論文 / 著書情報
Article / Book Information

題目(和文)	正規言語の零壱則
Title(English)	Zero-One Law for Regular Languages
著者(和文)	新屋良磨
Author(English)	Ryoma Sin'ya
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第10103号, 授与年月日:2016年3月26日, 学位の種別:課程博士, 審査員:鹿島 亮,小島 定吉,南出 靖彦,渡辺 治,寺嶋 郁二,金沢 誠
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第10103号, Conferred date:2016/3/26, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

ZERO-ONE LAW FOR REGULAR LANGUAGES

正規言語の零壱則

Ryoma Sin'ya

Tokyo Institute of Technology,
Department of Mathematical and Computing Sciences.

This thesis is an exposition of the author's research on automata theory and zero-one laws which had been done in 2013–2015 at Tokyo Institute of Technology and Télécom ParisTech. Most of the results in the thesis have been already published in [58, 57].

PROLOGUE

The notion of the regularity of a set of words, or regular language, is originally introduced by Kleene in 1951 [29]. The celebrated Kleene's theorem states that the class of regular languages (that is, definable by regular expressions) coincides with the class of recognisable languages (that is, recognisable by finite automata): *the birth of automata theory*. In the past half century, automata theory has been established as one of the most important foundations of computer science, a huge amount of remarkable research have been made. Automata are the simplest mathematical model of computation, so simple that they take forms in various diverse areas. An important notion often has several different characterisation, the class of regular languages is exactly such a type: it can be characterised via nondeterministic finite automata (Rabin-Scott [48]), via monadic second-order logic (Büchi [11]), via finite monoids (Myhill [43]), and via topological manner (Hunter [27]), etc...

Classifying regular languages

One of the most rich topic in the theory of regular languages is classifying regular languages. In this topic, the algebraic characterisation of regular languages – finite monoid recognisability – plays a crucial role. Many important subclasses of regular languages have been related with subclasses of finite monoids. Refrain from presenting a comprehensive history, I quote the following from the survey paper by Diekert et al. [15]: "When considering subclasses of regular languages, it turns out

that finite monoids are a very advantageous point of view. For instance, Schützenberger has shown that a language is star-free if and only if it is recognized by some finite and aperiodic monoid [54]. Brzozowski and Simon as well as McNaughton have shown independently that it is decidable whether a regular language is locally testable by describing an algebraic counterpart [10, 41]. Simon has characterized piecewise-testable languages in terms of finite $\mathcal{J}$-trivial monoids [56]. Inspired by these results, Eilenberg has proposed a general framework for such correspondences between classes of regular languages and classes of finite monoids [18]."

The framework stated in the above quotation is *Eilenberg's variety theory* which was introduced in the book "Automata, Languages and Machines: Volume B" [18] written by Eilenberg. A variety of languages is a class of regular languages closed under Boolean operations, left and right quotients and inverses of morphisms. The algebraic counterpart of a variety is a (pseudo)variety of finite monoids: a class of finite monoids closed under submonoids, quotients and finite direct products. The acclaimed Eilenberg's variety theorem [18] states that varieties of languages are in one-to-one correspondence with varieties of finite monoids. Since the work of Eilenberg, the theory has been deeply studied and it leads developments involving not only automata theory but also finite semigroup theory, to borrow Margolis's phrase [39]: "It is not an overstatement to say that since 1976 with the appearance of Eilenberg's Volume B, the vast majority of finite semigroup theory has been involved with the study of pseudovarieties of finite semigroups and monoids and their relationship to automata theory."

Every variety of languages captures some phenomenon which, sometimes trivial, sometimes essential, but always can be interpreted in various ways: *syntactically, algebraically*, and possibly, *logically*. The thesis sheds new light, by using variety theoretic techniques, on the relation between two different notions. The first notion, comes from basic semigroup theory, is the *existence of a zero element*. The second notion, comes from finite model theory, is a certain extreme phenomenon named *zero-one law*.

Zero-one law for finite graphs

In finite model theory, it is known that many logics can not express, intuitively speaking, any nontrivial counting property of graphs. This phenomenon is called as the *zero-one law for finite graphs* (*cf.* [35]). We say that a logic $\mathcal{L}$ over finite graphs has the zero-one law if every property Φ definable in $\mathcal{L}$ is either *almost surely true* or *almost surely false*, namely, either Φ is true for almost all finite graphs, or Φ is false for almost all finite graphs:

$$\lim_{n\to\infty} \frac{\text{the number of all } n\text{-vertices graphs that satisfies } \Phi}{\text{the number of all } n\text{-vertices graphs}} \in \{0,1\}.$$

It turns out that many nontrivial properties of finite graphs are either almost surely true or almost surely false. For example, on the one hand, almost all finite graphs are connected, rigid (i.e., have no nontrivial automorphism) and Hamiltonian, on the other hand, almost no finite graph is planar (*cf.* [13]).

The famed Fagin's theorem [20] states that first-order logic for finite graphs has the zero-one law. Moreover, any first-order definable property is almost surely true if and only if it is true on a certain infinite graph: the *random graph*. Fagin's beautiful characterisation leads to the fact that, for any first-order sentence Φ, it is decidable whether Φ is almost surely true or not (*cf.* Corollary 12.11 in [35]). After the work of Fagin, much ink has been spent on the zero-one law for logics over finite graphs. It is now known that many stronger logics (e.g., logic with a fixed point operator [7], finite variable infinitary logic [32] and certain fragments of second-order logic [33]) have the zero-one law. Here I would like to emphasise the remarkable fact about the zero-one law. It is known that finite satisfiability (i.e., the existence of a finite model) of first-order definable property for finite graphs is undecidable due to Trakhtenbrot's theorem [65]. Thus, for a given first-order sentence Φ, while it is undecidable whether Φ is true for all finite graphs, it is decidable whether Φ is true in almost all finite graphs! All these results can be easily extended to arbitrary finite relational structures (*cf.* [35]).

By contrast, though many logics have the zero-one law, their extensions with linear order no longer have it. In fact, while first-order logic over finite graphs has the zero-one law, its extension with a linear order does not [14].

Zero-one law for regular languages

A logic over finite words is one of the most important instance of logics with linear order in computer science. The question then naturally arises as to which logical fragments over finite words, or class of languages, have the zero-one law? The main topic of the thesis is this one: the zero-one law for finite words, or more emblematically, the *zero-one law for regular languages*. We call a language (i.e., set of finite words) L zero-one, or obeys the zero-one law, if L is either *almost empty* or *almost full*, namely, either L contains almost all finite words, or L does not contain almost all finite words:

$$\lim_{n\to\infty} \frac{\text{the number of all words of length } n \text{ in } L}{\text{the number of all words of length } n} \in \{0,1\}.$$

The original motivation of this work is the following question: *is there a nice (decidable) characterisation of the class of regular zero-one languages?* In this thesis I give an algebraic and automata theoretic characterisation of the zero-one law for regular languages. Roughly speaking, I prove the following "Zero-One Theorem" (precise statement is Theorem 2.3.1, Chapter 2): *a regular language L is zero-one if and only if its syntactic monoid has a zero element, or equivalently L or its complement includes a language of the form A^*wA^* for some word w*. The proof gives an effective automata characterisation of the zero-one law for regular languages, and it leads to a linear time algorithm for testing whether a given regular language is zero-one if it is given by an n-states deterministic automaton.

The key points of the proof of Zero-One Theorem are *closure properties of the class of zero-one languages* and *Eilenberg's lemma* which was crucial in Eilenberg's variety theorem.

Structure of the thesis

The thesis consists of six chapters. In Chapter 1, I give the necessary definitions and terminology of basic automata theory. Chapter 2 provides a detailed exposition of the notion of the zero-one law for regular languages. The main result of the thesis – Zero-One Theorem – will be stated in this chapter (Theorem 2.3.1). Closure properties of the class of all zero-one languages are investigated in Chapter 3. Eilenberg's lemma is also given in this chapter. An automata theoretic proof of Zero-One Theorem is given in Chapter 4. In this chapter, I introduce two new classes of automata: zero automata and quasi-zero automata. These classes of automata play a crucial role in the proof. Chapter 5 describes a linear time algorithm for testing whether a given regular language is zero-one (Theorem 5.1.1). Some logical aspects of the zero-one law for regular languages are also described in this chapter. Zero-One Theorem gives us a simple necessary and sufficient condition for a regular language to be zero-one, however, it is *not true beyond regular languages*. Simple counterexamples, zero-one languages whose syntactic monoid have no zero element, are given in Chapter 6. In this chapter, a new technique for proving non-regularity of languages is established.

I try to keep all chapters as self-contained as possible. At the end of each chapter, I provide "Bibliographic Notes" which can serve as a reader's guide to explore related works and topics. I use square brackets as an equivalent to "respectively", as in the following sentence: a language L is almost full [almost empty] if it contains [does not contain] almost all finite words.

R. SIN'YA

Tokyo, November 2015

ACKNOWLEDGMENTS

I gratefully acknowledge helpful discussions with Prof. Ryo Kashima on several points in the thesis. Special thanks also go to Prof. Yasuhiko Minamide and Prof. Makoto Kanazawa whose meticulous comments were an enormous help to me. I would like to acknowledge the encouragements of my colleagues, Naosuke Matsuda, Yoshiki Nakamura, and Takuro Umekita. My senior colleague Takeo Uramoto introduced me to the variety theory and has encouraged me throughout this research. I wish to express my gratitude to Prof. Masami Ito for his valuable advice. Grateful acknowledgement is made to The Wiley Publishing Company which provided this beautiful LaTeXtemplate.

I am grateful to Prof. Jacques Sakarovitch whose comments and suggestions were innumerably valuable throughout the course of my study. I decided to dive into automata theory, when I was a first year master's student, because I met his excellent book "*Elements of Automata Theory*" [50].

CONTENTS

CHAPTER 1

PRELIMINARIES

> Mais c'est plutôt le sens figuré qui m'intéresse. La théorie des automates comme connaissance de base, fondamentale, connue de tous et utilisée partous qui fait partie du «paysage intellectuel » depuis si longtemps qu'on ne l'y remarquerait plus. Et pourtant, elle y est, elle le structure, elle l'organise; la connaître permet de s'y orienter.
>
> —Jacques Sakarovitch, "*Éleménts de theorie des automates*".

> I am more interested, however, in the figurative sense: automata theory as a basic, fundamental subject, known and used by everyone, which has formed part of the intellectual landscape for so long that it no longer noticed. And yet, there it is, structuring it, organising it: and knowing it allows us to orient ourselves.
>
> —(English translation, "*Elements of Automata Theory*"[50]).

All automata considered in the thesis are *deterministic finite*, *complete*, and *accessible* (precise definition is given in this chapter). We refer the reader to [50, 34, 46] for background material.

1.1 Regular Languages

Let A be a nonempty finite set called an *alphabet*, whose elements are called *letters*. A finite sequence of elements of A is called a finite word over A, or just a *word*. We denote the sequence $(a_0, a_1 \cdots a_n)$ by mere juxtaposition:

$$a_0 a_1 \cdots a_n.$$

For a word $w = a_0 a_1 \cdots a_n$, the *length of* w is denoted by $|w| = n + 1$. We denote by A^* the set of all words over A, and denote by A^n the set of all words of length n over A. A set of words is endowed with the operation of the *concatenation*, which associates with two words $u = a_0 a_1 \cdots a_i$ and $v = b_0 b_1 \cdots b_j$ the word $uv = a_0 a_1 \cdots a_i b_0 b_1 \cdots b_j$. The concatenation is obviously associative. It has an identity, the *empty word*, denoted by ε, which is the empty sequence: $|\varepsilon| = 0$. Note that A^* always includes the empty word. We say that a word v is a *factor of* a word w if, there exists x, y in A^* such that $w = xvy$. For the word $w = a_0 a_1 \cdots a_n$, we denote by $w^r = a_n a_{n-1} \cdots a_0$ the *reversal of* w.

A *language over* A is a set of words over A, that is, a subset of A^*. The set of all words A^* over A is called the *full language*. We denote by $\overline{L} = A^* \setminus L$ the *complement* of L. The class of regular languages over A is the smallest class of languages that contains emptyset $\emptyset$ and each of the singleton $\{a\}$ for $a \in A$, and that is closed under the following three operations:

union: $L \cup K$;

concatenation: $LK = \{vw \mid v \in L, w \in K\}$;

Kleene star: $L^* = \bigcup_{n \in \mathbb{N}} L^n = \{\varepsilon\} \cup L \cup LL \cup LLL \cup \cdots$.

We shall identify the singleton $\{w\}$ with its unique element w in A^*. It is well known that the class of regular languages enjoys good closure properties (e.g., closed under the *complement* and *intersection*).

If a language L over A satisfies $A^*LA^* = L$, then L is called an *ideal language*. The language A^*wA^* for a word w in A^* is called the *ideal language generated by* w. A^*wA^* can be regarded as the set of all words that contain w as a factor. A word w is *forbidden for a language* L if it is a factor of no element of L, i.e., $A^*wA^* \cap L = \emptyset$. Dually, a word w is *admissible for a language* L if every word containing w as a factor is in L, i.e., $A^*wA^* \subseteq L$.

Let L be a language over A and let u be a word of A^*. The *left* [*right*] *quotient* $u^{-1}L$ $[Lu^{-1}]$ of L by u is defined by:

$$u^{-1}L = \{v \in A^* \mid uv \in L\} \quad [Lu^{-1} = \{v \in A^* \mid vu \in L\}].$$

The well-known Myhill-Nerode theorem [44] states that every regular language has only a finite number of left and right quotients.

