О сложности преобразования пар слов относительно операций выпадения-вставки специального вида

П. С. Дергач[1] С. Р. Амирова[1]

Данная статья посвящена поиску расстояния между парами слов в общем конечном алфавите под действием операции замены одной буквы в две (соседние) и вычислению соответствующей кратчайшей цепочки замен (в случае ее существования). Изначально задача ставилась в более общей формулировке для пары регулярных языков, но позднее постановка задачи была уточнена. При этом рассмотрены две возможности - с разрешением замены ранее отсутствовавших в исходном слове букв или с запретом таких операций. Данное направление актуально и может быть использовано, например, в теории помехоустойчивого кодирования. В частности, стоит упомянуть метрику Левенштейна, вдохновляющую на аналогичные исследования относительно нового вида операций буквенной замены.

Ключевые слова: распознавание текстов, расстояние Левенштейна, метрика, оптимальный алгоритм.

1. Введение

В области распознавания текстов многие классические приложения используют меры расстояния для определения сходства между данными. В частности, используются расстояние Хэмминга, которое подсчитывает различные позиции между двумя словами одинаковой длины, и расстояние Левенштейна [1] (расстояние редактирования), которое вычисляет количество операций, необходимых для преобразования одного слова в другое. Расстояние редактирования — это обычная метрика, используемая для расчета сходства между двумя последовательностями, которая включает операции вставки, удаления и замены символов[2]. По произвольной паре слов необходимо уметь эффективно находить расстояние

[1] *Дергач Пётр Сергеевич* — к.ф.-м.н., м.н.с. каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: dergachpes@gmail.com.

Dergach Peter Sergeevich — Ph.D., junior researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

[1] *Амирова Сабина Ровшан гызы* — выпускник Филиала МГУ имени М. В. Ломоносова в городе Баку, e-mail: sabina.mgu@mail.ru.

Amirova Sabina Rovshan — Graduate of the M. V. Lomonosov Moscow State University Branch in Baku.

между ними. Эта проблема возникает в нескольких областях, включая обработку естественного языка и вычислительную биологию.

В этой статье будет решаться такая же проблема, но для другого вида операций: замена одной буквы на две. Ставится **основная задача**: по произвольной паре слов найти эффективный алгоритм, проверяющий возможность преобразования слов друг в друга, рассчитывающий минимальное достаточное количество операций и дающий возможность эффективно восстановить соответствующую цепочку преобразований.

2. Основные определения и результаты

Определение 1. Алфавит A - некоторое конечное непустое множество символов.

Определение 2. Слово в алфавите A - конечная последовательность символов данного алфавита.

Определение 3. Множество всех слов данного алфавита обозначаем A^*.

Определение 4. Длиной $|\alpha|$ слова α назовем количество символов в этом слове.

Определение 5. Пусть $\alpha, \beta \in A^*$. Говорим, что слово α получается из слова β применением операции *, если найдется представление

$$\alpha = \alpha_1 a \alpha_2,$$
$$\beta = \alpha_1 b_1 b_2 \alpha_2$$

для некоторых $a, b_1, b_2 \in A$, $\alpha_1, \alpha_2 \in A^*$.

Определение 6. Пусть $\alpha, \beta \in A^*$. Если слово α можно превратить в слово β с помощью конечного применения операций *, то обозначаем это через $\alpha \xrightarrow{*} \beta$.

Определение 7. Пусть $\alpha, \beta \in A^*$. Если слово α можно превратить в слово β с помощью 1 применения операции *, то обозначаем это через $\alpha \xrightarrow{*,1} \beta$.

Определение 8. Пусть $\alpha, \beta \in A^*$. Если слово α можно превратить в слово β с помощью конечного применения операций *, и при этом мы имеем возможность заменять только первоначальные буквы α, то обозначаем это через $\alpha \xrightarrow{**} \beta$ (это означает, что мы не можем заменять новые буквы еще раз).

Определение 9. Для произвольной пары слов $\alpha, \beta \in A^*$ обозначаем через $f_1(\alpha, \beta)$ минимальное количество операций *, которых достаточно, чтобы превратить α в β . Если это не возможно, то $f_1(\alpha, \beta) = \infty$.

Определение 10. Для произвольной пары слов $\alpha, \beta \in A^*$ обозначаем через $f_2(\alpha, \beta)$ минимальное количество операций * с дополнительным ограничением на невозможность замены уже добавленных букв, которого достаточно, чтобы превратить α в β. Если это не возможно, то $f_2(\alpha, \beta) = \infty$.

Замечание 1. Введенные функции f_1, f_2 не являются метриками, поскольку они не удовлетворяют ни условию симметричности, ни неравенству треугольника.

