КИБЕРНЕТИКА

К.В. ШАХБАЗЯН, Ю.Г. ШУКУРЯН

УДК 519.6

# ЗАПРЕЩЕННЫЕ ТРЕКИ И ЗАПРЕЩЕННЫЕ ПОДТРЕКИ

**Ключевые слова:** *частично-коммутативный моноид, запрещенная строка, запрещенный трек, запрещенный подтрек.*

## ВВЕДЕНИЕ

Настоящая статья посвящена обобщениям на треки понятий запрещенных строк и запрещенных подпоследовательностей, применяемых к строкам [1–5]. Этими обобщениями являются следующие понятия.

Пусть $M(\Sigma, D)$ — трековый моноид и $s, t \in M(\Sigma, D)$ [6]. Трек $s$ называется минимальным запрещенным треком для $t$, если $s \notin \mathrm{Fact}(t)$, но все собственные факторы трека $s$ принадлежат $\mathrm{Fact}(t)$. Пусть $s = s_1 \dots s_n$ и $t = t_0 s_1 t_1 \dots s_n t_n$, где $s_1, \dots, s_n, \ t_0, \dots, t_n \in M(\Sigma, D)$. В этом случае $s$ называется подтреком трека $t$, и это отношение обозначается $s \le t$. Трек $s$ называется минимальным запрещенным подтреком для трека $t$, если $s \not\le t$, но каждый собственный подтрек $s$ является подтреком $t$, т.е. $u < s \to u \le t$ для каждого $u \in M(\Sigma, D)$.

Во многих задачах компьютерного анализа текстов большой интерес представляет рассмотрение запрещенных строк текстов, т.е. таких, которые не являются факторами текста. Понятие минимальной запрещенной строки эффективно объединяет отрицательную информацию о текстах и играет важную роль в приложениях. Например, метод сжатия текстов без потерь [7] и метод реконструкции ДНК с учетом множества ее фрагментов основан на «антисловарях» [8]. Это понятие используется в теории автоматов и символьной динамике, где каждое пространство сдвигов однозначно определяется множеством запрещенных строк [9]. Это обусловило интерес к переносу понятий запрещенных строк и запрещенных подпоследовательностей на треки.

В настоящей статье приведено решение следующих задач для запрещенных треков и запрещенных подтреков:

• $N \to L_f(N)$: по конечному антифакториальному языку треков $N$ распознать треки языка $L_f(N)$ — факториального трекового языка, для которого язык $N$ является языком минимальных запрещенных треков (на основе автомата Messner-а [10] для решения проблемы префикса в моноиде треков предложен алгоритм, распознающий $L_f(N)$ для конечного $N$);

• $t \to MFT(t)$: построить для заданного трека $t \in M(\Sigma, D)$ множество $MFT(t)$ минимальных запрещенных треков (предложен алгоритм, решающий проблему);

• $N \to L_{ST}(N)$: по конечному антиподтрековому языку треков $N$ построить автомат, допускающий язык $L = L_{ST}(N)$, для которого $N$ является языком минимальных запрещенных подтреков;

• $t \to MFST(t)$: построить для заданного трека $t \in M(\Sigma, D)$ множество $MFST(t)$ минимальных запрещенных подтреков (на основе алгоритма [11] для решения аналогичной проблемы для строк предложен алгоритм, строящий $MFST(t)$).

**МОНОИД ТРЕКОВ**

Далее использованы следующие понятия и обозначения.

Пусть $\Sigma$ — конечный алфавит, $D \subset \Sigma \times \Sigma$ — рефлексивное и симметричное отношение зависимости, $I = (\Sigma \times \Sigma) \setminus D$ — отношение независимости или перестановочности. Отношение $I$ индуцирует отношение эквивалентности $\approx$ на $\Sigma^*$. Две строки: $x, y \in \Sigma^*$, являются эквивалентными относительно $\approx$, если существует последовательность $z_1, \ldots, z_n \in \Sigma^*$ строк таких, что $x = z_1$, $y = z_n$ и для всех $i$ $(1 \le i < n)$ существуют строки $z_i', z_i'' \in \Sigma^*$ и буквы $a_i, b_i \in \Sigma$, удовлетворяющие условиям

$$\begin{cases} z_i = z_i' a_i b_i z_i'', \\ z_{i+1} = z_i' b_i a_i z_i'', \end{cases}$$

где $(a_i, b_i) \in I$, т.е. две строки эквивалентны по отношению $\approx$ тогда и только тогда, когда одну можно получить из другой путем перестановок соседних независимых букв. Классы эквивалентных строк из $\Sigma^*$ по отношению $\approx$ называются треками, а множество $M = M(\Sigma, D)$ является трековым моноидом. На элементах $M$ определено умножение — конкатенация. Обозначим $[x]$ трек $t$ для любой представляющей строки $x \in t$ (однобуквенный трек не будем брать в квадратные скобки), $|t|$ длину трека, которая является длиной любой представляющей его строки $x \in t$, и $\mathrm{Alph}(t)$ множество букв, входящих в трек $t$.

Для заданного трека $t \in M(\Sigma, D)$ множества префиксов $\mathrm{Pref}(t)$, суффиксов $\mathrm{Suff}(t)$ и факторов $\mathrm{Fact}(t)$ определяются обычным образом. Пусть $(\Sigma_1, \ldots, \Sigma_m)$ — покрытие алфавита зависимости $(\Sigma, D)$ кликами, т.е. семейство подмножеств $\Sigma_i$ таких, что

$$\bigcup_{i=1}^{m} \Sigma_i = \Sigma, \quad \Sigma_i \times \Sigma_i \subset D \ (i = 1, 2, \ldots, m), \quad (a, b) \in D \Leftrightarrow \exists\, i : a, b \in \Sigma_i.$$

Любой трек $t \in M(\Sigma, D)$ при заданном покрытии $(\Sigma_1, \ldots, \Sigma_m)$ можно представить $m$-кой строк, которую обозначим $\pi(t) = \{\pi_1(t), \ldots, \pi_m(t)\}$. Здесь $\pi_i(t) = \pi_{i1}(t), \ldots, \pi_{in_i}(t) \in \Sigma_i^*$, где $\pi_i(t)$ — проекция строки $y \in t$ на алфавит $\Sigma_i$, $n_i$ — ее длина.