EXAMPLE 1.1

Here we give a few simple examples of regular languages over $A = \{a, b\}$.

- Any finite set of words is obviously regular: it can be defined by the finite combination of concatenations and unions.
- The set of all words of even length is regular: it can be defined by $(AA)^* = \{w \in A^* \mid |w| \text{ is even}\}$.
- The set of all words beginning with the letter a in A is regular: it can be defined by $aA^* = \{aw \mid w \in A^*\}$.
- The set of all words that contain a sequence of a's followed by a sequence of b's is regular: it can be defined by $a^*b^* = \{a^n b^m \mid n, m \geq 0\}$.

Here we give two examples of *non-regular* languages.

- The set of all *palindromes* $\{w \in A^* \mid w = w^r\}$ is not regular.
- The Dyck language over $A = \{[,]\}$ (intuitively, the set of all balanced square brackets):
$$\{\varepsilon, [], [[]], [][], [[[]]], [[][]], [[]][], [][[]], [][][], \ldots\}$$
is not regular (more formal definition is given in Chapter 6).

The set of all palindromes and the Dyck language are classical examples of non-regular languages. The non-regularity of these languages can be easily proved via several ways, like as the *pumping lemma* or Myhill-Nerode theorem (*cf.* [50]). In Chapter 6, however, we give the proof of these non-regularity by using our new technique.

1.2 Automata and Counting

An *(complete deterministic finite) automaton* over A is a quintuple $\mathcal{A} = \langle Q, A, \cdot, q_0, F\rangle$ where:

- Q is a finite set of *states*;
- $\cdot : Q \times A \to Q$ is a *transition function*, which can be extended to a mapping $\cdot : Q \times A^* \to Q$ by $q \cdot \varepsilon = q$ and $q \cdot aw = (q \cdot a) \cdot w$ where $q \in Q, a \in A$ and $w \in A^*$;
- $q_0 \in Q$ is an *initial state*, and $F \subseteq Q$ is a set of *final states*.

The *language recognised by* $\mathcal{A}$ is denoted by $L(\mathcal{A}) = \{w \in A^* \mid q_0 \cdot w \in F\}$. We say that $\mathcal{A}$ *recognises* L if $L = L(\mathcal{A})$.

EXAMPLE 1.2

In this thesis, an automaton is illustrated by its transition diagram like Figure 1.1. Each final state will be indicated by an outgoing edge without a label, and the initial state will be indicated by an incoming edge without a label. One can easily observe that the automaton in Figure 1.1 has q_0 as its initial and finite state, and recognises the language $(\mathrm{TITECH})^*$.

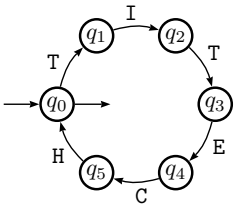


Figure 1.1 An automaton recognising $(\mathrm{TITECH})^*$.

It is a basic fact that, for any regular language L, there exists a unique automaton recognises L which has the minimum number of states: the *minimal automaton* of L and we denote it by $\mathcal{A}_L$. For each pair of states p, q in Q, we say that q *is reachable from* p if, there exists a word w such that $p \cdot w = q$. $\mathcal{A}$ is called *accessible* if every state q in Q is reachable from the initial state q_0. In this thesis, all considered automata are *accessible*. The following theorem is fundamental.

Theorem 1.2.1 (Kleene [29, 30]) *A language L is regular if and only if it is recognised by an automaton.*

The *counting function* $\gamma_n(L)$ of a language L counts the number of all words of length n in L:

$$\gamma_n(L) = |\{w \in L \mid |w| = n\}| = |L \cap A^n|.$$

If L is a regular language, we can represent its counting function $\gamma_n(L)$ by using the nth power of a certain matrix related to an automaton that recognises L. More precisely, for any regular language L and any automaton $\mathcal{A} = \langle Q, A, \cdot, q_0, F\rangle$ that recognises L, the following equation holds:

$$\gamma_n(L) = IM^nF$$

where M is the $|Q| \times |Q|$ matrix, I and F are the row and column vectors defined as follows:

$$M_{i,j} = |\{a \in A \mid q_i \cdot a = q_j\}|, \quad I_i = \begin{cases} 1 & \text{if } i = 0, \\ 0 & \text{if } i \neq 0, \end{cases} \quad F_i = \begin{cases} 1 & \text{if } q_i \in F, \\ 0 & \text{if } q_i \notin F. \end{cases}$$

M is called the *adjacency matrix of* $\mathcal{A}$, I $[F]$ is called the *initial* [*final*] *vector of* $\mathcal{A}$. Since $(M^n)_{i,j}$ equals to the number of all paths of length n from q_i to q_j, the right hand side of Equation 1.1 equals to the number of all paths of n from the initial state to final states, that is, the number of all words of length n in L.

EXAMPLE 1.3

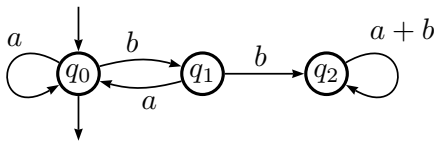


Figure 1.2 An automaton $\mathcal{A}_{fib}$.

Consider the automaton $\mathcal{A}_{fib}$ which recognises $L = \{a, ba\}^*$ illustrated in Figure 1.2. Let:

$$M = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 2 \end{bmatrix}, \qquad I = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \qquad F = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

are its adjacency matrix, initial and final vectors. Then the followings hold.

$$\begin{aligned}
\gamma_0(L) &= |\{\varepsilon\}| = 1, \\
\gamma_1(L) &= |\{a\}| = 1, \\
\gamma_2(L) &= |\{aa, ba\}| = 2, \\
\gamma_3(L) &= |\{aaa, aba, baa\}| = 3, \\
\gamma_4(L) &= |\{aaaa, aaba, abaa, baaa, baba\}| = 5, \\
&\vdots \\
\gamma_n(L) &= IM^nF = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 2 \end{bmatrix}^n \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} S \begin{bmatrix} 2 & 0 & 0 \\ 0 & \frac{1}{2}(1-\sqrt{5}) & 0 \\ 0 & 0 & \frac{1}{2}(1+\sqrt{5}) \end{bmatrix}^n S^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
&\quad \text{where} \quad S = \begin{bmatrix} 0 & \frac{1}{2}(1-\sqrt{5}) & \frac{1}{2}(1+\sqrt{5}) \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \\
&= \frac{1}{\sqrt{5}} \left\{ \left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right\}
\end{aligned}$$

The last equation means that $\gamma_n(L)$ equals to the $(n+1)$st Fibonacci number.

1.3 Monoids and Morphisms

A monoid M is a set equipped with an associative binary operation and the *identity element* 1 that satisfies $m1 = 1m = m$ for all m in M. In particular, the full language A^* is called the *free monoid over* A: its identity element is ε, and it is equipped with the concatenation as an associative binary operation. A *morphism* is a map ϕ from a monoid M into a monoid N that satisfies $\phi(1_M) = 1_N$ where $1_M[1_N]$ is an identity of $M[N]$, and ϕ preserves the binary operation:

$$\phi(xy) = \phi(x)\phi(y)$$

for every x, y in M. We say that *a monoid M recognises a language L over A* if, there exist a morphism $\phi : A^* \to M$ and a subset P of M such that:

$$\phi^{-1}(P) = L.$$

An element $\mathbf{0}$ of M is said to be a *zero* if, $\mathbf{0}x = x\mathbf{0} = \mathbf{0}$ holds for all x in M. A monoid M that have a zero element is said to be *a monoid with zero*.

EXAMPLE 1.4

Let $M_l = \{1, \overline{a}, \overline{b}\}$ be a finite monoid with the identity 1 whose product is defined as $\overline{a}x = \overline{a}$ and $\overline{b}x = \overline{b}$ for all x in M_l. Let $A = \{a, b\}$ be an alphabet and $\phi : A^* \to M_l$ be a morphism such that $\phi(a) = \overline{a}$ and $\phi(b) = \overline{b}$. Then M_l recognises three regular languages aA^*, bA^* and $\{\varepsilon\}$:

$$aA^* = \phi^{-1}(\overline{a}), \quad bA^* = \phi^{-1}(\overline{b}), \quad \phi^{-1}(1) = \{\varepsilon\}.$$

The *syntactic congruence* of a language L over A is the equivalence relation $\sim_L$ defined on A^* by $u \sim_L v$ if and only if $xuy \in L \Leftrightarrow xvy \in L$ holds for all x, y in A^*. The quotient $A^*/\sim_L$ is called the *syntactic monoid* of L and the natural morphism $\phi_L : A^* \to A^*/\sim_L$ is called the *syntactic morphism* of L. Let $\mathcal{A} = \langle Q, A, \cdot, q_0, F\rangle$ be an automaton. Each word w in A^* defines the transformation $w : q \mapsto q \cdot w$ on Q. The *transition monoid* of $\mathcal{A}$ is the transformation monoid generated by the generators $a : q \mapsto q \cdot a$ in A. It is well known that the syntactic monoid of a regular language is equal to the transition monoid of its minimal automaton. The following well-known theorem states that the converse is also true.

Theorem 1.3.1 (Myhill [43]) *A language L is regular if and only if it is recognised by some finite monoid. In particular, L is regular if and only if its syntactic monoid is finite.*

1.4 Bibliographic Notes

Kleene used the term *regular events* and in his 1951 paper "Representation of events in nerve nets and finite automata" [29] wrote: "We would welcome any suggestions

as to a more descriptive term". After that, in the later version of the paper [30], the above phrase was deleted. The definition of the syntactic monoid was firstly introduced by Schützenberger in 1956 [53]. It later appeared in the paper by Rabin and Scott [48], where the notion is credited to Myhill. For a semigroup S and its subset T, the *principal congruence determined by* T is the equivalent relation $\equiv_T$ defined on S by $u \equiv_T v$ if and only if $xuy \in T \Leftrightarrow xvy \in T$ holds for all x, y in S (*cf.* [28]). The syntactic congruence is a particular case of a principal congruence (when $S = A^*$). The notion of principal congruence has been studied, albeit with sometimes different meanings, from early 1940s: by Dubreil in 1941 [17], Teissier in 1951 [64], Pierce in 1954 [45]. A more detailed and complete history can be found in [12].

CHAPTER 2

ZERO-ONE LAW FOR REGULAR LANGUAGES

ある形式文法の族を導入した際に，まずそこで定義される言語の族の閉包性を調べ，決定問題を考え，正則集合との関係を見るという理論展開の鋳型はこのとき (Bar-Hillel et al. [3]) にできたといえる．

—Setsuo Arikawa, “数理言語学入門” (Japanese translation of [26]).

In this chapter, we provides a detailed exposition of the notion of the zero-one law for regular languages. The main result of the thesis – Zero-One Theorem – will be stated in this chapter (Theorem 2.3.1).

2.1 Zero-One Languages: $\mathcal{ZO}$ and $\mathcal{ZO}^{\mathrm{Reg}}$

Let L be a language over a non-empty finite alphabet A. Recall that the counting function $\gamma_n(L)$ of L counts the number of different words of length n in L: $\gamma_n(L) = |L \cap A^n|$. The *probability function* $\mu_n(L)$ of L is the fraction defined by:

$$\mu_n(L) = \frac{\gamma_n(L)}{\gamma_n(A^*)} = \frac{|L \cap A^n|}{|A^n|}.$$

The *asymptotic probability*, or *measure*, $\mu(L)$ of L is defined by:

$$\mu(L) = \lim_{n\to\infty} \mu_n(L)$$

if the limit exists. If two languages L and K over A are mutually disjoint ($L \cap K = \emptyset$), then clearly $\mu(L \cup K) = \mu(L) + \mu(K)$ and $\mu(\overline{L}) = 1 - \mu(L)$ hold if both $\mu(L)$ and $\mu(K)$ exist. We can regard $\mu_n(L)$ as the *probability* that a randomly chosen word of length n is in L, and $\mu(L)$ as its *asymptotic probability*. Then we introduce a new class of regular languages which is the main target of this thesis.

Definition 2.1.1 (zero-one language) A *zero-one language* L is a language whose asymptotic probability $\mu(L)$ is either zero or one. We denote by $\mathcal{ZO}$ *the class of all zero-one languages*, and by $\mathcal{ZO}^{\mathrm{Reg}}$ *the class of all zero-one regular languages*.

We call L *almost full* [*almost empty*] if $\mu(L) = 1$ [$\mu(L) = 0$] holds. We say that L *obeys the zero-one law* if L is either almost full or almost empty.

EXAMPLE 2.1

We now enumerate a few examples of $\mathcal{ZO}^{\mathrm{Reg}}$.

- The full language is almost full, and the empty language is almost empty. That is, the set of all words A^* over A satisfies $\mu(A^*) = 1$, and its complement $\emptyset$ satisfies $\mu(\emptyset) = 0$.
- Consider aA^* the set of all words which start with the letter a in A. Then the following holds:
$$\mu_n(aA^*) = \frac{|aA^{n-1}|}{|A^n|} = \frac{1}{|A|}.$$
Hence $\mu((aA)^*) = 1/|A|$ holds and aA^* is *not* zero-one if $|A| \geq 2$.
- Consider $(AA)^*$ the set of all words with even length. Then:
$$\mu_n((AA)^*) = \begin{cases} 1 & \text{if } n \text{ is even,} \\ 0 & \text{if } n \text{ is odd.} \end{cases}$$
Hence, its limit $\mu((AA)^*)$ does not exist.

