Journal of Discrete Mathematics and Its Applications 9 (2) (2024) 81–102

Journal of Discrete Mathematics and Its Applications

Available Online at: http://jdma.sru.ac.ir

Shahid Rajaee Teacher Training University

Research Paper

On an interesting hypothesis of the theory of formal languages

Boris Melnikov *

Shenzhen MSU-BIT University

Academic Editor: Majid Arezoomand

Abstract. The formulation of a hypothesis for any pair of nonempty finite languages is considered. The hypothesis consists in the formulation of the necessary conditions for the equality of infinite iterations of these languages, the paper provides some equivalent versions of this hypothesis. When fulfilling this hypothesis, we show the possibility of verifying the equality of infinite iterations of these languages in polynomial time. On the other hand, we present a plan for reducing the verification of the same equality to checking the completeness of the language of a specially constructed nondeterministic finite automaton, and such a check cannot be carried out in polynomial time. In this regard, the possibility of reducing the equality P=NP to the special hypothesis of the theory of formal languages is formulated.

Keywords. Formal languages, iterations of languages, binary relations, morphisms, inverse morphisms, algorithms, polynomial algorithms.

Mathematics Subject Classification (2010): 68R01, 68R99.

1 Introduction and motivation

The subject of this paper has the following two preambles, they are almost independent of each other.

The first preamble. In the 1990s, the author was engaged in describing subclasses of a class of context-free languages with a decidable equivalence problem. At that time, it was significant that such subclasses seemed important for the translation process (i.e., for creating

*Corresponding author (*Email address*: bormel@mail.ru)
Received 17 April 2024; Revised 25 April 2024; Accepted 17 May 2024
First Publish Date: 01 June 2024

 ©Shahid Rajaee Teacher Training University

automation systems for building translators), and also that the description of such subclasses should be as little as possible related to deterministic context-free languages (for which the equivalence problem was not yet solved at that time). Such subclasses were described by the author in two different publications published in the journals of Moscow State Lomonosov University: the first one was "more theoretical" [1], and the second one was "more practical" [2]. The material of the first paper was not subsequently published in English: primarily because the solvability of the equivalence problem in the class of deterministic CF-languages was already proved ([3], the partially proof was published some before), and the work on the description of alternative subclasses began to be of much less interest.

However, in the process of working on this topic, the author considered an auxiliary problem; it was a problem about the necessary and sufficient conditions for the equivalence of the so-called bracketed languages; note that some new results on this topic were published recently in [4], there is also a detailed definition of the mentioned bracket languages. And for it, another problem was solved, i.e. an "auxiliary for auxiliary" problem, and, as usual in such situations, the last problem has a very simple formulation. Briefly, this simple formulation can be stated as follows: it is necessary to formulate necessary and sufficient conditions for two finite formal languages having the same (infinite) set of infinite iterations corresponding to them (we conditionally call equivalence in the infinity of finite languages), see [5].[a] But shortly after that publication, the author discovered an incompletely proven fact (about it see below, Section 3); and therefore, we did not use in the next papers the results of [5], we used some special cases of this problem only.

The second preamble. Regardless of the search for necessary and sufficient equivalence conditions in the infinity of finite languages, the task of creating algorithms to verify this equivalence was solved. To describe such algorithms, 8 variants of finite automata were used, see [6, 7], and the two simplest of them are of the greatest interest. Those are so-called PRI and $NSPRI$ automata, see [8, 9] for them. It turned out that when fulfilling the hypothesis under consideration, we can describe a polynomial algorithm describing the operation of one of these automata. Therefore, if it is possible to prove the impossibility of such a polynomial algorithm in case of non-fulfillment of this hypothesis, then we will get a reduction of the well-known problem $P = NP$ to the hypothesis under consideration.

Thus, it is the hypothesis that is considered in this paper.

The paper has the following structure.

Section *2* discusses the basic designations and conventions for their use; some of these designations are non-standard. One of the most important concepts defined in this section is the binary relation $\leqslant|\geqslant$ defined on a set of finite languages over the considered alphabet, the whole material of the paper is connected with it.

In *Section 3* and some next sections, we consider the mentioned hypothesis. This hypothesis can be formulated very briefly as follows: infinite iterations of finite languages are the

[a] In 1991–93, the author had a long correspondence with professor Rohit Jivanlal Parikh (https://en.wikipedia.org/wiki/Parikh%27s_theorem, https://en.wikipedia.org/wiki/Rohit_Jivanlal_Parikh, etc.). Based on the results of this correspondence, the paper [5] was published.

same if and only if these languages can be represented as the same morphism of extended maximal prefix codes over an auxiliary alphabet. Particularly in Section 3, we consider some our previous results. Then *in Section 4*, we consider some main variants of its formulation; these variants do not use automata and trees.

Section 5 briefly discusses non-deterministic finite automata of a special kind, so-called petal automata (or semi-flower automata). We present the results of one of our previous papers, which can be formulated as follows: for any regular language and its table of the binary relation # (for more information about this relation, see Section 2), there is an algorithm for constructing a petal automaton having either the same table or such a table with an added column (i.e., with an added state in the canonical automata for the mirror language). Specifically in this paper, petal automata are used for two more reformulations of the main hypothesis considered in the paper, see Section *6*. Thus, the possible whole title of Section 6 could be as follows: "The auxiliary variants of the formulation using infinite trees". We use petal finite automata, as well as infinite iterative trees.

The above-mentioned plan of the reduction of $P = NP$ equality is given in Conclusion, *Section 7*. A more exact title for this section could be as follows: "On the plan of proving the possibility of reducing the equality of $P = NP$ to a hypothesis of the theory of formal languages" (meaning the hypothesis ($\exists\not\vee$)).

2 Preliminaries

This section discusses the basic designations and conventions for their use. It should be also noted that some of these designations are non-standard. One of the most important concepts defined in the section is the binary relation $\leqslant|\geqslant$ defined on a set of finite languages over the considered alphabet, the entire material of this paper is connected with it.

Thus, let us go directly to the notation.

We shall most often consider words and languages over the "main" alphabet Σ; the "auxiliary" alphabet will usually be Δ, sometimes with subscripts; all alphabets are always finite.

For the given word $u \in \Sigma^*$ and language $A \subseteq \Sigma^*$:

- the language $pref(u)$ is the set of prefixes of the word u (including u);
- $opref(u) = pref(u) \setminus \{u\}$;
- $pref(A) = \bigcup\limits_{u \in A} pref(u)$;
- $opref(A)$ is defined similarly.

Similarly for suffices, $suff$ and $osuff$.

If for two finite languages A and B (in our papers, they are most often considered over the "main" alphabet Σ), the condition

$$\big(\forall\, u \in A^*\big)\ \big(\exists\, v \in B^*\big)\ \big(u \in opref(v)\big),$$

is met, then we shall write $A \leqslant| B$ (or $B |\geqslant A$). If the conditions $A \leqslant| B$ and $A |\geqslant B$ are met simultaneously, then we shall write $A \leqslant|\geqslant B$.

