

# Some more on $\omega$ -finite automata and $\omega$ -regular languages.

## Part I: The main definitions and properties

B. F. Melnikov, A. A. Melnikova

**Abstract**—The finite and infinite iterations of finite and infinite languages arise in various problems of the formal languages theory. For instance, we can mention their application for the description of subclasses of the context-free languages class with the decidable equivalence problem. For infinite iterations of finite languages, we consider in this paper so-called strongly connected omega-automata and corresponding omega-regular languages. For them, many statements are fulfilled that are not satisfied in the more general case, i.e. when we use arbitrary omega-finite automata and corresponding omega-regular languages; the omega-iterations of languages include in the class of strongly connected omega-regular languages. The main of such properties and statements are non-existence of an omega-automaton for which there is no equivalent deterministic, and the possibility of checking the equivalence of two given omega-automata; in the general case (i.e., when we consider arbitrary omega-automata), both of these properties are not satisfied. We also describe transferring the usual procedure of determinization to the case of omega-automata and show the correctness of this procedure in the cases we are considering. We consider some examples, and in the second part of the paper, we shall consider the transfer of a well-known example (so called Waterloo automaton) to the case of omega-automata and omega-languages.

**Keywords**—nondeterministic finite automata, regular languages, omega-computations, omega-automata, omega-regular languages.

### I. INTRODUCTION

The main subject of this paper is the connection between the known facts related to  $\omega$ -regular languages<sup>1</sup> and iterations of languages (usually, infinite iterations of finite languages). However, we also consider some other statements related to  $\omega$ -languages, but not related to iterations of languages.

The iterations of finite and infinite languages arise in various problems of the formal languages theory. We cite only three relevant topics related to previous publications of the authors of this paper.

Received May 21, 2020.

Boris F. Melnikov, Shenzhen MSU–BIT University (email: bf-melnikov@yandex.ru).

Aleksandra A. Melnikova, Dimitrovgrad Engineering and Technology Institute – Branch of National Research Nuclear University “MEPhI” (email: super-avahi@yandex.ru).

<sup>1</sup>“ $\omega$ -regular languages” is an established spelling; the authors do not know how it complies with the rules of classical English. Similarly for “ $\omega$ -finite automata”. Certainly, we shall use this spelling.

However, the same spelling was also established in Russian mathematical literature, which carries some inaccuracy: “ $\omega$ ” is related to the *combination of words* “regular languages” (or “finite automata”), not to the word “regular” (or “finite”), and therefore should be written with a dash (and not with a hyphen).

- The description of subclasses of the context-free languages class with the decidable equivalence problem. See [1] etc.
- The use of iterations of languages for describing the structure of the set of loops of the transition graph of a considered nondeterministic finite automaton and the equivalent basis automaton. See [2], [3], [4] etc.
- The connection between iterations of languages and some problems of the theory of approximation of semi-groups. See [5], [6] etc.

Some clarification of the mentioned problems, as well as other links to our previous works are given below in preliminaries.

Thus, in connection with all of the above, we consider some facts related to  $\omega$ -finite automata<sup>2</sup> and  $\omega$ -regular languages. We specifically note, that the terminology and basic concepts used by our paper are somewhat different from those used in [7].

We can say that the main objects considered in this paper are so-called *strongly connected  $\omega$ -automata* and corresponding  $\omega$ -regular languages. For them, many statements are fulfilled that are not satisfied in the more general case, i.e. when we consider arbitrary  $\omega$ -finite automata and corresponding  $\omega$ -regular languages. It is important to note that the  $\omega$ -iterations of languages (they are of particular interest to us) do include in this class. The main of these properties and statements are non-existence of an  $\omega$ -automaton for which there is no equivalent deterministic, and the possibility of checking the equivalence of two given  $\omega$ -automata. We note again that in the general case (i.e., when we consider arbitrary  $\omega$ -automata), both of these properties are not satisfied.

The structure of the paper is as follows. Sections II, III, and IV includes preliminaries:

- in Section II, we give notation related to the “usual” finite automata and regular languages<sup>3</sup>;
- in Section III, we consider notation related to arbitrary  $\omega$ -languages; we also consider there some facts on  $\omega$ -languages which is necessary for the remained part of the paper;
- and in Section IV, we consider  $\omega$ -automata: among various versions of  $\omega$ -automata, we shall consider so-called Büchi-automata only; moreover, we shall mainly use a special simplification of Büchi-automata, which is sufficient for the purposes of this paper.

<sup>2</sup>Below, we usually shall write simply “ $\omega$ -automata” in the meaning “ $\omega$ -finite automata”.

<sup>3</sup>Below, we almost always shall not use quotes in such context.

In Section V, we consider a special subclass of the  $\omega$ -automata class. We called such objects iterating  $\omega$ -automata, and firstly introduce the definition and some examples for them. Corresponding  $\omega$ -languages are called iterating  $\omega$ -languages. And in Section VII, we introduce special subclasses of the classes of iterating  $\omega$ -automata and  $\omega$ -languages: i.e. the classes of strongly connected  $\omega$ -automata and  $\omega$ -languages.

Before, in Section VI, we describe in details the transferring of the usual procedure of determinization (for usual finite automata) to the case of  $\omega$ -automata; we have to describe it, because we shall use the description in proving some further facts. For instance, in Section VII we prove a fact (Theorem 1), which shows the validity of this procedure for the class of  $\omega$ -finite automata.