Замечание 2. Не для каждой пары слов значения $f_1(\alpha, \beta), f_2(\alpha, \beta)$ конечны. Например,

$$f_1(00, 111) = f_2(00, 111) = \infty.$$

Замечание 3. Приведем пример пары слов, иллюстрирующих различие f_1, f_2:

$$f_1(00, 11111) = 3.$$
$$f_2(00, 11111) = \infty.$$

Определение 11. ***Алгоритм Т.***

Пусть есть 2 слова α, $\beta \in A^*$:

$$\alpha = a_1 a_2 ... a_n,$$
$$\beta = b_1 b_2 ... b_m.$$

Пусть также известно, что длина α не превосходит длину β, то есть

$$n = |\alpha| \leq |\beta| = m.$$

Проверяем на равенство первые буквы слов.

а. Если $a_1 = b_1$, то отбрасываем по первой букве из каждого слова:

$$\alpha = a_1 \widehat{\alpha},$$
$$\beta = b_1 \widehat{\beta},$$
$$a_1 = b_1.$$

Если $\widehat{\alpha}, \widehat{\beta}$ не пусты, то запускаем алгоритм для новой пары $\widehat{\alpha}, \widehat{\beta}$.

Иначе алгоритм заканчивает работу.

Если $\widehat{\alpha}$ пусто, то алгоритм заканчивает работу со значением 1.

Если $\widehat{\alpha}$ не пусто, а $\widehat{\beta}$ пусто, то алгоритм заканчивает работу со значением 0.

b. Если $a_1 \neq b_1$ и $m \geq 2$, то отбрасываем 1 букву из первого слова и 2 буквы из второго слова:

$$\alpha = a_1\widehat{\alpha},$$
$$\beta = b_1b_2\widehat{\widehat{\beta}}.$$

Если $\widehat{\alpha}, \widehat{\widehat{\beta}}$ не пусты, то запускаем алгоритм для новой пары $\widehat{\alpha}, \widehat{\widehat{\beta}}$.

Иначе алгоритм заканчивает работу.

Если $\widehat{\alpha}$ пусто, то алгоритм заканчивает работу со значением 1.

Если $\widehat{\alpha}$ не пусто, а $\widehat{\widehat{\beta}}$ пусто, то алгоритм заканчивает работу со значением 0.

c. Если равенство $a_1 = b_1$ не выполняется и $m = 1$, алгоритм заканчивает работу со значением 0.

Определение 12. Если даны пара слов α, $\beta \in A^*$, причем $|\alpha| \leq |\beta|$, и алгоритм T закончил работу со значением 1, то говорим, что α влезает в β, и обозначим это через $\alpha \xrightarrow{T} \beta$. Иначе говорим, что α не влезает в β.

Теорема 1. *Существует линейный по времени выполнения алгоритм, эффективно вычисляющий по произвольной паре слов α, $\beta \in A^*$ значения $f_1(\alpha, \beta), f_2(\alpha, \beta)$, и, в случае их конечности, находящий последовательность соответствующих преобразований $\alpha \xrightarrow{*} \beta, \alpha \xrightarrow{**} \beta$.*

3. Вспомогательные утверждения

Лемма 1. *Пусть $\alpha, \beta \in A^*$, тогда:*

1) Если $\alpha \xrightarrow{T} \beta$, то $\alpha \xrightarrow{T} a\beta$,

2) Если $a\alpha \xrightarrow{T} \beta$, то $\alpha \xrightarrow{T} \beta$.

Доказательство. Обозначим

$$\alpha = a_1\widehat{\alpha},$$
$$\beta = b_1\widehat{\beta}.$$

Если при этом $\widehat{\beta}$ не пусто, то обозначим $\widehat{\beta} = b_2\widehat{\widehat{\beta}}$.

Сформулируем k-ое утверждение индукции – это утверждение леммы при ограничении $|\alpha| + |\beta| = k$. При k = 2 верно только первое свойство леммы (второе свойство в этом случае не выполнимо), при $k \geq 3$ оба свойства леммы верны.

База индукции: Проверим свойство 1 для случаев k=2, 3.

k = 2 и $|\alpha| + |\beta| = 2$. Значит $|\alpha| = |\beta| = 1$. То есть $\alpha = a_1$ и $\beta = b_1$. Так как $\alpha \xrightarrow{T} \beta$, то $a_1 = b_1$.

Значит $a\beta = ab_1$ и получаем $a_1 \xrightarrow{T} ab_1$.

Пусть теперь k = 3 и $|\alpha| + |\beta| = 3$. Тогда, так как $\alpha \xrightarrow{T} \beta$, то длины $|\alpha| = 1, |\beta| = 2$. Тогда $\alpha = a_1, \beta = b_1b_2$. В этом случае $a_1 \xrightarrow{T} b_1b_2$. Тогда верно и $a_1 \xrightarrow{T} ab_1b_2$.