(We shall specially note the difference in notation: in our recent papers on the one hand, and, on the other hand, in [5] and some other papers of the 1990s. Certainly, in this paper we shall use the notation given in this section only.)

Now we give some notation related to nondeterministic finite automata.

$$K = \left(Q, \Sigma, \delta, S, F \right) \tag{1}$$

is some automaton. δ is its transition function of the type

$$\delta : Q \times \Sigma \to \mathcal{P}(Q),$$

but not of the type

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q),$$

where the notation $\mathcal{P}(Q)$ denotes the superset (the power set) of the set Q; thus, we consider automaton without ε-transitions. We shall sometimes write some edge $\delta(q,a) \ni r$ in the form $q \xrightarrow[\delta]{a} r$, or, if it does not cause ambiguity, simply in the form $q \xrightarrow{a} r$.

The mirror automaton for the automaton of (1), i.e.,

$$\left(Q, \Sigma, \delta^R, F, S\right),$$

where

$$q' \xrightarrow[\delta^R]{a} q'' \quad \text{if and only if} \quad q'' \xrightarrow[\delta]{a} q',$$

will be denoted by K^R; note that K^R defines the mirror language L^R.

For automaton (1), the output language of the state q (denoted by $\mathcal{L}_K^{out}(q)$) is the language of the automaton

$$\left(Q, \Sigma, \delta, \{q\}, F \right).$$

Similarly, the input language of the state q (denoted by $\mathcal{L}_K^{in}(q)$) is the language of the automaton

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

For the considered language L, its automaton of canonical form will be denoted as $\widetilde{L}$. Let automata $\widetilde{L}$ and $\widetilde{L^R}$ for the given language L be as follows:

$$\widetilde{L} = \left(Q_\pi, \Sigma, \delta_\pi, \{s_\pi\}, F_\pi\right) \quad \text{and} \quad \widetilde{L^R} = \left(Q_\rho, \Sigma, \delta_\rho, \{s_\rho\}, F_\rho\right).$$

Moreover, we do not consider the language $L = \varnothing$, so both these automata *do have* initial states.

Relation $\# \subseteq Q_\pi \times Q_\rho$ is defined for pairs of states of automata $\widetilde{L}$ and $\widetilde{L^R}$ in the following way: $A \# X$ if and only if

$$\left(\exists uv \in L\right) \left(u \in \mathcal{L}_{\widetilde{L}}^{in}(A), v^R \in \mathcal{L}_{\widetilde{L^R}}^{in}(X)\right).$$

Note that such a definition is non-constructive; however, for example, [10] contains the equivalent constructive variant.

We consider deterministic finite automata as a special case of non-deterministic ones defined according to (1). We shall not give detailed definitions, they are standard; we only note that in our paper, so-called total automata will be specially used, i.e. those for which the following condition is met:

$$\left(\forall q \in Q\right)\ \left(\forall a \in \Sigma\right)\ \left(|\delta(q,a)| = 1\right).$$

At the same time, in some publications, this totality is considered as a mandatory property of a deterministic automaton, but we do not do this, and consider total automata to be the own subset of deterministic ones.

In this paper, we shall also use so-called petal (semi-flower) finite automata; see details in Section 5, where automata of the type $\mathcal{K}(A)$ for the given finite language $A \subseteq \Sigma^*$ are also defined. For some more information about notation on the finite automata, see [10].

Further, we believe that all the languages under consideration are not empty and do not contain ε. For some language

$$A \subseteq \Sigma^*, \quad A = \left\{ u_1, u_2, \ldots u_n \right\}$$

(we can assume that the words of the language are somehow ordered), we consider alphabet

$$\Delta_A = \left\{ d_1, d_2, \ldots d_n \right\}$$

(if it does not cause ambiguities, we usually write simply Δ). For this alphabet, we consider the morphism of the type

$$h_A : \Delta_A^* \to \Sigma^*$$

defined as follows:

$$h_A(d_1) = u_1\,,\ h_A(d_2) = u_2\,,\ \ldots\,,\ h_A(d_n) = u_n\,.$$

As usually, for each word $d_1 d_2 \ldots d_n \in \Delta^*$, we assume that

$$h(d_1 d_2 \ldots d_n) = h(d_1)\, h(d_2) \ldots h(d_n).$$

The subject of this paper is related to a more important and more complex problem (i.e., more complex than the construction of a morphism), namely, to the problem of constructing an *inverse morphism*. In [11–13], the beginning of consideration of this problem is given.

Let us formulate its condition in more detail. First, let us give such a natural definition. For a given finite language A, morphism h_A and some word $u \in \Sigma^*$, let us consider the *language*

$$h_A^{-1}(u) = \left\{ u_\Delta \in \Delta^* \,\middle|\, h_A(u_\Delta) = u \right\};$$

we specifically emphasize that this object is a *set*, possibly $\varnothing$. The same construction can be considered for some language (instead of the word u; let this language be B), and we shall be interested only in finite languages. Exactly,

$$h_A^{-1}(B) = \bigcup_{u \in B} h_A^{-1}(u).$$

Now let us consider another definition, almost unrelated to the previous one. Unlike other definitions of this paper, it uses the alphabet Δ as the "main" one. Over this alphabet, the set of maximal prefix codes (as a set of languages) will be denoted by $mp(\Delta)$.

(We shall not give definition of a maximal prefix code. Suffice it to say that this is a code that is maximal and prefix, all these concepts are present in the "usual student courses", i.e. the title already contains a definition.)

The set of languages, each of which *contains* some maximum prefix code *as a subset* (possibly improper one), will be denoted by $mp^+(\Delta)$. Thus,

$$mp(\Delta) \subset mp^+(\Delta),$$

and, certainly, for any alphabet, the inclusion is proper. We shall call each of these languages by an *extended maximal prefix code*.

Let us return to the consideration of the "main" alphabet Σ, as well as morphisms of the form

$$h_A : \Delta_A^* \to \Sigma^*.$$

Let we have some language

$$A_\Delta \in mp(\Delta);$$

then we assume that the following condition is met:

$$h_A(A_\Delta) \in mp(A);$$

this is how we define the set of languages $mp(A)$ over Σ. Similarly, for some language

$$A_\Delta \in mp^+(\Delta)$$

we assume that

$$h_A(A_\Delta) \in mp^+(A).$$

Therefore:

- $mp^+(\Delta)$ is the set of languages, each of which contains some maximal prefix code over Δ as a subset;
- $mp^+(\Delta)$ is the set of languages, each of which is the A-morphism of some language of the set $mp^+(\Delta)$; i.e., each of such languages is the special morphism of some extended maximal prefix code.

