ПРОГРАММИРОВАНИЕ, 2023, № 5, с. 3–18

ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ

УДК 519.6

ДВАДЦАТЬ ФУНКЦИЙ ПОДОБИЯ ДВУХ КОНЕЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

© 2023 г. И. Бурдонов[a],*, А. Максимов[a],**

[a] *Институт системного программирования РАН им. В.П. Иванникова*
109004, г. Москва, ул. А. Солженицына, д. 25, Россия

**E-mal: igor@ispras.ru*

***E-mal: andrew@ispras.ru*

Поступила в редакцию 09.01.2023 г.
После доработки 16.02.2023 г.
Принята к публикации 21.03.2023 г.

В статье рассматриваются различные числовые функции, определяющие степень “похожести” двух заданных конечных последовательностей. Эти меры подобия основаны на определяемом нами понятии вложения в последовательность. Частным случаем такого вложения является обычная подпоследовательность (subsequence). Другие случаи дополнительно требуют равенства расстояний между соседними символами подпоследовательности в обеих последовательностях. Это является обобщением понятия отрезка последовательности (substring), в котором эти расстояния единичны. Дополнительно может требоваться равенство расстояний от начала последовательностей до первого символа вложения или от последнего символа вложения до конца последовательностей. Кроме этих двух последних случаев, вложение может входить в последовательность несколько раз. В литературе используются такие функции как число общих вложений или числа пар вхождений вложений в последовательности. Кроме них, мы вводим еще три функции: сумма длин общих вложений, сумма минимумов числа вхождений общего вложения в обе последовательности и функция подобия на основе наибольшего по числу символов общего вложения. Всего рассматриваются 20 числовых функций, для 17 из которых предложены алгоритмы (в том числе новые) полиномиальной сложности, еще для двух функций алгоритмы имеют экспоненциальную сложность с уменьшенным показателем степени. В Заключении дается краткая сравнительная характеристика этих вложений и функций.

Ключевые слова: анализ последовательностей, общие подпоследовательности, наибольшие и максимальные общие подпоследовательности, каноническое вложение, соответствующие совместные вложения, комбинаторные алгоритмы для подпоследовательностей и вложений, аксиомы подобия

DOI: 10.31857/S0132347423050035, **EDN:** ZYAIZN

1. ВВЕДЕНИЕ

Анализ последовательностей широко используется в социальных, управленческих, политических, демографических, психологических науках, в химии, биоинформатике и при обработке текстов. Последние десятилетия появилось много работ, посвященных этой тематике [1–6]. Применяются различные метрики и меры сходства (подобия) последовательностей [4, 5].

В этой статье мы рассматриваем различные числовые функции, определяющие степень “похожести” двух заданных конечных последовательностей. Эти меры подобия основаны на определяемом нами понятии вложения в последовательность. Частным случаем такого вложения является подпоследовательность. Другие случаи учитывают дополнительно расстояния между символами подпоследовательности в обеих последовательностях. Например, последовательности “МЕТРИКА” и “МАРОККО” имеют наибольшую общую подпоследовательность “МРК”, с учетом расстояний между символами “Р-К”, с учетом расстояний между символами и от последнего символа вложения до конца последовательности “К-”, с учетом расстояний между символами и от начала последовательности до первого символа вложения “М----К”.

Понятие вложения вводится с использованием пустого символа, не принадлежащего алфавиту заданных последовательностей. Всего вводятся пять типов вложения: *E*-вложение получается заменой некоторых символов пустым символом, *L*-вложение получается из *E*-вложения удалением пустого префикса, *R*-вложение получается из *E*-вложения удалением пустого постфикса, *O*-вложение получается из *E*-вложения удалением пу-

стых префикса и постфикса, наконец, *A*-вложение получается из *E*-вложения удалением всех пустых символов, что совпадает с понятием подпоследовательности. Мнемоника обозначения вложений образована от английских слов: *E* – **E**mpty symbol (пустой символ), далее указание места, где пустых символов нет (there are no empty symbols): *L* – on the **L**eft (слева), *R* – on the **R**ight (справа), *O* – **O**utside (снаружи, т.е. слева и справа), *A* – **A**nywhere (в любом месте, т.е. нигде нет пустых символов).

Каждое вложение может иметь несколько вхождений в последовательность, т.е. несколько *E*-вложений, из которых данное вложение получается удалением префикса, постфикса, префикса и постфикса или всех пустых символов. Например, подпоследовательность “МРК” входит один раз в последовательность “МЕТРИКА” (соответствующее *E*-вложение “М--Р-К-”) и два раза в последовательность “МАРОККО” (соответствующие *E*-вложения “М-Р-К--” и “М-Р--К-”. Для вложений, которые могут содержать пустые символы, определяется понятие μ-длины как число непустых символов (для *A*-вложения совпадает с длиной подпоследовательности).

Для каждого из четырех типов вложения (*L*, *R*, *O* и *A*) определяются пять функций: 0) число общих вложений, 1) сумма μ-длин общих вложений, 2) сумма минимумов чисел вхождения общих вложений в заданные последовательности, 3) сумма произведений чисел вхождения общих вложений в заданные последовательности, а также 4) мера похожести, основанная на наибольшей (по длине) общей подпоследовательности (*longest common subsequence*, *lcs*).

Некоторые из этих двадцати функций хорошо известны, например, число общих подпоследовательностей (число общих *A*-вложений) или сумма произведений чисел вхождения общих подпоследовательностей в заданные последовательности (сумма произведений чисел *E*-вложений общих *A*-вложений в заданные последовательности) [3]. Другие функции, особенно учитывающие “расстояния” между символами, мы вводим в данной статье.

После настоящего Введения в разделе 2 вводятся основные понятия и обозначения. В разделе 3 рассматривается оптимизация общая для всех типов вложений: замена пустым символом тех символов, которые входят только в одну из двух последовательностей. Далее в разделах 4–7 рассматриваются четыре типа вложений *L*, *R*, *O* и *A*, и для каждого типа вложения – алгоритмы вычисления пяти функций 0, 1, 2, 3, 4. В Заключении подводятся итоги и намечаются направления дальнейших исследований.

2. ОПРЕДЕЛЕНИЯ И ОБОЗНАЧЕНИЯ

Для целых чисел i и j будем обозначать: $i..j = \{i, i+1, ..., j\}$, если $i \le j$, $i..j = \varnothing$, если $i > j$.

Конечная последовательность в алфавите H длиной $m \ge 0$ – это инъекция множества $1..m$ в множество H: $1..m \to H$. Множество конечных последовательностей в алфавите H обозначим H^*. Пустую последовательность (длины 0, пустая инъекция) обозначим (). Для непустой конечной последовательности x i-й элемент последовательности, $i \in 1..|x|$, обозначим $x_i = x(i)$. Отрезок $x_i, x_{i+1}, ..., x_j$ для $1 \le i \le j \le |x|$ обозначим $x[i..j]$. Для $i > j$ определим $x[i..j] = ()$. Префикс обозначим как $x[j] = x[1..j]$, пустой префикс (длины 0) определен, в том числе, и для пустой последовательности $()[0] = ()$. Конечная последовательность из $k \ge 0$ повторений символа $h \in H$ будем обозначать h^k: $|h^k| = k \;\&\; \forall\, i = 1..|k| h^k_i = h$. Также вместо h^1 будем писать просто h.

Конкатенация xy конечных последовательностей x и y определяется условиями: $|xy| = |x| + |y|$, $\forall\, i \in 1..|x|\ (xy)_i = x_i$ и $\forall\, j \in 1..|y|\ (xy)_{|x|+j} = y_j$. Нам также понадобится конкатенация пары (X, Y) конечных множеств конечных последовательностей с парой (z, t) конечных последовательностей, которую определим так: $(X, Y)(z, t) = \{(xz, yt) : (x, y) \in (X, Y)\}$. Будем считать, что в выражениях операция конкатенации приоритетнее операций над множествами (объединение, пересечение и разность).

Композицию функций f и g будем обозначать fg.

Введем *пустой символ* $\varepsilon \notin H$ и обозначим $H_\varepsilon = H \cup \{\varepsilon\}$.

Обозначим множество конечных последовательностей в алфавите H_ε:

- не начинающихся пустым символом $L(H) = \{v \in H_\varepsilon^* : |v| > 0 \Rightarrow v_1 \ne \varepsilon\}$;
- не заканчивающихся пустым символом $R(H) = \{v \in H_\varepsilon^* : |v| > 0 \Rightarrow v_{|v|} \ne \varepsilon\}$;
- не начинающихся и не заканчивающихся пустым символом

$$O(H) = \{v \in H_\varepsilon^* : |v| > 0 \Rightarrow v_1 \ne \varepsilon \;\&\; v_{|v|} \ne \varepsilon\} = L(H) \cap R(H).$$

Введем операции удаления пустых символов из конечной последовательности в алфавите H_ε:

- *удаление префикса пустых символов* λ: $H_\varepsilon^* \to L(H)$ определяется условием:

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

- *удаление постфикса пустых символов* ρ: $H_\varepsilon^* \to R(H)$ определяется условием:

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

• *удаление всех пустых символов* μ: $H_\varepsilon^* \to H^*$ определяется условием:

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

Определим для последовательности $x \in H_\varepsilon^*$:

• *E-вложение* получается из x заменой некоторых символов пустыми символами;

• *L-вложение* получается из *E*-вложения удалением префикса пустых символов;

• *R-вложение* получается из *E*-вложения удалением постфикса пустых символов;

• *O-вложение* получается из *E*-вложения удалением префикса и постфикса пустых символов, или из *L*-вложения удалением постфикса пустых символов, или из *R*-вложения удалением префикса пустых символов;

• *A-вложение* получается из *E*-, *L*-, *R*- или *O*-вложения удалением пустых символов.

Для $x \in H^*$ понятие *A*-вложения совпадает с понятием подпоследовательности, а *E*-вложение v соответствует понятию вхождения подпоследовательности $\mu(v)$ в x.

Обозначим множества вложений в последовательность $x \in H^*$:

• множество *E*-вложений в x:

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

• множество *L*-вложений в x:

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

• множество *R*-вложений в x:

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

• множество *O*-вложений в x:

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

• множество *A*-вложений в x:

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

Для последовательности x в алфавите H_ε^* введем понятие μ-*длины* x как число непустых символов в x, очевидно, равное $|\mu(x)|$. Для $x \in H^*$ μ-длина совпадает с длиной последовательности.