Thus, for some regular language L, the asymptotic probability $\mu(L)$ is either zero or one, for some, like $L = aA^*$ where $|A| \geq 2$, $\mu(L)$ could be a real number between zero and one, and for some, like $L = (AA)^*$, it may not even exist. It is previously known that there exists a cubic time algorithm computing $\mu(L)$ for any regular language L if L is given by an n-states automaton [8] and $\mu(L)$ is always rational [4, 51] (see Section 2.4).

Remark 2.1.1 Technically speaking, the function μ defined here is not a measure of the standard definition in measure theory (*cf.* [63]). The μ defined here obviously satisfies the following properties:

- $\mu(\emptyset) = 0$, $\mu(\overline{L}) = 1 - \mu(L)$ if $\mu(L)$ exists.
- the *(finite) additivity*: whenever two languages K, L are disjoint and both $\mu(K)$ and $\mu(L)$ exist, then $\mu(K \cup L) = \mu(K) + \mu(L)$.
- the *(finite) subadditivity*: $\mu(K \cup L) \leq \mu(K) + \mu(L)$ if both $\mu(K)$ and $\mu(L)$ exist.
- the *monotonicity*: $K \subseteq L$ implies $\mu(K) \leq \mu(L)$ if both $\mu(K)$ and $\mu(L)$ exist.

But μ does not satisfy the *countable additivity*. That is, $\mu(\bigcup_{w \in A^*} \{w\}) = \mu(A^*) = 1$, although $\mu(\{w\}) = 0$ holds for each word w in A^*.

2.2 Languages with Zero: $\mathcal{Z}$ and $\mathcal{Z}^{\mathsf{Reg}}$

In this thesis, we show that the following class of languages exactly captures the zero-one law for regular languages.

Definition 2.2.1 A *language with zero* is a language whose syntactic monoid has a zero element. We denote by $\mathcal{Z}$ *the class of all languages with zero*, and by $\mathcal{Z}^{\mathsf{Reg}}$ *the class of all regular languages with zero*.

EXAMPLE 2.2

We now enumerate a few examples related with $\mathcal{Z}^{\mathsf{Reg}}$.

- The trivial monoid $M = \{1\}$ is actually with zero: the identity element 1 is also a zero element. Hence the full language and the empty language over A are languages with zero, because the syntactic monoid of these languages is the trivial monoid.
- Let $L = A^*aA^*$ be the set of all words that contain a as a factor. One can easily verify that the syntactic monoid of L is the two element monoid with zero $M_L = \{0, 1\}$ and $\phi_L^{-1}(0) = A^*aA^*$. The identity element 1 represents "any word that does not contain a", and the zero element 0 represents "any word that contains at least one a". Thus $L = A^*aA^*$ is with zero.

- Consider aA^* the set of all words which start with the letter a in A. The syntactic monoid of aA^* is the monoid M_l defined in Example 1.4. Since M_l does not have a zero element, aA^* is not with zero.

2.3 Zero-One Theorem: $\mathcal{ZO}^{\mathrm{Reg}} = \mathcal{Z}^{\mathrm{Reg}}$

Now we give a precise statement of the our main result. The definition of two classes of automata – *zero automata* and *quasi-zero automata* – will be given in Chapter 4.

Theorem 2.3.1 (Zero-One Theorem) *Let L be a regular language and $\mathcal{A}_L$ be the minimal automaton of L. Then the following five conditions are equivalent.*

① *$\mathcal{A}_L$ is zero.*

② *L is with zero.*

③ *L or $\overline{L}$ contains an ideal language.*

④ *L obeys the zero-one law.*

⑤ *L is recognised by a quasi-zero automaton.*

The remarkable fact is that, $\mathcal{ZO}^{\mathrm{Reg}} = \mathcal{Z}^{\mathrm{Reg}}$ holds even though these two notions seem completely different from each other; $\mathcal{ZO}^{\mathrm{Reg}}$ is defined by the asymptotic behavior of its probability, $\mathcal{Z}^{\mathrm{Reg}}$ is defined by the existence of a zero of its syntactic monoid. The proof of this theorem is given in Chapter 4. We will prove this theorem as a cyclic chain of implications: ① $\Rightarrow$ ② $\Rightarrow$ ③ $\Rightarrow$ ④ $\Rightarrow$ ①, and ① $\Leftrightarrow$ ⑤ independently. We should notice that the most difficult part of this proof is the implication ④ $\Rightarrow$ ①, while the former part ① $\Rightarrow$ ② $\Rightarrow$ ③ $\Rightarrow$ ④ is easy. The key point of this difficult part is *closure properties of $\mathcal{ZO}$* which stated in the next chapter. Two automata characterisation ① and ⑤ play a crucial role in the proof.

2.4 Bibliographic Notes

Densities and algebraic coding theory

The notion of probability μ_n for regular languages has been studied by Berstel [4] from 1973, Salomaa and Soittola [51] from 1978 in the context of the *theory of formal power series*. They proved that $\mu_n(L)$ has finitely many accumulation points and each accumulation point is rational. Another approach, based on *Markov chain theory*, was presented by Bodirsky et al. [8]. They investigate the algorithmic complexity of computing accumulation points of L and introduced an $\mathrm{O}(n^3)$ algorithm to compute $\mu(L)$ for any regular language L (and hence whether L is zero-one), if L is given by an n-states automaton. There is an alternative definition of the asymptotic probability of L over A:

$$\mu^*(L) = \lim_{n\to\infty} \frac{\sum_{i=0}^{n} |L \cap A^i|}{\sum_{i=0}^{n} |A^i|}.$$

In fact, Berstel uses this definition of μ^* in his first research on this topic [4]. However, for any language L over A, $\mu^*(L)$ exists if and only if $\mu(L)$ exists and they are equal by well-known *Stolz–Cesáro theorem* and its partial converse (*cf.* Theorem 1.22 and 1.23 in [42]). Thus our Zero-One Theorem does not depend on the definition we use.

A similar notion, *density* of a language have also been studied in *algebraic coding theory* (*cf.* [5, 6]). A *probability distribution* π on A^* is a function $\pi : A^* \to [0,1]$ such that $\pi(\epsilon) = 1$ and $\sum_{a\in A} \pi(wa) = \pi(w)$ for all w in A^*. As a particular case, the *uniform Bernoulli distribution* is a morphism from A^* into $[0,1]$ such that $\pi(a) = 1/|A|$ for all a in A. We denote by $A^{(n)} = \{w \in A^* \mid |w| < n\}$ the set of all words of length less than n over A. The *density* $\delta(L)$ of L then defined by the following:

$$\delta(L) = \lim_{n\to\infty} \frac{1}{n}\pi\left(L \cap A^{(n)}\right)$$

where π is a probability distribution on A^*. A monoid M is called *well founded* if it has a unique minimal ideal, if moreover this ideal is the union of the minimal left ideals of M, and also of the minimal right ideals, and if the intersection of a minimal right ideal and of a minimal left ideal is a finite group. An elementary result from analysis shows that if π is the uniform Bernoulli distribution and $\mu(L)$ exists, then $\delta(L) = \lim_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n} \mu_i(L)$ also exists and $\delta(L) = \mu(L)$ holds. The converse, however, does not hold (e.g., $\delta((AA)^*) = 1/2$). In their book [6], Berstel et al. proved Theorem 13.4.5 which states that, for any well founded monoid M and morphism $\phi : A^* \to M$, $\delta(\phi^{-1}(m))$ has a limit for every m in M. Furthermore, this density is non-zero if and only if m in the minimal ideal K of M from which we obtain $\delta(\phi^{-1}(K)) = 1$. Since every monoid with zero is well founded, Theorem 13.4.5 implies that, every language with zero is zero-one (i.e., ② ⇒ ④, "easy part" of our Zero-One Theorem). Some other related results can be found in the *theory of probabilities on algebraic structures* initiated by Grenander [25] and Martin-Löf [40].

The point to observe is that the techniques presented in this thesis are purely automata theoretic. We did not use, to prove Zero-One Theorem, any probability theoretic tools: like as measure theory, formal power series, Markov chain, algebraic coding theory, etc. This point deserves explicit emphasise.

Languages defined by the counting function

There exist other classes of languages related to zero. We call the language A^* is *full*. If the counting function of a language L has *bounded density*, i.e., $\gamma_n(L) = \mathrm{O}(1)$ with respect to n, then L is called *slender*. A language is *sparse* if it has a polynomial density, i.e., $\gamma_n(L) = \mathrm{O}(n^k)$ for some $k > 0$. Finally, a language L is called *coslender* if its complement is slender, and a language L is called *cosparse* if its complement is sparse. Gehrke, Grigorieff and Pin [23] proved that both the class of all *sparse or cosparse languages* and the class of all *slender or coslender languages* are closed under Boolean operations, left and right quotients. Moreover, they showed that these two class of languages can be defined by certain *profinite equation* related to zero. Details of these results can be found in the book by Pin [46].

CHAPTER 3

CLOSURE PROPERTIES OF $\mathcal{ZO}$ AND EILENBERG'S LEMMA

> Wherever there is an algebraic structure for recognizing languages, there is an Eilenberg theorem. This theorem gives a bijective mapping between classes of languages with good closure properties (language varieties) and classes of monoids with good closure properties (monoid varieties).
>
> —Mikołaj Bojańczyk and Igor Walukiewicz, "*Forest Algebras*" [9].

In formal language theory, good closure properties of a class of languages sometimes imply a good structural theorem of that class. The key points of the proof of Zero-One Theorem are closure properties of the class of all zero-one languages $\mathcal{ZO}$. In this chapter, we introduce *Eilenberg's lemma* which is based on certain closure properties of a class of languages.

3.1 Closure Properties of $\mathcal{ZO}$

We first introduce the following lemma.

Lemma 3.1.1 *Let L be a language over A and w be a word in A^k for some $k \geq 0$. Then the asymptotic probability of L exists if and only if the asymptotic probability of the language wL $[Lw]$ exists. Moreover, these limits satisfy the equation $\mu(wL) = \mu(Lw) = |A|^{-k}\mu(L)$.*

Proof: Since wL and Lw clearly have the same counting function, we only have to prove the case of wL. For every u, v in A^k such that $u \neq v$, the two languages uL and vL are mutually disjoint and these counting functions coincides:

$$\gamma_n(uL) = \gamma_n(vL) = \begin{cases} 0 & n < k, \\ \gamma_{n-k}(L) & n \geq k. \end{cases}$$

This shows that uL and vL have the same asymptotic probability if its exists. We can easily verify that the following equation holds for any language L over A and $k \geq 0$:

$$\mu_{n+k}(A^k L) = \frac{|A^k L \cap A^{n+k}|}{|A^{n+k}|} = \frac{|A^k (L \cap A^n)|}{|A^k A^n|} = \frac{|L \cap A^n|}{|A^n|} = \mu_n(L).$$

It follows from what has been said that $\mu(A^k L)$ exists if and only if $\mu(L)$ exists and in that case they are equal $\mu(A^k L) = \mu(L)$. Hence it follows that $\mu(uL)$ exists if and only if $\mu(L)$ exists by the following equation:

$$\mu(L) = \mu\left(A^k L\right) = \sum_{u \in A^k} \mu(uL) = |A|^k \mu(uL).$$

■

Now we prove that $\mathcal{ZO}$ enjoys good closure properties that are necessary to apply Eilenberg's lemma introduced in the next section.

Proposition 3.1.1 *$\mathcal{ZO}$ is closed under Boolean operations, left and right quotients.*

Proof: It is obvious that $\mathcal{ZO}$ is closed under complement since $\mu(\overline{L}) = 1 - \mu(L) \in \{0, 1\}$. Next we assume that $\mu(L) = \mu(K) = 0$, then $\mu(L \cup K) = 0$ is obvious from the subadditivity of μ:

$$\mu(L \cup K) \leq \mu(L) + \mu(K) = 0.$$

Then one can easily verify that the following hold for L, K in $\mathcal{ZO}$:

- $\mu(L \cap K) = \min(\mu(L), \mu(K))$,
- $\mu(L \cup K) = \max(\mu(L), \mu(K))$.

We then prove that $\mathcal{ZO}$ is closed under left quotients. We only have to consider the left quotient by a letter $a^{-1}L$ since every left quotient $w^{-1}L = (a_0 \cdots a_n)^{-1}L$ is a successive application of letter quotients $a_n^{-1} \cdots (a_0^{-1}L)$. First we assume $\mu(L) = 0$. One can easily verify that $aa^{-1}L = L \cap aA^* \subseteq L$ and hence $\mu(aa^{-1}L) = \mu(L) = 0$ for each letter a. In addition, $\mu(aa^{-1}L)$ coincides with $\mu(a^{-1}L)$ for each letter a, because $\mu(aa^{-1}L) = |A|^{-1}\mu(a^{-1}L) = 0$ by Lemma 3.1.1 whence $\mu(a^{-1}L) = 0$.

Next we assume $\mu(L) = 1$. Then $\mu(\overline{L}) = 0$ and:

$$\begin{aligned} a^{-1}\overline{L} &= \{w \in A^* \mid aw \in \overline{L}\} \\ &= \{w \in A^* \mid aw \notin L\} = \overline{a^{-1}L} \end{aligned}$$

holds. We therefore obtain:

$$\begin{aligned} \mu(a^{-1}L) &= 1 - \mu(\overline{a^{-1}L}) \\ &= 1 - \mu(a^{-1}\overline{L}) = 1 - 0 = 1. \end{aligned}$$

We can prove that $\mathcal{ZO}$ is closed under right quotients by the same manner. ∎

Since the class of regular languages is closed under Boolean operations and quotients, the following corollary follows from Proposition 3.1.1.