3 The results of previous publications

In this and some next sections, we consider the hypothesis of the theory of formal languages, which we consider very important. Very briefly this hypothesis can be formulated as follows: infinite iterations of finite languages are the same if and only if these languages

an be represented as the same morphism of extended maximal prefix codes over an auxiliary alphabet. By the other words:

the obvious *sufficient* condition for fulfilling the relation $\leqslant|\geqslant$ is *necessary and sufficient*.

Particularly in this section, we consider some our previous results.

There is the following preamble, see also Introduction: to reformulate the equality condition $P = NP$, an author's previously published result is needed, see [5] (1993). However, *we currently do not consider this result be proven*. Specifically, using the terminology of this paper, we are talking about the *necessary* condition for the binary relation $A \leqslant|\geqslant B$ to be fulfilled.

Here are specific "claims" to *author's* paper [5]. Certainly, condition (6.2) contains a misprint, it should mean the following:

$$A^* \equiv \infty\, \mathcal{D} \cdot D^*;$$

note that we *cannot* rewrite this condition using the terminology of the current paper, i.e., the using binary relation $\leqslant|\geqslant$. There is also a typo in the formulation of Lemma 6: the initial condition must be $v \in A^*$.

(It is worth noting that $\mathcal{D}$ is some *assumed* language, we tried to prove it in [5] its *nonexistence*.)

However, of course, both of the above "claims" are not errors. We shall not specify a specific error (it is long, difficult and unnecessary to explain it), but instead note *the presence of a counterexample* to the provable in [5, Th. 1] condition $\mathcal{D} \leqslant|\geqslant A$ (here, for ease of reading, we again use the terminology and designations of the current paper, not of [5]).

Thus, the counterexample we need can be obtained as follows For the pair of languages

$$A = \{\, aaa, aabba, abba, bb \,\} \quad \text{and} \quad B = \{\, aaaa, abb, abba, bbb \,\},$$

we shall consider "significantly more complicated" languages

$$A' = (A \cup B)^2 \quad \text{and} \quad B' = (A \cup B)^3, \tag{2}$$

for the languages A and B. Using the previous results of [8,9], we obtain that the following facts hold.

- On the one hand, for a pair of languages (2), this resulting set of languages includes, among others, the language $\{a\}$, and, therefore, the word a is one of the words of the language $\mathcal{D}$, constructed for this pair of languages in the proof of [5, Th. 1];
- But on the other hand, the two languages (2) satisfy binary relations

  $$(A \cup B)^2 \leqslant|\geqslant (A \cup B)^3 \leqslant|\geqslant (A \cup B);$$

  at the same time, the condition

  $$(A \cup B) \leqslant|\geqslant (A \cup B \cup \{a\})$$

is false: for any $k \geqslant 1$, the word a^k is not included in the language $(A \cup B)^*$. However, based on the examples discussed in [8,9], we know that the word a (respectively, the language $\{a\}$) is obtained during the construction of the automaton $NSPRI(A,B)$ (respectively, of the automaton $PRI(A,B)$); therefore the same language $\{a\}$ must be included in the set of states of the automaton

$$PRI(A',B') = PRI(\,(A \cup B)^2, (A \cup B)^3\,).$$

The contradiction obtained here is the necessary counterexample to the statement [5, Th. 1]. However, we specifically note once again that we are not claiming that the theorem is incorrect: we have given a counterexample not to its formulation, but to the method of its proof.

As the conclusion of the section, let us note the following. Let some language B be fixed now; we can assume that *all* words included in the set of proper suffices of the language B belong to the set of states of a nondeterministic automaton $NSPRI(A,B)$, moreover, it is true for each finite language A. Therefore, all *subsets* of the set of such words belong to the set of states of a deterministic automaton $PRI(A,B)$. But, on the other hand, in real situations we can consider a significantly smaller number of states for both of these automata: in particular, in our examples the language

$$B = \{\, aaaa, abb, abba, bbb \,\}$$

has the following proper suffices of its words:

$$a, aa, aaa, b, ba, bb, bba$$

(in the alphabetical order). There are only 7 words, then we can formally assume that the number of states of the automaton PRI is equal to $2^7 = 128$.

4 The main variants of the formulation

... Thus, we have given a counterexample to one of the auxiliary proofs given in [5]. However, it is important to note that this is really *a counterexample to the proof given, but not at all to the statement being proved there*. And it will not change the situation at all (it "will not worsen" and "will not improve" the situation), if we instead of counterexample (i.e., "black box"), will answer the same question with the help of the "white box", i.e., if we shall indicate a specific claim to the proof. However, this is not necessary for the purposes of this paper.

All this allows us to formulate the statement under consideration *in the form of a hypothesis*, and further in this section such a formulation will be given. We believe that such a statement is a very important hypothesis of *the whole theory of formal languages*. (In [13], there are formulated two hypotheses, they are indicated below by (∃⁄) with strokes and (∃⁄∃⁄). The (∃⁄) with strokes are various options of the same hypothesis, and the (∃⁄∃⁄) can be considered as an "special option" of (∃⁄).)

Hypothesis $\nexists$ **.** For any finite languages $A, B \subseteq \Sigma^*$, the following fact holds. The equivalence $A \leqslant\!|\!\geqslant B$ is true if and only if there exists a finite language $D \subseteq \Sigma^*$, such that considering corresponding alphabet Δ (where $|\Delta| = |D|$), there exist extended maximal prefix codes

$$A_\Delta, B_\Delta \subseteq \Delta^*,$$

such that

$$A = h_D(A_\Delta), \;\; B = h_D(B_\Delta).$$

We shall briefly denote this hypothesis by ($\nexists$). □

Here are some *equivalent formulations* of the same hypothesis ($\nexists$); the equivalence follows from the material presented in this paper.

Hypothesis $\nexists'$**.** Let some finite language $D \subseteq \Sigma^*$ be given; at the same time, D is minimal in its own equivalence class. This means, that there are no language $\mathcal{D} \subseteq \pm^*$, such that $D \in mp^+(\mathcal{D})$. (When formulating the main version of this hypothesis, i.e. ($\nexists$), such an additional requirement is not necessary. Remark also that using one of the partial orders on the set of languages described in [12, Sect. XI], another explanation of minimality is also possible.)

Each finite language $A \subseteq \Sigma^*$ belonging to considered equivalence class of $A \leqslant\!|\!\geqslant D$, *can be constructed in the following* 4 *steps*.

1. We choose some alphabet Δ, such that $|\Delta| = |D|$.
2. We consider some language $A'_\Delta \in mp(\Delta)$.
3. We construct any language $A_\Delta \subseteq \Delta^*$ (certainly, depending on the given A), such that $A_\Delta \supseteq A'_\Delta$; i.e., informally, we add arbitrary words to the language A'_Δ; formally, $A_\Delta \in mp^+(\Delta)$.
4. We construct the language $h_D(A_\Delta)$.

The constructed language $h_D(A_\Delta)$ has to coincide with the given A. □

The next version of the same hypothesis we have actually already used above.

