Mathematical Problems of Computer Science 54, 69–79, 2020.

UDC 519.713, 004.43

An Approximate Method for Calculating the Distance Between Regular Languages for Multitape Finite Automata

Tigran A. Grigoryan

IT Educational and Research Center
Yerevan State University, Armenia
e-mail: tigran.grigoryan1995@gmail.com

Abstract

Sets of word tuples, accepted by multitape finite automata and a metric space for languages accepted by these automata, are considered. These languages are represented using the same notation as the known notation of regular expressions for languages accepted by one-tape automata. The only difference is the interpretation of the "concatenation" operation in the notation.

An algorithm is proposed for calculating the introduced distance between regular languages accepted by multitape finite automata.

Keywords: Multitape finite automata, Regular languages, Metric space.

1. Introduction

In 1959 M. O. Rabin and D. S. Scott introduced deterministic multitape finite automata and the problem of equivalence for these automata [1].

The equivalence problem for two-tape automata was proved to be solvable in 1973 by M.Bird [2]. In 1991 T. Harju and J. Karhumki proved the solvability of the equivalence problem for deterministic multitape automata without any restriction on the number of tapes [3] via a purely algebraic technique.

Many attempts were made to consider languages accepted by multitape automata. Some notable from our point of view attempts are briefly discussed below.

In [4], B. G. Mirkin has considered a special coding for the sets of words tuples accepted by multitape automata. The proofs and discussions are only for the case of $n = 2$ and there are no explanations/proofs on how this will extend to the case of $n > 2$.

Another paper [5] by P. H. Starke is dedicated to the following result: "An n-ary relation R over $W(X)$ is representable by a finite deterministic n-tape automaton iff there exists an admissible regular expression T such that $R = Val_n(T)$". In the same paper the

author mentions that "Unfortunately there are non-admissible regular expressions T such that $Val_n(T)$ is representable by a deterministic automaton".

A special coding suggested in [6] is used in this paper. This coding was also used in [7] to define regular expressions and regular events for multitape automata. The notation of regular expressions for one-tape automata [8] was used as a notation for languages accepted by multitape finite automata via interpreting the "concatenation" operation differently.

Introduction of the special binary coding for elements of a free partially commutative semigroup mentioned above leads to the consideration of multidimensional tape cells instead of the semigroup elements. This, in turn, gives an opportunity to compare them as integer vectors when analyzing behaviors of two automata on a given semigroup.

A metric is introduced based on these integer vectors. The metric intends to immerse the knowledge on coding in the notion of distance between regular expressions.

The introduced metric has some boundary cases where its value is not adequate. To adjust such cases, a new pseudo metric is introduced as an adjuster and is combined with the former one resulting in a more suitable metric.

A polynomial algorithm is proposed for calculating the distance between regular languages.

2. Preliminaries

Recall some definitions from [6, 9].

If X is an alphabet, then the set of all words in the alphabet X, including the empty word, will be denoted X^*, and the set of all n-element tuples of words will be denoted X^{n^*}.

Let G be a free semigroup, generated by the set of generators $Y = \{y_1, y_2, \ldots, y_n\}$. G is called a free partially commutative semigroup, if it is defined by a finite set of relations R of type $y_i y_j = y_j y_i$ [10]. We consider semigroups with identity elements (monoids) and use the notation $G = \langle Y \mid R \rangle$.

Let $K : Y^* \to \{0,1\}^{n^*}$, $n = |Y|$ be a homomorphism over the set Y^*, which maps words from Y^* to n-element vectors in binary alphabet $\{0,1\}$. The homomorphism K over the set of symbols of the set Y^* is defined by the equation:

$$K(y_i) = (a_{1i}, \ldots, a_{ni}), \text{ where } a_{ij} = \begin{cases} 1, & \text{if } i = j, \\ \text{e}, & \text{if } y_i y_j = y_j y_i, \\ 0, & \text{if } y_i y_j \neq y_j y_i. \end{cases}$$

At the same time $K(e) = (e, \ldots, e)$.

