МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2017 **№ 36**

Зарегистрирован в Федеральной службе по надзору
в сфере связи и массовых коммуникаций

Свидетельство о регистрации ПИ № ФС 77-33762 от 16 октября 2008 г.

Подписной индекс в объединённом каталоге «Пресса России» 38696

УЧРЕДИТЕЛЬ
Томский государственный университет

Адрес редакции и издателя: 634050, г. Томск, пр. Ленина, 36
E-mail: vestnik_pdm@mail.tsu.ru

В журнале публикуются результаты фундаментальных и прикладных научных исследований отечественных и зарубежных ученых, включая студентов и аспирантов, в области дискретной математики и её приложений в криптографии, компьютерной безопасности, кибернетике, информатике, программировании, теории надёжности, интеллектуальных системах.

Периодичность выхода журнала: 4 номера в год.

Редактор *Н. И. Шидловская*
Верстка *И. А. Панкратовой*

Подписано к печати 20.06.2017. Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 14,8. Уч.-изд. л. 16,3.
Тираж 300 экз. Заказ № 2609. Цена свободная. Дата выхода в свет 4.07.2017.

Отпечатано на оборудовании
Издательского Дома Томского государственного университета
634050, г. Томск, пр. Ленина, 36
Тел.: 8(3822)53-15-28, 52-98-49

СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

УДК 512.53

СЛУЧАЙНЫЕ УРАВНЕНИЯ НАД СВОБОДНЫМИ ПОЛУРЕШЁТКАМИ

М. А. Вахрамеев

Институт математики им. С. Л. Соболева СО РАН, г. Омск, Россия

Исследуются уравнения от одной переменной над свободными полурешётками. Установлено, что среднее число решений уравнения над свободной полурешёткой ранга n равно $\frac{3^n + 2 \cdot 2^n}{3 \cdot 2^n}$. Доказано, что среднее число неприводимых компонент алгебраических множеств, определяемых уравнениями над свободной полурешёткой счётного ранга, равно 1.

Ключевые слова: *свободная полурешётка, уравнение, неприводимые компоненты.*

DOI 10.17223/20710410/36/1

RANDOM EQUATIONS OVER FREE SEMILATTICES

M. A. Vakhrameev

Sobolev Institute of Mathematics, Omsk, Russia

E-mail: vahrmih@yandex.ru

In the paper, we study equations in one variable over free semilattices. We show that the average number of solutions of a random equation over a free semilattice of a rank n is equal to $\frac{3^n + 2 \cdot 2^n}{3 \cdot 2^n}$. It is proved that the average number of irreducible components of algebraic sets defined by equations over a free semilattice of a countable rank is equal to 1.

Keywords: *free semilattice, equation, irreducible components.*

Введение

В современной математике генерация и изучение свойств случайных алгебраических объектов является одним из важных и интересных направлений. Порождение случайных групповых уравнений рассматривалось в работах [1, 2], относительно разрешимости случайных уравнений в группах см. также [3]. С помощью работ [4, 5] понятие уравнения можно определить для произвольной алгебраической системы функционального языка. Таким образом, можно рассматривать и изучать характеристики уравнений над многими классами алгебраических систем, в частности над полурешётками [6].

В данной работе исследуются вероятностные характеристики множества решений уравнений над свободными полурешётками. Вычисляется среднее число решений случайно сгенерированного уравнения (теорема 3), а также среднее число неприводимых компонент алгебраического множества, определённого уравнением над свободной полурешёткой (теорема 4). Каждая теорема иллюстрируется примером для свободной полурешётки ранга 2.

1. Основные определения

Полурешёткой называется полугруппа, удовлетворяющая тождествам $xy = yx$ (коммутативности) и $xx = x$ (идемпотентности). В работе рассматривается свободная полурешётка F_n ранга n с множеством свободных порождающих $a_1, \ldots, a_n$ и с присоединённой единицей ε ($\forall x \in F_n\ (x\varepsilon = \varepsilon x = x)$).

Произвольный неединичный элемент a из F_n допускает единственное представление в виде произведения $a = a_{i_1}a_{i_2}\ldots a_{i_k}$, где $i_1 < i_2 < \ldots < i_k$, $k \leqslant n$. Число k будем называть *длиной* элемента a и обозначать $|a|$. Длину элемента ε полагаем равной 0.

Пусть $X = \{x_1,\ \ldots,\ x_m\}$ — множество переменных; F_n-термом $\tau(X)$ будем называть одно из следующих выражений: $w(X)c$, $w(X)$, c, где $w(X)$ — произведение переменных $x_i \in X$; c — элемент полурешётки F_n. *Уравнением* над F_n называется равенство двух F_n-термов: $s(X) = t(X)$, хотя бы один из которых содержит переменные. Элементы полурешётки F_n, входящие в запись уравнения, будем называть *константами*.

Решением уравнения $s(X) = t(X)$ называется набор значений переменных x_i, подстановка которых в уравнение $s(X) = t(X)$ обращает его в верное равенство. Уравнение, имеющее хотя бы одно решение, называется *совместным*. Уравнение, не имеющее решений, называется *несовместным*. Множество всех решений уравнения $s(X) = t(X)$ будем обозначать $V_{F_n}(s(X) = t(X))$.

Пример 1. Уравнение $a_1a_2x_1 = a_3a_4x_2$ является совместным. Одним из его решений является набор $x_1 = a_3a_4$, $x_2 = a_1a_2$. Уравнение $a_1a_2x_1 = a_3a_4$ является несовместным, поскольку при любом значении переменной x_1 в левой части уравнения будет присутствовать элемент a_2, который не содержится в правой части.

Все уравнения над полурешёткой F_n можно разделить на два типа:

1) уравнения, в которых переменные присутствуют только в одной части; будем называть такие уравнения *уравнениями* I *типа*;
2) уравнения, обе части которых содержат переменные — *уравнения* II *типа*.

Заметим, что любое уравнение II типа имеет решение над F_n. В самом деле, выбирая каждую из переменных x_k, входящих в уравнение, равной произведению констант левой и правой частей, получим верное равенство.

В работе рассматриваются уравнения над полурешёткой F_n от одной переменной. Множество всех таких уравнений обозначим через Eq_n. Будем обозначать: Eq_n^1 — множество всех уравнений I типа из Eq_n, Eq_n^2 — множество всех уравнений II типа из Eq_n. Уравнения из Eq_n^1 представимы либо в виде $c_1x = c_2$, $c_1, c_2 \in F_n$, либо в виде $c_3 = c_4x$, $c_3, c_4 \in F_n$, а уравнения из Eq_n^2 имеют вид $c_5x = c_6x$, $c_5, c_6 \in F_n$. Поскольку $|F_n| = 2^n$, легко видеть, что $|Eq_n^1| = 2 \cdot 2^n \cdot 2^n = 2 \cdot 4^n$, а $|Eq_n^2| = 2^n \cdot 2^n = 4^n$.

Все уравнения из множества Eq_n^2 совместны. Найдём количество совместных в Eq_n^1. Допустим, уравнение из Eq_n^1 имеет вид $c_1x = c_2$. Для того чтобы такое уравнение было совместным, необходимо, чтобы константа c_1 состояла только из букв, входя-

щих в состав константы c_2, и никаких других. Таким образом, количество совместных уравнений данного вида равно

$$\sum_{k=0}^{n} C_n^k \sum_{s=0}^{k} C_k^s = 3^n.$$

Здесь C_n^k — количество вариантов выбрать константу c_2, состоящую из k букв, а C_k^s — количество допустимых констант c_1, состоящих из s букв, входящих в состав c_2. Аналогично рассуждая для уравнений вида $c_3 = c_4 x$, получим, что количество совместных уравнений из Eq_n^1 равно $2 \cdot 3^n$.

2. Математическое ожидание числа решений случайно выбранного уравнения

Так как множества Eq_n^1 и Eq_n^2 конечны, на них можно ввести равномерную вероятностную меру и определить следующие случайные величины. Пусть ξ_1 — случайная величина, равная количеству решений случайно выбранного уравнения из Eq_n^1, а ξ_2 — случайная величина, равная количеству решений случайно выбранного уравнения из Eq_n^2. Отметим, что случайные величины ξ_1 и ξ_2 могут принимать целые значения от 0 до 2^n.

Теорема 1. Математическое ожидание числа решений случайно выбранного уравнения из Eq_n^1 равно 1:

$$\mathbb{E}\xi_1 = 1.$$

Доказательство. В Eq_n^1 содержится $2 \cdot 4^n$ уравнений, при этом только $2 \cdot 3^n$ из них являются совместными. Учитывая, что все варианты выбора уравнения равновероятны, получаем $\mathsf{P}[\xi_1 = 0] = \dfrac{2 \cdot 4^n - 2 \cdot 3^n}{2 \cdot 4^n} = \dfrac{4^n - 3^n}{4^n}$.

Далее, предположим, что уравнение $xa = b$, $a, b \in F_n$, имеет хотя бы одно решение. Это означает, что a может содержать только буквы из b. Допустим, что $|b| = s$, $|a| = t$, где $t \leqslant s \leqslant n$. Любое решение такого уравнения обязано содержать все буквы из b, не входящие в состав a, и при этом может содержать любые буквы из a. Следовательно, количество решений такого уравнения равно 2^t, т. е. однозначно определяется длиной элемента a. Получаем, что число всех уравнений, имеющих ровно 2^t решений, равно $C_n^t \sum\limits_{k=0}^{n-t} C_{n-t}^k = C_n^t 2^{n-t}$ (здесь C_n^t — число слов длины t, т. е. число всех вариантов выбора константы a, а $\sum\limits_{k=0}^{n-t} C_{n-t}^k$ — количество вариантов выбрать константу b, содержащую все буквы из a). Аналогичные рассуждения верны и для уравнений вида $c = xd$, $c, d \in F_n$. Таким образом, имеем

$$\mathsf{P}[\xi_1 = l] = \begin{cases} \dfrac{4^n - 3^n}{4^n}, & \text{если } l = 0; \\ \dfrac{2 \cdot C_n^i 2^{n-i}}{2 \cdot 4^n} = \dfrac{C_n^i 2^{n-i}}{4^n}, & \text{если } l = 2^i, \text{ где } 0 \leqslant i \leqslant n; \\ 0 & \text{иначе.} \end{cases} \tag{1}$$

Математическое ожидание ξ_1 равно

$$\mathbb{E}\xi_1 = 0 \cdot \frac{4^n - 3^n}{4^n} + \sum_{i=0}^{n} 2^i \frac{C_n^i 2^{n-i}}{4^n} = \frac{2^n}{4^n} \sum_{i=0}^{n} C_n^i = 1.$$

Теорема доказана. ■

Пример 2. Рассмотрим полурешётку F_2, порождённую множеством $\{a_1, a_2\}$, т. е. состоящую из элементов $\{\varepsilon, a_1, a_2, a_1a_2\}$. Выпишем все уравнения из множества Eq_2^1 (табл. 1).

Таблица 1

Уравнение	Количество решений	Уравнение	Количество решений
$x = \varepsilon$	1	$x = a_2$	1
$xa_1 = \varepsilon$	0	$xa_1 = a_2$	0
$xa_2 = \varepsilon$	0	$xa_2 = a_2$	2
$xa_1a_2 = \varepsilon$	0	$xa_1a_2 = a_2$	0
$x = a_1$	1	$x = a_1a_2$	1
$xa_1 = a_1$	2	$xa_1 = a_1a_2$	2
$xa_2 = a_1$	0	$xa_2 = a_1a_2$	2
$xa_1a_2 = a_1$	0	$xa_1a_2 = a_1a_2$	4
$\varepsilon = x$	1	$a_2 = x$	1
$\varepsilon = xa_1$	0	$a_2 = xa_1$	0
$\varepsilon = xa_2$	0	$a_2 = xa_2$	2
$\varepsilon = xa_1a_2$	0	$a_2 = xa_1a_2$	0
$a_1 = x$	1	$a_1a_2 = x$	1
$a_1 = xa_1$	2	$a_1a_2 = xa_1$	2
$a_1 = xa_2$	0	$a_1a_2 = xa_2$	2
$a_1 = xa_1a_2$	0	$a_1a_2 = xa_1a_2$	4

Составим ряд распределения случайной величины ξ_1:

ξ_1	0	1	2	3	4
P	14/32	8/32	8/32	0	2/32

Математическое ожидание $\mathbb{E}\xi_1 = 1 \cdot 8/32 + 2 \cdot 8/32 + 4 \cdot 2/32 = 1$ совпадает с результатом теоремы 1.

Теорема 2. Математическое ожидание числа решений случайно выбранного уравнения из Eq_n^2 равно

$$\mathbb{E}\xi_2 = \left(\frac{3}{2}\right)^n.$$

Доказательство. Поскольку $|Eq_n^2| = 4^n$, получаем

$$\mathbb{E}\xi_2 = \sum_{t=0}^{2^n} t\,\mathsf{P}[\xi_2 = t] = \sum_{t=0}^{2^n} \frac{tEq_n^2(t)}{4^n} = \frac{1}{4^n}\sum_{t=0}^{2^n} tEq_n^2(t),$$

где $Eq_n^2(t)$ — количество уравнений из Eq_n^2, имеющих ровно t решений.

Поскольку каждое уравнение участвует в полученной сумме ровно один раз, то

$$\frac{1}{4^n}\sum_{t=0}^{2^n} tEq_n^2(t) = \frac{1}{4^n}\sum_{s(x)=f(x)\in Eq_n^2} |V_{F_n}(s(x) = f(x))|,$$

где $|V_{F_n}(s(x) = f(x))|$ — количество решений уравнения $s(x) = f(x) \in Eq_n^2$.

Обозначим через $r_n(a)$ количество уравнений, удовлетворяющих точке $a \in F_n$. Изменив порядок суммирования, получаем

$$\mathbb{E}\xi_2 = \frac{1}{4^n}\sum_{s(X)=f(X)\in Eq_n^2} |V_{F_n}(s(x) = f(x))| = \frac{1}{4^n}\sum_{a\in F_n} r_n(a).$$

Пусть $xb = xc$ — произвольное уравнение из Eq_n^2 (все уравнения этого типа совместны) и $a \in F^n$ — его решение. Представим константы b и c в виде $b = da'$, $c = da''$, где d не содержит букв из a, a' и a'' могут содержать только буквы из a. Предположим, что $|a| = l$. Тогда a' и a'' — подмножества некоторого слова из l букв, d — подмножество некоторого слова из $n-l$ букв. Количество уравнений, удовлятворяющих точке a, равно

$$r_n(a) = \sum_{i=0}^{n-l} C_{n-l}^i \sum_{j=0}^{l} C_l^j \sum_{j=0}^{l} C_l^j = 2^{n-l} \cdot 2^l \cdot 2^l = 2^{n+l}.$$

Подставляя этот результат в выражение для математического ожидания, получаем

$$\mathbb{E}\xi_2 = \frac{1}{4^n} \sum_{a \in F_n} r_n(a) = \frac{1}{4^n} \sum_{l=0}^{n} C_n^l 2^{n+l} = \frac{1}{4^n} \cdot 2^n \sum_{l=0}^{n} C_n^l 2^l = \frac{1}{2^n} \cdot 3^n = \left(\frac{3}{2}\right)^n.$$

Теорема доказана. ■

Пример 3. Вновь рассмотрим полурешётку F_2 и выпишем уравнения из Eq_n^2 (табл. 2).

Таблица 2

Уравнение	Количество решений	Уравнение	Количество решений
$x = x$	4	$x = xa_2$	2
$xa_1 = x$	2	$xa_1 = xa_2$	1
$xa_2 = x$	2	$xa_2 = xa_2$	4
$xa_1a_2 = x$	1	$xa_1a_2 = xa_2$	2
$x = xa_1$	2	$x = xa_1a_2$	1
$xa_1 = xa_1$	4	$xa_1 = xa_1a_2$	2
$xa_2 = xa_1$	1	$xa_2 = xa_1a_2$	2
$xa_1a_2 = xa_1$	2	$xa_1a_2 = xa_1a_2$	4

Составим ряд распределения случайной величины ξ_2:

ξ_2	0	1	2	3	4
P	0	4/16	8/16	0	4/16

Видим, что математическое ожидание $\mathbb{E}\xi_2 = 1 \cdot 4/16 + 2 \cdot 8/16 + 4 \cdot 4/16 = 36/16 = 9/4$ совпадает с результатом теоремы 2.

Теперь найдём математическое ожидание случайной величины ξ, равной количеству решений случайно выбранного уравения из Eq_n.

Теорема 3. Математическое ожидание числа решений случайно выбранного уравнения из Eq_n равно

$$\mathbb{E}\xi = \frac{3^n + 2 \cdot 2^n}{3 \cdot 2^n}.$$

Доказательство. Случайная величина ξ может быть представлена в виде $\xi = \xi' + \xi''$, где

$$\xi' = \begin{cases} 0, & \text{если уравнение из } Eq_n^2, \\ \xi_1 & \text{иначе;} \end{cases} \qquad \xi'' = \begin{cases} 0, & \text{если уравнение из } Eq_n^1, \\ \xi_2 & \text{иначе.} \end{cases}$$

Поскольку $|Eq_n^1| = 2|Eq_n^2|$, то $\mathbb{E}\xi' = \frac{1}{3}\cdot 0 + \frac{2}{3}\,\mathbb{E}\xi_1 = \frac{2}{3}\,\mathbb{E}\xi_1$ и $\mathbb{E}\xi'' = \frac{2}{3}\cdot 0 + \frac{1}{3}\,\mathbb{E}\xi_2 = \frac{1}{3}\,\mathbb{E}\xi_2$. Используя теоремы 1 и 2, получаем

$$\mathbb{E}\xi = \mathbb{E}\xi' + \mathbb{E}\xi'' = \frac{2}{3}\,\mathbb{E}\xi_1 + \frac{1}{3}\,\mathbb{E}\xi_2 = \frac{2}{3} + \frac{1}{3}\left(\frac{3}{2}\right)^n = \frac{3^n + 2\cdot 2^n}{3\cdot 2^n}.$$

Теорема доказана. ■

Пример 4. Выпишем все уравнения от одной переменной над полурешёткой F_2 (табл. 3).

Таблица 3

Уравнения	Количество решений	Уравнение	Количество решений
$x = \varepsilon, \varepsilon = x$	1	$x = x$	4
$xa_1 = \varepsilon, \varepsilon = xa_1$	0	$xa_1 = x$	2
$xa_2 = \varepsilon, \varepsilon = xa_2$	0	$xa_2 = x$	2
$xa_1a_2 = \varepsilon, \varepsilon = xa_3$	0	$xa_1a_2 = x$	1
$x = a_1, a_1 = x$	1	$x = xa_1$	2
$xa_1 = a_1, a_1 = xa_1$	2	$xa_1 = xa_1$	4
$xa_2 = a_1, a_1 = xa_2$	0	$xa_2 = xa_1$	1
$xa_1a_2 = a_1, a_1 = xa_1a_2$	0	$xa_1a_2 = xa_1$	2
$x = a_2, a_2 = x$	1	$x = xa_2$	2
$xa_1 = a_2, a_2 = xa_1$	0	$xa_1 = xa_2$	1
$xa_2 = a_2, a_2 = xa_2$	2	$xa_2 = xa_2$	4
$xa_1a_2 = a_2, a_2 = xa_1a_2$	0	$xa_1a_2 = xa_2$	2
$x = a_1a_2, a_1a_2 = x$	1	$x = xa_1a_2$	1
$xa_1 = a_1a_2, a_1a_2 = xa_1$	2	$xa_1 = xa_1a_2$	2
$xa_2 = a_1a_2, a_1a_2 = xa_2$	2	$xa_2 = xa_1a_2$	2
$xa_1a_2 = a_1a_2, a_1a_2 = xa_1a_2$	4	$xa_1a_2 = xa_1a_2$	4

Составим таблицу распределения случайной величины ξ:

ξ	0	1	2	3	4
P	14/48	12/48	16/48	0	6/48

Математическое ожидание $\mathbb{E}\xi = 1\cdot 12/48 + 2\cdot 16/48 + 4\cdot 6/48 = 68/48 = 17/12$, очевидно, соответствует теореме 3.

3. Математическое ожидание числа неприводимых компонент множества решений случайно выбранного уравнения

Пусть F — свободная полурешётка счётного ранга с множеством свободных порождающих элементов $\{a_i : i \in I\}$ с присоединённой единицей ε. Множество $Y \subseteq F$ называется *алгебраическим*, если существует система уравнений над F с множеством решений Y (множеством решений системы уравнений является пересечение решений всех уравнений системы).

Непустое алгебраическое множество Y называется *неприводимым*, если его нельзя представить в виде

$$Y = Y_1 \cup Y_2 \cup \cdots \cup Y_n, \tag{2}$$

где Y_i — алгебраические множества, $Y_i \not\subseteq Y_j$ для всех $i \neq j$ и $n > 1$. Если алгебраическое множество представимо в виде объединения (2), где множества Y_i неприводимы, то множества Y_i называются *неприводимыми компонентами* множества Y.

Будем рассматривать уравнения над полурешёткой F, которые в своей записи содержат только константы из полурешётки F_n. Например, для $n = 2$ будем рассматривать те уравнения над F, в записи которых могут присутствовать только константы $\varepsilon, a_1, a_2, a_1a_2$, т. е. элементы полурешётки F_2.

Определим случайную величину ψ, равную количеству неприводимых компонент множества решений случайно выбранного уравнения над F с константами из F_n.

Теорема 4. Математическое ожидание числа неприводимых компонент множества решений случайно выбранного уравнения над F с константами из F_n равно

$$\mathbb{E}\psi = 1.$$

Доказательство. Рассмотрим уравнения I типа, т. е. уравнения вида $xa = a'$ или $a' = xa$, где $a, a' \in F_n$. Множество решений любого такого уравнения конечно, потому что x может состоять только из букв, входящих в состав a'. Это означает, что случайная величина ψ для любого из уравнений данного вида равна количеству решений этого уравнения. Таким образом, для уравнений I типа ψ может принимать следующие значения: $0, 1, 2, 4, \ldots, 2^k, \ldots, 2^n$.

Теперь рассмотрим уравнения II типа. Представим их в виде $xab = xa'b$, где $a, a', b \in F_n$ и $a \cap a' = \varnothing$. Поскольку любое решение такого уравнения имеет вид $x = aa't$, координатная полурешётка множества решений $V(xab = xa'b)$ вкладывается в полурешётку F^*, порождённую множеством $\{a_1 \ldots a_n \ldots\} \cup \{t\}$. В таком случае, как следует из работы [7], множество $V(xab = xa'b)$ является неприводимым. Следовательно, $\psi = 1$ для любого уравнения II типа.

Обобщим вышесказанное. Имеем, что $\psi = 0$ для $2(4^n - 3^n)$ несовместных уравнений I типа. Далее, $\psi = 1$ для всех уравнений II типа и для уравнений I типа, имеющих ровно одно решение, т. е. для $4^n + 2C_n^0 2^n$ уравнений. Наконец, по формуле (1) $\psi = 2^i$ для $2C_n^i 2^{n-i}$ уравнений I типа, где $i = 1, 2, \ldots, n$. Таким образом, получаем

$$\mathbb{E}\psi = 0 \cdot \frac{2(4^n - 3^n)}{3 \cdot 4^n} + 1 \cdot \frac{4^n + 2C_n^0 \cdot 2^n}{3 \cdot 4^n} + \sum_{i=1}^{n} 2^i \frac{2C_n^i 2^{n-i}}{3 \cdot 4^n} = \frac{1}{3} + \frac{2C_n^0 2^n}{3 \cdot 4^n} + \sum_{i=1}^{n} 2^i \frac{2C_n^i 2^{n-i}}{3 \cdot 4^n} =$$

$$= \frac{1}{3} + \sum_{i=0}^{n} 2^i \frac{2C_n^i 2^{n-i}}{3 \cdot 4^n} = \frac{1}{3} + \frac{2}{3} \sum_{i=0}^{n} \frac{C_n^i \cdot 2^n}{4^n} = \frac{1}{3} + \frac{2}{3} \cdot \frac{\sum\limits_{i=0}^{n} C_n^i}{2^n} = \frac{1}{3} + \frac{2}{3} = 1.$$

Теорема доказана. ■

ЛИТЕРАТУРА

1. *Gilman R., Myasnikov A., and Roman'kov V.* Random equations in nilpotent groups // J. Algebra. 2012. V. 352. No. 1. P. 192–214.
2. *Gilman R., Myasnikov A., and Roman'kov V.* Random equations in free groups // Groups Complexity Cryptol. 2011. V. 3. No. 2. P. 257–284.
3. *Roman'kov V.* Equations over groups // Groups Complexity Cryptol. 2012. V. 4. No. 2. P. 191–239.
4. *Daniyarova E., Miasnikov A., and Remeslennikov V.* Unification theorems in algebraic geometry // Algebra and Discrete Mathematics. Hackensack, 2008. P. 80–112.
5. *Даниярова Э. Ю., Мясников А. Г., Ремесленников В. Н* Алгебраическая геометрия над алгебраическими системами. II. Основания // Фундамент. и прикл. матем. 2012. Т. 17. № 1. С. 65–106.
6. *Шевляков А. Н.* Эквивалентные уравнения над полурешетками // Сиб. электрон. матем. изв. 2016. Т. 13. С. 478–490.

7. *Шевляков А. Н.* Элементы алгебраической геометрии над свободной полурешеткой // Алгебра и логика. 2015. Т. 54. № 3. С. 399–420.

REFERENCES

1. *Gilman R., Myasnikov A., and Roman'kov V.* Random equations in nilpotent groups. J. Algebra, 2012, vol. 352, no. 1, pp. 192–214.
2. *Gilman R., Myasnikov A., and Roman'kov V.* Random equations in free groups. Groups Complexity Cryptol., 2011, vol. 3, no. 2, pp. 257–284.
3. *Roman'kov V.* Equations over groups. Groups Complexity Cryptol., 2012, vol. 4, no. 2, pp. 191–239.
4. *Daniyarova E., Miasnikov A., and Remeslennikov V.* Unification theorems in algebraic geometry. Algebra and Discrete Mathematics. Hackensack, 2008, pp. 80–112.
5. *Daniyarova E. Yu., Myasnikov A. G., and Remeslennikov V. N* Algebraicheskaya geometriya nad algebraicheskimi sistemami. II. Osnovaniya [Algebraic geometry over algebraic structures. II. Foundations]. Fundam. Prikl. Mat., 2012, vol. 17, iss. 1, pp. 65–106. (in Russian)
6. *Shevlyakov A. N.* Ekvivalentnye uravneniya nad polureshetkami [Equivalent equations in semilattices]. Sib. Elektron. Mat. Izv., 2016, vol. 13, pp. 478–490. (in Russian)
7. *Shevlyakov A. N.* Elementy algebraicheskoy geometrii nad svobodnoy polureshetkoy [Elements of algebraic geometry over a free semilattice]. Algebra Logika, 2015, vol. 54, no. 3, pp. 399–420. (in Russian)

УДК 519.21

УЛУЧШЕНИЕ ОЦЕНКИ СКОРОСТИ СХОДИМОСТИ В МНОГОМЕРНОЙ ЦЕНТРАЛЬНОЙ ПРЕДЕЛЬНОЙ ТЕОРЕМЕ ДЛЯ СУММ ЛОКАЛЬНО ЗАВИСИМЫХ СЛУЧАЙНЫХ ВЕКТОРОВ

А. В. Волгин

Московский государственный университет информационных технологий, радиотехники и электроники, г. Москва, Россия

Приведена улучшенная явная оценка скорости многомерной нормальной аппроксимации сумм локально зависимых случайных векторов. При любой фиксации размерности векторов и числа слагаемых данный результат позволяет вычислять численную оценку скорости сходимости. Установлено, что главный член полученной оценки для сумм независимых и одинаково распределённых случайных векторов имеет порядок $d^{9/2}n^{-1/2}\ln n$, где d — размерность векторов и n — число слагаемых. Приведено применение результата в задаче о нормальной аппроксимации числа рёбер, инцидентных вершинам одного цвета, в регулярном графе.

