ISSN 0361-7688, Programming and Computer Software, 2023, Vol. 49, No. 5, pp. 373–387. © Pleiades Publishing, Ltd., 2023.
Russian Text © The Author(s), 2023, published in Programmirovanie, 2023, Vol. 49, No. 5.

Twenty Similarity Functions for Two Finite Sequences

I. Burdonov[a],* and A. Maksimov[a],**

[a] *Ivannikov Institute for System Programming, Russian Academy of Sciences, ul. Solzhenitsyna 25, Moscow, 109004 Russia*

**e-mail: igor@ispras.ru*

***e-mail: andrew@ispras.ru*

Received January 9, 2023; revised February 16, 2023; accepted March 21, 2023

Abstract—This paper considers various numerical functions that determine the degree of similarity between two finite sequences. These similarity measures are based on the concept of embedding for sequences, which we define here. A special case of this embedding is a subsequence. Other cases additionally require equal distances between adjacent symbols of a subsequence in both sequences. This is a generalization of the concept of the substring with unit distances. Moreover, equality of distances from the beginning of the sequences to the first embedded symbol or from the last embedded symbol to the end of the sequences may be required. In addition to the last two cases, an embedding can occur in the sequence more than once. In the literature, functions such as the number of common embeddings or the number of pairs of occurrences of embeddings in a sequence are used. We introduce three additional functions: the sum of lengths of common embeddings, the sum of the minimum numbers of occurrences of a common embedding in both sequences, and the similarity function based on the longest common embedding. In total, we consider 20 numerical functions; for 17 of these functions, algorithms (including new ones) of polynomial complexity are proposed; for two functions, algorithms of exponential complexity with a reduced exponent are proposed. In Conclusions, we briefly compare these embeddings and functions.

Keywords: sequence analysis, common subsequences, longest and maximal common subsequences, canonical embedding, matching joint embeddings, algorithms for subsequence and embeddings combinatorics, similarity axioms

DOI: 10.1134/S0361768823050031

1. INTRODUCTION

Sequence analysis is widely used in social, management, political, demographic, and psychological sciences, as well as in chemistry, bioinformatics, and text processing. In recent decades, many papers devoted to this problem have been published [1–6]. Various metrics and measures of similarity for sequences have been used [4, 5].

In this paper, we consider various numerical functions that determine the degree of similarity between two finite sequences. These similarity measures are based on the concept of embedding for sequences, which we define below. A special case of this embedding is a subsequence. Other cases additionally take into account distances between symbols of a subsequence in both sequences. For instance, sequences "METRICA" and "MOROCCO" have the longest common subsequence "MRC", taking into account the distances between symbols "R-C", the distances between the symbols and from the last embedded symbol to the end of the sequence "C-", and the distances between the symbols and from the beginning of the sequence to the first symbol of the "M—C" embedding.

The concept of embedding is introduced using an empty symbol, which does not belong to the alphabet of the sequences. In total, embeddings of five types are introduced: *E*-embedding is obtained by replacing some symbols with an empty symbol, *L*-embedding is obtained from *E*-embedding by removing the empty prefix, *R*-embedding is obtained from *E*-embedding by removing the empty postfix, *O*-embedding is obtained from *E*-embedding by removing the empty prefix and empty postfix, and *A*-embedding is obtained from *E*-embedding by removing all empty symbols, which coincides with the concept of a subsequence. The identifiers of the embeddings are derived from the corresponding English words: *E* is **E**mpty symbol, followed by an indication of a place where there are no empty symbols; *L* is on the **L**eft; *R* is on the **R**ight; *O* is **O**utside (i.e. on the left and right); and *A* is **A**nywhere (i.e., there are no empty symbols anywhere).

Each embedding can have multiple occurrences in a sequence, i.e., several *E*-embeddings, from which a given embedding is obtained by removing the prefix,

postfix, prefix and postfix, or all empty symbols. For instance, subsequence "MRC" occurs once in sequence "METRICA" (the corresponding E-embedding is "M--R-C-") and twice in sequence "MOROCCO" (the corresponding E-embeddings are "M-R-C--" and "M-R--C-"). For embeddings that can contain empty symbols, the concept of μ-length is defined as the number of non-empty symbols (for A-embedding, it coincides with the length of the subsequence).

For each of four embedding types (L, R, O and A), five functions are defined: (0) the number of common embeddings, (1) the sum of μ-lengths of common embeddings, (2) the sum of the minimum numbers of occurrences of common embeddings in given sequences, (3) the sum of products of the numbers of occurrences of common embeddings in given sequences, and (4) similarity measure based on the longest common subsequence (lcs).

Some of these 20 functions are well known, e.g., the number of common subsequences (the number of common A-embeddings) or the sum of products of the numbers of occurrences of common subsequences in given sequences (the sum of products of the numbers of E-embeddings of common A-embeddings in given sequences) [3]. The other functions, especially those that take into account distances between symbols, are introduced in this paper.

This paper is organized as follows. Section 2 introduces the basic concepts and notation. Section 3 considers an optimization common to embeddings of all types: replacing the symbols that occur only in one of two sequences with an empty symbol. Sections 4–7 discuss embeddings of four types (L, R, O, and A); for each type of embeddings, algorithms for evaluating five functions 0, 1, 2, 3, and 4 are considered. In Conclusions, the results are summarized and some directions for further research are outlined.

2. DEFINITIONS AND NOTATION

For integers i and j, we denote $i..j = \{i, i+1, ..., j\}$ if $i \le j$ and $i..j = \varnothing$ if $i > j$.

A finite sequence in alphabet H of length $m \ge 0$ is an injection of set $1..m$ into set H: $1..m \to H$. The set of finite sequences in alphabet H is denoted by H^*. An empty sequence (of zero length, an empty injection) is denoted by (). For a non-empty finite sequence x, its ith element ($i \in 1..|x|$) is denoted by $x_i = x(i)$. Segment $x_i, x_{i+1}, ..., x_j$ for $1 \le i \le j \le |x|$ is denoted by $x[i..j]$. For $i > j$, we define $x[i..j] = ()$. A prefix is denoted by $x[j] = x[1..j]$, and an empty prefix (of zero length) is also defined for an empty sequence $()[0] = ()$. A finite sequence of $k \ge 0$ repetitions of symbol $h \in H$ is denoted by h^k: $|h^k| = k \,\&\, \forall\, i = 1..|k| h_i^k = h$. Instead of h^1, we simply write h.

Concatenation xy of finite sequences x and y is defined as follows: $|xy| = |x| + |y|$, $\forall\, i \in 1..|x|\ (xy)_i = x_i$ and $\forall\, j \in 1..|y|\ (xy)_{|x|+j} = y_j$. We also define the concatenation of a pair (X, Y) of finite sets of finite sequences with a pair (z, t) of finite sequences: $(X, Y)(z, t) = \{(xz, yt) : (x, y) \in (X, Y)\}$. It is assumed that the concatenation has priority over operations on sets (union, intersection, and difference).

The composition of functions f and g is denoted by fg.

Let us introduce an *empty symbol* $\varepsilon \notin H$ and denote $H_\varepsilon = H \cup \{\varepsilon\}$.

The set of finite sequences in alphabet H_ε is denoted by

- $L(H) = \{v \in H_\varepsilon^* : |v| > 0 \Rightarrow v_1 \ne \varepsilon\}$ if the sequences do not begin with an empty symbol;
- $R(H) = \{v \in H_\varepsilon^* : |v| > 0 \Rightarrow v_{|v|} \ne \varepsilon\}$ if the sequences do not end with an empty symbol;
- $O(H) = \{v \in H_\varepsilon^* : |v| > 0 \Rightarrow v_1 \ne \varepsilon \,\&\, v_{|v|} \ne \varepsilon\} = L(H) \cap R(H)$ if the sequences do not begin and end with empty symbols.

Let us introduce the following operations for removing empty symbols from a finite sequence in H_ε:

- *removing the prefix of empty symbols* λ: $H_\varepsilon^* \to L(H)$ is defined by the condition

$$\forall v \in H_\varepsilon^* \lambda(\varepsilon \cdot v) = \lambda(v) \,\&\, \forall u \in L(H) \lambda(u) = u;$$

- *removing the postfix of empty symbols* ρ: $H_\varepsilon^* \to R(H)$ is defined by the condition

$$\forall v \in H_\varepsilon^* \rho(v \cdot \varepsilon) = \rho(v) \,\&\, \forall u \in R(H) \rho(u) = u;$$

- *removing all empty symbols* μ: $H_\varepsilon^* \to H^*$ is defined by the condition

$$\forall v, w \in H_\varepsilon^* \mu(v \cdot \varepsilon \cdot \mu) = \mu(v \cdot w) \,\&\, \forall u \in H^* \mu(u) = u.$$

For sequence $x \in H_\varepsilon^*$,

- *E-embedding* is obtained from x by replacing some symbols with empty symbols;
- *L-embedding* is obtained from E-embedding by removing the prefix of empty symbols;
- *R-embedding* is obtained from E-embedding by removing the postfix of empty symbols;
- *O-embedding* is obtained either from E-embedding by removing the prefix and postfix of empty symbols, or from L-embedding by removing the postfix of empty symbols, or from R-embedding by removing the prefix of empty symbols;
- *A-embedding* is obtained from E-, L-, R-, or O-embedding by removing empty symbols.

For $x \in H^*$, the concept of A-embedding coincides with the concept of a subsequence, while E-embed-

ding v corresponds to the occurrence of subsequence $\mu(v)$ in x.

Let us denote the following sets of embeddings in sequence $x \in H^*$:

• the set of E-embeddings in x is

$$E(x) = \{u \in H_\varepsilon^* : |u| = |x| \,\&\, \forall i \in 1..|u| u_i = x_i \vee u_i = \varepsilon\};$$

• the set of L-embeddings in x is

$$L(x) = \lambda E(x);$$

• the set of R-embeddings in x is

$$R(x) = \rho E(x);$$

• the set of O-embeddings in x is

$$O(x) = \lambda r E(x) = \lambda R(x) = \rho L(x);$$

• the set of A-embeddings in x is

$$A(x) = \mu E(x) = \mu L(x) = \mu R(x) = \mu O(x).$$

For a sequence x in alphabet H_ε^*, we introduce the concept of μ-*length* of x as the number of non-empty symbols in x, which is obviously equal to $|\mu(x)|$. For $x \in H^*$, its μ-length is equal to its actual length.

For sequences $x \in H^*$, we denote

• the set of E-embeddings in x for L-embedding by $u \in L(x)$: $l(u, x) = \{v \in E(x) : \lambda(v) = u\}$;

• the set of E-embeddings in x for R-embedding by $u \in R(x)$: $r(u, x) = \{v \in E(x) : \rho(v) = u\}$;

• the set of E-embeddings in x for O-embedding by $u \in O(x)$: $o(u, x) = \{v \in E(x) : \lambda\rho(v) = u\}$;

• the set of E-embeddings in x for A-embedding by $u \in A(x)$: $a(u, x) = \{v \in E(x) : \mu(v) = u\}$.