$K(y_i y_j)$, $i \neq j$ is defined in the following way:

$$K(y_i y_j) = (a_{1j} a_{1i}, \ldots, a_{nj} a_{ni}).$$

K maps the concatenation of semigroup elements $a = y_{i_1} \ldots y_{i_k}$ and $b = y_{j_1} \ldots y_{j_l}$ in the following way:

$$K(ab) = K((y_{i_1} \ldots y_{i_k})(y_{j_1} \ldots y_{j_l})) = (a_{1j_l} \ldots a_{1j_1} a_{1i_k} \ldots a_{1i_1}, \ldots, a_{nj_l} \ldots a_{nj_1} a_{ni_k} \ldots a_{ni_1}).$$

Lemma 1: [9] *Let y_i, y_j be generators of G, $y_i \neq y_j$, $g_1 = y_i y_j$, $g_2 = y_j y_i$ be elements of G, obtained after applying the operation of the semigroup G to generators y_i and y_j. Then*

$$g_1 = g_2 \Leftrightarrow K(y_i y_j) = K(y_j y_i).$$

This statement allows to consider the homomorphism K as a mapping not only over the Y^*, but also over the free partially commutative semigroup G.

An equivalence relation ρ over Y^* is specified as follows. If w_1 and w_2 are words from Y^*, $w_1,\ w_2 \in Y^*$, then $w_1 \rho w_2$ if and only if w_1 and w_2 are representations of the same element in G.

The relation ρ partitions Y^* into disjoint classes. These classes are called classes of commutation.

Lemma 2: [6, 11] *Any free partially commutative semigroup of n generators is isomorphic to some sub-semigroup of Cartesian product of n free semigroups with two generators.*

According to Lemma 2, instead of elements of the semigroup G we can consider their binary codings.

Any n-tuple of binary words can be considered as a tuple of integers (it will be denoted $Num(c_g)$, where c_g is the binary coded n-tuple of the semigroup element g). For that, we just treat each non-empty binary word (components of the tuple) as the binary representation of the integer (e.g., 010111 = 23) and use 0 for the empty word e [11].

Lemma 3: [11] *Any free partially commutative semigroup of n generators is isomorphic to some sub-semigroup of the n-dimensional space, where the semigroup operation is the concatenation of integer tuples.*

The multiplication of n-element tuples is defined as componentwise multiplication of corresponding binary words - components of tuples. The multiplication of binary words $B_1 = \beta_{11} \dots \beta_{1n_1}$ and $B_2 = \beta_{21} \dots \beta_{2n_2}$ is defined as $B_1 B_2 = \beta_{2n_2} \dots \beta_{21} \beta_{11} \dots \beta_{1n_1}$, i.e., the concatenation of new letters to the source word B_1 is performed starting from the leftmost letter and is added to the left of the word B_2.

In [7], the coding with n-tuples of binary words is considered to define regular expressions and regular events over a free partially commutative semigroup. These definitions allow to apply the already known notation of regular expressions for the case of multitape automata, however, the concatenation operation is interpreted differently.

Let R be a regular expression over a free partially commutative semigroup. By $E(R)$ we denote the regular event denoted by R.

For simplicity, we will use "word p belongs to regular event E" to indicate that "the equivalence class $[p]$ belongs to E". Also, we will use the elements of the partially commutative alphabet rather than their corresponding binary coded tuples in the notation of the regular expressions. For instance, for $Y = \{y_1, y_2\}$, where $y_1 y_2 = y_2 y_1$ by writing $y_1 y_2^* + y_2$ we will mean $(1, e)(e, 1)^* + (e, 1)$.

Next, we recall the definition of multitape finite automata (MFA).

Let Q be a finite set of states, X be an input alphabet, $\delta : Q \times X \to 2^Q$ be the transition function, $q_0 \in Q$ be the initial state and $F \subseteq Q$ be the set of final states. Assume that X can be divided into disjoint, ordered subsets $X = X_1 \cup \dots \cup X_n$ such that $X_i \cap_{i \neq j} X_j = \emptyset$ and $\forall x, x' \, (x \in X_i, x' \in X_j (i \neq j), xx' = x'x)$. Each subset X_i corresponds to i-th tape.

