

[10] Adamczewski, B., & Cassaigne, J. (2006). *Diophantine properties of real numbers generated by finite automata.* Compositio Mathematica, 142(6), 1351-1372.

[11] Allouche, J. P., & Shallit, J. (1992). *The ring of k-regular sequences.* Theoretical Computer Science, 98(2), 163-197.

[12] Carstensen-Opitz, C., Fine, B., Moldenhauer, A., & Rosenberger, G. (2019). *Abstract Algebra. In Abstract Algebra.* De Gruyter.

[13] Allender, E., Arvind, V., & Mahajan, M. (2003). *Arithmetic complexity, Kleene closure, and formal power series.* Theory of Computing Systems, 36, 303-328

[14] Han, G., & Hu, Y. (2019). *On the automaticity of sequences defined by the continued fractions* arXiv preprint arXiv:1908.02384.

[15] Norman, C. (2012). *The Polynomial Ring F [x] and Matrices over F [x].* Finitely Generated Abelian Groups and Similarity of Matrices over a Field, 165-202.

[16] Bostan, A., & Raschel, K. (2021). *Transcendence in Algebra, Combinatorics, Geometry and Number Theory.* Springer International Publishing.

[17] Wu, W. (2020). *Stieltjes continued fractions related to the paperfolding sequence and Rudin-Shapiro sequence.* Advances in Applied Mathematics, 118, 102040.

[18] Meduna, A. (2012). *Automata and languages: theory and applications.* Springer Science & Business Media.

[19] Rigo, M. (2014). *Formal languages, automata and numeration systems 1: Introduction to combinatorics on words (Vol. 1).* John Wiley & Sons.