An Automata Theoretic Approach to the Zero-One Law for Regular Languages: Algorithmic and Logical Aspects

Ryoma Sin'ya
Tokyo Institute of Technology.
shinya.r.aa@m.titech.ac.jp
École Nationale Supérieure des Télécommunications.
rshinya@enst.fr

A zero-one language L is a regular language whose asymptotic probability converges to either zero or one. In this case, we say that L obeys the zero-one law. We prove that a regular language obeys the zero-one law if and only if its syntactic monoid has a zero element, by means of Eilenberg's variety theoretic approach. Our 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. In addition, we discuss the logical aspects of the zero-one law for regular languages.

1 Introduction

Let L be a regular 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: $\gamma_n(L) = |L \cap A^n|$ where A^n is the set of all words of length n over A. 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* $\mu(L)$ of L is defined by $\mu(L) = \lim_{n\to\infty} \mu_n(L)$, if the limit exists. 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*. Here we introduce a new class of regular languages which is the main target of this paper.

Definition 1 (zero-one language)**.** A *zero-one language* L is a regular language whose asymptotic probability $\mu(L)$ is either zero or one. In this case, we say that L *obeys the zero-one law*. We denote by $\mathscr{ZO}$ *the class of all regular zero-one languages*.

As we will describe later (see Section 7), the notion of "zero-one law" defined here is a fundamental object in *finite model theory*.

Example 1. We now consider a few examples.

- The set of all words A^* over A satisfies $\mu(A^*) = 1$, and its complement $\emptyset$ satisfies $\mu(\emptyset) = 0$. These two languages obey the zero-one law.
- Consider aA^* the set of all words which start with the letter a in A. Then

$$\mu_n(aA^*) = \frac{|aA^{n-1}|}{|A^n|} = \frac{1}{|A|}.$$

Hence, its limit $\mu((aA)^*)$ is $1/|A|$ and aA^* is zero-one if and only if A is *unary*: $A = \{a\}$.

J. Esparza, E. Tronci (Eds.): Games, Automata,
Logics and Formal Verification 2015 (GandALF'15)
EPTCS 193, 2015, pp. 172–185, doi:10.4204/EPTCS.193.13

- 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 ([5], see Section 8).

Our results and contributions. In this paper, we show that the following class of languages exactly captures the zero-one law for regular languages.

Definition 2 ([16])**.** A *language with zero* is a regular language whose syntactic monoid has a zero element. We denote by $\mathscr{Z}$ *the class of all regular languages with zero*.

More precisely, we prove the following theorem, which states that $\mathscr{ZO}$ and $\mathscr{Z}$ are equivalent by means of a transparent condition of their automata: *zero automata* (Section 3) and *quasi-zero automata* (Section 6) which will be described later. The remarkable fact is that, $\mathscr{ZO} = \mathscr{Z}$ holds even though these two notions seem completely different from each other; $\mathscr{ZO}$ is defined by the asymptotic behavior of its probability, $\mathscr{Z}$ is defined by the existence of a zero of its syntactic monoid.

Theorem 1. *Let L be a regular language and $\mathscr{A}_L$ be the minimal automaton of L. Then the following four conditions are equivalent.*

① *$\mathscr{A}_L$ is zero.*

② *L is with zero.*

③ *L obeys the zero-one law.*

④ *L is recognised by a quasi-zero automaton.*

We will prove this theorem as a cyclic chain of implications: ① ⇒ ② ⇒ ③ ⇒ ①, and ① ⇔ ④ independently. We should notice that the most difficult part of this proof is the implication ③ ⇒ ①, while the former part ① ⇒ ② ⇒ ③ is easy. The key points of the proof of this part are *closure properties of* $\mathscr{ZO}$ and Lemma 1, which comes from Eilenberg's variety theorem. The automata characterisation ④ of Theorem 1 leads to a linear time algorithm for testing whether a given regular language is zero-one. In addition, our automata theoretic proof sheds new light on the relation between the zero-one law for regular languages and *logical fragments over finite words*.

Paper outline. The remainder of this paper is organised as follows. In Section 2, we first give the necessary definitions and terminology for languages, monoids, and automata. Lemma 1 will be introduced in this section. For the sake of completeness we include the proof of Lemma 1. Section 3 provides a detailed exposition of the notion of zero automata. Our automata theoretic proof of Theorem 1 consists of three parts: (i) Check certain closure properties of $\mathscr{ZO}$ (Section 4), (ii) Apply Lemma 1 to prove the implication ③ ⇒ ① (Section 5). (iii) Generalise the notion of zero automata, and prove ① ⇔ ④ (Section 6). In Section 6, we will give a linear time algorithm (Theorem 2). The logical aspects of our results are investigated in Section 7. Finally, we discuss some related works of our results and conclude this paper in Section 8. We try to keep all sections as self-contained as possible.

2 Preliminaries

In this paper, all considered automata are *deterministic finite, complete* and *accessible*. We refer the reader to the book by Sakarovitch [18] for background material.