Проверим свойство 2. k = 3 и $|\alpha| + |\beta| = 3$. Тогда, так как $a\alpha \xrightarrow{T} \beta$, то $|\alpha| = 1, |\beta| = 2$. Тогда $\alpha = a_1, \beta = b_1b_2$. Знаем, что $aa_1 \xrightarrow{T} b_1b_2$. Это возможно только в том случае, если $a = b_1, a_1 = b_2$. Тогда, очевидно, $a_1 \xrightarrow{T} b_1b_2$.

Переход индукции: Предположим, что мы доказали утверждение индукции для пар слов α и β с суммарной длиной не выше k.

Докажем утверждение индукции для пар слов α и β суммарной длиной k + 1.

Докажем первую часть леммы:

Случай 1:

$$a_1 = b_1,\ a_1 \neq a.$$

Тогда в результате применения алгоритма T для пары слов α, β произойдет замена $a_1 \xrightarrow{T} b_1$ и $\widehat{\alpha} \xrightarrow{T} \widehat{\beta}$.

Тогда для пары слов $\alpha, a\beta$ произойдет замена $a_1 \xrightarrow{T} ab_1$ и дальше уже факт, который знаем: $\widehat{\alpha} \xrightarrow{T} \widehat{\beta}$.

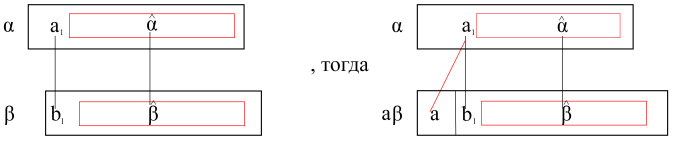

Рис. 1.

Замечание 4. Текст рассуждения здесь и далее не меняется в случае пустого $\widehat{\alpha}$. Далее про это отдельно говориться не будет.

Случай 2:

$$a_1 = b_1 = a$$

Для слов α, β произойдет замена $a_1 \xrightarrow{T} b_1$, и мы знаем, что $\widehat{\alpha} \xrightarrow{T} \widehat{\beta}$.

Так как $|\widehat{\alpha}| + |\widehat{\beta}| \leq k$, то по предположению индукции $\widehat{\alpha} \xrightarrow{T} b_1\widehat{\beta}$. И для слов $\alpha, a\beta$ произойдет замена $a_1 \xrightarrow{T} a$:

Случай 3:

$$a_1 \neq b_1,\ a_1 = a.$$

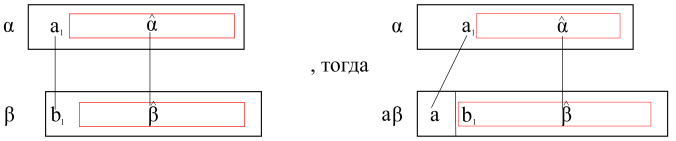

Рис. 2.

В слове β точно есть как минимум 2 символа b_1, b_2, в силу условия $|\alpha| + |\beta| = k$, $k \geq 3$.

Тогда, по нашему алгоритму, для слов α, β мы знаем, что $a_1 \xrightarrow{T} b_1 b_2$ и $\widehat{\alpha} \xrightarrow{T} \widehat{\widehat{\beta}}$.

Далее, для слов $\alpha, a\beta$ получаем $a_1 \xrightarrow{T} a$, и по предположению индукции 2 раза применяем первое свойство Леммы: $\widehat{\alpha} \xrightarrow{T} b_1 b_2 \widehat{\widehat{\beta}}$.

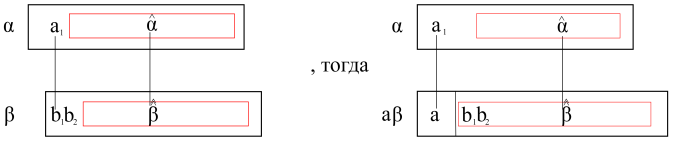

Рис. 3.

Случай 4:

$$a_1 \neq b_1,\ a_1 \neq a.$$

Тогда, по алгоритму для слов α, β, $a_1 \xrightarrow{T} b_1 b_2$, и $\widehat{\alpha} \xrightarrow{T} \widehat{\widehat{\beta}}$.

Далее, для слов $\alpha, a\beta$ получаем $a_1 \xrightarrow{T} a b_1$ и, по предположению индукции для первого свойства Леммы верно, что $\widehat{\alpha} \xrightarrow{T} b_2 \widehat{\widehat{\beta}}$.

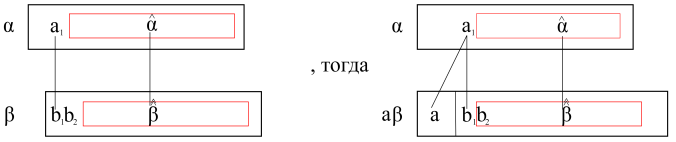

Рис. 4.