For sequences $x \in H^*$ and $y \in H^*$, we denote

• the set of pairs of E-embeddings for common L-embeddings by

$$L(x, y) = \cup\{l(u, x) \times l(u, y) : u \in L(x) \cap L(y)\};$$

• the set of pairs of E-embeddings for common R-embeddings by

$$R(x, y) = \cup\{r(u, x) \times r(u, y) : u \in R(x) \cap R(y)\};$$

• the set of pairs of E-embeddings for common O-embeddings by

$$O(x, y) = \cup\{o(u, x) \times o(u, y) : u \in O(x) \cap O(y)\};$$

• the set of pairs of E-embeddings for common A-embeddings by

$$A(x, y) = \cup\{a(u, x) \times a(u, y) : u \in A(x) \cap A(y)\}.$$

For sequences $x \in H^*$ and $y \in H^*$, we denote the longest (in terms of μ-length)

• common L-embeddings by $lcL(x, y) = max\{\mu(u) : u \in L(x) \cap L(y)\}$;

• common R-embeddings by $lcR(x, y) = max\{\mu(u) : u \in R(x) \cap R(y)\}$;

• common O-embeddings by $lcO(x, y) = max\{\mu(u) : u \in O(x) \cap O(y)\}$;

• common A-embeddings by $lcA(x, y) = max\{|u| : u \in A(x) \cap A(y)\}$.

It is natural to regard the maximum μ-length of a common embedding as another similarity function for sequences x and y: $lcI(x, y)$ for $I \in \{L, R, O, A\}$. In accordance with this criterion, sequence x is most "similar" to itself because it is the longest I-embedding in itself: $lcI(x, x) = max\{\mu(u) : u \in I(x)\} = |x| \geq max\{\mu(u) : u \in I(x) \cap I(y)\} = lcI(x, y)$.

We focus on functions of sequences $x \in H^*$ and $y \in H^*$ that are defined in Table 1.

Note that, for $I \in \{L, R, O, A\}$ and $j \in 0..4$, $I_j(x, y) = I_j(y, x)$. If $j \neq 1$, then $I_j(x, ()) = I_j((), y) = 1$; $I_1(x, ()) = I_1((), y) = 0$.

For non-empty sequence $x \in H^*$,

• the left-most E-embedding in x for L-embedding is denoted by $u \in L(x)$: $l_l(u, x) = v$ if $v \in l(u, x)$ and $\forall w \in l(u, x)\backslash\{v\}$ $w(min\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• the right-most E-embedding in x for L-embedding is denoted by $u \in O(x)$: $l_r(u, x) = v$ if $v \in l(u, x)$ and $\forall w \in l(u, x)\backslash\{v\}$ $w(max\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• the left-most E-embedding in x for R-embedding is denoted by $u \in R(x)$: $r_l(u, x) = v$ if $v \in r(u, x)$ and $\forall w \in r(u, x)\backslash\{v\}$ $w(min\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• the right-most E-embedding in x for R-embedding is denoted by $u \in O(x)$: $r_r(u, x) = v$ if $v \in r(u, x)$ and $\forall w \in r(u, x)\backslash\{v\}$ $w(max\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• the left-most E-embedding in x for O-embedding is denoted by $u \in O(x)$: $o_l(u, x) = v$ if $v \in o(u, x)$ and $\forall w \in o(u, x)\backslash\{v\}$ $w(min\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• the right-most E-embedding in x for O-embedding is denoted by $u \in O(x)$: $o_r(u, x) = v$ if $v \in o(u, x)$ and $\forall w \in o(u, x)\backslash\{v\}$ $w(max\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• the left-most E-embedding in x for A-embedding is denoted by $u \in A(x)$: $a_l(u, x) = v$ if $v \in a(u, x)$ and $\forall w \in a(u, x)\backslash\{v\}$ $w(min\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• the right-most E-embedding in x for A-embedding is denoted by $u \in A(x)$: $a_r(u, x) = v$ if $v \in a(u, x)$ and $\forall w \in a(u, x)\backslash\{v\}$ $w(max\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$.

In the literature, the left-most E-embeddings of common embeddings are also called (for subsequences) *canonical* embeddings [3].

For non-empty sequences $x \in H^*$ and $y \in H^*$, we denote

• the set of pairs of left-most E-embeddings for common L-embeddings by

$$L_l(x, y) = \{(l_l(u, x), l_l(u, y)) : u \in L(x) \cap L(y)\};$$

• the set of pairs of right-most E-embeddings for common L-embeddings by

$$L_r(x, y) = \{(l_r(u, x), l_r(u, y)) : u \in L(x) \cap L(y)\};$$

Table 1. Functions for common embeddings of two sequences x and y

Embedding	Function				
	Number of common embeddings	Sum of μ-lengths of common embeddings	Sum of minimum numbers of E-embeddings in common embeddings	Sum of products of numbers of E-embeddings in common embeddings	μ-Longest common embedding
	0	1	2	3	4
L	$L_0(x, y) = \|L(x) \cap L(y)\|$	$L_1(x, y) = \Sigma\{\|\mu(u)\| : u \in L(x) \cap L(y)\}$	$L_2(x, y) = \Sigma\{min\{\|l(u, x)\|, \|l(u, y)\|\} : u \in L(x) \cap L(y)\}$	$L_3(x, y) = \|L(x, y)\|$	$L_4(x, y) = lcL(x, y)$
R	$R_0(x, y) = \|R(x) \cap R(y)\|$	$R_1(x, y) = \Sigma\{\|\mu(u)\| : u \in R(x) \cap R(y)\}$	$R_2(x, y) = \Sigma\{min\{\|r(u, x)\|, \|r(u, y)\|\} : u \in R(x) \cap R(y)\}$	$R_3(x, y) = \|R(x, y)\|$	$R_4(x, y) = lcR(x, y)$
O	$O_0(x, y) = \|O(x) \cap O(y)\|$	$O_1(x, y) = \Sigma\{\|\mu(u)\| : u \in O(x) \cap O(y)\}$	$O_2(x, y) = \Sigma\{min\{\|o(u, x)\|, \|o(u, y)\|\} : u \in O(x) \cap O(y)\}$	$O_3(x, y) = \|O(x, y)\|$	$O_4(x, y) = lcO(x, y)$
A	$A_0(x, y) = \|A(x) \cap A(y)\|$	$A_1(x, y) = \Sigma\{\|u\| : u \in A(x) \cap A(y)\}$	$A_2(x, y) = \Sigma\{min\{\|a(u, x)\|, \|a(u, y)\|\} : u \in A(x) \cap A(y)\}$	$A_3(x, y) = \|A(x, y)\|$	$A_4(x, y) = lcA(x, y)$

• the set of pairs of left-most E-embeddings for common R-embeddings by

$$R_l(x, y) = \{(r_l(u, x), r_l(u, y)) : u \in R(x) \cap R(y)\};$$

• the set of pairs of right-most E-embeddings for common R-embeddings by

$$R_r(x, y) = \{(r_r(u, x), r_r(u, y)) : u \in R(x) \cap R(y)\};$$

• the set of pairs of left-most E-embeddings for common O-embeddings by

$$O_l(x, y) = \{(o_l(u, x), o_l(u, y)) : u \in O(x) \cap O(y)\};$$

• the set of pairs of right-most E-embeddings for common O-embeddings by

$$O_r(x, y) = \{(o_r(u, x), o_r(u, y)) : u \in O(x) \cap O(y)\};$$

• the set of pairs of left-most E-embeddings for common A-embeddings by

$$A_l(x, y) = \{(a_l(u, x), a_l(u, y)) : u \in A(x) \cap A(y)\};$$

• the set of pairs of right-most E-embeddings for common A-embeddings by

$$A_r(x, y) = \{(a_r(u, x), a_r(u, y)) : u \in A(x) \cap A(y)\}.$$

Note that $|O_l(x, y)| = |O_r(x, y)| = |O(x) \cap O(y)|$ and $|A_l(x, y)| = |A_r(x, y)| = |A(x) \cap A(y)|$.

Below, x and y denote two non-empty sequences in alphabet H with lengths $m = |x|$ and $n = |y|$. It is assumed that $m \le n$.

3. REPLACEMENT OF NON-COMMON SYMBOLS WITH EMPTY SYMBOLS

A general optimization for evaluation of all functions for all embeddings is to replace non-common symbols with empty symbols: for $i \in 1..m$, we define $x_i^\wedge = x_i$ if $x_i \in \mathbf{Im}\, y$ and $x_i^\wedge = \varepsilon_i$ if $x_i \notin \mathbf{Im}\, y$; $y^\wedge$ is defined similarly. This replacement is carried out in $\mathbf{O}(mn)$ time. In the case of A-embeddings, instead of replacing a non-common symbol with an empty symbol, we can simply remove the non-common symbol. The algorithms described below can be applied upon performing this replacement (removal for A-embeddings).

4. A-EMBEDDINGS (SUBSEQUENCES)

4.1. Number of Common A-Embeddings

Here, we prove a theorem similar to Lemma 6 from [3], but we prove it differently using different definitions and notation.

For non-empty sequence $z \in H^*$ and symbol $h \in H$, we define the maximum index by which symbol h is found in sequence z (or 0 if h is not included in z):

$p(z, h) = max\{i \in 1..|z| : z_i = h\}$ if $h \in \mathbf{Im}\, z$; $p(z, h) = 0$ if $h \notin \mathbf{Im}\, z$.

We denote $k = p(x[m-1], x_m\}$ and $l = p(y, x_m\}$.

Theorem 1.

$A_0(x, y) = A_0(x[m-1], y)$ if $x_m \notin \mathbf{Im}\, y$;

$A_0(x, y) = A_0(x[m-1], y) + A_0(x[m-1], y[l-1])$ if $x_m \in \mathbf{Im}\, y$ and $x_m \notin \mathbf{Im}\, x[m-1]$;

$A_0(x, y) = A_0(x[m-1], y) + A_0(x[m-1], y[l-1]) - A_0(x[k-1], y[l-1])$ if $x_m \in \mathbf{Im}\, y$ and $x_m \in \mathbf{Im}\, x[m-1]$.

Proof.

We consider sets of pairs $(e_r(u, x), e_r(u, y))$ of right-most E-embeddings for common A-embedding u of sequences x and y. Pairs of sequences of lengths m and n are denoted as follows:

$E = A_r(x, y)$;

$E_m = A_r(x[m-1], y)(\varepsilon, ())$;

$E_{ml} = A_r(x[m-1], y[l-1])(x_m, x_m\varepsilon^{n-l})$ if $x_m \in \mathbf{Im}\, y$; otherwise, $E_{ml} = \varnothing$;

$E_{kl} = A_r(x[k-1], y[l-1])(x_m\varepsilon^{m-k-1}x_m, x_m\varepsilon^{n-l})$ if $x_m \in \mathbf{Im}\, y$ and $x_m \in \mathbf{Im}\, x[m-1]$; otherwise, $E_{kl} = \varnothing$.

Since $A_0(x', y') = |A(x') \cap A(y')| = |A_r(x', y')|$ for any x' and y', the statement of the theorem can be rewritten as follows:

$|E| = |E_m|$ if $x_m \notin \mathbf{Im}\, y$;

$|E| = |E_m| + |E_{ml}|$ if $x_m \in \mathbf{Im}\, y$ and $x_m \notin \mathbf{Im}\, x[m-1]$;

$|E| = |E_m| + |E_{ml}| - |E_{kl}|$ if $x_m \in \mathbf{Im}\, y$ and $x_m \in \mathbf{Im}\, x[m-1]$.

With E-embeddings from set E_m having the form (...ε, ...) and sets E_{ml}, E_{kl} either being empty or E-embeddings from them having the form (...x_m, ...), we have $E_m \cap E_{ml} = E_m \cap E_{kl} = \varnothing$. For $m - 1 \geq k - 1$, we have $E_{ml} \supseteq E_{kl}$.

Let us consider the case where $x_m \notin \mathbf{Im}\, y$. Since $x_m \notin \mathbf{Im}\, y$, the transition from sequence $x[m-1]$ to sequence $x = x[m-1]x_m$ does not add new common A-embeddings. That is why $E = E_m$. Hence, $|E| = |E_m|$, which was to be proved in this case.