Для каждого заданного $t \in M(\Sigma, D)$ любой префикс $p \in \mathrm{Pref}(t)$ можно представить с помощью $m$-ки целых чисел по отношению к $\pi(t)$:

$$\underline{p} = (l_1, \ldots, l_m) \tag{1}$$

такой, что $\pi_i(p) = \pi_{il}(t), \ldots, \pi_{il_i}(t)$ [6].

Пусть $(\Sigma, D)$ — алфавит зависимости трекового моноида $M(\Sigma, D)$. Граф зависимости трека $t$ есть помеченный ациклический граф $G_t = [V, E, \lambda]$, где $V$ — множество вершин, $E \subseteq V \times V$ — множество дуг, граф $(V, E)$ ациклический и индуцирует частичный порядок на своих вершинах, $\lambda : V \to \Sigma$ — метки вершин такие, что $(\lambda(v_1), \lambda(v_2)) \in D$ тогда и только тогда, когда $(v_1, v_2) \in E \cup E^{-1} \cup id_V$. Отметим, что множество вершин графа $G_t$, помеченных одной и той же меткой

$V_b = \{v \in V \mid \lambda(v) = b\}$, упорядочено и поэтому имеет смысл говорить об $i$-й вершине во множестве $V_b$.

Если задан трек $t = [a_1 \dots a_n] \in M(\Sigma, D)$, то можно построить соответствующий ему граф зависимости $G_t$. Возьмем множество $n$ вершин, например, $V = \{1, 2, \dots, n\}$. Пометим вершину $i$ буквой $a_i$ и положим $(i, j) \in E \Leftrightarrow (a_i, a_j) \in D$, $i < j$. Этот граф соответствует треку $t$ в том смысле, что каждый порядок вершин, индуцируемый графом $G_t$, несет строку меток, принадлежащую треку $t$. И обратно, каждой строке, принадлежащей треку $t$, соответствует порядок на вершинах графа $G_t$.

Далее везде будем считать фиксированным покрытие $(\Sigma_1, \dots, \Sigma_m)$ кликами алфавита зависимости $(\Sigma, D)$. Тогда $m$ — число клик в покрытии. Обозначим $\alpha$ размер наибольшей клики в графе независимости алфавита $\Sigma$.

### ЗАПРЕЩЕННЫЕ ТРЕКИ

Пусть $M(\Sigma, D)$ — трековый моноид; $L \subseteq M(\Sigma, D)$ называется факториальным языком, если он содержит все факторы своих треков; $L \subseteq M(\Sigma, D)$ называется антифакториальным языком, если он не содержит ни одного фактора своих треков.

Пусть $L$ — факториальный язык. Трек $t$ называется запрещенным треком для $L$, если $t \notin L$. Трек t называется минимальным запрещенным треком для $L$, если $t$ — запрещенный трек для $L$, но все собственные факторы трека $t$ принадлежат $L$. Обозначим $MFT(L)$ множество минимальных запрещенных треков языка $L$. Пусть $L \subseteq M(\Sigma, D)$ — факториальный язык треков. Его дополнение $L^c$ является двухсторонним идеалом в $M(\Sigma, D)$. Очевидно, $MFT(L)$ — базис этого идеала, т.е.

$$L^c = M(\Sigma, D) \cdot MFT(L) \cdot M(\Sigma, D),$$

$$L \cap MFT(L) = \varnothing,$$

$$L = M(\Sigma, D) \setminus M(\Sigma, D) \cdot MFT(L) \cdot M(\Sigma, D).$$

Следовательно, $MFT(L)$ однозначно определяет $L$ и обратно.

**Пример 1.** Рассмотрим моноид $M(\Sigma, D)$, где $\Sigma = \{a, b, c, d\}$, $D = \{(d, a), (a, c), (c, b)\}$, и язык $L = \mathrm{Fact}(t)$, где $t = [dababc]$ (рис. 1).

Трек $s = [dabc]$ (рис. 2) является запрещенным треком для $L$, но он не входит в $MFT(L)$, так как один из его собственных факторов $s' = [dac] \notin L$. Легко видеть, что все собственные факторы $s'$ являются факторами $s$. Следовательно, $s' \in MFT(L)$.

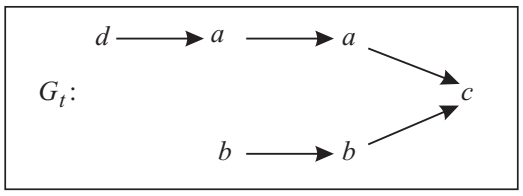


*Рис. 1.* Граф зависимости трека $t = [dababc]$

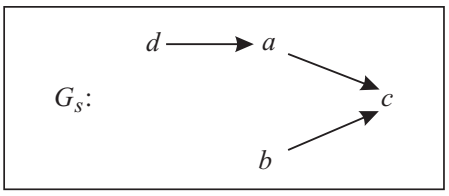


*Рис. 2.* Граф зависимости трека $s = [dabc]$

**Пример 2.** Рассмотрим моноид $M(\Sigma, D)$, где $\Sigma = \{a, b, c, d, e\}$, $D = \{(d, a), (a, c), (c, b), (b, e)\}$. Пусть $L = \mathrm{Fact}(t_1) \cup \mathrm{Fact}(t_2)$, где $t_1 = [daabbca]$, $t_2 = [dadcac]$ (рис. 3).