Languages and monoids. We denote by A^* $[A^n]$ the set of all words [of length n] over a nonempty finite alphabet A, and by $|w|$ the length of a word w in A^*. The empty word is denoted by ε. That is, A^* is the free monoid over A with the neutral element ε. We can easily verify that

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

holds for any language L of A^* and $k \geq 0$. 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 $\mu(A^k L) = \mu(L)$. If two languages L and K of A^* are mutually disjoint ($L \cap K = \emptyset$), then clearly $\mu(L \cup K) = \mu(L) + \mu(K)$ holds if both $\mu(L)$ and $\mu(K)$ exist. We say that v is a *factor of* w if, there exists x, y in A^* such that $w = xvy$. Let L be a language of 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

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

We denote by $\overline{L} = A^* \setminus L$ the *complement* of L. The *syntactic congruence* of L of A^* is the 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. If M is a monoid, an element $\mathbf{0}$ in M is said to be a *zero* if, $\mathbf{0}m = m\mathbf{0} = \mathbf{0}$ holds for all m in M.

Automata and an important lemma. An *(complete deterministic finite) automaton* over a finite alphabet A is a quintuple $\mathscr{A} = \langle Q, A, \cdot, q_0, F \rangle$ 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* $\mathscr{A}$ is denoted by $L(\mathscr{A}) = \{w \in A^* \mid q_0 \cdot w \in F\}$. We say that $\mathscr{A}$ *recognises* L if $L = L(\mathscr{A})$. 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 $\mathscr{A}_L$. Each word w in A^* defines the transformation $w : q \mapsto q \cdot w$ on Q. The *transition monoid* of $\mathscr{A}$ is equal to the transformation monoid generated by the generators A. It is well known that the syntactic monoid of a regular language is equal to the transition monoid of its minimal 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 $\mathscr{A}$ is minimal if and only if the following condition

$$p = q \quad \Leftrightarrow \quad \mathsf{Fut}(p) = \mathsf{Fut}(q) \tag{\textbf{M}}$$

holds for every pair of states p,q in Q. Myhill-Nerode theorem states that every regular language has only a finite number of left and right quotients.

In Section 5, to prove Theorem 1, we will use the following technical but important lemma. For the sake of completeness we include the proof, which is essentially based on "Proof of Theorem 3.2 and 3.2s" in the book [7] by Eilenberg.

Lemma 1. *Let $\mathscr{A}_L = \langle Q, A, \cdot, q_0, F \rangle$ 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}.*

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 $\bigcup_{w \in \emptyset} Lw^{-1} = \emptyset$:

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

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

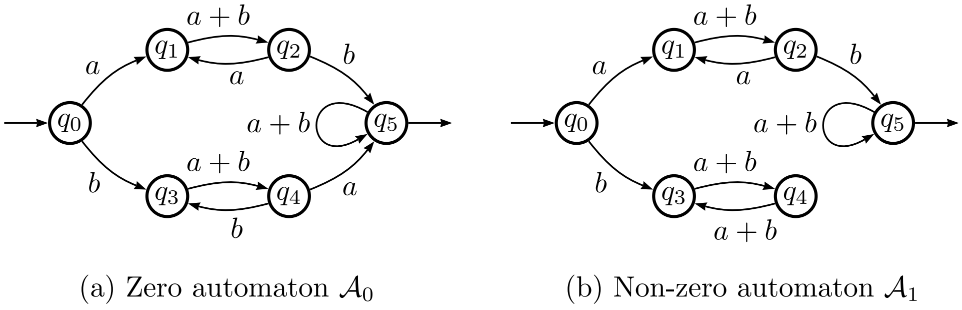
We prove first that the left hand side is contained in the right hand side in Equation (1). Let v be a word in $\mathsf{Past}(q)$. If a word w in $\mathsf{Fut}(q)$, then vw in L by the definition, and hence v in Lw^{-1}. If a word w not in $\mathsf{Fut}(q)$, then vw not in L by the definition, and hence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

Then we prove that the right hand side is contained in the left hand side in Equation (1). Let v be a word in right hand side in Equation (1). Let p be the state satisfies $q_0 \cdot v = p$, that is, v is a word in $\mathsf{Past}(p)$. For any w in $\mathsf{Fut}(q)$, by the form of Equation (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 $\mathscr{A}_L$. Hence we obtain v in $\mathsf{Past}(q)$ and thus the right hand side is contained in the left hand side in Equation (1). □

Remark 1. 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 taking submonoids, quotients and finite direct products (*cf.* [16]). Eilenberg's variety theorem [7] states that varieties of languages are in one-to-one correspondence with varieties of finite monoids. Lemma 1 shows us an importance of the Boolean operations taken in tandem with quotients. While this lemma is known (*cf.* [8]), 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.

3 Zero automata

In this seciton, we introduce a *zero automaton*, which plays a major role in our work. In contrast to the class of monoids with zero, their natural counterpart, the class of zero automata has not been given much



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

Figure 1: Zero and non-zero automata

attention. To the best of our knowledge, only few studies (e.g., [17]) have investigated zero automata in the context of the theory of synchronising word for *Černý's conjecture*.