Обозначим для последовательностей $x \in H^*$:

• множество *E*-вложений в x *L*-вложения $u \in L(x)$: $l(u, x) = \{v \in E(x) : \lambda(v) = u\}$;

• множество *E*-вложений в x *R*-вложения $u \in R(x)$: $r(u, x) = \{v \in E(x) : \rho(v) = u\}$;

• множество *E*-вложений в x *O*-вложения $u \in O(x)$: $o(u, x) = \{v \in E(x) : \lambda\rho(v) = u\}$;

• множество *E*-вложений в x *A*-вложения $u \in A(x)$: $a(u, x) = \{v \in E(x) : \mu(v) = u\}$;

Обозначим для последовательностей $x \in H^*$ и $y \in H^*$ множества пар *E*-вложений:

• множество пар *E*-вложений общих *L*-вложений

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

• множество пар *E*-вложений общих *R*-вложений

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

• множество пар *E*-вложений общих *O*-вложений

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

• множество пар *E*-вложений общих *A*-вложений

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

Обозначим для последовательностей $x \in H^*$ и $y \in H^*$ наибольшую μ-длину общего вложения:

• для *L*-вложений: $lcL(x, y) = max\{\mu(u) : u \in L(x) \cap L(y)\}$;

• для *R*-вложений: $lcR(x, y) = max\{\mu(u) : u \in R(x) \cap R(y)\}$;

• для *O*-вложений: $lcO(x, y) = max\{\mu(u) : u \in O(x) \cap O(y)\}$;

• для *A*-вложений: $lcA(x, y) = max\{|u| : u \in A(x) \cap A(y)\}$;

Наибольшую μ-длину общего вложения естественно рассматривать как еще одну функцию “похожести” последовательностей x и y: $lcI(x, y)$ для $I \in \{L, R, O, A\}$. По этому критерию последовательность x больше всего “похожа” на саму себя, поскольку является наибольшим *I*-вложением в себя: $lcI(x, x) = max\{\mu(u) : u \in I(x)\} = |x| \geq max\{\mu(u) : u \in I(x) \cap I(y)\} = lcI(x, y)$.

Нас будут интересовать функции от последовательностей $x \in H^*$ и $y \in H^*$, задаваемые табл. 1.

Заметим, что для $I \in \{L, R, O, A\}$ и $j \in 0..4$ $I_j(x, y) = I_j(y, x)$. Если $j \neq 1$, то $I_j(x, ()) = I_j((), y) = 1$; $I_1(x, ()) = I_1((), y) = 0$.

Обозначим для непустой последовательности $x \in H^*$:

• самое левое *E*-вложение в x *L*-вложения $u \in L(x)$: $l_l(u, x) = v$, если $v \in l(u, x)$ и $\forall w \in l(u, x)\backslash\{v\}$ $w(min\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• самое правое *E*-вложение в x *L*-вложения $u \in O(x)$: $l_r(u, x) = v$, если $v \in l(u, x)$ и $\forall w \in l(u, x)\backslash\{v\}$ $w(max\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$.

• самое левое *E*-вложение в x *R*-вложения $u \in R(x)$: $r_l(u, x) = v$, если $v \in r(u, x)$ и $\forall w \in r(u, x)\backslash\{v\}$ $w(min\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• самое правое *E*-вложение в x *R*-вложения $u \in O(x)$: $r_r(u, x) = v$, если $v \in r(u, x)$ и $\forall w \in r(u, x)\backslash\{v\}$ $w(max\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$.

Таблица 1. Функции для общих вложений двух последовательностей x и y

Функция\ Вложение	Число общих вложений	Сумма μ-длин общих вложений	Сумма минимумов чисел E-вложений общих вложений	Сумма произведений чисел E-вложений общих вложений	Наибольшая μ-длина общего вложения
	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	$A_0(x, y) = \|A(x) \cap A(y)\|$	$A_1(x, y) = \Sigma\{\|u\| : u \in A(x) \cap A(y)\}$	$A_2(x, y) = \Sigma\{min\{\|a(u, x)\|, \|a(u, y)\|\} : u \in A(x) \cap A(y)\}$	$A_3(x, y) = \|A(x, y)\|$	$A_4(x, y) = lcA(x, y)$

• самое левое E-вложение в x O-вложения $u \in O(x)$: $o_l(u, x) = v$, если $v \in o(u, x)$ и $\forall\, w \in o(u, x)\backslash\{v\}$ $w(min\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• самое правое E-вложение в x O-вложения $u \in O(x)$: $o_r(u, x) = v$, если $v \in o(u, x)$ и $\forall\, w \in o(u, x)\backslash\{v\}$ $w(max\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$.

• самое левое E-вложение в x A-вложения $u \in A(x)$: $a_l(u, x) = v$, если $v \in a(u, x)$ и $\forall\, w \in a(u, x)\backslash\{v\}$ $w(min\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$;

• самое правое E-вложение в x A-вложения $u \in A(x)$: $a_r(u, x) = v$, если $v \in a(u, x)$ и $\forall\, w \in a(u, x)\backslash\{v\}$ $w(max\{i \in 1..|x| : w_i \neq v_i\}) = \varepsilon$.

В литературе самые левые (*left-most*) E-вложения общих вложений называют также (для подпоследовательностей) каноническими (*canonical*) [3].

Обозначим для непустых последовательностей $x \in H^*$ и $y \in H^*$ множества пар E-вложений:

• множество пар самых левых E-вложений общих L-вложений

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

• множество пар самых правых E-вложений общих L-вложений

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

• множество пар самых левых E-вложений общих R-вложений

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

• множество пар самых правых E-вложений общих R-вложений

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

• множество пар самых левых E-вложений общих O-вложений

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

• множество пар самых правых E-вложений общих O-вложений

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

• множество пар самых левых E-вложений общих A-вложений

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

• множество пар самых правых E-вложений общих A-вложений

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

Заметим, что $|O_l(x, y)| = |O_r(x, y)| = |O(x) \cap O(y)|$ и $|A_l(x, y)| = |A_r(x, y)| = |A(x) \cap A(y)|$.

Далее через x и y будем обозначать две непустые последовательности в алфавите H с длинами $m = |x|$, $n = |y|$. Будем считать, что $m \leq n$.

3. ЗАМЕНА НЕОБЩИХ СИМВОЛОВ ПУСТЫМ СИМВОЛОМ

Общей оптимизацией для вычисления всех функций для всех вложений является замена необщих символов пустым символом: для $i \in 1..m$ определим $x_i^\wedge = x_i$, если $x_i \in \mathbf{Im}\, y$, $x_i^\wedge = \varepsilon_i$, если $x_i \notin \mathbf{Im}\, y$. Аналогично определяется $y^\wedge$. Эта замена выполняется за время $\mathbf{O}(mn)$. В случае A-вложений вместо замены необщего символа пустым символом

можно просто удалять необщий символ. Описанные ниже алгоритмы можно применять после выполнения этой замены (удаления для A-вложений).

4. A-ВЛОЖЕНИЯ (ПОДПОСЛЕДОВАТЕЛЬНОСТИ)

4.1. Число общих A-вложений

Здесь мы докажем теорему, аналогичную лемме 6 в [3], но дадим свое доказательство, поскольку мы используем другие определения и обозначения.

Для непустой последовательности $z \in H^*$ и символа $h \in H$ обозначим максимальный индекс, по которому в последовательности z находится символ h, или 0, если h не входит в z:

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

Обозначим: $k = p(x[m-1], x_m\}$, $l = p(y, x_m\}$.

Теорема 1.

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

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

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

Доказательство.

Рассматриваем множества пар $(e_r(u, x), e_r(u, y))$ самых правых E-вложений общих A-вложений u последовательностей x и y. Обозначим пары последовательностей длины m и n:

$E = A_r(x, y)$;

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

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

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

Поскольку для любых x', y' имеет место $A_0(x', y') = |A(x') \cap A(y')| = |A_r(x', y')|$, утверждение теоремы можно переписать в виде:

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

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

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

Поскольку E-вложения из множества E_m имеют вид $(...\varepsilon, ...)$, а множества E_{ml} и E_{kl} либо пусты, либо E-вложения из них имеют вид $(...x_m, ...)$, имеем $E_m \cap E_{ml} = E_m \cap E_{kl} = \varnothing$. Поскольку $m - 1 \geq k - 1$, имеем $E_{ml} \supseteq E_{kl}$.

Рассмотрим случай $x_m \notin \mathbf{Im}\, y$. Поскольку $x_m \notin \mathbf{Im}\, y$, переход от последовательности $x[m-1]$ к последовательности $x = x[m-1]x_m$ не добавляет новых общих A-вложений. Поэтому $E = E_m$. Отсюда $|E| = |E_m|$, что и требовалось доказать в этом случае.

Рассмотрим случай $x_m \in \mathbf{Im}\, y$ и $x_m \notin \mathbf{Im}\, x[m-1]$. Поскольку $x_m \in \mathbf{Im}\, y$, переход от последовательности $x[m-1]$ к последовательности $x = x[m-1]x_m$ может добавлять новые общие A-вложения, эти общие A-вложения должны заканчиваться на x_m, но этих общих A-вложений раньше, в $x[m-1]$ и y, не было, поскольку $x_m \notin \mathbf{Im}\, x[m-1]$. Заметим, что для такого нового общего A-вложения u в его самых правых E-вложениях в x и y последний непустой символ равен x_m по индексам m и l, соответственно, т.е. $e_r(u, x)_m = e_r(u, y)_l = x_m$. Имеем $E = E_m \cup E_{ml}$. Поскольку $E_m \cap E_{ml} = \varnothing$, имеем $|E| = |E_m| + |E_{ml}|$, что и требовалось доказать в этом случае.

Теперь рассмотрим случай $x_m \in \mathbf{Im}y$ и $x_m \in \mathbf{Im}\, x[m-1]$. Поскольку $x_m \in \mathbf{Im}y$, переход от последовательности $x[m-1]$ к последовательности $x = x[m-1]x_m$ может добавлять новые общие A-вложения u, но они новые только в том случае, когда таких A-вложений не было раньше в $x[m-1]$ и y. Эти новые общие A-вложения должны заканчиваться на x_m. Заметим, что если такое общее A-вложение u было раньше, то в его самых правых E-вложениях в $x[m-1]$ и y последний непустой символ равен x_m по индексам k и l, соответственно, т.е. $e_r(u, x[m-1])_k = e_r(u, y)_l = x_m$. В любом случае в самых правых E-вложениях u в x и y последний непустой символ равен x_m по индексам m и l, соответственно, т.е. $e_r(u, x)_m = e_r(u, y)_l = x_m$. Имеем $E = E_m \cup (E_{ml} \backslash E_{kl})$. Поскольку $E_m \cap E_{ml} = E_m \cap E_{kl} = \varnothing$, имеем $E_m \cap (E_{ml} \backslash E_{kl}) = \varnothing$ и поэтому $|E| = |E_m| + |E_{ml} \backslash E_{kl}|$. Поскольку $E_{ml} \supseteq E_{kl}$, имеем $|E_{ml} \backslash E_{kl}| = |E_{ml}| - |E_{kl}|$. Поэтому $|E| = |E_m| + |E_{ml}| - |E_{kl}|$, что и требовалось доказать в этом случае.