Definition 1: [1] *An n-tape automaton (MFA) is called a tuple $A = (Q, T, X, \delta, q_0, F)$, where $T : Q \to \{1, \dots, n\}$ is a function associating each state from Q with a certain tape and $Q = \cup_{i=1}^{n} Q_i$, such that $Q_i = \{q | q \in Q, T(q) = i\}$ $\forall i = 1, \dots, n$.*

Automaton is called deterministic (DMFA), if $\forall q \in Q,\ \forall x \in X\ |\delta(q,x)| \leq 1$. Otherwise, it is called nondeterministic (NMFA).

An NMFA with ε transitions (NMFA-ε) is a tuple $A = (Q, T, X, \delta, q_0, F)$, where $\delta : Q \times (X \cup \{\varepsilon\}) \to 2^Q$.

A path in an automaton graph [8] from an initial state to final state is called an accepted path. A string formed by concatenating labels of all transitions in an accepted path, is called an accepted extended word of multitape automaton [11].

An n-tuple of words $(w_1, w_2, \ldots, w_n) \in \prod_{i=1}^{n} X_i^*$ is accepted by an MFA A if and only if there exists an extended word $w \in X^*$ accepted by A, such that w_i $(i = 1, \ldots, n)$ is a word obtained from w by removing all symbols of all subsets X_j, $j \in \{1, \ldots, n\}, j \neq i$.

The set of all n-tuples accepted by A is called the language of the automaton A and is denoted by $L(A)$. Two automata are called equivalent, if they accept the same language, i.e., $A_1 \equiv A_2$ iff $L(A_1) = L(A_2)$.

Further, in this paper, we will only consider alphabets (and/or set of generators) X satisfying the following condition: X can be divided into disjoint, ordered subsets $X = X_1 \cup \ldots \cup X_n$ such that $X_i \cap_{i \neq j} X_j = \emptyset$ and $\forall x, x'\, (x \in X_i, x' \in X_j (i \neq j), xx' = x'x)$.

3. $\tilde{L}_k$ Distance

The result of Lemma 3 brings up a natural question of whether the Euclidean metric can be applied to regular languages over free partially commutative semigroups. Obviously, it can, however, the adequacy of such metric is questionable. To understand the issue of such a metric, we may look at Fig. 2 and Fig. 3 of [11]. It is clear that each time adding the same letter to the word moves it from $2^{n-1} - 1$ diagonal to $2^n - 1$ diagonal. For example, consider the sequence $e, y_1, y_1^2, y_1^3, \ldots$. The coordinates for its elements are $(0,0), (1,0), (3,0), (7,0), \ldots$ correspondingly, hence, the Euclidean distance between y^n and e is $2^n - 1$. In this section, we will introduce a new metric, which will transform this exponential growth of difference to a more natural and closer to linear one.

3.1. L Distance

Let G be a free partially commutative semigroup with n generators. Recall mappings Num and K discussed in Section 2. Let Num' be the composition of K and Num, i.e., $Num' := K \circ Num : G \to \mathbb{R}^n$. $Num'_i : G \to \mathbb{R}, (i = 1, \ldots, n)$ denotes the projection of the mapping $Num'(G)$ on the i-th axis of $\mathbb{R}^n$.

Definition 2: *Let g_1, g_2 be words in a free partially commutative semigroup G with n generators. The logarithmic distance between g_1 and g_2 is denoted by $L(g_1, g_2)$ and is equal to*

$$L(g_1, g_2) = \sqrt{\lg^2 \frac{Num'_1(g_1) + 1}{Num'_1(g_2) + 1} + \cdots + \lg^2 \frac{Num'_n(g_1) + 1}{Num'_n(g_2) + 1}}.$$

L is well-defined, as $\forall i = 1, \ldots, n, \forall g \in G\ Num'_i(g) + 1$ is a positive number.

Before continuing the investigation of this metric, let us see why it has this form. Let us consider the same sequence $e, y_1, y_1^2, y_1^3, \ldots$ with the corresponding coordinates $(0,0)$, $(1,0)$, $(3,0)$, $(7,0)$, ... The function $\lg^2 \frac{Num_1(g_1)+1}{Num_1(g_2)+1}$, indeed, maps their exponential difference to a linear one. Thus, $L(y_1, e) = L(y_1^2, y_1) = L(y_1^3, y_1^2) = \ldots = 1$, and generally, $L(y_1^m, y_1^k) =$

$|m-k|$. Additionally, adding one in numerator and denominator prevents the values in both of them from having the illegal value 0.