We are going to continue the consideration of  $\omega$ -finite automata and corresponding  $\omega$ -regular languages in Part II of this paper. At the end of this part, we present the main results that we are going to publish in Part II.

## II. PRELIMINARIES: THE MAIN NOTATION

For the usual finite automata and regular languages, we shall use notation of our previous papers. At first, see [8], [9]. Almost always, we considered nondeterministic finite automaton designated

$$K = (Q, \Sigma, \delta, S, F). \quad (1)$$

We specially note the following notation: the edge

$$\delta(q, a) \ni q'$$

(for finite automaton  $K$ ) will be denoted

$$q \xrightarrow[K]{a} q', \quad \text{or} \quad q \xrightarrow[\delta]{a} q', \quad \text{or simply} \quad q \xrightarrow{a} q'.$$

We also remark that we shall use designation  $\mathcal{L}^{in}$ , which naturally generalizes to  $\omega$ -automata defined below. For instance, for automaton 1 and its state  $q \in Q$ ,  $\mathcal{L}^{in}(q)$  is simply the language of the automaton

$$(Q, \Sigma, \delta, S, \{q\}).$$

## III. PRELIMINARIES: $\omega$ -LANGUAGES

Among the monographs devoted to  $\omega$ -languages, we note [7] and the classical books [10], [11], [12]. A detailed theory of  $\omega$ -languages is also presented in the classical papers [13], [14], [15], [16]. Note once again, that the terminology and basic concepts used by our paper are somewhat different from those used in [7].

Words over a given alphabet are *finite* sequences of the letters, and an *infinite* a sequence of letters of a given alphabet is called an infinite word or  $\omega$ -word over this alphabet. The set (finite or infinite) of  $\omega$ -words is called an  $\omega$ -language.

The set of all  $\omega$ -words over the alphabet  $\Sigma$  is denoted by  $\Sigma^\omega$ . We shall denote

$$\Sigma^\infty = \Sigma^* \cup \Sigma^\omega.$$

Below, the subsets of the set  $\Sigma^\infty$  will be usually denoted by  $\mathcal{A}$  and  $\mathcal{B}$ .

(Usually, the alphabet  $\Sigma$  is supposed to be finite. However, in some our papers it should be infinite, see at first [17]. See also [1], [18] etc. for some possible applications.)

For some given language  $L \subseteq \Sigma^*$  (“usual” language), we shall define  $\omega$ -languages

$$\lim(L) \quad \text{and} \quad \text{adh}(L).$$

Let us note in advance, that we can consider these  $\omega$ -languages as a connection between languages (subsets of the set  $\Sigma^*$ ) and  $\omega$ -languages (subsets of the set  $\Sigma^\omega$ ).

Thus, let  $\text{pref}(u)$  (where  $u \in \Sigma^*$ ) be the set of all the prefixes of  $u$ ; similarly  $\text{pref}(\alpha)$  (where  $\alpha \in \Sigma^\infty$ ) be the (infinite) set of all the prefixes of  $\alpha$ . For the languages ( $\omega$ -languages), we define

$$\text{pref}(\mathcal{A}) = \bigcup_{\alpha \in \mathcal{A}} \text{pref}(\alpha).$$

Then for the given language  $L \subseteq \Sigma^*$ , we set

$$\lim(L) = \left\{ \alpha \in \Sigma^\omega \mid (\forall k \in \mathbb{N}) (\exists u \in L) (|u| > k, u \in \text{pref}(\alpha)) \right\}$$

and

$$\text{adh}(L) = \left\{ \alpha \in \Sigma^\omega \mid (\forall w \in \text{pref}(\alpha)) (\exists x \in \Sigma^*) (wx \in L) \right\}$$

It is east to prove, that

$$\lim(L) \subseteq \text{adh}(L),$$

but, generally speaking,

$$\lim(L) \neq \text{adh}(L) :$$

the simplest example for the last inequality is the language  $L$ , which can be defined by the regular expression  $a^*b$ ; for it, we have

$$\lim(L) = \emptyset, \quad \text{but} \quad \text{adh}(L) = \{a^\omega\}.$$

We also note the equality

$$\text{adh}(L) = \lim(\text{pref}(L)),$$

which was proved in [12]; in the present paper; however, it does not play an important role for the further consideration.

We also note that for the infinite iterations of languages considered in some our works, the  $\omega$ -languages  $\lim$  and  $\text{adh}$  are one and the same.<sup>4</sup> This result can be considered as a consequence of the fact proved in [19, Th. 3.9].

Various methods are used to specify specific  $\omega$ -languages. Chronologically, the first to be applied the so-called L-systems, including their special cases: OL-systems, DTOL-systems, DOL-systems, etc. Apparently, these terms were introduced since 1968 only, see [20], [21], [22]. However, for example, the widely used *in various areas of mathematics* Thue–Morse  $\omega$ -word was actually described using DOL-systems much earlier, see [23], [24].

L-systems are described in detail in [12], some special cases of L-systems are defined in [25, Chap. 1, 5]. Often, the infinite iterations of languages that we consider cannot be generally defined using DOL systems, and OL-systems are apparently not a very convenient object to use, so we shall only define DTOL systems that we shall use in the future. The definition is given according to [16].

<sup>4</sup>In the continuation of this paper, we are going to consider infinite iterations of finite languages in much more detail.

