УДК 519.16

Ю. С. Окуловский

О ПОИСКЕ ВЛОЖЕНИЙ СЛОВ ЗИМИНА В ПРОИЗВОЛЬНЫЕ СЛОВА

1. Введение

Основной задачей комбинаторики слов является поиск регулярных структур в словах. В [1] было предложено использовать в одной из таких структур *шаблон* (точное определение дано ниже). В частности, в [1] указан алгоритм распознавания *неизбежных* шаблонов, т. е. таких, которые могут быть найдены в любом достаточно длинном слове. Независимо в [2] построена бесконечная серия слов (*слова Зимина*), которые не только являются неизбежными сами, но и содержат в качестве шаблонов все остальные неизбежные слова. Поскольку слова Зимина неизбежны, то естественно возникает вопрос об эффективном поиске фрагментов слова, соответствующих шаблону слова Зимина. Основным результатом данной работы является эффективный алгоритм такого поиска.

Данная задача естественно дополняет хорошо известную задачу о поиске в слове Зимина заданного шаблона. Последняя задача эквивалентна проверке шаблона на неизбежность, и в настоящее время предполагается (но не доказано), что эта проверка NP-полна (см., например, [3]).

2. Определения и обозначения

Конечным алфавитом $\Sigma = \{a_1, \ldots, a_n\}$ называется непустое конечное множество, $a_1, \ldots, a_n$ – буквы. Словом над алфавитом Σ называется произвольная конечная последовательность букв. Длина слова s, т. е. количество букв в нем, обозначается как $|s|$. Множество непустых слов над алфавитом Σ обозначается как Σ^+. На множестве слов вводится бинарная ассоциативная операция конкатенации: если $u = a_1 a_2 \ldots a_n$, $v = b_1 b_2 \ldots b_m$, то $u \cdot v = a_1 a_2 \ldots a_n b_1 b_2 \ldots b_m$. Пара $(\Sigma^+, \cdot)$ является свободной полугруппой.

Кроме основного алфавита Σ будем рассматривать вспомогательный счетный алфавит $\Delta = \{x_1, \ldots, x_n, \ldots\}$. Слова над алфавитом Δ будем называть *шаблонами*.

Под гомоморфизмом свободных полугрупп далее всюду понимается отображение, сохраняющее конкатенацию.

Слова Зимина [2] над алфавитом Δ рекуррентно задаются следующими выражениями:

$$Z_1 = x_1,$$
$$Z_{k+1} = Z_k \cdot x_{k+1} \cdot Z_k.$$

Шаблон u *вкладывается* в слово s, если $s = v \cdot \varphi(u) \cdot w$ для некоторого гомоморфизма φ и (возможно, пустых) слов v, w. *Неизбежность* шаблона u означает, что для любого Σ существует N_Σ такое, что для любого $s \in \Sigma^+$ из того, что $|s| > N_\Sigma$, следует, что шаблон u вкладывается в s. Для слов Зимина это условие может быть сформулировано так: для любых Σ, k существует $N_{\Sigma,k}$ такое, что шаблон Z_k вкладывается в любое слово длины большей, чем $N_{\Sigma,k}$. Подробнее см. [4].

Задача, алгоритм решения которой изложен в этой статье, состоит в поиске по данному слову s пары k, φ такой, что:

1) существуют слова u, v такие, что $s = u \cdot \varphi(Z_k) \cdot v$;

2) k – максимальное число, при котором выполняется 1.

3. Матрицы совпадений и их свойства

Далее в статье под матрицей мы всегда будем понимать квадратную булеву матрицу. Матрицы будем обозначать прописными латинскими буквами, а их элементы – строчными буквами с индексами.

Пусть s – произвольное слово над алфавитом Σ. *Матрицей совпадений* слова s будем называть $|s| \times |s|$-матрицу $P(s)$ такую, что $p_{ij} = 1$, если i-я и j-я буквы слова s совпадают, и $p_{ij} = 0$ иначе. В этом параграфе мы покажем, как по свойствам матриц совпадения указать максимальное k такое, что слово s является гомоморфным образом Z_k. На основании этого результата будет построен алгоритм, решающий поставленную задачу.