Lemma 4 below states that the defined distance is a metric on the free partially commutative semigroup G.

Lemma 4: *Let g_1, g_2 and g_3 be elements of a free partially commutative semigroup G with n generators, then*

1. $L(g_1, g_2) = 0 \Leftrightarrow g_1 = g_2$,
2. $L(g_1, g_2) = L(g_2, g_1)$,
3. $L(g_1, g_3) \leq L(g_1, g_2) + L(g_2, g_3)$.

Proof. Let us prove each property separately.

1. The equality $L(g_1, g_2) = 0$ takes place if and only if $\lg^2 \frac{Num'_i(g_1)+1}{Num'_i(g_2)+1} = 0,\ \forall i = 1, \ldots, n$. The later one is true if and only if $Num'_i(g_1) = Num'_i(g_2),\ \forall i = 1, \ldots, n$. After combining all the i-s, we will find that $L(g_1, g_2) = 0 \Leftrightarrow Num'(g_1) = Num'(g_2)$. From the fact that Num' mapping is an isomorphism follows that $Num'(g_1) = Num'(g_2) \Leftrightarrow g_1 = g_2$.

2. The proof of the second property follows from

$$\lg \frac{Num'_i(g_1)+1}{Num'_i(g_2)+1} = -\lg \frac{Num'_i(g_2)+1}{Num'_i(g_1)+1},\ \forall i = 1, \ldots, n,$$

hence,

$$\lg^2 \frac{Num'_i(g_1)+1}{Num'_i(g_2)+1} = \lg^2 \frac{Num'_i(g_2)+1}{Num'_i(g_1)+1},\ \forall i = 1, \ldots, n.$$

After combining all i-s, we get

$$\sqrt{\sum_{i=1}^{n} \lg^2 \frac{Num'_i(g_1)+1}{Num'_i(g_2)+1}} = \sqrt{\sum_{i=1}^{n} \lg^2 \frac{Num'_i(g_2)+1}{Num'_i(g_1)+1}}.$$

3. Let us denote $g_{1i} := Num'_i(g_1)$ and $g_{2i} := Num'_i(g_2)$. We need to prove that

$$\sqrt{\sum_{i=1}^{n} \lg^2 \frac{g_{1i}+1}{g_{2i}+1}} + \sqrt{\sum_{i=1}^{n} \lg^2 \frac{g_{2i}+1}{g_{3i}+1}} \geq \sqrt{\sum_{i=1}^{n} \lg^2 \frac{g_{1i}+1}{g_{3i}+1}}. \quad (1)$$

Let us square both sides of the inequality (1). We get

$$\sum_{i=1}^{n} \lg^2 \frac{g_{1i}+1}{g_{2i}+1} + \sum_{i=1}^{n} \lg^2 \frac{g_{2i}+1}{g_{3i}+1} + 2\sqrt{\sum_{i=1}^{n} \lg^2 \frac{g_{1i}+1}{g_{2i}+1}} \sqrt{\sum_{i=1}^{n} \lg^2 \frac{g_{2i}+1}{g_{3i}+1}} \geq \sum_{i=1}^{n} \lg^2 \frac{g_{1i}+1}{g_{3i}+1}. \quad (2)$$

Also, it is trivial that

$$\sum_{i=1}^{n} \lg^2 \frac{g_{1i}+1}{g_{3i}+1} = \sum_{i=1}^{n} \left(\lg \frac{g_{1i}+1}{g_{2i}+1} + \lg \frac{g_{2i}+1}{g_{3i}+1} \right)^2. \quad (3)$$

After putting the identity (3) into the inequality (2) and making some simple operations, we find that in order to prove (1) it is sufficient to show that

$$\sqrt{\sum_{i=1}^{n} \lg^2 \frac{g_{1i}+1}{g_{2i}+1}} \sqrt{\sum_{i=1}^{n} \lg^2 \frac{g_{2i}+1}{g_{3i}+1}} \geq \sum_{i=1}^{n} \lg \frac{g_{1i}+1}{g_{2i}+1} \lg \frac{g_{2i}+1}{g_{3i}+1}. \quad (4)$$