Ключевые слова: *многомерная центральная предельная теорема, локально зависимые случайные векторы.*

DOI 10.17223/20710410/36/2

IMPROVING THE RATE OF CONVERGENCE IN THE MULTIDIMENSIONAL CENTRAL LIMIT THEOREM FOR SUMS OF LOCALLY DEPENDENT RANDOM VECTORS

A. V. Volgin

Moscow Technological University, Moscow, Russia

E-mail: artem.volgin@bk.ru

Estimates of the rate of convergence in multidimensional limit theorems for sums of dependent random vectors are considered in many papers. The types of dependence in a sequence of random vectors can be different, for example, m-dependent and locally dependent sequences of random vectors. It is important that these estimates are implicit. They do not specify how the estimate depends on the dimension of random vectors.

In this connection, in one of the author's previous papers, an explicit estimate for the distance between a multidimensional normal distribution and the distribution of the sum of locally dependent random vectors was obtained. In this paper, we improve this estimate. Also, it is proved that for centered and normalized sums of independent random vectors, the order of this estimate is equal to $d^{9/2}n^{-1/2}\ln n$, where d is dimension and n is number of vectors.

Results of this paper have applications for discrete mathematical objects. For example, in the paper we consider a fixed regular graph. Each vertex is independently assigned one of the colors with a certain probability. A condition for the normal approximation of the number of edges incident to vertices of the same color is obtained.

Keywords: *multivariate CLT, locally dependent random vectors.*

Введение

Оценки скорости сходимости сумм случайных векторов с различными типами зависимостей рассматриваются во многих работах [1 – 5]. В [1, 2] приводятся оценки для сумм независимых и m-зависимых случайных векторов соответственно. В [3 – 5] с использованием метода Стейна рассматриваются оценки для более сложных типов зависимостей. Данные результаты имеют приложения для дискретных математических объектов. В п. 3 приводится применение основного результата данной работы в задаче о раскраске вершин регулярного графа.

Особенность оценок из упомянутых выше работ заключается в том, что для них не указывается явный вид зависимости от размерности векторов. Это обусловлено тем, что задача аппроксимации рассматривается для случая фиксированной размерности d и неограниченного роста числа слагаемых n; $d, n \in \mathbb{N}$.

В работе [6] анонсирован, а в [7] приведён результат об уточнении оценки скорости многомерной нормальной аппроксимации сумм локально зависимых случайных векторов, которая получена в [5]. Под уточнением подразумевается нахождение явного вида зависимости оценки от размерности d. При этом в [7] доказано, что главный член полученной оценки имеет порядок

$$(2\pi)^{d/2} d^3 n^{-1/2} \ln n. \tag{1}$$

В настоящей работе приводится улучшение оценки (1), главный член которой имеет порядок

$$d^{9/2} n^{-1/2} \ln n.$$

Далее кратко рассмотрим основные понятия и определения, введённые в [5, 7].

В [5] рассматривается оценка скорости сходимости в многомерной центральной предельной теореме для сумм зависимых ограниченных случайных векторов

$$\begin{gathered} W = \sum_{i=1}^{n} X_i, \quad X_i \in \mathbb{R}^d, \quad d, n \in \mathbb{N}, \\ |X_i| \leqslant B, \quad i = 1, \ldots, n, \end{gathered} \tag{2}$$

где через $|\cdot|$ обозначается сумма абсолютных значений всех элементов вектора (или матрицы); $B \in \mathbb{R}$ — неотрицательная величина, которая может зависеть от n и d.

Структура зависимости определяется следующим образом. Рассмотрим множество индексов $J = \{1, \ldots, n\}$. Для каждого $i \in J$ сумма W представляется двумя способами:

$$W = U_i + V_i \quad \text{и} \quad W = R_i + T_i, \tag{3}$$

при этом предполагается, что существуют неотрицательные величины $A_1, A_2 \in \mathbb{R}$, которые могут зависеть от n и d, такие, что

$$|U_i| \leqslant A_1, \quad |R_i| \leqslant A_2 \quad \text{и} \quad A_1 \leqslant A_2. \tag{4}$$

Интересующий нас результат работы [5] получен с использованием метода Стейна и заключается в оценке величины

$$\gamma = \sup_{h \in \mathscr{H}} |\mathbf{E}h(W) - \mathbf{E}h(Z)|,$$

где $\mathscr{H} = \{h : \mathbb{R}^d \to [-1, 1]\}$ — класс измеримых функций; Z — случайный вектор, имеющий d-мерное стандартное нормальное распределение.

Заметим, что для класса $\mathscr{H}_1$ индикаторов измеримых выпуклых подмножеств $\mathbb{R}^d$ справедливо выражение

$$\gamma = \sup_{h\in\mathscr{H}_1} |\mathbf{E}h(W) - \mathbf{E}h(Z)| = \sup_{F\in\mathscr{F}} \big|\mathbf{P}(W\in F) - \mathbf{P}(Z\in F)\big|,$$

где $\mathscr{F}$ — класс измеримых выпуклых подмножеств $\mathbb{R}^d$.

1. Основная теорема о точности многомерной нормальной аппроксимации

Обозначим: I — единичная матрица над $\mathbb{R}$ размера $d\times d$; A^{T} — матрица, транспонированная к матрице A; для $i=1,\dots,n$ пусть X_i, V_i, U_i, T_i — вектор-столбцы, $X_i^{\mathrm{T}}, V_i^{\mathrm{T}}, U_i^{\mathrm{T}}, T_i^{\mathrm{T}}$ — вектор-строки. Введём также следующие обозначения:

$$\chi_1 = \sum_{i=1}^n \mathbf{E}\big|\mathbf{E}(X_i|V_i)\big|, \quad \chi_2 = \sum_{i=1}^n \mathbf{E}\big|\mathbf{E}(X_iU_i^{\mathrm{T}}) - \mathbf{E}(X_iU_i^{\mathrm{T}}|T_i)\big|, \quad \chi_3 = \left|I - \sum_{i=1}^n \mathbf{E}(X_iU_i^{\mathrm{T}})\right|.$$

Теорема 1 [5]. Пусть $W=\sum\limits_{i=1}^n X_i$ — сумма ограниченных случайных векторов, таких, что имеют место представления (2)–(4). Тогда существует такая величина c, зависящая только от размерности векторов d, что

$$\gamma \leqslant c\big(aA_2 + naA_1A_2B(|\ln A_2B| + \ln n) + \chi_1 + (|\ln A_1B| + \ln n)(\chi_2+\chi_3)\big),$$

где a — величина, зависящая от конкретного класса функций $\mathscr{H}$.

Выбор представления (3) играет существенную роль в смысле уменьшения оценки сверху величины γ. Если, например, для любого $i\in J$ выбрать в качестве T_i вектор — константу, то величина χ_2 окажется равной нулю, однако при этом может увеличиться A_2. Если $\mathbf{E}W=0$ и ковариационная матрица вектора W является единичной, то при выборе $V_i=0$ получим, что $\chi_3=0$, однако это может привести к увеличению A_1.

В [5] указывается на возможность применения теоремы 1 для сумм *локально зависимых* ограниченных случайных векторов. Под этим подразумевается, что для любого $i\in J$ существуют множества $J_1^{(i)}, J_2^{(i)}$, $\varnothing \neq J_1^{(i)} \subseteq J_2^{(i)} \subset J$, для которых выполнены два условия: случайный вектор X_i не зависит от совокупности случайных векторов $\{X_j : j\in J\setminus J_1^{(i)}\}$ и множество случайных векторов $\{X_j : j\in J_1^{(i)}\}$ не зависит от совокупности случайных векторов $\{X_j : j\in J\setminus J_2^{(i)}\}$. Если для всех $i\in J$ выбрать $U_i = \sum\limits_{j\in J_1^{(i)}} X_j$, $R_i = \sum\limits_{j\in J_2^{(i)}} X_j$, $V_i = W - U_i$ и $T_i = W - R_i$, то в случае, когда случайный вектор W является суммой одинаково распределённых случайных векторов, имеет нулевое среднее и единичную ковариационную матрицу, величины в теореме 1 равны $\chi_1=\chi_2=\chi_3=0$, $A_1 = \max\limits_{i\in J}\Big(\mathrm{card}(J_1^{(i)})\Big)\,B$ и $A_2 = \max\limits_{i\in J}\Big(\mathrm{card}(J_2^{(i)})\Big)\,B$, где через $\mathrm{card}(\cdot)$ обозначается мощность множества. Для случая сумм независимых случайных векторов естественным является выбор $U_i=R_i=X_i$, $i=1,\dots,n$. В этом случае при центрировании и нормировании W оценка γ имеет порядок $n^{-1/2}\ln n$ [5].

Обозначим через α_d^2 квантиль уровня $7/8$ распределения хи-квадрат с d степенями свободы. Сформулируем основную теорему настоящей работы.

Теорема 2. Пусть $W=\sum\limits_{i=1}^n X_i$ — сумма случайных векторов, удовлетворяющих условиям (2)–(4) и $K = 20{,}28\,nA_1A_2B$, $|K|<1$. Тогда если $\mathscr{H} = \{h:\mathbb{R}^d\to[-1,1]\}$ — класс измеримых функций, то верно неравенство

$$\gamma \leqslant aA_2 + Kad\,|\ln K| + 4d\chi_1 + 4d^2\,|\ln K|\,(\chi_2+\chi_3) + 5a\alpha_d\frac{K}{1-K^2}, \tag{5}$$

где a — величина, зависящая от конкретного класса функций $\mathscr{H}$.

2. Доказательство теоремы 2

Вычислим те величины, которые в [5] при доказательстве теоремы 1 не определяются явно. Рассмотрим многомерный аналог уравнения Стейна [1, 5] относительно неизвестной функции $\Psi(x)$:

$$\Delta\Psi(x) - x\nabla\Psi(x) = h(x) - \mathbf{E}h(Z), \quad \Psi(x) : \mathbb{R}^d \to \mathbb{R}, \quad x \in \mathbb{R}^d, \tag{6}$$

где $\Psi(x)$ — трижды дифференцируема на $\mathbb{R}$; $\Delta\Psi(x) = \sum\limits_{i=1}^{d} \dfrac{\partial^2\Psi}{\partial x_i^2}(x)$ и $\nabla\Psi(x) = \left(\dfrac{\partial\Psi}{\partial x_1}, \ldots, \dfrac{\partial\Psi}{\partial x_d}\right)(x)$ — лапласиан и градиент функции $\Psi(x)$ соответственно.

Обозначим через Φ функцию распределения стандартного нормального закона. Рассмотрим функцию

$$h_s(x) = \int_{\mathbb{R}^d} h(s^{1/2}y + (1-s)^{1/2}x)\, d\Phi_d(y), \quad s \in (0,1),\ x, y \in \mathbb{R}^d,$$

где $\Phi_d(y) = \Phi(y_1) \cdot \ldots \cdot \Phi(y_d)$.

Как замечено в [5], если в уравнение (6) вместо h подставить функцию h_s, то оно будет иметь решение

$$\Psi_t(x) = -\frac{1}{2}\int_t^1 [h_s(x) - \mathbf{E}h(Z)]\,\frac{ds}{1-s}, \quad s, t \in (0,1),\ x \in \mathbb{R}^d.$$

Подставим данное решение в уравнение (6):

$$\Delta\Psi_t(W) - W\nabla\Psi_t(W) = h(W) - \mathbf{E}h(Z).$$

Согласно обобщению метода Стейна, в многомерном случае для того, чтобы оценить величину $\gamma = \sup\limits_{h\in\mathscr{H}}\big|\mathbf{E}h(W) - \mathbf{E}h(Z)\big|$, нужно оценить величину $\sup\limits_{h\in\mathscr{H}}\mathbf{E}\big|\Delta\Psi_t(W) - W\nabla\Psi_t(W)\big|$. Рассмотрим равенство

$$\gamma_t = \sup_{h\in\mathscr{H}}\big|\mathbf{E}h_t(W) - \mathbf{E}h_t(Z)\big|, \quad t \in (0,1).$$

В [1] приводится следующая лемма о связи γ и γ_t.

Лемма 1 [1]. Пусть Q — произвольная вероятностная мера, заданная на $\mathbb{R}^d$, в соответствии с которой выбирается вектор W. Тогда для любого $t \in (0,1)$ верно неравенство

$$\sup_{h\in\mathscr{H}}\big|\mathbf{E}h(W) - \mathbf{E}h(Z)\big| \leqslant \frac{4}{3}\sup_{h\in\mathscr{H}}\big|\mathbf{E}h_t(W) - \mathbf{E}h_t(Z)\big| + a\frac{5}{2}\alpha_d\frac{\sqrt{t}}{1-t},$$

где a — величина, зависящая от класса $\mathscr{H}$.

Из леммы 1 следует, что γ и γ_t связаны неравенством

$$\gamma \leqslant \frac{4}{3}\gamma_t + a\frac{5}{2}\alpha_d\frac{\sqrt{t}}{1-t}, \quad t \in (0,1). \tag{7}$$

Далее оценим величину γ_t. Обозначим через $\Psi_t^{(1)}(x)$ и $\Psi_t^{(2)}(x)$ градиент и матрицу вторых производных функции $\Psi_t(x)$ соответственно. Элементы градиента $\dfrac{\partial\Psi_t}{\partial x_p}(x)$,

$p = 1, \ldots, d$, обозначим через $\Psi^{(1)}_{t(p)}(x)$, элементы матрицы вторых производных $\dfrac{\partial^2 \Psi_t}{\partial x_p \partial x_q}(x)$, $p, q = 1, \ldots, d$, — через $\Psi^{(2)}_{t(pq)}(x)$. В [7] доказана следующая лемма, в которой приводятся оценки для $|\Psi^{(1)}_t(x)|$ и $|\Psi^{(2)}_t(x)|$.

Лемма 2 [7]. Для любой функции $h : \mathbb{R}^d \to [-1, 1]$ при $t \in (0, 1)$ справедливы неравенства

$$\sup_{x \in \mathbb{R}^d} \left|\Psi^{(1)}_t(x)\right| < d\sqrt{\frac{\pi}{2}},$$

$$\sup_{x \in \mathbb{R}^d} \left|\Psi^{(2)}_t(x)\right| < \left[\frac{d^2}{\pi} + d\left(\sqrt{\frac{2}{\pi e}} - \frac{1}{\pi}\right)\right] \ln(t^{-1}). \tag{8}$$

Обозначим через $\mathrm{Tr}(M)$ след матрицы M. Тогда верны равенства

$$\mathbf{E}h_t(W) - \mathbf{E}h_t(Z) = \mathbf{E}[\Delta\Psi_t(W) - W\nabla\Psi_t(W)] = \mathscr{A} - \mathscr{B} - \mathscr{C} + \mathscr{D},$$

где

$$\mathscr{A} = \mathbf{E}\mathrm{Tr}\left[\Psi^{(2)}_t(W)\left(I - \sum_{j=1}^n X_j U_j^{\mathrm{T}}\right)\right],$$

$$\mathscr{B} = \sum_{j=1}^n \mathbf{E}\left[X_j^{\mathrm{T}} \cdot \nabla\Psi_t(V_j)\right],$$

$$\mathscr{C} = \sum_{j=1}^n \mathbf{E}\left[X_j^{\mathrm{T}}\left(\nabla\Psi_t(W) - \nabla\Psi_t(V_j) - \Psi^{(2)}_t(V_j)U_j\right)\right]; \tag{9}$$

$$\mathscr{D} = \sum_{j=1}^n \mathbf{E}\mathrm{Tr}\left[X_j U_j^{\mathrm{T}}\left(\Psi^{(2)}_t(W) - \Psi^{(2)}_t(V_j)\right)\right]. \tag{10}$$

Заметим, что для оценки величины γ_t нужно оценить сумму $\left|\mathscr{A}\right| + \left|\mathscr{B}\right| + \left|\mathscr{C}\right| + \left|\mathscr{D}\right|$. Рассмотрим следующую лемму, доказанную в работе [7], о представлении в явном виде зависимости интересующих нас величин от размерности d. Обозначим через $\Psi^{(3)}_{t(pqr)}$ частную производную третьего порядка функции $\Psi_t(x)$ по переменным x_p, x_q и x_r, $p, q, r = 1, \ldots, d$.

Лемма 3 [7]. Пусть W, V и U — случайные векторы, принимающие значения в $\mathbb{R}^d$, $d \in \mathbb{N}$, $W = V + U$, и пусть Y — произвольная случайная величина. Предположим, что $|U| \leqslant C_1$, $|Y| \leqslant C_2$, где $C_1, C_2 \in \mathbb{R}$. Тогда для любого $\tau \in (0, 1)$ и $h \in \mathscr{H}$ верно неравенство

$$\left|\mathbf{E}Y\Psi^{(3)}_{t(pqr)}(V + \tau U)\right| < C_2\left(3{,}04\gamma/\sqrt{t} + 1{,}52aC_1/\sqrt{t} + ad\left|\ln t\right|\right),$$

где a — величина, зависящая от конкретного класса функций $\mathscr{H}$.

Для доказательства леммы 3 в [7] приводится следующая вспомогательная лемма.

Лемма 4 [7]. Пусть $\phi^{(3)}_{(pqr)} : \mathbb{R}^d \to \mathbb{R}$ — частная производная третьего порядка плотности распределения d-мерного стандартного нормального закона по переменным y_p, y_q и y_r, $p, q, r = 1, \ldots, d$. Тогда

а) для любых $p, q, r = 1, \ldots, d$

$$\int_{\mathbb{R}^d} |\phi^{(3)}_{(pqr)}(y)|dy < 1{,}52;$$

б) для любых $p, q, r = 1, \ldots, d$

$$\int_{\mathbb{R}^d} |y| \cdot |\phi^{(3)}_{(pqr)}(y)| dy < 1{,}21d + 0{,}65.$$

Замечание 1. В работе [7] приведены более грубые оценки выражений в леммах 3 и 4. При проведении дополнительных исследований удалось улучшить данные оценки.

Далее оценим величины $\mathscr{A}, \mathscr{B}, \mathscr{C}$ и $\mathscr{D}$.

1. Оценим величину $\left|\mathscr{C}\right|$. Обозначим через X_{jp} и U_{jp} координаты с номером $p = 1, \ldots, d$ векторов X_j и U_j соответственно. С учётом обозначений для градиента и матрицы вторых производных функции $\Psi_t(x)$ рассмотрим выражение, которое является частью формулы для $\mathscr{C}$:

$$\begin{gathered}\nabla\Psi_t(W) - \nabla\Psi_t(V_j) = \nabla\Psi_t(V_j + U_j) - \nabla\Psi_t(V_j) = \\ = \left(\Psi^{(1)}_{t(1)}(V_j + U_j) - \Psi^{(1)}_{t(1)}(V_j), \ldots, \Psi^{(1)}_{t(d)}(V_j + U_j) - \Psi^{(1)}_{t(d)}(V_j)\right)^{\mathrm{T}}.\end{gathered} \tag{11}$$

Применим формулу Тейлора с интегральной формой остаточного члена [8] в окрестности точки V_j к каждой координате вектора в правой части равенств (11):

$$\begin{gathered}\Psi^{(1)}_{t(p)}(V_j + U_j) - \Psi^{(1)}_{t(p)}(V_j) = U_{j1}\Psi^{(2)}_{t(p1)}(V_j) + \ldots + U_{jd}\Psi^{(2)}_{t(pd)}(V_j) + \\ + \int_0^1 (1-\tau) \sum_{q,r=1}^{d} \Psi^{(3)}_{t(pqr)}(V_j + \tau U_j) U_{jq} U_{jr} d\tau, \quad p = 1, \ldots, d,\ \tau \in (0,1).\end{gathered} \tag{12}$$

Теперь рассмотрим другое выражение, являющееся частью формулы для $\mathscr{C}$:

$$\Psi^{(2)}_t(V_j)U_j = \left(U_{j1}\Psi^{(2)}_{t(11)}(V_j) + \ldots + U_{jd}\Psi^{(2)}_{t(1d)}(V_j), \ldots, U_{j1}\Psi^{(2)}_{t(d1)}(V_j) + \ldots + U_{jd}\Psi^{(2)}_{t(dd)}(V_j)\right)^{\mathrm{T}}.$$

С учётом последнего равенства и формул (11), (12) рассмотрим следующие равенства:

$$\begin{gathered}\nabla\Psi_t(W) - \nabla\Psi_t(V_j) - \Psi^{(2)}_t(V_j)U_j = \\ = \left(\int_0^1 (1-\tau) \sum_{q,r=1}^{d} \Psi^{(3)}_{t(1qr)}(V_j + \tau U_j) d\tau, \ldots, \int_0^1 (1-\tau) \sum_{q,r=1}^{d} \Psi^{(3)}_{t(dqr)}(V_j + \tau U_j) d\tau\right)^{\mathrm{T}}.\end{gathered}$$

Учитывая последнее равенство, а также (9), получаем

$$\mathscr{C} = \sum_{j=1}^{n} \mathbf{E} \int_0^1 (1-\tau) \sum_{p,q,r=1}^{d} \Psi^{(3)}_{t(pqr)}(V_j + \tau U_j) X_{jp} U_{jq} U_{jr} d\tau,\ \tau \in (0,1).$$

Применяя лемму 3 при $U = U_j, Y = U_{jp}U_{jq}X_{jr}, j, p, q, r = 1, \ldots, d$, получаем следующую оценку:

$$\left|\mathscr{C}\right| \leqslant \frac{1}{2} n A_1^2 B \left(3{,}04\gamma/\sqrt{t} + 1{,}52aA_1/\sqrt{t} + ad|\ln t|\right). \tag{13}$$

2. Оценим величину $\left|\mathscr{D}\right|$. Согласно формуле Тейлора с интегральной формой остаточного члена [8], в окрестности точки V_j верно равенство

$$\Psi^{(2)}_{t(pq)}(W) - \Psi^{(2)}_{t(pq)}(V_j) = \sum_{r=1}^{d} \int_0^1 \Psi^{(3)}_{t(pqr)}(V_j + \tau U_j) U_{jr} d\tau,\ \tau \in (0,1). \tag{14}$$

Заметим, что верно следующее неравенство:

$$\operatorname{Tr}\left[X_j U_j^{\mathrm{T}}(U_{j1}+\ldots+U_{jd})\right] \leqslant A_1^2 B. \tag{15}$$

Обоснуем неравенство (15). Из условий

$$|X_j| = |X_{1j}|+\ldots+|X_{dj}| \leqslant B \quad \text{и} \quad |U_j^{\mathrm{T}}| = |U_{j1}|+\ldots+|U_{jd}| \leqslant A_1 \tag{16}$$

следует, что

$$|X_{1j}|\cdot|U_{j1}|+\ldots+|X_{dj}|\cdot|U_{jd}| \leqslant A_1 B. \tag{17}$$

Из (16) и (17) следует, что

$$\begin{gathered}\operatorname{Tr}\left[X_j U_j^{T}\,(U_{j1}+\ldots+U_{jd})\right] \leqslant \\ \leqslant (|X_{1j}|\cdot|U_{j1}|+\ldots+|X_{dj}|\cdot|U_{jd}|)\,|U_{j1}|+\ldots+(|X_{1j}|\cdot|U_{j1}|+\ldots+|X_{dj}|\cdot|U_{jd}|)\,|U_{jd}| \leqslant \\ \leqslant (|U_{j1}|+\ldots+|U_{jd}|)\,A_1 B \leqslant A_1^2 B.\end{gathered}$$

Таким образом, неравенство (15) обосновано.