A DTOL-system is a construction of the type

$$G = (\Sigma, h_1, \dots, h_n, w), \quad (2)$$

where:

- $\Sigma$  is the considered alphabet<sup>5</sup>;
- $n \geq 1$ ;
- each  $h_i$  is a morphism of the type

$$h_i : \Sigma^* \rightarrow \Sigma^*;$$

- $w \in \Sigma^*$  is an “axiom”<sup>6</sup>.

By definition, the language of such DTOL-system (2) (i.e. DTOL-language  $\mathcal{L}(G)$ ) is the set of words

$$\{u \in \Sigma^* \mid (\exists i_1, \dots, i_k \in \overline{1, n}) (u = h_{i_k} \circ \dots \circ h_{i_2} \circ h_{i_1}(w))\}.$$

Let us remark, that for  $n = 1$ , this definition defines a DOL-system; see [25, Chap. 1]. The  $\omega$ -DTOL-language is by definition  $\text{lim}(L(G))$ .

In addition to various L-systems,  $\omega$ -languages can be determined using various versions of  $\omega$ -automata and  $\omega$ -grammars. These methods for defining  $\omega$ -languages have analogies with defining context-free languages by push-down automata and context-free grammars.

#### IV. PRELIMINARIES: $\omega$ -AUTOMATA

Among various versions of  $\omega$ -automata, we shall consider so-called Büchi-automata only. (See [7] etc. For the primary source, see [26].) In fact, a Büchi-automaton can be defined in full accordance with the usual finite automaton: a tuple

$$K = (Q, \Sigma, \delta, S, F). \quad (3)$$

Formally, this coincides with (1); however, we have to make the following additions.

- Certainly, the set of final states  $F$  has a different meaning.
- In our previous papers, we considered the transition function  $\delta$  *without*  $\varepsilon$ -edges, i.e. it is a function of the type

$$\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q). \quad (4)$$

We make similar assumptions in this paper.

- In our previous papers, we sometimes considered  $\omega$ -automaton as a tetrad

$$\mathcal{K} = (Q, \Sigma, \delta, S), \quad (5)$$

but not as the five like (3); see [27], [28] and sections later in this paper.<sup>7</sup>

- In addition, we *have to define* the  $\omega$ -language accepted by the automata (3) and (5), considered as  $\omega$ -automata.

However, we shall not consider a detailed definition, but consider its *simplification*. Thus, given the last things, we shall use the following *modified* definitions.<sup>8</sup>

**Definition 1:** An  $\omega$ -automaton is

$$\mathcal{K} = (Q, \Sigma, \delta, S, F), \quad (6)$$

<sup>5</sup>Above, we briefly discussed the possibility of considering the infinite alphabet.

<sup>6</sup>Here, its use here is similar to that used in mathematical logic, the theory of formal grammars, etc.

<sup>7</sup>Unlike usual finite automata, we shall usually denote  $\omega$ -automata by  $\mathcal{K}$  etc. (not by  $K$ ).

<sup>8</sup>They could be considered as a special case of Büchi-automata.

where<sup>9</sup>:

- $Q$  is the set of states;
- $\Sigma$  is the alphabet;
- $\delta$  is the function of the type (4);
- $S \subseteq Q$  is the set of initial states;
- $F \subseteq Q$  is the set of special states, which could be called as final ones.  $\square$

Such automata are represented in the form of graphs in the usual way. Like our previous papers, we denote below all the states of usual automata (NFA) by *double* circles. (The single circles are used for so-called “proper” GNFPAs only, see [29] for details.) The initial and final states are marked by little input and output arrows respectively.

We specifically simplified two the following definitions a little: for our purposes, such a simplification is sufficient.

**Definition 2:** The  $\omega$ -word

$$a_1 a_2 \dots, \quad \text{where } a_i \in \Sigma \text{ for each } i \in \mathbb{N} \quad (7)$$

is an  $\omega$ -sequence of  $\omega$ -automaton (6), if there exists an infinite sequence of states

$$q_0, q_1, q_2, \dots, \quad \text{where } q_i \in Q \text{ for each } i \in \mathbb{N}_0, \quad (8)$$

for which the following conditions hold:

- $q_0 \in S$ ;
- $(\forall i \in \mathbb{N}) (\delta(q_{i-1}, a_i) \ni q_i)$ .  $\square$

**Definition 3:** The  $\omega$ -word

$$a_1 a_2 \dots, \quad \text{where } a_i \in \Sigma \text{ for each } i \in \mathbb{N}$$

is **accepted** by  $\omega$ -automaton (6), if it is the  $\omega$ -sequence of this  $\omega$ -automaton, and in addition, the following conditions hold:

- there exists  $f \in F$ , such that the sequence (8) contains  $f$  infinitely many times.<sup>10</sup>

The  $\omega$ -language accepting by  $\omega$ -automaton (6) (or simply “ $\omega$ -language of  $\omega$ -automaton”) is the set of  $\omega$ -words accepting by it.  $\square$

Of course, for such definitions we are interested in the usual problems of language theory. First, how to determine the equivalence of two  $\omega$ -automata (that is, do the given  $\omega$ -automata define the same  $\omega$ -language)? Secondly, does there exist a deterministic  $\omega$ -automaton equivalent to a given one?<sup>11</sup> Therefore, we must first determine the deterministic  $\omega$ -automaton, which is done in the usual way.<sup>12</sup>

**Definition 4:** The  $\omega$ -automaton of Definition 1 is **deterministic**, if the following conditions hold<sup>13</sup>:

- $|S| \leq 1$ ;
- $(\forall q \in Q, a \in \Sigma) (|\delta(q, a)| \leq 1)$ .  $\square$

<sup>9</sup>As usual.