For simplicity, let us denote $u_i := \lg \frac{g_{1i}+1}{g_{2i}+1}$ and $v_i := \lg \frac{g_{2i}+1}{g_{3i}+1}$. The inequality (4) takes the following form:

$$\sqrt{u_1^2 + u_2^2 + \ldots + u_n^2} \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2} \geq u_1 v_1 + u_2 v_2 + \ldots + u_n v_n. \quad (5)$$

(5) is the well-known Cauchy-Schwartz inequality, hence, the third property is also proved.

■

Next, we define logarithmic distance on the set of all regular events over a free partially commutative semigroup by inducing Hausdorff metric from L [12]. This distance will be called L_H distance.

Definition 3: *Let E_1 and E_2 be regular events over a free partially commutative semigroup. L_H distance between E_1 and E_2 is the quantity*

$$L_H(E_1, E_2) = \max \left\{ \sup_{r_1 \in E_1} \inf_{r_2 \in E_2} L(r_1, r_2), \sup_{r_2 \in E_2} \inf_{r_1 \in E_1} L(r_1, r_2) \right\}.$$

In Definition 3 we assume that L_H can have the value ∞.

For L_H metric to be well-defined, we assume that $L_H(\emptyset, \emptyset) = 0$ and $L_H(\emptyset, P) = L_H(P, \emptyset) = \infty, \forall P \neq \emptyset$.

The investigation of this metric shows that in most cases the evaluated distance of the words over a free partially commutative semigroup is adequate, however, after applying the homomorphism $K \circ Num$ on some words of the same length they appear close to each other on the same essential diagonal on $\mathbb{R}^n$ [6]. For instance, consider the words $a^k b$ and $b^k a$, where $k \in N$. The logarithmic distance of these words is $\sqrt{2} \lg \frac{2^k+1}{2^k}$, which goes to 0 as $k \to \infty$.

To adjust the introduced metric, we consider further a notion of a distance adjuster and combine it with the L_H distance.

3.2. Adjusted L Distance

Let $Y = y_1, y_2, \ldots, y_n$ be the set of generators of the free partially commutative semigroup G. $K : G \to \{0,1\}^{n^*}$ is the mapping from G into the semigroup of n-element tuples of binary words. For a word $g \in G$, the binary word of zeros and ones derived from g replacing all occurrences of the generator y_i by one and all occurrences of other generators by zero is called the mask for occurrences of generator $y_i \in Y$ for the word g [9].

Let $g \in G$, by $d_1^n(g)$ we will denote the n-element vector, for which i-th element is the number of ones in the mask for occurrences of generator y_i in the word g ($\forall i = 1, \ldots, n$). Now, we define a distance adjuster between words in G.

Let $g_1, g_2 \in G$. We denote the Euclidean distance between vectors $d_1^n(g_1)$ and $d_1^n(g_2)$ by $D(g_1, g_2)$.

It is easy to check that D is a metric on $\{d_1^n(g) \mid g \in G\}$. However, it is a pseudometric on G. Indeed, consider the free partially commutative semigroup $G = \langle y_1, y_2 \rangle$. Let us consider the words y_1y_2 and y_2y_1. From the definition of the function d_1^n we have that $d_1^n(y_1y_2) = d_1^n(y_2y_1) = (1, 1)$. Hence, the distance $D(y_1y_2, y_2y_1)$ equals to 0, despite the fact that $y_1y_2 \neq y_2y_1$. This means that the metric axiom $d(x, y) = 0 \Leftrightarrow x = y$ does not hold. Lemma 5 states that all pseudometric properties are satisfied.

Lemma 5: *Let g_1, g_2 and g_3 be any elements of a free partially commutative semigroup G with n generators, then*

1. $D(g_1, g_1) = 0$,
2. $D(g_1, g_2) = D(g_2, g_1)$,
3. $D(g_1, g_3) \leq D(g_1, g_2) + D(g_2, g_3)$.

Next we combine the metric L and the pseudometric D.