□

Теорема 1 определяет алгоритм вычисления $A_0(x, y)$. Число шагов алгоритма равно $\mathbf{O}(mn)$, что определяется числом функций вида $A_0(x[m-i], y[n-j])$, где $i \in 0..m$ и $j \in 0..n$, при условии, что каждая функция вычисляется не более одного раза (после чего ее значение сохраняется). На каждом шаге проверка условий $x_{m-i} \in \mathbf{Im}\, y[n-j]$ и $x_{m-i} \in \mathbf{Im}\, x[m-i-1]$ имеет сложность $\mathbf{O}(m+n)$, а остальные вычисления имеют сложность $\mathbf{O}(1)$. Сложность алгоритма равна $\mathbf{O}(m^2n + mn^2)$, что для $m \leq n$ равно $\mathbf{O}(mn^2)$.

4.2. Сумма длин общих A-вложений

Теорема 2.

A1(x, y) = A1(x[m − 1], y), если xm ∉ Im y;

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

$A_1(x, y) = A_1(x[m-1], y) + A_1(x[m-1], y[l-1]) + A_0(x[m-1], y[l-1]) - A_1(x[k-1], y[l-1]) -$

$A_0(x[k-1], y[l-1])$, если $x_m \in \mathbf{Im}\, y$ и $x_m \in \mathbf{Im}\, x[m-1]$.

Доказательство.

Доказательство аналогично доказательству теоремы 1. Используем те же обозначения:

$E = A_r(x, y)$;

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

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

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

Имеем $E_m \cap E_{ml} = E_m \cap E_{kl} = \varnothing$ и $E_{ml} \supseteq E_{kl}$.

Также имеем $A_0(x[m-1], y[l-1]) = |E_{ml}|$ и $|A_0(x[k-1], y[l-1])| = |E_{kl}|$.

Обозначим:

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

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

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

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

Длина общего A-вложения u равна числу непустых символов в каждом его E-вложении, в том числе, в самом правом E-вложении. Поэтому в этих обозначениях утверждение теоремы можно переписать в виде:

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

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

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

При переходе от последовательности $x[m-1]$ к последовательности $x = x[m-1]x_m$ число непустых символов в самом правом E-вложении в x общего A-вложения u не меняется, если в нем по индексу m оказывается пустой символ $e_r(u, x)_m = \varepsilon$, и увеличивается на 1 в противном случае, когда $e_r(u, x)_m = x_m$.

Это влечет следующие утверждения.

В случае $x_m \notin \mathbf{Im}\, y$ имеем $E = E_m$, для каждого $(u, v) \in E_m$ имеет место $u_m = \varepsilon$, поэтому $S = S_m$, что и требовалось доказать в этом случае.

В случае $x_m \in \mathbf{Im}\, y$ и $x_m \notin \mathbf{Im}\, x[m-1]$ имеем $E = E_m \cup E_{ml}$ и $E_m \cap E_{ml} = \varnothing$, для каждого $(u, v) \in E_m$ имеет место $u_m = \varepsilon$, для каждого $(u, v) \in E_{ml}$ имеет место $u_m = x_m$, поэтому $S = S_m + (S_{ml} + |E_{ml}|)$, что и требовалось доказать в этом случае.

В случае $x_m \in \mathbf{Im}\, y$ и $x_m \in \mathbf{Im}\, x[m-1]$ имеем $E = E_m \cup (E_{ml} \backslash E_{kl})$ и $E_m \cap E_{ml} = E_m \cap E_{kl} = \varnothing$, для каждого $(u, v) \in E_m$ имеет место $u_m = \varepsilon$, для каждого $(u, v) \in E_{ml}$ имеет место $u_m = x_m$, для каждого $(u, v) \in E_{kl}$ имеет место $u_m = x_m$, поэтому $S = S_m + (S_{ml} + |E_{ml}|) - (S_{kl} + |E_{kl}|)$, что и требовалось доказать в этом случае.

□

Теорема 2 определяет алгоритм вычисления $A_1(x, y)$, сложность которого по порядку, очевидно, не превышает сложности алгоритма вычисления $A_0(x, y)$, т.е. равна $\mathbf{O}(m^2n + mn^2)$, что для $m \le n$ равно $\mathbf{O}(mn^2)$.

4.3. Сумма минимумов чисел E-вложений общих A-вложений

Для функции $A_2(x, y)$ мы не знаем хорошего (отличного от полного перебора) алгоритма. Единственная полезная оптимизация – это предварительное удаление необщих символов.

4.4. Сумма произведений чисел E-вложений общих A-вложений

Здесь мы докажем теорему, аналогичную теореме 2 в [3] с тем отличием, что мы учитываем пустую подпоследовательность, а в теореме 2 в [3] она не учитывается. Также мы дадим свое доказательство, поскольку мы используем другие определения и обозначения.

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

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

Доказательство.

Обозначим следующие множества пар E-вложений общих A-вложений:

$E = A(x, y)$,

$E_{00} = A(x[m-1], y[n-1])(\varepsilon, \varepsilon)$ – пара последних элементов $(\varepsilon, \varepsilon)$,

$E_{01} = (A(x[m-1], y])(\varepsilon, ()))\backslash E_{00}$ – пара последних элементов (ε, y_n),

$E_{10} = (A(x, y[n-1])((), \varepsilon))\backslash E_{00}$ – пара последних элементов (x_m, ε),

$E_{11} = E\backslash(E_{00} \cup E_{01} \cup E_{10})$ – пара последних элементов (x_m, y_n), поскольку $E_{00} \cup E_{01} \cup E_{10}$ содержат все пары E-вложений общих A-вложений, в которых пара последних элементов содержит ε.

Очевидно, $E = E_{00} \cup E_{01} \cup E_{10} \cup E_{11}$. Пары последних элементов пар E-вложений из разных множеств E_{00}, E_{01}, E_{10}, E_{11} разные, поэтому эти множества попарно не пересекаются. Поэтому $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}|$, $|E_{00} \cup E_{01}| = |E_{00}| + |E_{01}|$, $|E_{00} \cup E_{10}| = |E_{00}| + |E_{10}|$.

Поскольку $A_3(x, y) = |A(x, y)|$, в этих обозначениях утверждение теоремы имеет вид

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

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

Рассмотрим случай $x_m \neq y_n$. В этом случае $E_{11} = \varnothing$, что влечет $|E_{11}| = 0$ и $|E| = |E_{00}| + |E_{01}| + |E_{10}|$, что и требовалось доказать в этом случае.

Рассмотрим случай $x_m = y_n = h$. Каждая пара E-вложений из E_{11} имеет вид $(u\varepsilon^{m-|u|-1}h, v\varepsilon^{n-|v|-1}h)$, где $(u\varepsilon^{m-|u|-1}\varepsilon, v\varepsilon^{n-|v|-1}\varepsilon) \in E_{00}$ (последовательности u и v могут быть обе пустыми). Будем говорить, что пара $(u\varepsilon^{m-|u|-1}h, v\varepsilon^{n-|v|-1}h)$ соответствует паре $(u\varepsilon^{m-|u|-1}\varepsilon, v\varepsilon^{n-|v|-1}\varepsilon)$. Это соответствие, очевидно, является биекцией множеств E_{11} и E_{00}. Тем самым, $|E_{11}| = |E_{00}|$. Поэтому $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{00}| = 2|E_{00}| + |E_{01}| + |E_{10}|$, что и требовалось доказать в этом случае.

□

Теорема 3 определяет алгоритм вычисления $A_3(x, y)$. Число шагов алгоритма равно $\mathbf{O}(mn)$, что определяется числом функций вида $A_3(x[m - i], y[n - j])$, где $i \in 0..m$ и $j \in 0..n$, при условии, что каждая функция вычисляется не более одного раза (после чего ее значение сохраняется). На каждом шаге вычисления имеют сложность $\mathbf{O}(1)$. Тем самым сложность алгоритма равна $\mathbf{O}(mn)$.

4.5. Функция похожести на основе наибольшей длины общего A-вложения

Задача вычисления длины наибольшей общей подпоследовательности (*longest common subsequence*, *lcs*) хорошо известна [1]. Простейший алгоритм сложности $\mathbf{O}(mn)$ основан на следующих соотношениях:

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

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

Тем самым, функция $A_4(x, y) = lcA(x, y)$ вычисляется за время $\mathbf{O}(mn)$.

5. L-ВЛОЖЕНИЯ

В отличие от A- и O-вложений L-вложению $u \in L(x)$ соответствует только одно E-вложение v такое, что $\lambda(v) = u$. Общему L-вложению $u \in L(x) \cap \cap L(y)$ соответствует пара E-вложений в x и y, которые могут отличаться только префиксом пустых символов. Поэтому множество $l(u, x)$ является синглетоном, $\{l_l(u, x)\} = \{l_r(u, x)\} = l(u, x)$, $L(x, y) = L_l(x, y) = L_r(x, y)$.

Обозначим через $\gamma(x, y)$ последовательность z длиной m (напомним, что $|x| = m \le n = |y|$), совпадающую с x и y по тем позициям, считая справа налево, по которым совпадают x и y, и содержащую пустой символ по остальным позициям: $|z| = m$ и $\forall i \in 1..m\ (x_{m+1-i} = y_{n+1-i} \Rightarrow z_{m+1-i} = x_{m+1-i})\ \&\ (x_{m+1-i} \neq y_{n+1-i} \Rightarrow z_{m+1-i} = \varepsilon)$. Функция $\gamma(x, y)$ устанавливает позиционное соответствие совпадающих символов x и y при счете позиций справа налево.

5.1. Число общих L-вложений

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

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

Доказательство.

Обозначим $L = L(x) \cap L(y)$ и $L_{-1} = L(x[m - 1]) \cap \cap L(y[n - 1])$. Если $x_m \neq y_n$, то L-вложению $u \in L_{-1}$ соответствует одно L-вложение $u\varepsilon \in L$. Тем самым, $L_0(x, y) = |L| = |L_{-1}| = L_0(x[m - 1], y[n - 1]))$. Если $x_m = y_n$, то L-вложению $u \in L_{-1}$ соответствуют два L-вложения $u\varepsilon \in L$ и $ux_m \in L$. Тем самым, $L_0(x, y) = |L| = 2|L_{-1}| = 2L_0(x[m - 1], y[n - 1]))$.

□

Теорема 4 определяет алгоритм вычисления $L_0(x, y)$. Для $m \le n$ число шагов алгоритма равно $\mathbf{O}(m)$, что определяется числом функций вида $L_0(x[m - i], y[n - i])$, где $i \in 0..m$. На каждом шаге вычисления имеют сложность $\mathbf{O}(1)$. Тем самым сложность алгоритма равна $\mathbf{O}(m)$.