Следующие свойства матриц совпадения тривиально вытекают из рефлексивности, симметричности и транзитивности отношения равенства:

$$\forall\, i \quad p_{ii} = 1, \tag{1}$$

$$\forall\, i, j \quad p_{ij} = p_{ji}, \tag{2}$$

$$\forall\, i, j, k \quad (p_{jk} = 1 \Rightarrow p_{ij} = p_{ik}). \tag{3}$$

Заметим, что всякая квадратная булева матрица со свойствами (1)–(3) определяет единственное, с точностью до переименования букв, слово.

Определим по индукции семейство множеств матриц $\{C^i\}$, $i \in \mathbb{N}$. C^1 есть множество матриц, все диагональные элементы которых равны 1. Матрица D принадлежит множеству C^k тогда и только тогда, когда она представима в виде

$$D = \left(\begin{array}{c|c|c} B & \dots & B \\ \hline \dots & A & \dots \\ \hline B & \dots & B \end{array}\right),$$

где $A \in C^1$, $B \in C^{k-1}$.

Лемма 3.1. *Для любых $s \in \Sigma^*$, $k \in N$*

$$s = \varphi(Z_k) \quad \Leftrightarrow \quad P(s) \in C^k.$$

Доказательство. Докажем эквивалентность индукцией по k.

База индукции. Любое слово является гомоморфным образом одной буквы, и матрица любого слова, согласно (1), принадлежит классу C^1, т. е. эквивалентность очевидна.

Шаг индукции. Необходимость. Пусть $s = \varphi(Z^k)$ для некоторого морфизма φ. Тогда $s = \varphi(Z^{k-1}) \cdot \varphi(x_k) \cdot \varphi(Z^{k-1})$ и матрица $P(s)$ имеет вид:

$$P(s) = \left(\begin{array}{c|c|c} P(\varphi(Z_{k-1})) & \dots & P(\varphi(Z_{k-1})) \\ \hline \dots & P(\varphi(x_k)) & \dots \\ \hline P(\varphi(Z_{k-1})) & \dots & P(\varphi(Z_{k-1})) \end{array}\right).$$

По предположению индукции, $P(\varphi(Z_{k-1})) \in C^{k-1}$, а $P(\varphi(x_k)) \in C^1$ по свойству (1), следовательно, $P(s) \in C^k$.

Достаточность. Пусть $P(s) \in C^k$. Следовательно, $P(s)$ представима в виде

$$P(s) = \left(\begin{array}{c|c|c} B & \dots & B \\ \hline \dots & A & \dots \\ \hline B & \dots & B \end{array}\right),$$

где $B \in C^{k-1}$, $A \in C^1$. Пусть размер матрицы B равен n_B, размер матрицы A есть n_A. Представим тогда s в виде $s = u \cdot v \cdot u$, где $|u| = n_B$, $|v| = n_A$. Очевидно в таком случае, что $P(u) = B \in C^{k-1}$, а следовательно, по предположению индукции, существует морфизм φ такой, что $u = \varphi(Z_{k-1})$. Доопределим φ, положив $\varphi(x_k) = v$. Тогда

$$s = u \cdot v \cdot u = \varphi(Z_{k-1}) \cdot \varphi(x_k) \cdot \varphi(Z_{k-1}) = \varphi(Z_{k-1} \cdot x_k \cdot Z_{k-1}) = \varphi(Z_k),$$

что и требовалось доказать.

Лемма 3.2. *Пусть P – матрица со свойствами* (1)–(3), $I = \{i_1, \ldots, i_k\}$, $J = \{j_1, \ldots, j_k\}$ *– множества индексов, L – $k \times k$-подматрица в P такая, что $\ell_{xy} = p_{i_x j_y}$, и пусть $L \in C^1$. Тогда L удовлетворяет свойствам* (1)–(3).