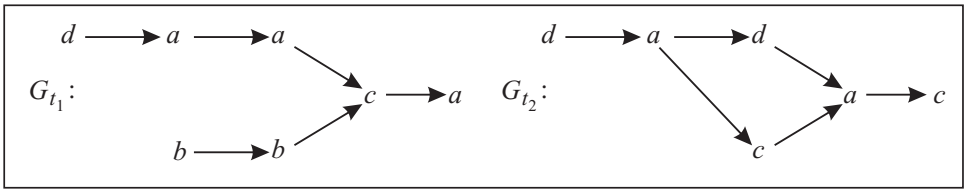


*Рис. 3*. Графы зависимости треков $t_1 = [daabbca]$, $t_2 = [dadcac]$

Очевидно, что трек $s = [daca]$ (рис. 4) — запрещенный для $L$, так как не является фактором $t_1$ или $t_2$. Его собственный фактор $s_1 = [dac]$ является фактором $t_2$, а собственный фактор $s_2 = [aca]$ является фактором $t_1$. Аналогично и остальные собственные факторы трека $s$ являются либо факторами $t_1$, либо $t_2$ и по определению принадлежат языку $L$. Следовательно, $s$ — минимальный запрещенный трек языка $L$.

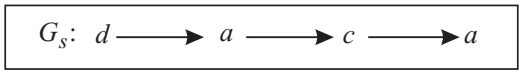


*Рис. 4*. Граф зависимости трека $s = [daca]$

Отметим особо, что трек $e$ является минимальным запрещенным треком для $L$, так как буква $e$ не входит в алфавиты треков $t_1$ и $t_2$.

**ПРОБЛЕМА $N \to L_f(N)$**

Заданы: моноид $M(\Sigma, D)$ и конечный антифакториальный язык $N = \{n_1, \ldots, n_k\} \subseteq M(\Sigma, D)$.

Построить алгоритм, распознающий треки, принадлежащие языку

$$L_f(N) = M(\Sigma, D) \setminus M(\Sigma, D) \cdot N \cdot M(\Sigma, D), \tag{2}$$

т.е. языку треков $L_f(N)$, для которого $N$ является множеством минимальных запрещенных треков.

**Утверждение 1.** Существует алгоритм, распознающий принадлежность трека $t$ языку $L_f(N)$ за время, линейное по $|t| \sum_{i=1}^{k} |n_i|$.

**Доказательство.** Воспользуемся автоматом $A_n$ Messner-a [11], который строится по треку $n \in M(\Sigma, D)$ за время, линейное по $|nt|$, и распознает язык $M(\Sigma, D) \cdot n$. Автомат $A_n$ минимальный. Согласно (2) для выяснения принадлежности трека $t$ языку $L_f(N)$ достаточно проверить, что ни один из треков $n_1, \ldots, n_k$ не является фактором трека $t$. Очевидно, что время этой проверки линейное по $|t| \sum_{i=1}^{k} |n_i|$.

**ПРОБЛЕМА $L \to MFT(L)$**

По факториальному языку треков $L$ построить множество $MFT(L)$ минимальных запрещенных треков. Здесь проблема решается для частного случая, когда $L$ является множеством факторов одного трека, т.е. $L = \mathrm{Fact}(t)$ при заданном $t \in M(\Sigma, D)$.

**Определение 1.** Пусть $s, t \in M(\Sigma, D)$ и $s \in \mathrm{Fact}(t)$. Если буквы $a, b \in \Sigma$ такие, что $[as] \in \mathrm{Fact}(t)$, $[sb] \in \mathrm{Fact}(t)$, $[asb] \notin \mathrm{Fact}(t)$, будем говорить, что фактор $s$ порождает минимальный запрещенный трек $[asb]$.

**Определение 2.** Пусть $s \in \mathrm{Fact}(t)$. Если $t = psu$, где $u \in M(\Sigma, D)$, то пару $(p, s)$ назовем вхождением фактора $s$ в трек $t$.

Пусть существует $n$ различных вхождений фактора $s$ в $t$: $\Omega(s) = ((p_1, s), \ldots, (p_n, s))$, где $p_1, \ldots, p_n \in \mathrm{Pref}(t)$. Для каждого вхождения определим следую-

щие множества букв:

$$\psi(p_i, s) = \{a \in \Sigma \mid p_i sa \in \mathrm{Pref}(t)\},\ \xi(p_i, s) = \{a \in \Sigma \mid \exists\, r \in \mathrm{Pref}(p_i) : ra = p_i\}.$$

Назовем окрестностью фактора $s$ во вхождении $(p_i, s)$ пару множеств $(\xi(p_i, s), \psi(p_i, s))$.

Отметим, что для всех $i = 1, \ldots, n$ имеет место $|\psi(p_i, s)|, |\xi(p_i, s)| \le m$. Определим соответствующий список окрестностей фактора $s$: $\Omega^*(s) = ((\psi(p_1, s), \xi(p_1, s)), \ldots, (\psi(p_n, s), \xi(p_n, s)))$. Длина списка $\Omega^*(s)$ не превышает $|t|^{\alpha}$.

**Утверждение 2.** Если $(p, s)$ — вхождение фактора $s$ в трек $t$ и $b \in \psi(p, s)$, $a \in \xi(p, s)$, то $asb \in \mathrm{Fact}(t)$.

**Следствие 1.** Если трек $s$ имеет единственное вхождение в трек $t$, то для $t$ не существует запрещенного трека, порожденного фактором $s$.

**Утверждение 3.** Фактор $s \in \mathrm{Fact}(t)$ порождает минимальный запрещенный трек $asb \notin \mathrm{Fact}(t)$, где $a \in \cup_i \xi(p_i, s)$, $b \in \cup_i \psi(p_i, s)$, тогда и только тогда, когда не существует вхождения $(p, s) \in \Omega(s)$ фактора $s$ такого, что $b \in \psi(p, s)$, $a \in \xi(p, s)$. Порождает ли данный фактор $s$ минимальный запрещенный трек, устанавливается по списку $\Omega^*(s)$ за время $O(|\Sigma|^2 |t|^{\alpha})$, где $\alpha$ — размер наибольшей клики в графе независимости алфавита $\Sigma$.

**Доказательство.** Предположим, что $asb$ — запрещенный трек $t$. Если $\exists\, (p, s) \in \Omega(s) : b \in \psi(p, s)$, $a \in \xi(p, s)$, то по утверждению 2 $asb \in \mathrm{Fact}(t)$, что противоречит предположению.