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

Доказательство.

Из теоремы 4 следует, что при добавлении к обеим последовательностям справа по одному символу число общих L-вложений увеличивается вдвое, если добавляются равные символы, и не меняется в противном случае. Поскольку $|L(x) \cap \cap L(())| = |L(()) \cap L(y)| = |\varepsilon| = 1$, а $|\mu\gamma(x, y)|$ равно числу совпадающих символов в одинаковых позициях x и y при отсчете справа налево, имеем $L_0(x, y) = |L(x) \cap L(y)| = 2^{|\mu\gamma(x, y)|}$, что и требовалось доказать.

□

Теорема 5 определяет алгоритм вычисления $L_0(x, y)$. Сложность алгоритма равна сложности вычисления $\mu\gamma(x, y)$, которая, очевидно, для $m \le n$ равна $\mathbf{O}(m)$, плюс сложность возведения числа два в степень $|\mu\gamma(x, y)|$, которая тоже равна $\mathbf{O}(m)$. Тем самым сложность алгоритма равна $\mathbf{O}(m)$.

5.2. Сумма μ-длин общих L-вложений

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

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

Доказательство.

Обозначим $L = L(x) \cap L(y)$ и $L_{-1} = L(x[m-1]) \cap \cap L(y[n-1])$. Если $x_m \neq y_n$, то L-вложению $u \in L_{-1}$ соответствует одно L-вложение $u\varepsilon \in L$ той же μ-длины. Тем самым, $L_1(x, y) = \Sigma\{|\mu(u)| : u \in L\} = \Sigma\{|\mu(u)| : u \in L_{-1}\} = L_1(x[m-1], y[n-1]))$. Если $x_m = y_n$, то L-вложению $u \in L_{-1}$ соответствуют два L-вложения $u\varepsilon \in L$ той же μ-длины и $ux_m \in L$ с μ-длиной на 1 большей. Тем самым, $L_1(x, y) = \Sigma\{|\mu(u)| : u \in L\} = \Sigma\{|\mu(u)| : u \in L_{-1}\} + (\Sigma\{|\mu(u)| : u \in L_{-1}\} + L_0(x[m-1], y[n-1])) = 2\Sigma\{|\mu(u)| : u \in L_{-1}\} + L_0(x[m-1], y[n-1]) = 2L_1(x[m-1], y[n-1] + L_0(x[m-1], y[n-1])$.

□

Теорема 6 определяет алгоритм вычисления $L_1(x, y)$, сложность которого по порядку, очевидно, не превышает сложности алгоритма вычисления $L_0(x, y)$, т.е. равна $\mathbf{O}(m)$.

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

Доказательство.

Утверждение теоремы непосредственно следует из того факта, что число общих L-вложений μ-длины i равно $C^i_{|\mu\gamma(x,y)|}$.

□

Теорема 7 определяет алгоритм вычисления $L_1(x, y)$. Сложность алгоритма равна сложности вычисления $\mu\gamma(x, y)$, которая, очевидно, для $m \le n$ равна $\mathbf{O}(m)$, плюс сложность вычисления факториалов $i!$ для $i \in 0..l$, равная $\mathbf{O}(m)$, плюс сложность суммирования l чисел, равная $\mathbf{O}(m)$. Тем самым сложность алгоритма равна $\mathbf{O}(m)$.

5.3. Сумма минимумов и сумма произведений чисел E-вложений общих L-вложений

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

Доказательство.

Утверждение теоремы непосредственно следует из того факта, что L-вложению $u \in L(x)$ соответствует ровно одно E-вложение v такое, что $\lambda(v) = u$: $L_2(x, y) = \Sigma\{min\{|l(u, x)|, |l(u, y)|\} : u \in L(x) \cap \cap L(y)\} = \Sigma\{min\{1, 1\} : u \in L(x) \cap L(y)\} = |L(x) \cap \cap L(y)| = L_0(x, y)$; $L_3(x, y) = |L(x, y)| = |\cup\{l(u, x) \times \times l(u, y) : u \in L(x) \cap L(y)\}| = |L(x) \cap L(y)| = L_0(x, y)$.

□

Теорема 8 определяет алгоритм вычисления $L_2(x, y)$ и $L_3(x, y)$ той же сложности, что алгоритм вычисления $L_0(x, y)$, т.е. для $m \le n$ сложности $\mathbf{O}(m)$.

5.4. Функция похожести на основе наибольшей длины общего L-вложения

Теорема 9.

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

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

Доказательство. Функция $\gamma(x, y)$ устанавливает позиционное соответствие совпадающих символов x и y при счете позиций справа налево. Поэтому $lcL(x, y) = |\mu\gamma(x, y)|$. Отсюда непосредственно следует утверждение теоремы.

□

Теорема 9 определяет алгоритм вычисления функции $L_4(x, y) = lcL(x, y)$ сложности $\mathbf{O}(m)$ для $m \le n$.

6. *R*-ВЛОЖЕНИЯ

Аналогично L-вложению R-вложению $u \in R(x)$ соответствует только одно E-вложение v такое, что $\rho(v) = u$. Общему R-вложению $u \in R(x) \cap R(y)$ соответствует пара E-вложений в x и y, отличающихся только постфиксом пустых символов. Поэтому множество $r(u, x)$ является синглетоном, $\{r_l(u, x)\} = \{r_r(u, x)\} = r(u, x)$, $R(x, y) = R_l(x, y) = R_r(x, y)$.

Обозначим через $\delta(x, y)$ последовательность z длиной m (напомним, что $|x| = m \le n = |y|$), совпадающую с x и y по тем позициям, считая слева направо, по которым совпадают x и y, и содержащую пустой символ по остальным позициям: $|z| = m$ и $\forall i \in 1..m\ (x_i = y_i \Rightarrow z_i = x_i)\ \&\ (x_i \neq y_i \Rightarrow z_i = \varepsilon)$. Функция $\delta(x, y)$ устанавливает позиционное соответствие совпадающих символов x и y при счете позиций слева направо.

6.1. Число общих R-вложений

Теорема 10. $R_0(x, y) = R_0(x[2..m], y[2..n]))$, если $x_1 \neq y_1$,

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

Доказательство.

Обозначим $R = R(x) \cap R(y)$ и $R_{-1} = R(x[2..m]) \cap \cap R(y[2..n])$. Если $x_1 \neq y_1$, то R-вложению $u \in R_{-1}$ соответствует одно R-вложение $\varepsilon u \in R$. Тем самым, $R_0(x, y) = |R| = |R_{-1}| = R_0(x[2..m], y[2..n]))$. Если $x_1 = y_1$, то R-вложению $u \in R_{-1}$ соответствуют два R-вложения $\varepsilon u \in R$ и $x_1 u$. Тем самым, $R_0(x, y) = = |R| = 2|R_{-1}| = 2R_0(x[2..m], y[2..n]))$.

□

Теорема 10 определяет алгоритм вычисления $R_0(x, y)$. Для $m \le n$ число шагов алгоритма равно $\mathbf{O}(m)$, что определяется числом функций вида $R_0(x[i..m], y[i..n])$, где $i \in 0..m$. На каждом шаге проверка вычисления имеет сложность $\mathbf{O}(1)$. Тем самым сложность алгоритма равна $\mathbf{O}(m)$.

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

Доказательство.

Из теоремы 10 следует, что при добавлении к обеим последовательностям слева по одному символу число общих R-вложений увеличивается вдвое, если добавляются равные символы, и не меняется в противном случае. Поскольку $|R(x) \cap R(()) | = |R(()) \cap R(y)| = |\varepsilon| = 1$, а $|\mu\delta(x, y)|$ равно числу совпадающих символов в одинаковых позициях x и y при отсчете слева направо, имеем $R_0(x, y) = |R(x) \cap R(y)| = 2^{|\mu\delta(x, y)|}$, что и требовалось доказать.

□

Теорема 11 определяет алгоритм вычисления $R_0(x, y)$. Сложность алгоритма равна сложности вычисления $\mu\delta(x, y)$, которая, очевидно, для $m \le n$ равна $\mathbf{O}(m)$, плюс сложность возведения числа два в степень $|\mu\delta(x, y)|$, которая тоже равна $\mathbf{O}(m)$. Тем самым сложность алгоритма равна $\mathbf{O}(m)$.

6.2. Сумма μ-длин общих R-вложений

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

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

Доказательство.

Обозначим $R = R(x) \cap R(y)$ и $R_{-1} = R(x[2..m]) \cap \cap R(y[2..n])$. Если $x_1 \ne y_1$, то R-вложению $u \in R_{-1}$ соответствует одно R-вложение $\varepsilon u \in R$ той же μ-длины. Тем самым, $R_1(x, y) = \Sigma\{|\mu(u)| : u \in R\} = \Sigma\{|\mu(u)| : u \in R_{-1}\} = R_1(x[2..m], y[2..n]))$. Если $x_1 = y_1$, то R-вложению $u \in R_{-1}$ соответствуют два R-вложения $\varepsilon u \in R$ той же μ-длины и $x_1 u \in R$ с μ-длиной на 1 большей. Тем самым, $R_1(x, y) = \Sigma\{|\mu(u)| : u \in R\} = \Sigma\{|\mu(u)| : u \in R_{-1}\} + (\Sigma\{|\mu(u)| : u \in R_{-1}\} + R_0(x[2..m], y[2..n])) = 2\Sigma\{|\mu(u)| : u \in R_{-1}\} + R_0(x[2..m], y[2..n]) = 2R_1(x[2..m], y[2..n] + R_0(x[2..m], y[2..n])$.

□

Теорема 12 определяет алгоритм вычисления $R_1(x, y)$, сложность которого по порядку, очевидно, не превышает сложности алгоритма вычисления $R_0(x, y)$, т.е. для $m \le n$ равна $\mathbf{O}(m)$.

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

Доказательство.

Утверждение теоремы непосредственно следует из того факта, что число общих R-вложений μ-длины i равно $C^i_{|\mu\delta(x,y)|}$.

□

Теорема 13 определяет алгоритм вычисления $R_1(x, y)$. Сложность алгоритма равна сложности вычисления $\mu\delta(x, y)$, которая, очевидно, для $m \le n$ равна $\mathbf{O}(m)$, плюс сложность вычисления факториалов $i!$ для $i \in 0..r$, равная $\mathbf{O}(m)$, плюс сложность суммирования r чисел, равная $\mathbf{O}(m)$. Тем самым сложность алгоритма равна $\mathbf{O}(m)$.

6.3. Сумма минимумов и сумма произведений чисел E-вложений общих R-вложений

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

Доказательство.