Учитывая формулы (14), (15), лемму 3 и равенство (10), получаем

$$\begin{gathered}\left|\mathscr{D}\right| \leqslant \sum_{j=1}^{n}\mathbf{E}\operatorname{Tr}\left[X_j U_j^{\mathrm{T}}\sum_{r=1}^{d}\int_0^1 |\Psi^{(3)}_{t(pqr)}(V_j+\tau U_j)|\cdot\left|U_{jr}\right|d\tau\right] \leqslant \\ \leqslant \sum_{j=1}^{n}\mathbf{E}\left(\sum_{i=1}^{d}|X_{ij}|\cdot|U_{ji}|\sum_{r=1}^{d}|U_{jr}|\int_0^1 \Psi^{(3)}_{t(pqr)}(V_j+\tau U_j)d\tau\right) \leqslant \\ \leqslant \sum_{j=1}^{n}A_1^2 B\left(\sum_{r=1}^{d}\int_0^1\left|\mathbf{E}\Psi^{(3)}_{t(pqr)}(V_j+\tau U_j)\right|d\tau\right) \leqslant nA_1^2 B\left(3{,}04\gamma/\sqrt{t}+1{,}52aA_1/\sqrt{t}+ad|\ln t|\right).\end{gathered} \tag{18}$$

3. Для нахождения оценки величины $\left|\mathscr{B}\right|$ воспользуемся следующим свойством условного математического ожидания [9]: для произвольных случайных величин ξ, η и любой функции $f = f(\eta)$ верно равенство

$$\mathbf{E}\left[f(\eta)\mathbf{E}(\xi|\eta)\right] = \mathbf{E}\left[\xi f(\eta)\right]. \tag{19}$$

С учётом (19), для любого $j = 1,\ldots,d$ верны следующие равенства:

$$\mathbf{E}\left(X_j^{\mathrm{T}}\nabla\Psi_t(V_j)\right) = \sum_{p=1}^{d}\mathbf{E}\left(X_{jp}\Psi^{(1)}_{t(p)}(V_j)\right) = \sum_{p=1}^{d}\mathbf{E}\left(\Psi^{(1)}_{t(p)}(V_j)\mathbf{E}\left(X_{jp}|V_j\right)\right),$$

следовательно,

$$\left|\mathscr{B}\right| \leqslant \sum_{p=1}^{d}\mathbf{E}\left(\left|\Psi^{(1)}_{t(p)}(V_j)\right|\cdot|\mathbf{E}(X_{jp}|V_j)|\right). \tag{20}$$

Из леммы 2 следует, что $\sum\limits_{p=1}^{d}\left|\Psi^{(1)}_{t(p)}(V_j)\right| < d\sqrt{\dfrac{\pi}{2}}$, а значит, для любого $p = 1,\ldots,d$ верно неравенство

$$\left|\Psi^{(1)}_{t(p)}(V_j)\right| < d\sqrt{\frac{\pi}{2}}. \tag{21}$$

Из (20) и (21) следует, что верно неравенство

$$\left|\mathscr{B}\right| \leqslant d\sqrt{\frac{\pi}{2}}\sum_{j=1}^{n}\sum_{p=1}^{d}\mathbf{E}\left|\mathbf{E}(X_{jp}|V_j)\right|. \tag{22}$$

4. Оценим величину $\left|\mathscr{A}\right|$. Заметим, что справедливы следующие равенства:

$$\begin{aligned}\operatorname{Tr}\left[\Psi_t^{(2)}(W)\left(I-\sum_{j=1}^{n}X_jU_j^{\mathrm{T}}\right)\right]&=\sum_{p=1}^{d}\sum_{q=1}^{d}\Psi_{t(pq)}^{(2)}(W)\left(\delta_{pq}-\sum_{j=1}^{n}X_{jq}U_{jp}\right)=\\&=\sum_{p=1}^{d}\sum_{q=1}^{d}\Psi_{t(pq)}^{(2)}(W)\left(\delta_{pq}-\sum_{j=1}^{n}\mathbf{E}(X_{jq}U_{jp})+\sum_{j=1}^{n}\mathbf{E}(X_{jq}U_{jp})-\sum_{j=1}^{n}X_{jq}U_{jp}\right),\end{aligned}\tag{23}$$

где δ_{pq} — символ Кронекера.

Рассмотрим первые два слагаемых в правой части (23). Согласно оценке (8) из леммы 2, верно неравенство

$$\begin{aligned}&\mathbf{E}\left|\sum_{p=1}^{d}\sum_{q=1}^{d}\Psi_{t(pq)}^{(2)}(W)\left[\delta_{pq}-\sum_{j=1}^{n}\mathbf{E}(X_{jq}U_{jp})\right]\right|\leqslant\\&\leqslant\left[\frac{d^2}{\pi}+d\left(\sqrt{\frac{2}{\pi e}}-\frac{1}{\pi}\right)\right]\cdot\left|\ln t\right|\cdot\sum_{p=1}^{d}\sum_{q=1}^{d}\left|\delta_{pq}-\sum_{j=1}^{n}\mathbf{E}(X_{jp}U_{jp})\right|.\end{aligned}\tag{24}$$

Перепишем последние два слагаемых в правой части выражения (23) в следующем виде:

$$\sum_{j=1}^{n}\sum_{p=1}^{d}\sum_{q=1}^{d}\left[\Psi_{t(pq)}^{(2)}(W)-\Psi_{t(pq)}^{(2)}(T_j)+\Psi_{t(pq)}^{(2)}(T_j)\right]\left(\mathbf{E}(X_{jq}U_{jp})-X_{jq}U_{jp}\right).\tag{25}$$

Применяя формулу Тейлора [8] для выражения $\Psi_{t(pq)}^{(2)}(W)-\Psi_{t(pq)}^{(2)}(T_j)$ и лемму 3 в условиях, когда $U=R_j$ и $Y=R_{jr}X_{jq}U_{jp}$, $j=1,\ldots,d$, оценим первые два слагаемых в (25):

$$\begin{aligned}&\sum_{j=1}^{n}\sum_{p=1}^{d}\sum_{q=1}^{d}\left|\mathbf{E}\left(\left[\Psi_{t(pq)}^{(2)}(W)-\Psi_{t(pq)}^{(2)}(T_j)\right]\left[\mathbf{E}(X_{jq}U_{jp})-X_{jq}U_{jp}\right]\right)\right|\leqslant\\&\leqslant nA_1A_2B\left(3{,}04\delta/\sqrt{t}+1{,}52aA_2/\sqrt{t}+ad\left|\ln t\right|\right).\end{aligned}\tag{26}$$

Оценим последнее слагаемое в (25). Согласно оценке (8) из леммы 2, имеет место неравенство

$$\begin{aligned}&\sum_{j=1}^{n}\sum_{p=1}^{d}\sum_{q=1}^{d}\left|\mathbf{E}\Psi_{t(pq)}^{(2)}(T_j)\left[\mathbf{E}(X_{jq}U_{jp})-X_{jq}U_{jp}\right]\right|\leqslant\left(\frac{d^2}{\pi}+d\left(\sqrt{\frac{2}{\pi e}}-\frac{1}{\pi}\right)\right)\times\\&\times\left|\ln t\right|\sum_{j=1}^{n}\sum_{p=1}^{d}\sum_{q=1}^{d}\mathbf{E}\left|\mathbf{E}(X_{jq}U_{jp}-\mathbf{E}(X_{jq}U_{jp}|T_j))\right|.\end{aligned}\tag{27}$$

Учитывая (7), неравенство $A_1\leqslant A_2$ и оценки, полученные в выражениях (13)–(22), (24), (26) и (27), получим

$$\begin{aligned}\gamma\leqslant\frac{4}{3}\Bigg\{&\frac{5}{2}nA_1A_2B\left(3{,}04\gamma/\sqrt{t}+1{,}52aA_1/\sqrt{t}+ad|\ln t|\right)+\\&+d\sqrt{\frac{\pi}{2}}\sum_{j=1}^{n}\sum_{p=1}^{d}\mathbf{E}\left|\mathbf{E}(X_{jp}|V_j)\right|+\left(\frac{d^2}{\pi}+d\left(\sqrt{\frac{2}{\pi e}}-\frac{1}{\pi}\right)\right)\left|\ln t\right|\times\\&\times\left(\sum_{p=1}^{d}\sum_{q=1}^{d}\left|\delta_{pq}-\sum_{j=1}^{n}\mathbf{E}(X_{jp}U_{jp})\right|+\right.\\&\left.+\sum_{j=1}^{n}\sum_{p=1}^{d}\sum_{q=1}^{d}\mathbf{E}\left|\mathbf{E}(X_{jq}U_{jp}-\mathbf{E}(X_{jq}U_{jp}|T_j))\right|\right)\Bigg\}+a\frac{5}{2}\alpha_d\frac{\sqrt{t}}{1-t}.\end{aligned}\tag{28}$$

Перепишем (28) с учётом обозначений $\chi_1 = \sum\limits_{i=1}^{n} \mathbf{E}\big|\mathbf{E}(X_i|V_i)\big|$, $\chi_2 = \sum\limits_{i=1}^{n} \mathbf{E}\big|\mathbf{E}(X_iU_i^{\mathrm{T}}) - \mathbf{E}(X_iU_i^{\mathrm{T}}|T_i)\big|$, $\chi_3 = \left|I - \sum\limits_{i=1}^{n} \mathbf{E}(X_iU_i^{\mathrm{T}})\right|$:

$$\gamma \leqslant 10{,}14nA_1A_2B\gamma/\sqrt{t} + 5{,}07naA_1^2A_2B/\sqrt{t} + 3{,}34nA_1A_2Bad|\ln t| + 1{,}68d\chi_1 + \\ +(0{,}43d^2 + 0{,}23d)\big|\ln t\big|(\chi_2 + \chi_3) + a2{,}5\alpha_d\frac{\sqrt{t}}{1-t}.$$

Используя обозначение $\sqrt{t} = K = 20{,}28nA_1A_2B$, получаем следующую оценку:

$$\gamma \leqslant 0{,}5aA_2 + 0{,}66Kad\,|\ln K| + 3{,}36d\chi_1 + 3{,}6d^2\,|\ln K|\,(\chi_2 + \chi_3) + 5a\alpha_d\frac{K}{1-K^2}. \qquad (29)$$

При округлении сомножителей в (29) получаем искомую оценку (5). Теорема 2 доказана. ■

Замечание 2. При фиксированном d главным членом в правой части неравенства (5) является слагаемое $Kad\big|\ln K\big|$, которое зависит от величин a, d, A_1, A_2 и B. В [4] замечено, что если $\mathscr{H}_1$ — класс индикаторов измеримых выпуклых подмножеств $\mathbb{R}^d$, то верно неравенство $a \leqslant 2d^{1/2}$. В связи с этим выражение в правой части (5) имеет порядок $n^{-1/2}\ln n$.

С учётом оценки для величины a переформулируем теорему 2.

Следствие 1. Пусть выполнены условия теоремы 2. Тогда верна оценка

$$\gamma = \sup_{F\in\mathscr{F}}\big|\mathbf{P}(W\in F) - \mathbf{P}(Z\in F)\big| \leqslant \\ \leqslant 2d^{1/2}A_2 + 2Kd^{3/2}\,|\ln K| + 4d\chi_1 + 4d^2\,|\ln K|\,(\chi_2+\chi_3) + 10d^{1/2}\alpha_d\frac{K}{1-K^2}, \qquad (30)$$

где $\mathscr{F}$ — класс измеримых выпуклых подмножеств $\mathbb{R}^d$; Z — случайный вектор, имеющий d-мерное стандартное нормальное распределение.

Доказательство следствия 1 заключается в непосредственной подстановке в неравенство (29) величины $2d^{1/2}$ вместо a.

Рассмотрим случай, когда W — центрированная и нормированная сумма независимых одинаково распределённых случайных векторов и $U_i = R_i = X_i$, $i = 1,\ldots,n$. Тогда $\chi_1 = \chi_2 = \chi_3 = 0$ и $A_1 = A_2 = B = d/\sqrt{n}$. В этом случае оценка (30) имеет порядок $d^{9/2}n^{-1/2}\ln n$.

В заключение рассмотрим случай, когда W — центрированная и нормированная сумма m-зависимых случайных векторов и компоненты всех векторов являются множеством попарно независимых случайных величин, при этом $\mathbf{E}W = 0$, ковариационная матрица вектора W является единичной, а также имеют место представления

$$U_i = \{j : |j-i| \leqslant m\}\bigcap J, \quad R_i = \{j : |j-i| \leqslant 2m\}\bigcap J,$$
$$\chi_1 = \chi_2 = \chi_3 = 0, \;\; A_1 = \frac{2d(m+1)}{\sqrt{n}}, \; A_2 = \frac{4d(m+1)}{\sqrt{n}}, \; B = \frac{d}{\sqrt{n}}.$$

В этом случае оценка (30) имеет порядок $d^{9/2}m^2n^{-1/2}\ln n$.

3. Задача о раскраске вершин в графе

Рассмотрим применение следствия 1 в известной (см., например, [3]) задаче о нормальной аппроксимации числа рёбер, инцидентных вершинам одного цвета, в регулярном m-графе. Зафиксируем неориентированный граф с n вершинами, каждая из которых имеет степень m. Число рёбер в данном графе равно $N = nm/2$. Предположим, что каждая вершина независимо окрашивается в цвет c_i с вероятностью $\pi_i > 0$, $i = 1, \ldots, d$, и выполнено условие $\sum\limits_{i=1}^{d} \pi_i = 1$. Рассмотрим случайный вектор $W = (W_1, \ldots, W_d)$, где W_i — случайная величина, которая равна числу рёбер, соединяющих вершины одного цвета c_i, $i = 1, \ldots, d$.

На рис. 1 изображён регулярный 6-граф. Его вершины раскрашены в три цвета: c_1, c_2 и c_3, которые обозначены символами ★, ▲ и ● соответственно. В цвет c_1 окрашены вершины 1, 3 и 6, в c_2 — вершина 2, в c_3 — вершины 4 и 5. Рёбра $(1, 3)$, $(1, 6)$ и $(3, 6)$ инцидентны вершинам цвета c_1, ребро $(4, 5)$ — вершинам цвета c_3. Поэтому реализация $\widehat{W}$ случайного вектора W в данном случае имеет вид $\widehat{W}_1 = 3$, $\widehat{W}_2 = 0$, $\widehat{W}_3 = 1$.

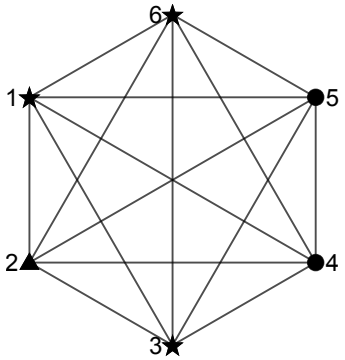

Рис. 1. Регулярный 6-граф с раскрашенными вершинами

Заметим, что для каждой координаты случайного вектора W имеет место следующее равенство:

$$W_i = \sum_{j=1}^{N} X_{ji}, \tag{31}$$

где X_{ji} — индикатор события, при котором обе вершины, инцидентные ребру с номером j, окрашены в один цвет c_i, $i = 1, \ldots, d$, $j = 1, \ldots, N$.

Обозначим $\lambda = \mathbf{E}W = N(\pi_1^2, \ldots, \pi_d^2)$. С учётом (31) элементы ковариационной матрицы $\Sigma = (\sigma_{ij})_{d\times d}$ случайного вектора W имеют следующий вид:

$$\sigma_{ii} = \mathbf{E}W_i^2 - (\mathbf{E}W_i)^2 = N\pi_i^2 + 2N(m-1)\pi_i^3 + \binom{n-2}{2} N\pi_i^4 - N^2\pi_i^4 =$$
$$= N\pi_i^2(1-\pi_i^2) + 2N(m-1)(\pi_i^3 - \pi_i^4), \quad i = 1, \ldots, d,$$
$$\sigma_{ij} = \mathbf{E}W_iW_j - \mathbf{E}W_i\mathbf{E}W_j = -N(2m-1)\pi_i^2\pi_j^2, \quad i, j = 1, \ldots, d, \;\; i \neq j.$$

Обозначим $L = \left[\left(\min\limits_{1\leqslant i\leqslant d} \{\pi_i^2(1-\pi_i)\} \right)^{-1/2} \right]$, где $[\cdot]$ означает целую часть числа. В [3] показано, что $||\Sigma^{-1/2}|| \leqslant N^{-1/2}L$; через $||\cdot||$ обозначается наибольшее абсолютное значение элемента вектора или матрицы.

Применим следствие 1 к вектору $\Sigma^{-1/2}(W-\lambda)$ с учетом равенства

$$B = dN^{-1/2}L. \tag{32}$$

Обозначим множество номеров рёбер графа через $J = \{1, \dots, N\}$. Для $j \in J$ рассмотрим множества индексов

$$J_1^{(j)} = \{t \in J : \text{ребро } t \text{ имеет общую вершину с ребром } j\}, \quad J_2^{(j)} = \bigcup_{t \in J_1^{(j)}} J_1^{(t)}. \tag{33}$$

С учётом (33) для $j \in J$ выберем $U_j = \sum\limits_{t \in J_1^{(j)}} X_t$, $R_j = \sum\limits_{t \in J_2^{(j)}} X_t$, где $X_t = (X_{t1}, \dots, X_{td})$. В этом случае верны оценки

$$A_1 \leqslant (2m-1)B, \quad A_2 \leqslant (2m-1)^2 B, \quad \chi_1 = \chi_2 = \chi_3 = 0. \tag{34}$$

Подставляя в неравенство (30) параметры из формул (32) и (34), получаем, что главный член оценки имеет порядок $\varphi_1 = N^{-1/2}d^{9/2}(2m-1)^3 L^3 \ln N$.

Рассмотрим следующее условие. Пусть параметры d, m, L изменяются так, что верна формула

$$\varphi_1 \to 0, \quad N \to \infty. \tag{35}$$

С учётом следствия 1 при выполнении условия (35) выполняется сходимость

$$\sup_{F \in \mathscr{F}} \left| \mathbf{P}(W \in F) - \mathbf{P}(Z \in F) \right| \to 0, \quad N \to \infty, \tag{36}$$

где $\mathscr{F}$ — класс измеримых выпуклых подмножеств $\mathbb{R}^d$; Z — случайный вектор, имеющий d-мерное стандартное нормальное распределение.

Условие (36) означает, что при увеличении числа рёбер N и изменении числа цветов раскраски d в соответствии с (35) случайный регулярный m-граф с раскрашенными вершинами будет обладать следующим свойством: для каждого $i = 1, \dots, d$ количество рёбер, инцидентных вершинам цвета c_i, будет сближаться (в смысле формулы (36)) с многомерным нормальным распределением $\mathcal{N}(\lambda, \Sigma)$.

Заключение

В работе с использованием метода Стейна получен результат по улучшению известной ранее [7] оценки скорости сходимости в центральной предельной теореме для сумм локально зависимых случайных векторов. Данный результат позволяет вычислять точные оценки скорости сходимости, а также имеет приложения для дискретных математических объектов. Получено условие многомерной нормальной аппроксимации числа рёбер в раскрашенном регулярном графе, инцидентных вершинам одного цвета.

Автор выражает благодарность О. В. Денисову за ряд полезных замечаний, способствовавших существенному улучшению статьи.

ЛИТЕРАТУРА

1. *Götze F.* On the rate of convergence in the multivariate CLT // Ann. Probab. 1991. V. 19. P. 724–739.
2. *Sunklodas J.* Convergence rate estimate in central limit theorem for m-dependent random vectors // Lithuanian Math. J. 1978. V. 18(4). P. 566–575.

3. *Goldstein L. and Rinott Y.* Multivariate normal approximations by Stein's method and size bias couplings // Appl. Probab. 1996. V. 33. P. 1–17.
4. *Reinert G. and Röllin A.* Multivariate normal approximation with Stein's method of exchangeable pairs under a general linearity condition // Ann. Probab. 2009. V. 37. P. 2150–2173.
5. *Rinott Y. and Rotar V. I.* A multivariate CLT for local dependence with $n^{-1/2}\log n$ rate and applications to multivariate graph related statistics // J. Multivariate Analysis. 1996. V. 56. P. 333–350.
6. *Волгин А. В.* Оценка скорости сходимости в многомерной центральной предельной теореме // Прикладная дискретная математика. Приложение. 2013. № 6. С. 11–12.
7. *Волгин А. В.* Об оценке точности многомерной нормальной аппроксимации сумм локально зависимых случайных векторов // Обозрение прикл. и промышл. матем. 2015. Т. 22(4). С. 11–30.
8. *Зорич В. А.* Математический анализ. Ч. I. М.: МЦНМО, 2002. 664 с.
9. *Ширяев А. Н.* Вероятность-1. М.: МЦНМО, 2007. 552 с.

REFERENCES

1. *Götze F.* On the rate of convergence in the multivariate CLT. Ann. Probab., 1991, vol. 19, pp. 724–739.
2. *Sunklodas J.* Convergence rate estimate in central limit theorem for m-dependent random vectors. Lithuanian Math. J., 1978, vol. 18(4), pp. 566–575.
3. *Goldstein L. and Rinott Y.* Multivariate normal approximations by Stein's method and size bias couplings. Appl. Probab., 1996, vol. 33, pp. 1–17.
4. *Reinert G. and Röllin A.* Multivariate normal approximation with Stein's method of exchangeable pairs under a general linearity condition. Ann. Probab., 2009, vol. 37, pp. 2150–2173.
5. *Rinott Y. and Rotar V. I.* A multivariate CLT for local dependence with $n^{-1/2}\log n$ rate and applications to multivariate graph related statistics. J. Multivariate Analysis, 1996, vol. 56, pp. 333–350.
6. *Volgin A. V.* Otsenka skorosti skhodimosti v mnogomernoy tsentral'noy predel'noy teoreme [An improved estimate for the convergence rate in the multidimensional central limit theorem]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2013, no. 6, pp. 11–12. (in Russian)
7. *Volgin A. V.* Ob otsenke tochnosti mnogomernoy normal'noy approksimatsii summ lokal'no zavisimykh sluchaynykh vektorov [On the accuracy estimation for the multidimensional normal approximation of locally dependent random vectors sums]. Obozrenie Prikl. i Promyshl. Matem., 2015, vol. 22(4), pp. 11–30. (in Russian)
8. *Zorich V. A.* Matematicheskiy analiz [Mathematical Analysis]. P. I. Moscow, MCCME Publ., 2002. 664 p. (in Russian)
9. *Shiryaev A. N.* Veroyatnost'-1 [Probability-1]. Moscow, MCCME Publ., 2007. 552 p. (in Russian)

УДК 621.391:519.7+621.391.1:004.7

ПРИМЕНЕНИЕ СУММ ГАУССА ДЛЯ ВЫЧИСЛЕНИЯ ТОЧНЫХ ЗНАЧЕНИЙ ЧИСЛА ПОЯВЛЕНИЙ ЭЛЕМЕНТОВ ПОЛЯ НА ЦИКЛАХ ЛИНЕЙНЫХ РЕКУРРЕНТНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

М. М. Глухов*, О. В. Камловский**

*Московский технологический университет (МИРЭА), г. Москва, Россия
**ООО «Центр сертификационных исследований», г. Москва, Россия

Рассматривается задача получения точных значений для числа появлений элементов на циклах линейных рекуррентных последовательностей немаксимального периода над произвольными конечными полями. Для решения данной задачи применяется аппарат сумм Гаусса.

Ключевые слова: *линейные рекуррентные последовательности, суммы Гаусса, число появлений элементов на циклах.*

DOI 10.17223/20710410/36/3

APPLICATION OF GAUSS SUMS TO CALCULATE THE EXACT VALUES OF THE NUMBER OF APPEARANCES OF ELEMENTS ON CYCLES OF LINEAR RECURRENCES

M. M. Glukhov*, O. V. Kamlovskii**

*Moscow Technological University (MIREA), Moscow, Russia
**Certification Research Center, Moscow, Russia

E-mail: ov-kam@yandex.ru

Using Gauss sums, we solve the problem of obtaining the formulas for the exact values $N(z, u)$ of appearances of z among the elements $u(0)$, $u(1)$, …, $u(T-1)$ of a linear recurrence sequence (LRS) u generated by an irreducible polynomial of a degree m over a field $P = \mathrm{GF}(q)$ in the case, when the period of u is equal to $T = (q^m - 1)/d$, where $d|(p^j+1)$ for some natural number j and $p = \mathrm{char}\, P$, that is, p is a semiprimitive number modulo d. Such a sequence u is obtained from a LRS of the maximal period $q^m - 1$ by regular sampling with step d.

The results of the article generalize the formulas for $N(z, u)$ which are well-known in the case of prime q or $z = 0$. In fact, we give some formulas for $N(z, u)$ in the following cases: 1) $d = 2$; 2) $d > 2$ and $z = 0$; 3) $d > 2$, $z \neq 0$, and $d = d_1$ or $d_1 = 1$, where $d_1 = ((q^m - 1)/(q - 1), d)$; 4) $d > 2$, $z \neq 0$, $d_1 = 2$, and $d/2$ is odd or $(p^{l_1} + 1)/(d/2)$ is even, where l_1 is the least positive integer such that $(d/2) \mid (p^{l_1} + 1)$. Thus, as a corollary, we have a complete solution of the problem in the situation when d is a prime number.

Keywords: *linear recurrent sequences, Gauss sum.*

Введение

Решается задача получения точных значений частот $N(z,u)$ появлений элемента z поля $P=\mathrm{GF}(q)$ среди элементов $u(0)$, $u(1)$, …, $u(T-1)$ линейной рекуррентной последовательности (ЛРП) $u=(u(i))_{i=0}^{\infty}$ над полем P с неприводимым характеристическим многочленом $f(x)\in P[x]$ степени m и периода $T=(q^m-1)/d$ [1, 2]. На число d накладывается следующее условие: d делит p^j+1 для некоторого натурального числа j, где p — характеристика поля. В этом случае говорят, что число p *полупримитивно* по модулю d [3]. Такие последовательности u получаются в результате регулярной выборки с шагом d из ЛРП максимального периода q^m-1 [4, 5].

В [6, теоремы 6, 7] с использованием сумм Гаусса данная задача решена для случая, когда P — простое поле, т. е. $q=p$. При этом выписаны только типы распределений, но не указано значение $N(z,u)$ при каждой конкретной ЛРП u. Позже, в работе [3], задача была решена для случая, когда $z=0$, а P — произвольное поле. Полное решение рассматриваемой задачи приводится в [7, теорема 1.1], однако при доказательстве в самом начале работы автор использовал ошибочное равенство (2.3) для суммы Гаусса, которое привело к ошибочным формулам. Отметим также, что случай, когда $d=2$, а P — произвольное поле, рассмотрен в [1, теорема 23, с. 359], где выписаны только возможные типы распределений.

Цель данной работы — обобщить известные факты о частотах $N(z,u)$ и изложить доказательства в рамках одной статьи. Основными результатами являются теоремы 4, 5, 7, 8, 9, в частности, позволяющие получить полное решение поставленной задачи в ситуации, когда d — простое число.

1. Некоторые свойства характеров конечных полей

Укажем некоторые свойства характеров конечных полей и дадим необходимые определения. Доказательства приведённых результатов подробно изложены в [2].

Пусть $(G,\cdot)$ — конечная абелева группа с нейтральным элементом e. *Характером* группы G называется любой гомоморфизм χ группы G в мультипликативную группу $\mathbb{C}^*$ поля комплексных чисел. Обозначим $\hat{G}$ множество всех характеров группы G. Множество $\hat{G}$ не пусто, так как там содержится, например, характер χ_0, определённый равенствами $\chi_0(g)=1$ для всех $g\in G$. Характер χ_0 называется *тривиальным*, а все остальные характеры из множества $\hat{G}$ называются *нетривиальными*.

Зададим операцию произведения характеров, положив для любых $\chi_1,\chi_2\in\hat{G}$ $\chi_1\cdot\chi_2=\chi$, где $\chi: G\to\mathbb{C}$ и $\chi(g)=\chi_1(g)\chi_2(g)$ для всех $g\in G$. Тогда $(\hat{G},\cdot)$ — конечная абелева группа, изоморфная группе G. Обратным к характеру $\chi\in\hat{G}$ является характер $\bar{\chi}$ (*сопряжённый* к характеру χ), определённый равенством $\bar{\chi}(g)=\overline{\chi(g)}$ для всех $g\in G$, где черта означает комплексное сопряжение.