Corollary 3.1.1 *$\mathcal{ZO}^{\mathsf{Reg}}$ is closed under Boolean operations, left and right quotients.*

Proposition 3.1.2 *$\mathcal{ZO}$ is not closed under inverses of morphisms.*

Proof: Let $L = (aa)^*$ be a language over $A = \{a, b\}$, let $\phi : A^* \to A^*$ be the morphism such that $\phi(a) = \phi(b) = a$. One can easily verify $\mu(L) = 0$ and hence L is in $\mathcal{ZO}^{\mathsf{Reg}}$. Then the inverse image of L is $\phi^{-1}((aa)^*) = (AA)^*$, but $(AA)^*$ is not in $\mathcal{ZO}$ as we stated in Example 2.2. ∎

Corollary 3.1.2 *$\mathcal{ZO}^{\mathsf{Reg}}$ is not closed under inverses of morphisms.*

The counterexample given in Proposition 3.1.2 can be found in Pin's book [46]. He uses it to prove that $\mathcal{Z}^{\mathsf{Reg}}$ is not closed under inverses of morphisms.

3.2 Eilenberg's Lemma

Let $\mathcal{A}_L = \langle Q, A, \cdot, q_0, F\rangle$ be an automaton. For any subset P of Q, the *past* of P is the language denoted by $\mathsf{Past}(P)$ and defined by:

$$\mathsf{Past}(P) = \{w \in A^* \mid q_0 \cdot w \in P\}.$$

Dually, the *future* of a subset P of Q is the language denoted by $\mathsf{Fut}(P)$ and defined by:

$$\mathsf{Fut}(P) = \{w \in A^* \mid \exists p \in P, p \cdot w \in F\}.$$

It is well known that, an (accessible) automaton $\mathcal{A}$ is minimal if and only if the following condition holds:

$$\forall p, q \in Q\Big(p = q \Leftrightarrow \mathsf{Fut}(p) = \mathsf{Fut}(q)\Big). \tag{\textbf{M}}$$

In the next chapter, to prove Zero-One Theorem, we will use the following technical but important lemma.

Lemma 3.2.1 *Let $\mathcal{A}_L = \langle Q, A, \cdot, q_0, F\rangle$ be the minimal automaton of a language L. Then for any subset P of Q, its past $\mathsf{Past}(P)$ can be expressed as a finite Boolean combination of languages of the form Lw^{-1} where w in A^*.*

Proof: We only have to prove that, for any state q in Q, its past $\mathsf{Past}(q)$ can be expressed as a Boolean combination of languages of the form Lw^{-1}. Our goal is to prove the following equation with the usual conventions $\bigcap_{w\in\emptyset} Lw^{-1} = A^*$ and $\bigcup_{w\in\emptyset} Lw^{-1} = \emptyset$:

$$\mathsf{Past}(q) = \left(\bigcap_{w\in\mathsf{Fut}(q)} Lw^{-1}\right) \setminus \left(\bigcup_{w\notin\mathsf{Fut}(q)} Lw^{-1}\right). \tag{3.1}$$

The finiteness of this Boolean combination follows from Myhill-Nerode theorem.

We prove first that the left hand side is contained in the right hand side. Let v be a word in $\mathsf{Past}(q)$. If a word w in $\mathsf{Fut}(q)$, then vw in L by the definition, and hence v in Lw^{-1}. If a word w not in $\mathsf{Fut}(q)$, then vw not in L by the definition, and hence v not in Lw^{-1}. It follows that the left hand side is contained in the right hand side in Equation (3.1).

Then we prove that the right hand side is contained in the left hand side. Let v be a word in right hand side. Let p be the state satisfying $q_0 \cdot v = p$, that is, v is a word in $\mathsf{Past}(p)$. For any w in $\mathsf{Fut}(q)$, by the form of Equation (3.1), v is in Lw^{-1} from which we get vw in L whence $p \cdot w$ in F. That is, w also belongs to $\mathsf{Fut}(p)$. Conversely, for any w not in $\mathsf{Fut}(q)$, vw is not in L and thus v not in Lw^{-1}. That is, w does not belong to $\mathsf{Fut}(p)$. It follows that p and q have the same future $\mathsf{Fut}(p) = \mathsf{Fut}(q)$ from which we get $p = q$ by Condition (**M**) of the minimality of $\mathcal{A}_L$. Hence we obtain v in $\mathsf{Past}(q)$ and thus the right hand side is contained in the left hand side in Equation (3.1). ∎

3.3 Consequence of Eilenberg's Lemma for $\mathcal{ZO}^{\mathsf{Reg}}$

We will use the following lemma, which is a direct consequence of Lemma 3.2.1 and Proposition 3.1.1.

Lemma 3.3.1 *Let L be a regular language in $\mathcal{ZO}^{\mathsf{Reg}}$, $\mathcal{A}_L = \langle Q, A, \cdot, q_0, F\rangle$ be its minimal automaton. Then, for any subset P of Q, its past $\mathsf{Past}(P)$ is also in $\mathcal{ZO}^{\mathsf{Reg}}$.*

Proof: By Lemma 3.2.1, for any subset P of Q, its past $\mathsf{Past}(P)$ can be expressed as a finite Boolean combination of languages of the form Lw^{-1}. It follows that

$\mathsf{Past}(P)$ obeys the zero-one law, since L is in $\mathcal{ZO}^{\mathsf{Reg}}$ and $\mathcal{ZO}^{\mathsf{Reg}}$ is closed under Boolean operations and quotients by Corollary 3.1.1. ∎

3.4 Bibliographic Notes

Lemma 3.2.1 shows us an importance of the Boolean operations taken in tandem with quotients. While this lemma is known as a folklore (*cf.* [19]), which is an "automaton version" of a key lemma in Eilenberg's variety theorem, we have not found any literature that includes a complete proof. The proof given in this thesis is essentially based on "Proof of Theorem 3.2 and 3.2s" in Eilenberg's Volume B [18]. The original Eilenberg's lemma states about a monoid, not an automaton, as the following kind.

Lemma 3.4.1 (Eilenberg [18]) *Let M_L be the syntactic monoid of a regular language L over A, and let $\phi_L : A^* \to M_L$ be the syntactic morphism of L. Then for each element m of M_L, its inverse $\phi_L^{-1}(m)$ can be expressed as a finite Boolean combination of languages of the form $u^{-1}Lv^{-1}$ where u, v in A^*.*

Eilenberg used this lemma to prove his variety theorem: the existence of the one-to-one correspondence between varieties of languages and varieties of finite monoids.

Technically speaking, $\mathcal{ZO}^{\mathsf{Reg}}$ is *not a variety*. Recall that a variety of languages is a class of regular languages closed under Boolean operations, left and right quotients and inverses of morphisms. Proposition 3.1.2 shows that $\mathcal{ZO}^{\mathsf{Reg}}$ is not closed under inverses of morphisms. Since the work of Eilenberg, the theory have been extended several times by relaxing the definition of a variety of languages. Straubing [60] introduced the notion of $\mathcal{C}$*-varieties*: here $\mathcal{C}$ denotes some natural class of morphisms. A similar notion was introduced independently by Ésik and Ito [19]. The definition of a $\mathcal{C}$-variety of languages is similar to Eilenberg original definition except that it only requires closure under inverse images of morphisms belonging to $\mathcal{C}$. More recently, Gehrke et al. [23] proved that any *lattice of languages* (a class of regular languages closed under union and intersection) can be defined by a set of *profinite equations*, a result that subsumes Eilenberg's variety theorem. See [46, 61, 47] for more details.

CHAPTER 4

EQUIVALENCE OF $\mathcal{ZO}^{\mathbf{Reg}}$ AND $\mathcal{Z}^{\mathbf{Reg}}$

> The Variety Theorem provides the framework for talking about recognisable languages and finite monoids. It says that if you have a pseudovariety of finite monoids then there will be an associated variety of languages, although it will not tell you what these languages look like; that involves extra work and depends on the properties of the pseudovariety of monoids in question. Likewise, if you have a variety of languages, the Theorem tells us that there is an associated pseudovariety of finite monoids, but again it will not tell us what they look like; you have to do extra work.
>
> —Mark V. Lawson, "*Finite Automata*" [34].

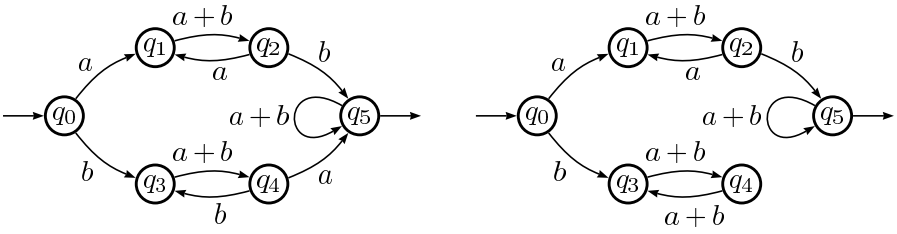
An automata theoretic proof of Zero-One Theorem is given in this chapter. In this chapter, we introduce two classes of automata: zero automata and quasi-zero automata. Zero automata plays major role in the proof of the difficult implication ④ ⇒ ①.

4.1 Zero Automata

Let $\mathcal{A}$ be an automaton $\langle Q, A, \cdot, q_0, F\rangle$. We write $p \to^* q$ if a state q is reachable from a state p. It is clear that the reachability relation $\to^*$ forms a *preorder over* Q, that is, a reflexive and transitive relation over Q. The equivalence relation $\leftrightarrow^*$ defined on Q by $p \leftrightarrow^* q$ if and only if $p \to^* q$ and $q \to^* p$ hold. A subset P of Q is called *strongly connected component* if every state q in P is reachable from every other state in P, i.e., , $p \leftrightarrow^* q$ holds for every p, q in P. A state q in Q is said to be *sink*, if $q \cdot a = q$ holds for every letter a in A. We say that a subset P of Q is *sink*, if it is strongly connected and there is no transition from any state p in P to a state which does not in P. That is, $Q \setminus P$ are not reachable from P. The family of all sink components of $\mathcal{A}$ is denoted by $\mathsf{Sink}(\mathcal{A})$. A sink component P is *trivial* if it consists of some single state $P = \{p\}$. We shall identify a singleton $\{p\}$ with its unique element p. Sink components can be regarded as *maximal equivalence classes with respect to* $\leftrightarrow^*$ *over* Q. That is, if P is sink, then P is strongly connected and $p \to^* q$ implies q in P for every p in P and q in Q. Note that, since every finite set equipped with a preorder has at least one maximal equivalence class, every (complete) automaton has at least one sink component. One can easily verify that, for every state q, there exists a sink component that is reachable from q. A word w is a *synchronising word of* $\mathcal{A}$ if, there exists a certain state q in Q, $p \cdot w = q$ holds for every state p in Q. That is, w is the *constant map* from Q to q. We call an automaton *synchronising* if it has a synchronising word. Note that any synchronising automaton has at most one sink state. As we will prove in the next section, the following class of automata captures precisely the zero-one law for regular languages.

Definition 4.1.1 ([49]) A *zero automaton* is a synchronising automaton with a sink state.

EXAMPLE 4.1



(a) Zero automaton $\mathcal{A}_0$ (b) Non-zero automaton $\mathcal{A}_1$

Figure 4.1 Zero and non-zero automata

Consider two automata $\mathcal{A}_0$ and $\mathcal{A}_1$ illustrated in Figure 4.1. $\mathcal{A}_0$ is a zero automaton but $\mathcal{A}_1$ is not, though both automata have a sink state q_5. A synchronisation word of $\mathcal{A}_0$ is $aabb$: one can easily verify that $q_i \cdot aabb = q_5$ for every

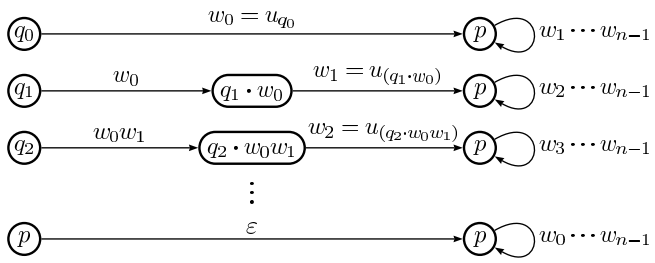


Figure 4.2 Synchronising word $v_{n-1} = w_0 \cdots w_{n-1}$ in the proof of Lemma 4.1.1

state q_i. It is clear that $\mathcal{A}_1$ in Figure 4.1 does not have a synchronising word since it has two sink components. The only difference between $\mathcal{A}_0$ and $\mathcal{A}_1$ is the transition result of $q_4 \cdot a$; which equals to q_5 in $\mathcal{A}_0$, while which equals to q_3 in $\mathcal{A}_1$. We can easily verify that, $\mathcal{A}_0$ has a unique sink component q_5, while $\mathcal{A}_1$ has two sink components $\{q_3, q_4\}$ and q_5.

The definition of zero automata can be rephrased as follows.

Lemma 4.1.1 *Let $\mathcal{A} = \langle Q, A, \cdot, q_0, F\rangle$ be an automaton. Then $\mathcal{A}$ is zero if and only if $\mathcal{A}$ has a unique sink component and it is trivial, i.e., $\mathsf{Sink}(\mathcal{A}) = \{\{p\}\}$ for a certain sink state p.*

Proof: First we assume $\mathcal{A}$ is zero with a sink state p. Then there exists a synchronising word w and it clearly satisfies $q \cdot w = p$ for each q in Q since p is sink. This shows that there is no sink component in $Q \setminus p$.