<sup>10</sup>We repeat that we deliberately simplified this definition a little. There are many options for accepting an  $\omega$ -word by an  $\omega$ -automaton. The details are given in the cited works related to  $\omega$ -automata. Among them, we especially note the papers of R. Cohen and Y. Gold, that analyze the relationship between some variants of accepting.

<sup>11</sup>In another way, this question can be reformulated as follows. Does there exist a corresponding deterministic  $\omega$ -automaton for any  $\omega$ -language that can be defined by an  $\omega$ -automaton?

<sup>12</sup>This definition is also slightly simplified compared to those given in classical publications.

<sup>13</sup>As usual, we consider  $S$  and each value of the function  $\delta$  as the sets.

A well-known example answers the second question negatively. Namely, let us consider the following automaton (Fig. 1):

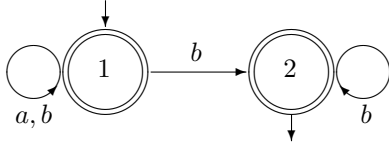


Fig. 1

It is evident, that the  $\omega$ -language of this  $\omega$ -automaton could be determined by the expression<sup>14</sup>

$$(a + b)^* \cdot b^\omega. \quad (9)$$

The *usual* procedure of making the deterministic automaton<sup>15</sup> gives the following  $\omega$ -automaton (Fig. 2):

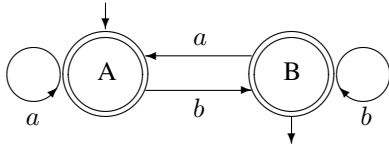


Fig. 2

(where  $A$  corresponds to the set  $\{1\}$ , and  $B$  corresponds to the set  $\{1, 2\}$ ). Certainly, it is not equivalent to the  $\omega$ -automaton of Fig. 1: for instance, the  $\omega$ -language of the last  $\omega$ -automaton contains  $\omega$ -word

$$(ab)^\omega,$$

which is not included in the  $\omega$ -language of the first  $\omega$ -automaton.<sup>16</sup>

Thus, we answered negatively to the second of the questions asked above; that is, not for any  $\omega$ -automaton there exists an equivalent deterministic one. Further we shall consider particular cases in which a different situation is observed.

All this allows us to consider the following definition, or rather, makes it meaningful.

**Definition 5:** The class of **deterministic  $\omega$ -languages** includes only those  $\omega$ -languages, each of which can be accepted by an deterministic  $\omega$ -automaton.  $\square$

In the conclusion of the section, we note once again that the designation  $\mathcal{L}^{in}$  that we shall use in the future is not necessary to redefine for  $\omega$ -automata.

### V. ITERATING $\omega$ -AUTOMATA: THE DEFINITION

In this section, we shall consider a special subclass of the  $\omega$ -automata class.

As we have already noted, in many of our previous works, we considered problems related to iteration of languages, as well as to binary relations related to such iterations of (usually finite) languages. See the papers [1], [5], [6], [17], [18], [19] cited before, and also [31], [32] etc.

<sup>14</sup>We shall not define  *$\omega$ -regular expressions* strictly, we only note that there are several similar approaches to this thing; for instance, see below. Anyway, this expression is certainly clear.

<sup>15</sup>We simply transfer the same procedure to the case of  $\omega$ -automata. More details are below.

<sup>16</sup>We also note that  $(ab)^\omega$  can be considered both as an  $\omega$ -word and as an “regular  $\omega$ -expression”.

A generalization of such iterations to case of  $\omega$ -languages are  $\omega$ -iterations. Above, we defined them using L-systems; now, we shall consider a simpler definition of them. For the following, the iterating set does not contain the empty word (i.e.  $A \not\ni \varepsilon$  when we use notation  $A^\omega$ ); this limitation is because otherwise, the following definition accepts not only  $\omega$ -words, but also usual words. Like alphabet  $\Sigma$ , the set  $A$  is usually finite; however, as we noted before, in [17] we considered cases, when either  $\Sigma$  or  $A$  should be infinite; we also already cited [1], [18] for some possible applications.

**Definition 6:** An **iterating  $\omega$ -language** (of the language  $A$ ) is

$$A^\omega = \prod_{i \in \mathbb{N}} u_i, \quad \text{where } u_i \in A \text{ for each } i \in \mathbb{N}.$$

(Here, we used the natural generalization of the concatenation operation, when its right operand is an  $\omega$ -word.)  $\square$

We also note that according to, for example, already cited papers [13], [14], [22], we can assume that the infinite iterations of languages discussed below are special subsets of the so-called  $\omega$ -deterministic context-free languages class (this is a more general object than the  $\omega$ -languages of the deterministic  $\omega$ -finite automata). However, in our case (i.e. for iterating  $\omega$ -languages) we can consider exactly the (nondeterministic)  $\omega$ -finite automaton defined by the next Definition 7. We note in advance that we strictly define an object schematically depicted on the following Fig. 3.