Definition 4: *Let g_1 and g_2 be words in G. The vector $(L(g_1, g_2), D(g_1, g_2))$ is called the vector of adjusted logarithmic metric for the words g_1 and g_2.*

The first component of the vector of adjusted logarithmic distance is a value expressing the difference in patterns of the words g_1 and g_2. Meanwhile, the second component is the difference in the number of occurrences for each letter of the set of generators Y.

Definition 5: *Let $g_1, g_2 \in G$. The L_2 norm of the vector of adjusted logarithmic metric for g_1 and g_2 is called an adjusted logarithmic distance between the words g_1 and g_2 and denoted by $\tilde{L}$:*

$$\tilde{L}(g_1, g_2) = \sqrt{L(g_1, g_2)^2 + D(g_1, g_2)^2}.$$

Theorem 1: *The distance function $\tilde{L}$ is a metric, in other words, let g_1, g_2 and g_3 be the elements of a free partially commutative semigroup G, then*

1. $\tilde{L}(g_1, g_2) = 0 \Leftrightarrow g_1 = g_2$,
2. $\tilde{L}(g_1, g_2) = \tilde{L}(g_2, g_1)$,
3. $\tilde{L}(g_1, g_3) \leq \tilde{L}(g_1, g_2) + \tilde{L}(g_2, g_3)$.

Proof. The proof is based on the results of Lemma 4 and Lemma 5. From these two lemmas we have that

(a) $L(g_1, g_2) = 0 \Leftrightarrow g_1 = g_2$,

(b) $L(g_1, g_2) = L(g_2, g_1)$,

(c) $L(g_1, g_3) \leq L(g_1, g_2) + L(g_2, g_3)$,

(d) $D(g_1, g_1) = 0$,

(e) $D(g_1, g_2) = D(g_2, g_1)$,

(f) $D(g_1, g_3) \leq D(g_1, g_2) + D(g_2, g_3)$.

The truthiness of the properties 2 and 3 is obvious. Indeed, $\tilde{L}(g_1, g_2) = \tilde{L}(g_2, g_1)$ follows directly from (b) and (e), and $\tilde{L}(g_1, g_3) \leq \tilde{L}(g_1, g_2) + \tilde{L}(g_2, g_3)$ follows directly from (c) and (f).

Now, let us prove the property 1.

If $\tilde{L}(g_1, g_2) = 0$, then $L(g_1, g_2) = 0$. From the property (a) it follows that the latter can hold only and only if $g_1 = g_2$. So, $\tilde{L}(g_1, g_2) = 0 \Rightarrow g_1 = g_2$.

Now, we prove the opposite. If $g_1 = g_2$, then from the properties (a) and (d) it follows that $L(g_1, g_2) = 0$ and $D(g_1, g_2) = 0$, consequently, $\tilde{L}(g_1, g_2) = 0$. ■

3.3. Definition of $\tilde{L}_H$ and $\tilde{L}_k$ Distances

Once more, we use Hausdorff distance to induce the adjusted $\tilde{L}_H$ metric on the set of all regular events over a free partially commutative semigroup.

Definition 6: *Let E_1 and E_2 be regular events over a free partially commutative semigroup. Adjusted L_H distance between E_1 and E_2 is called the quantity*

$$\tilde{L}_H(E_1, E_2) = \max\left\{\sup_{r_1 \in E_1}\inf_{r_2 \in E_2}\tilde{L}(r_1, r_2), \sup_{r_2 \in E_2}\inf_{r_1 \in E_1}\tilde{L}(r_1, r_2)\right\}.$$

We apply the following assumptions for $\tilde{L}_H$ as well: $\tilde{L}_H(\emptyset, \emptyset) = 0$ and $\tilde{L}_H(\emptyset, P) = \tilde{L}_H(P, \emptyset) = \infty, \forall P \neq \emptyset$.

In Section 4 it will be shown that regular expressions over a free partially commutative semigroup are representable as nondeterministic multitape finite automata. The equivalence problem for the latter ones is proved to be unsolvable [1]. To be able to calculate the distance between them, we have to be able to tell when their distance is 0, which is, as already stated, unsolvable. Hence, the calculation of $\tilde{L}_H$ is unsolvable, so, we introduce a new distance, which takes into account the words accepted by regular expressions, having up to some fixed length. This new distance is an approximation of $\tilde{L}_H$.