Let us consider the case where $x_m \in \mathbf{Im}\, y$ and $x_m \notin \mathbf{Im}\, x[m-1]$. Since $x_m \in \mathbf{Im}\, y$, the transition from sequence $x[m-1]$ to sequence $x = x[m-1]x_m$ can add new common A-embeddings; these common A-embeddings must end with x_m. However, these common A-embeddings were absent in $x[m-1]$ and y before because $x_m \notin \mathbf{Im}\, x[m-1]$. Note that, for this new common A-embedding u in its right-most E-embeddings in x and y, the last non-empty symbol is x_m by indices m and l, respectively, i.e., $e_r(u, x)_m = e_r(u, y)_l = x_m$. We have $E = E_m \cup E_{ml}$. Since $E_m \cap E_{ml} = \varnothing$, we have $|E| = |E_m| + |E_{ml}|$, which was to be proved in this case.

Let us consider the case where $x_m \in \mathbf{Im}\, y$ and $x_m \in \mathbf{Im}\, x[m-1]$. Since $x_m \in \mathbf{Im}\, y$, the transition from sequence $x[m-1]$ to sequence $x = x[m-1]x_m$ can add new common A-embedding u; however, they are new only if there were no such A-embeddings in $x[m-1]$ and y before. These new common A-embeddings must end with x_m. Note that, if this common A-embedding u was present before, then, in its right-most E-embeddings in $x[m-1]$ and y, the last non-empty symbol is x_m by indices k and l, respectively, i.e., $e_r(u, x[m-1])_k = e_r(u, y)_l = x_m$. In any case, in right-most E-embeddings u in x and y, the last non-empty symbol is x_m by indices m and l, respectively, i.e., $e_r(u, x)_m = e_r(u, y)_l = x_m$. We have $E = E_m \cup (E_{ml} \backslash E_{kl})$. With $E_m \cap E_{ml} = E_m \cap E_{kl} = \varnothing$, we have $E_m \cap (E_{ml} \backslash E_{kl}) = \varnothing$ and, therefore, $|E| = |E_m| + |E_{ml} \backslash E_{kl}|$. With $E_{ml} \supseteq E_{kl}$, we have $|E_{ml} \backslash E_{kl}| = |E_{ml}| - |E_{kl}|$. Hence, $|E| = |E_m| + |E_{ml}| - |E_{kl}|$, which was to be proved.

□

Theorem 1 defines an algorithm for evaluating $A_0(x, y)$. The number of algorithm steps is $\mathbf{O}(mn)$, which is determined by the number of functions of form $A_0(x[m-i], y[n-j])$, where $i \in 0..m$ and $j \in 0..n$, provided that each function is evaluated at most once (after which its value is saved). At each step, the check of conditions $x_{m-i} \in \mathbf{Im}\, y[n-j]$ and $x_{m-i} \in \mathbf{Im}\, x[m-i-1]$ has complexity $\mathbf{O}(m+n)$, while the other computations have complexity $\mathbf{O}(1)$. The complexity of the algorithm is $\mathbf{O}(m^2n + mn^2)$, which corresponds to $\mathbf{O}(mn^2)$ for $m \leq n$.

4.2. Sum of Lengths of Common A-Embeddings

Theorem 2.

$A_1(x, y) = A_1(x[m-1], y)$ if $x_m \notin \mathbf{Im}\, y$;

$A_1(x, y) = A_1(x[m-1], y) + A_1(x[m-1], y[l-1]) + A_0(x[m-1], y[l-1])$ if $x_m \in \mathbf{Im}\, y$ and $x_m \notin \mathbf{Im}\, x[m-1]$;

$A_1(x, y) = A_1(x[m-1], y) + A_1(x[m-1], y[l-1]) + A_0(x[m-1], y[l-1]) - A_1(x[k-1], y[l-1]) - A_0(x[k-1], y[l-1])$ if $x_m \in \mathbf{Im}\, y$ and $x_m \in \mathbf{Im}\, x[m-1]$.

Proof.

The proof is similar to that of Theorem 1. We use the same notation:

$E = A_r(x, y)$;

$E_m = A_r(x[m-1], y)(\varepsilon, ())$;

$E_{ml} = A_r(x[m-1], y[l-1])(x_m, x_m\varepsilon^{n-l})$ if $x_m \in \mathbf{Im}\, y$; otherwise, $E_{ml} = \varnothing$;

$E_{kl} = A_r(x[k-1], y[l-1])(x_m\varepsilon^{m-k-1}x_m, x_m\varepsilon^{n-l})$ if $x_m \in \mathbf{Im}\, y$ and $x_m \in \mathbf{Im}\, x[m-1]$; otherwise, $E_{kl} = \varnothing$.

We have $E_m \cap E_{ml} = E_m \cap E_{kl} = \varnothing$ and $E_{ml} \supseteq E_{kl}$.

In addition, we have $A_0(x[m-1], y[l-1]) = |E_{ml}|$ and $|A_0(x[k-1], y[l-1])| = |E_{kl}|$.

Let us denote

$S = \Sigma\{|u| : u \in A(x) \cap A(y)\ \&\ (e_r(u, x), e_r(u, y)) \in E\}$;

$S_m = \Sigma\{|u| : u \in A(x) \cap A(y)\ \&\ (e_r(u, x), e_r(u, y)) \in E_m\}$;

$S_{ml} = \Sigma\{|u| : u \in A(x) \cap A(y)\ \&\ (e_r(u, x), e_r(u, y)) \in E_{ml}\}$ if $x_m \in \mathbf{Im}\, y$; otherwise $E_{ml} = \varnothing$;

$S_{kl} = \Sigma\{|u| : u \in A(x) \cap A(y)\ \&\ (e_r(u, x), e_r(u, y)) \in E_{kl}\}$ if $x_m \in \mathbf{Im}\, y$ and $x_m \in \mathbf{Im}\, x[m-1]$; otherwise $E_{kl} = \varnothing$.

The length of common A-embedding u is equal to the number of non-empty symbols in each of its E-embeddings, including the right-most E-embedding. Therefore, in this notation, the statement of the theorem can be rewritten as follows:

$S = S_m$ if $x_m \notin \mathbf{Im}\, y$;

$S = S_m + S_{ml} + |E_{ml}|$ if $x_m \in \mathbf{Im}\, y$ and $x_m \notin \mathbf{Im}\, x[m-1]$;

$S = S_m + S_{ml} + |E_{ml}| - S_{kl} - |E_{kl}|$ if $x_m \in \mathbf{Im}\, y$ and $x_m \in \mathbf{Im}\, x[m-1]$.

When passing from sequence $x[m-1]$ to sequence $x = x[m-1]x_m$, the number of non-empty symbols in the right-most E-embedding in x of common A-embedding u does not change if it includes, by index m, an empty symbol $e_r(u, x)_m = \varepsilon$; otherwise, it is incremented by 1 when $e_r(u, x)_m = x_m$.

This implies the following conditions.

In the case where $x_m \notin \mathbf{Im}\, y$, we have $E = E_m$; for each $(u, v) \in E_m$, we have $u_m = \varepsilon$; therefore, $S = S_m$, which was to be prove in this case.

For $x_m \in \mathbf{Im}\, y$ and $x_m \notin \mathbf{Im}\, x[m-1]$, we have $E = E_m \cup E_{ml}$ and $E_m \cap E_{ml} = \varnothing$; for each $(u, v) \in E_m$, holds $u_m = \varepsilon$; for each $(u, v) \in E_{ml}$, holds $u_m = x_m$; therefore, $S = S_m + (S_{ml} + |E_{ml}|)$, which was to be proved in this case.

For $x_m \in \mathbf{Im}\, y$ and $x_m \in \mathbf{Im}\, x[m-1]$, we have $E = E_m \cup (E_{ml} \backslash E_{kl})$ and $E_m \cap E_{ml} = E_m \cap E_{kl} = \varnothing$; for each $(u, v) \in E_m$, holds $u_m = \varepsilon$; for each $(u, v) \in E_{ml}$, holds $u_m = x_m$; for each $(u, v) \in E_{kl}$, holds $u_m = x_m$; therefore, $S = S_m + (S_{ml} + |E_{ml}|) - (S_{kl} + |E_{kl}|)$, which was to be proved in this case.

□

Theorem 2 defines an algorithm for evaluating $A_1(x, y)$; its complexity obviously does not exceed by an order of magnitude the complexity of the algorithm for evaluating $A_0(x, y)$, i.e., it is $\mathbf{O}(m^2 n + mn^2)$, which corresponds to $\mathbf{O}(mn^2)$ for $m \le n$.

4.3. Sum of the Minimum Numbers of E-Embeddings in Common A-Embeddings

For function $A_2(x, y)$, we are not aware of a good algorithm (except exhaustive search). The only useful optimization is preliminary removal of non-common symbols.

4.4. Sum of Products of the Numbers of E-Embeddings in Common A-Embeddings

Here, we prove a theorem similar to Theorem 2 from [3]; the difference is that we take into account an empty subsequence, whereas Theorem 2 from [3] does not, and use different definitions and notation.

Theorem 3. $A_3(x, y) = A_3(x[m-1], y) + A_3(x, y[n-1]) - A_3(x[m-1], y[n-1])$ if $x_m \ne y_n$;

$A_3(x, y) = A_3(x[m-1], y) + A_3(x, y[n-1])$ if $x_m = y_n$.

Proof.

We denote the following sets of pairs of E-embeddings for common A-embeddings as follows:

$E = A(x, y)$,

$E_{00} = A(x[m-1], y[n\text{-}1])(\varepsilon, \varepsilon)$ is the pair of last elements $(\varepsilon, \varepsilon)$,

$E_{01} = (A(x[m-1], y])(\varepsilon, ()))\backslash E_{00}$ is the pair of last elements (ε, y_n),

$E_{10} = (A(x, y[n-1])((), \varepsilon))\backslash E_{00}$ is the pair of last elements (x_m, ε),

$E_{11} = E\backslash(E_{00} \cup E_{01} \cup E_{10})$ is the pair of last elements (x_m, y_n), since $E_{00} \cup E_{01} \cup E_{10}$ contain all pairs of E-embeddings of common A-embeddings in which the pair of last elements contains ε.

Obviously, $E = E_{00} \cup E_{01} \cup E_{10} \cup E_{11}$. Pairs of last elements of pairs of E-embeddings from different sets E_{00}, E_{01}, E_{10}, and E_{11} are different, which is why these sets are pairwise disjoint. Therefore, $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}|$, $|E_{00} \cup E_{01}| = |E_{00}| + |E_{01}|$, and $|E_{00} \cup E_{10}| = |E_{00}| + |E_{10}|$.

With $A_3(x, y) = |A(x, y)|$, in this notation, the statement of the theorem has the following form:

$|E| = (|E_{00}| + |E_{01}|) + (|E_{00}| + |E_{10}|) - |E_{00}| = |E_{00}| + |E_{01}| + |E_{10}|$ if $x_m \ne y_n$,

$|E| = (|E_{00}| + |E_{01}|) + (|E_{00}| + |E_{10}|) = 2|E_{00}| + |E_{01}| + |E_{10}|$ if $x_m = y_n$.

Let us consider the case where $x_m \ne y_n$. In this case, $E_{11} = \varnothing$; this implies $|E_{11}| = 0$ and $|E| = |E_{00}| + |E_{01}| + |E_{10}|$, which was to be proved in this case.

Let us consider the case where $x_m = y_n = h$. Each pair of E-embeddings from E_{11} has form $(u\varepsilon^{m-|u|-1}h, v\varepsilon^{n-|v|-1}h)$, where $(u\varepsilon^{m-|u|-1}\varepsilon, v\varepsilon^{n-|v|-1}\varepsilon) \in E_{00}$ (sequences u and v can both be empty). We say that pair $(u\varepsilon^{m-|u|-1}h, v\varepsilon^{n-|v|-1}h)$ corresponds to pair $(u\varepsilon^{m-|u|-1}\varepsilon, v\varepsilon^{n-|v|-1}\varepsilon)$. This correspondence is obviously a bijection of sets E_{11} and E_{00}. Therefore, $|E_{11}| = |E_{00}|$. Hence, $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{00}| = 2|E_{00}| + |E_{01}| + |E_{10}|$, which was to be proved.