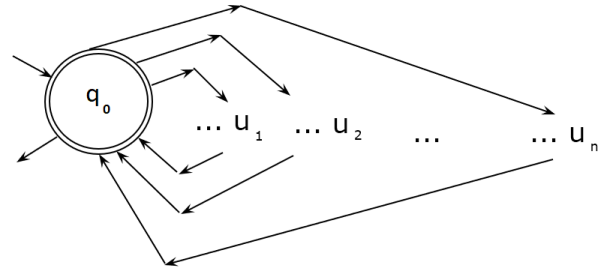


Fig. 3

For Definition 7, we shall use notation

$$\overline{1, n} = \{1, 2, \dots, n\}$$

and set

$$A = \{u_1, u_2, \dots, u_n\}.$$

Let us especially note, that the use of  $u_i$  differs from that used in the previous Definition 6.

**Definition 7:** Let

$$u_i = a_{i,1} a_{i,2} \dots a_{i,l_i},$$

where  $a_{i,j} \in \Sigma$ ,  $l_i \in \mathbb{N}$  for  $i \in \overline{1, n}$ .<sup>17</sup>

Then an **iterating  $\omega$ -automaton** (of the language  $A$ ) is

$$(Q, \Sigma, \delta, \{q_0\}, \{q_0\}),$$

where

$$Q = \{q_0\} \cup \bigcup_{\substack{i \in \overline{1, n}, \\ j \in \overline{1, l_i - 1}}} q_{i,j},$$

<sup>17</sup>Let us note once again, that we do not consider empty words here, i.e. each  $l_i > 0$ .

and the transition function  $\delta$  is defined in the following way.

- For each  $i \in \overline{1, n}$ :
  - $\delta(q_0, a_{i,1}) = \{q_{i,1}\}$ ;
  - for each  $j \in \overline{1, l_i - 2}$ ,  $\delta(q_{i,j}, a_{i,j+1}) = \{q_{i,j+1}\}$ ;
  - $\delta(q_{i,l_i-1}, a_{i,l_i}) = \{q_0\}$ .
- For all remaining pairs  $q \in Q$  and  $a \in \Sigma$ , we set

$$\delta(q, a) = \emptyset. \quad \square$$

In any infinite path of the transition graph of the  $\omega$ -automaton of Definition 7, the state  $q_0$  occurs infinite number of times. Therefore Definitions 6 and 7 do define the same  $\omega$ -language.

#### VI. THE FORMAL DEFINITION OF DETERMINIZATION PROCEDURE IN CASE OF $\omega$ -AUTOMATA

In this section, we consider the usual procedure of determinization; more precisely, we consider its transferring to the case of  $\omega$ -automata. This procedure is valid standard; however, it must be considered in detail for the following three reasons:

- firstly, for usual automata, the different formulations of the corresponding algorithm are given in different monographs (and especially the specific description of such algorithms);
- secondly, we have to describe it, because we shall use the description in proving some facts;
- thirdly, we have to describe it, because not quite standard is working with the set  $F$ : in the usual case, they are used to describe the sets of useless and not-useless states, and in our case, as can be seen from the above examples, it is hardly possible to draw a corresponding analogy.<sup>18</sup>

We note once again that, according to the material of Section IV, that we do not guarantee equivalence of the both  $\omega$ -automata: the given one and the one obtained after the transformation.<sup>19</sup>

Below,  $\mathcal{P}(A)$  denotes the superset for the given set  $A$ .<sup>20</sup> Similar (but not the same) notation will be used for the automata. Thus.

**Definition 8:** Let the  $\omega$ -automaton  $\mathcal{K}$  (6) be given.  $\omega$ -automaton  $\mathcal{P}(\mathcal{K})$  is defined in the following way.

$$\mathcal{P}(\mathcal{K}) = (\mathcal{Q}, \Sigma, \delta_{\mathcal{Q}}, \{S\}, \mathcal{F}),$$

where:

- $\mathcal{Q} = \mathcal{P}(Q)$ ;
- $\delta_{\mathcal{Q}}$  is defined as follows: the edge

$$q \xrightarrow[\delta_{\mathcal{Q}}]{a} q', \quad \text{where } q, q' \in \mathcal{P}(Q), a \in \Sigma$$

exists if and only if

$$(\exists q \in q, \exists q' \in q') (q \xrightarrow[\delta]{a} q');$$

- $\mathcal{F} = \{f \in \mathcal{P}(Q) \mid (\exists f \in F) (f \in f)\}$ .  $\square$

<sup>18</sup>However we shall use the standard (for the usual case) algorithms.

<sup>19</sup>We also note in advance, that in the particular cases of interest to us, such equivalence does hold.

<sup>20</sup>Other papers sometimes use the equivalent notation  $2^A$ .

The following fact is very easily proved<sup>21</sup>: if  $\mathcal{K}$  and  $\mathcal{P}(\mathcal{K})$  are considered as the usual automata, then they are equivalent. However we note once again, that, generally speaking,  $\omega$ -automata  $\mathcal{K}$  and  $\mathcal{P}(\mathcal{K})$  are not equivalent.

Authors hope that the examples of Section IV should not cause any questions; moreover, they can be considered as the examples to the definition given in this section.

#### VII. STRONGLY CONNECTED $\omega$ -AUTOMATA: THE DEFINITION AND THE MAIN PROPERTY

Let us note at first, that the notation of the graph theory is consistent with the classical book [30].