Пусть $b \in \psi(p_i, s)$, $a \in \xi(p_i, s)$, $i \ne j$, но $\nexists\, (p, s) \in \Omega(s) : b \in \psi(p, s)$, $a \in \xi(p, s)$. Предположим $asb \in \mathrm{Fact}(t)$. Тогда трек $t$ можно представить в виде $t = w_1 asb w_2$, т.е. существует вхождение $(w_1 a, s)$ с окрестностью $(\xi(w_1 a, s), \psi(w_1 a, s))$ такой, что $b \in \psi(s)$, $a \in \xi(s)$, что противоречит предположению. Приведенная оценка основана на том, что число префиксов трека $t$ имеет порядок $O(|t|^{\alpha})$.

**Пример 3.** Рассмотрим трек $t$ из примера 1. В нем фактор $a$ имеет два вхождения и соответственно две окрестности: $(\{d\}, \{a\})$ и $(\{a\}, \{c\})$. По утверждению 3 треки $[dac], [aaa]$ являются минимальными запрещенными треками, порождаемыми фактором $a$. Трек $b$ имеет окрестности $(\varnothing, \{b\})$ и $(\{b\}, \{c\})$. Следовательно, фактор $b$ порождает единственный минимальный запрещенный трек $[bbb]$.

### АЛГОРИТМ РЕШЕНИЯ ПРОБЛЕМЫ $N \to L_f(N)$

Алгоритм накапливает множество факторов трека $t$ вместе с окрестностями всех вхождений факторов, а затем проверяет условие утверждения 3 для каждого фактора $s \in \mathrm{Fact}(t)$. Алгоритм состоит из четырех этапов.

**1. Получение графа префиксов трека $t$ с помощью алгоритма Avellon-a [5].** Пусть $t \in M(\Sigma, D)$. Алгоритм строит граф $G_{\mathrm{Pref}}(t) = (\mathrm{Pref}(t), U_t, \mu_t)$ префиксов трека $t$, где $U_t = \{(u, r) \mid u, r \in \mathrm{Pref}(t), \exists\, a \in \Sigma : r = ua\}$, метка на дуге $\mu_t(u, r) = a$, если $r = ua$. Вершинами графа $G_{\mathrm{Pref}}(t)$ являются наборы $\underline{p} = (p_1, \ldots, p_m)$ вида (1), представляющие соответствующие префиксы $u \in \mathrm{Pref}(t)$ длиной проекций префикса относительно проекций $\pi(t)$. Для каждого префикса $u \in \mathrm{Pref}(t)$ алгоритм строит $A(u)$ — список дуг вида $(u, r)$, где $u, r \in \mathrm{Pref}(t)$, исходящих из вершины $u$ в графе $G_t$. При этом в списке вместе с дугой содержится ее метка.

Число префиксов трека $t$ имеет порядок $O(|t|^{\alpha})$, где $\alpha$— размер наибольшей клики в графе независимости алфавита $\Sigma$ и сложность алгоритма есть $O(|t|^{\alpha})$.

**Пример 4.** Рассмотрим моноид $M(\Sigma, D)$, где $\Sigma = \{a, b; c, d\}$, $D = \{(d, a), (a, c), (d, c), (c, b)\}$ и трек $t = [dabac]$ (рис. 5). Задано покрытие графа

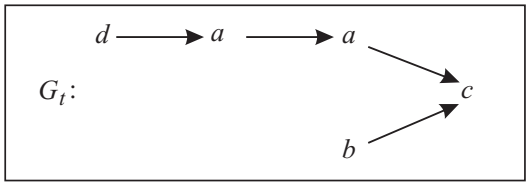


*Рис. 5*. Граф зависимости трека $t = [dabac]$

$(\Sigma, D)$ кликами $\Sigma = \{d, a, c\} \cup \{b, c\}$. Проекции трека $t$ на клики: $\pi_1(t) = daac$, $\pi_2(t) = bc$.

Трек $[ab]$ имеет два вхождения в трек $t$: $\Omega([ab]) = ((d, [ab]), ([da], [ab]))$. Легко видеть, что $\psi(d, [ab]) = \{a\}$, $\xi(d, [ab]) = \{d\}$, $\psi([da], [ab]) = \{c\}$, $\xi([da], [ab]) = \{a\}$. В графе $G_{\mathrm{Pref}(t)}$ (рис. 6) этим двум вхождениям соответствуют окрестность $(\{a\}, \{d\})$ фактора $[ab]$ во вхождении $(d, [ab])$ и окрестность $(\{c\}, \{a\})$ фактора $[ab]$ во вхождении $([da], [ab])$. Из рассмотрения этих окрестностей по утверждению 3 получаем, что фактор $[ab]$ порождает два минимальных запрещенных трека для трека $t$: $[dabc]$ и $[aaba]$.

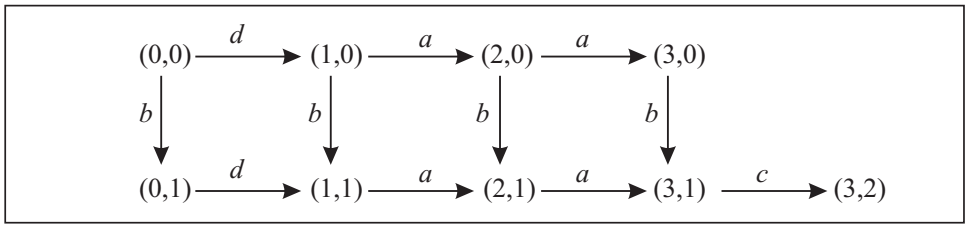


*Рис. 6*. Граф $G_{\mathrm{Pref}(t)}$

**2. Сбор информации об окрестностях вхождений каждого фактора $s$ в трек $t$.** При сборе этой информации перебираются все пары $(r, s)$, где $r$ и $rs$ — вершины графа $G_{\mathrm{Pref}(t)}$, лежащие на одном пути из начальной вершины, т.е. $r, rs \in \mathrm{Pref}(t)$, $s \in \mathrm{Fact}(t)$. Пусть префиксы $r, rs \in \mathrm{Pref}(t)$. Тогда известны их представления вида (1): $\underline{r} = (l_1, \ldots, l_m)$ и $\underline{rs} = (k_1, \ldots, k_m)$. При этом очевидно, что проекция фактора $s$ на клику $\Sigma_i$ есть $\pi_i(s) = \pi_{i,l_i+1}(t) \ldots \pi_{i,k_i}(t) \in \Sigma_i^*$.