Пусть χ — произвольный нетривиальный характер конечной абелевой группы G, тогда справедливо соотношение

$$\sum_{g\in G}\chi(g)=\begin{cases}|G|, & \text{если } \chi=\chi_0,\\ 0, & \text{если } \chi\neq\chi_0.\end{cases}\tag{1}$$

Пусть H — подгруппа конечной абелевой группы G. Обозначим через A множество всех характеров χ группы G, которые удовлетворяют условию $\chi(h)=1$ для всех $h\in H$. В этом случае говорят, что характер χ *аннулирует* подгруппу H. Тогда A является подгруппой группы $\hat{G}$ и имеет место равенство

$$|A|=\frac{|G|}{|H|}.\tag{2}$$

Рассмотрим конечное поле $P = \mathrm{GF}(q)$ из q элементов, где $q = p^s$, p — простое число. С полем P связаны две абелевы группы: $(P^*, \cdot)$ — мультипликативная группа поля P и $(P, +)$ — аддитивная группа поля P. Рассмотрим характеры аддитивной группы поля P. Такие характеры будем называть *аддитивными* характерами поля P. Для каждого элемента $b \in P$ рассмотрим отображение $\chi_b : P \to \mathbb{C}^*$, осуществляемое по правилу

$$\chi_b(x) = e^{2\pi i \frac{\mathrm{tr}_p^q(bx)}{p}},$$

для всех $x \in P$, где tr_p^q — функция следа из поля P в простое подполе. Тогда группа аддитивных характеров поля P имеет вид $\{\chi_b : b \in P\}$. При $b = 0$ получим тривиальный характер χ_0, а при $b = e$ — характер χ_e, который называется *каноническим* аддитивным характером поля P.

В дальнейшем будет полезна конструкция поднятия аддитивного характера χ поля $P = \mathrm{GF}(q)$. Пусть $\chi = \chi_b$, где $b \in P$, $b \neq 0$; $Q = \mathrm{GF}(q^m)$ — расширение степени m поля P. Зададим отображение $\chi' : Q \to \mathbb{C}^*$, осуществляемое для всех $z \in Q$ по правилу

$$\chi'(z) = \chi(\mathrm{tr}_q^{q^m}(z)). \tag{3}$$

Тогда

$$\chi'(z) = e^{2\pi i \frac{\mathrm{tr}_p^{q^m}(bz)}{p}}, \quad z \in Q,$$

а значит, χ' является нетривиальным аддитивным характером поля Q. Характер χ' называется *поднятием* характера χ до поля Q. Если χ — канонический аддитивный характер поля P, то его поднятие до поля Q также будет каноническим аддитивным характером поля Q.

Рассмотрим теперь группу мультипликативных характеров поля $P = \mathrm{GF}(q)$. Группа P^* циклическая, поэтому группа мультипликативных характеров — циклическая группа, порождённая характером ψ_1, который задаётся по правилу

$$\psi_1(g^k) = e^{2\pi i \frac{k}{q-1}}, \quad k = 0, 1, \ldots, q-2,$$

где g — некоторый примитивный элемент поля P. Для всех $j \in \{0, 1, \ldots, q-2\}$ обозначим через ψ_j характер ψ_1^j, тогда группа мультипликативных характеров поля P имеет вид $\{\psi_0, \psi_1, \ldots, \psi_{q-2}\} = \langle\psi_1\rangle$. Характер ψ_0 является тривиальным.

Пусть $P = \mathrm{GF}(q)$, ψ — мультипликативный характер поля P, а χ — аддитивный характер поля P. *Сумма Гаусса* $G(\psi, \chi)$ определяется следующим равенством:

$$G(\psi, \chi) = \sum_{c \in P^*} \psi(c)\chi(c).$$

Теорема 1. Для сумм Гаусса справедливы следующие соотношения:

$$G(\psi, \chi) = \begin{cases} q-1, & \text{если } \psi = \psi_0,\ \chi = \chi_0, \\ -1, & \text{если } \psi = \psi_0,\ \chi \neq \chi_0, \\ 0, & \text{если } \psi \neq \psi_0,\ \chi = \chi_0; \end{cases}$$

если $\psi \neq \psi_0$, $\chi \neq \chi_0$, то

$$|G(\psi, \chi)| = \sqrt{q};$$

если χ — произвольный аддитивный характер поля $P = \mathrm{GF}(q)$, то для всех $c \in P^*$

$$\chi(c) = \frac{1}{q-1} \sum_{\psi} G(\bar{\psi}, \chi)\psi(c), \tag{4}$$

где суммирование осуществляется по всем мультипликативным характерам ψ поля P.

2. Сведение задачи к исследованию сумм Гаусса

Пусть $u = (u(i))_{i=0}^{\infty}$ — произвольная чисто периодическая последовательность элементов поля $P = \mathrm{GF}(q)$ периода $T(u)$. Для каждого элемента $z \in P$ обозначим через $N(z, u)$ число появлений z среди элементов $u(0)$, $u(1)$, …, $u(T(u)-1)$.

Покажем, что исследование частоты $N(z, u)$ сводится к исследованию суммы

$$\sigma_z(u) = \sum_{c \in P^*} \chi(-cz) \sum_{i=0}^{T(u)-1} \chi(cu(i)),$$

где χ — нетривиальный аддитивный характер поля P. С использованием равенства (1) и описания всех аддитивных характеров поля P получим

$$N(z, u) = \sum_{i=0}^{T(u)-1} \frac{1}{q} \sum_{c \in P} \chi(c(u(i) - z)),$$

откуда

$$N(z, u) = \frac{T(u)}{q} + \frac{\sigma_z(u)}{q}. \tag{5}$$

Для каждого элемента $z \in P$ обозначим λ_z — отображение из поля P в множество $\mathbb{C}^*$, задаваемое по правилу $\lambda_z(x) = \chi(-zx)$ для всех $x \in P$. Очевидно, что λ_z является аддитивным характером поля P.

Обозначим через $L_P(f)$ множество всех ЛРП над полем P с характеристическим многочленом $f(x)$. Приведём результаты работы [3] (см. также [2, теорема 8.84]). С целью полноты изложения дадим их доказательство в удобных для нас обозначениях. Часть фрагментов доказательства будет использоваться в дальнейшем.

Теорема 2. Пусть $f(x)$ — неприводимый многочлен степени m над полем $P = \mathrm{GF}(q)$, $f(x) \neq x$, $t = T(f) = (q^m - 1)/d$, α — корень $f(x)$ в поле $Q = \mathrm{GF}(q^m)$, u — ненулевая ЛРП из множества $L_P(f)$, имеющая представление $u(i) = \mathrm{tr}_q^{q^m}(a\alpha^i)$, $i \geqslant 0$, $a \in Q$. Тогда

$$\sigma_0(u) = -\frac{q-1}{d} + \frac{q-1}{d} \sum_{\psi' \in B \setminus \{\psi'_0\}} \psi'(a) G(\bar{\psi}', \chi'), \tag{6}$$

а при $z \neq 0$

$$\sigma_z(u) = \frac{1}{d} - \frac{1}{d} \sum_{\psi' \in B \setminus \{\psi'_0\}} \psi'(a) G(\bar{\psi}', \chi') + \frac{1}{d} \sum_{\psi' \in A \setminus B} \psi'(a) G(\bar{\psi}', \chi') G(\psi, \lambda_z), \tag{7}$$

где ψ — ограничение характера ψ' на поле P; A — группа мультипликативных характеров поля Q, аннулирующих элемент α; ψ'_0 — её нейтральный элемент; B — её подгруппа, состоящая из характеров, аннулирующих группу P^*, причём

$$|A| = \frac{q^m - 1}{t}, \quad |B| = \frac{q^m - 1}{[t, q-1]}.$$

Доказательство. Заметим, что элемент a в представлении элементов ЛРП u с использованием функции след из условия теоремы — однозначно определённый ненулевой элемент поля Q. Тогда если χ' — поднятие характера χ до поля Q, определённое равенством (3), то

$$\sigma_z(u) = \sum_{c \in P^*} \chi(-cz) \sum_{i=0}^{t-1} \chi'(ca\alpha^i).$$

С использованием соотношения (4) представим $\sigma_z(u)$ в виде

$$\sigma_z(u) = \sum_{c\in P^*} \chi(-cz) \sum_{i=0}^{t-1} \left(\frac{1}{q^m-1} \sum_{\psi'} G(\bar{\psi}', \chi')\psi'(ca\alpha^i) \right),$$

где суммирование ведётся по всем мультипликативным характерам ψ' поля Q. Изменив порядок суммирования, получим

$$\sigma_z(u) = \frac{1}{q^m-1} \sum_{c\in P^*} \chi(-cz) \sum_{\psi'} G(\bar{\psi}', \chi')\psi'(ca) \sum_{i=0}^{t-1} \psi'(\alpha^i). \tag{8}$$

Заметим, что если $\psi'(\alpha) \neq 1$, то так как $\operatorname{ord}\alpha = t$, справедливы равенства

$$\sum_{i=0}^{t-1} \psi'(\alpha^i) = \sum_{i=0}^{t-1} (\psi'(\alpha))^i = \frac{(\psi'(\alpha))^t - 1}{\psi'(\alpha) - 1} = \frac{\psi'(\alpha^t) - 1}{\psi'(\alpha) - 1} = \frac{\psi'(e) - 1}{\psi'(\alpha) - 1} = 0.$$

Таким образом, в правой части равенства (8) достаточно ограничиться суммированием по всем мультипликативным характерам ψ', для которых $\psi'(\alpha) = 1$. Значит,

$$\sigma_z(u) = \frac{t}{q^m-1} \sum_{c\in P^*} \chi(-cz) \sum_{\psi'\in A} G(\bar{\psi}', \chi')\psi'(ca).$$

Тогда

$$\sigma_z(u) = \frac{1}{d} \sum_{\psi'\in A} \psi'(a) G(\bar{\psi}', \chi') \sum_{c\in P^*} \psi'(c)\lambda_z(c) = \frac{1}{d} \sum_{\psi'\in A} \psi'(a) G(\bar{\psi}', \chi') G(\psi, \lambda_z), \tag{9}$$

где ψ — ограничение характера ψ' на поле P.

Пусть $z = 0$, тогда характер λ_z тривиален и, согласно теореме 1,

$$G(\psi, \lambda_z) = \begin{cases} q-1, & \text{если } \psi = \psi_0, \\ 0, & \text{если } \psi \neq \psi_0. \end{cases}$$

Множество B состоит из всех мультипликативных характеров ψ' поля Q, для которых выполнены соотношения $\psi'(\alpha) = 1$, $\psi'(c) = 1$ для всех $c \in P^*$. Тогда, согласно равенству (9), получим

$$\sigma_0(u) = \frac{q-1}{d} \sum_{\psi'\in B} \psi'(a) G(\bar{\psi}', \chi'). \tag{10}$$

Найдём $|B|$. Пусть θ — примитивный элемент поля P, тогда множество B состоит из всех мультипликативных характеров ψ' поля Q, для которых $\psi'(\alpha) = 1$ и $\psi'(\theta) = 1$, т. е. таких характеров, которые аннулируют группу $H = \langle \alpha, \theta \rangle$, порождённую элементами α и θ. Порядок h группы H равен $[\operatorname{ord}\alpha, q-1] = [t, q-1]$, поэтому, согласно равенству (2),

$$|B| = \frac{q^m-1}{h} = \frac{q^m-1}{[t, q-1]}.$$

Выделив в равенстве (10) отдельно слагаемое, соответствующее тривиальному характеру $\psi'_0 \in B$, и воспользовавшись теоремой 1, будем иметь

$$\sigma_0(u) + \frac{q-1}{d} = \frac{q-1}{d} \sum_{\psi'\in B\setminus\{\psi'_0\}} \psi'(a) G(\bar{\psi}', \chi').$$

Пусть теперь $z \neq 0$, тогда λ_z является нетривиальным аддитивным характером поля P и, согласно теореме 1, $G(\psi, \lambda_z) = -1$ для всех $\psi' \in B$. Тогда из равенства (9) получим

$$\sigma_z(u) = -\frac{1}{d} \sum_{\psi' \in B} \psi'(a) G(\bar{\psi}', \chi') + \frac{1}{d} \sum_{\psi' \in A \setminus B} \psi'(a) G(\bar{\psi}', \chi') G(\psi, \lambda_z).$$

Выделяя отдельно слагаемое, соответствующее тривиальному характеру ψ'_0, согласно теореме 1, получим

$$\sigma_z(u) - \frac{1}{d} = -\frac{1}{d} \sum_{\psi' \in B \setminus \{\psi'_0\}} \psi'(a) G(\bar{\psi}', \chi') + \frac{1}{d} \sum_{\psi' \in A \setminus B} \psi'(a) G(\bar{\psi}', \chi') G(\psi, \lambda_z).$$

Осталось заметить, что множество A состоит из всех мультипликативных характеров, которые аннулируют группу $\langle \alpha \rangle$, поэтому, согласно равенству (2),

$$|A| = \frac{q^m - 1}{\operatorname{ord} \alpha} = \frac{q^m - 1}{t}.$$

Теорема доказана. ■

3. Линейные рекурренты максимального периода

Приведём первый простейший случай, когда теорема 2 позволяет найти точные значения частот появлений элементов на циклах ЛРП.

Пусть $P = \mathrm{GF}(q)$, $f(x) \in P[x]$ — многочлен максимального периода $T(f) = q^m - 1$. Каждая ненулевая ЛРП $u \in L_P(f)$ является ЛРП максимального периода $T(u) = q^m - 1$. Как показано в теореме 2, в этом случае $|A| = |B| = 1$, $\sigma_0(u) = q - 1$, а при $z \neq 0$ выполнено $\sigma_z(u) = 1$. Таким образом, из равенства (5) получим хорошо известный результат [4, 5] для числа $N(z, u)$ появлений элемента $z \in P$ среди элементов $u(0)$, $u(1)$, …, $u(T(f) - 1)$:

$$N(z, u) = \begin{cases} q^{m-1} - 1, & \text{если } z = 0, \\ q^{m-1}, & \text{если } z \neq 0. \end{cases}$$

Отметим, что в работах [4, 5] для доказательства этих формул используется комбинаторный метод.

4. Линейные рекурренты периода $(q^m - 1)/2$

Пусть $P = \mathrm{GF}(q)$, где q — нечётное число, $f(x)$ — неприводимый многочлен степени m над полем P, имеющий период $T(f) = (q^m - 1)/2$. Получим точные значения для числа $N(z, u)$ появлений элемента $z \in P$ среди элементов $u(0)$, $u(1)$, …, $u(T(f) - 1)$ для всех ЛРП $u \in L_P(f)$.

Нам понадобятся некоторые дополнительные сведения о суммах Гаусса над конечными полями. Рассмотрим отображение $\eta : P^* \to \mathbb{C}^*$, определённое для всех $a \in P^*$ по правилу

$$\eta(a) = \begin{cases} 1, & \text{если } a \text{ является квадратом в поле } P, \\ -1, & \text{если } a \text{ не является квадратом в поле } P. \end{cases}$$

Отображение η является мультипликативным характером поля P, имеющим порядок 2. Характер η называется *квадратичным* мультипликативным характером поля P. Если γ — примитивный элемент поля P, то $\eta(a) = 1$ тогда и только тогда, когда

$a \in \langle\gamma^2\rangle$, где $\langle\gamma^2\rangle = H$ — группа порядка $(q-1)/2$, порождённая элементом γ^2. Согласно равенству (2), η является единственным нетривиальным характером, аннулирующим группу H. Для суммы Гаусса $G(\psi, \chi)$, где $\psi = \eta$, а χ — канонический аддитивный характер поля P, можно точно вычислить её значение.

Теорема 3 [2, теорема 5.15]. Пусть $P = \mathrm{GF}(q)$, где $q = p^s$, p — простое число, $s \in \mathbb{N}$. Тогда

$$G(\eta, \chi) = \begin{cases} (-1)^{s-1}\sqrt{q}, & \text{если } p \equiv 1 \pmod 4, \\ (-1)^{s-1} i^s \sqrt{q}, & \text{если } p \equiv 3 \pmod 4. \end{cases}$$

Нам понадобится несколько вспомогательных результатов.

Лемма 1. Пусть η — квадратичный характер поля $P = \mathrm{GF}(q)$, $q = p^s$, p — простое число, $s \in \mathbb{N}$, e — единица поля P. Тогда

$$\eta(-e) = \begin{cases} 1, & \text{если } p \equiv 1 \pmod 4, \\ (-1)^s, & \text{если } p \equiv 3 \pmod 4. \end{cases}$$

Доказательство. Пусть γ — примитивный элемент поля P, тогда элемент $\gamma^{(q-1)/2}$ отличен от e и является решением уравнения $x^2 = e$. Значит, $\gamma^{(q-1)/2} = -e$ и $\eta(-e) = \eta(\gamma^{(q-1)/2}) = (\eta(\gamma))^{(q-1)/2} = (-1)^{(q-1)/2}$, так как γ не является квадратом в поле P. Таким образом, $\eta(-e) = 1$ тогда и только тогда, когда $(q-1)/2$ — чётное число. Это равносильно тому, что $4 \,|\, (p^s - 1) = (p-1)(1 + p + \ldots + p^{s-1})$. Если $p \equiv 1 \pmod 4$, то данное соотношение верно для всех $s \in \mathbb{N}$. Если $p \equiv 3 \pmod 4$, то $p-1 \equiv 2 \pmod 4$, и рассматриваемое соотношение выполнено, если $2 \,|\, (1+p+\ldots+p^{s-1})$, т. е. если s — чётное число. ■

Лемма 2. Пусть η' — квадратичный характер поля $Q = \mathrm{GF}(q^m)$, где q — нечётное число, ψ — ограничение характера η' на поле $P = \mathrm{GF}(q)$. Тогда

1) если m — нечётное число, то $\psi = \eta$ — квадратичный характер поля P;
2) если m — чётное число, то $\psi = \psi_0$ — тривиальный характер поля P.

Доказательство. Пусть γ' и γ — примитивные элементы полей Q и P соответственно. Так как ψ аннулирует группу $H = \langle\gamma^2\rangle$, то из равенства (2) либо $\psi = \eta$, либо $\psi = \psi_0$. Характер η' аннулирует группу $H' = \langle(\gamma')^2\rangle$, поэтому $\psi = \psi_0$ тогда и только тогда, когда P^* — подгруппа группы H'. Это выполнено, только если $q-1$ делит $(q^m - 1)/2 = (q-1)(1 + q + \ldots + q^{m-1})/2$, т. е. если m — чётное число. ■

Докажем теорему, обобщающую на случай произвольного q результаты из [6, теорема 7], где рассматривается ситуация, когда $q = p$ — простое число.

Теорема 4. Пусть $f(x)$ — неприводимый многочлен степени $m = 2\lambda+1$ над полем $P = \mathrm{GF}(q)$, где $q = p^s$, p — нечётное простое число, $s \in \mathbb{N}$. Тогда если $T(f) = (q^m - 1)/2$ и ненулевая ЛРП $u \in L_P(f)$ имеет представление

$$u(i) = \mathrm{tr}_P^Q(a\alpha^i), \quad i \geqslant 0,$$

то

$$N(0, u) = \frac{q^{m-1} - 1}{2},$$

а если $z \neq 0$, то

$$N(z,u) = \begin{cases} \dfrac{q^{m-1} + \eta'(az)q^{\lambda}}{2}, & \text{если } p \equiv 1 \pmod 4, \\ \dfrac{q^{m-1} + \eta'(az)(-1)^{\lambda s}q^{\lambda}}{2}, & \text{если } p \equiv 3 \pmod 4, \end{cases}$$

где η' — квадратичный характер поля $Q = \mathrm{GF}(q^m)$.

Доказательство. Согласно соотношению (5), справедливо равенство

$$N(z,u) = \frac{T(u)}{q} + \frac{1}{q}\sigma_z(u),$$

где

$$\sigma_z(u) = \sum_{c \in P^*} \chi(-cz) \sum_{i=0}^{T(u)-1} \chi(cu(i)), \tag{11}$$

χ — канонический аддитивный характер поля P. Изучим величину $\sigma_z(u)$.

1) Пусть $z = 0$. Тогда из равенства (6) получим

$$\sigma_0(u) = -\frac{q-1}{2} + \frac{q-1}{2} \sum_{\psi' \in B \setminus \{\psi'_0\}} \psi'(a) G(\overline{\psi'}, \chi'), \tag{12}$$

где χ' и ψ'_0 — канонический аддитивный и тривиальный мультипликативный характеры поля Q соответственно; B — множество всех мультипликативных характеров ψ' поля Q, аннулирующих группу $\langle \alpha, \gamma \rangle$, порождённую корнем α многочлена $f(x)$ и примитивным элементом γ поля P. Так как $\psi'(\alpha) = 1$ для каждого характера $\psi' \in B$ и $|\langle \alpha \rangle| = T(f) = (q^m - 1)/2$, то из (2) либо $\psi' = \eta'$, либо $\psi' = \psi'_0$. Согласно доказательству леммы 2, число $q - 1$ не делит $(q^m - 1)/2$, а значит, $\langle \alpha \rangle \neq \langle \alpha, \gamma \rangle$. Таким образом, $\langle \alpha, \gamma \rangle = Q^*$, $B = \{\psi'_0\}$ и из равенства (12) получим

$$\sigma_0(u) = -\frac{q-1}{2}, \quad N(0,u) = \frac{T(u)}{q} + \frac{\sigma_0(u)}{q} = \frac{q^{m-1} - 1}{2}.$$

2) Пусть $z \neq 0$. Тогда из равенства (7) и соотношения $B = \{\psi'_0\}$ будем иметь

$$\sigma_z(u) = \frac{1}{2} + \frac{1}{2} \sum_{\psi' \in A \setminus \{\psi'_0\}} \psi'(a) G(\overline{\psi'}, \chi') G(\psi, \lambda_z),$$

где A — множество всех мультипликативных характеров ψ' поля Q, аннулирующих группу $\langle \alpha \rangle$; ψ — ограничение характера ψ' на поле P; λ_z — аддитивный характер поля P, определённый равенством $\lambda_z(x) = \chi(-zx)$, $x \in P$. Как отмечалось при доказательстве п. 1, элемент α аннулируется только характерами ψ'_0, η'. Значит, $A = \{\psi'_0, \eta'\}$. Поэтому с использованием леммы 2 и равенства $\overline{\eta'} = \eta'$ получим

$$\sigma_z(u) = \frac{1}{2} + \frac{1}{2}\eta'(a) G(\eta', \chi') G(\eta, \lambda_z).$$

Для суммы Гаусса $G(\eta, \lambda_z)$ имеем

$$G(\eta, \lambda_z) = \sum_{c \in P^*} \eta(c)\lambda_z(c) = \sum_{c \in P^*} \eta(c)\chi(-cz) = \sum_{b \in P^*} \eta(-bz^{-1})\chi(b) =$$
$$= \eta(-z^{-1}) G(\eta, \chi) = \eta(-e)\eta(z^{-1}) G(\eta, \chi) = \eta(-e)\eta(z) G(\eta, \chi).$$

Таким образом,

$$\sigma_z(u) = \frac{1}{2} + \frac{1}{2}\eta'(az)\eta(-e)G(\eta,\chi)G(\eta',\chi'). \tag{13}$$

Согласно теореме 3 и лемме 1,

$$\eta(-e)G(\eta,\chi) = \begin{cases} (-1)^{s-1}\sqrt{q}, & \text{если } p \equiv 1 \pmod 4, \\ -i^s\sqrt{q}, & \text{если } p \equiv 3 \pmod 4, \end{cases}$$

$$G(\eta',\chi') = \begin{cases} (-1)^{ms-1}q^{m/2}, & \text{если } p \equiv 1 \pmod 4, \\ (-1)^{ms-1}i^{ms}q^{m/2}, & \text{если } p \equiv 3 \pmod 4. \end{cases}$$

Тогда из равенства (13) получим

$$\sigma_z(u) = \begin{cases} \dfrac{1+\eta'(az)q^{(m+1)/2}}{2}, & \text{если } p \equiv 1 \pmod 4, \\ \dfrac{1+(-1)^{ms}\eta'(az)i^{(m+1)s}q^{(m+1)/2}}{2}, & \text{если } p \equiv 3 \pmod 4, \end{cases}$$

а значит, согласно (11),

$$N(z,u) = \frac{T(u)}{q} + \frac{\sigma_z(u)}{q} = \begin{cases} \dfrac{q^{m-1}+\eta'(az)q^{\lambda}}{2}, & \text{если } p \equiv 1 \pmod 4, \\ \dfrac{q^{m-1}+\eta'(az)(-1)^{\lambda s}q^{\lambda}}{2}, & \text{если } p \equiv 3 \pmod 4. \end{cases}$$

Теорема доказана. ■

Следствие 1. В условиях теоремы 4 имеют место соотношения

$$N(0,u) = \frac{q^{m-1}-1}{2}, \quad N(z,u) = \frac{q^{m-1} \pm q^{\lambda}}{2},\ z \in P^*,$$

причём у каждой ЛРП u для половины элементов $z \in P^*$ в последней формуле надо взять знак «+», а для другой половины — знак «−».

Рассмотрим теперь случай, когда m — чётное число. Следующая теорема обобщает на случай произвольного q результаты из [6, теорема 7].

Теорема 5. Пусть $f(x)$ — неприводимый многочлен степени $m = 2\lambda$ над полем $P = \mathrm{GF}(q)$, где $q = p^s$, p — нечётное простое число, $s \in \mathbb{N}$. Тогда если $T(f) = (q^m-1)/2$ и ненулевая ЛРП $u \in L_P(f)$ имеет представление $u(i) = \mathrm{tr}_P^Q(a\alpha^i)$, $i \geqslant 0$, то

$$N(0,u) = \begin{cases} \dfrac{q^{m-1}-(q-1)\eta'(a)q^{\lambda-1}-1}{2}, & \text{если } p \equiv 1 \pmod 4, \\ \dfrac{q^{m-1}-(q-1)\eta'(a)(-1)^{\lambda s}q^{\lambda-1}-1}{2}, & \text{если } p \equiv 3 \pmod 4, \end{cases}$$

а если $z \neq 0$, то

$$N(z,u) = \begin{cases} \dfrac{q^{m-1}+\eta'(a)q^{\lambda-1}}{2}, & \text{если } p \equiv 1 \pmod 4, \\ \dfrac{q^{m-1}+\eta'(a)(-1)^{\lambda s}q^{\lambda-1}}{2}, & \text{если } p \equiv 3 \pmod 4, \end{cases}$$

где η' — квадратичный характер поля $Q = \mathrm{GF}(q^m)$.

Доказательство. С использованием доказательства леммы 2 при чётном m имеем P^* — подгруппа группы $\langle\alpha\rangle$, поэтому в обозначениях доказательства теоремы 4 $B = \{\psi_0', \eta'\} = A$.