Denote by $W_k(P)$ ($k \in \mathbb{N}$ is a fixed number) the following subset of the regular event P:

$$W_k(P) = \{p | p \in P, |p| \leq k\},$$

where $|p|$ is the length of the word p.

Definition 7: *Let E_1 and E_2 be regular events over a free partially commutative semigroup. $\tilde{L}_k$ distance between E_1 and E_2 for a fixed $k \in \mathbb{N}$ is called the quantity*

$$\tilde{L}_k(E_1, E_2) = \tilde{L}_H(W_k(E_1), W_k(E_2)).$$

It is obvious that $\tilde{L}_k$ is a pseudometric.

4. An Algorithm for Calculating the $\tilde{L}_k$ Distance

Denote by $L(R)$ the language of n-tuples of words recognized by the regular expression R.

Theorem 2: (On synthesis of MFA) *There exists an algorithm, which synthesizes an NMFA-ε A from a given regular expression R over a partially commutative semigroup, such that $L(A) = L(R)$.*

One such an algorithm for synthesizing NMFA-ε from a given regular expression might be Thompson's construction [13] used for synthesizing one-tape automata. Indeed, one can easily show that this construction builds an NMFA-ε A such that $L(A) = L(R)$.

Consider regular expressions R_1 and R_2 over a free partially commutative semigroup. The following algorithm calculates the $\tilde{L}_k$ ($k \in \mathbb{N}$) distance between $E(R_1)$ and $E(R_2)$.

1. Construct NMFA-ε for R_1 and R_2 using Thompson's construction, i.e., A_1 and A_2, correspondingly.

2. Find all the extended words accepted by A_1 and A_2 having length less than or equal to k, i.e., $W_k(A_1)$ and $W_k(A_2)$, correspondingly.

3. Calculate $\tilde{L}_H$ distance between the finite sets $W_k(A_1)$ and $W_k(A_2)$.

Now, we calculate the complexity of the proposed algorithm.

Let l_1 and l_2 be the numbers of operations $(+, ^*, \cdot)$ in R_1 and R_2, correspondingly. The first step of the algorithm takes $O(l_i)$ time for R_i ($i = 1, 2$) and constructs an automaton with at most $2l_i$ states [13].

At the second step, the complexity of the construction of set $W_k(A_i)$ ($i = 1, 2$) is $O\left((2l_i)^{2k}\right)$.

At the third step, we construct the binary codings of the words, then their corresponding integer vectors. This takes $c_1 k$ time for a word having k length, where c_1 is some constant. The calculation of the distance between two integer vectors is $c_2 m$, where m is the number of letters in the alphabet.

So, the overall complexity of the proposed algorithm can be estimated as $O(km(2l_1 + 2l_2)^{2k})$.

5. Conclusion

In this paper, a special binary coding of the elements in a free partially commutative semigroup [6] has been considered. This coding is used to define regular expressions for multitape finite automata and a distance, which is shown to be a metric. As the calculation problem of this metric is unsolvable, in order to provide an approximate solution for this problem, a modification of the metric was considered.

A method, having a polynomial complexity, was proposed for approximate calculation of the distance between those regular expressions.

References

[1] M. O. Rabin and D. S. Scott, "Finite Automata and Their Decision Problems", *IBM J. Res. Dev.*, vol. 3, no. 2, pp. 114–125, 1959.

[2] M. Bird, "The Equivalence Problem for Deterministic Two-Tape Automata", *J. Comput. Syst. Sci.*, vol. 7, no. 2, pp. 218–236, 1973.

[3] T. Harju and J. Karhumaki, "The Equivalence Problem of Multitape Finite Automata", *Theor. Comput. Sci.*, vol. 78, no. 2, pp. 347-355, 1991.

[4] B. G. Mirkin, "On the theory of multitape automata", *Cybernetics*, vol. 2, no. 5, pp. 9-14, 1966. doi:10.1007/BF01073664.

[5] P. H. Starke, "On the Representability of Relations by Deterministic and Nondeterministic MultiTape Automata", *International Symposium on Mathematical Foundations of Computer Science*, pp. 114-124, 1975.