Now we prove the converse direction, we assume $\mathcal{A}$ has a unique sink component and it is trivial, say p. We can verify that for every state q in Q, there exists a word w in A^*, such that $q \cdot w = p$. Indeed, if there does not exist such word w for some q, then the set of all reachable states from q : $\{r \in Q \mid \exists w \in A^*, q \cdot w = r\}$ must contains at least one sink component which does not contain p. This contradicts with the uniqueness of the sink component p in $\mathcal{A}$. The existence of a synchronising word w is guaranteed, because we can concretely construct it as follows. Let n be the number of states $n = |Q|$ and let $Q = \{q_0, \cdots, q_{n-1} = p\}$. We define a word sequence w_i inductively by $w_0 = u_{q_0}$ and $w_i = u_{(q_i \cdot v_{i-1})}$ where each u_{q_i} is a shortest word satisfying $q_i \cdot u_{q_i} = p$, and v_{i-1} is the word of the form $w_0 \cdots w_{i-1}$. As shown in Figure 4.2, we can easily verify that the word $v_{n-1} = w_0 \cdots w_{n-1}$ is a synchronising word satisfying $q \cdot v_{n-1} = p$ for each q in Q.

For example, consider the zero automaton $\mathcal{A}_0$ in Figure 4.1. Then each u_{q_i}, w_{q_i} and v_{q_i} are defined as follows.

	u_{q_i}	w_{q_i}	v_{q_i}
q_0	aab	aab	aab
q_1	ab	b	$aabb$
q_2	b	ε	$aabb$
q_3	aa	ε	$aabb$
q_4	a	ε	$aabb$

The obtained word $v_{q_4} = aabb$ is a synchronising word of $\mathcal{A}_0$. ∎

4.2 Proof of Zero-One Theorem (1)

We show the implication ① ⇒ ② ⇒ ③ ⇒ ④ ⇒ ①. The former implication ① ⇒ ② ⇒ ③ ⇒ ④ is easy, but we include a complete proof here to be self-contained.

4.2.1 ① ⇒ ② ($\mathcal{A}_L$ is zero ⇒ L is with zero)

Let $\mathcal{A}_L = \langle Q, A, \cdot, q_0, F\rangle$ be the minimal automaton of L and assume that $\mathcal{A}_L$ is zero with a sink state p. Let M be the transition monoid of $\mathcal{A}_L$ and $\phi : A^* \to M$ be the syntactic morphism of L. Then we can verify that M has a zero element $\mathbf{0}$ as the transformation $\mathbf{0} : q \mapsto p$ for all q in Q, that is, $\mathbf{0}$ is the constant map from Q to p. The existence of $\mathbf{0}$ is guaranteed since $\mathcal{A}_L$ is synchronising. Indeed, for any synchronising word w, $\phi(w) = \mathbf{0}$ holds. One can easily verify that $m\mathbf{0} = \mathbf{0}m = \mathbf{0}$ for all m in M. This proves that M the syntactic monoid of L has the zero. ■

4.2.2 ② ⇒ ③ (L is with zero ⇒ L or $\overline{L}$ contains an ideal language)

Let L be a regular language in $\mathcal{Z}^{\mathsf{Reg}}$, M be its syntactic monoid with a zero element $\mathbf{0}$ and $\phi : A^* \to M$ be its syntactic morphism. Choose a word $w_{\mathbf{0}}$ from the preimage of $\mathbf{0}$: $w_{\mathbf{0}} \in \phi^{-1}(\mathbf{0})$. Note that the word $w_{\mathbf{0}}$ is always exists by the definition of the syntactic monoid.

Now we prove that L contains the ideal language $A^*w_{\mathbf{0}}A^*$ if $w_{\mathbf{0}}$ is in L. By the definition of zero, we have

$$\phi(xw_{\mathbf{0}}y) = \phi(x)\phi(w_{\mathbf{0}})\phi(y) = \phi(x)\mathbf{0}\phi(y) = \mathbf{0}$$

for any words x, y in A^*. That is, if w contains $w_{\mathbf{0}}$ as a factor, then $\phi(w) = \phi(w_{\mathbf{0}}) = \mathbf{0}$ holds and hence w also in L. This implies that the language of the form $A^*w_{\mathbf{0}}A^*$, the set of all words that contains $w_{\mathbf{0}}$ as a factor, is contained in L. Dually, we can prove that $\overline{L}$ contains $A^*w_{\mathbf{0}}A^*$ if $w_{\mathbf{0}}$ is not in L. ■

4.2.3 ③ ⇒ ④ (L or $\overline{L}$ contains an ideal language ⇒ L obeys the zero-one law)

We assume that L contains A^*wA^* for some word w. The probability $\mu_n(A^*wA^*)$ is nothing but the probability that a randomly chosen word of length n contains w as a factor. The *infinite monkey theorem* (*cf.* Note I.35 in [21]), sometimes called *Borges's theorem*, ensures that $\mu_n(A^*wA^*)$ tends to one if n tends to infinity.

Infinite Monkey Theorem. *Take any fixed finite set* Π *of words in* A^*. *A random word in* A^* *of length* n *contains all the words of the set* Π *as factors with probability tending to one exponentially fast as* n *tends to infinity.*

This and the monotonicity of μ shows $\mu(L) = \mu(A^*wA^*) = 1$. Conversely, if the complement $\overline{L}$ contains A^*wA^*, one can easily verify that $\mu(L) = 1 - \mu(\overline{L}) = 0$. ■

4.2.4 ④ ⇒ ① (L obeys the zero-one law ⇒ $\mathcal{A}_L$ is zero)

Let L be a regular language in $\mathcal{ZO}^{\mathsf{Reg}}$ and $\mathcal{A}_L = \langle Q, A, \cdot, q_0, F\rangle$ be its minimal automaton, let $\mathsf{Sink}(\mathcal{A}_L) = \{P_1, \cdots, P_k\}$ for some $k \geq 1$. Our goal is to prove $k = 1$ and $\mathsf{Sink}(\mathcal{A}_L) = \{\{p\}\}$ for a certain sink state p. It follows that $\mathcal{A}_L$ is zero by Lemma 4.1.1.

For any sink component P_i, there exists a word w_i such that $q_0 \cdot w_i$ in P_i because $\mathcal{A}_L$ is accessible. Lemma 3.1.1 and $\mu(A^*) = 1$ implies:

$$\mu(w_i A^*) = |A|^{-|w_i|}\mu(A^*) = |A|^{-|w_i|} > 0. \tag{4.1}$$

Since P_i is sink, the language $w_i A^*$ is contained in $\mathsf{Past}(P_i)$. For each $\mathsf{Past}(P_i)$ has the asymptotic probability and it is either zero or one by Lemma 3.3.1. The monotonicity of μ and Inequation (4.1) imply:

$$\mu(\mathsf{Past}(P_i)) = 1 \tag{4.2}$$

holds for every sink component P_i.

Now we prove $k = 1$. By Equation (4.2), we can easily verify that

$$\mu\left(\bigcup_{i=1}^{k} \mathsf{Past}(P_i)\right) = \sum_{i=1}^{k} \mu(\mathsf{Past}(P_i)) = k$$

holds because $\mathcal{A}_L$ is deterministic and thus all $\mathsf{Past}(P_i)$ are mutually disjoint. This clearly shows $k = 1$, that is, there exists a unique sink component, say P, in $\mathcal{A}_L$: $\mathsf{Sink}(\mathcal{A}_L) = \{P\}$.

Next we let $P = \{p_1, \cdots, p_n\}$ and prove $n = 1$. Since P satisfies $\mu(\mathsf{Past}(P)) = 1$ by Equation (4.2), there exists exactly one state p in P that satisfies $\mu(\mathsf{Past}(p)) = 1$ by Lemma 3.3.1. Further, because P is strongly connected, for every state p_i in P, there exists a word w_i such that $p \cdot w_i = p_i$ and thus $\mathsf{Past}(p_i)$ contains $\mathsf{Past}(p)w_i$. Lemma 3.1.1 and $\mu(\mathsf{Past}(p)) = 1$ implies:

$$\mu(\mathsf{Past}(p)w_i) = |A|^{-|w_i|}\mu(\mathsf{Past}(p)) = |A|^{-|w_i|} > 0. \tag{4.3}$$

Each $\mathsf{Past}(p_i)$ has the asymptotic probability and it is either zero or one by Lemma 3.3.1. The monotonicity of μ and Inequation (4.3) imply:

$$\mu(\mathsf{Past}(p_i)) = 1 \tag{4.4}$$

holds for every p_i in P. From Equation (4.4), we obtain:

$$\mu(\mathsf{Past}(P)) = \sum_{i=1}^{n} \mu(\mathsf{Past}(p_i)) = \sum_{i=1}^{n} 1 = n = 1,$$

because $\mathcal{A}_L$ is deterministic and thus all $\mathsf{Past}(p_i)$ are mutually disjoint. We now obtain $n = 1$, that is, P is singleton and hence $\mathsf{Sink}(\mathcal{A}_L) = \{p\}$. That is, $\mathcal{A}_L$ is zero. ■

4.3 Quasi-Zero Automata

Let $\mathcal{A} = \langle Q, A, \cdot, q_0, F\rangle$ be an automaton. The *Nerode equivalence* $\sim$ of $\mathcal{A}$ is the relation defined on Q by $p \sim q$ if and only if $\mathsf{Fut}(p) = \mathsf{Fut}(q)$. One can easily verify that $\sim$ is actually a congruence, in the sense that $p \sim q$ implies $p \cdot w \sim q \cdot w$ for all $w \in A^*$. Hence it follows that there is a well defined new automaton $\mathcal{A}/\!\sim$, *the quotient automaton of* $\mathcal{A}$:

$$\mathcal{A}/\!\sim = \langle Q/\!\sim, A, \cdot, [\widetilde{q_0}], F/\!\sim\rangle$$

where $[\widetilde{q}]$ is the equivalence class modulo $\sim$ of q, $S/\!\sim = \{[\widetilde{q}] \mid q \in S\}$ is the set of the equivalence classes modulo $\sim$ of a subset $S \subseteq Q$, and where the transition function $\cdot : Q/\!\sim \times A \to Q/\!\sim$ is defined by $[\widetilde{p}] \cdot a = [\widetilde{p \cdot a}]$. We define the natural mapping $\widetilde{\phi} : Q \to Q/\!\sim$ by $\widetilde{\phi}(q) = [\widetilde{q}]$. Condition (**M**) for minimal automata implies that, for any automaton $\mathcal{A}$, its quotient automaton $\mathcal{A}/\!\sim$ is the minimal automaton of $L(\mathcal{A})$. We shall identify the quotient automaton $\mathcal{A}/\!\sim$ with the minimal automaton of $L(\mathcal{A})$ (*cf.* [50]).

We now introduce a new class of automata which is a generalisation of the class of zero automata.

Definition 4.3.1 (quasi-zero automaton) An automaton $\mathcal{A} = \langle Q, A, \cdot, q_0, F\rangle$ is *quasi-zero* if either $\bigcup \mathsf{Sink}(\mathcal{A}) \subseteq F$ or $\bigcup \mathsf{Sink}(\mathcal{A}) \cap F = \emptyset$ holds.

Since every zero automaton $\mathcal{A}$ satisfies $\bigcup \mathsf{Sink}(\mathcal{A}) = \{p\}$ for a certain state p (Lemma 4.1.1), every zero automaton is quasi-zero.

Before proving the equivalence ① $\Leftrightarrow$ ⑤ in Zero-One Theorem, we introduce the following lemma.

Lemma 4.3.1 *Let $\mathcal{A} = \langle Q, A, \cdot, q_0, F\rangle$ be an automaton and $\mathcal{A}/\!\sim$ be its quotient automaton. Then the following hold:*

1. *For any sink component P in $\mathcal{A}$, $P/\!\sim$ is also a sink component in $\mathcal{A}/\!\sim$.*
2. *For any sink component R in $\mathcal{A}/\!\sim$, there is at least one sink component P in $\mathcal{A}$ satisfying $P/\!\sim = R$.*

Proof: (1) Let P be a sink component in $\mathcal{A}$. Since P is strongly connected, for each pair of states $[\widetilde{p}], [\widetilde{q}]$ in $P/\!\sim$, there exists a word w satisfying $p \cdot w = q$ and hence $[\widetilde{p}] \cdot w = [\widetilde{p \cdot w}] = [\widetilde{q}]$. That is, $[\widetilde{p}] \to^* [\widetilde{q}]$ holds and one can easily verify that $[\widetilde{q}] \to^* [\widetilde{p}]$ also holds. This shows that $P/\!\sim$ is strongly connected. Moreover, for any $[\widetilde{p}]$ in $P/\!\sim$ and for any word w, $[\widetilde{p}] \cdot w = [\widetilde{p \cdot w}]$ is also in $P/\!\sim$ because P is sink and $p \cdot w$ is in P. That is, $P/\!\sim$ is sink component in $\mathcal{A}/\!\sim$.