Let $\mathscr{A}$ be an automaton $\langle Q,A,\cdot,q_0,F\rangle$. 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 $p\cdot w = q$. $\mathscr{A}$ is called *accessible* if every state q in Q is reachable from the initial state q_0. A subset P of Q is called *strongly connected component*, if for each state q in P, q is reachable from every other state 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*, analogously, if 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. Note that, every (complete) automaton has at least one strongly connected sink component. The family of all strongly connected sink components of $\mathscr{A}$ is denoted by $\mathsf{Sink}(\mathscr{A})$. A strongly connected component P is *trivial* if it consists of some single state $P = \{p\}$. We shall identify a singleton $\{p\}$ with its unique element p. A word w is a *synchronising word of $\mathscr{A}$* 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 Section 5, the following class of automata captures precisely the zero-one law for regular languages.

Definition 3 ([17])**.** A *zero automaton* is a synchronising automaton with a sink state.

Example 2. Consider two automata $\mathscr{A}_0$ and $\mathscr{A}_1$ illustrated in Figure 1. $\mathscr{A}_0$ is a zero automaton but $\mathscr{A}_1$ is not, though both automata have a sink state q_5. The only difference between $\mathscr{A}_0$ and $\mathscr{A}_1$ is the transition result of $q_4\cdot a$; which equals to q_5 in $\mathscr{A}_0$, while which equals to q_3 in $\mathscr{A}_1$. We can easily verify that, $\mathscr{A}_0$ has a unique strongly connected sink component q_5, while $\mathscr{A}_1$ has two strongly connected sink components $\{q_3,q_4\}$ and q_5.

Definition 3 can be rephrased as follows.

Lemma 2. *Let $\mathscr{A} = \langle Q,A,\cdot,q_0,F\rangle$ be an automaton. Then $\mathscr{A}$ is zero if and only if $\mathscr{A}$ has a unique strongly connected sink component and it is trivial, i.e.,* $\mathsf{Sink}(\mathscr{A}) = \{\{p\}\}$ *for a certain sink state p.*

Proof. First we assume $\mathscr{A}$ 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 strongly connected sink component in $Q\setminus p$.

Now we prove the converse direction, we assume $\mathscr{A}$ has a unique strongly connected 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 strongly connected sink component which does not contain p. This contradicts with the uniqueness of the closed strongly connected component

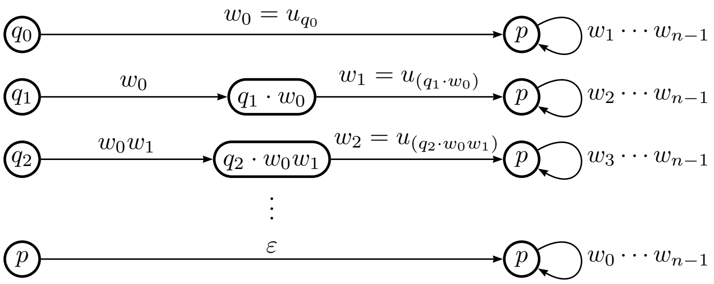


Figure 2: Synchronising word $v_{n-1} = w_0 \cdots w_{n-1}$ in the proof of Lemma 2

p in $\mathscr{A}$. 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 satisfies $q_i \cdot u_{q_i} = p$, and v_{i-1} is the word of the form $w_0 \cdots w_{i-1}$. As shown in Figure 2, we can easily verify that the word $v_{n-1} = w_0 \cdots w_{n-1}$ is a synchronising word satisfies $q \cdot v_{n-1} = p$ for each q in Q.

For example, consider the zero automaton $\mathscr{A}_0$ in Figure 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	$aabb$
q_5	ε	ε	$aabb$

The obtained word $v_{q_4} = aabb$ is a synchronising word which satisfies $q_i \cdot aabb = q_5$ for all q_i in $\mathscr{A}_0$. It is clear that the non-zero automaton $\mathscr{A}_1$ in Figure 1 does not have a synchronising word since it has two strongly connected sink components. □

4 Closure properties of $\mathscr{ZO}$

We first introduce the following lemma.

Lemma 3. *Let L be a language of A^* and w be a word in A^k. Then the asymptotic probability of L exists if and only if the asymptotic probability of the language wL $[Lw]$ exists. Moreover, these limits satisfies 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 language uL and vL are obviously mutually disjoint and these counting functions satisfies

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

This shows that uL and vL have the same counting function and thus have the same asymptotic probability if its exists. We can easily verify that

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

holds for any w in A^k. □

Now we prove the following proposition, which states the necessary closure properties of the class $\mathscr{ZO}$ for Lemma 1.

Proposition 1. *$\mathscr{ZO}$ is closed under Boolean operations, left and right quotients.*

Proposition 1. We first prove that $\mathscr{ZO}$ is closed under Boolean operations, and then prove that $\mathscr{ZO}$ is closed under quotients.