Легко видеть, что если задано вхождение $(r, s) \in \Omega(s)$, то $\psi(r, s) = \{a \in \Sigma \mid \exists\, u \in A(rs) : \mu_t(u) = a\}$, где $A(rs)$ вычисляется одновременно с $G_{\mathrm{Pref}(t)}$; $\xi(r, s)$ можно вычислить по следующей формуле: если $\underline{r} = (l_1, \ldots, l_m)$, то $\xi(r, s) = \{\pi_{l_1}, \ldots, \pi_{l_m}\}$. Следовательно, множества $\psi(r, s)$ и $\xi(r, s)$ можно вычислить при рассмотрении вхождения $(r, s) \in \Omega(s)$. Число факторов трека $t$ имеет порядок $\mathrm{O}(|t|^{2\alpha})$, где $\alpha$ — размер наибольшей клики в графе независимости алфавита $\Sigma$. Поскольку для сравнения факторов и для вычисления одной окрестности $(\xi(r, s), \psi(r, s))$ требуется $\mathrm{O}(t)$ шагов, сложность этого этапа есть $\mathrm{O}(|t|^{2\alpha+1})$.

**3. Структура для сбора информации об окрестностях вхождений каждого фактора $s$ в трек $t$.** Собранную информацию необходимо сохранять в некотором хранилище информации, приспособленном для запоминания множеств треков с сопутствующей информацией. Это хранилище должно быть структурой, обеспечивающей быстрый поиск треков во множестве, включение новых треков, изменение сопутствующей информации. В работе [13] предложена специальная структура $F$ для описания множеств треков. Эта структура является обобщением префиксных деревьев, используемых для хранения строк. Напомним, что в префиксных деревьях имеется одна вершина для каждого общего префикса и строки запоминаются как последовательности меток путей, ведущих от корня к финальным вершинам.

Для описания множеств треков предложена структура $F = (\tau_1, \ldots, \tau_m, \Psi)$, где $\tau_i = (V_i, E_i, \alpha_i)$, $i = 1, \ldots, m$, — префиксное дерево для хранения множества строк $\omega(\tau_i)$ в алфавите $\Sigma_i$; $V_i$ — вершины дерева; $E_i$ — дуги дерева; $\alpha_i$ — функ-

ция пометки дуг буквами из $\Sigma_i$; $\Psi$ — префиксное дерево для хранения множества строк длины $m$ в алфавите $\bigcup_{i=1}^{m} V_i$. Каждая такая строка принадлежит $V_1 \times \ldots \times V_m$.

Структура $F$ используется для хранения множества треков $\omega(F)$ в следующем смысле. Трек $s$ принадлежит множеству $\omega(F)$ тогда и только тогда, когда выполнены условия: в каждом дереве $\tau_i$ существует путь из корня в некоторую вершину $v_i$, несущий строку $\pi_i(s)$, $i = 1, \ldots, m$, причем в дереве $\Psi$ существует путь из корня в висячую вершину, несущий строку $v_1 \ldots v_m$.

Структуру $F$ можно использовать для хранения множества всех факторов трека $t$ и словаря запрещенных треков $MTF(t)$. Структура $F$ позволяет за линейное время устанавливать принадлежность трека множеству $\omega(F)$ или добавлять новый трек [13].

Для сбора информации о вхождениях каждого фактора $s$ в трек $t$ изменим структуру $F$, добавив в нее возможность задания окрестностей треков. А именно, рассмотрим структуру $F = (\tau_1, \ldots, \tau_m, \Psi')$, здесь $\Psi'$ — дерево $\Psi$, в котором добавлена возможность в каждой висячей вершине хранить списки пар вида $(\gamma, \delta)$, где $\gamma, \delta \subseteq \Sigma$ и $|\gamma|, |\delta| \le m$. Тогда, если висячая вершина дерева $\Psi$ соответствует фактору $s$, свяжем с нею список окрестностей треков $\Omega^*(s) = ((\psi(p_1, s), \xi(p_1, s)), \ldots, (\psi(p_n, s), \xi(p_n, s)))$. Список $\Omega^*(s)$ используется для проверки условий утверждения 3, т.е. порождает ли фактор $s$ минимальный запрещенный трек, и в случае положительного ответа для нахождения этого запрещенного трека.

**4. Сравнение окрестностей факторов.** Отметим, что каждый фактор может иметь $O(|t|^{2\alpha})$ вхождений и соответственно столько же окрестностей. Для каждого фактора $s$ осуществляется поиск таких пар букв $(a, b)$ алфавита $\Sigma$, для которых выполняется условие утверждения 3. В этом случае $asb$ — минимальный запрещенный трек $t$.

Очевидно, этот этап алгоритма требует $O(|t|^{2\alpha+1})$ шагов.

**Утверждение 4.** Существует алгоритм построения множества $MTF(L)$ запрещенных треков языка $L = \mathrm{Fact}(t)$ для трека $t \in M(\Sigma, D)$ со сложностью $O(|t|^{2\alpha+1})$, где $\alpha$ — размер наибольшей клики в графе независимости алфавита $\Sigma$.

### ЗАПРЕЩЕННЫЕ ПОДТРЕКИ

Если $G_1$ — подграф графа $G_2$, то будем писать $G_1 \le G_2$.

**Определение 3.** Пусть $s, t \in M(\Sigma, D)$ таковы, что выполнено $s = s_1 \ldots s_n$ и $t = t_0 s_1 t_1 \ldots s_n t_n$, где $s_1, \ldots, s_n, t_0, \ldots, t_n \in M(\Sigma, D)$. Из этого равенства следует, что $G_s \le G_t$. В этом случае $s$ назовем подтреком трека $t$ и обозначим это отношение $s \le t$ (обозначим $\not\le$ отрицание отношения $\le$). Отношение $s < t$ означает $s \le t$, но $s \ne t$.