□

Theorem 3 defines an algorithm for evaluating $A_3(x, y)$. The number of algorithm steps is $\mathbf{O}(mn)$, which is determined by the number of functions of form $A_3(x[m-i], y[n-j])$, where $i \in 0..m$ and $j \in 0..n$, provided that each function is evaluated at most once (after which its value is saved). At each step, the computations have complexity $\mathbf{O}(1)$. Thus, the complexity of the algorithm is $\mathbf{O}(mn)$.

4.5. Similarity Function Based on the Longest Common A-Embedding

The problem of calculating the length of the longest common subsequence (lcs) is well known [1]. The simplest algorithm with complexity $\mathbf{O}(mn)$ is based on the following relations:

$lcA(x, y) = lcA(x[m-1], y[n-1]) + 1$ if $x_m = y_n$;

$lcA(x, y) = max\{lcA(x, y[n-1]), lcA(x[m-1], y)\}$ if $x_m \neq y_n$.

Thus, function $A_4(x, y) = lcA(x, y)$ is evaluated in $\mathbf{O}(mn)$ time.

5. L-EMBEDDINGS

In contrast to A- and O-embeddings, L-embedding $u \in L(x)$ has only one E-embedding v such that $\lambda(v) = u$. Common L-embedding $u \in L(x) \cap L(y)$ has a pair of E-embeddings in x and y, which can differ only in the prefix of empty symbols. That is why set $l(u, x)$ is a singleton, $\{l_l(u, x)\} = \{l_r(u, x)\} = l(u, x)$ and $L(x, y) = L_l(x, y) = L_r(x, y)$.

By $\gamma(x, y)$, we denote a sequence z of length m (recall that $|x| = m \leq n = |y|$) that coincides with x and y in those positions (from right to left) in which x and y coincide, as well as has empty symbols in the other positions: $|z| = m$ and $\forall\, i \in 1..m\ (x_{m+1-i} = y_{n+1-i} \Rightarrow z_{m+1-i} = x_{m+1-i})\ \&\ (x_{m+1-i} \neq y_{n+1-i} \Rightarrow z_{m+1-i} = \varepsilon)$. Function $\gamma(x, y)$ establishes a positional correspondence between coinciding symbols of x and y when counting positions from right to left.

5.1. Number of Common L-Embeddings

Theorem 4. $L_0(x, y) = L_0(x[m-1], y[n-1]))$ if $x_m \neq y_n$,

$L_0(x, y) = 2L_0(x[m-1], y[n-1])$ if $x_m = y_n$.

Proof.

We denote $L = L(x) \cap L(y)$ and $L_{-1} = L(x[m-1]) \cap L(y[n-1])$. If $x_m \neq y_n$, then L-embedding $u \in L_{-1}$ corresponds to one L-embedding $u\varepsilon \in L$. Hence, $L_0(x, y) = |L| = |L_{-1}| = L_0(x[m-1], y[n-1]))$. If $x_m = y_n$, then L-embedding $u \in L_{-1}$ corresponds to two L-embeddings $u\varepsilon \in L$ and $ux_m \in L$. Hence, $L_0(x, y) = |L| = 2|L_{-1}| = 2L_0(x[m-1], y[n-1]))$.

□

Theorem 4 defines an algorithm for evaluating $L_0(x, y)$. For $m \leq n$, the number of algorithm steps is $\mathbf{O}(m)$, which is determined by the number of functions of form $L_0(x[m-i], y[n-i])$, where $i \in 0..m$. At each step, the computations have complexity $\mathbf{O}(1)$. Thus, the complexity of the algorithm is $\mathbf{O}(m)$.

Theorem 5. $L_0(x, y) = 2^{|\mu\gamma(x, y)|}$.

Proof.

Theorem 4 implies that, when adding one symbol to both sequences on the right, the number of common L-embeddings doubles if the same symbols are added; otherwise, it does not change. With $|L(x) \cap L(())| = |L(()) \cap L(y)| = |\varepsilon| = 1$ and $|\mu\gamma(x, y)|$ being equal to the number of coinciding symbols in the same positions of x and y from right to left, we have $L_0(x, y) = |L(x) \cap L(y)| = 2^{|\mu\gamma(x, y)|}$, which was to be proved.

□

Theorem 5 defines an algorithm for evaluating $L_0(x, y)$. The complexity of the algorithm corresponds to the complexity of evaluating $\mu\gamma(x, y)$, which is obviously $\mathbf{O}(m)$ for $m \leq n$, plus the complexity of raising 2 to power $|\mu\gamma(x, y)|$, which is also $\mathbf{O}(m)$. Thus, the complexity of the algorithm is $\mathbf{O}(m)$.

5.2. Sum of μ-Lengths of Common L-Embeddings

Theorem 6. $L_1(x, y) = L_1(x[m-1], y[n-1]))$ if $x_m \neq y_n$,

$L_1(x, y) = 2L_1(x[m-1], y[n-1] + L_0(x[m-1], y[n-1])$ if $x_m = y_n$.

Proof.

We denote $L = L(x) \cap L(y)$ and $L_{-1} = L(x[m-1]) \cap L(y[n-1])$. If $x_m \neq y_n$, then L-embedding $u \in L_{-1}$ corresponds to one L-embedding $u\varepsilon \in L$ of the same μ-length. Hence, $L_1(x, y) = \Sigma\ \{|\mu(u)| : u \in L\} = \Sigma\ \{|\mu(u)| : u \in L_{-1}\} = L_1(x[m-1], y[n-1]))$. If $x_m = y_n$, then L-embedding $u \in L_{-1}$ corresponds to two L-embeddings: $u\varepsilon \in L$ of the same μ-length and $ux_m \in L$ with a μ-length greater by 1. Hence, $L_1(x, y) = \Sigma\ \{|\mu(u)| : u \in L\} = \Sigma\ \{|\mu(u)| : u \in L_{-1}\} + (\Sigma\ \{|\mu(u)| : u \in L_{-1}\} + L_0(x[m-1], y[n-1])) = 2\Sigma\ \{|\mu(u)| : u \in L_{-1}\} + L_0(x[m-1], y[n-1]) = 2L_1(x[m-1], y[n-1] + L_0(x[m-1], y[n-1])$.

□

Theorem 6 defines an algorithm for evaluating $L_1(x, y)$; its complexity obviously does not exceed by an order of magnitude the complexity of the algorithm for evaluating $L_0(x, y)$, i.e., it is $\mathbf{O}(m)$.

Theorem 7. $L_1(x, y) = 1{*}C_l^1 + 2{*}C_l^2 + \ldots + l{*}C_l^1$ (A001787), where $l = |\mu\gamma(x, y)|$.

Proof.

The statement of the theorem follows directly from the fact that the number of common L-embeddings of μ-length i is $C^i_{|\mu\gamma(x,y)|}$.

□

Theorem 7 defines an algorithm for evaluating $L_1(x, y)$. The complexity of the algorithm is equal to the complexity of evaluating $\mu\gamma(x, y)$, which is obviously $\mathbf{O}(m)$ for $m \leq n$, plus the complexity of calculating factorials $i!$ for $i \in 0..l$, which is $\mathbf{O}(m)$, plus the complexity of adding l numbers, which is $\mathbf{O}(m)$. Thus, the complexity of the algorithm is $\mathbf{O}(m)$.

5.3. Sum of the Minimum Numbers and Sum of Products of the Numbers of E-Embeddings in Common A-Embeddings

Theorem 8. $L_2(x, y) = L_3(x, y) = L_0(x, y)$.

Proof.

The statement of the theorem follows directly from the fact that L-embedding $u \in L(x)$ has exactly one E-embedding v such that $\lambda(v) = u$: $L_2(x, y) = \Sigma\{min\{|l(u, x)|, |l(u, y)|\} : u \in L(x) \cap L(y)\} = \Sigma\{min\{1,$

$1\} : u \in L(x) \cap L(y)\} = |L(x) \cap L(y)| = L_0(x, y)$; $L_3(x, y) = |L(x, y)| = |\cup \{l(u, x) \times l(u, y) : u \in L(x) \cap L(y)\}| = |L(x) \cap L(y)| = L_0(x, y)$.

□

Theorem 8 defines an algorithm for evaluating $L_2(x, y)$ and $L_3(x, y)$ that has the same complexity as the algorithm for evaluating $L_0(x, y)$, i.e., $\mathbf{O}(m)$ for $m \le n$.

5.4. Similarity Function Based on the Longest Common L-Embedding

Theorem 9.

$lcL(x, y) = lcL(x[m - 1], y[n - 1]) + 1$ if $x_m = y_n$;

$lcL(x, y) = lcL(x[m - 1], y[n - 1])$ if $x_m \ne y_n$.

Proof. Function $\gamma(x, y)$ establishes a positional correspondence between coinciding symbols of x and y when counting positions from right to left. Hence, $lcL(x, y) = |\mu\gamma(x, y)|$, which directly implies the statement of the theorem.

□

Theorem 9 defines an algorithm for evaluating function $L_4(x, y) = lcL(x, y)$ that has complexity $\mathbf{O}(m)$ for $m \le n$.

6. *R*-EMBEDDINGS

Like *L*-embedding, *R*-embedding $u \in R(x)$ has only one *E*-embedding v such that $\rho(v) = u$. Common *R*-embedding $u \in R(x) \cap R(y)$ has a pair of *E*-embeddings in x and y that differ only in the postfix of empty symbols. That is why set $r(u, x)$ is a singleton, $\{r_l(u, x)\} = \{r_r(u, x)\} = r(u, x)$ and $R(x, y) = R_l(x, y) = R_r(x, y)$.

By $\delta(x, y)$, we denote a sequence z of length m (recall that $|x| = m \le n = |y|$) that coincides with x and y in those positions (from left to right) in which x and y coincide, as well as has empty symbols in the other positions: $|z| = m$ and $\forall\, i \in 1..m\ (x_i = y_i \Rightarrow z_i = x_i)\ \&\ (x_i \ne y_i \Rightarrow z_i = \varepsilon)$. Function $\delta(x, y)$ establishes a positional correspondence between coinciding symbols of x and y when counting positions from left to right.

6.1. Number of Common R-Embeddings

Theorem 10. $R_0(x, y) = R_0(x[2..m], y[2..n]))$ if $x_1 \ne y_1$,

$R_0(x, y) = 2R_0(x[2..m], y[2..n])$ if $x_1 = y_1$.

Proof.

We denote $R = R(x) \cap R(y)$ and $R_{-1} = R(x[2..m]) \cap R(y[2..n])$. If $x_1 \ne y_1$, then *R*-embedding $u \in R_{-1}$ corresponds to one *R*-embedding $\varepsilon u \in R$. Hence, $R_0(x, y) = |R| = |R_{-1}| = R_0(x[2..m], y[2..n]))$. If $x_1 = y_1$, then *R*-embedding $u \in R_{-1}$ corresponds to two *R*-embeddings $\varepsilon u \in R$ and $x_1 u$. Hence, $R_0(x, y) = |R| = 2|R_{-1}| = 2R_0(x[2..m], y[2..n]))$.

□

Theorem 10 defines an algorithm for evaluating $R_0(x, y)$. For $m \le n$, the number of algorithm steps is $\mathbf{O}(m)$, which is determined by the number of functions of form $R_0(x[i..m], y[i..n])$, where $i \in 0..m$. At each step, the computations have complexity $\mathbf{O}(1)$. Thus, the complexity of the algorithm is $\mathbf{O}(m)$.