1) Пусть $z = 0$. Из равенства (12) получим

$$\sigma_0(u) = -\frac{q-1}{2} + \frac{q-1}{2}\eta'(a)G(\eta', \chi'),$$

а значит, согласно теореме 3,

$$\sigma_0(u) = \begin{cases} \dfrac{-(q-1)(1+\eta'(a)q^{m/2})}{2}, & \text{если } p \equiv 1 \pmod 4, \\ \dfrac{-(q-1)(1+\eta'(a)i^{ms}q^{m/2})}{2}, & \text{если } p \equiv 3 \pmod 4. \end{cases}$$

Тогда будем иметь

$$N(0,u) = \frac{T(u)}{q} + \frac{\sigma_0(u)}{q} = \begin{cases} \dfrac{q^{m-1} - (q-1)\eta'(a)q^{\lambda-1} - 1}{2}, & \text{если } p \equiv 1 \pmod 4, \\ \dfrac{q^{m-1} - (q-1)\eta'(a)(-1)^{\lambda s}q^{\lambda-1} - 1}{2}, & \text{если } p \equiv 3 \pmod 4. \end{cases}$$

2) Пусть $z \neq 0$. Из равенства (7) получим

$$\sigma_z(u) = \frac{1}{2} - \frac{1}{2}\eta'(a)G(\eta', \chi'),$$

а значит, согласно теореме 3,

$$\sigma_z(u) = \begin{cases} \dfrac{1+\eta'(a)q^{m/2}}{2}, & \text{если } p \equiv 1 \pmod 4, \\ \dfrac{1+\eta'(a)i^{ms}q^{m/2}}{2}, & \text{если } p \equiv 3 \pmod 4. \end{cases}$$

Тогда будем иметь

$$N(z,u) = \frac{T(u)}{q} + \frac{\sigma_z(u)}{q} = \begin{cases} \dfrac{q^{m-1} + \eta'(a)q^{\lambda-1}}{2}, & \text{если } p \equiv 1 \pmod 4, \\ \dfrac{q^{m-1} + \eta'(a)(-1)^{\lambda s}q^{\lambda-1}}{2}, & \text{если } p \equiv 3 \pmod 4. \end{cases}$$

Теорема доказана. ■

Следствие 2. В условиях теоремы 5 имеют место соотношения

$$N(0,u) = \frac{q^{m-1} \pm (q-1)q^{\lambda-1} - 1}{2}, \quad N(z,u) = \frac{q^{m-1} \mp q^{\lambda-1}}{2},\ z \in P^*,$$

причём у каждой ЛРП u все элементы $z \in P^*$ появляются одинаково часто.

5. Суммы Гаусса в полупримитивном случае

Пусть $P = \mathrm{GF}(q)$ — поле порядка $q = p^s$, где p — простое число, $s \in \mathbb{N}$, $f(x)$ — неприводимый многочлен степени m над полем P, $f(x) \neq x$. Рассмотрим ситуацию, когда $T(f) = (q^m - 1)/d$, где d — делитель числа $q^m - 1$, причём найдётся натуральное число j, такое, что $d \,|\, (p^j + 1)$.

Так как ранее был изучен случай $d = 1$ и 2, далее будем рассматривать случай $d > 2$. Найдём точные значения для числа $N(z, u)$ появлений элемента $z \in P$ среди элементов $u(0)$, $u(1)$, …, $u(T(f) - 1)$ в каждой ненулевой ЛРП $u \in L_P(f)$. Отметим, что в этом случае $T(u) = T(f)$. Нам понадобится ряд вспомогательных результатов.

Лемма 3. Пусть число p полупримитивно по модулю d, $q = p^s$, $d \,|\, (q^m - 1)$, $d > 2$ и $l = \min\{j \in \mathbb{N} : d \,|\, (p^j + 1)\}$. Тогда $2l \,|\, sm$.

Доказательство. Рассмотрим кольцо $\mathbb{Z}/d$ классов вычетов по модулю d. Из условия следует, что для класса $\beta = [p]_d$ справедливы равенства $\beta^{ms} = [1]_d$, $\beta^l = [-1]_d$, $\beta^{2l} = [1]_d$. Пусть $k = \operatorname{ord} \beta$ — мультипликативный порядок элемента β. Покажем, что $k = 2l$, тогда отсюда будет следовать, что $2l \,|\, sm$. Ясно, что $k \,|\, 2l$. Допустим, что $k < 2l$, тогда $k \leqslant l$ и из равенств $\beta^k = [1]_d$, $\beta^l = [-1]_d$ получим $\beta^k + \beta^l = \beta^k([1]_d + \beta^{l-k}) = [0]_d$. В силу обратимости элемента β отсюда будем иметь

$$\beta^{l-k} = [-1]_d. \tag{14}$$

Так как $d > 2$, то $[1]_d \neq [-1]_d$, а значит, $l - k \neq 0$. Покажем, что $k \neq 0$. В случае $k = 0$ имеем $\beta = [1]_d$, $d \,|\, (p - 1)$, $d \,|\, (p^l + 1)$, следовательно, $d \,|\, (p^l + p)$, $d \,|\, p(p^{l-1} + 1)$ и, значит, $d \,|\, (p^{l-1} + 1)$. Чтобы не получить противоречие с выбором l в условии, имеем $l = 1$, что приводит к противоречию с условием на d. Таким образом, $k \neq 0$, и в равенстве (14) получаем $0 < l - k < l$, что противоречит выбору l. ■

Доказательство следующего результата подробно изложено в [2, теорема 5.16].

Лемма 4 (теорема Штикельбергера). Пусть в условиях леммы 3 ψ — мультипликативный характер порядка d в поле $T = \mathrm{GF}(p^{2l})$, χ — канонический аддитивный характер поля T, тогда

$$G(\psi, \chi) = \begin{cases} -p^l, & \text{если } d \text{ чётно, а } \dfrac{p^l + 1}{d} \text{ нечётно,} \\ p^l, & \text{если } d \text{ нечётно или } \dfrac{p^l + 1}{d} \text{ чётно.} \end{cases}$$

Следствие 3. В условиях леммы 4 для каждого $i = 1, 2, \ldots, d - 1$

$$G(\psi^i, \chi) = \begin{cases} (-1)^i p^l, & \text{если } d \text{ чётно, а } \dfrac{p^l + 1}{d} \text{ нечётно,} \\ p^l, & \text{если } d \text{ нечётно или } \dfrac{p^l + 1}{d} \text{ чётно.} \end{cases}$$

Доказательство. Пусть $G_i = G(\psi^i, \chi)$, $d_i = \operatorname{ord} \psi^i$, где $i = 1, 2, \ldots, d - 1$. Тогда

$$d_i = \frac{d}{(d, i)}, \quad \frac{p^l + 1}{d_i} = \frac{(p^l + 1)(d, i)}{d}. \tag{15}$$

Рассмотрим несколько случаев.

1) Если d — нечётное число, то из равенств (15) получим d_i — нечётное число, $d_i \,|\, (p^l + 1)$. Тогда по лемме 4 $G_i = p^l$.

2) Если d — чётное число и $(p^l+1)/d$ — чётное число, то, согласно (15), $(p^l+1)/d_i$ — чётное число, и по лемме 4 $G_i = p^l$.
3) Если d — чётное число, а $(p^l+1)/d$ — нечётное число, то рассмотрим два случая:
   3а) если i — чётное число, то, согласно (15), $(p^l+1)/d_i$ — чётное число, и по лемме 4 $G_i = p^l$;
   3б) если i — нечётное число, то, согласно (15), $(p^l+1)/d_i$ — нечётное число, и по лемме 4 $G_i = -p^l$.

Следствие доказано. ■

Нам понадобится конструкция поднятия мультипликативного характера. Пусть $T = \mathrm{GF}(q_1)$ и $Q = \mathrm{GF}(q_2)$ — расширение степени r поля T, т. е. $q_2 = q_1^r$. Определим *норму* элемента $a \in Q^*$ над полем T следующим равенством:

$$N_{q_1}^{q_2}(a) = a \cdot a^{q_1} \cdot a^{q_1^2} \cdots a^{q_1^{r-1}} = a^{\frac{q_1^r-1}{q_1-1}} = a^{\frac{q_2-1}{q_1-1}}.$$

Так как $(N_{q_1}^{q_2}(a))^{q_1-1} = e$, таким образом определено отображение $N_{q_1}^{q_2} : Q^* \to T^*$, которое обладает следующими свойствами:

1) $N_{q_1}^{q_2}(ab) = N_{q_1}^{q_2}(a)N_{q_1}^{q_2}(b)$ для всех $a, b \in Q^*$;
2) $N_{q_1}^{q_2}$ сюръективно.

Пусть ψ — мультипликативный характер поля T. Рассмотрим отображение $\psi' : Q^* \to \mathbb{C}^*$, осуществляемое по правилу

$$\psi'(a) = \psi(N_{q_1}^{q_2}(a)), \quad a \in Q^*.$$

Из свойства 1 для нормы получим, что ψ' является мультипликативным характером поля Q. Его называют *поднятием мультипликативного характера* ψ до поля Q. Обозначим ψ_0 и ψ'_0 тривиальные мультипликативные характеры полей T и Q соответственно. Заметим, что для каждого $k \in \mathbb{N}_0$ справедливы равенства

$$(\psi')^k(a) = (\psi'(a))^k = (\psi(N_{q_1}^{q_2}(a)))^k = \psi^k(N_{q_1}^{q_2}(a)), \quad a \in Q^*. \tag{16}$$

Отсюда с учётом сюръективности отображения $N_{q_1}^{q_2}$ получим, что $(\psi')^k = \psi'_0$ тогда и только тогда, когда $\psi^k = \psi_0$. Таким образом, для порядков характеров ψ и ψ' имеем

$$\operatorname{ord}\psi = \operatorname{ord}\psi'. \tag{17}$$

Следующий результат подробно доказан в [2, теорема 5.14].

Лемма 5 (теорема Дэвенпорта — Хассе). Пусть χ — аддитивный, а ψ — мультипликативный характеры поля $T = \mathrm{GF}(q_1)$, не являющиеся одновременно тривиальными. Тогда если $Q = \mathrm{GF}(q_2)$ — расширение степени r поля T и χ', ψ' — поднятия характеров χ и ψ соответственно до поля Q, то

$$G(\psi', \chi') = (-1)^{r-1} G(\psi, \chi)^r.$$

Теорема 6. Пусть простое число p полупримитивно по модулю d, $d > 2$, $d \,|\, (q^m - 1)$, $q = p^s$ и число l выбрано так, как в лемме 3. Тогда если ψ' — мультипликативный характер поля $Q = \mathrm{GF}(q^m)$, имеющий порядок d, а χ' — канонический аддитивный характер поля Q, то для всех $i = 1, 2, \ldots, d-1$

$$G((\psi')^i, \chi') = \begin{cases} (-1)^{(i+1)r-1} q^{m/2}, & \text{если } d \text{ чётно, а } \dfrac{p^l+1}{d} \text{ нечётно,} \\ (-1)^{r-1} q^{m/2}, & \text{если } d \text{ нечётно или } \dfrac{p^l+1}{d} \text{ чётно,} \end{cases}$$

где $r = (ms)/(2l)$.

Доказательство. Из леммы 3 следует, что поле $T = GF(p^{2l})$ является подполем поля Q. Так как $d \mid (p^{2l}-1)$, в циклической группе всех мультипликативных характеров поля T есть подгруппа порядка d. Разные характеры из этой подгруппы имеют разные поднятия до поля Q, которые, согласно (17), имеют такие же порядки. Таким образом, найдётся мультипликативный характер ψ поля T, такой, что $\operatorname{ord}\psi = d$ и его поднятие до поля Q совпадает с ψ'. Кроме того, согласно (16), поднятие характера ψ^i до поля Q будет совпадать с характером $(\psi')^i$ для всех $i = 1, 2, \ldots, d-1$. Для завершения доказательства остаётся воспользоваться следствием 3 и леммой 5. ■

Лемма 6. Пусть в условиях теоремы 6 d_1 — натуральный делитель числа d, $\psi_1' = (\psi')^{d/d_1}$, γ — примитивный элемент поля Q. Тогда для каждого $a \in Q^*$:

1) $\psi_1'(a) = 1$ тогда и только тогда, когда $a \in \langle\gamma^{d_1}\rangle$;
2) $\psi_1'(a) = -1$ тогда и только тогда, когда d_1 — чётное число, $a \notin \langle\gamma^{d_1}\rangle$ и $a \in \langle\gamma^{d_1/2}\rangle$;
3) $\psi_1'(a^{-1}) = (-1)^{\varepsilon}$ для некоторого $\varepsilon \in \mathbb{N}_0$ тогда и только тогда, когда $\psi_1'(a) = (-1)^{\varepsilon}$.

Доказательство.

1) Рассмотрим группу всех мультипликативных характеров поля Q, аннулирующих подгруппу $\langle\gamma^{d_1}\rangle$ группы Q^*. Из равенства (2) следует, что её порядок равен d_1. Таким образом, группа $\langle\psi_1'\rangle$, имеющая порядок d_1, совпадает с группой всех характеров, аннулирующих группу $\langle\gamma^{d_1}\rangle$. Так как ψ_1' — её образующий, то $\psi_1'(a) = 1$ тогда и только тогда, когда $a \in \langle\gamma^{d_1}\rangle$.

2) Равенство $\psi_1'(a) = -1$ равносильно тому, что $(\psi_1'(a))^2 = \psi_1'(a^2) = 1$, а $\psi_1'(a) \neq 1$, т. е. по п. 1 $a^2 \in \langle\gamma^{d_1}\rangle$ и $a \notin \langle\gamma^{d_1}\rangle$. Если d_1 — нечётное число, то соотношение $(\psi_1'(a))^{d_1} = (-1)^{d_1} = -1$ противоречит условию $\operatorname{ord}\psi' = d$. Пусть d_1 — чётное число. Опишем все элементы $a \in Q^*$, такие, что $a^2 \in \langle\gamma^{d_1}\rangle$. Используя представление $a = \gamma^s$, где $s = 0, 1, \ldots, q^m - 2$, получим, что $\gamma^{2s} \in \langle\gamma^{d_1}\rangle$ тогда и только тогда, когда $d_1 \mid 2s$, т. е. $(d_1/2) \mid s$. Таким образом, $a = \gamma^{d_1 t/2}$, где $t = 0, 1, \ldots, (2(q^m-1)/d_1) - 1$, или $a \in \langle\gamma^{d_1/2}\rangle$.

3) Доказательство непосредственно следует из п. 1 и 2. ■

6. Полупримитивный случай. Нулевые элементы на циклах

Рассмотрим вопрос о распределении нулей на циклах ЛРП. Следующая теорема обобщает результаты [3, формулы (3.2)], где указаны только типы распределений, но не получены условия, при которых каждая конкретная ЛРП u имеет данный тип.

Теорема 7. Пусть $f(x)$ — неприводимый многочлен степени m над полем $\mathrm{GF}(q)$, $q = p^s$, p — простое число, $s \in \mathbb{N}$, $T(f) = (q^m-1)/d$, $d > 2$, число p полупримитивно по модулю d, $d_1 = ((q^m-1)/(q-1), d)$, $r = (ms)/(2l)$, где l определено в лемме 3. Тогда для каждой ненулевой ЛРП $u \in L_P(f)$, имеющей представление $u(i) = \mathrm{tr}_P^Q(a\alpha^i)$, $i \geqslant 0$, справедливо:

1) если dr/d_1 — чётное, или d — нечётное, или $(p^l+1)/d$ — чётное, то

$$N(0,u) = \begin{cases} \dfrac{q^{m-1} + (-1)^r (q-1) q^{m/2-1} - 1}{d}, & \text{если } a \notin \langle\gamma^{d_1}\rangle, \\ \dfrac{q^{m-1} + (-1)^{r-1} (q-1)(d_1-1) q^{m/2-1} - 1}{d}, & \text{если } a \in \langle\gamma^{d_1}\rangle; \end{cases}$$

2) если dr/d_1 — нечётное, d — чётное, $(p^l+1)/d$ — нечётное, то d_1 — чётное и

$$N(0,u)=\begin{cases}\dfrac{q^{m-1}+(-1)^r(q-1)q^{m/2-1}-1}{d}, & \text{если } a\in\langle\gamma^{d_1}\rangle \text{ или } a\notin\langle\gamma^{d_1/2}\rangle,\\[2ex] \dfrac{q^{m-1}+(-1)^{r-1}(q-1)(d_1-1)q^{m/2-1}-1}{d}, & \text{если } a\notin\langle\gamma^{d_1}\rangle \text{ и } a\in\langle\gamma^{d_1/2}\rangle.\end{cases}$$

Доказательство. С использованием равенства (5) получим

$$N(0,u)=\frac{T(u)}{q}+\frac{\sigma_0(u)}{q},$$

где $\sigma_0(u)=\sum\limits_{c\in P^*}\sum\limits_{i=0}^{T(f)-1}\chi(cu(i))$; χ — канонический аддитивный характер поля P. Из равенства (6) будем иметь

$$\sigma_0(u)=-\frac{q-1}{d}+\frac{q-1}{d}\sum_{\psi'\in B\setminus\{\psi'_0\}}\psi'(a)G(\overline{\psi'},\chi'),$$

где χ' и ψ'_0 — канонический аддитивный и тривиальный мультипликативный характеры поля Q соответственно; B — множество всех мультипликативных характеров ψ' поля Q, аннулирующих группу $\langle\alpha,\gamma\rangle$, порождённую корнем α многочлена $f(x)$ и примитивным элементом γ поля P. Порядок группы $\langle\alpha,\gamma\rangle$ делится на $(q^m-1)/d$, а значит, он равен $(q^m-1)/d'$, где d' — делитель d. Как показано при доказательстве теоремы 2,

$$d'=|B|=\frac{q^m-1}{[(q^m-1)/d,q-1]}=\frac{d(q^m-1)}{(q^m-1)(q-1)}((q^m-1)/d,q-1)=$$
$$=\frac{d}{q-1}((q^m-1)/d,q-1)=\frac{(q^m-1,d(q-1))}{q-1}=((q^m-1)/(q-1),d)=d_1.$$

Тогда группа B является циклической группой, порождённой характером $(\psi')^{d/d_1}=\psi'_1$, где ψ' — мультипликативный характер порядка d поля $Q=\mathrm{GF}(q^m)$. Заметим, что если характеры пробегают всю группу B, то характеры, сопряжённые к ним, также пробегают всю группу B, поэтому

$$\begin{aligned}\sigma_0(u)&=-\frac{q-1}{d}+\frac{q-1}{d}\sum_{i=1}^{d_1-1}(\psi'_1)^i(a)G(\overline{(\psi'_1)^i},\chi')=\\ &=-\frac{q-1}{d}+\frac{q-1}{d}\sum_{i=1}^{d_1-1}(\psi'_1)^i(a^{-1})G((\psi'_1)^i,\chi')=\\ &=-\frac{q-1}{d}+\frac{q-1}{d}\sum_{i=1}^{d_1-1}(\psi')^{di/d_1}(a^{-1})G((\psi')^{di/d_1},\chi').\end{aligned}\tag{18}$$

Рассмотрим два случая.

С л у ч а й 1. Пусть d — нечётное число или $(p^l+1)/d$ — чётное число. Тогда по теореме 6

$$\sigma_0(u)=-\frac{q-1}{d}+\frac{q-1}{d}\sum_{i=1}^{d_1-1}(\psi'_1)^i(a^{-1})(-1)^{r-1}q^{m/2}=\frac{q-1}{d}\left(-1+(-1)^{r-1}q^{m/2}\sum_{i=1}^{d_1-1}(\psi'_1(a^{-1}))^i\right),$$

а значит,

$$\sigma_0(u) = \frac{q-1}{d}\left(-1 + (-1)^r q^{m/2} + (-1)^{r-1} q^{m/2} \sum_{i=0}^{d_1-1} (\psi_1'(a^{-1}))^i\right). \tag{19}$$

Заметим, что

$$\sum_{i=0}^{d_1-1} (\psi_1'(a^{-1}))^i = \begin{cases} \dfrac{(\psi_1'(a^{-1}))^{d_1}-1}{\psi_1'(a^{-1})-1}, & \text{если } \psi_1'(a^{-1}) \neq 1, \\ d_1, & \text{если } \psi_1'(a^{-1}) = 1, \end{cases}$$

а так как $\operatorname{ord}\psi_1' = d_1$, то

$$\sum_{i=0}^{d_1-1} (\psi_1'(a^{-1}))^i = \begin{cases} 0, & \text{если } \psi_1'(a^{-1}) \neq 1, \\ d_1, & \text{если } \psi_1'(a^{-1}) = 1. \end{cases}$$

Тогда из равенства (19) и леммы 6 получим

$$\sigma_0(u) = \begin{cases} \dfrac{q-1}{d}\left(-1 + (-1)^r q^{m/2}\right), & \text{если } a \notin \langle\gamma^{d_1}\rangle, \\ \dfrac{q-1}{d}\left(-1 + (-1)^{r-1}(d_1-1) q^{m/2}\right), & \text{если } a \in \langle\gamma^{d_1}\rangle. \end{cases}$$

Подставляя найденные значения $\sigma_0(u)$ в равенство для частот $N(0,u)$, получим

$$N(0,u) = \frac{q^m-1}{dq} + \frac{\sigma_0(u)}{q} = \begin{cases} \dfrac{q^{m-1}-1+(-1)^r(q-1)q^{m/2-1}}{d}, & \text{если } a \notin \langle\gamma^{d_1}\rangle, \\ \dfrac{q^{m-1}-1+(-1)^{r-1}(q-1)(d_1-1)q^{m/2-1}}{d}, & \text{если } a \in \langle\gamma^{d_1}\rangle. \end{cases}$$

С л у ч а й 2. Пусть d — чётное число, $(p^l+1)/d$ — нечётное число. Из равенства (18) с использованием теоремы 6 будем иметь

$$\sigma_0(u) = \frac{q-1}{d}\left(-1 + \sum_{i=1}^{d_1-1} (\psi_1'(a^{-1}))^i (-1)^{(ki+1)r-1} q^{m/2}\right),$$

где $k = d/d_1$. Тогда

$$\sigma_0(u) = \frac{q-1}{d}\left(-1 + (-1)^{r-1} q^{m/2} \sum_{i=1}^{d_1-1} (\psi_1'(a^{-1})(-1)^{kr})^i\right),$$

и значит,

$$\sigma_0(u) = \frac{q-1}{d}\left(-1 + (-1)^r q^{m/2} + (-1)^{r-1} q^{m/2} \sum_{i=0}^{d_1-1} (\psi_1'(a^{-1})(-1)^{kr})^i\right).$$

Заметим, что

$$\sum_{i=0}^{d_1-1} (\psi_1'(a^{-1})(-1)^{kr})^i = \begin{cases} \dfrac{(\psi_1'(a^{-1})(-1)^{kr})^{d_1}-1}{\psi_1'(a^{-1})(-1)^{kr}-1}, & \text{если } \psi_1'(a^{-1}) \neq (-1)^{kr}, \\ d_1, & \text{если } \psi_1'(a^{-1}) = (-1)^{kr}. \end{cases}$$

Так как d — чётное число и $\operatorname{ord}\psi_1' = d_1$, то

$$(\psi_1'(a^{-1})(-1)^{kr})^{d_1} = (\psi_1')^{d_1}(a^{-1})(-1)^{krd_1} = (-1)^{dr} = 1,$$

поэтому

$$\sum_{i=0}^{d_1-1}(\psi_1'(a^{-1})(-1)^{kr})^i = \begin{cases} 0, & \text{если } \psi_1'(a^{-1}) \neq (-1)^{kr}, \\ d_1, & \text{если } \psi_1'(a^{-1}) = (-1)^{kr}. \end{cases}$$

Если теперь $kr = dr/d_1$ — чётное число, то ответ совпадает с ответом, полученным для случая 1. Пусть kr — нечётное число. Тогда $k = d/d_1$ — нечётное, и так как d — чётное, то d_1 — чётное число. В итоге по аналогии с доказательством случая 1 и с учётом леммы 6 получим

$$N(0,u) = \begin{cases} \dfrac{q^{m-1} - 1 + (-1)^r(q-1)q^{m/2-1}}{d}, & \text{если } a \in \langle\gamma^{d_1}\rangle \text{ или } a \notin \langle\gamma^{d_1/2}\rangle, \\ \dfrac{q^{m-1} - 1 + (-1)^{r-1}(q-1)(d_1-1)q^{m/2-1}}{d}, & \text{если } a \notin \langle\gamma^{d_1}\rangle, a \in \langle\gamma^{d_1/2}\rangle. \end{cases}$$

Теорема доказана. ■

7. Полупримитивный случай, $d = d_1$. Ненулевые элементы на циклах

Исследование частот появлений ненулевых элементов разобьём на несколько случаев. Сначала рассмотрим ситуацию, когда $d = d_1$.

Теорема 8. Пусть в условиях теоремы 7 $d = d_1$, тогда все ненулевые элементы $z \in P$ появляются в ЛРП u одинаково часто и справедливы следующие равенства:

1) если r — чётное число, или d — нечётное число, или $\dfrac{p^l+1}{d}$ — чётное число, то

$$N(z,u) = \begin{cases} \dfrac{q^{m-1} + (-1)^{r-1}q^{m/2-1}}{d}, & \text{если } a \notin \langle\gamma^d\rangle, \\ \dfrac{q^{m-1} + (-1)^r(d-1)q^{m/2-1}}{d}, & \text{если } a \in \langle\gamma^d\rangle; \end{cases}$$

2) если r — нечётное число, d — чётное число, $\dfrac{p^l+1}{d}$ — нечётное число, то

$$N(z,u) = \begin{cases} \dfrac{q^{m-1} + (-1)^{r-1}q^{m/2-1}}{d}, & \text{если } a \in \langle\gamma^d\rangle \text{ или } a \notin \langle\gamma^{d/2}\rangle, \\ \dfrac{q^{m-1} + (-1)^r(d-1)q^{m/2-1}}{d}, & \text{если } a \notin \langle\gamma^d\rangle \text{ и } a \in \langle\gamma^{d/2}\rangle. \end{cases}$$

В частности, если $q = p$, то $d = d_1$ и число $N(z,u)$ появлений каждого ненулевого элемента $z \in P$ удовлетворяет указанным формулам.