Теперь проверяем свойство 2.

Так как $a\alpha \xrightarrow{T} \beta$, то $m \geq 2$.

Случай 1:

$$a = b_1 = a_1$$

Для пары слов $a\alpha, \beta$ знаем, что $a \xrightarrow{T} b_1$, и, по алгоритму $a_1\widehat{\alpha} \xrightarrow{T} \widehat{\beta}$.

Тогда, по предположению индукции из второго свойства Леммы получаем $\widehat{\alpha} \xrightarrow{T} \widehat{\beta}$.

И, так как $a_1 \xrightarrow{T} b_1$ и $\widehat{\alpha} \xrightarrow{T} \widehat{\beta}$, то $\alpha \xrightarrow{T} \beta$.

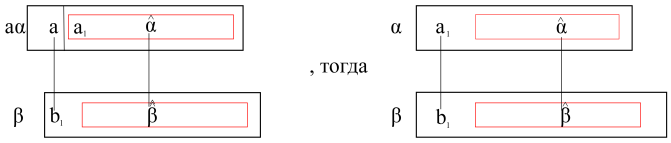

Рис. 5.

Случай 2:

$$a = b_1,\ a_1 \neq b_1.$$

Для пары слов $a\alpha, \beta$ знаем, что $a\alpha \xrightarrow{T} \beta$, и, по алгоритму $a \xrightarrow{T} b_1$ и $a_1\widehat{\alpha} \xrightarrow{T} b_2\widehat{\widehat{\beta}}$.

Для слов α, β заметим, что $a_1 \xrightarrow{T} b_1b_2$ и надо показать, что $\widehat{\alpha} \xrightarrow{T} \widehat{\widehat{\beta}}$.

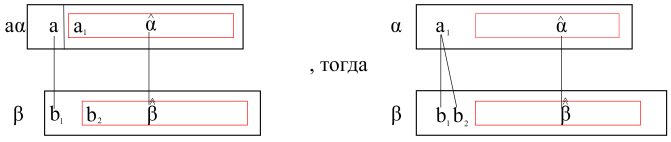

Рис. 6.

Разберем два подслучая:

Случай 2.1:

$$a_1 = b_2, \text{ тогда } \widehat{\alpha} \xrightarrow{T} \widehat{\widehat{\beta}}$$

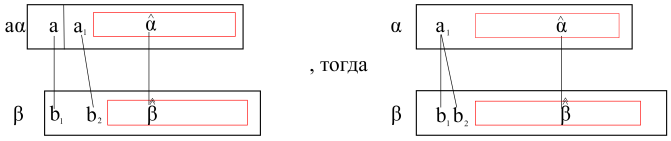

Рис. 7.

Случай 2.2

$a_1 \neq b_2$, тогда $m \geq 3$ и обозначим $\widehat{\widehat{\beta}} = b_3\widehat{\widehat{\beta}}$.

Тогда $\widehat{\alpha} \xrightarrow{T} \widehat{\widehat{\beta}}$. По первому свойству Леммы верно, что $\widehat{\alpha} \xrightarrow{T} b_3\widehat{\widehat{\beta}}$.

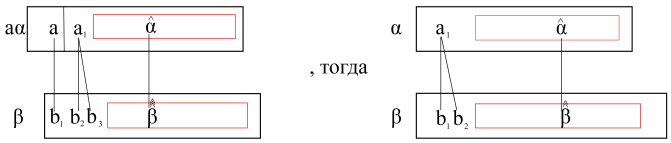

Рис. 8.

Случай 3:

$$a \neq b_1,\ a_1 = b_1.$$

Для слов $a\alpha, \beta$ знаем, что $a \xrightarrow{T} b_1b_2$, и $a_1\widehat{\alpha} \xrightarrow{T} \widehat{\widehat{\beta}}$, и по предположению индукции для второго свойства Леммы верно $\widehat{\alpha} \xrightarrow{T} \widehat{\widehat{\beta}}$.

Далее, для слов α, β получим $a_1 \xrightarrow{T} b_1$ и, по преположению индукции для первого свойства Леммы $\widehat{\alpha} \xrightarrow{T} b_2\widehat{\widehat{\beta}}$. Индукция по обоим свойствам.

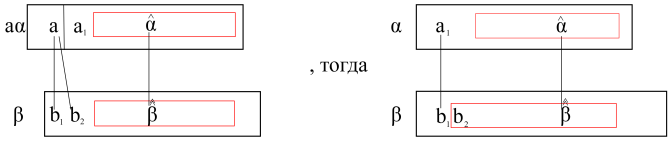

Рис. 9.

Случай 4:

$$a \neq b_1,\ b_1 \neq a_1.$$

Для слов $a\alpha, \beta$ знаем, что $a \xrightarrow{T} b_1b_2$ и $a_1\widehat{\alpha} \xrightarrow{T} \widehat{\widehat{\beta}}$.