(2) Let R be a sink component in $\mathcal{A}/\!\sim$ and S be its preimage $S = \widetilde{\phi}^{-1}(R)$. Clearly, for any state s in S and for any word w, $s \cdot w$ is in S since $[\widetilde{s \cdot w}] = [\widetilde{s}] \cdot w$ is in R. Since every finite set equipped with a preorder has at least one maximal equivalence class, S has at least one sink component, say P. Let p be a state in P. Since R is strongly connected, for any state in $[\widetilde{r}]$ in R, there exists a word w such that $[\widetilde{p}] \cdot w = [\widetilde{p \cdot w}] = [\widetilde{r}]$. This shows that for every state $[\widetilde{r}]$ in R, there exists a state $p \cdot w$ in P for some w because P is sink. That is, $P/\!\sim = R$. ∎

4.4 Proof of Zero-One Theorem (2)

The following proposition shows that the minimal automaton of any quasi-zero automaton is zero and *vice versa* (this justifies the term "quasi-zero"). This proposition shows exactly the equivalence ① ⇔ ⑤ in Zero-One Theorem.

Proposition 4.4.1 *An automaton $\mathcal{A} = \langle Q, A, \cdot, q_0, F\rangle$ is quasi-zero if and only if $\mathcal{A}/\!\sim$ is zero.*

4.4.1 ① ⇒ ⑤ ($\mathcal{A}/\!\sim$ is zero ⇒ $\mathcal{A}$ is quasi-zero)

Let p be the unique sink state of $\mathcal{A}/\!\sim$. To prove this direction, it is enough to consider the case when $p \in F/\!\sim$, i.e., $\mathsf{Fut}(p) = A^*$. We now show

$$\bigcup \mathsf{Sink}(\mathcal{A}) \subseteq F \tag{4.5}$$

by contradiction. Let us assume that Inclusion (4.5) does not hold, that is, we assume there exists a non-final state q in $\bigcup \mathsf{Sink}(\mathcal{A})$. Let P be the sink component of $\mathcal{A}$ that contains q. Since P is sink, $\widetilde{\phi}(P)$ is also sink in $\mathcal{A}/\!\sim$ by Lemma 4.3.1. Moreover, $\widetilde{\phi}(P)$ does not contain the sink state p, because $q \notin F$ implies that, for any state q' in P, $\mathsf{Fut}(q') \neq A^*$ from which we obtain $\mathsf{Fut}([\widetilde{q'}]) \neq \mathsf{Fut}(p)$ and $[\widetilde{q'}] \neq p$. That is, $\mathcal{A}/\!\sim$ has at least two sink components $\widetilde{\phi}(P)$ and p. This is contradiction. ∎

4.4.2 ⑤ ⇒ ① ($\mathcal{A}$ is quasi-zero ⇒ $\mathcal{A}/\!\sim$ is zero)

To prove this direction, it is enough to consider the case when $\bigcup \mathsf{Sink}(\mathcal{A}) \subseteq F$. Since $\mathcal{A}$ is quasi-zero, all states in $\bigcup \mathsf{Sink}(\mathcal{A})$ have the same future A^*, i.e., $\mathsf{Fut}(q) = A^*$ for every state q in $\bigcup \mathsf{Sink}(\mathcal{A})$, because $\bigcup \mathsf{Sink}(\mathcal{A}) \subseteq F$ implies $q \cdot w \in F$ for every state q in $\bigcup \mathsf{Sink}(\mathcal{A})$ and every word w. This implies that $\big(\bigcup \mathsf{Sink}(\mathcal{A})\big)/\!\sim$ consists of a single equivalence class, say p. Moreover, this equivalence class p is a sink state in $\mathcal{A}/\!\sim$ by the definition of sink and Condition (**M**) of the minimality of $\mathcal{A}/\!\sim$. We now show that, by contradiction, $\mathcal{A}/\!\sim$ has only one sink component p:

$$\bigcup \mathsf{Sink}(\mathcal{A}/\!\sim) = \{p\} \tag{4.6}$$

from which we obtain $\mathcal{A}/\!\sim$ is zero by Lemma 4.1.1. Let us assume that Equation (4.6) does not hold, that is, we assume there exists another sink component R in $\mathcal{A}/\!\sim$ that does not contain p. By Lemma 4.3.1, there exists a sink component P in $\mathcal{A}$ such that $P/\!\sim = R$. This implies that $P \not\subseteq F$ because R does not contain p. This is contradicts with the assumption $\bigcup \mathsf{Sink}(\mathcal{A}) \subseteq F$. This completes the proof of Zero-One Theorem. ∎

4.5 Bibliographic Notes

From the proof in this chapter, we can obtain the followings as a corollary.

Corollary 4.5.1 *Let L be a regular language and $\mathcal{A}_L$ be the minimal automaton of L. Then the following four conditions are equivalent.*

1. *L is almost full.*
2. *L contains an ideal language.*
3. *$\mathcal{A}_L$ is zero and its sink state is final.*
4. *L is recognised by a quasi-zero automaton $\mathcal{A}$ such that all states in $\bigcup \mathsf{Sink}(\mathcal{A})$ are final.*

The direction ③ $\Rightarrow$ ④ of Zero-One Theorem is nothing but the well known Infinite Monkey Theorem, as we proved in Section 4.2.3. The remarkable fact of this theorem is that its converse ④ $\Rightarrow$ ③ is also true. Things are, however, getting more complicated if we consider beyond regular languages. There exist several simple counterexamples that imply $\mathcal{ZO} \neq \mathcal{Z}$ and we will explain such languages in Chapter 6.

In contrast to the class of monoids with zero, their natural counterpart, the class of zero automata has not been given much attention. To the best of our knowledge, only few studies (e.g., [49]) have investigated zero automata in the context of the theory of synchronising word for *Černý's conjecture*.

CHAPTER 5

ALGORITHMIC AND LOGICAL ASPECTS OF $\mathcal{ZO}^{\mathbf{Reg}}$

> There are many brilliant surveys on formal language theory. Quite many surveys cover first-order and monadic second-order definability. But there are also nuggets below. There are deep theorems on proper fragments of first-order definability.
>
> —Diekert et al., "*A Survey on Small Fragments of First-Order Logic over Finite Words*" [15].

In this chapter, we describes a linear time algorithm for testing whether a given regular language is zero-one if it is given by an n-states automaton (recall that all automata considered in the thesis are deterministic). Some logical aspects of the zero-one law for regular languages are also investigated.

5.1 Linear Time Algorithm for Testing Membership

The equivalence of zero-automata and the zero-one law gives us an effective algorithm. For a given n-states automaton $\mathcal{A}$, we can determine whether $L(\mathcal{A})$ obeys the zero-one law by the following steps: (i) Minimise $\mathcal{A}$ to obtain its minimal automaton $\mathcal{A}/\!\sim$. (ii) Calculate the family of all strongly connected components P of $\mathcal{A}/\!\sim$. (iii) Check whether P contains exactly one strongly connected sink component and it is trivial, i.e., whether $\mathcal{A}/\!\sim$ is a zero automaton (Lemma 4.1.1). It is well known that Hopcroft's automaton minimisation algorithm has an $\mathrm{O}(n \log n)$ time complexity and Tarjan's strongly connected components algorithm has an $\mathrm{O}(n + n|A|) = \mathrm{O}(n)$ complexity where $n|A|$ means the number of *edges*. Hence we can minimise $\mathcal{A}$ to obtain $\mathcal{A}/\!\sim$ in $\mathrm{O}(n \log n)$ on the step (i), and can calculate P in $\mathrm{O}(n)$ on the step (ii). One can easily verify that the step (iii) above can be done in $\mathrm{O}(n)$. To sum up, we have an $\mathrm{O}(n \log n)$ algorithm for testing whether a given regular language obeys the zero-one law.

We can obtain, however, more efficient algorithm *by avoiding minimisation*. Quasi-zero automata gives us more effective algorithm.

Theorem 5.1.1 *There is an $\mathrm{O}(n)$ algorithm for testing whether a given regular language is zero-one, if its is given by an n-states automaton.*

Proof: For a given n-states automaton $\mathcal{A}$, we can determine whether $L(\mathcal{A})$ obeys the zero-one law by the following steps: (i) Calculate the family of all strongly connected components P of $\mathcal{A}$. (ii) Extract all strongly connected sink components from P to obtain $\mathsf{Sink}(\mathcal{A})$. (iii) Check whether, in $\bigcup \mathsf{Sink}(\mathcal{A})$, either all states are final or all states are non-final, i.e., whether $\mathcal{A}$ is quasi-zero. By Zero-One Theorem, $L(\mathcal{A})$ obeys the zero-one law if and only if $\mathcal{A}$ is quasi-zero. Hence this algorithm is correct. All steps (i) $\sim$ (iii) can be done in $\mathrm{O}(n)$, this ends the proof. ∎

5.2 Logical Fragments over Finite Words

We denote by $\mathrm{MSO}[<]$ *monadic second-order logic over finite words* and denote by $\mathrm{FO}[<]$ *first-order logic over finite words*. We can interpret words as logical structures with a linear order composed of a sequence of positions labeled over a finite alphabet A, $<$ denotes the linear order over the natural numbers. Given a word $w = a_0 a_1 \cdots a_n$ in A^* where each a_i is a letter, we define the *structure* $M_w = \langle U, <, (P_a)_{a \in A} \rangle$ *of* w as follows: the *universe* U is $\{0, 1, \cdots, n\}$ which corresponds to *positions* in the word, $<$ the usual linear order on the natural numbers, and the *unary predicate* P_a of a letter a in A is defined as:

$$P_a(i) \text{ is true } \Leftrightarrow a_i = a.$$

We shall identify each predicate P_a as the set of positions $P_a = \{i \in U \mid a_i = a\}$. For a logical sentence Φ of some logic $\mathcal{L}$, we denote by $L(\Phi)$ the *language defined by* Φ:

$$L(\Phi) = \{w \in A^* \mid M_w \models \Phi\}.$$

We say that a language L *is definable in a logic* $\mathcal{L}$ if, there exists a sentence Φ of $\mathcal{L}$ such that $L(\Phi) = L$.

EXAMPLE 5.1

The structure of a word $w = abaab$ is defined by

$$M_w = \langle \{1, 2, 3, 4, 5\}, <, P_a, P_b \rangle$$

where $<$ is the usual ordering, and $P_a[P_b]$ contain positions in w where $a[b]$ occurs: that is, $P_a = \{1, 3, 4\}$ and $P_b = \{2, 5\}$.

We can easily observe that first-order logic over finite words $\mathrm{FO}[<]$ does not have the zero-one law.

EXAMPLE 5.2

A simple counterexample is the language aA^* which can be defined by the $\mathrm{FO}[<]$ sentence $\Phi_{aA^*} = \exists i\big(P_a(i) \wedge \forall j(i \leq j)\big)$. aA^* satisfies $\mu_n(aA^*) = 1/|A|$ as we stated in Example 2.2, hence Φ_{aA^*} does not obey the zero-one law in general. It follows that $\mathrm{FO}[<]$ does not have the zero-one law.

We summarise well-known logical and algebraic characterisations of classes of languages, including the class of zero-one languages $\mathcal{ZO}^{\mathrm{Reg}}$, in Table 5.1. We use standard abridged notation for the following first-order fragments over finite words:

- $\mathrm{FO}^n[<]$ for first-order logic with distinct n variables;
- $\Sigma_n[<]$ for FO formulas with n blocks of quantifiers and starting with a block of existential quantifiers;
- $\mathbb{B}\Sigma_n[<]$ for the Boolean closure of $\Sigma_n[<]$.

Details and full proofs of these results can be found in a very nice survey [15] by Diekert et al. A *monomial* over A is a language of the form $A_0^* a_1 A_1^* a_2 \cdots a_k A_k^*$ where a_i in A and A_i is a subset of A for each i. A monomial $A_0^* a_1 A_1^* a_2 \cdots a_k A_k^*$ is *unambiguous* if for all $w \in A_0^* a_1 A_1^* a_2 \cdots a_k A_k^*$ there exists exactly one factorisation $w = w_0 a_1 w_1 a_w \cdots a_k w_k$ with w_i in A_i^* for each i. A language L over A is called:

- *star-free* if it is expressible by union, concatenation and complement, but does not use Kleene star;
- *polynomial* if it is a finite union of monomials;
- *unambiguous polynomial* if it is a finite disjoint union of unambiguous monomials;
- *simple polynomial* if it is a finite union of languages of the form $A^* a_1 A^* a_2 \cdots a_k A^*$.
- *piecewise testable* if it is a finite Boolean combination of simple polynomials;

Table 5.1 Language Hierarchy

Languages	Monoids	Logic
regular	finite	$\mathrm{MSO}[<]$
star-free	*aperiodic*	$\mathrm{FO}[<]$
polynomials		$\Sigma_2[<]$
unambiguous polynomials	**DA**	$\mathrm{FO}^2[<]$
zero-one	zero	?
piecewise testable	$\mathcal{J}$-trivial	$\mathbb{B}\Sigma_1[<]$
simple polynomial		$\Sigma_1[<]$
$\mathbb{B}\{A^* \mid A \subseteq \Sigma\}$	*semilattice*	$\mathrm{FO}^1[<]$

The question then arises as to *which fragments of* $\mathrm{FO}[<]$ *over finite words have the zero-one law*. The algebraic characterisation of the zero-one law partially answers this question. Since every $\mathcal{J}$*-trivial* syntactic monoid has a zero element (*cf.* [46]), Zero-One Theorem leads to the following corollary.