Hypothesis $\nexists''$**.** The proposition formulated in [5, Th. 1] is true, i.e., the sufficient condition of equivalence $A \leqslant\!|\!\geqslant B$ formulated there is necessary and sufficient one. □

Now, we formulate an equivalent version of the hypothesis based on its possible non-fulfillment. To do this, we shall answer the question of how to successfully formulate what "can happen" with this *non*-fulfillment. Apparently, it is best to answer this question based on the formulation ($\nexists'$). Since *in a potential counterexample*, not all words of the language A belong to *minimal* set (language) D^*, then there exists some minimal language in its equivalence class D (let us assume that this language sets the equivalence class we are considering) and a word *not* belonging to the set D^* (let it be $u \in \Sigma^*$), such that

$$(D \cup \{u\}) \leqslant\!|\!\geqslant D$$

(or, equivalently, $(D \cup \{u\}) \leqslant| D$ here). Therefore, we formulate the hypothesis simply as *nonexistence* of such a pair.

Hypothesis $\nexists'''$. There exist *no pair* (D,u), where

$$D \subseteq \Sigma^* \quad \text{and} \quad u \in \Sigma^*,\ u \notin D^*,$$

such that:

- the language D is the minimal language in its equivalence class (see before some comments about such minimality);
- $(D \cup \{u\}) \leqslant|\geqslant D$. □

We shall mostly consider the last version of the hypothesis, i.e., $(\nexists''')$. It is clear that we *know no* such pair (u,D), but we also do not know the proof of the non-existence of such a pair (otherwise, all this could not be called a hypothesis); however, now *we assume that such a pair exists*, and we shall consider the corresponding language D (we repeat once again that it is the minimal one of its equivalence class) and the word u that does not belong to the language D^*.

Within the assumptions made, it is not difficult to verify the following fact formulated below in this paragraph. Let v be *any* word of the language D. Then we "remove" v from D and "add" the previously marked word u instead, denoting the resulting language D_{u-v}. Then it is false that

$$D \leqslant|\geqslant D_{u-v}\,. \tag{3}$$

However, this fact does follow:

- neither opportunity,
- nor impossibility

of *existence of several* "violations" of the hypothesis ($\nexists$), *for the same* language D. Moreover, we cannot say that a situation is impossible when the "substitution" of some words of the language D (not less than 1 words) for some other words (each of which *violates* the hypothesis $(\nexists''')$ as the word u, and, therefore, violates also the main formulation, i.e., the hypothesis ($\nexists$)) gives the language belonging to the equivalence class by the relation $\leqslant|\geqslant$, i.e., the class formed by the considered language D. The equivalence similar to (3) can be write as follows:

$$D \leqslant|\geqslant D_{F-G}$$

(where G is the set of words to be deleted, and F is the set to be added).

Let us formulate all this in the form of the following hypothesis ($\nexists\nexists$).

Hypothesis $\nexists\nexists$. The ($\nexists$) hypothesis does not hold. However, at the same time *there not exist* the finite language $D \subseteq \Sigma^*$, for which the following holds.

First, for this language D the hypothesis ($\nexists$) *is not executed* for each of the following words:

$$u_1, u_2, \ldots, u_n$$

(according to the above explanations, not only the inequality $n \geqslant 1$ is fulfilled, but also $n \geqslant 2$; in addition, none of these words are included in the language of D^*).

Secondly, for these languages D and F there exists some nonempty set $G \subseteq D$, such that

$$D \not\leqslant\!|\!\geqslant D_{F-G}.$$

We shall be briefly denote this hypothesis by the notation ($\nexists/\nexists$). □

The inclusions of the two formulated hypotheses can be depicted in the following Fig. 1..

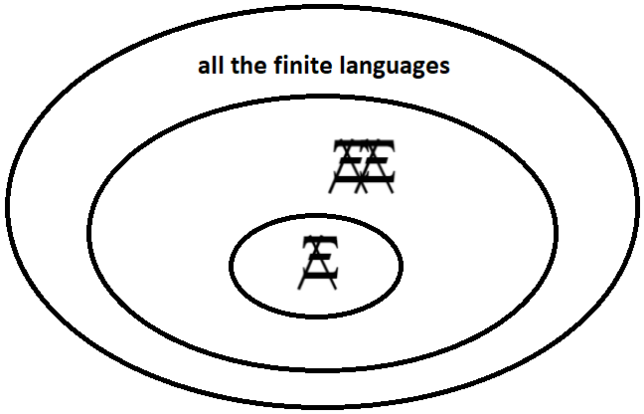

Figure 1. The inclusions of the two hypotheses.

In the figure, each "point" of the drawing corresponds to a language. More precisely, these "points" are:

- a pair (language D, word u) for the first hypothesis
- or a triple (language D, language F, language G) for the second one.

In the figure, we mean in both cases the variants of the language D only. At the same time, certainly, two the following sets of languages cannot be empty:

- the "internal" set corresponding to the potential fulfillment of the hypothesis ($\nexists$),
- and the "external" set corresponding to the potential non-fulfillment of the hypothesis ($\nexists$), but at the same time to the fulfillment of the hypothesis ($\nexists/\nexists$).

The last fact is obvious due to the existence of many trivial examples. For this, it is enough to consider the language $D = \{a, b\}$ over the alphabet $\Sigma = \{a, b\}$. (Simplifying, we can say, that "here the language coincides with the alphabet".) Certainly, there is no "bad" word u for this example, because there exist no word not belonging to Σ^*. But at the same time, although the above sets of languages cannot be empty, they can coincide with "external" sets of languages. Moreover, *the hypotheses consist in such a possible coincidence of sets of languages.*

5 Petal automata and some auxiliary variants of the formulation of the hypothesis

This section discusses (non-deterministic) finite automata of a special kind, [14, 15]. In the only publication found, the term "semi-flower automata" is used; it is certainly not very successful. Therefore we propose our title for them, i.e., "petal automata".