Theorem 11. $R_0(x, y) = 2^{|\mu\delta(x, y)|}$.

Proof.

Theorem 10 implies that, when adding one symbol to both sequences on the left, the number of common *R*-embeddings doubles if the same symbols are added; otherwise, it does not change. With $|R(x) \cap R(())| = |R(()) \cap R(y)| = |\varepsilon| = 1$ and $|\mu\delta(x, y)|$ being equal to the number of coinciding symbols in the same positions of x and y from left to right, we have $R_0(x, y) = |R(x) \cap R(y)| = 2^{|\mu\delta(x, y)|}$, which was to be proved.

□

Theorem 11 defines an algorithm for evaluating $R_0(x, y)$. The complexity of the algorithm corresponds to the complexity of evaluating $\mu\delta(x, y)$, which is obviously $\mathbf{O}(m)$ for $m \le n$, plus the complexity of raising 2 to power $|\mu\delta(x, y)|$, which is also $\mathbf{O}(m)$. Thus, the complexity of the algorithm is $\mathbf{O}(m)$.

6.2. Sum of μ-Lengths of Common R-Embeddings

Theorem 12. $R_1(x, y) = R_1(x[2..m], y[2..n]))$ if $x_1 \ne y_1$,

$R_1(x, y) = 2R_1(x[2..m], y[2..n] + R_0(x[2..m], y[2..n])$ if $x_1 = y_1$.

Proof.

We denote $R = R(x) \cap R(y)$ and $R_{-1} = R(x[2..m]) \cap R(y[2..n])$. If $x_1 \ne y_1$, then *R*-embedding $u \in R_{-1}$ corresponds to one *R*-embedding $\varepsilon u \in R$ of the same μ-length. Hence, $R_1(x, y) = \Sigma\{|\mu(u)| : u \in R\} = \Sigma\{|\mu(u)| : u \in R_{-1}\} = R_1(x[2..m], y[2..n]))$. If $x_1 = y_1$, then *R*-embedding $u \in R_{-1}$ corresponds to two *R*-embeddings: $\varepsilon u \in R$ of the same μ-length and $x_1 u \in R$ with a μ-length greater by 1. Hence, $R_1(x, y) = \Sigma\{|\mu(u)| : u \in R\} = \Sigma\{|\mu(u)| : u \in R_{-1}\} + (\Sigma\{|\mu(u)| : u \in R_{-1}\} + R_0(x[2..m], y[2..n])) = 2\Sigma\{|\mu(u)| : u \in R_{-1}\} + R_0(x[2..m], y[2..n]) = 2R_1(x[2..m], y[2..n] + R_0(x[2..m], y[2..n])$.

□

Theorem 12 defines an algorithm for evaluating $R_1(x, y)$; its complexity obviously does not exceed by an order of magnitude the complexity of the algorithm for evaluating $R_0(x, y)$; i.e., it is $\mathbf{O}(m)$ for $m \le n$.

Theorem 13. $R_1(x, y) = 1*C_r^1 + 2*C_r^2 + \ldots + r*C_r^r$ (A001787), where $r = |\mu\delta(x, y)|$.

Proof.

The statement of the theorem follows directly from the fact that the number of common *R*-embeddings of μ-length i is $C^i_{|\mu\delta(x,y)|}$.

□

Theorem 13 defines an algorithm for evaluating $R_1(x, y)$. The complexity of the algorithm is equal to the complexity of evaluating $\mu\delta(x, y)$, which is obviously $\mathbf{O}(m)$ for $m \le n$, plus the complexity of calculating factorials $i!$ for $i \in 0..r$, which is $\mathbf{O}(m)$, plus the complexity of adding r numbers, which is $\mathbf{O}(m)$. Thus, the complexity of the algorithm is $\mathbf{O}(m)$.

6.3. Sum of the Minimum Numbers and Sum of Products of the Numbers of E-Embeddings in Common R-Embeddings

Theorem 14. $R_2(x, y) = R_3(x, y) = R_0(x, y)$.

Proof.

The statement of the theorem follows directly from the fact that R-embedding $u \in R(x)$ has exactly one E-embedding v such that $\rho(v) = u$: $R_2(x, y) = \Sigma\{min\{|r(u, x)|, |r(u, y)|\} : u \in R(x) \cap R(y)\} = \Sigma\{min\{1, 1\} : u \in R(x) \cap R(y)\} = |R(x) \cap R(y)| = R_0(x, y)$; $R_3(x, y) = |R(x, y)| = |\cup \{r(u, x) \times r(u, y) : u \in R(x) \cap R(y)\}| = |R(x) \cap R(y)| = R_0(x, y)$.

□

Theorem 14 defines an algorithm for evaluating $R_2(x, y)$ and $R_3(x, y)$ that has the same complexity as the algorithm for evaluating $R_0(x, y)$, i.e., $\mathbf{O}(m)$ for $m \le n$.

6.4. Similarity Function Based on the Longest Common R-Embedding

Theorem 15.

$lcR(x, y) = lcR(x[2..m], y[2..n]) + 1$ if $x_1 = y_1$;

$lcR(x, y) = lcR(x[2..m], y[2..n])$ if $x_1 \ne y_1$.

Proof. Function $\delta(x, y)$ establishes a positional correspondence between coinciding symbols of x and y when counting positions from left to right. Hence, $lcR(x, y) = |\mu\delta(x, y)|$, which directly implies the statement of the theorem.

□

Theorem 15 defines an algorithm for evaluating function $R_4(x, y) = lcR(x, y)$ that has complexity $\mathbf{O}(m)$ for $m \le n$.

7. *O*-EMBEDDINGS

7.1. Number of Common O-Embeddings

For $x_m = y_n = h$, we denote the sets of indices that determine the position of symbol h in x and y by $I = \{i \in 1..m : x_i = h\}$ and $J = \{j \in 1..n : y_j = h\}$, respectively.

For $i \in I$ and $j \in J$, we denote $L_{i,j} = L(x[i-1]) \cap L(y[j-1])$.

Denote also $K = (I \times J)\backslash\{(m, n)\}$ and $L_h(x, y) = L_{m,n}\backslash\cup \{L_{i,j} : (i, j) \in K\}$.

For $i \in I$ and $j \in J$, $u_{i,j}$ denotes the maximum common L-embedding in $x[i-1]$ and $y[j-1]$: $u_{i,j} = \lambda(v_{i,j})$, where $v_{i,j}$ is a common E-embedding in $x[i-1]$ and $y[j-1]$, which is defined by the following condition: $\forall\, t = 1..min\{i, j\} - 1\ (x_{i-t} = y_{i-t} \Rightarrow v_{i,j}(i-t) = x_{i-t})$ & $(x_{i-t} \ne y_{i-t} \Rightarrow v_{i,j}(i-t) = \varepsilon)$. The set of all common L-embeddings in $x[i-1]$ and $y[j-1]$ is $\lambda E(u_{i,j})$; i.e., it is obtained from $u_{i,j}$ by all possible replacements of certain symbols with empty symbols, followed by the removal of the prefix of empty symbols.

Lemma 1. Suppose that $x_m = y_n = h$. Evaluation of $|L_h(x, y)|$ is equivalent to calculating the number of terms in the PDNF of the conjunction of disjunctions of variables without negations, where the number of variables is equal to the number of non-empty symbols in the maximum common L-embedding $u_{m,n}$ in $x[m-1]$ and $y[n-1]$.

Proof.

$L_h(x, y) = L_{m,n}\backslash\cup\{L_{i,j} : (i, j) \in K\} = L_{m,n}\backslash\cup\{L_{m,n} \cap L_{i,j} : (i, j) \in K\}$. To obtain $L_h(x, y)$, we need to remove set $L_{m,n} \cap L_{i,j}$ from $L_{m,n}$ for each $(i, j) \in K$.

Let us define the intersection $u^{\wedge}_{i,j}$ of embeddings $u_{m,n}$ and $u_{i,j}$ that have length $|u_{m,n}|$ and coincide with $u_{m,n}$ and $u_{i,j}$ in those positions (from right to left) in which they mutually coincide, with the other positions containing empty symbols: $|u^{\wedge}_{i,j}| = |u_{m,n}|$ and $\forall\, t = 1..|u_{m,n}|$ $(t > |u_{i,j}| \Rightarrow u^{\wedge}_{i,j}(|u_{m,n}| - t) = \varepsilon)$ & $(t \le |u_{i,j}|$ & $u_{m,n}(|u_{m,n}| - t) = u_{i,j}(|u_{i,j}| - t) \Rightarrow u^{\wedge}_{i,j}(|u_{m,n}| - t) = u_{m,n}(|u_{m,n}| - t))$ & $(t \le |u_{i,j}|$ & $u_{m,n}(|u_{m,n}| - t) \ne u_{i,j}(|u_{i,j}| - t) \Rightarrow u^{\wedge}_{i,j}(|u_{m,n}| - t) = \varepsilon)$. Obviously, $u^{\wedge}_{m,n} = u_{m,n}$.

The symbols in the positions where $u_{m,n}$ contains empty symbols are removed from $u^{\wedge}_{i,j}$, and the result is denoted by $w_{i,j}$: if $u^{\wedge}_{i,j} = u_1h_1u_2h_2...u_kh_{k-1}u_k$, $u_{m,n} = v_1\varepsilon v_2\varepsilon...v_{k-1}\varepsilon v_k$ for both $i \in 1..k$ $|u_i| = |v_i|$ and $v_i \in H^*$, then $w_{i,j} = u_1u_2...u_{k-1}u_k$. Obviously, $w_{m,n} = \mu(u_{m,n})$.

All $w_{i,j}$ have the same length $|w_{m,n}|$, equal to the number of non-empty symbols in $u_{m,n}$. Each L-embedding from $L_{m,n} \cap L_{i,j}$ is in one-to-one correspondence with an E-embedding in $w_{i,j}$, with the number of these embeddings being 2^k, where $k = |\mu(w_{i,j})|$ is the number of non-empty symbols in $w_{i,j}$.

Each $t \in 1..|w_{m,n}|$ is associated with a Boolean variable α_t, which means that there can be a non-empty symbol in position t. Then, E-embeddings in $w_{i,j}$ are given by a Boolean function $F_{i,j} = \&\{\neg\alpha_t : t \in 1..|w_{m,n}|$ & $w_{i,j}(t) = \varepsilon\}$, which is **true** on only those sets α_1, ..., in which $\alpha_t =$ **false** for all indices t by which an empty symbol is found in $w_{i,j}$ and, therefore, in any E-embedding in $w_{i,j}$. If there is a non-empty symbol by index t in $w_{i,j}$, then some E-embeddings in $w_{i,j}$ contain an empty symbol in this position, while the others contain non-empty symbols, which means that function

$F_{i,j}$ does not depend on α_t. Obviously, $F_{m,n} = \&\varnothing =$ **true**.

Difference $L_{m,n}\backslash\cup\{L_{m,n} \cap L_{i,j} : (i,j)\in K\}$ is given by a Boolean function $F = F_{m,n}\backslash\vee\{F_{i,j} : (i,j)\in K\} = \&\{\neg F_{i,j} : (i,j)\in K\}$, and $\neg F_{i,j} = \vee\{\alpha_t : t \in 1..|w_{m,n}| \,\&\, w_{i,j}(t) = \varepsilon\}$. Thus, F is a conjunction of disjunctions of variables without negations, and $|L_h(x,y)|$ is equal to the number of terms in the PDNF of function F.

□

Theorem 16.