Доказательство. Пусть $z \in P$, $z \neq 0$, тогда из равенства (5) получим

$$N(z,u) = \frac{T(u)}{q} + \frac{\sigma_z(u)}{q},$$

где $\sigma_z(u) = \sum\limits_{c\in P^*} \chi(-cz) \sum\limits_{i=0}^{T(u)-1} \chi(cu(i))$. С использованием формулы (7) получим

$$\sigma_z(u) = \frac{1}{d}\left(1 - \sum_{\psi'\in B\setminus\{\psi'_0\}} \psi'(a)G(\overline{\psi'},\chi') + \sum_{\psi'\in A\setminus B} \psi'(a)G(\overline{\psi'},\chi')G(\psi,\lambda_z)\right),$$

где B (A) — множество всех мультипликативных характеров поля Q, аннулирующих группу $\langle\alpha, P^*\rangle$ ($\langle\alpha\rangle$). Как показано при доказательстве теоремы 7, $|B| = d_1$, а по доказательству теоремы 2 $|A| = (q^m-1)/|\langle\alpha\rangle| = d$. По условию $d = d_1$, поэтому $A = B$ и

$$\sigma_z(u) = \frac{1}{d}\left(1 - \sum_{\psi'\in B\setminus\{\psi'_0\}} \psi'(a)G(\overline{\psi'},\chi')\right).$$

Таким образом, $\sigma_z(u)$, а значит, $N(z,u)$ не зависят от элемента $z \in P^*$ и справедливо равенство

$$N(z,u) = (T(u) - N(0,u))/(q-1).$$

Подставляя в полученную формулу значения $N(0,u)$, подсчитанные в теореме 7, получим искомые равенства.

Заметим, что если $q = p$, то из соотношений

$$d \,|\, (p^l+1), \quad d \,|\, (p^l+1)(p^l-1), \quad (p^l+1)(p^l-1) \,|\, (p^m-1), \quad (p-1) \,|\, (p^l-1)$$

получим $(p-1) \,|\, (p^m-1)/d$, а значит,

$$d_1 = \frac{d((p^m-1)/d, p-1)}{p-1} = d.$$

Теорема доказана. ■

Отметим, что типы распределений, указанные в теореме 8, ранее были получены в [6, теорема 6] для случая $q = p$.

Рассмотрим наиболее интересный случай $p = 2$.

Следствие 4. Пусть $f(x)$ — неприводимый многочлен степени m над полем $P = \mathrm{GF}(2)$, $T(f) = (2^m-1)/3$, тогда $m = 2\lambda$ и для каждой ненулевой ЛРП u, имеющей представление $u(i) = \mathrm{tr}_P^Q(a\alpha^i)$, $i \geqslant 0$, где $Q = \mathrm{GF}(2^m)$, $a, \alpha \in Q$, справедливы равенства

$$N(0,u) = \begin{cases} \dfrac{2^{m-1} + (-1)^\lambda 2^{\lambda-1} - 1}{3}, & \text{если } a \text{ не является кубом в } Q, \\ \dfrac{2^{m-1} + (-1)^{\lambda-1} 2^\lambda - 1}{3}, & \text{если } a \text{ является кубом в } Q; \end{cases}$$

$$N(1,u) = \begin{cases} \dfrac{2^{m-1} + (-1)^{\lambda-1} 2^{\lambda-1}}{3}, & \text{если } a \text{ не является кубом в } Q, \\ \dfrac{2^{m-1} + (-1)^\lambda 2^\lambda}{3}, & \text{если } a \text{ является кубом в } Q. \end{cases}$$

Доказательство. Достаточно заметить, что в условиях теорем 7 и 8 $l = 1$, $s = 1$, $d = d_1 = 3$. ■

Следствие 5. Пусть $f(x)$ — неприводимый многочлен степени m над полем $P = \mathrm{GF}(2)$, $T(f) = (2^m - 1)/5$, тогда $m = 2\lambda$, λ — чётное число и для каждой ненулевой ЛРП u, имеющей представление $u(i) = \mathrm{tr}_P^Q(a\alpha^i)$, $i \geqslant 0$, где $Q = \mathrm{GF}(2^m)$, $a, \alpha \in Q$, справедливы равенства

$$N(0,u) = \begin{cases} \dfrac{2^{m-1} + (-1)^{\lambda/2} 2^{\lambda-1} - 1}{5}, & \text{если не существует } \sqrt[5]{a} \text{ в поле } Q, \\ \dfrac{2^{m-1} + (-1)^{\lambda/2-1} 2^{\lambda+1} - 1}{5}, & \text{если существует } \sqrt[5]{a} \text{ в поле } Q; \end{cases}$$

$$N(1,u) = \begin{cases} \dfrac{2^{m-1} + (-1)^{\lambda/2-1} 2^{\lambda-1}}{5}, & \text{если не существует } \sqrt[5]{a} \text{ в поле } Q, \\ \dfrac{2^{m-1} + (-1)^{\lambda/2} 2^{\lambda+1}}{5}, & \text{если существует } \sqrt[5]{a} \text{ в поле } Q. \end{cases}$$

Доказательство. Достаточно заметить, что в условиях теорем 7 и 8 $l = 2$, $s = 1$, $d = d_1 = 5$. ■

По аналогии со следствиями 4 и 5 нетрудно получить точные значения для числа элементов на цикле любой двоичной ЛРП u с неприводимым характеристическим многочленом $f(x) \in \mathrm{GF}(2)[x]$ степени m и периода $(2^m - 1)/d$, где $d = 9, 11, 13, 17$, а также для других чисел d, при которых число 2 полупримитивно по модулю d (см. таблицу таких чисел на с. 47).

8. Полупримитивный случай, $d \neq d_1$, $d_1 = 1$. Ненулевые элементы на циклах

Рассмотрим теперь частный случай, когда $d > 2$ и

$$d_1 = \left(\frac{q^m - 1}{q - 1}, d \right) = 1.$$

Согласно доказательству теоремы 7, в этой ситуации имеют место соотношения

$$\left(\frac{q^m - 1}{d}, q - 1 \right) = \frac{d_1(q-1)}{d} = \frac{q-1}{d}.$$

В дальнейшем нам понадобятся несколько вспомогательных результатов.

Лемма 7. Пусть ψ' — мультипликативный характер порядка d поля $Q = \mathrm{GF}(q^m)$, $d_1 = ((q^m - 1)/(q-1), d)$. Тогда ограничение ψ характера ψ' на поле $P = \mathrm{GF}(q)$ является мультипликативным характером порядка d/d_1.

Доказательство. Пусть ψ_1' — мультипликативный характер поля Q порядка $\operatorname{ord} \psi_1' = q^m - 1$, определённый равенствами

$$\psi_1'(\gamma^j) = e^{2\pi i (j/(q^m-1))}, \quad j = 0, 1, \ldots, q^m - 2,$$

где γ — примитивный элемент поля Q. Характер ψ' в силу равенства $\operatorname{ord} \psi' = d$ принадлежит циклической группе, порождённой характером $(\psi_1')^{(q^m-1)/d}$. Найдётся натуральное число k, такое, что $\psi' = (\psi_1')^{(q^m-1)k/d}$, причём $(d, k) = 1$. Пусть $\gamma_0 = \gamma^{(q^m-1)/(q-1)}$. Ясно, что γ_0 является примитивным элементом поля $P = \mathrm{GF}(q)$. Справедливы равенства

$$\psi(\gamma_0^j) = \psi'(\gamma_0^j) = (\psi_1')^{(q^m-1)k/d}(\gamma_0^j) = (\psi_1')^{(q^m-1)k/d}(\gamma^{(q^m-1)j/(q-1)}) = e^{2\pi i((q^m-1)/d)kj/(q-1)} = \psi(\gamma_0)^j.$$

Отсюда получим

$$\operatorname{ord}\psi = \min\left\{j \in \mathbb{N} : (q-1) \,\Big|\, \frac{q^m-1}{d}kj\right\} =$$
$$= \min\left\{j \in \mathbb{N} : \frac{q-1}{((q-1),(q^m-1)/d)} \,\Big|\, \frac{(q^m-1)/d}{((q-1),(q^m-1)/d)}kj\right\} =$$
$$= \min\left\{j \in \mathbb{N} : \frac{q-1}{((q-1),(q^m-1)/d)} \,\Big|\, kj\right\},$$

а так как $(q-1,(q^m-1)/d) = d_1(q-1)/d$, то, учитывая $(k,d)=1$, будем иметь

$$\operatorname{ord}\psi = \min\left\{j \in \mathbb{N} : \frac{d}{d_1} \,\Big|\, kj\right\} = \min\left\{j \in \mathbb{N} : \frac{d}{d_1} \,\Big|\, j\right\} = \frac{d}{d_1}.$$

Лемма доказана. ■

Лемма 8. Пусть в условиях леммы 7 $d_1 = 1$, тогда для каждого $i = 1, 2, \ldots, d-1$

$$G(\psi^i,\chi) = \begin{cases} (-1)^{(i+1)r/m-1}\sqrt{q}, & \text{если } d \text{ чётно, а } \dfrac{p^l+1}{d} \text{ нечётно,} \\ (-1)^{r/m-1}\sqrt{q}, & \text{если } d \text{ нечётно или } \dfrac{p^l+1}{d} \text{ чётно,} \end{cases}$$

где χ — канонический аддитивный характер поля P; параметры l и r определены в теореме 6.

Доказательство. Согласно лемме 7, характер ψ имеет порядок d. Число l — наименьшее из натуральных j, таких, что $d \,|\, (p^j+1)$. Из равенства $((q^m-1)/d, q-1) = (q-1)/d$ следует, что число d делит $q-1 = p^s-1$. Отсюда с использованием леммы 3 (подставляя $m=1$) получим $2l \,|\, s$. Для завершения доказательства остаётся воспользоваться теоремой 6 (применённой к случаю $m=1$). ■

Следующая лемма доказывается аналогично лемме 6.

Лемма 9. Пусть ψ' — мультипликативный характер поля Q порядка d, γ — примитивный элемент поля Q. Тогда для каждого $a \in Q^*$:

1) $\psi'(a) = 1$ тогда и только тогда, когда $a \in \langle\gamma^d\rangle$;
2) $\psi'(a) = -1$ тогда и только тогда, когда d — чётное число, $a \notin \langle\gamma^d\rangle$ и $a \in \langle\gamma^{d/2}\rangle$.

Теорема 9. Пусть в условиях теоремы 7 $d_1 = 1$, $z \neq 0$. Тогда:

1) если $r(m+1)/m$ — чётное, или d — нечётное, или $(p^l+1)/d$ — чётное, то

$$N(z,u) = \begin{cases} \dfrac{q^{m-1} + (-1)^{r(m+1)/m}(d-1)q^{(m-1)/2}}{d}, & \text{если } -z^{-1}a \in \langle\gamma^d\rangle, \\ \dfrac{q^{m-1} - (-1)^{r(m+1)/m}q^{(m-1)/2}}{d}, & \text{если } -z^{-1}a \notin \langle\gamma^d\rangle; \end{cases}$$

2) если $r(m+1)/m$ — нечётное, d — чётное, $(p^l+1)/d$ — нечётное, то

$$N(z,u) = \begin{cases} \dfrac{q^{m-1} - (d-1)q^{(m-1)/2}}{d}, & \text{если } -az^{-1} \notin \langle\gamma^d\rangle \text{ и } -az^{-1} \in \langle\gamma^{d/2}\rangle, \\ \dfrac{q^{m-1} + q^{(m-1)/2}}{d}, & \text{если } -az^{-1} \in \langle\gamma^d\rangle \text{ или } -az^{-1} \notin \langle\gamma^{d/2}\rangle. \end{cases}$$

Доказательство. С использованием равенства (5) и теоремы 2 получим

$$|B| = d_1 = 1, \quad N(z,u) = \frac{T(u)}{q} + \frac{\sigma_z(u)}{q},$$

где

$$\sigma_z(u) = \frac{1}{d} + \frac{1}{d}\sum_{\psi' \in A\setminus\{\psi'_0\}} \psi'(a)G(\overline{\psi'},\chi')G(\psi,\lambda_z) = \frac{1}{d} + \frac{1}{d}\sum_{\psi' \in A\setminus\{\psi'_0\}} \overline{\psi'(a)}G(\psi',\chi')G(\bar{\psi},\lambda_z).$$

Рассмотрим два случая.

С л у ч а й 1. Пусть d — нечётное число или $(p^l+1)/d$ — чётное число. Тогда по теореме 6

$$\sigma_z(u) = \frac{1}{d} + \frac{1}{d}\sum_{i=1}^{d-1} (\bar{\psi}')^i(a)G((\psi')^i,\chi')G(\bar{\psi}^i,\lambda_z) =$$

$$= \frac{1}{d} + \frac{1}{d}\sum_{i=1}^{d-1} (\bar{\psi}')^i(a)(-1)^{r-1}q^{m/2}G(\bar{\psi}^i,\lambda_z) = \frac{1}{d} + \frac{1}{d}(-1)^{r-1}q^{m/2}\sum_{i=1}^{d-1} (\psi')^i(a)G(\psi^i,\lambda_z),$$

где ψ' — мультипликативный характер порядка d поля Q. Для суммы Гаусса $G(\psi^i,\lambda_z)$ имеем

$$G(\psi^i,\lambda_z) = \sum_{x\in P^*} \psi^i(x)\chi(-zx) = \sum_{y\in P^*} \psi^i(-z^{-1}y)\chi(y) = \psi^i(-z^{-1})G(\psi^i,\chi).$$

Поэтому с использованием леммы 8 получим

$$\sigma_z(u) = \frac{1}{d} + \frac{1}{d}(-1)^{r-1}q^{m/2}(-1)^{r/m-1}\sqrt{q}\sum_{i=1}^{d-1} (\psi')^i(a)\psi^i(-z^{-1}) =$$

$$= \frac{1}{d} + \frac{1}{d}(-1)^{r(m+1)/m}q^{(m+1)/2}\sum_{i=1}^{d-1} (\psi')^i(-z^{-1}a).$$

Остаётся вычислить сумму

$$\sum_{i=1}^{d-1} (\psi')^i(-z^{-1}a) = -1 + \sum_{i=0}^{d-1} (\psi')^i(-z^{-1}a) = \begin{cases} d-1, & \text{если } \psi'(-z^{-1}a) = 1, \\ -1, & \text{если } \psi'(-z^{-1}a) \neq 1, \end{cases}$$

которая, согласно лемме 9, принимает вид

$$\sum_{i=1}^{d-1} (\psi')^i(-z^{-1}a) = \begin{cases} d-1, & \text{если } -z^{-1}a \in \langle\gamma^d\rangle, \\ -1, & \text{если } -z^{-1}a \notin \langle\gamma^d\rangle. \end{cases}$$

Таким образом,

$$\sigma_z(u) = \begin{cases} \dfrac{1}{d} + \dfrac{1}{d}(-1)^{r(m+1)/m}q^{(m+1)/2}(d-1), & \text{если } -z^{-1}a \in \langle\gamma^d\rangle, \\ \dfrac{1}{d} - \dfrac{1}{d}(-1)^{r(m+1)/m}q^{(m+1)/2}, & \text{если } -z^{-1}a \notin \langle\gamma^d\rangle, \end{cases}$$

и значит,

$$N(z,u) = \begin{cases} \dfrac{q^{m-1} + (-1)^{r(m+1)/m}(d-1)q^{(m-1)/2}}{d}, & \text{если } -z^{-1}a \in \langle\gamma^d\rangle, \\ \dfrac{q^{m-1} - (-1)^{r(m+1)/m}q^{(m-1)/2}}{d}, & \text{если } -z^{-1}a \notin \langle\gamma^d\rangle. \end{cases}$$

С л у ч а й 2. Пусть d — чётное число, $(p^l+1)/d$ — нечётное число. С использованием теоремы 6 получим

$$\sigma_z(u) = \frac{1}{d} + \frac{1}{d}\sum_{i=1}^{d-1} (\bar{\psi}')^i(a)G((\psi')^i, \chi')G(\bar{\psi}^i, \lambda_z) =$$

$$= \frac{1}{d} + \frac{1}{d}\sum_{i=1}^{d-1} (\bar{\psi}')^i(a)(-1)^{(i+1)r-1}q^{m/2}G(\bar{\psi}^i, \lambda_z) = \frac{1}{d} + \frac{q^{m/2}}{d}\sum_{i=1}^{d-1} (\psi')^i(a)(-1)^{(i+1)r-1}G(\psi^i, \lambda_z).$$

По аналогии с доказательством случая 1, используя лемму 8, будем иметь

$$\sigma_z(u) = \frac{1}{d} + \frac{q^{(m+1)/2}}{d}\sum_{i=1}^{d-1} (\psi')^i(a)(-1)^{(i+1)r-1}\psi^i(-z^{-1})(-1)^{(i+1)r/m-1} =$$

$$= \frac{1}{d} + \frac{q^{(m+1)/2}}{d}\sum_{i=1}^{d-1} (\psi')^i(-z^{-1}a)(-1)^{(i+1)(m+1)r/m} =$$

$$= \frac{1}{d} + \frac{(-1)^{(m+1)r/m}q^{(m+1)/2}}{d}\sum_{i=1}^{d-1} (\psi')^i(-z^{-1}a)(-1)^{i(m+1)r/m}.$$

Рассмотрим два подслучая:

2а) Если число $\dfrac{(m+1)r}{m} = \dfrac{(m+1)s}{2l}$ чётно, то по аналогии со случаем 1 получим

$$N(z,u) = \frac{T(u)}{q} + \frac{\sigma_z(u)}{q} = \begin{cases} \dfrac{q^{m-1} + (d-1)q^{(m-1)/2}}{d}, & \text{если } -z^{-1}a \in \langle\gamma^d\rangle, \\ \dfrac{q^{m-1} - q^{(m-1)/2}}{d}, & \text{если } -z^{-1}a \notin \langle\gamma^d\rangle. \end{cases}$$

2б) Если число $\dfrac{(m+1)r}{m} = \dfrac{(m+1)s}{2l}$ нечётно, то по аналогии со случаем 1 получим

$$\sigma_z(u) = \frac{1}{d} - \frac{q^{(m+1)/2}}{d}\sum_{i=1}^{d-1} (\psi')^i(-z^{-1}a)(-1)^i =$$

$$= \frac{1}{d} - \frac{q^{(m+1)/2}}{d} \cdot \begin{cases} d-1, & \text{если } \psi'(-z^{-1}a) = -1, \\ -1, & \text{если } \psi'(-z^{-1}a) \neq -1 \end{cases} =$$

$$= \begin{cases} \dfrac{1}{d} - \dfrac{q^{(m+1)/2}(d-1)}{d}, & \text{если } -az^{-1} \notin \langle\gamma^d\rangle \text{ и } -az^{-1} \in \langle\gamma^{d/2}\rangle, \\ \dfrac{1}{d} + \dfrac{q^{(m+1)/2}}{d}, & \text{если } -az^{-1} \in \langle\gamma^d\rangle \text{ или } -az^{-1} \notin \langle\gamma^{d/2}\rangle. \end{cases}$$

Таким образом,

$$N(z,u) = \begin{cases} \dfrac{q^{m-1} - (d-1)q^{(m-1)/2}}{d}, & \text{если } -az^{-1} \notin \langle\gamma^d\rangle \text{ и } -az^{-1} \in \langle\gamma^{d/2}\rangle, \\ \dfrac{q^{m-1} + q^{(m-1)/2}}{d}, & \text{если } -az^{-1} \in \langle\gamma^d\rangle \text{ или } -az^{-1} \notin \langle\gamma^{d/2}\rangle. \end{cases}$$

Теорема доказана. ■

Заметим, что теоремы 8 и 9 позволяют найти точные значения частот $N(z,u)$, $z \neq 0$, в каждой ЛРП u периода $T(u) = (q^m-1)/d$, где $d > 2$, d — произвольное простое число,

такое, что p полупримитивно по модулю d (в этом случае $d_1 = 1$ или $d_1 = d$). Кроме того, напомним, что теорема 7 описывает частоты $N(0, u)$ при каждом не обязательно простом числе d.

Приведём несколько наиболее интересных следствий из полученных результатов. Пусть u — ЛРП порядка m периода $T(u) = (2^{8m} - 1)/3$ над полем $\mathrm{GF}(2^8)$. Нетрудно видеть, что 3 делит $2^{8m} - 1$ при всех $m \in \mathbb{N}$, при этом

$$d_1 = \left(\frac{2^{8m} - 1}{2^8 - 1}, 3\right) = \begin{cases} 3, & \text{если 3 делит } m, \\ 1, & \text{если 3 не делит } m. \end{cases}$$

Каждая регулярная выборка из ЛРП максимального периода $2^{8m} - 1$ с шагом выбора 3 приведёт к рассматриваемой последовательности u [4].

Следствие 6. Пусть $f(x)$ — неприводимый многочлен степени m над полем $P = \mathrm{GF}(q) = \mathrm{GF}(2^8)$, $T(f) = (2^{8m} - 1)/3$, тогда для каждой ненулевой ЛРП u, имеющей представление $u(i) = \mathrm{tr}_P^Q(a\alpha^i)$, $i \geqslant 0$, где $Q = \mathrm{GF}(2^{8m})$, $a, \alpha \in Q$, справедливы следующие равенства:

1) если m кратно 3, то

$$N(0, u) = \begin{cases} \dfrac{q^{m-1} + (q-1)q^{m/2-1} - 1}{3}, & \text{если } a \text{ не является кубом в } Q, \\ \dfrac{q^{m-1} - 2(q-1)q^{m/2-1} - 1}{3}, & \text{если } a \text{ является кубом в } Q; \end{cases}$$

для $z \neq 0$

$$N(z, u) = \begin{cases} \dfrac{q^{m-1} - q^{m/2-1}}{3}, & \text{если } a \text{ не является кубом в } Q, \\ \dfrac{q^{m-1} + 2q^{m/2-1}}{3}, & \text{если } a \text{ является кубом в } Q; \end{cases}$$

2) если m не кратно 3, то $N(0, u) = \dfrac{q^{m-1} - 1}{3}$; для $z \neq 0$

$$N(z, u) = \begin{cases} \dfrac{q^{m-1} + 2q^{(m-1)/2}}{3}, & \text{если } z^{-1}a \text{ является кубом в } Q, \\ \dfrac{q^{m-1} - q^{(m-1)/2}}{3}, & \text{если } z^{-1}a \text{ не является кубом в } Q. \end{cases}$$

Доказательство. Достаточно заметить, что в условиях теорем 7–9 $d = 3$, $l = 1$, $s = 8$, $r = 4m$. ■

Пусть u — ЛРП порядка m периода $T(u) = (2^{8m} - 1)/5$ над полем $\mathrm{GF}(2^8)$. Нетрудно видеть, что 5 делит $2^{8m} - 1$ при всех $m \in \mathbb{N}$, при этом

$$d_1 = \left(\frac{2^{8m} - 1}{2^8 - 1}, 5\right) = \begin{cases} 5, & \text{если 5 делит } m, \\ 1, & \text{если 5 не делит } m. \end{cases}$$

Каждая регулярная выборка из ЛРП максимального периода $2^{8m} - 1$ с шагом выбора 5 приведёт к рассматриваемой последовательности u [4].

Следствие 7. Пусть $f(x)$ — неприводимый многочлен степени m над полем $P = \mathrm{GF}(q) = \mathrm{GF}(2^8)$, $T(f) = (2^{8m} - 1)/5$, тогда для каждой ненулевой ЛРП u, имеющей

представление $u(i) = \mathrm{tr}_P^Q(a\alpha^i)$, $i \geqslant 0$, где $Q = \mathrm{GF}(2^{8m})$, $a, \alpha \in Q$, справедливы следующие равенства:

1) если m кратно 5, то

$$N(0,u) = \begin{cases} \dfrac{q^{m-1} + (q-1)q^{m/2-1} - 1}{5}, & \text{если } a \text{ не является корнем пятой степени в } Q, \\ \dfrac{q^{m-1} - 4(q-1)q^{m/2-1} - 1}{5}, & \text{если } a \text{ является корнем пятой степени в } Q; \end{cases}$$

для $z \neq 0$

$$N(z,u) = \begin{cases} \dfrac{q^{m-1} - q^{m/2-1}}{5}, & \text{если } a \text{ не является корнем пятой степени в } Q, \\ \dfrac{q^{m-1} + 4q^{m/2-1}}{5}, & \text{если } a \text{ является корнем пятой степени в } Q; \end{cases}$$

2) если m не кратно 5, то $N(0,u) = \dfrac{q^{m-1} - 1}{5}$; для $z \neq 0$

$$N(z,u) = \begin{cases} \dfrac{q^{m-1} + 4q^{(m-1)/2}}{5}, & \text{если } z^{-1}a \text{ является корнем пятой степени в } Q, \\ \dfrac{q^{m-1} - q^{(m-1)/2}}{5}, & \text{если } z^{-1}a \text{ не является корнем пятой степени в } Q. \end{cases}$$

Доказательство. Достаточно заметить, что в условиях теорем 7–9 $d = 5$, $l = 2$, $s = 8$, $r = 2m$. ■

Приведём таблицы небольших модулей, по которым полупримитивны простые числа 2, 3, 5, 7. При этом будем использовать введённые в лемме 3 обозначения d и l.

$p = 2$

d	3	5	9	11	13	17	19	25	27	29	33	37	41	43	53	57
l	1	2	3	5	6	4	9	10	9	14	5	18	10	7	26	9

$p = 3$

d	4	5	7	10	14	17	19	25	28	29	31	34	37	38	41	43
l	1	2	3	2	3	8	9	10	3	14	15	8	9	9	4	21

$p = 5$

d	3	6	7	9	13	14	17	18	21	23	26	27	29	34	37	41
l	1	1	3	3	2	3	8	3	3	11	2	9	7	8	18	10

$p = 7$

d	4	5	8	10	11	13	17	22	23	25	26	34	41	43	44	46
l	1	2	1	2	5	6	8	5	11	2	6	8	20	3	5	11

9. Полупримитивный случай, $d \neq d_1$, $d_1 \neq 1$. Ненулевые элементы на циклах

Сначала рассмотрим частный случай, когда $d > 2$ и

$$d_1 = \left(\frac{q^m - 1}{q - 1}, d\right) = 2.$$

В этом случае, очевидно, подразумевается, что d чётно, $q = p^s$ нечётно и m чётно.

Лемма 10. В условиях теоремы 7 при $d > 2$ и $d_1 = \left(\dfrac{q^m - 1}{q - 1}, d\right) = 2$ для любого ненулевого элемента $z \in \mathrm{GF}(q)$ справедливо равенство

$$\sigma_z(u) = \frac{1}{d} + \frac{1}{d}\sum_{i=1}^{d-1} \bar{\psi}'^i(a) G(\psi'^i, \chi') G(\bar{\psi}^i, \lambda_z).$$

Доказательство. Для доказательства рассмотрим отдельно каждую из сумм равенства (7):

$$\sigma_z(u) = \frac{1}{d} - \frac{1}{d}\sum_{\psi' \in B\setminus\{\psi'_0\}} \psi'(a)G(\bar{\psi}', \chi') + \frac{1}{d}\sum_{\psi' \in A\setminus B} \psi'(a)G(\bar{\psi}', \chi')G(\psi, \lambda_z).$$

В рассматриваемом случае группа B есть группа $\langle \psi'^{d/2} \rangle$, где ψ' — мультипликативный характер порядка d поля $Q = \mathrm{GF}(q^m)$. Тогда первую сумму можно переписать следующим образом:

$$W = \sum_{\psi'' \in B\setminus\{\psi'_0\}} \psi''(a)G(\bar{\psi}'', \chi') = \psi'^{d/2}(a)G(\bar{\psi}'^{d/2}, \chi') = \eta'(a)G(\eta', \chi'),$$

где η' — квадратичный характер поля Q. Теперь по теореме 3 имеем

$$W = \begin{cases} -\eta'(a)q^{m/2}, & \text{если } p \equiv 1 \pmod 4, \\ -\eta'(a)(-1)^{sm/2}q^{m/2}, & \text{если } p \equiv 3 \pmod 4. \end{cases}$$

Вторую сумму равенства (7) перепишем следующим образом:

$$V = \sum_{\psi'' \in A\setminus B} \psi''(a)G(\bar{\psi}'', \chi')G(\psi, \lambda_z) = \sum_{i=1}^{d-1} \bar{\psi}'^i(a)G(\psi'^i, \chi')G(\bar{\psi}^i, \lambda_z) - W \cdot G(\psi^{d/2}, \lambda_z),$$

где ψ — ограничение характера ψ' на поле P.