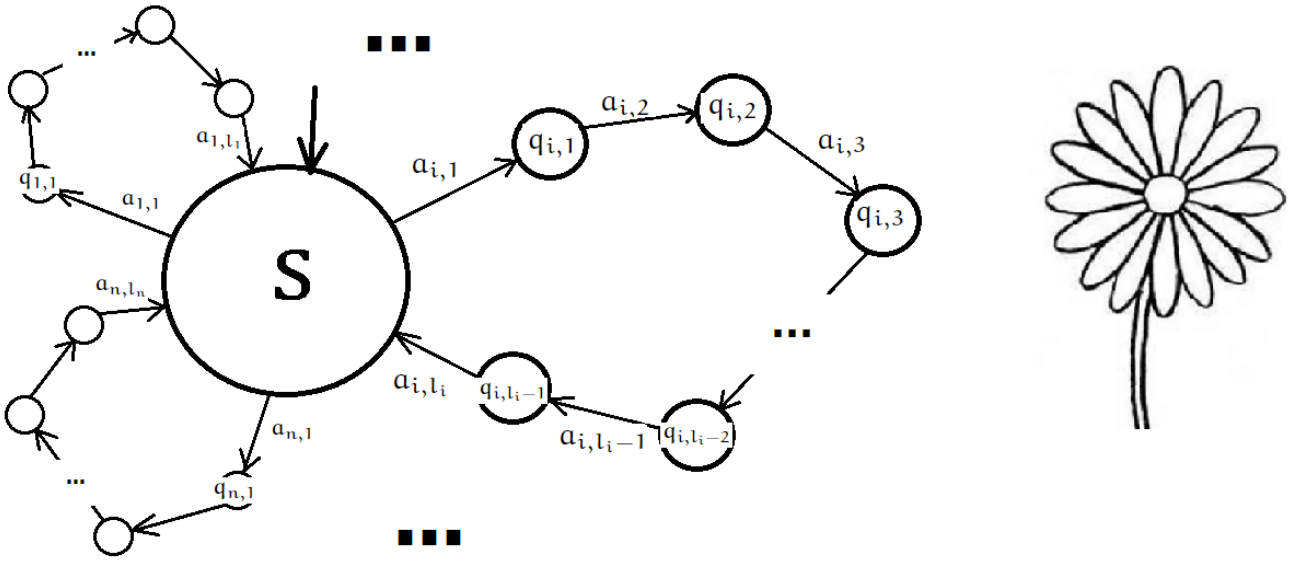

Figure 2. A detailed general diagram of the transition graph of the petal automaton and a schematic drawing of a flower (a daisy) for the analogy.

The motivation for their consideration follows from the question that arises: what can they have to do with the considered *set of problems*? After all, the main goal of almost all the problems briefly described above in this paper is to investigate the binary relation $\leqslant|\geqslant$; at the same time, we have noted in previous publications that there are two possible options for such a study:

- either to get some new necessary and / or sufficient conditions for fulfilling this relationship;
- or describe some new possibilities for its application in some other problems of the theory of formal languages.

But in both cases, the necessary "input data" are *two* finite languages (not one)...

However, the answer to this question is also almost obvious: yes, for all the problems described before, we really consider pairs of finite languages; at the same time, for each of the languages of such a pair, it is desirable (and usually possible) to replace its consideration with consideration of the "minimal" language (which is "minimal" in its corresponding equivalence class with respect to $\leqslant|\geqslant$). Note that possible approaches to defining the concept of "minimality" were discussed, for example, in [12]; it is not very principal.

Let us move on to the direct definition of the variant of petal automata that will be used in this paper. Let some finite language be given, it is

$$A = \{ u_1, u_2, \ldots, u_i, \ldots, u_n \}, \quad \text{where } u_1, \ldots, u_i, \ldots, u_n \in \Sigma^*, \tag{4}$$

and $A \not\ni \varepsilon$. Let also

$$u_i = a_{i,1} a_{i,2} \ldots a_{i,l_i} \ \ (l_i \geqslant 1) \ \text{ for each } \ i = 1,2,\ldots,n.$$

For this finite language A, let us consider nondeterministic finite automaton

$$\mathcal{K}(A) = (Q, \Sigma, \delta, \{s\}, Q), \tag{5}$$

where

$$Q = \{s\} \quad \cup \bigcup_{i \leq n,\ j < l_i} q_{i,j},$$

and the transition function δ, which is defined as follows for each $i = 1, \ldots, n$.

- If $|u_i| = 1$, then $\delta(s, a_{i,1}) \ni s$.
- Otherwise:
  - $\delta(s, a_{i,1}) \ni q_{i,1}$;
  - $\delta(q_{i,l_i-1}, a_{i,l_i}) = \{s\}$;
  - for each j such that $2 \leqslant j \leqslant l_i - 1$, we set $\delta(q_{i,j-1}, a_{i,j}) = \{q_{i,j}\}$.

  (We supposed $A \not\ni \varepsilon$, therefore $|u_i| \geqslant 1$.)

Such a definition describes the automaton shown in Fig. 2.; it is also obvious that the automaton (5) defines the language $pref(A^*)$.

As an example, let us consider the language that in the examples of previous publications was considered "as the second", i.e., in our usual notation, as the language B; exactly,

$$B = \{ aaaa, abb, abba, bbb \}.$$

We obtain the automaton shown on Fig. 3.; similarly to Fig. 2., all its states are outputs. As can be seen from this figure, to simplify the notation, we have designated the state of s as 0, and instead of each $q_{i,j}$ we write ij (this is possible here, because the maximum value of the state number is 4). We shall continue to consider this automaton in the remainder of this section.

As we have already noted, this section provides a formulation of a variant of the ($\exists\!\not\vee$) hypothesis using a non-deterministic automaton $\mathcal{K}(A)$. Note in advance that we shall not prove the equivalence of this formulation and the formulations given earlier. The possible scheme of this proof is as follows. When negating the formulation ($\exists\!\not\vee'''$) we consider the petal automaton $\mathcal{K}(D)$ for language D used in that formulation; the word u has the same

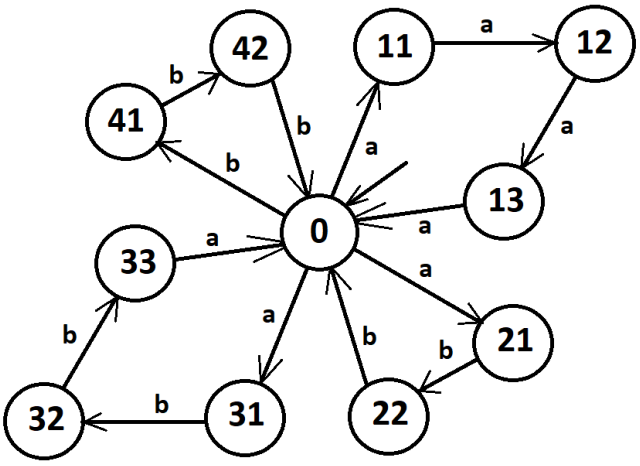

Figure 3. Petal automaton for the considered language B.

meaning. Generally speaking, this word is an input for several states of the automaton $\mathcal{K}(D)$, and at the same time, it is important that among these states there is no "main" state of the petal automaton (otherwise, we have a word from D^*). However, the union of the output languages of these states should contain the language of the given petal automaton as a subset, which is formulated in this version of the hypothesis.

Hypothesis $\nexists\!\!\!/^{IV}$. There exists *no* language $A \subseteq \Sigma^*$, such that the following condition holds.