$O_0(x, y) = O_0(x[m-1], y) + O_0(x, y[n-1]) - O_0(x[m-1], y[n-1])$ if $x_m \neq y_n$.

$O_0(x, y) = O_0(x[m-1], y) + O_0(x, y[n-1]) - O_0(x[m-1], y[n-1]) + L_h(x, y])$ if $x_m = y_n$.

Proof.

Let us denote the following sets of pairs of E-embeddings:

$E = O_r(x, y)$,

$E_{00} = O_r(x[m-1], y[n-1])(\varepsilon, \varepsilon)$ is the pair of last elements $(\varepsilon, \varepsilon)$,

$E_{01} = (O_r(x[m-1], y])(\varepsilon, ()))\backslash E_{00}$ is the pair of last elements (ε, y_n),

$E_{10} = (O_r(x, y[n\text{-}1])((), \varepsilon))\backslash E_{00}$ is the pair of last elements (x_m, ε),

$E_{11} = E\backslash(E_{00} \cup E_{01} \cup E_{10})$ is the pair of last elements (x_m, y_n), since $E_{00} \cup E_{01} \cup E_{10}$ contain all pairs of E-embeddings of common O-embeddings in which the pair of last elements contains ε.

Obviously, $E = E_{00} \cup E_{01} \cup E_{10} \cup E_{11}$. Pairs of last elements of pairs of E-embeddings from different sets E_{00}, E_{01}, E_{10}, and E_{11} are different, which is why these sets are pairwise disjoint. Therefore, $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}|$, $|E_{00} \cup E_{01}| = |E_{00}| + |E_{01}|$, and $|E_{00} \cup E_{10}| = |E_{00}| + |E_{10}|$.

Since $O_0(x, y) = |O(x) \cap O(y)| = |O_r(x, y)|$ and $L_0(x, y) = L_3(x, y) = |L(x, y)|$ for any x and y, the statement of the theorem can be rewritten as follows:

$|E| = (|E_{00}| + |E_{01}|) + (|E_{00}| + |E_{10}|) - |E_{00}| = |E_{00}| + |E_{01}| + |E_{10}|$ if $x_m \neq y_n$,

$|E| = (|E_{00}| + |E_{01}|) + (|E_{00}| + |E_{10}|) - |E_{00}| + |L_h(x, y)| = |E_{00}| + |E_{01}| + |E_{10}| + |L_h(x, y)|$ if $x_m = y_n$.

Let us consider the case where $x_m \neq y_n$. In this case, $E_{11} = \varnothing$; this implies that $|E_{11}| = 0$ and $|E_{00}| = |E_{00}| + |E_{01}| + |E_{10}|$, which was to be proved in this case.

Let us consider the case where $x_m = y_n = h$. In this case, each pair of E-embeddings from E_{11} has form $(\varepsilon^{m-|v|-k-1}v\varepsilon^k h,\ \varepsilon^{n-|v|-k-1}v\varepsilon^k h)$, where $(\varepsilon^{m-|v|-k-1}v\varepsilon^k,\ \varepsilon^{n-|v|-k-1}v\varepsilon^k) \in L_h(x, y)$. More specifically, when passing from $x[m-1]$ and $y[n-1]$ to x and y, this pair is formed by adding symbol h to the right of the pair of E-embeddings of common L-embeddings in $x[m-1]$ and $y[n-1]$ that did not exist before, i.e., they are not pairs of E-embeddings of common L-embeddings in $x[i-1]$ and $y[j-1]$, where $x_i = y_j = h$ and $i < m$ or $j < n$. We say that pair $(\varepsilon^{m-|v|-k-1}v\varepsilon^k h,\ \varepsilon^{n-|v|-k-1}v\varepsilon^k h)$ corresponds to pair $(\varepsilon^{m-|v|-k-1}v\varepsilon^k,\ \varepsilon^{n-|v|-k-1}v\varepsilon^k)$. This correspondence is obviously a bijection of sets E_{11} and $L_h(x, y)$. Therefore, $|E_{11}| = |L_h(x, y)|$. Hence, $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}| = |E_{00}| + |E_{01}| + |E_{10}| + |L_h(x, y)|$, which was to be proved.

□

Theorem 16 defines an algorithm for evaluating $O_0(x, y)$. The number of algorithm steps is $\mathbf{O}(mn)$, which is determined by the number of functions of form $O_0(x[m-i], y[n-j])$, where $i \in 0..m$ and $j \in 0..n$, provided that each function is evaluated at most once (after which its value is saved). At each step, the computations have complexity $\mathbf{O}(1)$. Thus, the complexity of the algorithm is $\mathbf{O}(mn)^*\mathbf{O}(L_h(x, y))$, where $\mathbf{O}(L_h(x, y))$ is the complexity of evaluating function $|L_h(x, y)|$.

Lemma 2. Suppose that we have k not necessarily distinct sets $a(s)$, $s \in 1..k$. For $S \subseteq 1..k$, $S \neq \varnothing$, the intersection of sets $a(s)$ whose indices run over set S is denoted by $c(S) = \cap\{a(s) : s \in S\}$, and the sum of the numbers of subsets in set $c(S)$ over all S for which $|S| = l$ is denoted by $E(l) = \Sigma\{2^{|c(S)|} : S \subseteq 1..k \,\&\, |S| = l\}$. Then, the number of distinct sets embedded in at least one of the sets $a(s)$, $s \in 1..k$, is equal to the alternating sum $E(1) - E(2) + E(3) - E(4) \ldots (-1)^{k+1}E(k)$, and the sum F of sizes of these sets incremented by 1 is equal to the alternating sum $F(1) - F(2) + F(3) - F(4) \ldots (-1)^{k+1}F(k)$, where $F(l) = \Sigma\{(|c(S)| + 2)\, 2^{|c(S)|-1} : S \subseteq 1..k \,\&\, |S| = l\}$. These alternating sums can be calculated in $\mathbf{O}(2^k)$ time, provided that intersections of two sets (as well as integer arithmetic operations) can be performed in $\mathbf{O}(1)$ time.

Proof.

Let us denote the number of distinct sets embedded in at least one of sets $a(s)$, $s \in 1..k$, by $E(a(1), \ldots, a(k))$, and the sum of their sizes, incremented by 1, by $F(a(1), \ldots, a(k))$.

We prove the theorem by induction on k. For $k = 1$, there is a single set $S \subseteq 1..1$, $S \neq \varnothing$, namely, $S = \{1\}$, and $c(\{1\}) = \cap\{a(s) : s \in \{1\}\} = a(1)$. The number of distinct subsets of set $a(1)$ is $2^{|a(1)|}$ and $E(a(1)) = E(1) = 2^{|a(1)|}$, while the sum of their lengths, incremented by 1, is $1^*C_r^0 + 2^*C_r^1 + \ldots + r^*C_r^{-1} + (r+1)^*C_r^r = (r+2)2^{r-1}$ (A001792), and $F(a(1)) = F(1) = (r+2)2^{r-1}$, where $r = |a(1)|$.

Suppose that the statement holds for k. Let us prove it for $k+1$.

The number of distinct sets embedded in at least one of the sets $a(s)$, where $s \in 1..k+1$, is equal to the number of distinct sets embedded in at least one of the sets $a(s)$, where $s \in 1..k$, i.e., $E(a(1), \ldots, a(k))$, plus the number of distinct sets embedded in $a(k+1)$, i.e., $2^{|a(k+1)|}$, except those embedded in the intersection between $a(k+1)$

and the union of sets $a(1)$, ..., $a(k)$, the number of which is $E(a(k+1) \cap a(1), ..., a(k+1) \cap a(k))$.

Let us denote $E_{k+1} = E(a(1), ..., a(k+1))$, $E_k = E(a(1), ..., a(k))$, and $E_k^\wedge = E(a(k+1) \cap a(1), ..., a(k+1) \cap a(k))$. We have $E_{k+1} = E_k + 2^{|a(k+1)|} - E_k^\wedge$.

Let us now consider $S \subseteq 1..k+1$. Suppose that $|S| = 1$; then, $S = \{i\}$, $i \in 1..k+1$. For $i \in 1..k$, term $2^{|c(S)|} = 2^{|a(i)|}$ occurs once in sum E_k with sign "+"; as a result, for $i \in 1..k+1$, term $2^{|c(S)|} = 2^{|a(i)|}$ occurs once in sum E_{k+1} with the same sign. Suppose that $|S| > 1$. If $k+1 \notin S$, then term $2^{|c(S)|}$ occurs once in sum E_k with sign "$(-1)^{|S|+1}$"; as a result, for $i \in 1..k+1$, term $2^{|c(S)|}$ occurs once in sum E_{k+1} with the same sign. If $k+1 \in S$, then $c(S) = \{a(i_1) \cap ... \cap a(i_{|S|-1}) \cap a(k+1)\} = \{(a(k+1) \cap a(i_1)) \cap ... \cap (a(k+1) \cap a(i_{|S|-1}))\}$. Therefore, term $2^{|c(S)|}$ occurs once in sum $E_k^\wedge$ with sign "$(-1)^{|S|}$"; however, with sum $E_k^\wedge$ being subtracted, term $2^{|c(S)|}$ occurs once in sum E_{k+1} with sign "$(-1)^{|S|+1}$".

Similarly, the sum of lengths (incremented by 1) of distinct sets embedded in at least one of sets $a(s)$, where $s \in 1..k+1$, is equal to the sum of lengths (incremented by 1) of distinct sets embedded in at least one of sets $a(s)$, where $s \in 1..k$, i.e., $F(a(1), ..., a(k))$, plus the sum of lengths (incremented by 1) of distinct sets embedded in $a(k+1)$, i.e., $(|a(k+1)| + 2)2^{|a(k+1)|-1}$, except those embedded in the intersection between $a(k+1)$ and the union of sets $a(1)$, ..., $a(k)$, the sum of lengths (incremented by 1) of which is $F(a(k+1) \cap a(1), ..., a(k+1) \cap a(k))$. We denote $F_{k+1} = F(a(1), ..., a(k+1))$, $F_k = F(a(1), ..., a(k))$, and $F_k^\wedge = F(a(k+1) \cap a(1), ..., a(k+1) \cap a(k))$. We have $F_{k+1} = F_k + (|a(k+1)| + 2)2^{|a(k+1)|-1} - F_k^\wedge$. Let us consider $S \subseteq 1..k+1$. Suppose that $|S| = 1$; then, $S = \{i\}$, $i \in 1..k+1$. For $i \in 1..k$, term $(|c(S)| + 2)2^{|c(S)|-1} = (|a(i)| + 2)2^{|a(i)|-1}$ occurs once in sum F_k with sign "+"; as a result, for $i \in 1..k+1$, term $(|c(S)| + 2)2^{|c(S)|-1} = (|a(i)| + 2)2^{|a(i)|-1}$ occurs once in sum F_{k+1} with the same sign. Suppose that $|S| > 1$. If $k+1 \notin S$, then term $(|c(S)| + 2)2^{|c(S)|-1}$ occurs once in sum F_k with sign "$(-1)^{|S|+1}$"; as a result, for $i \in 1..k+1$, term $(|c(S)| + 2)2^{|c(S)|-1}$ occurs once in sum F_{k+1} with the same sign. If $k+1 \in S$, then term $(|c(S)| + 2)2^{|c(S)|-1}$ occurs once in sum $F_k^\wedge$ with sign "$(-1)^{|S|}$"; however, with $F_k^\wedge$ being subtracted, term $(|c(S)| + 2)2^{|c(S)|-1}$ occurs once in sum F_{k+1} with sign "$(-1)^{|S|+1}$".

The statement is proved.