По лемме 2 $\psi^{d/2} = \psi_0$ и, значит, $G(\psi^{\frac{d}{2}}, \lambda_z) = -1$. Таким образом,

$$V = \sum_{i=1}^{d-1} \bar{\psi}'^i(a)G(\psi'^i, \chi')G(\bar{\psi}^i, \lambda_z) + W$$

и

$$\begin{aligned}\sigma_z(u) &= \frac{1}{d} - \frac{1}{d}W + \frac{1}{d}\left(\sum_{i=1}^{d-1} \bar{\psi}'^i(a)G(\psi'^i, \chi')G(\bar{\psi}^i, \lambda_z) + W\right) = \\ &= \frac{1}{d} + \frac{1}{d}\sum_{i=1}^{d-1} \bar{\psi}'^i(a)G(\psi'^i, \chi')G(\bar{\psi}^i, \lambda_z).\end{aligned}$$

Лемма доказана. ■

Далее, воспользуемся идеей доказательства теоремы 9, в которой подсчитано значение суммы $\sigma_z(u)$ при условии, что порядок характера ψ есть d (а в нашем случае, согласно лемме 7, $d/2$), и докажем справедливость следующей теоремы.

Теорема 10. Пусть в условиях теоремы 7 $d > 2$, $d_1 = \left(\dfrac{q^m - 1}{q - 1}, d\right) = 2$, l_1 — наименьшее натуральное число, такое, что $\dfrac{d}{2} \Big| (p^{l_1} + 1)$, $r_1 = \dfrac{ms}{2l_1}$. Тогда:

1) если $\dfrac{p^l + 1}{d}$ чётно, то

$$\sigma_z(u) = \begin{cases} \dfrac{1}{d} + \dfrac{1}{d}(-1)^r q^{(m+1)/2} \displaystyle\sum_{i=1}^{d-1} (-1)^{(i+1)r_1/m} \psi'^i(-z^{-1}a), \text{если } d/2 \text{ чётно, а } \dfrac{p^{l_1}+1}{d/2} \text{ нечётно}, \\ \dfrac{1}{d} + \dfrac{1}{d}(-1)^{r+r_1/m} q^{(m+1)/2} \displaystyle\sum_{i=1}^{d-1} \psi'^i(-z^{-1}a), \text{если } d/2 \text{ нечётно или } \dfrac{p^{l_1}+1}{d/2} \text{ чётно}; \end{cases}$$

2) если $d/2$ нечётно или $\dfrac{p^{l_1}+1}{d/2}$ чётно, то

$$\sigma_z(u)=\begin{cases}\dfrac{1}{d}+\dfrac{1}{d}(-1)^{r+r_1/m}q^{(m+1)/2}(d-1), & \text{если } \psi'(-z^{-1}a)=1,\\ \dfrac{1}{d}-\dfrac{1}{d}(-1)^{r+r_1/m}q^{(m+1)/2}, & \text{если } \psi'(-z^{-1}a)\neq 1.\end{cases}$$

Доказательство. Напомним, что $G(\psi^i,\lambda_z)=\psi^i(-z^{-1})G(\psi^i,\chi)$ и $\left(\dfrac{q^m-1}{q-1},d\right)=2$. Из последнего равенства следует, что либо $d|(q-1)$, либо, если это не так, то $\dfrac{d}{2}\Big|(q-1)$ и, значит, $d|(q^2-1)$.

Рассмотрим случай $d|(q-1)$. Согласно теореме 6, для каждого $i=1,2,\ldots,d-1$

$$G(\psi^i,\chi)=\begin{cases}(-1)^{(i+1)r_1/m-1}\sqrt{q}, & \text{если } d/2 \text{ чётно, а } \dfrac{p^{l_1}+1}{d/2} \text{ нечётно},\\ (-1)^{r_1/m-1}\sqrt{q}, & \text{если } d/2 \text{ нечётно или } \dfrac{p^{l_1}+1}{d/2} \text{ чётно};\end{cases}$$

$$G(\psi'^i,\chi')=\begin{cases}(-1)^{(i+1)r-1}q^{m/2}, & \text{если } \dfrac{p^l+1}{d} \text{ нечётно},\\ (-1)^{r-1}q^{m/2}, & \text{если } \dfrac{p^l+1}{d} \text{ чётно}.\end{cases}$$

Тогда если $\dfrac{p^l+1}{d}$ чётно, то

$$\sigma_z(u)=\frac{1}{d}+\frac{1}{d}\sum_{i=1}^{d-1}\bar{\psi}'^i(a)(-1)^{r-1}q^{m/2}G(\bar{\psi}^i,\lambda_z)=\frac{1}{d}+\frac{1}{d}(-1)^{r-1}q^{m/2}\sum_{i=1}^{d-1}\psi'^i(a)G(\psi^i,\lambda_z)=$$

$$=\frac{1}{d}+\frac{1}{d}(-1)^{r-1}q^{m/2}\sum_{i=1}^{d-1}\psi'^i(a)\psi^i(-z^{-1})G(\psi^i,\chi)=$$

$$=\begin{cases}\dfrac{1}{d}+\dfrac{1}{d}(-1)^{r}q^{(m+1)/2}\sum\limits_{i=1}^{d-1}(-1)^{(i+1)r_1/m}\psi'^i(-z^{-1}a), & \text{если } d/2 \text{ чётно, а } \dfrac{p^{l_1}+1}{d/2} \text{ нечётно},\\ \dfrac{1}{d}+\dfrac{1}{d}(-1)^{r+r_1/m}q^{(m+1)/2}\sum\limits_{i=1}^{d-1}\psi'^i(-z^{-1}a), & \text{если } d/2 \text{ нечётно или } \dfrac{p^{l_1}+1}{d/2} \text{ чётно}.\end{cases}$$

Во втором случае, когда $d/2$ нечётно или $\dfrac{p^{l_1}+1}{d/2}$ чётно, получим

$$\sigma_z(u)=\begin{cases}\dfrac{1}{d}+\dfrac{1}{d}(-1)^{r+r_1/m}q^{(m+1)/2}(d-1), & \text{если } \psi'(-z^{-1}a)=1,\\ \dfrac{1}{d}-\dfrac{1}{d}(-1)^{r+r_1/m}q^{(m+1)/2}, & \text{если } \psi'(-z^{-1}a)\neq 1.\end{cases}$$

Теорема доказана. ■

Следствие 8. В условиях теоремы 10 при нечётном $d/2$ или чётном $\dfrac{p^{l_1}+1}{d/2}$ для любого ненулевого элемента поля z справедливо

$$N(z,u)=\begin{cases}\dfrac{q^{m-1}+(-1)^{r+r_1/m}q^{(m-1)/2}(d-1)}{d}, & \text{если } -z^{-1}a\in\langle\gamma^d\rangle,\\ \dfrac{q^{m-1}-(-1)^{r+r_1/m}q^{(m-1)/2}}{d}, & \text{если } -z^{-1}a\notin\langle\gamma^d\rangle.\end{cases}$$

ЛИТЕРАТУРА

1. *Глухов М. М., Елизаров В. П., Нечаев А. А.* Алгебра: учебник. Т. 2. М.: Гелиос АРВ, 2003. 416 с.
2. *Лидл Р., Нидеррайтер Г.* Конечные поля. М.: Мир, 1988.
3. *McEliece R. J.* Irreducible cyclic codes and Gauss sums // Combinatorics. 1975. P. 185–202.
4. *Цирлер Н.* Линейные возвратные последовательности // Кибернетический сборник. 1963. № 6. С. 55–79.
5. *Лаксов Д.* Линейные рекуррентные последовательности над конечными полями // Математика. Сборник переводов. 1967. Т. 11. № 6. С. 145–158.
6. *Baumert L. D. and McEliece R. J.* Weights of irreducible cyclic codes // Information and Control. 1972. V. 20. P. 158–175.
7. *Nelubin A. S.* Distribution of elements on cycles of linear recurrences over Galois fields // Formal Power Series and Algebraic Combinatorics. 12-th Intern. Conf. FPSAC. Moscow, 2000. P. 534–542.

REFERENCES

1. *Glukhov M. M., Elizarov V. P., and Nechaev A. A.* Algebra [Algebra], vol. 2. Moscow, Gelios ARV Publ., 2003. 416 p. (in Russian)
2. *Lidl R. and Niederreiter H.* Finite Fields. Addison-Wesley Publ., 1983.
3. *McEliece R. J.* Irreducible cyclic codes and Gauss sums. Combinatorics, 1975, pp. 185–202.
4. *Tsirler N.* Lineynye vozvratnye posledovatel'nosti [Linear recurrence sequences]. Kiberneticheskiy Sbornik, 1963, no. 6, pp. 55–79. (in Russian)
5. *Laksov D.* Lineynye rekurrentnye posledovatel'nosti nad konechnymi polyami [Linear recurring sequences over finite fields]. Matematika. Sbornik perevodov, 1967, vol. 11, no. 6, pp. 145–158. (in Russian)
6. *Baumert L. D. and McEliece R. J.* Weights of irreducible cyclic codes. Information and Control, 1972, vol. 20, pp. 158–175.
7. *Nelubin A. S.* Distribution of elements on cycles of linear recurrences over Galois fields. Formal Power Series and Algebraic Combinatorics. 12-th Intern. Conf. FPSAC, Moscow, 2000, pp. 534–542.

UDC 512.54 DOI 10.17223/20710410/36/4

ON SOLVABILITY OF REGULAR EQUATIONS IN THE VARIETY OF METABELIAN GROUPS[1]

V. A. Roman'kov

Dostoevsky Omsk State University, Omsk, Russia

We study the solvability of equations over groups within a given variety or another class of groups. The classes of nilpotent and solvable groups were considered as main classes to investigate from such point of view. The natural analogues of the famous Kervaire — Laudenbach and Levin conjectures were raised to the challenge. It was also noted that the "solvable" version of the known theorem by Brodskiĭ is not true. In this paper, for each $n \in \mathbb{N}, n \geqslant 2$, we prove that every regular equation over the free metabelian group M_n is solvable in the class $\mathcal{M}$ of all metabelian groups. Moreover, there is a metabelian group $\tilde{M}_n$ that contains a solution of every unimodular equation over M_n. These results are extended to the class of rigid metabelian groups. Also, we give an example showing that there exists an equation over a locally indicable torsion-free metabelian group G that has no solution in any solvable overgroup of G. It follows that solvable versions of the Levin conjecture are not true. Another example presents an unimodular equation over a locally indicable torsion-free metabelian group G that has no solution in any metabelian overgroup of G. Hence, the Kervaire — Laudenbach conjecture is not valid for the variety of all metabelian groups. We prove that there is an unimodular equation over a finite metabelian group G that has no solutions in any finite metabelian overgroup of G. This means that analog of the famous theorem by Gerstenhaber and Rothaus (about solvability of each unimodular equation over a finite group G in some finite overgroup of G) is not valid for the class of finite metabelian groups.

Keywords: *Kervaire — Laudenbach conjecture, Levin conjecture, solvable group, metabelian group, rigid group, nilpotent group, locally indicable group, regular equation, solvability over group.*

Introduction

The purpose of this note is to study solvability of equations over groups in varieties or other classes of groups, emphasizing the variety of metabelian groups.

The process of solving equations is central to much of mathematics. In general setting, there are two questions to answer when presented with an equation. The first question is about existing of a solution, the second one is about finding a solution if it exists. Equations in groups are the old and well-established area of group theory. For instance, see the surveys [1 – 4].

An equation in k variables over a group G is an expression of the form

$$u(x_1, \ldots, x_k) = 1, \tag{1}$$

where

$$u(x_1, \ldots, x_k) = g_0 x_{i_1}^{\varepsilon_1} g_1 \ldots g_{n-1} x_{i_n}^{\varepsilon_n} \in F(X) * G.$$

Here, each g_j is a group element, each exponent ε_j is 1 or -1, each x_{i_j} is taken from an alphabet of variables $X = \{x_1, \ldots, x_k, \ldots\}$, and $F(X)$ denotes the free group with basis X.

[1]This research was supported by Russian Science Foundation (project 16-11-10002).

For brevity, we often use the word "equation" to refer a word $u(x_1, \dots, x_k)$, i. e., the left-hand side of an equation of the form (1). Also, an equation can be written in the form $u(x_1, \dots, u_k) = v(x_1, \dots, x_k)$, where $u(x_1, \dots, x_k), v(x_1, \dots, x_k) \in F(X) * G$. This equation is equivalent to $u(x_1, \dots, x_k)v(x_1, \dots, x_k)^{-1} = 1$.

By the preceding convention, the free product $G_X = F(X) * G$ is the space of all equations with variables in X and coefficients in G. We assume that $n \geqslant 1$ and if $i_j = i_{j+1}$ and $\varepsilon_{i_j} + \varepsilon_{i_{j+1}} = 0$, then $g_j \neq 1$. Also, we assume that if $g_0 = 1$ and $i_1 = i_n$, then $\varepsilon_1 + \varepsilon_n \neq 0$. If not, we can conjugate the both sides of (1) by $x_{i_1}^{-\varepsilon_1}$ to get an equivalent equation with a shorter length. We exclude the case that u is conjugate to an element of G. The g_j's are called the *coefficients* or *constants* of the equation and n is called the *length* of the equation. For every positive integer k, let $X_k = \{x_1, \dots, x_k\} \subseteq X$. The free product $G_{X_k} = F(X_k) * G$ is the space of all equations over G in k variables.

A *solution* of (1) in G is an assignment $x_j \mapsto h_j \in G$ such that

$$g_0 h_{i_1}^{\varepsilon_1} g_1 \dots g_{n-1} h_{i_n}^{\varepsilon_n} = 1.$$

Let $u = u(x_1, \dots, x_k) \in G_X$ and let U denote the normal closure of u in the group G_X. We say that (the equation) $u = 1$ has a *solution* (equivalently, is *solvable*) over the group G if there is an overgroup H of G such that $u = 1$ has a solution in H. In this case, we say that $u = 1$ has a solution or $u = 1$ is *solvable*. Clearly, $u = 1$ has a solution if and only if the canonical map from G to the group $H = G_X/U$ is an embedding.

An one-variable equation of the form $g_0 x^{\varepsilon_1} g_1 \dots g_{n-1} x^{\varepsilon_n} = 1$ of length n with the exponent sum (*exponent*) $\varepsilon = \sum_{i=1}^{n} \varepsilon_i$ is called *regular* or *non-singular* if $\varepsilon \neq 0$, *unimodular* if $\varepsilon = 1$, and *singular* if $\varepsilon = 0$. Similarly, we call the underlined word $u = g_0 x^{\varepsilon_1} g_1 \dots g_{n-1} x^{\varepsilon_n}$ *regular, unimodular* or *singular*, respectively.

Not every singular equation is solvable. If $g, f \in G$ and $|g| \neq |f|$ (by $|v|$ we denote the order of element v), then the equation $gxfx^{-1} = 1$ is obviously non-solvable.

For any word $w = w(x_1, \dots, x_k) \in G_X$ and for each $j = 1, \dots, k$, we denote the exponent sum of x_j in w by $\exp_j(w)$. Let

$$\{u_1 = 1, \dots, u_m = 1\} \tag{2}$$

be a system of equations in variables $x_1, \dots, x_k$. Let $\sigma_{ij} = \exp_j(u_i)$ for $i = 1, \dots, m$; $j = 1, \dots, k$. The system (2) is called *independent* if the matrix (σ_{ij}) has rank m.

There are the following three major results on solvability of equations in groups.

Theorem 1 (M. Gerstenhaber, O. S. Rothaus, 1962 [5]). Every independent system of equations over a compact connected Lie group has a solution.

It follows that every independent system of equations over a finite group or, more generally, over a locally residually finite group [6] has a solution. In particular, every regular equation over a finite or, more generally, locally residually finite group has a solution.

Theorem 2 (A. A. Klyachko, 1993 [7]). If G is a torsion-free group, then every unimodular equation in one variable over G is solvable.

Theorem 3. (S. D. Brodskiĭ, 1980 [8], 1984 [9]; J. Howie, 1982 [10]; S. M. Gersten, 1985 [11]; S. Krstić, 1985 [12]). If G is locally indicable, then every independent system of equations over G is solvable. Moreover, if G is locally p-indicable for some prime number p, then every p-independent system of equations over G is solvable.

Recall that G is *locally indicable* if every finitely generated subgroup of G maps onto the infinite cyclic group C, and *locally p-indicable* (p is a prime) if every finitely generated subgroup of G maps onto the cyclic group C_p of order p. A system of m equations is called *p-independent* if the corresponding matrix (σ_{ij}) has rank m mod p.

S. D. Brodskiĭ proved the locally indicable statement in the case of a single equation. In fact, he proved that quotient $G{*}H/\mathrm{ncl}(w)$ of a free product $G{*}H$ of two non-trivial locally indicable groups by a normal closure $\mathrm{ncl}(w)$ of element $w \in G{*}H$, which is not conjugate to element of $G \cup H$, contains the natural images of the factors. Independently, J. Howie proved the locally indicable statement for arbitrary independent system. S. M. Gersten established the locally p-indicable version, and S. Krstić gave an alternative proof of the locally p-indicable statement.

Not every singular equation is solvable. If $g, f \in G$ and $|g| \neq |f|$, then the equation $gxfx^{-1} = 1$ is obviously non-solvable.

Since not every equation is solvable over an arbitrary group, some restrictions on group or equation are required. The best known restrictions are the Kervaire — Laudenbach conjecture and the Levin conjecture.

Conjecture 1 (**Kervaire — Laudenbach conjecture**). Every regular equation in one variable over an arbitrary group is solvable.

Conjecture 2 (**Levin conjecture**). Every equation over arbitrary torsion-free group is solvable.

In general, these two conjectures are still open.

At this point, we need to introduce the following notation: $[g, f]$ means the commutator $gfg^{-1}f^{-1}$, g^f stands for conjugate gfg^{-1}, G' means the derived subgroup (commutant) of group G, $\mathbb{Z}$ is the ring of integers, $\mathbb{N}$ is the set of positive integers, $\mathbb{Q}$ denotes the field of rationals. By $\mathbb{Z}[G]$, we denote the group ring of group G.

Recall that a group G is said to be *metabelian* if there exists a short exact sequence $A \hookrightarrow G \twoheadrightarrow B$ of groups with abelian A and B. In particular, one can take $A = G'$ and $B = G/G'$. Hence, metabelian groups are precisely the solvable groups of derived length at least two. We think of A (in particular, G') as a $\mathbb{Z}[B]$-module via conjugation; that is, we define $a \circ q = gag^{-1}$, where $g \mapsto q$ under the standard homomorphism $G \twoheadrightarrow B$. Since A is abelian, this is well-defined.

1. Solvability of equations over a group in a given class of groups

In [4], a new direction was initiated to study solvability of equations over groups within a given variety or another class of groups. The classes of nilpotent and solvable groups were considered as main classes to investigate from such point of view. The natural analogues of the Kervaire — Laudenbach and Levin conjectures were raised to the challenge.

When we study the solvability of equations over a group G in a given class $\mathcal{C}$ of groups, there is an important issue — what is the proper space of equations for a given group G? The free product G_X seems suitable if the class $\mathcal{C}$ coincides with the class $\mathcal{G}$ of all groups. On the other hand, if $\mathcal{C}$ is the class $\mathcal{A}$ of abelian groups (and G is an abelian group), then it is natural to consider an equation $g_1x_1^{-1}g_2x_2x_1 = 1$ as essentially the same as $g_1g_2x_2 = 1$. More generally, it is natural to consider two equations as essentially the same if one can be transformed into the other by applying identities of the variety of abelian groups. So the natural space of equations in variables X over an abelian group G, when we restrict ourselves by the class $\mathcal{A}$, is the direct product $G \times A(X)$ of G and a free abelian group $A(X)$ with basis X. Similarly, if G is in a variety $\mathcal{V}$, and we study solvability of equations over G in $\mathcal{V}$, then the space of equations in variables X over G *relative to* $\mathcal{V}$ is the free product in the

variety $\mathcal{V}$ of G and a $\mathcal{V}$-free group with basis X. Following this approach to a metabelian group G, we define the space of *metabelian* equations as the set of elements of a free product $G *_{\mathcal{M}} F_{\mathcal{M}}(X)$, where $\mathcal{M}$ is the variety of all metabelian groups, $*_{\mathcal{M}}$ is the free product in $\mathcal{M}$, and $F_{\mathcal{M}}(X)$ is a free metabelian group with basis X. Let M_n denote the free metabelian group of rank n.

2. Solvability of equations over a group in the variety $\mathcal{M}$ of metabelian groups: negative results

We are interesting in solvability of equations over metabelian groups in the class $\mathcal{M}$. We say that the equation $u = 1$ has a *solution* (equivalently, is *solvable*) over the metabelian group G in the class $\mathcal{M}$ if there is an overgroup $H \in \mathcal{M}$ of G such that $u = 1$ has a solution in H.

It has been noted in [4] that the "solvable" version of theorem 3 is not true. Indeed, an equation of the form

$$xgx^{-1} = [g, hgh^{-1}], \tag{3}$$

where g and h are elements of a torsion-free solvable group G, for which $[g, hgh^{-1}] \neq 1$ (such elements exist in every non-abelian solvable group), has no solution in any solvable overgroup of G. It also follows that solvable versions of the Levin conjecture are not true. Hence, these conjectures are not true in the class $\mathcal{M}$.

Indeed, let the equation (3) has a solution in a solvable group H of the derived length r and $H \geqslant G$. Suppose $g \in H^{(k)}$, where $H^{(k)}$ denotes the k-th member of the derived series of H. Then the right side of (3) belongs to the $(k+1)$-th member $H^{(k+1)} = [H^{(k)}, H^{(k)}]$ of the derived series of H. By continuing in this way, we get $g \in H^{(r)} = \{1\}$. This contradicts the our assumption.

The following example shows that there is not only regular even unimodular equation that is not solvable over some torsion-free metabelian group in the class $\mathcal{M}$.

Example 1. Let G be subgroup of group $\mathrm{GL}_2(\mathbb{Q})$ generated by two matrices:

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \text{ and } \mathrm{B} = \begin{pmatrix} -2 & 0 \\ 0 & 2 \end{pmatrix}.$$

Then the following unimodular equation is not solvable in every metabelian group $H \geqslant G$:

$$x^2 B x^{-1} B^{-1} = [A, B]. \tag{4}$$

Proof. Note that $[A, B] = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$. Let $x = h$ is a solution of the equation (4) in a metabelian group $H \geqslant G$. Then $h = [A, B][h, B]^{-1} \in H'$. We consider H' as a module over group ring $\mathbb{Z}[H]$. Group H acts on H' by conjugation. As H' acts trivially, we get a module H' over group ring $\mathbb{Z}[H/H']$. We denote the image of $f \in H$ in H/H' by $\bar{f}$. For each $\bar{f} \in H/H'$ and each $u \in H'$, we write $u^{\bar{f}}$ as a result of action of $\bar{f}$ on u. The equality corresponding to (4) can be written in the form

$$h^{2-\bar{B}} = [A, B].$$

Then

$$h^{(2-\bar{B})(2+\bar{B})} = h^{4-\bar{B}^2} = 1, \tag{5}$$

because $B^2 = 4E$ (E is identical matrix). Hence, the action of $2 + \bar{B}$ on the left-hand side of (5) gives trivial element as a result. But direct computation gives the following result of the action of $2 + \bar{B}$ on the right-hand side of equality (4):

$$[A, B]^{2+\bar{B}} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = [A, B]. \tag{6}$$

In view of (5), the nontriviality of the right-hand side of (6) contradicts our assumption about solvability of (4) in H. ■

The group G in Example 1 is obviosly torsion-free. Moreover, it is locally indicable. The equation (4) is unimodular. Thus, the statement of Example 1 implies that both Theorems 2 and 3 are not true in the class $\mathcal{M}$ of metabelian groups. Moreover, it is easy to modify Example 1 considering subgroup G_p of $\mathrm{GL}_2(\mathbb{Z}_p)$ generated by matrices A and B over the modular ring (finite prime field) $\mathbb{Z}_p$ (p is an odd prime) and to prove that the equation (4) is unsolvable in the class $\mathcal{M}_{\text{fin}}$ of all finite metabelian groups. The argument is practically the same as in the proof above. This shows that Theorem 1 is wrong in $\mathcal{M}_{\text{fin}}$.

3. Solvability of equations over a group in the variety $\mathcal{M}$ of metabelian groups: positive results

Let M_n be the free metabelian group of rank n with basis $\{f_1, \ldots, f_n\}$ and $A_n = M_n/M_n'$. Then A_n is a free abelian group with the basis $\{a_1, \ldots, a_n\}$ corresponding to the basis $\{f_1, \ldots, f_n\}$, that is, $f_i \mapsto a_i$ under the standard homomorphism $\mu : M_n \twoheadrightarrow A_n$, $i = 1, \ldots, n$. For brevity, we denote $\bar{g} = \mu(g)$ for each $g \in M_n$. Thus, $a_i = \bar{f}_i$, $i = 1, \ldots, n$.

Now, we apply the partial Fox derivatives and the Magnus embedding. For the basic notions and properties of these tools see [13].