Трек $s \in M(\Sigma, D)$ есть запрещенный подтрек для трека $t \in M(\Sigma, D)$, если $s \not\le t$.

Трек $s \in M(\Sigma, D)$ есть запрещенный подтрек для языка треков $L \subseteq M(\Sigma, D)$, если $s$ является запрещенным подтреком для всех $t \in M(\Sigma, D)$.

Запрещенный трек s является минимальным запрещенным подтреком для $t$, если каждый собственный подтрек $s$ является подтреком $t$, т.е. $u < s \to u \le t$ для каждого $u \in M(\Sigma, D)$.

Обозначим $MFST(t)$ множество всех минимальных запрещенных подтреков для $t$.

Язык $L \subseteq M(\Sigma, D)$ называется подтрековым, если он содержит все подтреки своих треков, и антиподтрековым, если ни один из его треков не является подтреком трека, принадлежащего $L$.

**Пример 5.** Рассмотрим моноид треков $M(\Sigma, D)$, где $\Sigma = \{a, b, c, d\}$, $D = \{(d, a), (a, c), (c, b)\}$, а также треки $t = [dababc]$, $u = [acb]$. Очевидно, что $u \not\leq t$. Однако $u \notin MFT(v)$, так как $[cb] \not\leq t$.

Пусть $L \subseteq M(\Sigma, D)$ — подтрековый язык. Отметим, что $MFST(L)$ однозначно характеризует $L$:

$$\begin{cases} L = M(\Sigma, D) \setminus e(MFST(L)), \\ MFST(L) = \{u \mid u \notin L,\ S(u) \setminus \{u\} \subseteq L\}, \end{cases} \tag{3}$$

где $e(MFST(L))$ — множество треков, имеющих подтрек, принадлежащий $MFST(L)$, $S(u)$ — множество всех подтреков трека $u$.

**Следствие 2.** Языки $L$ и $MFST(L)$ либо одновременно рациональны, либо нерациональны.

Соотношения (3) устанавливают биекцию между подтрековыми и антиподтрековыми языками.

#### ПРОБЛЕМА $N \to L_{ST}(N)$

Для данного антиподтрекового языка $N$ построить автомат, распознающий язык $L = L_{ST}(N)$, для которого $N = MFST(L)$.

**Утверждение 5.** Для любого конечного множества $N = \{u_1, \ldots, u_n\}$ треков моноида $M(\Sigma, D)$ существует конечный детерминированный автомат $B(N)$, допускающий язык $L = L_{ST}(N)$.

**Доказательство.** В работе [14] утверждение 2 состоит в том, что для любого трека $p \in M(\Sigma, D)$ существует автомат Зеленки $A(p)$, который допускает $t \in M(\Sigma, D)$ тогда и только тогда, когда $p \leq t$. Следовательно, существует автомат $\overline{A}(p)$, допускающий $t \in M(\Sigma, D)$ тогда и только тогда, когда $p \not\leq t$, а автомат $B(N)$ — прямое произведение автоматов $\overline{A}(u_1), \ldots, \overline{A}(u_n)$, допускает язык $L_{ST}(N)$.

#### ПРОБЛЕМА $t \to MFST(t)$

Задан трек $t \in M(\Sigma, D)$. Построить множество $MFST(t)$ минимальных запрещенных подтреков.

**Утверждение 6.** Пусть $s, t \in M(\Sigma, D)$. Тогда

- отношение $s \leq t$ имеет место тогда и только тогда, когда для каждой строки $y$ трека $t$ существует строка $x$ трека $s$ такая, что $x \leq y$;
- отношение $s \not\leq t$ имеет место тогда и только тогда, когда для каждой строки $y$ трека $t$ и каждой строки $x$ трека $s$ имеет место $x \not\leq y$;
- отношение $s \in MFST(t)$ имеет место тогда и только тогда, когда для каждой строки $y \in t$ и каждой строки $x \in s$ имеет место $x \in MFS(y)$, т.е. $s \in MFST(t) \Leftrightarrow \forall\ y \in t\ \forall\ x \in s : x \in MFS(y)$. Здесь $MFS(y)$ — множество минимальных запрещенных подпоследовательностей строки $y$ [8].

**Пример 6.** Рассмотрим моноид $M(\Sigma, D)$, где $\Sigma = \{a, b, c\}$, $D = \{(a, c), (c, b)\}$, а также треки $t = [acb]$, $s = [ba]$. Очевидно, что $ba \not\leq acb$. Однако $[ba] \leq [acb]$.

Из утверждения 6 следует, что множество $MFST(t)$ можно получить в процессе порождения множества $MFS(y)$, если $y \in t$. Надо только иметь метод отсева тех строк $x \in MFS(y)$, для которых $[x] \notin MFST(t)$.

Дальнейшие рассуждения базируются на теореме о запрещенных подпоследовательностях для строк, приведенной в [8]. Прежде всего рассмотрим некую подпоследовательность, связанную с каждой минимальной запрещенной подпос-

ледовательностью, которая позволит строить $MFST(t)$ путем небольшого видоизменения рекуррентных формул, полученных в [8] для построения $MFS(y)$, если $y \in t$.

В работе [8] приведена теорема 4.1, описывающая рекуррентный процесс построения минимальных запрещенных подпоследовательностей данной строки.

Представим ее. Пусть $x \in X$ и $u \in X^*$ — произвольные. Если $x \in \mathrm{Alph}(u)$, то $u = u_1 x u_2$, где $u_1, u_2 \in X^*$ и $x \notin \mathrm{Alph}(u_2)$. Если $z \in \mathrm{Alph}(xu_2x)$, то обозначим $T(z)$ множество всех букв, входящих в строку $xu_2x$ после $z$. Тогда

$$MFS(ux) = MFS(u) \setminus X^* x \cup \bigcup\nolimits_{z \in \mathrm{Alph}(xu_2)} (MFS(u) \cap X^* zx) T(z). \qquad (4)$$

Если $x \notin \mathrm{Alph}(u)$, то

$$MFS(ux) = MFS(u) \setminus \{x\} \cup x\mathrm{Alph}(ux). \qquad (5)$$

Эта теорема использована в on-line алгоритме [8] для вычисления множества запрещенных подпоследовательностей $MFS(u)$ для строки $u$. Алгоритм работает экспоненциальное время по $|u|$.