The number of (not necessarily distinct) sets $c(S)$ is equal to the number of non-empty subsets S of set $1..k$, which is equal to $2^k - 1$. That is why the complexity of evaluating E is $\mathbf{O}(2^k)$, provided that intersections of two sets (as well as integer arithmetic operations) can be performed in $\mathbf{O}(1)$ time.

□

Let us denote the set of indices of sequence x by which symbol h is found: $K_x(h) = \{i \in 1..m : x(i) = h\}$. For $i \in K_x(h)$, the set of pairs (symbol in x, symbol's position in x relative to position i), except for pair $(h, 0)$, is denoted by $P_x(i) = \{(x(t), t-i) : t \in 1..m \,\&\, t \neq i\}$. Suppose that symbol h occurs $|K_x(h)| > 0$ times in x and $|K_y(h)| > 0$ times in y. We denote $k = |K_x(h)|*|K_y(h)|$. For $i \in K_x(h)$ and $j \in K_y(h)$, for brevity, we denote $P(i,j) = P_x(i) \cap P_y(j)$. For $S \subseteq K_x(h) \times K_y(h)$, $S \neq \varnothing$, we denote $c(S) = \cap\{P(i,j) : (i,j) \in S\}$.

Theorem 17. The number of distinct common O-embeddings in x and y that contain symbol h is equal to the alternating sum $E = E(1) - E(2) + E(3) - E(4) ... (-1)^{k+1}E(k)$, where term $E(l)$ is a sum of the numbers of all subsets of set $c(S)$ over all S of size l, $|S| = l$: $E(l) = \Sigma\{2^{|c(S)|} : S \subseteq 1..k \,\&\, |S| = l\}$. The complexity of the computations is $\mathbf{O}(n2^k)$.

Proof.

Let us consider common O-embedding u and a pair of its E-embeddings $v(x) \in o(u, x)$ and $v(y) \in o(u, y)$ such that $v(x)_i = x_i = h$ and $v(y)_j = y_j = h$. It is in one-to-one correspondence with a subset of the set of pairs $P(i,j)$. We need to calculate the number of distinct sets embedded in at least one of sets $P(i,j)$, where $i \in K_x(h)$ and $j \in K_y(h)$. The number of these sets of pairs $P(i,j)$ is k. By Lemma 2, it is equal to $E = E(1) - E(2) + E(3) - E(4) ... (-1)^{k+1}E(k)$, where $E(l) = \Sigma\{2^{|c(S)|} : S \subseteq 1..k \,\&\, |S| = l\}$, $l = 1..k$, is the sum of the numbers of all subsets of set $c(S)$ over all $S \subseteq K_x(h) \times K_y(h)$ of size l, i.e., $|S| = l$, and $c(S) = \cap\{P(i,j) : (i,j) \in S\}$.

We iterate over sequence x of length m and calculate $K_x(h)$. While searching through $K_x(h)$, we calculate sets $P_x(i)$. Calculating set $P_x(i)$ requires searching through sequence x of length m. Thus, all sets $P_x(i)$, $i \in K_x(h)$, are calculated in $\mathbf{O}(m|K_x(h)|)$. Similarly, all sets $P_y(j)$, $j \in K_y(h)$, are calculated in $\mathbf{O}(n|K_x(h)|)$. We can assume that set $P_x(i) = \{(x(t), t-i) : t \in 1..m \,\&\, t \neq i\}$ is linearly arranged in ascending order with respect to index $t - i$; the size of this set is $m - 1$. Similarly for set $P_y(j)$, with the size of this set being $n - 1$. Then, to construct intersection $P(i,j) = P_x(i) \cap P_y(j)$, we need to iterate over these sets, i.e., we need time $\mathbf{O}(n + m)$, with all sets $P(i,j)$, $i \in K_x(h)$ and $j \in K_y(h)$, being calculated in $\mathbf{O}(k(n+m))$. Similarly, each intersection of sets $P(i,j)$ is constructed in $\mathbf{O}(n+m)$ time. The complexity of calculating sum E by Lemma 2 is $\mathbf{O}(2^k)$; however, provided that the intersection of two sets is constructed in $\mathbf{O}(1)$ time, in this case, $\mathbf{O}((n+m)2^k)$ time is required. The overall complexity is $\mathbf{O}(m) + \mathbf{O}(n) + \mathbf{O}(m|K_x(h)|) + \mathbf{O}(n|K_y(h)|) + \mathbf{O}(k(n+m)) + \mathbf{O}((n+m)2^k) = \mathbf{O}((n+m)2^k)$, which corresponds to $\mathbf{O}(n2^k)$ for $m \le n$.

□

Theorem 17 defines the following algorithm for evaluating $O_0(x, y)$.

1. SUMMA = 0.

2. Search through sequence x to find non-empty symbol h in x.

2.1. If it is not found, then the algorithm terminates.

2.2. If it is found, then search for symbol h in y.

2.2.1. If it is not found, then replace h in x with an empty symbol and go to Step 2.

2.2.2. If it is found, then do the following.

2.2.2.1. Calculate set $K_x(h)$ of indices in x by which h is found, as well as set $K_y(h)$ of indices in y by which h is found.

2.2.2.2. For each pair $(i, j) \in K_x(h) \times K_y(h)$, construct a set of pairs $P(i, j)$.

2.2.2.3. For each set of pairs of indices $S \subseteq K_x(h) \times K_y(h)$, $S \neq \varnothing$, construct intersection $c(S) = \cap \{P(i, j) : (i, j) \in S\}$.

2.2.2.4. Calculate $E = E(1) - E(2) + E(3) - E(4) \ldots (-1)^{k+1}E(k)$.

2.2.2.5. SUMMA = SUMMA + E.

2.2.2.6. Replace h in x and y with empty symbols and go to Step 2.

7.2. Sum of μ*-Lengths of Common O-Embeddings*

Theorem 18. The sum of μ-lengths of distinct common O-embeddings in x and y that contain symbol h is equal to the alternating sum $F = F(1) - F(2) + F(3) - F(4) \ldots (-1)^{k+1}F(k)$, where $F(l) = \Sigma\{(|c(S)| + 2)\, 2^{|c(S)|-1} : S \subseteq 1..k \,\&\, |S| = l\}$ is the sum of cardinalities of all subsets of all sets $c(S)$ for $|S| = l$. The complexity of the computations is $\mathbf{O}(n2^k)$.

Proof is similar to that of Theorem 17. □

Theorem 18 defines the following algorithm for evaluating $O_1(x, y)$.

1. Search through sequence x to find non-empty symbol h in x.

1.1. If it is not found, then the algorithm terminates.

1.2. If it is found, then search for symbol h in y.

1.2.1. If it is not found, then replace h in x with an empty symbol and go to Step 2.

1.2.2. If it is found, then do the following.

1.2.2.1. Calculate set $K_x(h)$ of indices in x by which h is found, as well as set $K_y(h)$ of indices in y by which h is found.

1.2.2.2. For each pair $(i, j) \in K_x(h) \times K_y(h)$, construct a set of pairs $P(i, j)$.

1.2.2.3. For each set of pairs of indices $S \subseteq K_x(h) \times K_y(h)$, $S \neq \varnothing$, construct intersection $c(S) = \cap \{P(i, j) : (i, j) \in S\}$.

1.2.2.4. Calculate $F = F(1) - F(2) + F(3) - F(4) \ldots (-1)^{k+1}F(k)$.

1.2.2.5. SUMMA = SUMMA + F.

1.2.2.6. Replace h in x and y with empty symbols and go to Step 2.

7.3. Sum of the Minimum Numbers of E-Embeddings in Common O-Embeddings

Theorem 19. The sum of the minimum numbers of E-embeddings of common O-embeddings in x and y that contain symbol h is $\Sigma\{min\{|I|, |J|\} * 2^{|A(I, J)|-1} : I \subseteq K_x(h) \,\&\, I \neq \varnothing \,\&\, J \subseteq K_y(h) \,\&\, J \neq \varnothing\}$, where $A(I, J) = (\cap\{P(i, j) : i \in I, j \in J\})\backslash(\cup\{P(i, j): i \in K_x(h)\backslash I \vee j \in K_y(h)\backslash J\})$.

The complexity of the computations is $\mathbf{O}(n2^k)$.

Proof. Set $A(I, J)$ represents all common O-embeddings that contain symbol h and have E-embeddings in x, represented by set I and E-embeddings in y, represented by set J. For each of these O-embeddings, the minimum number of their E-embeddings in x and y is $min\{|I|, |J|\}$. The number of these O-embeddings is $2^{|A(I, J)|}$, which is why the sum of the minimum numbers of E-embeddings in these common O-embeddings is $min\{|I|, |J|\} * 2^{|A(I, J)|}$. By summing over all pairs of sets $I \subseteq K_x(h)$, $I \neq \varnothing$, and $J \subseteq K_y(h)$, $J \neq \varnothing$, we obtain the desired sum of the minimum numbers of E-embeddings of common O-embeddings in x and y that contain symbol h.

When calculating this sum, the operations of addition, multiplication, powering 2, and calculating the minimum of two numbers are performed $\mathbf{O}(2^k)$ times. To evaluate all $A(I, J)$, the operations for calculating the difference of two sets, intersecting two sets, and uniting two sets are performed $\mathbf{O}(2^k)$ times; each of these operations is performed in $\mathbf{O}(n + m)$ time.

We search through sequence x of length m and calculate $K_x(h)$. While searching through $K_y(h)$, we calculate sets $P_x(i)$. Calculating set $P_x(i)$ requires searching through sequence x of length m. Thus, all sets $P_x(i)$, $i \in K_x(h)$, are calculated in $\mathbf{O}(m|K_x(h)|)$. Similarly, all sets $P_y(j)$, $j \in K_y(h)$, are calculated in $\mathbf{O}(n|K_y(h)|)$. We can assume that set $P_x(i) = \{(x(t), t - i) : t \in 1..m \,\&\, t \neq i\}$ is linearly arranged in the ascending order with respect to index $t - i$; the size of this set is $m - 1$. Similarly for set $P_y(j)$, with the size of this set being $n - 1$. Then, to construct intersection $P(i, j) = P_x(i) \cap P_y(j)$, we need to search through these sets, i.e., we need time $\mathbf{O}(n + m)$, with all sets $P(i, j)$, $i \in K_x(h)$ and $j \in K_y(h)$, being calculated in $\mathbf{O}(k(n + m))$ time. To calculate set $A(I, J)$, the operations for calculating the difference of two sets, intersecting two sets, and uniting two sets are performed in $\mathbf{O}(n + m)$ time, and the number of these sets $A(I, J)$ is $\mathbf{O}(2^k)$. Thus, all sets $A(I, J)$ are constructed in $\mathbf{O}((n + m)2^k)$ time, after which the arithmetic operations are performed $\mathbf{O}(2^k)$ times to calculate the sum. The overall complexity is $\mathbf{O}(k(n + m)) + \mathbf{O}((n + m)2^k)$

$+ \mathbf{O}(2^k) = \mathbf{O}((n+m)2^k)$, which corresponds to $\mathbf{O}(n2^k)$ for $m \le n$.

□

Theorem 19 defines the following algorithm for evaluating $O_2(x, y)$.

1. Search through sequence x to find non-empty character h in x.

1.1. If it is not found, then the algorithm terminates.

1.2. If it is found, then search for symbol h in y.

1.2.1. If it is not found, then replace h in x with an empty symbol and go to Step 2.

1.2.2. If it is found, then do the following.

1.2.2.1. Calculate set $K_x(h)$ of indices in x by which h is found, as well as set $K_y(h)$ of indices in y by which h is found.

1.2.2.2. For each pair $(i, j) \in K_x(h) \times K_y(h)$, construct a set of pairs $P(i, j)$.

1.2.2.3. For each $I \subseteq K_x(h)$, $I \neq \varnothing$, and $J \subseteq K_y(h)$, $J \neq \varnothing$, construct $A(I, J)$.