Далее, для слов α, β имеем $a_1 \xrightarrow{T} b_1b_2$, тогда по предположению индукции для второго свойства леммы, $\widehat{\alpha} \xrightarrow{T} \widehat{\widehat{\beta}}$

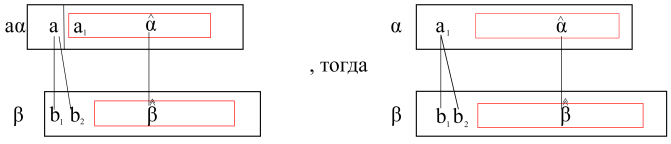

Рис. 10.

Утверждение леммы доказано. □

Лемма 2. *Пусть* $\alpha_1 \xrightarrow{T} \alpha_2$ *и* $\alpha_2 \xrightarrow{T} \alpha_3$, *тогда* $\alpha_1 \xrightarrow{T} \alpha_3$.

Доказательство. Будем доказывать утверждение индукцией по длине

$$|\alpha_1| + |\alpha_2| + |\alpha_3| = k,\ k \geq 3.$$

База индукции: k = 3, тогда все первые буквы одинаковы $a = a = a$, тогда, очевидно, $\alpha_1 \xrightarrow{T} \alpha_3$.

Переход индукции: пусть доказано утверждение для k, докажем его для k + 1.

Расммотрим 5 случаев:

1. Первые буквы попарно равны.
2. Первые буквы α_1 и α_2 равны, но не равны первой букве α_3;
3. Первые буквы α_2 и α_3 равны, но не равны первой букве α_1;
4. Первые буквы α_1 и α_3 равны, но не равны первой букве α_2;
5. Первые буквы попарно не равны.

Случай 1: Первые буквы попарно равны, то есть

$$\alpha_1 = a\widehat{\alpha_1},$$
$$\alpha_2 = a\widehat{\alpha_2},$$
$$\alpha_3 = a\widehat{\alpha_3}.$$

Тогда, так как $\alpha_1 \xrightarrow{T} \alpha_2$, то $\widehat{\alpha_1} \xrightarrow{T} \widehat{\alpha_2}$.

Так как $\alpha_2 \xrightarrow{T} \alpha_3$, то $\widehat{\alpha_2} \xrightarrow{T} \widehat{\alpha_3}$.

Заметим, что $a \xrightarrow{T} a$, тогда по предположению индукции верно, что $\widehat{\alpha_1} \xrightarrow{T} \widehat{\alpha_3}$. Отсюда окончательно получаем $\alpha_1 \xrightarrow{T} \alpha_3$.

Случай 2: Первые буквы α_1 и α_2 равны, но не равны первой букве α_3:

$$\alpha_1 = a\widehat{\alpha_1},$$
$$\alpha_2 = a\widehat{\alpha_2},$$
$$\alpha_3 = \bar{a}\widehat{\alpha_3}.$$

Так как $\alpha_2 \xrightarrow{T} \alpha_3$, а $a \neq \bar{a}$, то $|\alpha_3| \geq 2$:

$$\alpha_3 = \bar{a}b\widehat{\widehat{\alpha_3}}.$$

Так как $\alpha_1 \xrightarrow{T} \alpha_2$, то $\widehat{\alpha_1} \xrightarrow{T} \widehat{\alpha_2}$.

Так как $\alpha_2 \xrightarrow{T} \alpha_3$, то $\widehat{\alpha_2} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$.

Тогда, по предположению индукции $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$.

Заметим, что $a \xrightarrow{T} \bar{a}b$ и $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$, значит верно, что $\alpha_1 \xrightarrow{T} \alpha_3$.

Случай 3: Первые буквы α_2 и α_3 равны, но не равны первой букве α_1:

$$\alpha_1 = a\widehat{\alpha_1},$$
$$\alpha_2 = \bar{a}\widehat{\alpha_2},$$
$$\alpha_3 = \bar{a}\widehat{\alpha_3}.$$
Так как $\alpha_1 \xrightarrow{T} \alpha_2$, а $a \neq \bar{a}$, то $|\alpha_2| \geq 2$:
$$\alpha_2 = \bar{a}b\widehat{\widehat{\alpha_2}},$$
Так как $\alpha_2 \xrightarrow{T} \alpha_3$, то $|\alpha_3| \geq 2$:
$$\alpha_3 = \bar{a}c\widehat{\widehat{\alpha_3}}.$$
Так как $\alpha_1 \xrightarrow{T} \alpha_2$, то $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_2}}$.
Так как $\alpha_2 \xrightarrow{T} \alpha_3$, то $b\widehat{\widehat{\alpha_2}} \xrightarrow{T} c\widehat{\widehat{\alpha_3}}$.