Доказательство. Свойство (1) очевидно выполняется, поскольку $L \in C^1$. Для проверки свойств (2) и (3) докажем следующую цепочку равенств:

$$p_{i_x j_y} = p_{i_x i_y} = p_{j_x i_y} = p_{j_x j_y}. \tag{4}$$

В самом деле, $p_{i_y j_y} = \ell_{yy} = 1$, так как $L \in C^1$; тогда $p_{j_y i_y} = 1$ по свойству (2) и $p_{i_x j_y} = p_{i_x i_y}$ по свойству (3). Остальные два равенства доказываются полностью аналогично. Используя (4), (2) и определение L, получаем

$$\ell_{xy} = p_{i_x j_y} = p_{j_x i_y} = p_{i_y j_x} = \ell_{yx},$$

т. е. L удовлетворяет (2). Далее, из $\ell_{yz} = 1$ по определению следует, что $p_{i_y j_z} = 1$, откуда $p_{j_y j_z} = 1$ согласно (4). Из свойства (3) матрицы P следует, что $p_{i_x j_y} = p_{i_x j_z}$, т. е. $\ell_{xy} = \ell_{xz}$. Следовательно, L удовлетворяет (3).

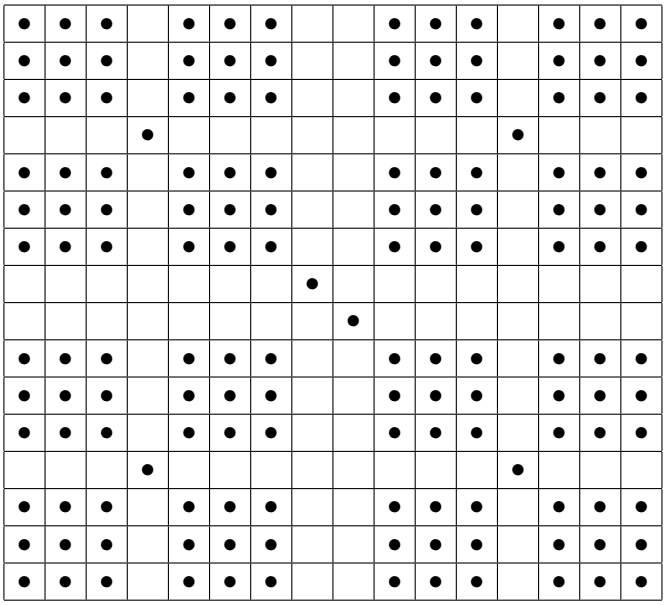
Лемма 3.3. *Пусть P – $n \times n$-матрица со свойствами* (1)–(3), $k < n/2$, $I = (1, \ldots, k)$, $J = (n{-}k{+}1, \ldots, n)$, *$L$ – $k \times k$-подматрица в P такая, что $\ell_{xy} = p_{i_x j_y}$ и $L \in C^k$. Тогда $P \in C^{k+1}$.*

Доказательство. Пусть A, B, D – $k \times k$-подматрицы в P, определяемые соответственно индексными множествами (I, I), (J, J) и (J, I). Тогда P имеет вид:

$$P(s) = \left(\begin{array}{c|c|c} A & \ldots & D \\ \hline \ldots & R & \ldots \\ \hline L & \ldots & B \end{array}\right).$$

Из симметричности P следует $D = L^t$, но L симметрична в силу свойства (2), которое распространено на нее по лемме 3.2. Это значит, что $D = L$. Поскольку $\ell_{xy} = p_{i_x j_y} = p_{i_x i_y} = a_{xy}$ согласно (4), получаем $A = L$ и, аналогично, $B = L$. Тем самым «по углам» матрицы P расположены четыре одинаковые подматрицы из C^k, а матрица $R \in C^1$, ведь ее диагональ совпадает с диагональю P. Отсюда получаем $P \in C^{k+1}$ по определению.

Возьмем матрицу со свойствами (1)–(3) и рассмотрим ее диагонали, параллельные главной диагонали. Диагонали, полностью заполненные единицами, мы будем называть единичными. Построим для примера матрицу совпадений слова $s = aaabaaacdaaabaaa$ (см. рис.). Чтобы облегчить восприятие, не будем писать нули в ячейках, а вместо единиц выставим маркеры. Единичные диагонали в этом случае имеют длины 1, 2, 3, 7 и 16.