Прежде всего выведем следствие из этой теоремы, связывающее структуру строки $u$ с ее минимальными запрещенными подпоследовательностями.

Пусть $u = y_0 \dots y_n \in X^*$, $x \in X$. Обозначим $|u| = n+1$ длину строки $u$, а $|u|_x$ число вхождений буквы $x$ в строку $u$. Если $y_i = x$ и $|y_0 \dots y_i|_x = j$, то будем число $i$ называть координатой $j$-го вхождения буквы $x$ в строку $u$.

**Лемма.** Пусть $u = y_0 \dots y_n \in X^*$. Для того чтобы $a = a_0 \dots a_k \in MFS(u)$:

- при $k = 0$ необходимо и достаточно, чтобы $a_0 \notin \mathrm{Alph}(u)$;
- при $k \geq 1$ необходимо и достаточно, чтобы в строку $u$ входила в качестве подпоследовательности строка

$$A(a,u) = A(a_0 \dots a_k, u) = a_1 a_0 a_2 a_1 a_3 a_2 \dots a_{k-1} a_{k-2} a_k a_{k-1},$$

причем для соответствующих координат букв этой подпоследовательности в строке $u$:

$$i_1, i'_0, i_2, i'_1, i_3, i'_2, \dots, i_{k-1}, i'_{k-2}, i_k, i'_{k-1}$$

должны выполнятся приведенные далее условия.

1. Первое вхождение буквы $a_0$ в строку $u$ имеет координату $i'_0$, последнее вхождение буквы $a_k$ в строку $u$ имеет координату $i_k$, т. е. $y_{i'_0} = a_0$, $y_{i_k} = a_k$.

2. Числа $i_j, i'_j$ — суть координаты двух последовательных вхождений одной и той же буквы $a_j$ в строку $u$, т.е. $y_{i_j} = y_{i'_j} = a_j$ и если $i_j < l < i_j'$, то $y_l \neq a_j$, $j = 1, \dots, k-1$.

3. Между двумя последовательными вхождениями буквы $a_j$ с координатами $i_j, i'_j$ в строку $u$ находятся вхождения букв $a_{j-1}$ и $a_{j+1}$: $i_j \leq i'_{j-1} < i_{j+1} \leq i'_j$, $j = 1, \dots, k-1$. При этом $i_{j+1}$ — самое правое из вхождений буквы $a_{j+1}$ в строку $a_{i'_{j-1}} a_{i'_{j+1}+1} \dots a_{i'_{j-1}-1} a_{i'_j}$.

4. Равенство $i_{j+1} = i'_j$ выполняется тогда и только тогда, когда в строке $u$ имеет место $a_j = a_{j+1}$.

**Доказательство.** Случай $k = 0$ очевиден. Рассмотрим случай $k \geq 1$. Пусть задана строка $u = y_0 \dots y_n \in X^*$ и пусть строка $a = a_0 \dots a_k$ такова, что в строке $u$ существует подпоследовательность $A(a,u)$, удовлетворяющая условиям леммы. Пусть $a_j \neq a_{j+1}$, $j = 0, \dots, k-1$. Покажем, что в этом случае $a \in MFS(u)$. Легко видеть, что строка $a$ не является подпоследовательностью строки $u$. Пусть теперь из строки $a$ удалена произвольная буква $a_j$. Очевидно, что

$$y_{i'_0} y_{i'_1} \dots y_{i'_{j-1}} y_{i_{j+1}} \dots y_{i_{k-1}} y_{i_k} = a_0 a_1 \dots a_{j-1} a_{j+1} \dots a_k \leq u.$$

Следовательно, каждая собственная подпоследовательность строки $a$ является подпоследовательностью строки $u$, что означает $a \in MFS(u)$.

Случай, когда существует $j$ такое, что $a_j = a_{j+1}$, рассматривается аналогично.

Покажем теперь обратное. Предположим, что лемма верна для строки $u$, т.е. каждой строке $a = a_0 \dots a_k \in MFS(u)$ соответствует строка $A(a,u)$, удовлетворяющая условиям 1–4. Докажем, что лемма верна для строки $ux$, где $x \in X$.

Рассмотрим подробнее рекуррентный процесс построения множества $MFS(ux)$ по множеству $MFS(u)$, описанный теоремой 4.1 в [8].

Очевидно, что если $a = a_0 \dots a_k \in MFS(u)$ и $a_k \neq x$, то $A(a,u)$ удовлетворяет условиям леммы и на строке $ux$. Следовательно, в этом случае $a = a_0 \dots a_k \in$ $\in MFS(ux)$.

Пусть теперь $a_k = x$. При этом возможны следующие варианты: $u = y_0 \dots y_n$, $x \notin \mathrm{Alph}(u)$, т.е. $x$ не входит в строку $u$. Согласно (4) $MFS(ux) = MFS(u) \cup$ $\cup \bigcup_0^n xy_i$. Тогда строке $a = xy_i$ соответствует строка $A(a,ux) = y_i x$, которая очевидно является подпоследовательностью строки $ux = y_0 \dots y_n x$.

Пусть $u = y_0 \dots y_n \in X^*$, $x \in X$ и $x$ входит в строку $u = u_1 x u_2$, $x \notin \mathrm{Alph}(u_2)$, где выделено самое правое вхождение буквы $x$ в $u$. Предположим, что уже построена подпоследовательность $A(a,u)$ строки $u$ для каждой строки $a \in MFS(u)$.

Формула (3) теоремы 4.1 из [8] определяет следующий порядок действий для добавления новых запрещенных подпоследовательностей во множество $MFS(ux)$, которые строятся по строкам из $MFS(u)$, оканчивающимся буквой $x$. Пусть $a = rzx \in MFS(u)$, где $r \in X^*$, $z \in X$. Тогда строка $A(a,u) = A(rzx,u) = sxz$, где $s \in X^*$.