For automaton $\mathcal{K}(A)$, its language $L = \mathcal{L}(\mathcal{K}(A))$, and its set of states Q, there exists a word

$$u \in L \setminus A^*,$$

such that for the subset of the set of states

$$Q' = \{\, q \in Q \,|\, \mathcal{L}^{in}_{\mathcal{K}(A)}(q) \ni u \,\}$$

the following is true:

$$\bigcup_{q \in Q'} \mathcal{L}^{out}_{\mathcal{K}(A)}(q) \supseteq A^*. \qquad \square$$

Let us especially note that such a check (i.e., checking the possible existence of the required word u) is possible *for a particular automaton* $\mathcal{K}(A)$, or, in other words, *for a particular language* A. In addition, it is obvious that the author has no examples of such languages (otherwise we would not need to formalize all this as a hypothesis). Thus, for all languages considered at the moment, the "bad" condition formulated in the hypothesis is not fulfilled.

Let us consider an example (a special case): for the petal automaton shown in Fig. 3., and the words *abbab*. The corresponding *input* states are 22, 32 and 41 (note that there is no 0 state among them, this suits us), therefore, to test *the special case* of the hypothesis, it must be shown that the union of the *output* languages of these states does not contain the language

of the given petal automaton as a subset. And this follows at least from the fact that all the words of the output languages of these states begin with b, while in the language of the original automaton, there are also some words beginning with a.

It can also be noted that all the various variants of the sets of vertices Q' (corresponding to various input words u) can be obtained in the process of determinization of the considered automaton $\mathcal{K}(A)$, using the usual algorithm of such determinization. (When applying the determinization algorithm, in which all the subsets of states of the given automaton $\mathcal{K}(A)$ as the so-called aggregate states are considered, we need to considering those aggregate states only, that are achievable.) All this makes it possible to formulate another equivalent version of the hypothesis we are considering.

In the remainder of this section, we shall use deterministic automata, more precisely, the standard determinization procedure. This procedure has been described many times in the literature since the 1950s, therefore, there is no need to give specific references; however, we shall use automata obtained after their procedure, and in this regard we need to briefly describe its specific version. Thus, we define the *standard determinization procedure* as follows.

We shall formulate the next version of the hypothesis for canonical automata (and not for such deterministic ones, which are obtained by determinization of petal automata, but without canonization of them); apparently, the difference is not fundamental, since the union of equivalent states of a deterministic automaton is not of interest for this paper.

We consistently form the canonical automaton

$$\hat{K} = (\,\hat{Q}, \Sigma, \hat{\delta}, \{S\}, \hat{F}\,),$$

which is equivalent to automaton (1); here, using designations of (1), $\hat{Q}, \hat{F} \subseteq \mathcal{P}(Q)$. For the induction basis of such forming, we set

$$\hat{\delta} = \varnothing, \quad \hat{Q} = \{S\};$$

these sets, generally speaking, will be changed. Next, we describe the induction step.

If we have already built values $\hat{\delta}$ for all states of $\hat{Q}$ and for all letters of Σ, then we build the set

$$\hat{F} = \left\{\, \hat{q} \in \hat{Q} \,\middle|\, (\exists f \in \hat{q})\,(f \in F) \,\right\}$$

and finish the process of building automaton $\hat{K}$.

Otherwise, we choose some state $\hat{q} \in \hat{Q}$, for which transitions $\hat{\delta}(\hat{q}, a)$ are not formed yet; we form them for each letter $a \in \Sigma$. Namely,

$$\hat{\delta}(\hat{q}, a) = \bigcup_{q \in \hat{q}} \delta(q, a). \tag{6}$$

We note two following auxiliary comments:

- in (6), we can use the sign $=$, but not $\ni$, since nothing else will be added to the set $\hat{\delta}(\hat{q}, a)$ after the only execution of (6);

- the possible further equivalent transformation of the resulting automaton, i.e. obtaining its canonical version, is not necessary for this paper.

Hypothesis $\exists\!\!\not\vee^{V}$. There exist *no language* $A \subseteq \Sigma^*$, such that the following holds. For automaton $\mathcal{K}(A)$, we obtain the equivalent deterministic automaton $\hat{\mathcal{K}}(A)$ using the standard determinization procedure described before (we also use all notation of it).

For $\hat{\mathcal{K}}(A)$, we choose some state $\hat{q} \in \hat{Q}$ such that $\hat{q} \not\ni s$, and for it, consider deterministic automaton

$$\hat{\mathcal{K}}_{\hat{q}}(A) = (\hat{Q}, \Sigma, \hat{\delta}, \{\hat{q}\}, \hat{F}).$$

We note two following auxiliary comments:

- recall that the state s used for choosing $\hat{q}$ is the "main" state of nondeterministic automaton $\mathcal{K}(A)$;
- under such constraints, at least one such state $\hat{q}$ gives the fulfillment of the condition under consideration (the non-fulfillment of which is postulated in the present version of the hypothesis).

Then

$$\mathcal{L}(\hat{\mathcal{K}}_{\hat{q}}(A)) \supseteq \mathcal{L}(\hat{\mathcal{K}}(A)) = \mathcal{L}(\mathcal{K}(A)) = A^*. \qquad \square$$

Let us continue to consider the example of the automaton shown in Fig. 3.. First, let us consider the process of determinization of this automaton, see Table 1.. We should note that in it, we denoted aggregate states by applying such conventions:

- bold font indicates those ones, which do not contain the state 0 as their element;
- of the last ones, those are highlighted with a gray background, in which there are transitions on both letters of the alphabet (from the absence of at least one such transition it immediately follows, that the automaton with such a starting state is obviously not suitable, i.e., it does not require further verification); we have 2 such states.

We also recall that according to our agreements, all states of the considered automata are output; therefore, we do not note this fact in the tables.

By reinterpreting the aggregate states in the usual way, we obtain an equivalent deterministic automaton given in Table 2.. As we have already noted, it is canonical. Matching the name of the automaton with the name of its state is made for readability and should not cause problems.

We already know that to test a special case of the hypothesis, we need to consider 2 automata, obtained from Table 2. by replacing the starting state with one of the two that we

Table 1. The process of determinization of the initial petal automaton.

		a	b
→	0	11 21 31	41
	11 21 31	12	22 32
	41	—	42
	12	13	—
	22 32	—	0 33
	42	—	0
	13	0	—
	0 33	0 11 21 31	41
	0 11 21 31	11 12 21 31	22 32 41
	11 12 21 31	12 13	22 32
	22 32 41	—	0 33 42
	12 13	0 13	—
	0 33 42	0 11 21 31	0 41
	0 13	0 11 21 31	41
	0 41	11 21 31	41 42
	41 42	—	0 42
	0 42	11 21 31	0 41

Table 2. The resulting equivalent deterministic automaton.

	K	a	b
→	A	B	C
	B	D	E
	C	—	F
	D	G	—
	E	—	H
	F	—	A
	G	A	—
	H	J	C
	J	K	L
	K	M	E
	L	—	N
	M	P	—
	N	J	Q
	P	J	C
	Q	B	R
	R	—	S
	S	B	Q

Table 3. The first deterministic automaton to be checked.