Случай 3.1: $b = c$, получим $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_2}}$ и $\widehat{\widehat{\alpha_2}} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$. Значит, по предположению индукции, $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$.

Так как $a \xrightarrow{T} \bar{a}c$ и $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$, то $\alpha_1 \xrightarrow{T} \alpha_3$.

Случай 3.2: $b \neq c$, тогда $|\alpha_3| \geq 3$ и обозначим $\alpha_3 = \bar{a}cd\widehat{\widehat{\alpha_3}}$.

$\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_2}}$ и $\widehat{\widehat{\alpha_2}} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$. Получили $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$, и из первого свойства Леммы 1 верно $\widehat{\alpha_1} \xrightarrow{T} d\widehat{\widehat{\alpha_3}}$.

Так как $a \xrightarrow{T} \bar{a}c$ и $\widehat{\alpha_1} \xrightarrow{T} d\widehat{\widehat{\alpha_3}}$, то $\alpha_1 \xrightarrow{T} \alpha_3$.

Случай 4: Первые буквы α_1 и α_3 равны, но не равны первой букве α_2:

$$\alpha_1 = a\widehat{\alpha_1},$$
$$\alpha_2 = \bar{a}\widehat{\alpha_2},$$
$$\alpha_3 = a\widehat{\alpha_3}.$$
Так как $\alpha_1 \xrightarrow{T} \alpha_2$, а $a \neq \bar{a}$, то $|\alpha_2| \geq 2$:
$$\alpha_2 = \bar{a}b\widehat{\widehat{\alpha_2}},$$
Так как $\alpha_2 \xrightarrow{T} \alpha_3$, то $|\alpha_3| \geq 2$:
$$\alpha_3 = ad\widehat{\widehat{\alpha_3}}.$$

Так как $\alpha_1 \xrightarrow{T} \alpha_2$, то $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_2}}$.

Так как $\alpha_2 \xrightarrow{T} \alpha_3$, то $b\widehat{\widehat{\alpha_2}} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$. По второму свойству Леммы 1 получим, что $\widehat{\widehat{\alpha_2}} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$.

И, по предположению индукции, $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$. Тогда, по первому свойству Леммы 1 получим, что $\widehat{\alpha_1} \xrightarrow{T} d\widehat{\widehat{\alpha_3}}$.

Так как $a = a$ и $\widehat{\alpha_1} \xrightarrow{T} d\widehat{\widehat{\alpha_3}}$, то $\alpha_1 \xrightarrow{T} \alpha_3$.

Случай 5: Первые буквы попарно не равны:

$$\alpha_1 = a\widehat{\alpha_1},$$
$$\alpha_2 = \bar{a}\widehat{\alpha_2},$$
$$\alpha_3 = c\widehat{\alpha_3}.$$

Так как $\alpha_1 \xrightarrow{T} \alpha_2$, а $a \neq \bar{a}$, то $|\alpha_2| \geq 2$:

$$\alpha_2 = \bar{a}b\widehat{\widehat{\alpha_2}},$$

Так как $\alpha_2 \xrightarrow{T} \alpha_3$, а $\bar{a} \neq c$, то $|\alpha_3| \geq 2$:

$$\alpha_3 = cd\widehat{\widehat{\alpha_3}}.$$

Так как $\alpha_1 \xrightarrow{T} \alpha_2$, то $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_2}}$.

Так как $\alpha_2 \xrightarrow{T} \alpha_3$, то $b\widehat{\widehat{\alpha_2}} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$. По второму свойству Леммы 1 получим, что $\widehat{\widehat{\alpha_2}} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$.

И, по предположению индукции $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$.

Так как $a \xrightarrow{T} cd$ и $\widehat{\alpha_1} \xrightarrow{T} \widehat{\widehat{\alpha_3}}$, то $\alpha_1 \xrightarrow{T} \alpha_3$.

Утверждение леммы доказано. □

Лемма 3. *Если верно* $\alpha_1 \xrightarrow{*,1} \alpha_2$*, то верно и* $\alpha_1 \xrightarrow{T} \alpha_2$.

Доказательство. Пусть

$$\alpha_1 = \gamma_1 a \gamma_2,$$
$$\alpha_2 = \gamma_1 bc \gamma_2.$$
$$\gamma_1, \gamma_2 \in A^*$$

Алгоритм идет по началу.

1. Первый случай:

$$\gamma_1 = \gamma_1, \text{ тогда } \gamma_1 \xrightarrow{T} \gamma_1,$$
$$\text{Если a} \neq \text{b, тогда } a \xrightarrow{T} bc,$$
$$\gamma_2 = \gamma_2, \text{ тогда } \gamma_2 \xrightarrow{T} \gamma_2.$$

2. Второй случай:

$$\gamma_1 = \gamma_1, \text{ тогда } \gamma_1 \xrightarrow{T} \gamma_1,$$
$$\text{a =b, тогда } a \xrightarrow{T} b.$$

$\gamma_2 \xrightarrow{T} c\gamma_2$ - второе свойство Леммы 1, так как $\gamma_2 \xrightarrow{T} \gamma_2$.