Для каждой буквы $b$, входящей в строку $xu_2x$ после буквы $z$, строка $ab = rzxb \in MFS(ux)$ и соответствующая строка $A(ab,ux) = sxzbx$. Для нее $t'_{k-1} = n+2$, $t_{k+1}$ равно координате последнего вхождения буквы $b$ в строку $u$. Координаты остальных букв не меняются. Очевидно, что $A(ab,ux)$ удовлетворяет условиям леммы.

Вернемся к трекам $t \in M(\Sigma, D)$ и к минимальным запрещенным подтрекам, входящим во множество $MFST(t)$. Будем называть цепью либо однобуквенный трек, либо трек $a = [a_0 \dots a_k] \in M(\Sigma, D)$, $k \geq 1$, для которого выполняется $(a_i, a_{i+1}) \in D$ при всех $i = 0,1, \dots, k-1$.

**Теорема 1.** Пусть $s, t \in M(\Sigma, D)$. Если $s \in MFST(t)$, то $s$ — цепь.

**Доказательство.** Предположим, что $s \in MFST(t)$. Рассмотрим произвольные строки $u = y_0 \dots y_n \in t$ и $a = a_0 \dots a_k \in s$ в том случае, когда в строке $a$ нет стоящих рядом одинаковых букв. (Если таковые имеются, доказательство аналогично.) Согласно утверждению 6, если $a \in s$, то $a_0 \dots a_k \in MFS(u)$. Согласно лемме строка

$$A(a,u) = a_1 a_0 a_2 a_1 a_3 a_2 \dots a_{i-1} a_{i-2} a_i a_{i-1} \dots a_{k-1} a_{k-2} a_k a_{k-1}$$

является подпоследовательностью строки $u = y_0 \dots y_n$.

Предположим, что трек $s$ — не цепь. Тогда существуют $a_i \neq a_{i-1}$ такие, что $(a_i, a_{i-1}) \in I$. Следовательно, $a' = a_0 \dots a_i a_{i-1} \dots a_k \in s$. Однако, как легко видеть, строка $a'$ является подпоследовательностью строки $A(a,u)$ и тем самым является подпоследовательностью строки $u = y_0 \dots y_n$, что противоречит предположению.

**Следствие 3.** Пусть $t \in M(\Sigma, D)$ и строка $u \in t$. Тогда $MFST(t) =$ $= \{x \in MFS(u) \mid [x] \text{ — цепь}\}$.

Отсюда следует, что формулы теоремы 4.1 из [8] легко изменить так, чтобы они описывали процесс построения множества минимальных запрещенных тре-

ков $MFST(t)$, где $t \in M(\Sigma, D)$. А именно, нужно в формулу (3) теоремы 4.1 добавить условие, которое разрешает порождение только цепей. Для этого необходимо переопределить $T(z)$ как множество таких букв $y$, которые входят в строку $xu_2x$ после $z$, и таких, что $(y,x) \in D$. Кроме того, в случае $x \notin \mathrm{Alph}(u)$ нужно вместо (4) использовать формулу

$$MFST(tx) = MFST(t) \setminus \{x\} \cup \{xy |\ y \in \mathrm{Alph}(tx), (y,x) \in D\}.$$

**ЗАКЛЮЧЕНИЕ**

Статья содержит решения нескольких задач о запрещенных треках и подтреках. В частности, предложены алгоритмы, строящие множества минимальных запрещенных треков и минимальных запрещенных подтреков для заданного трека.

СПИСОК ЛИТЕРАТУРЫ


1. Mignosi F., Restivo A., Sciortino M. Words and forbidden factors // Theoretical Comput. Sci. – 2001. — **273**, N 1–2. — P. 99–117.
2. Crochemore M., Rytter W. Jewels of stringology // World Sci. Publ. Co. Rtc. Ltd, 2002. – 320 p.
3. Computing forbidden words of regular languages / M. Beal, M. Crochemore, F. Mignosi at al. // Fundamenta Inform. 2003. — **56**, N 1–2. — P. 121–135.
4. Crochemore M., Mignosi F., Restivo A. Automata and forbidden words // Inform. Process. Lett. — 1998. — **67**. — P. 111–117.
5. Petcovic T., Ciricv, Bogdanovic S. Minimal forbidden subwords // Inform. Process. Lett. — 2004. — **92**. — P. 211–218.
6. Dikert V., Rozenberg G. The book of traces // Handbook Formal Languages, Beyond Words. N.Y.: Springer-Verlag. — 1997. — **3**. — 568 p.
7. Crochemore M., Mignosi F., Restivo A., Salemi S. Data compression using antidictionaries // Proc IEEE (Special issue on lossless data compression). — 2000. — **88**, N 11. — P. 1756–1768.
8. Mignosi F., Restivo A., Sciortino M. Forbidden factors and fragment assembly // LNCS. — 2002. — **2295**. — P. 349–358.
9. Beal M., Mignosi F., Restivo A., Sciortino M. Forbidden words in symbolic dynamics // Advances in Applied Mathematics. — 2000. — **25**, N 2. — P. 163–193.
10. Beal M., Mignosi F., Restivo A. Minimal forbidden words and symbolic dynamics // Proceedings of the 13th Annual Symp. on Theoretical Aspects of Comput. Sci. — 1996. — N 22–24. — P. 555–566.
11. Messner J. Pattern Matching in trace monoids // STACS. — 1997. — **97**. — P. 571–582.
12. Avellone A., Goldwurm M. Analysis of algorithms for the recognition of rational and context-free trace languages // Theoretical Inform. and Appl. — 1998. — **32**. — N 4–6. — P. 141–152.
13. Шахбазян К.В., Шукурян Ю.Г. Асинхронные автоматы, сравнивающие треки // Кибернетика и системный анализ. — 2012. — № 3. — P. 3–11.
14. Шахбазян К.В., Шукурян Ю.Г. Вхождения в моноиде треков // Там же. — 2010. — № 4. — P. 31–38.


*Поступила 09.07.2012*