$\mathscr{ZO}$ is closed under Boolean operations. Let L, K be two languages in $\mathscr{ZO}$. It is obvious that $\mathscr{ZO}$ is closed under complement since $\mu(\overline{L}) = 1 - \mu(L) \in \{0,1\}$, and we can easily verify that the following equations holds.

- $\mu(L \cup K) = 0$ if $\mu(L) = 0$ and $\mu(K) = 0$;
- $\mu(L \cap K) = 0$ if either $\mu(L) = 0$ or $\mu(K) = 0$;
- $\mu(L \cup K) = 1$ if either $\mu(L) = 1$ or $\mu(K) = 1$;
- $\mu(L \cap K) = 1$ if $\mu(L) = 1$ and $\mu(K) = 1$.

$\mathscr{ZO}$ is closed under quotients. We first prove that $\mathscr{ZO}$ is closed under left quotients. Let L be a regular language in $\mathscr{ZO}$ and we assume that L does not contain ε without loss of generality. First we assume $\mu(L) = 0$. By the definition of left quotients, one can easily verify that

$$L = \bigcup_{a \in A} L \cap aA^* = \bigcup_{a \in A} aa^{-1}L$$

holds (since $\varepsilon \notin L$) and all these sets $aa^{-1}L$ $(= L \cap aA^*)$ are mutually disjoint. It follows that the following equation holds.

$$\begin{aligned} \mu(L) &= \lim_{n\to\infty} \frac{|L \cap A^n|}{|A^n|} = \lim_{n\to\infty} \frac{|\left(\bigcup_{a\in A} aa^{-1}L\right) \cap A^n|}{|A^n|} = \lim_{n\to\infty} \frac{|\bigcup_{a\in A}(aa^{-1}L \cap A^n)|}{|A^n|} \\ &= \lim_{n\to\infty} \sum_{a\in A} \frac{|aa^{-1}L \cap A^n|}{|A^n|} = \sum_{a\in A} \mu(aa^{-1}L) = 0. \end{aligned}$$

That is, the asymptotic probability $\mu(aa^{-1}L)$ equals to zero for each a in A, since these summation converges to zero. In addition, $\mu(aa^{-1}L)$ coincides with $\mu(a^{-1}L)$ for any a in A, because $\mu(aa^{-1}L) = |A|^{-1}\mu(a^{-1}L) = 0$ by Lemma 3 whence $\mu(a^{-1}L) = 0$.

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

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

holds. We therefore obtain:

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

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

5 Equivalence of $\mathscr{ZO}$ and $\mathscr{Z}$

We will use the following lemma, which is a direct consequence of Lemma 1 and Proposition 1.

Lemma 4. *Let L be a regular language in $\mathscr{ZO}$, $\mathscr{A}_L = \langle Q, A, \cdot, q_0, F\rangle$ be its minimal automaton. Then, for any subset P of Q in $\mathscr{A}_L$, its past* $\mathsf{Past}(P)$ *is also in $\mathscr{ZO}$.*

Proof. By Lemma 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 $\mathscr{ZO}$ and $\mathscr{ZO}$ is closed under Boolean operations and quotients by Proposition 1. □

Lemma 4 will be used for proving the direction ③ ⇒ ①. Now we give a proof.

Proof of Theorem 1. We show the implication ① ⇒ ② ⇒ ③ ⇒ ①. The former implication ① ⇒ ② ⇒ ③ is easy and almost folklore, but we include a proof here to be self-contained.

① ⇒ ② ($\mathscr{A}_L$ is zero ⇒ L is with zero). Let $\mathscr{A}_L = \langle Q, A, \cdot, q_0, F\rangle$ be the minimal automaton of L and it is zero with a sink state p. Let M be the transition monoid of $\mathscr{A}_L$ 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 $\mathscr{A}_L$ 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.

② ⇒ ③ (L is with zero ⇒ L obeys the zero-one law). Let L be a regular language in $\mathscr{Z}$, M be its syntactic monoid with a zero element $\mathbf{0}$ and $\phi : A^* \to M$ be its syntactic morphism. We choose a word $w_\mathbf{0}$ from the preimage of $\mathbf{0}$: $w_\mathbf{0} \in \phi^{-1}(\mathbf{0})$.

Now we prove $\mu(L) = 1$ if $w_\mathbf{0}$ 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. Let $L_{w_\mathbf{0}} = A^* w_\mathbf{0} A^*$ be the set of all words that contain $w_\mathbf{0}$ as a factor. Then clearly $L_{w_\mathbf{0}}$ is contained in L from which we get $\mu_n(L_{w_\mathbf{0}}) \le \mu_n(L)$ for all n. The probability $\mu_n(L_{w_\mathbf{0}})$ is nothing but the probability that a randomly chosen word of length n contains $w_\mathbf{0}$ as a factor. The following well known elementally fact, sometimes called *Borges's theorem* (*cf.* Note I.35 in [10]), ensures that $\mu_n(L_{w_\mathbf{0}})$ tends to one if n tends to infinity. This shows $\mu(L) = \mu(L_{w_\mathbf{0}}) = 1$ and we can prove $\mu(L) = 0$ if $w_\mathbf{0}$ not in L by the same manner.