Утверждение теоремы непосредственно следует из того факта, что R-вложению $u \in R(x)$ соответствует ровно одно E-вложение v такое, что $\rho(v) = u$: $R_2(x, y) = \Sigma\{min\{|r(u, x)|, |r(u, y)|\} : u \in R(x) \cap \cap R(y)\} = \Sigma\{min\{1, 1\} : u \in R(x) \cap R(y)\} = |R(x) \cap \cap R(y)| = R_0(x, y)$; $R_3(x, y) = |R(x, y)| = |\cup \{r(u, x) \times \times r(u, y) : u \in R(x) \cap R(y)\}| = |R(x) \cap R(y)| = R_0(x, y)$.

□

Теорема 14 определяет алгоритм вычисления $R_2(x, y)$ и $R_3(x, y)$ той же сложности, что алгоритм вычисления $R_0(x, y)$, т.е. для $m \le n$ сложности $\mathbf{O}(m)$.

6.4. Функция похожести на основе наибольшей длины общего R-вложения

Теорема 15.

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

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

Доказательство. Функция $\delta(x, y)$ устанавливает позиционное соответствие совпадающих символов x и y при счете позиций слева направо. Поэтому $lcR(x, y) = |\mu\delta(x, y)|$. Отсюда непосредственно следует утверждение теоремы.

□

Теорема 15 определяет алгоритм вычисления функции $R_4(x, y) = lcR(x, y)$ сложности $\mathbf{O}(m)$ для $m \le n$.

7. *O*-ВЛОЖЕНИЯ

7.1. Число общих O-вложений

Для случая $x_m = y_n = h$ обозначим через $I = \{i \in \in 1..m : x_i = h\}$ и $J = \{j \in 1..n : y_j = h\}$ множества индексов по которым находится символ h в x и y, соответственно.

Обозначим для $i \in I, j \in J$: $L_{i,j} = L(x[i - 1]) \cap \cap L(y[j - 1])$.

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

Для $i \in I$ и $j \in J$ обозначим через $u_{i,j}$ максимальное общее L-вложение в $x[i - 1]$ и $y[j - 1]$: $u_{i,j} = = \lambda(v_{i,j})$, где $v_{i,j}$ общее E-вложение в $x[i - 1]$ и $y[j - 1]$, определяемое условием: $\forall t = 1..min\{i, j\} - 1$ ($x_{i-t} =$

$= y_{i-t} \Rightarrow v_{i,j}(i-t) = x_{i-t}$) & ($x_{i-t} \neq y_{i-t} \Rightarrow v_{i,j}(i-t) = \varepsilon$). Множество всех общих L-вложений в $x[i-1]$ и $y[j-1]$ равно $\lambda E(u_{i,j})$, т.е. получается из $u_{i,j}$ всеми возможными заменами некоторых символов на пустой символ и последующим удалением префикса пустых символов.

Лемма 1. Пусть $x_m = y_n = h$. Вычисление $|L_h(x, y)|$ эквивалентно вычислению числа слагаемых СДНФ некоторой конъюнкции дизъюнкций переменных без отрицаний, где число переменных равно числу непустых символов в максимальном общем L-вложении $u_{m,n}$ в $x[m-1]$ и $y[n-1]$.

Доказательство.

$L_h(x, y) = L_{m,n} \setminus \cup \{L_{i,j} : (i, j) \in K\} = L_{m,n} \setminus \cup \{L_{m,n} \cap L_{i,j} : (i, j) \in K\}$. Для получения $L_h(x, y)$ нам нужно из $L_{m,n}$ удалить множество $L_{m,n} \cap L_{i,j}$ для каждых $(i, j) \in K$.

Определим "пересечение" $u^{\wedge}_{i,j}$ вложений $u_{m,n}$ и $u_{i,j}$, имеющее длину $|u_{m,n}|$ и совпадающее с $u_{m,n}$ и $u_{i,j}$ в тех позициях, считая справа налево, в которых они совпадают между собой, а во всех остальных позициях содержит пустой символ: $|u^{\wedge}_{i,j}| = |u_{m,n}|$ и $\forall\, t = 1..|u_{m,n}|$ ($t > |u_{i,j}| \Rightarrow u^{\wedge}_{i,j}(|u_{m,n}| - t) = \varepsilon$) & ($t \le |u_{i,j}|$ & $u_{m,n}(|u_{m,n}| - t) = u_{i,j}(|u_{i,j}| - t) \Rightarrow u^{\wedge}_{i,j}(|u_{m,n}| - t) = u_{m,n}(|u_{m,n}| - t)$) & ($t \le |u_{i,j}|$ & $u_{m,n}(|u_{m,n}| - t) \neq u_{i,j}(|u_{i,j}| - t) \Rightarrow u^{\wedge}_{i,j}(|u_{m,n}| - t) = \varepsilon$). Очевидно, $u^{\wedge}_{m,n} = u_{m,n}$.

Удалим из вложения $u^{\wedge}_{i,j}$ символы в тех позициях, в которых $u_{m,n}$ содержит пустой символ, и обозначим результат $w_{i,j}$: если $u^{\wedge}_{i,j} = u_1 h_1 u_2 h_2 \ldots u_k h_{k-1} u_k$, $u_{m,n} = v_1 \varepsilon v_2 \varepsilon \ldots v_{k-1} \varepsilon v_k$ и для $i \in 1..k$ $|u_i| = |v_i|$ и $v_i \in H^*$, то $w_{i,j} = u_1 u_2 \ldots u_{k-1} u_k$. Очевидно, $w_{m,n} = \mu(u_{m,n})$.

Все $w_{i,j}$ имеют одинаковую длину $|w_{m,n}|$, равную числу непустых символов в $u_{m,n}$. Каждому L-вложению из $L_{m,n} \cap L_{i,j}$ взаимно-однозначно соответствует E-вложение в $w_{i,j}$, число таких вложений равно 2^k, где $k = |\mu(w_{i,j})|$ число непустых символов в $w_{i,j}$.

Поставим каждому $t \in 1..|w_{m,n}|$ в соответствие булеву переменную α_t, означающую, что в позиции t может быть непустой символ. Тогда E-вложения в $w_{i,j}$ задаются булевой функцией $F_{i,j} = \&\{\neg\alpha_t : t \in 1..|w_{m,n}| \;\&\; w_{i,j}(t) = \varepsilon\}$, которая принимает значение **true** на тех и только тех наборах α_1, ..., в которых $\alpha_t =$ **false** для всех тех индексов t, по которым в $w_{i,j}$ и, следовательно, в любом E-вложении в $w_{i,j}$ находится пустой символ. Если по индексу t в $w_{i,j}$ находится непустой символ, то в одних E-вложениях в $w_{i,j}$ в этой позиции будет пустой символ, а в других непустой символ, что означает, что функция $F_{i,j}$ не зависит от α_t. Очевидно, что $F_{m,n} = \&\varnothing =$ **true**.

Разность $L_{m,n} \setminus \cup \{L_{m,n} \cap L_{i,j} : (i, j) \in K\}$ задается булевой функцией $F = F_{m,n} \setminus \vee \{F_{i,j} : (i, j) \in K\} = \& \{\neg F_{i,j} : (i, j) \in K\}$, и $\neg F_{i,j} = \vee\{\alpha_t : t \in 1..|w_{m,n}| \;\&\; w_{i,j}(t) = \varepsilon\}$. Таким образом, F – это конъюнкция дизъюнкций переменных без отрицаний, и число $|L_h(x, y)|$ равно числу слагаемых в СДНФ функции F.

□

Теорема 16.

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

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

Доказательство.

Обозначим следующие множества пар E-вложений:

$E = O_r(x, y)$,

$E_{00} = O_r(x[m-1], y[n-1])(\varepsilon, \varepsilon)$ – пара последних элементов $(\varepsilon, \varepsilon)$,

$E_{01} = (O_r(x[m-1], y])(\varepsilon, ()))\setminus E_{00}$ – пара последних элементов (ε, y_n),

$E_{10} = (O_r(x, y[n-1])((), \varepsilon))\setminus E_{00}$ – пара последних элементов (x_m, ε),

$E_{11} = E\setminus(E_{00} \cup E_{01} \cup E_{10})$ – пара последних элементов (x_m, y_n), поскольку $E_{00} \cup E_{01} \cup E_{10}$ содержат все пары E-вложений общих O-вложений, в которых пара последних элементов содержит ε.

Очевидно, $E = E_{00} \cup E_{01} \cup E_{10} \cup E_{11}$. Пары последних элементов пар E-вложений из разных множеств E_{00}, E_{01}, E_{10}, E_{11} разные, поэтому эти множества попарно не пересекаются. Поэтому $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}|$, $|E_{00} \cup E_{01}| = |E_{00}| + |E_{01}|$, $|E_{00} \cup E_{10}| = |E_{00}| + |E_{10}|$.

Поскольку для любых x, y имеет место $O_0(x, y) = |O(x) \cap O(y)| = |O_r(x, y)|$ и $L_0(x, y) = L_3(x, y) = |L(x, y)|$, утверждение теоремы можно переписать в виде:

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

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

Рассмотрим случай $x_m \neq y_n$. В этом случае $E_{11} = \varnothing$, что влечет $|E_{11}| = 0$ и $|E_{00}| = |E_{00}| + |E_{01}| + |E_{10}|$, что и требовалось доказать в этом случае.

Рассмотрим случай $x_m = y_n = h$. В этом случае каждая пара E-вложений из E_{11} имеет вид $(\varepsilon^{m-|v|-k-1} v \varepsilon^k h, \varepsilon^{n-|v|-k-1} v \varepsilon^k h)$, где $(\varepsilon^{m-|v|-k-1} v \varepsilon^k, \varepsilon^{n-|v|-k-1} v \varepsilon^k) \in L_h(x, y)$, т.е. при переходе от $x[m-1]$ и $y[n-1]$ к x и y эта пара образована добавлением справа символа h к паре E-вложений общих L-вложений в $x[m-1]$ и $y[n-1]$, но таких, которых не было раньше, т.е. которые не являются парой E-вложений общих L-вложений в $x[i-1]$ и $y[j-1]$, где $x_i = y_j = h$ и $i < m$ или $j < n$. Будем говорить, что пара $(\varepsilon^{m-|v|-k-1} v \varepsilon^k h,$

$\varepsilon^{n-|v|-k-1}v\varepsilon^k h$) соответствует паре ($\varepsilon^{m-|v|-k-1}v\varepsilon^k$, $\varepsilon^{n-|v|-k-1}v\varepsilon^k$). Это соответствие, очевидно, является биекцией множеств E_{11} и $L_h(x, y)$. Тем самым, $|E_{11}| = |L_h(x, y)|$. Поэтому $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}| = |E_{00}| + |E_{01}| + |E_{10}| + |L_h(x, y)|$, что и требовалось доказать в этом случае.