Матрица совпадений слова *aaabaaacdaaabaaa*

Теорема 3.1. *Матрица P принадлежит множеству C^k тогда и только тогда, когда в P можно выбрать единичные диагонали с длинами $d_1, \ldots, d_k$ так, что $d_i < d_{i+1}/2$ для всех i.*

Доказательство. Необходимость непосредственно следует из леммы 3.1. Достаточность следует из леммы 3.3. В самом деле, рассмотрим d_1 и d_2. Поскольку $d_1 < d_2/2$, то $d_2 \times d_2$-подматрица в нижнем левом углу находится в условиях леммы 3.3, а значит, принадлежит к C^2. В свою очередь, поскольку $d_2 < d_3/2$, то угловая $d_3 \times d_3$-подматрица находится в условиях леммы и принадлежит к C^3 и т. д.

4. Алгоритм поиска образа слова Зимина

4.1. Основной алгоритм

Напомним, что задача состоит в том, чтобы по слову s определить максимальное k и морфизм φ такие, что $s = u \cdot \varphi(Z_k) \cdot v$. В частном случае $s = \varphi(Z_k)$ задача может быть решена следующим алгоритмом.

Алгоритм 1. Вход: s. Выход: k, φ.

Шаг 1. Найти максимальный по мощности набор единичных диагоналей матрицы $P(s)$, удовлетворяющих условию теоремы 3.1.

Шаг 2. Если $(d_1, \ldots, d_r)$ – длины диагоналей найденного набора, то $k = r$.

Шаг 3. Найти морфизм φ из условий $|\varphi(Z_i)| = d_i$, $i = 1, \ldots, k$.

Корректность шага 2 следует из леммы 3.1 и теоремы 3.1, а условие на шаге 3 определяется тем, что d_i – это размер подматрицы из множества C^i в матрице $P(s)$.

Предложение 4.1. *Шаг 1 корректно реализуется жадным алгоритмом.*

Доказательство. Рассмотрим выполнение шага 1 жадным алгоритмом: начиная с главной диагонали, в качестве следующей каждый раз выбирается ближайшая единичная, удовлетворяющая требуемому неравенству (для того чтобы получить числа $d_1, \ldots, d_k$, найденные диагонали пишутся в возрастающем порядке, но для доказательства корректности жадного алгоритма нам будет удобнее считать, что диагонали упорядочены по убыванию – эту последовательность обозначим как $f_1, \ldots, f_k$).

Пусть $f_1, \ldots, f_k$ – это диагонали, найденные жадным алгоритмом, а $g_1, \ldots, g_m$ – диагонали, выбранные другим алгоритмом (и также упорядоченные по убыванию). Покажем по индукции, что $f_i \geq g_i$ для всех допустимых значений i. Отметим, что для $i = 1$ это выполняется.

Далее, $g_{i+1} < g_i/2 \leq f_i/2$, а это значит, что на $(i+1)$-м шаге жадный алгоритм может выбрать g_{i+1}, если нет более близкой (а значит, и более длинной) единичной диагонали. Отсюда $g_i \leq f_i$. Поскольку диагонали $g_1, \ldots, g_m$ не длиннее, чем диагонали $f_1, \ldots, f_k$, то и число их не больше, а значит, $k \geq m$, что и требовалось доказать.

Предложение 4.2. *Сложность алгоритма 1 составляет* $O(|s|^2)$.

Доказательство. В самом деле, каждая диагональ матрицы совпадений просматривается не более одного раза, а это значит, что в худшем случае придется посчитать лишь всю матрицу совпадения. Поэтому алгоритм работает за время, не превышающее $O(|s|^2)$.

Примечание. Для работы алгоритму не требуется сама матрица слова, а значит, ее не нужно хранить. Поиск единичных диагоналей в $P(s)$ производится непосредственным сравнением соответствующих букв слова s.