The following definition is very natural. It introduces special subclasses of the classes of  $\omega$ -automata and  $\omega$ -languages considered in the previous section.

**Definition 9:** An  $\omega$ -automaton is called **strongly connected** one, if its transition graph is strongly connected.  $\square$

Certainly, automaton on Fig. 1 is *not* strongly connected.

Obviously, the definitions for both pairs of classes we introduced (i.e. iterating/strongly connected  $\omega$ -languages/ $\omega$ -automata) can be reformulated for “usual” regular languages and nondeterministic finite automata. In this case, the definitions for the corresponding “usual” languages are naturally introduced. Moreover, it can be proved that such definitions are meaningful, i.e. they introduce the own subclasses of the regular languages class.<sup>22</sup> We are going to return to this issue in a future publication.

The following theorem seems to be simple, but the authors did not find it in the available literature.

**Theorem 1:** Let the strongly connected  $\omega$ -automaton  $\mathcal{K}$  (6) be given. Then the  $\omega$ -automata  $\mathcal{K}$  and  $\mathcal{P}(\mathcal{K})$  are equivalent.

*Proof.* We have to prove the following 4 parts:

- 1) each  $\omega$ -sequence of  $\omega$ -automaton  $\mathcal{K}$  belongs to the set of  $\omega$ -sequences of  $\omega$ -automaton  $\mathcal{P}(\mathcal{K})$ ; below, this part will be denoted by “ $\mathcal{K} \Rightarrow \mathcal{P}(\mathcal{K})$  (seq)”;
- 2) each  $\omega$ -sequence of  $\omega$ -automaton  $\mathcal{P}(\mathcal{K})$  belongs to the set of  $\omega$ -sequences of  $\omega$ -automaton  $\mathcal{K}$ ; below, this part will be denoted by “ $\mathcal{K} \Leftarrow \mathcal{P}(\mathcal{K})$  (seq)”;
- 3) the presence an infinite number of occurrences of at least one state included in the set  $F$  in the sequence of states<sup>23</sup> of  $\mathcal{K}$  induces the presence an infinite number of occurrences of at least one state included in the set  $F$  in the sequence of states<sup>24</sup> of  $\mathcal{P}(\mathcal{K})$ ; below, this part will be denoted by “ $\mathcal{K} \Rightarrow \mathcal{P}(\mathcal{K})$  (fin)”;
- 4) the converse to the previous fact; below, this part will be denoted by “ $\mathcal{K} \Leftarrow \mathcal{P}(\mathcal{K})$  (fin)”.

	seq	fin
$\mathcal{K} \Rightarrow \mathcal{P}(\mathcal{K})$	(1)	(3)
$\mathcal{P}(\mathcal{K}) \Rightarrow \mathcal{K}$	(2)	(4)

Thus, let us prove these facts.

<sup>21</sup>Similar statements are included in standard student courses in discrete mathematics, theory of formal languages, etc.

<sup>22</sup>In contrast, for example, to the property of unambiguosity: we can consider unambiguous finite automata, but it makes *no sense* to define “the class of unambiguous regular languages”.

<sup>23</sup>We consider the accepting an  $\omega$ -word by the given  $\omega$ -automaton. I.e., we consider Definition 2 for the given  $\omega$ -word and the  $\omega$ -automaton  $\mathcal{K}$ .

<sup>24</sup>Here, we consider Definition 2 for the given  $\omega$ -word and the  $\omega$ -automaton  $\mathcal{P}(\mathcal{K})$ .

1)  $\mathcal{K} \Rightarrow \mathcal{P}(\mathcal{K})$  (seq).

For this thing, there is enough prove by induction, that for the given  $\omega$ -word (7) and for each  $n \in \mathbb{N}_0$ , the following holds:

$$(\exists q \in Q) (\mathcal{L}_{\mathcal{P}(\mathcal{K})}^{in}(q) \ni a_1 a_2 \dots a_n). \quad (10)$$

Besides, we can assume the existence of the set of states (8) of the automaton  $\mathcal{K}$ .

For the base of induction, we choose  $q_0 = \{S\}$ . Let us prove the step.

We assume (10). By Definition 2, we have

$$q_n \xrightarrow[\delta]{a_{n+1}} q_{n+1}. \quad (11)$$

By Definition 8, we select  $q'$ : namely, we select it using the edge (11). Then we obtain (10) for  $n + 1$  instead of  $n$  and  $q'$  instead of  $q$ , and we can say, that we built the necessary sequence of states for the automaton  $\mathcal{P}(\mathcal{K})$ , Q.E.D.

2)  $\mathcal{K} \Leftarrow \mathcal{P}(\mathcal{K})$  (seq).

Let us reformulate a part of Definition 8 for  $\omega$ -automaton  $\mathcal{P}(\mathcal{K})$ . We have, that for the given  $\omega$ -word (7), there exists an infinite sequence of states

$$q_0, q_1, q_2, \dots, \quad \text{where } q_i \in Q \text{ for each } i \in \mathbb{N}_0, \quad (12)$$

for which the following conditions hold:

- $q_0 = \{S\}$ ;
- $(\forall i \in \mathbb{N}) (\delta(q_{i-1}, a_i) = \{q_i\})$

(we also changed both “ $\in$ ” and “ $\ni$ ” for the equality because of the determinism).