Borges's 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.*

③ ⇒ ① (L obeys the zero-one law ⇒ $\mathscr{A}_L$ is zero). Let L be a regular language in $\mathscr{ZO}$ and $\mathscr{A}_L = \langle Q, A, \cdot, q_0, F\rangle$ be its minimal automaton, let $\mathsf{Sink}(\mathscr{A}_L) = \{P_1, \cdots, P_k\}$ for some $k \ge 0$. Our goal is to prove $k = 1$ and $\mathsf{Sink}(\mathscr{A}_L) = \{\{p\}\}$ for a certain sink state p. It follows that $\mathscr{A}_L$ is zero by Lemma 2.

For any strongly connected sink component P_i, there exists a word w_i such that $q_0 \cdot w_i$ in P_i because $\mathscr{A}_L$ is accessible. Since P_i is sink, the language $w_i A^*$ is contained in $\mathsf{Past}(P_i)$ from which we get

$$0 < \mu(w_i A^*) = |A|^{-|w_i|}\mu(A^*) = |A|^{-|w_i|} \le \mu(\mathsf{Past}(P_i)) \qquad (2)$$

for each P_i by Lemma 3. Lemma 4 and Equation (2) implies that the asymptotic probability $\mu(\mathsf{Past}(P_i))$ surely exists and satisfies

$$\mu(\mathsf{Past}(P_i)) = 1 \tag{3}$$

for every strongly connected sink component P_i.

Now we prove $k = 1$. By Equation (3), 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 $\mathscr{A}_L$ is deterministic and thus all $\mathsf{Past}(P_i)$ are mutually disjoint. This clearly shows $k = 1$, that is, there exists a unique strongly connected sink component, say P, in $\mathscr{A}_L$: $\mathsf{Sink}(\mathscr{A}_L) = \{P\}$.

Next we let $P = \{p_1, \cdots, p_n\}$ and prove $n = 1$. Since P satisfies $\mu(\mathsf{Past}(P)) = 1$ by Equation (3), there exists exactly one state p in P satisfies $\mu(\mathsf{Past}(p)) = 1$ by Lemma 4. Further, because P is strongly connected, for every state p_i in P, there exists a word w_i such that $p \cdot w_i = p_i$. It follows that $\mathsf{Past}(p)w_i \subseteq \mathsf{Past}(p_i)$ and thus

$$0 < \mu(\mathsf{Past}(p)w_i) = |A|^{-|w_i|}\mu(\mathsf{Past}(p)) = |A|^{-|w_i|} \leq \mu(\mathsf{Past}(p_i)) = 1 \tag{4}$$

holds for every state p_i in P by Lemma 3 and Lemma 4. Equation (3) and (4) implies

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

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

Remark 2. It is interesting that, though we use Borges's theorem to prove the direction ② ⇒ ③, Theorem 1 is a vast generalisation of Borges's theorem, since any language of the form A^*KA^* where K is regular is always recognised by a zero automaton (but the converse is not true). To state Theorem 1 more precisely, by the proof above we can easily verify that, a zero-one language L satisfies $\mu(L) = 1$ [$\mu(L) = 0$] if and only if its minimal automaton $\mathscr{A}_L$ is zero and the sink state of $\mathscr{A}_L$ is final [non-final].

6 Linear time algorithm for testing the zero-one law

The equivalence of zero-automata and the zero-one law gives us an effective algorithm. For a given n-states automaton $\mathscr{A}$, we can determine whether $L(\mathscr{A})$ obeys the zero-one law by the following steps: (i) Minimise $\mathscr{A}$ to obtain its minimal automaton $\mathscr{B}$. (ii) Calculate the family of all strongly connected components P of $\mathscr{B}$. (iii) Check whether P contains exactly one strongly connected sink component and it is trivial, i.e., whether $\mathscr{B}$ is a zero automaton (Lemma 2). 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 $\mathscr{A}$ to obtain $\mathscr{B}$ 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, if its is given by an n-states deterministic finite automaton. We can obtain, however, more efficient algorithm *by avoiding minimisation*.

In order to do that, there is a need for further investigation of the structure of zero automata.

Quasi-zero automata and more effective algorithm. Let $\mathscr{A} = \langle Q, A, \cdot, q_0, F \rangle$ be an automaton. The *Nerode equivalence* $\sim$ of $\mathscr{A}$ 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 F is saturated by $\sim$ and $p \sim q$ implies $p \cdot w \sim q \cdot w$ for all $w \in A^*$. Hence it follows that there is a well defined new automaton $\mathscr{A}/\!\sim$, *the quotient automaton of* $\mathscr{A}$:

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

where $[q]_\sim$ is the equivalence class modulo $\sim$ of q, $S/\!\sim = \{[q]_\sim \mid q \in S\}$ 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 $[p]_\sim \cdot a = [p \cdot a]_\sim$. We define the natural mapping $\phi_\sim : Q \to Q/\!\sim$ by $\phi_\sim(q) = [q]_\sim$. Condition (**M**) for minimal automata implies that, for any automaton $\mathscr{A}$, its quotient automaton $\mathscr{A}/\!\sim$ is the minimal automaton of $L(\mathscr{A})$. We shall identify the quotient automaton $\mathscr{A}/\!\sim$ with the minimal automaton of $L(\mathscr{A})$ (*cf.* [18]).