[6] A. A. Letichevsky, A. S. Shoukourian and S. K. Shoukourian, "The equivalence problem of deterministic multitape finite automata: a new proof of solvability using a multidimensional tape", *International Conference on Language and Automata Theory and Applications*, pp. 392–402, 2010.

[7] T. A. Grigoryan, "Some Results on Regular Expressions for Multitape Finite Automata", *Proceedings of the YSU: Physical and Mathematical Sciences*, vol. 53, no. 2, pp. 82-90, 2019.

[8] V. M. Glushkov, "The Abstract Theory of Automata", *Russian Mathematical Surveys*, vol. 16, no. 5, pp. 1-53, 1961.

[9] A. B. Godlevskii, A. A. Letichevskii and S. K. Shukuryan, "Reducibility of program-scheme functional equivalence on a nondegenerate basis of rank unity to the equivalence of automata with multidimensional tapes", *Cybernetics*, vol. 16, no. 6, pp. 793-799, 1980.

[10] A. H. Clifford and G. B. Preston, *The Algebraic Theory of Semigroups, Second Edition, ser. Mathematical Surveys and Monographs*, American Mathematical Society, vol. 7.1, 1961.

[11] H. A. Grigoryan and S. K. Shoukourian. "Polynomial algorithm for equivalence problem of deterministic multitape finite automata", *Theor. Comput. Sci.*, vol. 833, pp. 120-132, 2020.

[12] F. Hausdorff, *Set Theory, Fourth Edition*, Chelsea Pub Co, 1991.

[13] A. V. Aho, R. Sethi R and J. D. Ullman, *Compilers: Principles, Techniques, and Tools, World Student Series Edition, ser. Addison-Wesley Series in Computer Science*, 1986.

Submitted 09.06.2020, accepted 05.10.2020.

Բազմաժապավեն վերջավոր ավտոմատների համար կանոնավոր լեզուների միջև հեռավորությունը հաշվող մոտավոր եղանակ

Տիգրան Ա. Գրիգորյան

Երևանի պետական համալսարան
e-mail: tigran.grigoryan1995@gmail.com

Ամփոփում

Դիտարկվում են բազմաժապավեն վերջավոր ավտոմատների կողմից ճանաչվող բառերի կորտեժների բազմություններ և այդ ավտոմատների կողմից ճանաչվող լեզուների վրա սահմանված մետրիկական տարածություն: Այս լեզուները ներկայացված են նույն գրելաձևով, ինչ մեկ ժապավենանոց ավտոմատների կողմից ճանաչվող կանոնավոր լեզուների համար կանոնավոր արտահայտությունները: Միակ տարբերությունը <<կոնկատենացիա>> գործողության մեկնաբանությունն է:

Առաջարկվում է ալգորիթմ, որով հաշվվում է բազմաժապավեն վերջավոր ավտոմատների կողմից ճանաչվող կանոնավոր արտահայտությունների միջև հեռավորությունը՝ ըստ ներմուծված մետրիկայի:

Բանալի բառեր՝ բազմաժապավեն վերջավոր ավտոմատներ, կանոնավոր արտահայտություններ, մետրիկական տարածություն:

Приближенный метод вычисления расстояния между регулярными языками для многоленточных конечных автоматов

Тигран А. Григорян

Ереванский государственный университет
e-mail: tigran.grigoryan1995@gmail.com

Аннотация

Рассматриваются регулярные выражения для многоленточных конечных автоматов. Каждое регулярное выражение описывает язык - множество кортежей слов, принимаемых данным многоленточным конечным автоматом. Также рассмотрено метрическое пространство языков, принимаемых многоленточными конечными автоматами. Эти языки представлены с помощью той же нотации, которая используется в регулярных выражениях для языков, распознаваемых одноленточными автоматами. Единственная разница - в иной интерпретации каждой операции "конкатенация" нотации.

Определена метрика и предложен алгоритм для вычисления расстояния между регулярными выражениями, принимаемыми многоленточными конечными автоматами.

Ключевые слова: многоленточные конечные автоматы, регулярные выражения, метрическое пространство.