Утверждение леммы доказано. □

Лемма 4. *Пусть* $\alpha, \beta \in A^*$, $|\alpha| = n, |\beta| = m$ *и* $n \leq m \leq 2n$.
Тогда $\alpha \xrightarrow{T} \beta$ *равносильно* $\alpha \xrightarrow{*} \beta$.

Доказательство. Если $\alpha \xrightarrow{T} \beta$, то алгоритм заканчивает работу и α может перейти в начало β. То есть $\beta = \beta_1\beta_2$, причем $\alpha \xrightarrow{*} \beta_1$. Так как операция * позволяет нам заменять уже замененные буквы α еще раз в буквы β, то последнюю букву α заменяем в две новые пока β не закончится, то есть $\beta_1 \xrightarrow{*} \beta_1\beta_2$. Таким образом $\alpha \xrightarrow{*} \beta$.

Докажем в обратную сторону. Пусть $\alpha \xrightarrow{*} \beta$, тогда существует такая последовательность замен, которая слово α превращает в слово β.

Возникает цепочка шагов $\alpha = \alpha_1 \xrightarrow{*,1} \alpha_2 \xrightarrow{*,1} ... \xrightarrow{*,1} \alpha_n = \beta$.

По Лемме 3 мы знаем, что если верно $\alpha_1 \xrightarrow{*,1} \alpha_2$, то верно и $\alpha_1 \xrightarrow{T} \alpha_2$.

Значит $\alpha_1 \xrightarrow{T} \alpha_2$. Также для $\alpha_2 \xrightarrow{T} \alpha_3$ и остальных пар. Значит, по Лемме 2 $\alpha_1 \xrightarrow{T} \alpha_n$, то есть $\alpha \xrightarrow{T} \beta$.

Утверждение леммы доказано. □

Лемма 5. *Пусть* $\alpha, \beta \in A^*$, $|\alpha| = n$, $|\beta| = m$, $m \geq n$.

$$\textit{Если } \alpha \xrightarrow{T} \beta \textit{ или } m > 2n, \textit{ то } f_1(\alpha, \beta) = m - n.$$
$$\textit{Иначе } f_1(\alpha, \beta) = \infty.$$

Доказательство. Если $\alpha \xrightarrow{T} \beta$ и $m \leq 2n$, то по Лемме 4 есть цепочка преобразований $\alpha \xrightarrow{*} \beta$. Таким образом, $f_1(\alpha, \beta)$ конечно и равно $m - n$, так как каждая операция увеличивает длину на единицу.

Если $m > 2n$, то каждую букву α заменяем в две буквы β, а потом, так как операция * позволяет нам заменять уже замененные буквы еще раз, то последнюю букву заменяем на 2 новые пока β не закончится. Таким образом, $f_1(\alpha, \beta)$ конечно и равно $m - n$.

Если не верно, что $\alpha \xrightarrow{T} \beta$, но $m \leq 2n$, то по Лемме 4 неверно $\alpha \xrightarrow{*} \beta$. Таким образом, $f_1(\alpha, \beta) = \infty$. □

Лемма 6. *Пусть* $\alpha, \beta \in A^*$, $|\alpha| = n$, $|\beta| = m$, $m \geq n$.

$$\textit{Если } \alpha \xrightarrow{T} \beta \textit{ и } 2n \geq m, \textit{ то } f_2(\alpha, \beta) = m - n.$$
$$\textit{Если } m > 2n, \textit{ либо не верно, что } \alpha \xrightarrow{T} \beta, \textit{ то } f_2(\alpha, \beta) = \infty.$$

Доказательство. Если $\alpha \xrightarrow{T} \beta$ и $m \leq 2n$, то верно, что $\beta = \beta_1\beta_2$, где $\alpha \xrightarrow{**} \beta_1$. Разделим буквы слова α на два типа: те, которые мы не меняли при замене, и те, которые заменяли одну в две. Пусть букв, которых мы не меняли при замене k штук. Тогда $|\beta_1| = 2(n - k) + k$. Тогда в β_2 осталось $m - (2(n - k) + k) = m - 2n + k$ букв. Тогда, так как $m \leq 2n$, то $m - 2n + k \leq k$.

Далее заменяем последние $m - 2n + k$ нетронутых букв α по одной в две буквы β, начиная с крайней правой, и получаем соответствующую

цепочку замен $\alpha \xrightarrow{**} \beta$. Количество операций при этом, очевидно, равно $m - n$.