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

Definition 4 (quasi-zero automaton)**.** An automaton $\mathscr{A} = \langle Q, A, \cdot, q_0, F \rangle$ is *quasi-zero* if either $\bigcup \mathsf{Sink}(\mathscr{A}) \subseteq F$ or $\bigcup \mathsf{Sink}(\mathscr{A}) \cap F = \emptyset$ holds.

Since every zero automaton $\mathscr{A}$ satisfies $\bigcup \mathsf{Sink}(\mathscr{A}) = \{p\}$ for a certain state p (Lemma 2), every zero automaton is quasi-zero. The following proposition shows that the minimal automaton of any quasi-zero automaton is zero and *vice versa* (this justifies the term "quasi-zero").

Proposition 2. *An automaton $\mathscr{A} = \langle Q, A, \cdot, q_0, F \rangle$ is quasi-zero if and only if $\mathscr{A}/\!\sim$ is zero.*

Proof. This proposition shows exactly the equivalence ① $\Leftrightarrow$ ④ in Theorem 1.

① $\Rightarrow$ ④ **($\mathscr{A}/\!\sim$ is zero $\Rightarrow$ $\mathscr{A}$ is quasi-zero).** Let p be the unique sink state of $\mathscr{A}/\!\sim$. 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

$$\bigcup \mathsf{Sink}(\mathscr{A}) \subseteq F \tag{5}$$

by contradiction. Let us assume that Inclusion (5) does not hold, that is, we assume there exists a non-final state q in $\bigcup \mathsf{Sink}(\mathscr{A})$. Let P be the strongly connected sink component of $\mathscr{A}$ that contains q. Since P is sink and strongly connected, $\phi_\sim(P)$ is sink and strongly connected in $\mathscr{A}/\!\sim$ too. Moreover, $\phi_\sim(P)$ 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 $\mathsf{Fut}([q']_\sim) \neq \mathsf{Fut}(p)$ and $[q']_\sim \neq p$. That is, $\mathscr{A}/\!\sim$ has at least two strongly connected sink components $\phi_\sim(P)$ and p. This is contradiction.

④ $\Rightarrow$ ① **($\mathscr{A}$ is quasi-zero $\Rightarrow$ $\mathscr{A}/\!\sim$ is zero).** To prove this direction, it is enough to consider the case when $\bigcup \mathsf{Sink}(\mathscr{A}) \subseteq F$. Since $\mathscr{A}$ is quasi-zero, all states in $\bigcup \mathsf{Sink}(\mathscr{A})$ have the same future A^*, i.e., $\mathsf{Fut}(q) = A^*$ for every state q in $\bigcup \mathsf{Sink}(\mathscr{A})$, because $\bigcup \mathsf{Sink}(\mathscr{A}) \subseteq F$ implies $q \cdot w \in F$ for every state q in $\bigcup \mathsf{Sink}(\mathscr{A})$ and every word w in A^*. This implies that $\bigcup \mathsf{Sink}(\mathscr{A})/\!\sim$ consists of a single equivalence class, say p. Moreover, this equivalence class p is a sink state in $\mathscr{A}/\!\sim$ by the definition of sink and Condition (**M**) of the minimality of $\mathscr{A}/\!\sim$. We now show that, by contradiction, $\mathscr{A}/\!\sim$ has only one strongly connected sink component p:

$$\bigcup \mathsf{Sink}(\mathscr{A}/\!\sim) = \{p\} \tag{6}$$

from which we obtain $\mathscr{A}/\!\sim$ is zero by Lemma 2. Let us assume that Inclusion (6) does not hold, that is, we assume there exists a strongly connected sink component $R = \{r_1, \cdots, r_n\}$ of $\mathscr{A}/\!\sim$, which does not contain p. Recall that each state r_i of $\mathscr{A}/\!\sim$ is an equivalence class, i.e., a set of states, of $\mathscr{A}$. Let $S = \phi_\sim^{-1}(R)$ be a set of states of $\mathscr{A}$. Since R is strongly connected sink component of $\mathscr{A}/\!\sim$, its preimage S contains at least one strongly connected sink component, say P, of $\mathscr{A}$. For every state q in P, $\mathsf{Fut}(q)$ is not equal to $A^* = \mathsf{Fut}(p)$, because $p \notin \phi_\sim(P) \subseteq \phi_\sim(S) = R$ implies $[q]_\sim \neq p$. This contradicts with the assumption that $\mathsf{Fut}(q) = A^*$ for every state q in $\bigcup \mathsf{Sink}(\mathscr{A})$. This completes the proof of Theorem 1. □

By using this proposition, we obtain a linear time algorithm by avoiding minimisation as stated in the following theorem.