□

Теорема 16 определяет алгоритм вычисления $O_0(x, y)$. Число шагов алгоритма равно $\mathbf{O}(mn)$, что определяется числом функций вида $O_0(x[m - i], y[n - j])$, где $i \in 0..m$ и $j \in 0..n$, при условии, что каждая функция вычисляется не более одного раза (после чего ее значение сохраняется). На каждом шаге вычисления имеют сложность $\mathbf{O}(1)$. Тем самым сложность алгоритма равна $\mathbf{O}(mn)$*$\mathbf{O}(L_h(x, y))$, где $\mathbf{O}(L_h(x, y))$ сложность вычисления функции $|L_h(x, y)|$.

Лемма 2. Пусть имеется k не обязательно различных множеств $a(s)$, $s \in 1..k$. Обозначим для $S \subseteq 1..k$, $S \neq \varnothing$, пересечение множеств $a(s)$, индексы которых пробегают множество S, через $c(S) = \cap\{a(s) : s \in S\}$ и сумму числа подмножеств множества $c(S)$ по всем S, для которых $|S| = l$, через $E(l) = \Sigma\{2^{|c(S)|} : S \subseteq 1..k \ \& \ |S| = l\}$. Тогда число различных множеств, вложенных хотя бы в одно из множеств $a(s)$, $s \in 1..k$, равно чередующейся сумме $E(1) - E(2) + E(3) - E(4) ... (-1)^{k+1}E(k)$, а сумма F размеров этих множеств, увеличенных на 1, равна чередующейся сумме $F(1) - F(2) + F(3) - F(4) ... (-1)^{k+1}F(k)$, где $F(l) = \Sigma\{(|c(S)| + 2)\, 2^{|c(S)|-1} : S \subseteq 1..k \ \& \ |S| = l\}$. Эти чередующиеся суммы могут быть вычислены за время $\mathbf{O}(2^k)$ при условии, что за время $\mathbf{O}(1)$ могут выполняться операции пересечения двух множеств (а также арифметические операции над целыми числами).

Доказательство.

Обозначим число различных множеств, вложенных хотя бы в одно из множеств $a(s)$, $s \in 1..k$, через $E(a(1), ..., a(k))$, а сумму их размеров, увеличенных на 1, через $F(a(1), ..., a(k))$.

Будем вести доказательство индукцией по k. Для $k = 1$ имеется единственное множество $S \subseteq 1..1$, $S \neq \varnothing$, а именно $S = \{1\}$, и $c(\{1\}) = \cap\{a(s) : s \in \{1\}\} = a(1)$. Число различных подмножеств множества $a(1)$ равно $2^{|a(1)|}$ и $E(a(1)) = E(1) = 2^{|a(1)|}$, а сумма их длин, увеличенных на 1, равна $1*C_r^0 + 2*C_r^1 + ... + r*C_r^{r-1} + (r + 1)*C_r^r = (r + 2)2^{r-1}$ (A001792), и $F(a(1)) = F(1) = (r + 2)2^{r-1}$, где $r = |a(1)|$.

Пусть утверждение верно для k и докажем его для $k + 1$.

Число различных множеств, вложенных хотя бы в одно из множеств $a(s)$, где $s \in 1..k + 1$, равно числу различных множеств, вложенных хотя бы в одно из множеств $a(s)$, где $s \in 1..k$, т.е. $E(a(1), ..., a(k))$, плюс число различных множеств, вложенных в $a(k + 1)$, т.е. $2^{|a(k+1)|}$, кроме тех, что вложены в пересечение $a(k + 1)$ с объединением множеств $a(1)$, ..., $a(k)$, число которых равно $E(a(k + 1) \cap \cap a(1), ..., a(k + 1) \cap a(k))$.

Обозначим $E_{k+1} = E(a(1), ..., a(k + 1))$, $E_k = E(a(1), ..., a(k))$, $E_k^\wedge = E(a(k + 1) \cap a(1), ..., a(k + 1) \cap \cap a(k))$. Имеем $E_{k+1} = E_k + 2^{|a(k+1)|} - E_k^\wedge$.

Рассмотрим $S \subseteq 1..k + 1$. Пусть $|S| = 1$, тогда $S = \{i\}$, $i \in 1..k + 1$. Для $i \in 1..k$ слагаемое $2^{|cS)|} = 2^{|a(i)|}$ один раз входит в сумму E_k со знаком "+", в результате для $i \in 1..k + 1$ слагаемое $2^{|cS)|} = 2^{|a(i)|}$ один раз входит в сумму E_{k+1} с тем же знаком. Пусть $|S| > 1$. Если $k + 1 \notin S$, то слагаемое $2^{|c(S)|}$ один раз входит в сумму E_k со знаком "$(-1)^{|S|+1}$", в результате для $i \in \in 1..k + 1$ слагаемое $2^{|c(S)|}$ один раз входит в сумму E_{k+1} с тем же знаком. Если $k + 1 \in S$, то $c(S) = \{a(i_1) \cap ... \cap a(i_{|S|-1}) \cap a(k + 1)\} = \{(a(k + 1) \cap \cap a(i_1)) \cap ... \cap (a(k + 1) \cap a(i_{|S|-1}))\}$. Поэтому слагаемое $2^{|c(S)|}$ один раз входит в сумму $E_k^\wedge$ со знаком "$(-1)^{|S|}$", но, поскольку сумма $E_k^\wedge$ вычитается, слагаемое $2^{|c(S)|}$ один раз входит в сумму E_{k+1} со знаком "$(-1)^{|S|+1}$".

Аналогично сумма длин, увеличенных на 1, различных множеств, вложенных хотя бы в одно из множеств $a(s)$, где $s \in 1..k + 1$, равно сумме длин, увеличенных на 1, различных множеств, вложенных хотя бы в одно из множеств $a(s)$, где $s \in 1..k$, т.е. $F(a(1), ..., a(k))$, плюс сумма длин, увеличенных на 1, различных множеств, вложенных в $a(k + 1)$, т.е. $(|a(k + 1)| + 2)2^{|a(k+1)|-1}$, кроме тех, что вложены в пересечение $a(k + 1)$ с объединением множеств $a(1)$, ..., $a(k)$, сумма длин которых, увеличенных на 1, равна $F(a(k + 1) \cap a(1), ..., a(k + 1) \cap a(k))$. Обозначим $F_{k+1} = F(a(1), ..., a(k + 1))$, $F_k = F(a(1), ..., a(k))$, $F_k^\wedge = F(a(k + 1) \cap a(1), ..., a(k + 1) \cap a(k))$. Имеем $F_{k+1} = F_k + (|a(k + 1)| + 2)2^{|a(k+1)|-1} - F_k^\wedge$. Рассмотрим $S \subseteq 1..k + 1$. Пусть $|S| = 1$, тогда $S = \{i\}$, $i \in 1..k + 1$. Для $i \in 1..k$ слагаемое $(|c(S)| + 2)2^{|c(S)|-1} = (|a(i)| + 2)2^{|a(i)|-1}$ один раз входит в сумму F_k со знаком "+", в результате для $i \in 1..k + 1$ слагаемое $(|c(S)| + 2)2^{|c(S)|-1} = (|a(i)| + 2)2^{|a(i)|-1}$ один раз входит в сумму F_{k+1} с тем же знаком. Пусть $|S| > 1$. Если $k + 1 \notin S$, то слагаемое $(|c(S)| + 2)2^{|c(S)|-1}$ один раз входит в сумму F_k со знаком "$(-1)^{|S|+1}$", в результате для $i \in 1..k + 1$ слагаемое $(|c(S)| + 2)2^{|c(S)|-1}$ один раз входит в сумму F_{k+1} с тем же знаком. Если $k + 1 \in S$, то слагаемое $(|c(S)| + 2)2^{|c(S)|-1}$ один раз входит в сумму $F_k^\wedge$ со знаком "$(-1)^{|S|}$", но, поскольку сумма $F_k^\wedge$ вычитается, слагаемое $(|c(S)| + 2)2^{|c(S)|-1}$ один раз входит в сумму F_{k+1} со знаком "$(-1)^{|S|+1}$".

Утверждение доказано.

Число (не обязательно различных) множеств $c(S)$ равно числу непустых подмножеств S множества $1..k$, которое равно $2^k - 1$, поэтому сложность вычисления E равна $\mathbf{O}(2^k)$ при условии, что за время $\mathbf{O}(1)$ могут выполняться операции пересечения двух множеств (а также арифметические операции над целыми числами).

□

Обозначим множество индексов последовательности x, по которым находится символ h: $K_x(h) = \{i \in 1..m : x(i) = h\}$. Для $i \in K_x(h)$ обозначим множество пар (символ в x, позиция символа в x относительно позиции i), кроме пары $(h, 0)$: $P_x(i) = \{(x(t), t - i) : t \in 1..m \,\&\, t \neq i\}$. Пусть символ h входит $|K_x(h)| > 0$ раз в x и $|K_y(h)| > 0$ раз в y. Обозначим $k = |K_x(h)|*|K_y(h)|$. Для $i \in K_x(h)$ и $j \in K_y(h)$ для краткости обозначим $P(i,j) = P_x(i) \cap P_y(j)$. Для $S \subseteq \subseteq K_x(h) \times K_y(h)$, $S \neq \varnothing$ обозначим $c(S) = \cap\{P(i,j) : (i,j) \in S\}$.

Теорема 17. Число различных общих O-вложений в x и y, содержащих символ h, равно чередующейся сумме $E = E(1) - E(2) + E(3) - E(4) \ldots (-1)^{k+1}E(k)$, где слагаемое $E(l)$ это сумма числа всех подмножеств множества $c(S)$ по всем S размера l, $|S| = l$: $E(l) = \Sigma\{2^{|c(S)|} : S \subseteq 1..k \,\&\, |S| = l\}$. Сложность вычисления равна $\mathbf{O}(n2^k)$.

Доказательство.

Рассмотрим общее O-вложение u и пару его E-вложений $v(x) \in o(u, x)$ и $v(y) \in o(u, y)$ таких, что $v(x)_i = x_i = h$ и $v(y)_j = y_j = h$. Этому взаимно-однозначно соответствует подмножество множества пар $P(i,j)$. Нам нужно вычислить число различных множеств, вложенных хотя бы в одно из множеств $P(i,j)$, где $i \in K_x(h)$ и $j \in K_y(h)$. Число таких множеств пар $P(i,j)$ равно k. По лемме 2 оно равно $E = E(1) - E(2) + E(3) - E(4) \ldots (-1)^{k+1}E(k)$, где $E(l) = \Sigma\{2^{|c(S)|} : S \subseteq 1..k \,\&\, |S| = l\}$, $l = 1..k$, сумма числа всех подмножеств множества $c(S)$ по всем $S \subseteq K_x(h) \times K_y(h)$ размера l, т.е. $|S| = l$, а $c(S) = \cap\{P(i, j) : (i,j) \in S\}$.