1.2.2.4. Calculate $M = \Sigma\{min\{|I|, |J|\} * 2^{|A(I,J)|-1} : I \subseteq K_x(h)\ \&\ I \neq \varnothing\ \&\ J \subseteq K_y(h)\ \&\ J \neq \varnothing\}$.

1.2.2.5. SUMMA = SUMMA + M.

1.2.2.6. Replace h in x and y with empty symbols and go to Step 2.

7.4. Sum of Products of the Numbers of E-Embeddings in Common O-Embeddings

Theorem 20. $O_3(x, y) = O_3(x[m-1], y) + O_3(x, y[n-1]) - O_3(x[m-1], y[n-1])$ if $x_m \neq y_n$;

$O_3(x, y) = O_3(x[m-1], y) + O_3(x, y[n-1]) - O_3(x[m-1], y[n-1]) + L_3(x[m-1], y[n-1])$ if $x_m = y_n$.

Proof. Let us denote the following sets of pairs of E-embeddings:

$E = O(x, y)$,

$L_{00} = L(x[m-1], y[n-1])$,

$E_{00} = O(x[m-1], y[n-1])(\varepsilon, \varepsilon)$ is the pair of last elements $(\varepsilon, \varepsilon)$,

$E_{01} = (O(x[m-1], y])(\varepsilon, ()))\backslash E_{00}$ is the pair of last elements (ε, y_n),

$E_{10} = (O(x, y[n-1])((), \varepsilon))\backslash E_{00}$ is the pair of last elements (x_m, ε),

$E_{11} = E\backslash(E_{00} \cup E_{01} \cup E_{10})$ is the pair of last elements (x_m, y_n), since $E_{00} \cup E_{01} \cup E_{10}$ contain all pairs of E-embeddings of common O-embeddings in which the pair of last elements contains ε.

Obviously, $E = E_{00} \cup E_{01} \cup E_{10} \cup E_{11}$. Pairs of last elements of pairs of E-embeddings from different sets E_{00}, E_{01}, E_{10}, and E_{11} are different, which is why these sets are pairwise disjoint. Therefore, $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}|$, $|E_{00} \cup E_{01}| = |E_{00}| + |E_{01}|$, and $|E_{00} \cup E_{10}| = |E_{00}| + |E_{10}|$.

In this notation, the statement of the theorem has the following form:

$|E| = (|E_{00}| + |E_{01}|) + (|E_{00}| + |E_{10}|) - |E_{00}| = |E_{00}| + |E_{01}| + |E_{10}|$ if $x_m \neq y_n$,

$|E| = (|E_{00}| + |E_{01}|) + (|E_{00}| + |E_{10}|) - |E_{00}| + |L_{00}| = |E_{00}| + |E_{01}| + |E_{10}| + |L_{00}|$ if $x_m = y_n$.

Let us consider the case where $x_m \neq y_n$. In this case, $E_{11} = \varnothing$; this implies that $|E_{11}| = 0$ and $|E_{00}| = |E_{00}| + |E_{01}| + |E_{10}|$, which was to be proved in this case.

Let us consider the case where $x_m = y_n = h$. Each pair of E-embeddings from E_{11} has form $(\varepsilon^{m-|v|-k-1}v\varepsilon^k h, \varepsilon^{n-|v|-k-1}v\varepsilon^k h)$, where $(\varepsilon^{m-|v|-k-1}v\varepsilon^k, \varepsilon^{n-|v|-k-1}v\varepsilon^k) \in L_{00}$. We say that pair $(\varepsilon^{m-|v|-k-1}v\varepsilon^k h, \varepsilon^{n-|v|-k-1}v\varepsilon^k h)$ corresponds to pair $(\varepsilon^{m-|v|-k-1}v\varepsilon^k, \varepsilon^{n-|v|-k-1}v\varepsilon^k)$. This correspondence is obviously a bijection of sets E_{11} and L_{00}. Therefore, $|E_{11}| = |L_{00}|$. Hence, $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}| = |E_{00}| + |E_{01}| + |E_{10}| + |L_{00}|$, which was to be proved.

□

Theorem 20 defines an algorithm for evaluating $O_3(x, y)$. The number of algorithm steps is $\mathbf{O}(mn)$, which is determined by the number of functions of form $O_3(x[m-i], y[n-j])$ and $L_3(x[m-i], y[n-i])$, where $i \in 0..m$ and $j \in 0..n$, provided that each function is evaluated at most once (after which its value is saved). At each step, the computations have complexity $\mathbf{O}(1)$. Thus, the complexity of the algorithm is $\mathbf{O}(mn)$.

7.5. Similarity Function Based on the Longest Common O-Embedding

Theorem 21.

$lcO(x, y) = max\{lcL(x[m-1], y[n-1]) + 1, lcO(x[m-1], y[n-1])\}$ if $x_m = y_n$:

$lcO(x, y) = max\{lcO(x, y[n-1]), lcO(x[m-1], y)\}$ if $x_m \neq y_n$.

Proof. If $x_m = y_n$, then the right-most E-embeddings of a common O-embedding can have either (1) symbol $x_m = y_n$ or (2) an empty symbol in position m in x and in position n in y. In case 1, the longest common O-embedding has form ux_m, where u is a common L-embedding of prefixes of $x[m-1]$ and $y[n-1]$; i.e., its μ-length exceeds by 1 that of the longest common L-embedding of prefixes of $x[m-1]$ and $y[n-1]$. In case 2, the longest common O-embedding is the longest O-embedding of prefixes of $x[m-1]$ and $y[n-1]$ that has the same μ-length. This implies the statement of the theorem for $x_m = y_n$.

For $x_m \neq y_n$, the right-most E-embeddings of a common O-embedding have an empty symbol either (1) in position m in x or (2) in position n in y. In case 1, the longest common O-embedding is the longest O-embedding of prefixes of $x[m-1]$ and sequence y

that has the same μ-length. In case 2, the longest common O-embedding is the longest O-embedding of sequence x and prefix of $y[n-1]$ that has the same μ-length. This implies the statement of the theorem for $x_m \neq y_n$.

□

Theorem 21 defines an algorithm for evaluating function $O_4(x, y) = lcO(x, y)$, which has complexity $\mathbf{O}(mn)$.

8. CONCLUSIONS

Some of the similarity functions introduced in this paper play an auxiliary role. For instance, L_3 is used to evaluate O_3, lcL is used to evaluate lcO, while A_0 is independently important and is also used to evaluate A_1. However, there are cases where these auxiliary functions play the main role. L-embeddings are useful when the distance between embedded symbols is important, as well as when the distance from the last embedded symbol to the end of a sequence is taken into account; R-embeddings are useful when the distance from the beginning of the sequence to the first embedded symbol is important.

When comparing functions of different types (regardless of the type of embedding), we can note the following. The number of common embeddings (function 0) is a good numerical characteristic; however, it does not take into account the length of embeddings. For instance, sequences 11112222 and 1122111 have 9 common subsequences (including the empty one), and the sum of their lengths is 17. The former sequence also has 9 common subsequences with sequence 22221111; however, the sum of their lengths is 20 because there are two long subsequences 1111 and 2222. Therefore, the sum of lengths of common embeddings (function 1) is of independent importance.

Both these characteristics (functions 0 and 1) do not take into account that one common embedding can occur in one sequence many times and occur in another sequence a few times. This is taken into account by the sum of the numbers of pairs of occurrences of common embeddings (the sum of products of the numbers of occurrences of common embeddings: function 3). Using the above example, sequences 11112222 and 1122111 have 279 pairs of occurrences of common embeddings, and sequences 11112222 and 22221111 have 139 pairs due to the fact that, in the former pair of sequences, there are many occurrences of common embeddings 12, 112, and 122 in both the sequences that are absent in the latter pair.

At the same time, function 3 does not satisfy the natural *axiom of direction of similarity* [5], also known as the *bound by self-similarity* [4]: $f(x, y) \leq min\{f(x, x), f(y, y)\}$. In particular, this function is strictly increasing when the same non-empty sequence x is compared with sequences xx, xxx, $xxxx$, ... This flaw is overcome by function 4 (the sum of the minimum numbers of occurrences of common embeddings). Unfortunately, this function is poorly amenable to algorithmic optimization, in particular, for A-embeddings, i.e., subsequences; we do not aware of any algorithm for it, except exhaustive search. Partial optimization could be carried out using an efficient algorithm for enumerating common subsequences the complexity $C(x, y)$ of which is lower than that of exhaustive search. Since the calculation of the number of occurrences of embedding u in sequence x has complexity $\mathbf{O}(|u|*|x|)$ (see Lemma 8 [3]), we would have an algorithm for evaluating function 4 that has complexity $C(x, y)*\mathbf{O}(mn)$. Moreover, if there were an efficient (possibly on a subclass of sequences) algorithm for enumerating subsequences of only one sequence that has complexity $C_1(x, y)$, then we would have an algorithm for evaluating function 4 that has complexity $C_1(x, y)*\mathbf{O}(mn)$.

Function 5, which is based on the longest common embedding, also satisfies the direction axiom. However, it has the following disadvantage: it does not take into account common embeddings that are not part of the longest common embedding. In our example, sequences 11112222 and 1122111 have the longest common subsequence 1122, which does not include subsequence 111; sequences 11112222 and 22221111 have two longest common subsequences: 1111 (which does not include all subsequences of twos) and 2222 (which does not include all subsequences of ones).

The problem of enumerating common embeddings is also investigated. As an example, we can mention the work [6], which proposed an algorithm for enumerating maximum (rather than longest) common subsequences, which needs polynomial time and memory for each maximum common subsequence. These maximum common embeddings have a useful property: each common embedding is part of one or several maximum embeddings. It could be possible to propose a similarity function based on maximum common embeddings: the number of these embeddings, the sum of their lengths, etc. This could be a subject of further research.

Another direction for further research could be similarity functions for sequences in an alphabet with weighted symbols. An empty symbol could be assigned a zero weight. In turn, an embedding could correspond to the sum of weights of its constituent symbols rather than to their number. A negative weight could be a "penalty" for using this symbol in a common embedding.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Wagner, R. and Fischer, M., The string-to-string correction problem, *J. ACM,* 1974, vol. 21, no. 1, pp. 168–173. https://dl.acm.org/doi/10.1145/321796.321811
2. Wang, H., All common subsequences, *Proc. 20th Int. Joint Conf. Artificial Intelligence (IJCAI),* Hyderabad, India, 2007. https://www.aaai.org/Papers/IJCAI/2007/IJCAI07-101.pdf.
3. Elzinga, C., Rahmann, S., and Wang, H., Algorithms for subsequence combinatorics, *Theor. Comput. Sci.,* 2008, vol. 409, no. 3, pp. 394–404. https://doi.org/10.1016/j.tcs.2008.08.035
4. *Proc. Int. Conf. Sequence Analysis and Related Methods (LaCOSAII),* Lausanne, Switzerland, 2016. https://www.academia.edu/83294569/Proceedings_of_the_International_Conference_on_Sequence_Analysis_and_Related_Methods_LaCOSA_II_Lausanne_Switzerland_June_8_10_2016.
5. Znamenskii, S.V., Model and axioms of similarity metrics, *Program. Sist.: Teor. Prilozh.,* 2017, vol. 8, no. 4, pp. 347–357. https://doi.org/10.25209/2079-3316-2017-8-4-347-357
6. Conte, A., Grossi, R., Punzi, G., et al., Enumeration of maximal common subsequences between two strings, *Algorithmica,* 2022, vol. 84, pp. 757–783. https://doi.org/10.1007/s00453-021-00898-5

Translated by Yu. Kornienko