Corollary 5.2.1 *The Boolean closure of existential first-order logic over finite words has the zero-one law.*

One can easily verify that the sentence Φ_{aA^*} in example 5.2, which only uses two variables i and j, is in $\mathrm{FO}^2[<]$. It follows that $\mathrm{FO}^2[<]$ does not have the zero-one law, hence Corollary 5.2.1 shows us a "separation" between $\mathrm{FO}^2[<]$ and $\mathbb{B}\Sigma_1[<]$. It must be noted that the class of zero-one languages $\mathcal{ZO}^{\mathsf{Reg}}$ and unambiguous polynomials are incomparable. To take a simple example, consider two languages $(aa)^*$ and aA^* over $A = \{a, b\}$. The language $(aa)^*$ is zero-one but not unambiguous polynomial since its syntactic monoid is not *aperiodic* (i.e., having no nontrivial subgroup). Conversely, aA^* is not zero-one but unambiguous polynomial since it is definable in $\mathrm{FO}^2[<]$ as we have stated in Example 5.2. An interesting open problem is whether there exists a logical fragment that exactly captures the zero-one law (Figure 5.1).

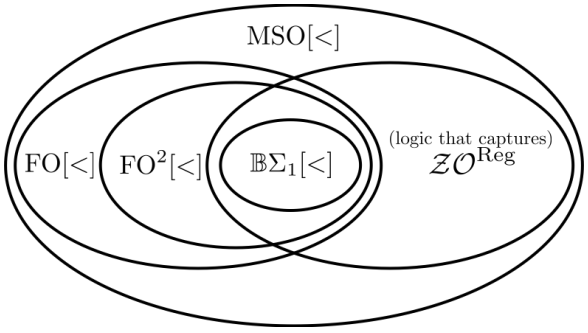


Figure 5.1 Logical fragments and $\mathcal{ZO}^{\mathsf{Reg}}$

5.3 Bibliographic Notes

Zero-one law for finite graphs That first-order logic has the zero-one law was proved first by Glebskii et al. in 1969 [24], and independently by Fagin in 1976 [20]. As Kolaitis and Vardi put it: " In the past, 0-1 laws for various logics $\mathcal{L}$ were proved by establishing first a transfer theorem for L of the following kind:

> There is a certain infinite structure $\mathcal{R}$ over the vocabulary σ such that for any property P expressible in $\mathcal{L}$ we have:
>
> $$\mathcal{R} \models P \Leftrightarrow P \text{ is almost surely true.}$$

This method was discovered by Fagin (1976) in his proof of the zero-one law for first-order logic on finite structures." And such infinite structure $\mathcal{R}$ is called the *random structure*. There are other known methods for proving the zero-one law: the *quantifier elimination* (i.e., every formula with just one quantifier in front of it is almost everywhere equivalent to a quantifier-free formula) and *game theoretic approaches* (e.g., *pebble games*). [32]. These three methods are rely on the *extension axioms* introduced by Gaifman [22]. Blass, Gurevich, and Kozen [7], and independently, Talanov and Knyazev [62] proved that *first-order logic with a fixed point operator* has the zero-one law. Kolaitis and Vardi [32] gave three different proofs for the zero-one law for *finite variable infinitary first-order logic*; the first proof is by the transfer theorem, the second proof is by the quantifier elimination, and the third proof is by the pebble games. The study of the zero-one law for fragments of *existential second-order logic* was initiated by Kolaitis and Vardi [31] and they provide a survey on this topic [33]. We refer to Chapter 12 of the book [35] by Libkin for more details.

Zero-one law and logical fragments over finite words
Ehrenfeucht has shown that *first-order logic with linear order* has the *convergence law*: every definable property has an asymptotic probability (the proof can be found in Lynch [36]). Lynch [37] proved that *first-order logic with unary functions* also has the convergence law.

The connection between logic and languages firstly discovered by Büchi in 1960 [11]. He gave an effective transformations of $\mathrm{MSO}[<]$ sentences into finite automata and vice versa. This shows that the definability in $\mathrm{MSO}[<]$ captures exactly the class of regular languages. Since the work of Bühi, many connections between logic and languages have been shown as we summarised in Table 5.1. Some results about the zero-one law for finite words considered in this thesis is also given by Lynch [38]. He proved that first-order logic over finite words has the convergence law, that is, in our terms, $\mu(L)$ exists for every first-order definable language L. Moreover, he proved that monadic second-order logic over finite words has the following *weak convergence law*: for every regular language L, there is a positive integer a such that for all non-negative integer $b < a$:

$$\lim_{n\to\infty} \mu_{an+b}(L)$$

exists. Lynch uses the game theoretic approach (*Ehrenfeucht-Fraïssé game*) and Markov chains [38]. The second result is related to the previous result by Berstel [4], Salomaa and Soittola [51]: $\mu_n(L)$ has finitely many accumulation points and each accumulation point is rational for any regular language L. We refer the reader to Compton's comprehensive survey about zero-one laws for various logics and structures [13] for more history on this topic.

CHAPTER 6

BEYOND REGULAR LANGUAGES

> Concevons qu'on ait dressé un million de singes á frapper au hasard sur les touches d'une machine á ècrire et que, sous la surveillance de contremaîtres illettrés, ces singes dactylographes travaillent avec ardeur dix heures par jour avec un million de machines à écrire de types variés. Les contremaîtres illettrés rassembleraient les feuilles noircies et les relieraient en volumes. Et au bout d'un an, ces volumes se trouveraient renfermer la copie exacte des livres de toute nature et de toutes langues conservés dans les plus riches bibliothéques du monde. Telle est la probabilité pour qu'il se produise pendant un instant très court, dans un espace de quelque étendue, un écart notable de ce que la mècanique statistique considère comme la phénomène le plus probable.
>
> —Émile Borel, "*La mécanique statistique et l'irréversibilité*".

The implication ③ $\Rightarrow$ ④ of Zero-One Theorem is nothing but the well-known Infinite Monkey Theorem. In general, it is very difficult to extend some result about regular languages into beyond regular languages. Many deep results in the theory of regular languages, of course, heavily depend on the regularity of regular languages. *Zero-One Theorem is not true beyond regular languages*. Some counterexamples are given in Section 6.2. These languages obey the zero-one law but are not with zero. This implies that $\mathcal{ZO}$ properly contains $\mathcal{Z}$, while $\mathcal{ZO}^{\mathsf{Reg}}$ coincides with $\mathcal{Z}^{\mathsf{Reg}}$.

6.1 Zero-One Theorem for Proving Non-Regularity

First of all, we prove that $\mathcal{Z}$ is contained in $\mathcal{ZO}$. This is easy and the following proposition is folklore (*cf.* [61]), but we include the proof for self-containedness.

Proposition 6.1.1 *A language L over A is with zero if and only if L or $\overline{L}$ contains an ideal language.*

Proof: The "only if" part is what we exactly proved in Section 4.2.2. Note that we did not use any assumption of the regularity of L in Section 4.2.2. Now we prove the "if" part. Let M_L be the syntactic monoid of L and ϕ_L be the syntactic morphism of L. We can assume that L contains an ideal language, say A^*wA^* for some word w, without loss of generality. Then, for any v, x, y in A^*, xwy, $xvwy$ and $xwvy$ are obviously in A^*wA^*. Hence w, vw and wv are all equivalent on the syntactic congruence of L: $\phi_L(vw) = \phi_L(wv) = \phi_L(w)$. Because ϕ_L is surjective, we obtain the following equation for all x in M_L:

$$x\phi_L(w) = \phi_L(w)x = \phi_L(w).$$

This implies that $\phi_L(w)$ is a zero element of M_L. ∎

Proposition 6.1.1 and Infinite Monkey Theorem immediately imply the following.

Corollary 6.1.1 *$\mathcal{ZO}$ contains $\mathcal{Z}$.*

Zero-One Theorem implies that if L is in $\mathcal{ZO}$ but not in $\mathcal{Z}$, then L is not regular. Before proving the non-regularity of the set of all palindromes and the Dyck language, we sum up the above discussion in the following lemmata. Recall that a word w is forbidden [admissible] for a language L if $A^*wA^* \cap L = \emptyset$ [$A^*wA^* \subseteq L$] holds.

Lemma 6.1.1 (Zero Lemma) *Let L be an almost empty language over A. If L does not have a forbidden word, then L is not regular.*

Proof: From the assumption $\mu(L) = 0$, we can easily verify that L does not contain an ideal language by Infinite Monkey Theorem. Assume that L does not have a forbidden word. Then for any word w not in L, the ideal language A^*wA^* is not disjoint from L: $A^*wA^* \cap L \neq \emptyset$. Hence every ideal language is not contained in $\overline{L}$. That is, if L is almost empty and does not have a forbidden word, then L is not with zero. Since L is in $\mathcal{ZO}$ but not in $\mathcal{Z}$, L is not regular by Zero-One Theorem. ∎

Corollary 6.1.2 *Let L be an almost full language over A. If L does not have an admissible word, then L is not regular.*

6.2 Counterexamples

6.2.1 Palindromes

Recall that the set of all palindromes P over A is defined as follows:

$$P = \{w \in A^* \mid w = w^r\}.$$

Note that, if A is singleton ($|A| = 1$), then $P = A^*$ and hence P is regular.

Proposition 6.2.1 *The set of all palindromes P over A is not regular if A consists of at least two letters.*

Proof: One can easily verify that:

$$\mu_n(P) = \begin{cases} \frac{|A|^{n/2}}{|A|^n} = \frac{1}{|A|^{n/2}} & \text{if } n \text{ is even,} \\ \frac{|A| \times |A|^{(n-1)/2}}{|A|^n} = \frac{1}{|A|^{(n-1)/2}} & \text{if } n \text{ is odd.} \end{cases}$$

Hence its limit $\mu(P)$ converges to zero. That is, P is almost empty. Moreover, for every word w in A^*, the word ww^r is in P. This shows that P does not have a forbidden word. Hence P is not regular by Zero Lemma. ∎

Corollary 6.2.1 *P is in $\mathcal{ZO}$ but not in $\mathcal{Z}$.*

Corollary 6.2.2 *$\mathcal{ZO}$ properly contains $\mathcal{Z}$.*

6.2.2 Dyck Language

Recall that the Dyck language D over $A = \{[,]\}$ is the set of all balanced square brackets:

$$D = \{\varepsilon, [], [[]], [][], [[[]]], [[][]], [][][], [[]][], [][[]], [][][], \ldots\}.$$

Here we give a more formal definition of D. Let w be a word over $A = \{[,]\}$. We define the trim function $\mathsf{Trim} : A^* \to A^*$ that maps a word w to more shorter word by *deleting all factors of the form* $[]$ *in* w. For example, the words $[]$, $[[]$ and $[[]][[$ are mapped by deleting all doubly underlined factors $\underline{\underline{[]}}$ as follows:

$$\mathsf{Trim}(\underline{\underline{[]}}) = \varepsilon, \quad \mathsf{Trim}([\underline{\underline{[]}}) = [, \quad \mathsf{Trim}([\underline{\underline{[]}}]\underline{\underline{[]}}[) = [][.$$

It is clear that by the definition of Trim, for every word w, Trim has the fixed point by starting with w: there exists some $m \geq 1$ that satisfies $\mathsf{Trim}^n(w) = \mathsf{Trim}^m(w)$ for all $n \geq m$, and we call such $\mathsf{Trim}^m(w)$ the *reduced word of* w. We denote by $\mathsf{Trim}^*(w)$ the reduced word of w. Then the Dyck language D can be defined as follows:

$$D = \{w \in A^* \mid \mathsf{Trim}^*(w) = \varepsilon\}.$$

Proposition 6.2.2 *The Dyck language D over $A = \{[,]\}$ is not regular.*

Proof: It is well known that $\gamma_{2n}(D)$ for each $n \geq 1$:

$$\gamma_2(D) = 1, \quad \gamma_4(D) = 2, \quad \gamma_6(D) = 5, \quad \gamma_8(D) = 14, \quad \cdots$$

is equal to the nth *Catalan number* which has $\Theta(\frac{4^n}{n^{3/2}})$ asymptotic complexity (*cf.* [21]). Thus we obtain the following equation:

$$\mu_n(D) = \begin{cases} \Theta\left(\frac{1}{n^{3/2}}\right) & \text{if } n \text{ is even,} \\ 0 & \text{if } n \text{ is odd.} \end{cases}$$

Hence its limit $\mu(D)$ converges to zero. That is, D is almost empty. Now we show that D does not have a forbidden word. Let w be an arbitrary word in A^*. By definition, the reduced word of w is of the form:

$$\mathsf{Trim}^*(w) =]^n[^m$$

for some $n, m \geq 0$. Then the word $[^n w]^m$ is in D since the following equation holds:

$$\mathsf{Trim}^*([^n w]^m) = \mathsf{Trim}^*([^n]^n[^m]^m) = \varepsilon.$$

Hence D is not regular by Zero Lemma. ∎

Corollary 6.2.3 *D is in $\mathcal{ZO}$ but not in $\mathcal{Z}$.*

6.3 Bibliographic Notes

It is well known that the syntactic monoid of the Dyck language D is infinite and has two generators p, q satisfying $pq = 1$ and $qp \neq 1$. This monoid is called *bicyclic monoid*, and D can be represented by the inverse of the identity 1 of the bicyclic monoid: $D = \phi_D^{-1}(1)$ (*cf.* [52]). It is also known that the free monoid over A is the syntactic monoid of the set of all palindromes P over A, if A consists of at least two letters (*cf.* Theorem 2.2.5 in [16], Exercises 27 of Section 4.9 in [55]). That is, the syntactic congruence $\sim_P$ of P is the *identity*: $u \sim_P v \Leftrightarrow u = v$ for all words u, v in A^*. A language whose syntactic congruence is the identity, like as P, is called *disjunctive* (*cf.* [28, 16]).