Продемонстрируем работу алгоритма на слове $s = aaabaaacdaaabaaa$. Его матрица совпадений приведена на рисунке. Жадный алгоритм выберет диагонали с длинами 16, 7, 3. После этого он уже не может выбрать диагональ длины 2, поэтому остается последняя диагональ длины 1. Длины выбранных

диагоналей, таким образом, – 1, 3, 7, 16. Следовательно, в s вкладывается слово Z_4. Применяя третий шаг, находим гомоморфизм:

$$\begin{array}{lll} |\varphi(Z_1)| = 1 & Z_1 = x_1 & \varphi(x_1) = a \\ |\varphi(Z_2)| = 3 & Z_2 = Z_1 x_2 Z_1 & \varphi(x_2) = a \\ |\varphi(Z_3)| = 7 & Z_3 = Z_2 x_3 Z_2 & \varphi(x_3) = b \\ |\varphi(Z_4)| = 16 & Z_4 = Z_3 x_4 Z_3 & \varphi(x_4) = cd. \end{array}$$

4.2. Модификации алгоритма

Для решения задачи представления слова s в виде $s = \varphi(Z_k) \cdot v$ с наибольшим возможным k можно использовать алгоритм 1 для $|v| = 0, \ldots, |s|-1$. Однако есть возможность существенно ускорить вычисления.

Пусть $d_1, \ldots, d_m$ – номера всех единичных диагоналей для слова u. Покажем, как при помощи $O(m)$ операций получить список единичных диагоналей для $u' = u \cdot a$, $a \in \Sigma$.

Составим односвязный список, каждый элемент которого соответствует единичной диагонали. Далее пройдем по всем элементам списка, для каждого из них проверим, продолжается ли диагональ на новой букве a, и удалим элемент из списка, если диагональ оборвалась. Вместе с этим будем строить набор диагоналей, необходимый для поиска слова Зимина (т. е. таких, что $d'_i < d'_{i+1}/2$). Наконец, для нижнего левого элемента матрицы совпадений проведем проверку на равенство единице и включим элемент в список, если проверка успешна. Таким образом за $O(m)$ операций мы переходим к слову u'.

Для решения задачи о представлении s в указанном виде нужно $|s|-1$ раз применить данный шаг, попутно выбрав наибольший по мощности список диагоналей, по которому восстанавливаются k и φ (шаги 2 и 3 алгоритма 1).

Время работы данного алгоритма есть $O(p)$, где p – суммарная длина единичных отрезков всех диагоналей, начинающихся в первом столбце матрицы сравнения. В худшем случае (если слово состоит из одинаковых букв) $p = |s|^2$. Оценим среднее значение p.

Для случайного слова вероятность совпадения двух его букв составляет $1/|\Sigma|$. Таким образом, диагональ матрицы совпадения начинается с t единиц с вероятностью $1/|\Sigma|^t$. Значит, математическое ожидание длины начального единичного участка диагонали составляет $1/(|\Sigma| - 1)$. Всего диагоналей $|s|$, поэтому $p = |s|/(|\Sigma| - 1)$. Отсюда следует, что приведенная модификация алгоритма работает в среднем за время $O(|s|)$ при фиксированном алфавите.

Наконец, решение основной задачи, т. е. представление слова s в виде $s = u \cdot \varphi(Z_k) \cdot v$ с наибольшим возможным k достигается путем применения модифицированного алгоритма к слову s без ℓ начальных букв ($\ell = 0, \ldots, |s|-1$).

Время работы такого алгоритма – $O(|s|^2)$ в среднем случае и $O(|s|^3)$ – в худшем, когда все буквы слова одинаковы.

Литература

1. Bean D. R., Ehrenfeucht A., McNulty G. Avoidable patterns in stings of symbols // Pacific J. Math. 1979. Vol. 85. P. 261–294.
2. Зимин А. И. Блокирующие множества термов // Мат. сб. 1982. Т. 119. С. 363–375.
3. Currie J. Pattern avoidance: themes and variations // Proc. WORDS'03 Int. Conf. Turku, 2003. P. 14–26.
4. Шур А. М. Комбинаторика слов: Учеб. пособие. Екатеринбург: Изд-во Урал. ун-та, 2003.