Если $m \leq 2n$ и не верно, что $\alpha \xrightarrow{T} \beta$, то, по Лемме 4, не верно $\alpha \xrightarrow{*} \beta$, тем более не верно $\alpha \xrightarrow{**} \beta$. И, значит, $f_2(\alpha, \beta) = \infty$.

Если $m > 2n$, то нам не хватит букв α, так как даже если мы будем заменять каждую букву α в две буквы β, то максимальная длина β может быть только 2n.

Поэтому $f_2(\alpha, \beta) = \infty$.

Утверждение леммы доказано. □

4. Доказательство основных утверждений

Теорема 1. *Существует линейный по времени выполнения алгоритм, эффективно вычисляющий по произвольной паре слов α, $\beta \in A^*$ значения $f_1(\alpha, \beta), f_2(\alpha, \beta)$, и, в случае их конечности, находящий последовательность соответствующих преобразований $\alpha \xrightarrow{*} \beta, \alpha \xrightarrow{**} \beta$.*

Доказательство. Пусть $|\alpha| = n, |\beta| = m$. Без ограничения общности считаем, что $m \leq n$.

Если $m \leq 2n$, то запускаем Алгоритм T.

Если Алгоритм T закончился со значением 1, то по Леммам 5 и 6 верно, что значения $f_1(\alpha, \beta) = f_2(\alpha, \beta) = m - n$. Соответствующая цепочка преобразований строится из доказательств Лемм 5 и 6.

Если Алгоритм T закончился со значением 0, то по Леммам 5 и 6 верно, что значения $f_1(\alpha, \beta) = f_2(\alpha, \beta) = \infty$. Соответствующей цепочки преобразований нет в силу доказательств Лемм 5 и 6.

Если $m > 2n$, то по Лемме 5 верно $f_1(\alpha, \beta) = m - n$, и из доказательства Леммы 5 получим соответствующую цепочку преобразований. А из доказательства Леммы 6 получаем $f_2(\alpha, \beta) = \infty$.

Утверждение теоремы доказано. □

5. Заключение

Приведем основные результаты данной статьи:

- Была исследована тема метрической близости языков и ее применение для исправления ошибок в словах, в частности, замены одной буквы на две.
- Была выполнена основная задача: предложен отптимательный по времени работы алгоритм, позволяющий по произвольной паре

слов $\alpha, \beta \in A^*$ проверить возможность преобразования одного слова в другое нашими операциями, вычисляющий минимальное достаточное количество шагов и эффективно восстанавливающий соответствующую последовательность преобразований.

В результате исследования были выявлены возможные направления для дальнейшей работы в этой области.

В частности, можно рассмотреть расширение алгоритма для других вариантов замен, а также его применение не к паре слов, а к паре языков.

Список литературы

[1] В.И.Левенштейн., "Двоичные коды с исправлением выпадений и вставок символа. Проблема передачи информации", *Докл. АН СССР*, 1965, 845 - 848.

[2] Томас Кормен, Чарльз Лейзерсон, Рональд Ривест, Клиффорд Штайн, "Алгоритмы. Построение и анализ", *Второе издание*, 2011.

[3] Б.Д. Кудряшов, "Основы теории кодирования", *Учебное пособие*, 2016.

[4] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, "Introduction to Automata Theory, Languages, and Computation", *3rd edition*, 2007.

[5] Austin J. Parker, Kelly B. Yancey, and Matthew P. Yancey, "egular Language Distance and Entropy", 2017.

On the complexity of converting pairs of words with respect to drop-in operations of a special type

Dergach P.S., Amirova S.R.

This article is devoted to finding the distance between pairs of words in a general finite alphabet under the action of the operation of replacing one letter into two (adjacent) and calculating the corresponding shortest chain of substitutions (if it exists). Initially, the problem was posed in a more general formulation for a pair of regular languages, but later the formulation of the problem was clarified. At the same time, two possibilities are considered - with the permission to replace letters that were previously absent in the original word or with the prohibition of such operations. This direction is relevant and can be used, for example, in the theory of noise-resistant coding. In particular, it is worth mentioning the Levenstein metric, which inspires similar research on a new type of letter substitution operations.

Keywords: text recognition, Levenshtein distance, metric, optimal algorithm.

References

[1] levenshtein V.I., “Binary codes capable of correcting deletions, insertions, and reversals”, 1965, 845 – 848.

[2] Thomas H.Cormen, Charles E.Leiserson, Ronald L.Rivest, Clifford Stein., “Introduction to Algorithms”, 2011.

[3] Kudryavtsev V.B., “Correction Coding Theory”, 2016.

[4] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, “Introduction to Automata Theory, Languages, and Computation”, 2007.

[5] Austin J. Parker, Kelly B. Yancey, and Matthew P. Yancey, “Regular Language Distance and Entropy”, 2017.