EPILOGUE

In this thesis, the class of regular zero-one languages was characterised in various ways: (i) syntactic monoids with zero (ii) ideal languages (iii) zero automata and (iv) quasi-zero automata. The last characterisation – quasi-zero automata – gives us a linear time algorithm for testing the zero-one law. I can conclude that, from the algebraic point of view, the zero-one law for regular languages is completely unraveled.

One line of extension of my results is investigating zero-one laws *beyond regular languages*. For example, characterising the zero-one law for *visibly pushdown languages* (VPLs), for *deterministic context-free languages* (DCFLs), for *unambiguous context-free languages* (UCFLs), and for *context-free languages* (CFLs). Visibly pushdown languages are introduced by Alur et al. [1] in 2004, and it is known that the class of VPLs enjoys good closure properties (e.g., Boolean operations, Kleene star, and concatenation). Moreover, VPLs has a congruence-based characterisation that resembles the syntactic congruence for regular languages [2].

Another line of extension of my results is investigating zero-one laws for *another structures*. For example, characterising the zero-one law for *regular tree languages* (*cf.* [9]), *ω-regular languages*.

The key points of Zero-One Theorem are the closure properties of $\mathcal{ZO}$ (closed under Boolean operations and quotients), and this points are also applicable beyond regular languages, and possibly, for another structures.

Once again, I would like to emphasise the fact about the zero-one law which was stated in PROLOGUE:

> It is known that the finite satisfiability (i.e., the existence of a finite model) of first-order definable property for finite graphs is undecidable due to Trakhtenbrot's theorem [65]. Thus, for a given first-order sentence Φ, while it is undecidable whether Φ is true for all finite graphs, it is decidable whether Φ is true for almost all finite graphs!

In this sense we can regard the zero-one law as a *theoretically nice approximation of the satisfiability problem*. Though regular languages are very tractable (many decision problems are decidable) and no approximation would be needed in most cases, this point of view – theoretically nice approximation of some decision problem – might be worthwhile for more intractable language classes. In particular, it is well known that the *universality problem (deciding whether a given language is full) for CFLs is undecidable* (*cf.* [59]). Zero-one laws beyond regular languages could be an interesting subject for research in future.

REFERENCES

[1] Rajeev Alur, Viraj Kumar, P. Madhusudan & Mahesh Viswanathan (2004): *Visibly pushdown languages*. In: *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, ACM Press, pp. 202–211.

[2] Rajeev Alur, Viraj Kumar, P. Madhusudan & Mahesh Viswanathan (2005): *Congruences for Visibly Pushdown Languages*. In Luís Caires, GiuseppeF. Italiano, Luís Monteiro, Catuscia Palamidessi & Moti Yung, editors: *Automata, Languages and Programming*, *Lecture Notes in Computer Science* 3580, Springer Berlin Heidelberg, pp. 1102–1114.

[3] Yehoshua Bar-Hillel, Micha Perles & Eli Shamir (1961): *On Formal Properties of Simple Phrase Structure Grammars*. *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung* 14, pp. 143–172.

[4] Jean Berstel (1973): *Sur la densité asymptotique de langages formels*. In: *International Colloquium on Automata, Languages and Programming (ICALP, 1972)*, North-Holland, France, pp. 345–358.

[5] Jean Berstel & Dominique Perrin (1985): *Theory of codes*. Pure and applied mathematics, Academic Press, Orlando, San Diego, New York.

[6] Jean Berstel, Dominique Perrin & Christophe Reutenauer (2009): *Codes and Automata (Encyclopedia of Mathematics and Its Applications)*, first edition. Cambridge University Press, New York, NY, USA.

[7] Andreas Blass, Yuri Gurevich & Dexter Kozen (1985): *A Zero-One Law for Logic with a Fixed-Point Operator*. *Information and Control* 67(1-3), pp. 70–90.

[8] Manuel Bodirsky, Tobias Gärtner, Timo von Oertzen & Jan Schwinghammer (2004): *Efficiently Computing the Density of Regular Languages*. In Martín Farach-Colton, editor: *LATIN 2004: Theoretical Informatics*, *Lecture Notes in Computer Science* 2976, Springer Berlin Heidelberg, pp. 262–270.

[9] Mikołaj Bojańczyk & Igor Walukiewicz (2008): *Forest algebras*. In: *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*., pp. 107–132.

[10] Janusz A. Brzozowski & Imre Simon (1971): *Characterizations of Locally Testable Events*. In: *SWAT (FOCS)*, IEEE Computer Society, pp. 166–176.

[11] Julius Richard Büchi (1960): *Weak second-order arithmetic and finite automata*. *Z. Math. Logik und Grundl. Math.* 6, pp. 66–92.

[12] Alfred H. Clifford & Gordon B. Preston (1967): *The Algebraic Theory of Semigroups, Volume II*. Mathematical Surveys, American Mathematical Society, Providence, Rhode Island.

[13] Kevin J. Compton (1989): *Laws in Logic and Combinatorics*. In Ivan Rival, editor: *Algorithms and Order*, *NATO ASI Series* 255, Springer Netherlands, pp. 353–383.

[14] Kevin J. Compton, C.Ward Henson & Saharon Shelah (1987): *Nonconvergence, undecidability, and intractability in asymptotic problems*. *Ann. Pure Appl. Logic* 36, pp. 207–224.

[15] Volker Diekert, Paul Gastin & Manfred Kufleitner (2008): *A Survey on Small Fragments of First-Order Logic over Finite Words*. *International Journal of Foundations of Computer Science* 19(3), pp. 513–548.

[16] Pal Domosi, Sándor. Horváth & Masami Ito (2014): *Context-Free Languages and Primitive Words*. World Scientific Publishing Company Pte Limited.

[17] Paul Dubreil (1941): *Contribution à la théorie des demi-groupes*. *Mémoires de l'Académie des sciences* 63, pp. 1–52.

[18] Samuel Eilenberg & Bret Tilson (1976): *Automata, languages and machines. Volume B*. Pure and applied mathematics, Academic Press, New-York, San Franciso, London.

[19] Zoltán Ésik & Masami Ito (2003): *Temporal Logic with Cyclic Counting and the Degree of Aperiodicity of Finite Automata*. *Acta Cybernetica* 16(1), pp. 1–28.

[20] Ronald Fagin (1976): *Probabilities on Finite Models*. *J. Symb. Log.* 41(1), pp. 50–58.

[21] Philippe Flajolet & Robert Sedgewick (2009): *Analytic Combinatorics*, first edition. Cambridge University Press, New York, NY, USA.

[22] Haim Gaifman (1964): *Concerning measures in first order calculi*. *Israel Journal of Mathematics* 2(1), pp. 1–18.

[23] Mai Gehrke, Serge Grigorieff & Jean-Éric Pin (2008): *Duality and Equational Theory of Regular Languages*. In: *Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Part II*, ICALP '08, Springer-Verlag, Berlin, Heidelberg, pp. 246–257.

[24] Y. V. Glebskii, D. I. Kogan, M. I. Liogonkii & V. A. Talanov (1969): *Range and degree of realizability of formulas in the restricted predicate calculus*. *Cybernetics* 5, pp. 142–154.

[25] Ulf Grenander (1963): *Probabilities on algebraic structures*. Wiley, New York.

[26] Maurice Gross & André Lentin (1970): *Notions sur les grammaires formelles*. Collection Programmation, Gauthier-Villars.

[27] Robert P. Hunter (1988): *Certain Finitely Generated Compact Zero Dimensional Semigroups*. Journal of the Australian Mathematical Society (Series A) 44, pp. 265–270.

[28] Helmut Jürgensen (2001): *Disjunctivity*. In Masami Ito, Gheorghe Paun & Sheng Yu, editors: *Words, Semigroups, and Transductions*, World Scientific, pp. 255–274.

[29] Stephen C. Kleene (1951): *Representation of events in nerve nets and finite automata*. Rand Corporatin.

[30] Stephen C. Kleene (1956): *Representation of events in nerve nets and finite automata*. In Claude Shannon & John McCarthy, editors: *Automata Studies*, Princeton University Press, Princeton, NJ, pp. 3–41.

[31] Phokion G. Kolaitis & Moshe Y. Vardi (1987): *The Decision Problem for the Probabilities of Higher-order Properties*. In: *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, ACM, New York, NY, USA, pp. 425–435.

[32] Phokion G. Kolaitis & Moshe Y. Vardi (1992): *Infinitary logics and 0-1 laws*. Information and Computation 98(2), pp. 258–294.

[33] Phokion G. Kolaitis & Moshe Y. Vardi (2000): *0-1 Laws for Fragments of Existential Second-Order Logic: A Survey.* In Mogens Nielsen & Branislav Rovan, editors: *MFCS*, *Lecture Notes in Computer Science* 1893, Springer, pp. 84–98.

[34] Mark V. Lawson (2005): *Finite Automata.* Birkhäuser.

[35] Leonid Libkin (2004): *Elements of Finite Model Theory*. SpringerVerlag.

[36] James F. Lynch (1980): *Almost sure theories*. Ann. Mathematical Logic 18, pp. 91–135.

[37] James F. Lynch (1985): *Probabilities of first-order sentences about unary functions*. Trans. Amer. Math. Soc. 287, pp. 543–568.

[38] James F. Lynch (1993): *Convergence laws for random words*. Australas. J. Combin. 7, pp. 145–156.

[39] Stuart W. Margolis (2014): *The q-theory of finite semigroups: history and mathematics*.

[40] Per Martin-Löf (1965): *Probability theory on discrete semigroups*. Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete 4(1), pp. 78–102.

[41] Robert McNaughton (1974): *Algebraic Decision Procedures for Local Testability*. Mathematical Systems Theory 8(1), pp. 60–76.

[42] Marian Mureşan (2009): *A concrete approach to classical analysis*. CMS books in mathematics , Springer.

[43] John R. Myhill (1957): *Finite Automata and the Representation of Events*. Technical Report WADC TR-57-624, Wright-Paterson Air Force Base.

[44] Anil Nerode (1958): *Linear automaton transformations*. Proceedings of the American Mathematical Society 9(4), pp. 541–544.

[45] Richard S. Pierce (1954): *Homomorphisms of Semigroups*. Annals of Mathematics 59, pp. 287–291.

[46] Jean-Éric Pin: *Mathematical foundations of automata theory*. Available at http://www.liafa.jussieu.fr/~jep/PDF/MPRI/MPRI.pdf.

[47] Jean-Éric Pin (1986): *Varieties of formal languages*. Plenum Publishing Corp., New York. With a preface by Marcel-Paul Schützenberger, Translated from the French by A. Howie.

[48] Michael O. Rabin & Dana S. Scott (1959): *Finite Automata and Their Decision Problems*. *IBM J. Res. Dev.* 3(2), pp. 114–125.

[49] Igor Rystsov (1997): *Reset words for commutative and solvable automata*. *Theoretical Computer Science* 172(1–2), pp. 273–279.

[50] Jacques Sakarovitch (2009): *Elements of Automata Theory*. Cambridge University Press, New York, NY, USA.

[51] Arto Salomaa & Matti Soittola (1978): *Automata Theoretic Aspects of Formal Power Series*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[52] Mark V. Sapir (2014): *Combinatorial algebra syntax and semantics*. Springer Monographs in Mathematics, Springer, Cham.

[53] Marcel-Paul Schützenberger (1956): *Une théorie algébrique du codage*. *Séminaire Dubreil. Algèbre et Théorie des Nombres* 9, pp. 1–24.

[54] Marcel-Paul Schützenberger (1965): *On finite monoids having only trivial subgroups*. *Information and Control* 8(2), pp. 190–194.

[55] Jeffrey Shallit (2008): *A Second Course in Formal Languages and Automata Theory*, first edition. Cambridge University Press, New York, NY, USA.

[56] Imre Simon (1975): *Piecewise Testable Events*. In: *Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages*, Springer-Verlag, London, UK, UK, pp. 214–222.

[57] Ryoma Sin'ya (2014): *Graph Spectral Properties of Deterministic Finite Automata*. In Arseny M. Shur & Mikhail V. Volkov, editors: *Developments in Language Theory*, *Lecture Notes in Computer Science* 8633, pp. 76–83.

[58] Ryoma Sin'ya (2015): *An Automata Theoretic Approach to the Zero-One Law for Regular Languages: Algorithmic and Logical Aspects*. In: *Proceedings Sixth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2015*, pp. 172–185.

[59] Michael Sipser (2006): *Introduction to the theory of computation: second edition*, second edition. PWS Pub., Boston.

[60] Howard Straubing (2002): *On Logical Descriptions of Regular Languages*. In Sergio Rajsbaum, editor: *LATIN 2002: Theoretical Informatics*, *Lecture Notes in Computer Science* 2286, Springer Berlin Heidelberg, pp. 528–538.

[61] Howard Straubing & Pascal Weil (2015): *Varieties*. *CoRR* abs/1502.03951.

[62] V. A. Talanov & V. V. Knyazev (1986): *The asymptotic truth value of infinite formulas (in Russian)*. In: *All-Union seminar on discrete mathematics and its applications*, pp. 56–61.

[63] Terence Tao (2011): *An Introduction to Measure Theory*. Graduate studies in mathematics, American Mathematical Soc.

[64] Marianne Teissier (1951): *Sur les équivalences regulières dans les demi-groupes*. *Comptes rendus de l'Académie des sciences* 232, pp. 1987–1989.

[65] Boris Trakhtenbrot (1950): *The Impossibility of an Algorithm for the Decidability Problem on Finite Classes*. In: *Proceedings of the USSR Academy of Sciences (in Russian)*, 70, pp. 569–572.