Theorem 2. *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 deterministic finite automaton.*

Proof. For a given n-states automaton $\mathscr{A}$, we can determine whether $L(\mathscr{A})$ obeys the zero-one law by the following steps: (i) Calculate the family of all strongly connected components P of $\mathscr{A}$. (ii) Extract all strongly connected sink components from P to obtain $\mathsf{Sink}(\mathscr{A})$. (iii) Check whether, in $\bigcup \mathsf{Sink}(\mathscr{A})$, either all states are final or all states are non-final, i.e., whether $\mathscr{A}$ is quasi-zero. By Theorem 1, $L(\mathscr{A})$ obeys the zero-one law if and only if $\mathscr{A}$ is quasi-zero. Hence this algorithm is correct. All steps (i) $\sim$ (iii) can be done in $\mathrm{O}(n)$, this ends the proof. □

7 Logical aspects of the zero-one law

There are different manners to define a language: a set of *finite words*. In the descriptive approach, the words of a language are characterised by a property. The automata approach is a special case of the descriptive approach. Another variant of the descriptive approach consists in defining languages by logical formulae: we regard words as *finite structures with a linear order composed of a sequence of positions labeled over finite alphabet*. The zero-one law, which is defined in this paper, has been studied extensively in finite model theory (*cf.* Chapter 12 “Zero-One Laws” of [14]). This notion can be applied to logics over, not only finite words, but also arbitrary *finite structures*, such as *finite graphs*: we regard graphs as finite structures with a set of nodes and their edge relation. We say that a logic $\mathscr{L}$, over fixed finite structures, has the zero-one law if every property Φ definable in $\mathscr{L}$ satisfies $\mu(\Phi) \in \{0,1\}$ (μ is defined analogously). Broadly speaking, every property Φ is either *almost surely true* or *almost surely false*. Fagin’s theorem [9] states that *first-order logic* FO for finite graphs has the zero-one law. Moreover, an FO sentence Φ is almost surely true (i.e., $\mu(\Phi) = 1$) if and only if Φ is true on a certain infinite graph: the *random graph*. This characterisation leads to the fact that, for any FO sentence Φ, it is decidable whether $\mu(\Phi) = 1$ (*cf.* Corollary 12.11 in [14]). 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 logics (e.g., *logic with a fixed point operator* [4], *finite variable infinitary logic* [12] and certain fragments of *second-order logic* [13]) have the zero-one law.

By contrast, though many logics have the zero-one law, their extensions with ordering (like as logics over finite words), no longer have it. In fact, over both finite graphs and finite words, while first-order logic FO has the zero-one law, its extension with a linear order FO[<] does not.

Example 3. A simple counterexample is the language $(aA)^*$ which can be defined by the FO[<] sentence $\Phi_{aA^*} = \exists i\,(\forall j(i < j) \wedge P_a(i))$. The variables i and j of this sentence represent *position* in a word. The

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

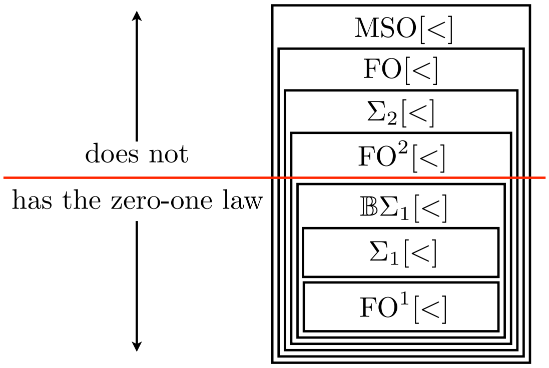


Figure 3: Logical and algebraic characterisations of well known subclasses of regular languages.

sentence $P_a(i)$ is interpreted to mean "the i-th letter is a". This language aA^* satisfies $\mu_n(aA^*) = 1/|A|$ as we stated in Section 1, hence Φ_{aA^*} does not obey the zero-one law in general. It follows that FO[<] for finite words 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 $\mathscr{ZO}$, in Figure 3. Details and full proofs of these results can be found in a very nice survey [6] by Diekert *et al.* In Figure 3, we use standard abridged notation: $\mathrm{FO}^n[<]$ for first-order logic with n variables; $\Sigma_n[<]$ for FO formulae with n blocks of quantifiers and starting with a block of existential quantifiers; $\mathbb{B}\Sigma_n[<]$ for the Boolean closure of $\Sigma_n[<]$. A *monomial* over A is a language of the form $A_0^*a_1A_1^*a_2\cdots a_kA_k^*$ where a_i in A and $A_i \subseteq A$ for each i, and is *unambiguous* if for all $w \in A_0^*a_1A_1^*a_2\cdots a_kA_k^*$ there exists exactly one factorisation $w = w_0a_1w_1a_w\cdots a_kw_k$ 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;
- *piecewise testable* if it is a finite Boolean combination of simple polynomials;
- *simple polynomial* if it is a finite union of languages of the form $A^*a_1A^*a_2\cdots a_kA^*$.