Then there is enough prove by induction, that:

- for the given  $\omega$ -word (7);
- for the corresponding sequence of states

$$q_0, q_1, q_2, \dots \quad (q_i \in Q \text{ for each } i \in \mathbb{N}_0), \quad (13)$$

(we obtained it changing the sequence (8); we can see, that it coincides with the sequence (12) used when proving the direct inclusion);

- and for each  $n \in \mathbb{N}_0$ ,

the following condition holds:

$$(\exists q \in Q) (\mathcal{L}_{\mathcal{K}}^{in}(q) \ni a_1 a_2 \dots a_n). \quad (14)$$

By Definition 2 for automaton  $\mathcal{P}(\mathcal{K})$ , we have its edge

$$\delta(q_i, a_{i+1}) = \{q_{i+1}\}$$

for the letter  $a_{i+1}$  of the given  $\omega$ -word (7).

By Definition 8 for automaton  $\mathcal{P}(\mathcal{K})$ , we obtain existing an edge of the type (11), where

$$q_n \in q_n$$

(it is a consequence of the induction hypothesis), and also

$$q_{n+1} \in q_{n+1}.$$

The last fact proves the induction step, Q.E.D.

3)  $\mathcal{K} \Rightarrow \mathcal{P}(\mathcal{K})$  (fin).

Let us consider some  $\omega$ -word of  $\omega$ -automaton  $\mathcal{K}$ . By Definition 3, the corresponding sequence of states (8) has the following property: for some  $f \in F$ , this sequence includes a subsequence (let it be the sequence  $\sigma$ ), which includes  $f$  infinitely many times. For each such inclusion, we have a corresponding state of the sequence (13); and because this

state (considered as a set of states of  $Q$ ) includes  $f$ , it belongs to  $\mathcal{F}$ . Thus, each state of  $\sigma$  corresponds to a state of  $\mathcal{F}$ ; then, because  $\sigma$  is infinite, there exists a state of  $\mathcal{F}$  which enters  $\sigma$  an infinite number of times. Then, by Definition 8, the considered  $\omega$ -word belongs to  $\omega$ -language of the automaton  $\mathcal{K}$ . Q.E.D.

4)  $\mathcal{K} \Leftarrow \mathcal{P}(\mathcal{K})$  (fin).

Let us consider some  $\omega$ -word of  $\omega$ -automaton  $\mathcal{K}$ . By Definitions 3 and 8, the corresponding infinite sequence of states (13) contains infinitely many states  $f \in \mathcal{P}(Q)$ , such that  $f \ni f$  for some  $f \in F$ . If for each  $f \in F$ , the corresponding sequence of the states of  $\mathcal{K}$  contains the finite number of  $f$  only, then for none  $f \in \mathcal{F}$ , we could not form the infinite number of  $f$  in the sequence (13), and this contradicts the possibility of accepting the considered  $\omega$ -word by the given  $\omega$ -automaton  $\mathcal{K}$ . Q.E.D.  $\square$

To conclude this section, let us give the following natural definition.

**Definition 10:** The  $\omega$ -language is called **strongly connected** if it can be accepted by an strongly connected  $\omega$ -automaton.  $\square$

As we saw from the previous exposition, this definition is meaningful, i.e. the subclass of strongly connected  $\omega$ -languages is the own subclass of the  $\omega$ -languages class. However, we have not yet answered the question whether this subclass includes any deterministic  $\omega$ -regular language.

## ON PART II OF THIS PAPER

In Part II of this paper, we shall continue the consideration of the subclasses of  $\omega$ -regular languages and  $\omega$ -finite automata classes we have defined in Part I. For instance, we shall consider an effective algorithm for determining the fact that some  $\omega$ -language is the iterating one. We also shall consider the transfer of a well-known example (so called Waterloo automaton) to the case of  $\omega$ -automata and  $\omega$ -languages.

## REFERENCES

- [1] Melnikov B., Kashlakova E. Some grammatical structures of programming languages as simple bracketed languages // Informatica (Lithuanian Academy of Sciences). – 2000. – Vol. 11. No. 4. – P. 441–454.
- [2] Melnikov B., Melnikova A. Edge-minimization of non-deterministic finite automata // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 2001. – Vol. 8. No. 3. – P. 469–479.
- [3] Melnikov B., Melnikova A. Extended basis finite automaton. Part I. The basic definitions // International Journal of Open Information Technologies. – 2019. – Vol. 7. No. 10. – P. 1–8. (in Russian)
- [4] Melnikov B., Melnikova A. Extended basis finite automaton. Part II. Description of auxiliary languages and some properties // International Journal of Open Information Technologies. – 2020. – Vol. 8. No. 5. – P. 1–7. (in Russian)
- [5] Dang V., Korabel'shchikova S., Melnikov B. Semigroups approximation with respect to some ad hoc predicates // Arctic Environmental Research. – 2017. – Vol. 17. No. 2. – P. 133–140.
- [6] Melnikov B., Korabel'shchikova S., Dolgov V. On the task of extracting the root from the language // International Journal of Open Information Technologies. – 2019. – Vol. 7. No. 3. – P. 1–6.
- [7] Khousainov B., Nerode A. Automata Theory and its Applications // Progress in Computer Science and Applied Logic, Vol. 21. – Springer, Berlin, 2001.
- [8] Melnikov B. Once more on the edge-minimization of nondeterministic finite automata and the connected problems // Fundamenta Informaticae. – 2010. – Vol. 104. No. 3. – P. 267–283.
- [9] Melnikov B. The complete finite automaton // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.