	K1	a	b	
	A	B	C	A⩽B ⇒
→	B	D	E	B⩽D & C⩽E ⇒
	C	—	F	... E⩽Ø ...
	D	G	—	
	E	—	H	
	F	—	A	
	G	A	—	
	H	J	C	
	J	K	L	
	K	M	E	
	L	—	N	
	M	P	—	
	N	J	Q	
	P	J	C	
	Q	B	R	
	R	—	S	
	S	B	Q	

Table 4. The second deterministic automaton to be checked.

	K2	a	b	
	A	B	C	A⩽K ⇒
	B	D	E	B⩽M & C⩽E ⇒
	C	—	F	... E⩽Ø ...
	D	G	—	
	E	—	H	
	F	—	A	
	G	A	—	
	H	J	C	
	J	K	L	
→	K	M	E	
	L	—	N	
	M	P	—	
	N	J	Q	
	P	J	C	
	Q	B	R	
	R	—	S	
	S	B	Q	

previously marked with a gray background; in the new notation, these are the states B and K.

For the input state B, we obtain the automaton K_1, see Table 3..

To the right of the table, there is a brief explanation that this automaton is not a counterexample to the hypothesis $\exists\!\!\not| \, ^{V}$; in this text, the $\leqslant$ sign means that the output language of the "smaller" state *must be* a subset of the language of the "greater" state. Therefore the condition $\mathcal{L}^{out}_{K_1}(A) \subseteq \mathcal{L}^{out}_{K_1}(B)$ must be met, and from here, considering both possible transitions from these states, we get the need to fulfill the condition

$$\mathcal{L}^{out}_{K_1}(B) \subseteq \mathcal{L}^{out}_{K_1}(D) \ \ \& \ \ \mathcal{L}^{out}_{K_1}(C) \subseteq \mathcal{L}^{out}_{K_1}(E);$$

the first of the conditions in this conjunction is impossible due to the required transitions by the letter b and the deliberate non-fulfillment of the condition $\mathcal{L}^{out}_{K_1}(E) = \varnothing$.

The second automaton necessary to test the hypothesis (K_2, its input is the state K) is given in Table 4.; the verification is carried out similarly.

For the formulation of the next version of the hypothesis, we use constructions obtained on the basis of the petal finite automata.

6 The auxiliary variants using infinite trees

Thus, like previous formulations, we consider the language A used for obtaining automaton $\mathcal{K}(A)$. This section uses another equivalent reformulation of the same hypothesis, exactly, the one in terms of infinite iterative trees, see [8,9] etc. For its "initialization", we shall apply a variant of the hypothesis ($\exists\!\!\not| \, '''$).

First, we have to define strictly the simple version of infinite iterating.

Definition 1. For the nonempty finite language $A \subseteq \Sigma^*$, such that $A \not\ni \varepsilon$, we define the infinite iterating tree (we shall denote it by $IIT(A)$) as follows.

- The vertices of the tree $IIT(A)$ are a subset of all the vertices of the infinite tree corresponding to the language Σ^*.
- This subset is the vertices of the type $pref(A^*)$.
- The vertices corresponding to the words of the set A^* are called marked, the other ones are called unmarked.

The (infinite) subtrees of this tree are defined in the usual way. For their denoting, we use their roots. □

For some examples, see also [8,9].

Now we can give another equivalent formulation of the hypothesis ($\exists\!\!\not| \, $).

Hypothesis $\exists\!\!\not| \, ^{VI}$. There exist *no language* $A \subseteq \Sigma^*$, such that the tree $IIT(A)$ built for it it has the following property. Among the unmarked vertices, there exist some v, such that the v-subtree contains the given tree (i.e., the ε-subtree) as the subtree. □

7 Conclusion. The possible reduction of P = NP equality to considered hypothesis

A more exact title for this section could be as follows: "On the plan of proving the possibility of reducing the equality of $P = NP$ to a hypothesis of the theory of formal languages" (meaning ($\nexists$)).

Thus, *let everywhere else* the condition ($\nexists$) is met. Knowing that it is fulfilled, we somehow want to determine for some finite languages $A, B \subseteq \Sigma^*$, whether the equality of $A \leqslant\!|\!\geqslant B$ is fulfilled. Let us describe *two possible ways to get an answer* to this question.

The first way. Firstly, we construct an optimal inverse morphism for each of these two languages as an auxiliary problem; according to the hypothesis ($\nexists$) (to its "main" variant), we should get matching languages, i.e. the same final language $D \subseteq \Sigma^*$, such that

$$A, B \in mp^+(D).$$

Note that the notation D has in this case the same meaning that was used in [5].

For this goal, we firstly consider the language A (we "temporarily forget" the language B). For A, we make the following steps.

- First, we consider its *potential roots* ([11–13]):

$$A' = \overline{\sqrt[*]{A}} = \left\{ u \in \Sigma^* \,\middle|\, (\exists n \in \mathbb{N})\,(u^n \in A) \right\};$$

  we specifically note that the equality $n = 1$ is allowed in this case.

- Secondly, we recall that the problem we are considering can in principle be solved by iterating over the set of all subsets of the set of potential roots. (Evidently, such an algorithm is exponential.)

- Thirdly, we consider the special subset of the set of potential roots. Exactly, we consider those of them that *are not represented* in the form of the following nontrivial concatenation of its other potential roots:

$$A'' = \left\{ u \in A' \,\middle|\, (\nexists u_1, u_2, \ldots, u_k \in A', k \geqslant 2)\,(u = u_1 u_2 \ldots u_k) \right\}.$$

- Fourthly, we note that the language A'' can be built on the basis of the language A using a polynomial algorithm; this is obvious.

- Fifthly, as it is easy to see (for example, based on the material [11], or simply based on the definition of the binary relation $\leqslant\!|\!\geqslant$) that $A'' \leqslant\!|\!\geqslant A'$. In this regard, we shall further consider those potential roots only, that are included in A''.

In the resulting language, let us choose some word $u \in A''$ and consider the problem of determining whether it is true that

$$A'' \leqslant\!|\!\geqslant A'' \setminus \{u\}. \tag{7}$$

Due to the fulfillment of the hypothesis ($\nexists$), if at the same time the condition (7) also holds, we obtain that $u \in (A'')^*$; this gives a trivial (and polynomial) algorithm for checking the needed condition (7).

Therefore, *sequentially* deleting all the words u such that the condition (7) is not met (i.e., in other words, deleting all the words $u \in A''$, such that

$$A'' \mid\!> A'' \setminus \{u\}$$

holds), we obtain the required language D which is *the minimal one of the equivalence class defined by the considered language* A. Since one such deletion is polynomial, the entire algorithm for constructing the required language D is also polynomial.

Remark once again, that the "minimality of the language" can be defined in various possible obvious ways. Some of them have been described in our previous papers. In this paper, we shall not consider this issue in detail.