The question then arises as to *which fragments of* FO[<] *over finite words have the zero-one law.* The algebraic characterisation of the zero-one law partially answers this question. Since every $\mathscr{J}$*-trivial* syntactic monoid has a zero element (*cf.* [16]), Theorem 1 leads to the following corollary.

Corollary 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 3, 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 1 shows us a "separation line" (red line in Figure 3). It must be noted that the class of zero-one languages $\mathscr{ZO}$ 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 3. An interesting open problem is whether there exists a logical fragment that exactly captures the zero-one law.

8 Related works

The notion of probability μ_n for regular languages has been studied by Berstel [1] from 1973, and by Salomaa and Soittola [19] 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.* [5]. 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 deterministic finite automaton.

A similar notion, *density* of a language have also been studied in *algebraic coding theory* (*cf.* [2, 3]). A *probability distribution* π on A^* is a function $\pi : A^* \to [0,1]$ such that $\pi(\varepsilon) = 1$ and $\sum_{a\in A} \pi(wa) = \pi(w)$ for all w in A^*. As a particular case, a *Bernoulli distribution* is a morphism from A^* into $[0,1]$ such that $\sum_{a\in A} \pi(a) = 1$. Clearly, a Bernoulli distribution is a probability distribution. We denote by $A^{(n)} = A^0 \cup A \cup \cdots \cup A^{n-1}$ the set of all words of length less than n over a finite alphabet A. The *density* $\delta(L)$ of L is a limit defined by

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

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 the sequence $\pi(L \cap A^n)$ has a limit, then $\delta(L)$ also has a limit, and both are equal. The converse, however, does not hold (e.g., $\delta((AA)^*) = 1/2$). In their book [3], 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 ⇒ ③, "easy part" of our Theorem 1). Some other related results can be found in the *theory of probabilities on algebraic structures* initiated by Grenander [11] and Martin-Löf [15].

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

Acknowledgement. I wish to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper, especially, who informed me the previous works in algebraic coding theory (Theorem 13.4.5 in [3]). Special thanks also go to Prof. Yasuhiko Minamide (Tokyo Institute of Technology) whose meticulous comments for Lemma 1 were an enormous help to me. I am grateful to Prof. Jacques Sakarovitch (Télécom ParisTech) whose comments and suggestions (and his excellent book [18]) were innumerably valuable throughout the course of my study. This work was supported by JSPS KAKENHI Grant Number $26 \cdot 11962$.

References

[1] 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.

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

[3] Jean Berstel, Dominique Perrin & Christophe Reutenauer (2009): *Codes and Automata (Encyclopedia of Mathematics and Its Applications)*, 1st edition. Cambridge University Press, New York, NY, USA.

[4] 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, doi:10.1016/S0019-9958(85)80027-9.

[5] Manuel Bodirsky, Tobias Grtner, Timo von Oertzen & Jan Schwinghammer (2004): *Efficiently Computing the Density of Regular Languages*. In Martn Farach-Colton, editor: LATIN 2004: Theoretical Informatics, Lecture Notes in Computer Science 2976, Springer Berlin Heidelberg, pp. 262–270, doi:10.1007/978-3-540-24698-5_30.

[6] 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, doi:10.1142/S0129054108005802.

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

[8] 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. Available at `http://www.inf.u-szeged.hu/actacybernetica/edb/vol16n1/Esik_2003_ActaCybernetica.xml`.

[9] Ronald Fagin (1976): *Probabilities on Finite Models*. J. Symb. Log. 41(1), pp. 50–58, doi:10.1017/S0022481200051756.

[10] Philippe Flajolet & Robert Sedgewick (2009): *Analytic Combinatorics*, 1 edition. Cambridge University Press, New York, NY, USA, doi:10.1017/CBO9780511801655.

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

[12] Phokion G. Kolaitis & Moshe Y. Vardi (1992): *Infinitary logics and 01 laws*. Information and Computation 98(2), pp. 258 – 294, doi:10.1016/0890-5401(92)90021-7. Available at `http://www.sciencedirect.com/science/article/pii/0890540192900217`.

[13] 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, doi:10.1007/3-540-44612-5_6. Available at `http://dblp.uni-trier.de/db/conf/mfcs/mfcs2000.html#KolaitisV00`.

[14] Leonid Libkin (2004): *Elements of Finite Model Theory*. SpringerVerlag, doi:10.1007/978-3-662-07003-1.

[15] Per Martin-Löf (1965): *Probability theory on discrete semigroups*. Zeitschrift fr Wahrscheinlichkeitstheorie und Verwandte Gebiete 4(1), pp. 78–102, doi:10.1007/BF00535486.

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

[17] Igor Rystsov (1997): *Reset words for commutative and solvable automata*. Theoretical Computer Science 172(12), pp. 273 – 279, doi:10.1016/S0304-3975(96)00136-3.

[18] Jacques Sakarovitch (2009): *Elements of Automata Theory*. Cambridge University Press, New York, NY, USA, doi:10.1017/CBO9781139195218.

[19] Arto Salomaa & M. Soittola (1978): *Automata Theoretic Aspects of Formal Power Series*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, doi:10.1007/978-1-4612-6264-0.