Просматриваем последовательность x длиной m, вычисляем $K_x(h)$. Просматривая $K_x(h)$, вычисляем множества $P_x(i)$. Вычисление множества $P_x(i)$ требует просмотра последовательности x длиной m. Тем самым, все множества $P_x(i)$, $i \in K_x(h)$, вычисляются за $\mathbf{O}(m|K_x(h)|)$. Аналогично все множества $P_y(j)$, $j \in K_y(h)$, вычисляются за $\mathbf{O}(n|K_x(h)|)$. Можно считать, что множество $P_x(i) = \{(x(t), t - i) : t \in 1..m \,\&\, t \neq i\}$ линейно упорядочено по возрастанию относительного индекса $t - i$; размер этого множества равен $m - 1$. Аналогично для множества $P_y(j)$; размер этого множества равен $n - 1$. Тогда для построения пересечения $P(i,j) = P_x(i) \cap P_y(j)$ требуется просмотр этих множеств, т.е. время $\mathbf{O}(n + m)$, а все множества $P(i,j)$, $i \in K_x(h)$ и $j \in K_y(h)$, вычисляются за время $\mathbf{O}(k(n + m))$. Аналогично каждое пересечение множеств $P(i, j)$ строится за время $\mathbf{O}(n + m)$. Сложность вычисления суммы E по лемме 2 равна $\mathbf{O}(2^k)$, но при условии, что пересечение двух множеств строится за время $\mathbf{O}(1)$, поэтому в данном случае потребуется время $\mathbf{O}((n + m)2^k)$. Общая сложность равна $\mathbf{O}(m) + \mathbf{O}(n) + \mathbf{O}(m|K_x(h)|) + \mathbf{O}(n|K_y(h)|) + \mathbf{O}(k(n + m)) + \mathbf{O}((n + m)2^k) = \mathbf{O}((n + m)2^k)$, что для $m \leq n$ равно $\mathbf{O}(n2^k)$.

□

Теорема 17 определяет следующий алгоритм вычисления $O_0(x, y)$:

1. SUMMA = 0.

2. Просматривая последовательность x, ищем непустой символ h, входящий в x.

2.1. Если не нашли, то конец алгоритма.

2.2. Если нашли, то ищем символ h в y.

2.2.1. Если не нашли, то в x заменяем h на пустой символ и переходим на п. 2.

2.2.2. Если нашли, то:

2.2.2.1. Вычисляем множество $K_x(h)$ индексов в x, по которым находится h, и множество $K_y(h)$ индексов в y, по которым находится h.

2.2.2.2. Для каждой пары $(i, j) \in K_x(h) \times K_y(h)$ строим множество пар $P(i,j)$.

2.2.2.3. Для каждого множества пар индексов $S \subseteq K_x(h) \times K_y(h)$, $S \neq \varnothing$ строим пересечение $c(S) = \cap\{P(i,j) : (i,j) \in S\}$.

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

2.2.2.5. SUMMA = SUMMA + E.

2.2.2.6. В x и в y заменяем h на пустой символ и переходим на п. 2.

7.2. Сумма μ-длин общих O-вложений

Теорема 18. Сумма μ-длин различных общих O-вложений в x и y, содержащих символ h, равно чередующейся сумме $F = F(1) - F(2) + F(3) - F(4) \ldots (-1)^{k+1}F(k)$, где $F(l) = \Sigma\{(|c(S)| + 2)\, 2^{|c(S)|-1} : S \subseteq \subseteq 1..k \,\&\, |S| = l\}$ сумма мощностей всех подмножеств всех множеств $c(S)$ для $|S| = l$. Сложность вычисления $\mathbf{O}(n2^k)$.

Доказательство аналогично доказательству теоремы 17.

□

Теорема 18 определяет следующий алгоритм вычисления $O_1(x, y)$:

1. Просматривая последовательность x, ищем непустой символ h, входящий в x.

1.1. Если не нашли, то конец алгоритма.

1.2. Если нашли, то ищем символ h в y.

1.2.1. Если не нашли, то в x заменяем h на пустой символ и переходим на п. 2.

1.2.2. Если нашли, то:

1.2.2.1. Вычисляем множество $K_x(h)$ индексов в x, по которым находится h, и множество $K_y(h)$ индексов в y, по которым находится h.

1.2.2.2. Для каждой пары $(i, j) \in K_x(h) \times K_y(h)$ строим множество пар $P(i, j)$.

1.2.2.3. Для каждого множества пар индексов $S \subseteq K_x(h) \times K_y(h)$, $S \neq \varnothing$ строим пересечение $c(S) = \cap \{P(i, j) : (i, j) \in S\}$.

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

1.2.2.5. SUMMA = SUMMA + F.

1.2.2.6. В x и в y заменяем h на пустой символ и переходим на п. 2.

7.3. Сумма минимумов чисел E-вложений общих O-вложений

Теорема 19. Сумма минимумов чисел E-вложений общих O-вложений в x и y, содержащих символ h, равна $\Sigma\{min\{|I|, |J|\} * 2^{|A(I, J)|-1} : I \subseteq K_x(h)\ \&\ I \neq \varnothing\ \&\ J \subseteq K_y(h)\ \&\ J \neq \varnothing\}$, где $A(I, J) = (\cap\{P(i, j) : i \in I, j \in J\})\backslash(\cup\{P(i, j): i \in K_x(h)\backslash I \vee j \in K_y(h)\backslash J\})$.

Сложность вычисления равна $\mathbf{O}(n2^k)$.

Доказательство. Множество $A(I, J)$ представляет все общие O-вложения, содержащие символ h, которые имеют E-вложения в x, представляемые множеством I, и E-вложения в y, представляемые множеством J. Для каждого из таких O-вложений минимум чисел их E-вложений в x и в y равен $min\{|I|, |J|\}$. Число таких O-вложений равно $2^{|A(I, J)|}$, поэтому сумма минимумов чисел E-вложений таких общих O-вложений равна $min\{|I|, |J|\} * 2^{|A(I, J)|}$. Суммируя по всем парам множеств $I \subseteq K_x(h)$, $I \neq \varnothing$ и $J \subseteq K_y(h)$, $J \neq \varnothing$, получаем искомую сумму минимумов чисел E-вложений общих O-вложений в x и y, содержащих символ h.

При вычислении этой суммы операции сложения, умножения, возведения в степень числа 2 и вычисление минимума из двух чисел выполняются $\mathbf{O}(2^k)$ раз. Для вычисления всех $A(I, J)$ операции вычисления разности двух множеств, пересечения двух множеств, объединения двух множеств выполняются $\mathbf{O}(2^k)$ раз, а каждая такая операция выполняется за время $\mathbf{O}(n + m)$.

Просматриваем последовательность x длиной m, вычисляем $K_x(h)$. Просматривая $K_y(h)$, вычисляем множества $P_x(i)$. Вычисление множества $P_x(i)$ требует просмотра последовательности x длиной m. Тем самым, все множества $P_x(i)$, $i \in K_x(h)$, вычисляются за $\mathbf{O}(m|K_x(h)|)$. Аналогично все множества $P_y(j)$, $j \in K_y(h)$, вычисляются за $\mathbf{O}(n|K_y(h)|)$. Можно считать, что множество $P_x(i) = \{(x(t), t - i) : t \in 1..m\ \&\ t \neq i\}$ линейно упорядочено по возрастанию относительного индекса $t - i$; размер этого множества равен $m - 1$. Аналогично для множества $P_y(j)$; размер этого множества равен $n - 1$. Тогда для построения пересечения $P(i, j) = P_x(i) \cap P_y(j)$ требуется просмотр этих множеств, т.е. время $\mathbf{O}(n + m)$, а все множества $P(i, j)$, $i \in K_x(h)$ и $j \in K_y(h)$, вычисляются за время $\mathbf{O}(k(n + m))$. Для вычисления множества $A(I, J)$ операции вычисления разности двух множеств, пересечения двух множеств, объединения двух множеств выполняются за время $\mathbf{O}(n + m)$, а число таких множеств $A(I, J)$ равно $\mathbf{O}(2^k)$. В результате все множества $A(I, J)$ строятся за время $\mathbf{O}((n + m)2^k)$, после чего для вычисления суммы арифметические операции выполняются $\mathbf{O}(2^k)$ раз. Общая сложность равна $\mathbf{O}(k(n + m)) + \mathbf{O}((n + m)2^k) + \mathbf{O}(2^k) = \mathbf{O}((n + m)2^k)$, что для $m \leq n$ равно $\mathbf{O}(n2^k)$.

□

Теорема 19 определяет следующий алгоритм вычисления $O_2(x, y)$:

1. Просматривая последовательность x, ищем непустой символ h, входящий в x.

1.1. Если не нашли, то конец алгоритма.

1.2. Если нашли, то ищем символ h в y.

1.2.1. Если не нашли, то в x заменяем h на пустой символ и переходим на п. 2.

1.2.2. Если нашли, то:

1.2.2.1. Вычисляем множество $K_x(h)$ индексов в x, по которым находится h, и множество $K_y(h)$ индексов в y, по которым находится h.

1.2.2.2. Для каждой пары $(i, j) \in K_x(h) \times K_y(h)$ строим множество пар $P(i, j)$.

1.2.2.3. Для каждых $I \subseteq K_x(h)$, $I \neq \varnothing$, $J \subseteq K_y(h)$, $J \neq \varnothing$ строим $A(I, J)$.

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

1.2.2.5. SUMMA = SUMMA + M.

1.2.2.6. В x и в y заменяем h на пустой символ и переходим на п. 2.

7.4. Сумма произведений чисел E-вложений общих O-вложений

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

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

Доказательство. Обозначим следующие множества пар E-вложений:

$E = O(x, y)$,

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

$E_{00} = O(x[m-1], y[n-1])(\varepsilon, \varepsilon)$ – пара последних элементов $(\varepsilon, \varepsilon)$,

$E_{01} = (O(x[m-1], y])(\varepsilon, ()))\backslash E_{00}$ – пара последних элементов (ε, y_n),

$E_{10} = (O(x, y[n-1])((), \varepsilon))\backslash E_{00}$ – пара последних элементов (x_m, ε),

$E_{11} = E\backslash(E_{00} \cup E_{01} \cup E_{10})$ – пара последних элементов (x_m, y_n), поскольку $E_{00} \cup E_{01} \cup E_{10}$ содержат все пары E-вложений общих O-вложений, в которых пара последних элементов содержит ε.

Очевидно, $E = E_{00} \cup E_{01} \cup E_{10} \cup E_{11}$. Пары последних элементов пар E-вложений из разных множеств E_{00}, E_{01}, E_{10}, E_{11} разные, поэтому эти множества попарно не пересекаются. Поэтому $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}|$, $|E_{00} \cup E_{01}| = |E_{00}| + |E_{01}|$, $|E_{00} \cup E_{10}| = |E_{00}| + |E_{10}|$.