- [10] Eilenberg S. Automata, Languages and Machines. Vol. A // Academic Press, N.Y., 1974.
- [11] Eilenberg S. Automata, Languages and Machines. Vol. B // Academic Press, N.Y., 1976.
- [12] Rozenberg G., Salomaa A. The Mathematical Theory of L Systems // Academic Press, N.Y., 1980.
- [13] Cohen R., Gold Y. Theory of  $\omega$ -Languages. I: Characterizations of  $\omega$ -Context-Free Languages // Journal of Computer and System Sciences. – 1977. – Vol. 15. – P. 169–184.
- [14] Cohen R., Gold Y. Theory of  $\omega$ -Languages. II: A Study of Various Models of  $\omega$ -Type Generation and Recognitions // Journal of Computer and System Sciences. – 1977. – Vol. 15. – P. 185–208.
- [15] Čulik II K., Pachi J. Equivalence Problems for Mapping on Infinite Strings // Information and Control. – 1981. – Vol. 49. – P. 52–63.
- [16] Čulik II K., Salomaa A. Test Sets and Cheking Words for Homomorphism Equivalence // Journal of Computer and System Sciences. – 1980. – Vol. 20. – P. 379–395.
- [17] Brosalina A., Melnikov B. Commutation in global supermonoid of free monoids // Informatica (Lithuanian Academy of Sciences). – 2000. – Vol. 11. No. 4. – P. 353–370.
- [18] Dubasova A., Melnikov B. On an extension of the context-free languages class // Programming and Computer Software (Russian Academy of Sciences). – 1995. – No. 6. – P. 46–55. (in Russian)
- [19] Melnikov B. An algorithm for checking the equality of infinite iterations of finite languages // Bulletin of Moscow University, series Computational Mathematics and Cybernetics. – 1996. – No. 4. – P. 49–54. (in Russian)
- [20] Lindenmayer A. Mathematical models for cellular interaction in development. I // Journal of Theoretical Biology. – 1968. – Vol. 18. – P. 280–298.
- [21] Lindenmayer A. Mathematical models for cellular interaction in development. II // Journal of Theoretical Biology. – 1968. – Vol. 18. – P. 299–315.
- [22] Wood D. Grammar and L Forms: an Introductions // Springer, Berlin, 1980.
- [23] Thue A. Über unendlichen Zeichenreihen // Math.-Naturv. Klasse Skrifter udgivne af Videnskabsselskabet i Christiania. – 1906. – P. 1–22. (in German)
- [24] Thue A. Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen // Math.-Naturv. Klasse Skrifter udgivne af Videnskabsselskabet i Christiania. – 1912. – P. 1–35. (in German)
- [25] Salomaa A. Jewels of Formal Language Theory // Computer Science Press, Rockville, 1981.
- [26] Büchi J. On a decision method in restricted second order arithmetic // In: Proceedings of International Congress on Logic, Method, and Philosophy of Science. – Stanford University Press, Stanford, 1962.<sup>25</sup> – P. 425–435.
- [27] Melnikov B. On  $\omega$ -languages of special billiards // Discrete Mathematics (Russian Academy of Sciences). – 2002. – No. 3. – P. 95–108. (in Russian)
- [28] Melnikov B. On  $\omega$ -languages of special billiards // Discrete Mathematics and Applications. 2002. – Vol. 12. No. 5. – P. 501–514.
- [29] Melnikov B., Melnikova A. Pseudo-automata for generalized regular expressions // International Journal of Open Information Technologies. 2018. – Vol. 6. No. 1. – P. 1–8.
- [30] Harary F. Graph Theory // Addison-Wesley, Reading, 1969.
- [31] Melnikov B. The equality condition for infinite catenations of two sets of finite words // International Journal of Foundation of Computer Sciences. 1993. – Vol. 4. No. 3. – P. 267–274.
- [32] Melnikov B. The star-height of a finite automaton and some related questions // International Journal of Open Information Technologies. 2018. – Vol. 6. No. 7. – P. 1–5.

**Boris Feliksovich MELNIKOV,**  
 Professor of Shenzhen MSU – BIT University, China  
 (<http://szmsubit.ru/>),  
 Professor of Russian State Social University  
 (<http://www.rgsu.net/>),  
 email: [bf-melnikov@yandex.ru](mailto:bf-melnikov@yandex.ru),  
 mathnet.ru: personid=27967,  
 elibrary.ru: authorid=15715,  
 scopus.com: authorId=55954040300,  
 ORCID: orcidID=0000-0002-6765-6800.

**Aleksandra Aleksandrovna MELNIKOVA,**  
 Associated Professor of  
 Dimitrovgrad Engineering and Technology Institute –  
 Branch of National Research Nuclear University “MEPhI”  
 (<https://diti-mephi.ru/>),  
 email: [super-avahi@yandex.ru](mailto:super-avahi@yandex.ru),  
 mathnet.ru: personid=148963,  
 elibrary.ru: authorid=143351,  
 ORCID: orcidID=0000-0002-1658-6857.

<sup>25</sup>The congress was held in 1960. This sometimes leads to incorrect citation of this paper in some sources, as well as to an incorrect indication of the time of “creation” of considered automata and the superiority in their description.