Remark also that we are considered not a general algorithm for constructing the required language D with the necessary strict proofs, but the *scheme* of such an algorithm only. We propose to give a full description in one of the next publications.

The second way. Let us start the description of this path with information about the "inverse problem". Before, automaton $NSPRI(A,B)$ for the given languages $A, B \subseteq \Sigma^*$ was described; it is a non-deterministic finite automaton, the result of which gives an answer to the question whether the relation $A \leqslant\!\mid\!\geqslant B$ is fulfilled: it is fulfilled if and only if the language of the automaton $NSPRI(A,B)$ is the "complete" (the "universal") language Σ^*. In [9], we showed the possibility of constructing such an automaton using two given finite languages $A, B \subseteq \Sigma^*$ in polynomial time.

However, in the general case, the answer to the last question (i.e., whether the language of some nondeterministic finite automaton coincides with the language Σ^*) is an NP-complete problem (even when all states are output, as in the case we are considering) and this gives an "indirect argument" about the NP-completeness of the "direct problem", which is more interesting to us.

We shall formulate it (i.e., the "direct problem") as follows. First, it is desirable for us to formally describe the restrictions imposed on a (nondeterministic) automaton $NSPRI$; such restrictions do exist (see [8,9] etc.), for example, the following one. According to the properties of this automaton, the presence of any word $u \in \Sigma^*$ as a designation of some state of this automaton implies the presence any suffix of u as the other states of this automaton, i.e., any word belonging to the set $suff(u)$; in the formulas,

$$u \in \mathcal{Q} \quad \Longrightarrow \quad \big(\forall v \in pref(u)\big)\ \big(v \in \mathcal{Q}\big); \tag{8}$$

"inside" this section, we shall call (8) by the restriction (Ω).

Let us note that there may be other restrictions on the $NSPRI$ automaton (unrelated to (Ω)), but also note that we do not need to strictly prove the fact that any particular set of restrictions gives a description of any possible $NSPRI$ automaton: even if these restrictions

satisfy a larger set of automata, we shall be able to complete the consideration of such a "direct problem". However, of course, *the presence of these constraints should not make it possible to answer in polynomial time the question whether an automaton satisfying such constraints defines the "universal" language* Σ^*; and (Ω) (more precisely, a set of constraints containing (Ω) only) is such a restriction.

After describing the constraints, it is necessary to describe the procedure for constructing a pair (A, B), which should be obtained in polynomial time on the basis of the given automaton (which is a potential automaton $NSPRI$). To describe such a procedure, we propose to give in one of the following publications an algorithm for a possible *subproblems* of this problem, i.e., the descriptions of a polynomial-time procedure for adding a transition. Let us describe such a subtask in a little more detail.

- Let for some languages A and B, *we already have* the corresponding automaton $NSPRI(A, B)$.
- Let, in addition, some value of the transition function

$$\delta(q, a) \ni q', \tag{9}$$

  be given; in the given automaton $NSPRI(A, B)$, this value should be missing.
- Then we need to specify an algorithm that modifies the given automaton $NSPRI(A, B)$ in polynomial time, i.e., for example, adds some words to the languages A and / or B, resulting in a modification of the original $NSPRI(A, B)$ automaton, and it is an automaton, which has all the existing transitions, and also a transition (9); at the same time, it is possible to change the languages A and / or B, but such a change must also be performed in polynomial time.

Thus, if we have a polynomial-time algorithm for constructing a pair of languages (A, B) for a given automaton $NSPRI$, then we determine for this pair of languages, whether the condition $A \leqslant| \geqslant B$ is met:

- in polynomial time (certainly, if the hypothesis ($\exists\not/$) is fulfilled);
- according to the above "first way".

Therefore, in the case of the ($\exists\not/$) hypothesis, we answer the question whether the language of the automaton $NSPRI$ is Σ^* also in polynomial time.

References

[1] B. Melnikov, On a classification of sequential context-free languages and grammars, Vestnik of Moscow University, series "Computational mathematics and cybernetics", 3 (1993) 64–69 (in Russian).
[2] O. Dubasova, B. Melnikov, On an extension of the class of context-free languages, Programming and Computer Software, 21(6) (1995) 299–306.

[3] G. Sénizergues, L(A)=L(B)? Decidability results from complete formal systems, Theoretical Computer Science, 251(1–2) (2001) 1–166.
[4] B. Melnikov, A. Melnikova, A polynomial algorithm for construction an automaton for checking the equality of infinite iterations of two finite languages, Lecture Notes in Networks and Systems, 502 (2022) 521–530.
[5] B. Melnikov, The equality condition for infinite catenations of two sets of finite words, International Journal of Foundation of Computer Science, 4(3) (1993) 267–274.
[6] B. Melnikov, L. Meng, Eight variants of finite automata for checking the fulfillment of the coverage relation of iterations of two finite languages. Part I, International Journal of Open Information Technologies, 11(11) (2023) 1–9 (in Russian).
[7] B. Melnikov, L. Meng, Eight variants of finite automata for checking the fulfillment of the coverage relation of iterations of two finite languages. Part II, International Journal of Open Information Technologies, 12(2) (2024) 1–11 (in Russian).
[8] B. Melnikov, Variants of finite automata corresponding to infinite iterative morphism trees. Part I, International Journal of Open Information Technologies, 9(7) (2021) 5–13 (in Russian).
[9] B. Melnikov, Variants of finite automata corresponding to infinite iterative morphism trees. Part II, International Journal of Open Information Technologies, 9(10) (2021) 1–8 (in Russian).
[10] B. Melnikov, Once more on the edge-minimization of nondeterministic finite automata and the connected problems, Fundamenta Informaticae, 104(3) (2010) 267–283.
[11] B. Melnikov, Semi-lattices of the subsets of potential roots in the problems of the formal languages theory, Part I. Extracting the root from the language, International Journal of Open Information Technologies, 10(4) (2022) 1–9 (in Russian).
[12] B. Melnikov, Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part II. Constructing an inverse morphism, International Journal of Open Information Technologies, 10(5) (2022) 1–8 (in Russian).
[13] B. Melnikov, Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part III. The condition for the existence of a lattice, International Journal of Open Information Technologies, 10(7) (2022) 1–9 (in Russian).
[14] B. Melnikov, Petal finite automata: basic definitions, examples and their relation to complete automata. Part I, International Journal of Open Information Technologies, 10(9) (2022) 1–11 (in Russian).
[15] B. Melnikov, Petal finite automata: basic definitions, examples and their relation to complete automata. Part II, International Journal of Open Information Technologies, 10(10) (2022) 1–10 (in Russian).

Citation: Boris Melnikov, On an interesting hypothesis of the theory of formal languages, J. Disc. Math. Appl. 9(2) (2024) 81–102.
https://doi.org/10.22061/jdma.2024.10789.1069

COPYRIGHTS
©2024 The author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, as long as the original authors and source are cited. No permission is required from the authors or the publishers.