В этих обозначениях утверждение теоремы имеет вид

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

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

Рассмотрим случай $x_m \neq y_n$. В этом случае $E_{11} = \varnothing$, что влечет $|E_{11}| = 0$ и $|E_{00}| = |E_{00}| + |E_{01}| + |E_{10}|$, что и требовалось доказать в этом случае.

Рассмотрим случай $x_m = y_n = h$. Каждая пара E-вложений из E_{11} имеет вид $(\varepsilon^{m-|v|-k-1}v\varepsilon^k h, \varepsilon^{n-|v|-k-1}v\varepsilon^k h)$, где $(\varepsilon^{m-|v|-k-1}v\varepsilon^k, \varepsilon^{n-|v|-k-1}v\varepsilon^k) \in L_{00}$. Будем говорить, что пара $(\varepsilon^{m-|v|-k-1}v\varepsilon^k h, \varepsilon^{n-|v|-k-1}v\varepsilon^k h)$ соответствует паре $(\varepsilon^{m-|v|-k-1}v\varepsilon^k, \varepsilon^{n-|v|-k-1}v\varepsilon^k)$. Это соответствие, очевидно, является биекцией множеств E_{11} и L_{00}. Тем самым, $|E_{11}| = |L_{00}|$. Поэтому $|E| = |E_{00}| + |E_{01}| + |E_{10}| + |E_{11}| = |E_{00}| + |E_{01}| + |E_{10}| + |L_{00}|$, что и требовалось доказать в этом случае.

□

Теорема 20 определяет алгоритм вычисления $O_3(x, y)$. Число шагов алгоритма равно $\mathbf{O}(mn)$, что определяется числом функций вида $O_3(x[m-i], y[n-j])$ и $L_3(x[m-i], y[n-i])$, где $i \in 0..m$ и $j \in 0..n$, при условии, что каждая функция вычисляется не более одного раза (после чего ее значение сохраняется). На каждом шаге вычисления имеют сложность $\mathbf{O}(1)$. Тем самым сложность алгоритма равна $\mathbf{O}(mn)$.

7.5. Функция похожести на основе наибольшей длины общего O-вложения

Теорема 21.

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

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

Доказательство. Если $x_m = y_n$, то самые правые E-вложения общего O-вложения могут иметь в позиции m в x и в позиции n в y либо 1) символ $x_m = y_n$, либо 2) пустой символ. Если общее O-вложение наибольшей длины относится к случаю 1, то оно имеет вид ux_m, где u общее L-вложение префиксов $x[m-1]$ и $y[n-1]$, т.е. его μ-длина на 1 больше наибольшей длины общего L-вложения префиксов $x[m-1]$ и $y[n-1]$. Если общее O-вложение наибольшей длины относится к случаю 2, то оно является O-вложением наибольшей длины префиксов $x[m-1]$ и $y[n-1]$ и имеет такую же μ-длину. Отсюда следует утверждение теоремы для случая $x_m = y_n$.

Если $x_m \neq y_n$, то самые правые E-вложения общего O-вложения имеют пустой символ либо 1) в позиции m в x, либо 2) в позиции n в y. Если общее O-вложение наибольшей длины относится к случаю 1, то оно является O-вложением наибольшей длины префикса $x[m-1]$ и последовательности y и имеет такую же μ-длину. Если общее O-вложение наибольшей длины относится к случаю 2, то оно является O-вложением наибольшей длины последовательности x и префикса $y[n-1]$ и имеет такую же μ-длину. Отсюда следует утверждение теоремы для случая $x_m \neq y_n$.

□

Теорема 21 определяет алгоритм вычисления функции $O_4(x, y) = lcO(x, y)$ сложности $\mathbf{O}(mn)$.

8. ЗАКЛЮЧЕНИЕ

Некоторые из введенных нами функций подобия играют вспомогательную роль. Например, L_3 используется для вычисления O_3, а lcL используется для вычисления lcO, A_0 имеет как самостоятельное значение, так и используется для вычисления A_1. Хотя могут быть приложения, в которых эти вспомогательные функции окажутся как раз основными. L-вложения полезны там, где важно не только расстояние между символами вложения, но и расстояние от последнего символа вложения до конца последовательности, соответственно, R-вложения полезны там, где важно расстояние от начала последовательности до первого символа вложения.

Сравнивая разные типы функций (независимо от типа вложения), можно отметить следующее. Число общих вложений (функция 0) является хорошей числовой характеристикой, но она не учитывает длины вложений. Например, последовательности 11112222 и 1122111 имеют 9 общих подпоследовательностей (включая пустую), а сумма их длин равна 17, в то же время первая последовательность имеет с последовательностью 22221111

тоже 9 общих подпоследовательностей, но сумма их длин равна 20 за счет того, что есть две длинные подпоследовательности 1111 и 2222. Поэтому сумма длин общих вложений (функция 1) имеет самостоятельное значение.

Обе эти характеристики (функции 0 и 1) не учитывают тот факт, что одно общее вложение может входить в одну последовательность много раз, а в другую мало. Это пытается учесть сумма числа пар вхождений общих вложений (сумма произведений числа вхождений общих вложений – функция 3). Для того же примера: последовательности 11112222 и 1122111 имеют 279 пар вхождений общих вложений, а последовательности 11112222 и 22221111 – 139 пар за счет того, что в первой паре последовательностей очень много вхождений в обе последовательности общих вложений 12, 112 и 122, отсутствующих для второй пары.

В то же время функция 3 не удовлетворяет естественной *аксиоме направленности (direction) сходства* [5], иначе называемой *ограниченность самоподобием (bounded by self-similarity)* [4]: $f(x, y) \le min\{f(x, x), f(y, y)\}$. В частности эта функция строго возрастает, когда одна и та же непустая последовательность x сравнивается с последовательностями xx, xxx, $xxxx$, ... Этот недостаток преодолевает функция 4 – сумма минимумов числа вхождений общих вложений. К сожалению, эта функция плохо поддается алгоритмической оптимизации, в частности, для A-вложений, т.е. подпоследовательностей, мы не знаем алгоритма, отличного от полного перебора. Частичную оптимизацию можно было бы провести на основе эффективного алгоритма перечисления общих подпоследовательностей сложности $C(x, y)$ меньшей, чем сложность полного перебора. Поскольку вычисление числа вхождений данного вложения u в данную последовательность x имеет сложность $\mathbf{O}(|u|*|x|)$ [3, лемма 8], мы имели бы алгоритм вычисления функции 4 сложности $C(x, y)*\mathbf{O}(mn)$. Также если бы был эффективный (возможно, на подклассе последовательностей) алгоритм перечисления подпоследовательностей только одной данной последовательности сложности $C_1(x, y)$, то мы имели бы алгоритм вычисления функции 4 сложности $C_1(x, y)*\mathbf{O}(mn)$.

Функция 5, основанная на наибольшей длине общего вложения, также удовлетворяет аксиоме направленности. Но у нее тот недостаток, что не учитываются общие вложения, не являющиеся частью наибольшего (по длине) общего вложения. В нашем примере последовательности 11112222 и 1122111 имеют наибольшую общую подпоследовательность 1122, частью которой не является подпоследовательность 111, а последовательности 11112222 и 22221111 имеют две наибольшие общие подпоследовательности: 1111, не учитывающую все подпоследовательности из двоек, и 2222, не учитывающую все подпоследовательности из единиц.

Проблема перечисления общих вложений также исследуется. В качестве примера можно привести работу [6], в которой предлагается алгоритм перечисления максимальных по вложенности (а не по длине) общих подпоследовательностей, которому требуется полиномиальные время и память на каждую найденную максимальную общую подпоследовательность. Такие максимальные общие вложения обладают тем полезным свойством, что каждое общее вложение является частью одного или нескольких максимальных вложений. Можно было бы предложить функцию подобия, основанную на максимальных по вложенности общих вложениях: число таких вложений, сумма их длин и др. Это могло бы стать темой дальнейших исследований.

Другим направлением дальнейших исследований могли бы стать функции подобия последовательностей в алфавите, в котором символам приписаны те или иные веса. Пустому символу можно было бы приписать нулевой вес. Соответственно, вложению соответствует не число входящих в него символов, а сумма их весов. Отрицательный вес мог бы означать “штраф” за использование такого символа в общем вложении.

СПИСОК ЛИТЕРАТУРЫ

1. *Wagner R., Fischer M.* The string-to-string correction problem // Journal of the ACM. 1974. V. 21. № 1. P. 168–173. https://dl.acm.org/doi/10.1145/321796.321811
2. *Wang H.* All common subsequences, in: M.M. Veloso (Ed.), IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 2007. https://www.aaai.org/Papers/IJCAI/2007/IJCAI07-101.pdf
3. *Cees Elzinga, Sven Rahmann, Hui Wang*: Algorithms for Subsequence Combinatorics. Theoretical Computer Science. 2008. V. 409. № 3. P. 394–404. https://doi.org/10.1016/j.tcs.2008.08.035
4. *Gilbert Ritschard and Matthias Studer* (*editors*). Proceedings of the International Conference on Sequence Analysis and Related Methods (LaCOSA II). Lausanne, Switzerland, June 8–10, 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. *Знаменский С.В.* Модель и аксиомы метрик сходства, Программные системы: теория и приложения. 2017. Т. 8. Вып. 4. С. 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. V. 84. P. 757–783. https://doi.org/10.1007/s00453-021-00898-5

TWENTY SIMILARITY FUNCTIONS OF TWO FINITE SEQUENCES

© 2023 г. I. Burdonov[a,#], and A. Maksimov[a,##]

[a] *Ivannikov Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia*

[#] *e-mail: igor@ispras.ru*

[##] *e-mail: andrew@ispras.ru*

The article discusses the various numerical functions that determine the degree of "similarity" of the two given final sequences. These similarity measures are based on the concept we define of embedding in a sequence. A special case of such an attachment is the usual sub-subsequence. Other cases further require equality of distances between adjacent sub-sequence symbols in both sequences. This is generalization of the concept of a sequence segment (substring) in which these distances are unit. In addition, equality of distances from the beginning of the sequences to the first embedding symbol or from the last embedding symbol to the end of the sequences may be required. Except these last two cases, the attachment can be in a sequence several times. The literature uses functions such as the number of common attachments or the number of attachment occurrence pairs in a sequence. In addition to them, we enter three more functions: the sum of the lengths of total investments, the sum of the minima of the number of occurrences of a common embedding in both sequences and the similarity function based on the largest number of symbols of the common embedding. In total, 20 numerical functions are considered, for 17 of which algorithms (including new ones) of polynomial complexity are proposed, for two more functions, algorithms have exponential complexity with reduced a measure of degree. The Conclusion gives a brief comparative description of these investments and functions.