The partial Fox derivatives on the group ring $\mathbb{Z}[M_n]$ may be defined as the mappings

$$\partial/\partial f_j : \mathbb{Z}[M_n] \to \mathbb{Z}[A_n] \quad \text{for} \quad 1 \leqslant j \leqslant n,$$

satisfying the following conditions for all $\alpha, \beta \in \mathbb{Z}$, $u, v \in \mathbb{Z}[M_n]$, and $g, h \in M_n$:

$$\begin{gathered} \partial f_i/\partial f_j = \delta_{ij} \text{ (where } \delta_{ij} \text{ is the Kronecker delta)}, \\ \partial(\alpha u + \beta v)/\partial f_j = \alpha \partial u/\partial f_j + \beta \partial v/\partial f_j, \\ \partial(gh)/\partial f_j = \partial g/\partial f_j + \bar{g} \cdot \partial h/\partial f_j. \end{gathered}$$

For any $n \in \mathbb{N}$, $n \geqslant 2$, the free metabelian group M_n can be treated via the Magnus embedding as a subgroup of the wreath product $W = A_n \mathrm{Wr} A_n$, where A_n is the free abelian group of rank n. In linear representation, the group W is the group of matrices of the form

$$\begin{pmatrix} a & \sum\limits_{i=1}^{n} u_i t_i \\ 0 & 1 \end{pmatrix},$$

where $\{t_1, \ldots, t_n\}$ is a basis of the free module T over $\mathbb{Z}[A_n]$, $a \in A_n$, $u_j \in \mathbb{Z}[A_n]$ for $j = 1, \ldots, n$. Hence,

$$W = \begin{pmatrix} A_n & \sum\limits_{i=1}^{n} \mathbb{Z}[A_n] \cdot t_i \\ 0 & 1 \end{pmatrix}.$$

The group M_n is embedded into W by the following map, where $g \in M_n$ and $\mu : M_n \to A_n$ is the standard homomorphism,

$$\nu : g \mapsto \begin{pmatrix} \mu(g) & \sum\limits_{j=1}^{n} \partial g/\partial f_j \cdot t_j \\ 0 & 1 \end{pmatrix}.$$

In particular, $\nu(f_i) = \begin{pmatrix} a_i & t_i \\ 0 & 1 \end{pmatrix}$ for $i = 1, \ldots, n$.

The group ring $\mathbb{Z}[A_n]$ is the ring of Laurent polynomials $\mathbb{Z}[a^{\pm 1}, \ldots, a_n^{\pm 1}]$.

This ring is an integral domain; so, we can consider the field of fractions $\mathrm{Frac}(\mathbb{Z}[A_n])$ of $\mathbb{Z}[A_n]$. Recall that the fraction u/v denotes the equivalence class of pairs (u, v), where (u, v) is equivalent to (r, s) if and only if $us = rv$. The field of fractions $\mathrm{Frac}(\mathbb{Z}[A_n])$ is defined as the set of all such fractions u/v. The sum of u/v and r/s is defined as $us + rv/vs$, and the product of u/v and r/s is defined as ur/vs. The embedding of $\mathbb{Z}[A_n]$ into $\mathrm{Frac}(\mathbb{Z}[A_n])$ maps each $u \in \mathbb{Z}[A_n]$ to the fraction $u/1$.

The wreath product W (thus, the free metabelian group M_n) can be considered as subgroup of the following metabelian group:

$$\tilde{W} = \begin{pmatrix} A_n & \sum\limits_{i=1}^{n} \mathrm{Frac}(\mathbb{Z}[A_n])t_i \\ 0 & 1 \end{pmatrix}.$$

Now, we are ready to prove the following theorem.

Theorem 4. For each $n \in \mathbb{N}$, $n \geqslant 2$, every regular equation over the free metabelian group M_n is solvable in the class $\mathcal{M}$ of all metabelian groups. Moreover, there is a metabelian group $\tilde{M}_n$ that contains a solution of every unimodular equation over M_n.

Proof. Let

$$g_0 \cdot x^{\varepsilon_1} \cdot g_1 \cdot \ldots \cdot g_{n-1} \cdot x^{\varepsilon_n} = 1 \tag{7}$$

be a regular equation over M_n with exponent $\varepsilon \neq 0$. This equation induces equation of the form $x^{\varepsilon}\bar{g} = 1$ over the free abelian group $A_n = M_n/M_n'$. Here, $\bar{g}$ is the image of $g = g_0 g_1 \ldots g_{n-1}$ in A_n.

Suppose that $\varepsilon = 1$, i. e., the equation (7) is unimodular. In this case, we set $x = g^{-1}y$, where y is a new variable. The new induced equation over A_n is expressed as $y = 1$. Then (7) is equivalent to an equation of the form

$$g_0 \cdot (g^{-1}y)^{\varepsilon_1} \cdot g_1 \cdot \ldots \cdot g_{n-1} \cdot (g^{-1}y)^{\varepsilon_n} = 1. \tag{8}$$

The equation (8) has a solution y in the group $\tilde{W}$ such that y belongs to the abelian normal subgroup $\tilde{W}_1 = \begin{pmatrix} 1 & \sum\limits_{i=1}^{n} \mathrm{Frac}(\mathbb{Z}[A_n])t_i \\ 0 & 1 \end{pmatrix}$. Indeed, we consider this abelian normal subgroup as a module via conjugations over group ring $\mathbb{Z}[A_n]$ as we did before. Then (8) can be rewritten in the form

$$y^{\bar{u}} = v, \text{ where } \bar{u} \in \mathbb{Z}[A_n],\ v \in M_n'.$$

Since $M_n' \leqslant \tilde{W}_1$ and $\tilde{W}_1$ is a vector space, it will be enough to prove that $\bar{u} \neq 0$. We can take $y = v^{\bar{u}^{-1}}$ and succeed. Since (8) is unimodular, $\bar{v} = \sum\limits_{i=1}^{k+1} f_i - \sum\limits_{j=1}^{k} h_j$ for some $k \in \mathbb{N}$

and $f_i, h_j \in A_n$. Let $\delta : \mathbb{Z}[A_n] \to \mathbb{Z}$ be a trivialization homomorphism that sends every group element $a \in A_n$ to 1. Then $\delta(\bar{v}) = 1$ and $\bar{v} \neq 0$. We proved that every unimodular equation (7) is solvable in metabelian group $\tilde{M}_n$.

Suppose that $\varepsilon > 1$, i. e., the equation (7) is regular, but not unimodular. We deal with M_n as a subgroup of other group $H_n \simeq M_n$ with the basis $\{h_1, \ldots, h_n\}$ via embedding corresponding to map $f_i \mapsto h_i^{\varepsilon}$ for $i = 1, \ldots, n$. This embedding is well-defined. Indeed, a subgroup of a free metabelian group is free if and only if it is generated by a set of elements, which are independent modulo the derived group [14]. Thus, a subgroup of M_n generated by some k independent modulo M_n' elements is isomorphic to M_k with the basis consisting of k given elements. Denote $A_n = M_n/M_n'$ and $B_n = H_n/H_n'$. Then A_n is a free abelian subgroup of B_n, $A_n \simeq B_n$, with the basis $\{b_1, \ldots, b_n\}$ corresponding to $\{h_1, \ldots, h_n\}$. As above, (7) induces an equation of the form $x^{\varepsilon}\bar{g} = 1$ over the free abelian group A_n, or, more generally, over B_n. There is a unique solution $x = g$ of this induced equation. The rest of the proof is practically the same as the proof above in the case of unimodular equation. The group $\tilde{W}$ in this proof is constructed for bigger free metabelian group M_n. There is a small difference in the last step in these two proofs. Now, $\bar{v} = \sum\limits_{i=1}^{k+\varepsilon} f_i - \sum\limits_{j=1}^{k} h_j$. Then $\delta(\bar{v}) = \varepsilon$. But we have $\bar{v} \neq 0$, and succeed again. ■

Remark 1. We see from the proof that in the case of regular and not unimodular equation (7), this equation is solvable in metabelian overgroup isomorphic to $\tilde{W}$. The difference with the unimodular case is in embedding of M_n into $\tilde{W}$. This embedding depends on ε.

We can now give a generalization of Theorem 4 to the class of rigid metabelian groups. Recall that a group G is called *m-rigid* if it contains a finite series of normal subgroups

$$G > G_1 > \ldots > G_{m+1} = 1,$$

with abelian quotients G_i/G_{i+1} (thus G is solvable) such that each quotient G_i/G_{i+1} considered as a module over $\mathbb{Z}[G/G_i]$ via conjugation has no module torsion, i. e., for each nontrivial $g \in G_i/G_{i+1}$ and for every nontrivial $v \in \mathbb{Z}[G/G_i]$, it follows $g^v \neq 1$. Such a series (if it exists) is uniquely defined by the group G that is solvable of the derived length exactly m [15]. A group G is called *rigid* if it is m-rigid for some m. If G is metabelian, then $m = 2$.

In the class of all rigid groups, we distinguish *divisible* groups G with quotients G_i/G_{i+1}, whose elements are divisible by any nontrivial element of the respective group ring $\mathbb{Z}[G/G_i]$, i. e., for each $g \in G_i/G_{i+1}$ and for every nontrivial $v \in \mathbb{Z}[G/G_i]$, there exists $h \in G_i/G_{i+1}$ such that $h^v = g$. It has been shown in [16] that every m-rigid group can be embedded into divisible m-rigid group. In particular, every metabelian rigid group can be embedded into metabelian divisible rigid group. It has been proved in [16] that there exists a minimal divisible m-rigid completion $\hat{G}$ of the given m-rigid group G preserving linear independence, and $\hat{G}$ is uniquely defined up to G-isomorphism.

Theorem 5. Every regular equation over a rigid metabelian group M is solvable in the class $\mathcal{M}$ of all metabelian groups. Moreover, every regular equation over a rigid metabelian group M has a solution in the divisible rigid completion $\hat{M}$.

A proof of Theorem 5 is practically such as the proof of Theorem 4.

REFERENCES

1. *Noskov G. A., Remeslennikov V. N., and Roman'kov V. A.* Infinite groups. J. Sov. Math., 1982, vol. 18, pp. 669–735.
2. *Remeslennikov V. N. and Roman'kov V. A.* Model-theoretic and algorithmic questions in group theory. J. Sov. Math., 1985, vol. 31, pp. 2887–2939.
3. *Kharlampovich O. and Myasnikov A.* Equations and Algorithmic Problems in Groups. Publicacoes Math., IMPA, Brazil, 2008. 79 p.
4. *Roman'kov V.* Equations over groups. Groups, Complexity, Cryptology, 2012, vol. 4, pp. 191–239.
5. *Gerstenhaber V. and Rothaus O. S.* The solutions of sets of equations in groups. Proc. Nat. Acad. Sci.U.S., 1962, vol. 48, pp. 1951–1953.
6. *Rothaus O. S.* On the non-triviality of some group extensions given by generators and relators. Ann. Math., 1977, vol. 106, pp. 599–612.
7. *Klyachko A.* Funny property of sphere and equations over groups. Comm. Algebra, 1993, vol. 21, pp. 2555–2575.
8. *Brodskiĭ S. D.* Equations over groups and groups with a single defining relation. Russian Math. Surveys, 1980, vol. 35, pp. 165.
9. *Brodskiĭ S. D.* Equations over groups and groups with a single defining relator. Siberian Math. J., 1984, vol. 25, pp. 84–103.
10. *Howie J.* On locally indicable groups. Math. Z., 1982, vol. 180, pp. 445–461.
11. *Gersten S. M.* Reducible diagrams and equations over groups. In: Gersten, S. M., ed. Essays in Group Theory. MSRI Publ., Springer Verlag, 1985, vol. 8, pp. 15–73.
12. *Krstić S.* Systems of equations over locally $p-$indicable groups. Invent. Math., 1985, vol. 81, pp. 373–378.
13. *Timoshenko E. I.* Endomorfizmy i universal'nye teorii razreshimykh grupp [Endomorphisms and universal theories of solvable groups]. Novosibirsk, NSTU Publishers, 2011. 327 p. (NSTU Monographs Ser.) (in Russian).
14. *Baumslag G.* Some theorems on the free groups of certain product varieties. J. Combinatorial Theory, 1967, vol. 2, pp. 77–99.
15. *Myasnikov A. and Romanovskiy N. S.* Krull dimension of solvable groups. J. Algebra, 2010, vol. 324, pp. 2814–2831.
16. *Romanovskiy N. S.* Divisible rigid groups. Algebra and Logic, 2008, vol. 47, pp. 426–434.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7

КРИПТОАВТОМАТЫ С ФУНКЦИОНАЛЬНЫМИ КЛЮЧАМИ[1]

Г. П. Агибалов

Национальный исследовательский Томский государственный университет, г. Томск, Россия

Определяется понятие криптографического автомата (называемого также криптоавтоматом) как некоторого класса C автоматных сетей с фиксированной структурой N, построенных с помощью операций последовательного, параллельного и с обратной связью соединений инициальных конечных автоматов с функциями переходов и выходов, принадлежащими произвольным функциональным классам. Ключ криптоавтомата может содержать в себе начальные состояния и функции переходов и выходов некоторых компонент в N так, что задание любого конкретного ключа k влечёт за собой выбор вполне определённой автоматной сети N_k в C в качестве нового криптографического алгоритма. В случае обратимого автомата сети этот алгоритм может выступать в роли алгоритма шифрования. Функционирование сети N_k в дискретном времени описывается канонической системой уравнений конечного автомата, сопоставляемого этой сети. Её структура описывается системой уравнений, объединяющей в себе системы канонических уравнений компонент сети. Криптоанализ криптоавтомата осуществляется путём решения функциональной или структурной системы уравнений сети N_k и доопределения возникающих при этом частичных функций её компонент в заданных классах. В роли одного из инструментов решения автоматных уравнений используется метод DSS, который в применении к некоторой системе уравнений E является итерацией следующей тройки действий: 1) E разделяется (Divided) на две подсистемы E' и E'', где E' легко решаема; 2) E' решается (Solved); 3) решение E' подставляется (Substituted) в E''. В криптографических системах криптоавтоматы находят применение в качестве их компонент — криптографических генераторов, блоков замены, фильтров, комбайнеров, ключевых хеш-функций, а также систем шифрования, симметричных и с открытым ключом, и схем цифровой подписи. Определение и криптоанализ иллюстрируются на примере автономного криптоавтомата, обобщающего известную схему криптографического генератора с альтернативным управлением, или с перемежающимся шагом, построенного на регистрах сдвига с линейной обратной связью. Представлен ряд атак на этот криптоавтомат с ключами различных типов, сочетающих в себе комбинаторные или функциональные свойства, выраженные начальными состояниями или функциями выходов компонент в схеме криптоавтомата.

Ключевые слова: *конечный автомат, автоматная сеть, криптоавтомат, криптоавтомат с альтернативным управлением, криптоанализ, метод DSS, доопределение частичных функций.*

[1]Работа поддержана грантом РФФИ, проект № 17-01-00354.

DOI 10.17223/20710410/36/5

CRYPTAUTOMATA WITH FUNCTIONAL KEYS

G. P. Agibalov

National Research Tomsk State University, Tomsk, Russia

E-mail: agibalov@isc.tsu.ru

In this paper, we describe the cryptautomata and some cryptanalysis techniques for them. In cryptographic systems, the cryptautomata are widely used as its primitives including key-stream generators, s-boxes, cryptofilters, cryptocombiners, key hash functions as well as symmetric and public-key ciphers, digital signature schemes. Here, a cryptautomaton is defined as a class C of automata networks of a fixed structure N constructed by means of the series, parallel, and feedback connection operations over initial finite automata (finite state machines) with transition and output functions taken from some predetermined functional classes. A cryptautomaton key can include initial states, transition and output functions of some components in N. The choosing a certain key k produces a certain network N_k from C to be a cryptographic algorithm. In case of invertibility of N_k, this algorithm can be used for encryption. The operation (functioning) of any network N_k in the discrete time is described by the canonical system of equations of its automaton. The structure of N_k is described by the union of canonical systems of equations of its components. The cryptanalysis problems for a cryptautomaton are considered as the problems of solving the operational or structural system of equations of N_k with the corresponding unknowns that are key k variables and (or) plaintexts (input sequences). For solving such a system E, the method DSS is used. It is the iteration of the following three actions: 1) E is Divided into subsystems E' and E'', where E' is easy solvable; 2) E' is Solved; 3) the solutions of E' are Substituted into E'' by turns. The definition and cryptanalysis of a cryptautomaton are illustrated by giving the example of the autonomous cryptautomaton with the alternative control. It is a generalization of the LFSR-based cryptographic alternating step generator. We present a number of attacks on this cryptautomaton with the states or output functions of its components as a key.

Keywords: *finite automaton, automata network, cryptautomaton, cryptautomaton generator with alternative control, cryptanalysis, linearization attack, "devide-and-solve-and-substitute", partially defined function completion.*

1. Автоматные сети

В современной криптографии важное место занимают криптосистемы и их компоненты, представляющие собой сети, построенные с помощью операций последовательного, параллельного и с обратной связью соединений инициальных конечных автоматов. Первые две операции бинарные. Для любых двух автоматов $A_i = (X_i, S_i, Y_i, g_i, f_i, s_i(1))$ с конечными входными алфавитами X_i, выходными алфавитами Y_i, множествами состояний S_i, начальными состояниями $s_i(1)$ и с функциями переходов и выходов соответственно $g_i : X_i \times S_i \to S_i$ и $f_i : X_i \times S_i \to Y_i$, $i \in \{1, 2\}$, их результатами являются автоматы $A = (X, S, Y, g, f, s(1))$, в которых $S = S_1 \times S_2$, $s(1) = s_1(1)s_2(1)$ и для любых $x \in X$ и $s = s_1 s_2 \in S$:

— для последовательного соединения $A_1 \cdot A_2$ верно $X = X_1$, $Y = Y_2$, $g(x, s) = (g_1(x, s_1), g_2(f_1(x, s_1), s_2))$ и $f(x, s) = f_2(f_1(x, s_1), s_2)$;

— для параллельного соединения $(A_1 \parallel A_2)h$ с отображением связи $h : Y_1 \times Y_2 \to Y$ верно $X = X_1 = X_2$, $g(x,s) = (g_1(x,s_1), g_2(x,s_2))$ и $f(x,s) = h(f_1(x,s_1), f_2(x,s_2))$.

Третья операция унарная: в автомате A сети hA_1, являющейся результатом её применения к автомату A_1 с отображением связи $h : X \times Y_1 \to X_1$, верно $S = S_1$, $Y = Y_1$, $g(x,s) = g_1(h(x, f_1(s)), s)$, $f(x,s) = f_1(s)$ для произвольного X и любых $x \in X$ и $s \in S$.

В определении последнего соединения предполагается, что A_1 есть автомат Мура, в нём функция выходов зависит только от состояния.

Формально автоматная сеть и её компоненты определяются индуктивно следующим образом:

1. Всякий конечный автомат является автоматной сетью и одновременно её единственной компонентой.

2. Последовательное, параллельное и с обратной связью соединения автоматных сетей есть автоматная сеть. Её компонентами являются компоненты этих сетей.

3. Других автоматных сетей нет.

По определению, всякая автоматная сеть однозначно определяется своими компонентами и отображениями связи между ними.

2. Канонические уравнения

Функционально любую автоматную сеть можно описать системой уравнений, называемой её канонической системой уравнений (КСУ) и определяемой по индукции построения сети следующим образом.

1. Каноническая система уравнений автомата $A = (X, S, Y, g, f, s(1))$ есть система уравнений от переменных $x(t), s(t), y(t)$, $t = 1, 2, \ldots$, со значениями в X, S, Y соответственно, записываемая как

$$\begin{aligned} &y(t) = f(x(t), s(t)),\\ &s(t+1) = g(x(t), s(t)),\ t \geqslant 1,\\ &s(1) \text{ — начальное условие.} \end{aligned}$$

Далее эта система обозначается КСУ$(A; x, s, y, s(1))$.

2. Каноническая система уравнений сети $A_1 \cdot A_2$ записывается в виде

$$\text{КСУ}(A_1; x, s_1, y_1, s_1(1)),\ \text{КСУ}(A_2; y_1, s_2, y, s_2(1))$$

и обозначается КСУ$(A_1 \cdot A_2; x, s_1, s_2, y, s_1(1)s_2(1)$.

3. Каноническая система уравнений сети $(A_1 \parallel A_2)h$ записывается в виде

$$\text{КСУ}(A_1; x, s_1, y_1, s_1(1)),\ \text{КСУ}(A_2; x, s_2, y_2, s_2(1)),\ y(t) = h(y_1(t), y_2(t)),\ t \geqslant 1,$$

и обозначается КСУ$((A_1 \parallel A_2)h; x, s_1, s_2, y, s_1(1)s_2(1))$.

4. Каноническая система уравнений сети hA_1 записывается в виде

$$\text{КСУ}(A_1; x_1, s_1, y_1, s_1(1)),\ x_1(t) = h(x(t), y_1(t)),\ y(t) = y_1(t),\ t \geqslant 1,$$

и обозначается КСУ$(hA_1; x, s_1, y, s_1(1))$.

По определению, переменные в КСУ любой автоматной сети N подразделяются на входную — x, выходную — y, внутренние — s_i (они же переменные состояний компонент в N) и вспомогательные — x_j, y_k (посредством их осуществляется связь между компонентами в N). Внутренние и вспомогательные переменные называются иногда промежуточными. Переменная t трактуется как дискретное время, а $u(t)$ — как значение любой другой переменной u в момент времени t. Значение переменной состояния некоторой компоненты сети в момент $t = 1$ называется начальным состоянием этой компоненты.

3. Определение криптоавтомата

Понятие криптоавтомата восходит к работе [1], где под ним подразумевается конечный автомат с ключом и приводятся примеры описания такими криптоавтоматами генераторов ключевого потока MUGI и KNOT — в поточных шифрах, блоков замены L, M, R, S и управляющего ими блока stepping — в японской шифрмашине Purple и симметричного конечно-автоматного шифра Закревского. Здесь мы расширяем это понятие, подразумевая под криптоавтоматом класс автоматных сетей с ключом, который может включать в себя и начальные состояния компонент сети, и их функции переходов и выходов так, что любое фиксирование значения ключа выделяет в классе некоторую конкретную сеть для выполнения соответствующего криптографического преобразования.

Условимся далее множество всех функций, имеющих одинаковые области соответственно определения и значений и обладающих некоторыми фиксированными свойствами, называть (функциональным) классом. Так, можно говорить, например, о классах булевых функций, зависящих от одних и тех же множеств переменных и обладающих ограниченной сложностью задания или вычисления и одинаковыми криптографическими свойствами — нелинейностью, корреляционной иммунностью и т. п. Аналогично, можно говорить и об автоматных классах, или классах автоматов с функциями переходов и выходов из некоторых фиксированных функциональных классов. Наконец, можно говорить и о классах автоматных сетей, в которых между компонентами разных сетей существует взаимно однозначное соответствие, такое, что соответствующие компоненты принадлежат одному и тому же автоматному классу. Таким образом, автоматные сети из одного и того же класса имеют общую структуру (схему соединений компонент) и могут различаться лишь начальными состояниями компонент и их функциями переходов и выходов в своих классах. Принимая за ключ конкретные значения этих величин (начальных состояний и (или) функций переходов и выходов) в каких-либо компонентах автоматных сетей класса, мы тем самым выделяем в этом классе конкретную автоматную сеть, подобно тому как, фиксируя значение ключа в криптосистеме, мы превращаем её в конкретный криптоалгоритм. Эти рассуждения приводят нас к следующим определениям.

Определим *класс автоматной сети* N как множество $C(N)$ всех автоматных сетей, которые получаются из N с помощью операций замены в N начальных состояний некоторых компонент, и (или) функций переходов некоторых компонент, и (или) функций выходов некоторых компонент с сохранением их функциональных классов. Ясно, что этими операциями можно любую сеть в $C(N)$ получить из любой другой сети в $C(N)$, т. е. для любой сети N' в $C(N)$ имеет место равенство $C(N') = C(N)$, в связи с чем можно говорить о *классе автоматных сетей* как о любом таком их множестве C, в котором существует сеть N со свойством $C(N) = C$, или, что то же самое, для всех N в C верно это свойство. В дальнейшем класс автоматных сетей понимается именно в этом смысле.

Тройка $\Sigma = (C, I, K)$ называется *криптографическим автоматом*, или сокращённо *криптоавтоматом*, если C есть некоторый класс автоматных сетей, $C = C(N)$ для некоторой сети N, $I = \{I_\mathrm{s}, I_\mathrm{t}, I_\mathrm{o}\}$, где I_s, I_t и I_o суть множества, некоторые из которых непустые и состоят из номеров компонент в N, чьи начальные состояния (s — state), функции переходов (t — transition) и функции выходов (o — output) соответственно составляют ключ криптоавтомата, и K — множество всех возможных значений этого ключа. В нём C называется *сетевым классом*, I — *носителем ключа* и K — *ключевым пространством* криптоавтомата. Так, пусть N состоит из компонент $A_1, A_2, \ldots, A_r$, где

A_i для каждого $i \in \{1, 2, \ldots, r\}$ есть автомат с множеством состояний S_i и с функциями переходов и выходов из некоторых функциональных классов G_i и F_i соответственно. Тогда $I_\mathrm{s}, I_\mathrm{t}, I_\mathrm{o}$ суть подмножества в $\{1, 2, \ldots, r\}$, такие, что для любого $i \in \{1, 2, \ldots, r\}$ если $i \in I_\mathrm{s}$, $i \in I_\mathrm{t}$ или $i \in I_\mathrm{o}$, то соответственно для каждого $s \in S_i$, $g \in G_i$ или $f \in F_i$ найдётся сеть в C, в i-й компоненте которой s является начальным состоянием, g — функцией переходов или f — функцией выходов соответственно, а если $i \notin I_\mathrm{s}$, $i \notin I_\mathrm{t}$ или $i \notin I_\mathrm{o}$, то соответственно начальные состояния, функции переходов или функции выходов i-й компоненты во всех сетях в C одинаковы. Кроме того, ключевое пространство криптоавтомата Σ равно $K = K_\mathrm{s} \times K_\mathrm{t} \times K_\mathrm{o}$, где $K_\mathrm{s} = \prod\limits_{i \in I_\mathrm{s}} S_i$, $K_\mathrm{t} = \prod\limits_{i \in I_\mathrm{t}} G_i$, $K_\mathrm{o} = \prod\limits_{i \in I_\mathrm{o}} F_i$ и $\prod$ — символ декартова произведения.

Каждый ключ $k = k_\mathrm{s} k_\mathrm{t} k_\mathrm{o}$ в ключевом пространстве K криптоавтомата Σ состоит из трёх частей, не все из которых пустые: k_s — набор значений начальных состояний компонент криптоавтомата с номерами в I_s, k_t — набор функций переходов компонент криптоавтомата с номерами в I_t и k_o — набор функций выходов компонент криптоавтомата с номерами в I_o. Его задание выделяет в C однозначно некоторую конкретную сеть N_k, в которой начальные состояния компонент A_i для $i \in I_\mathrm{s}$ принадлежат набору k_s, функции переходов компонент A_i, $i \in I_\mathrm{t}$, — набору k_t и функции выходов компонент A_i, $i \in I_\mathrm{o}$, — набору k_o.

Мы допускаем к рассмотрению и автономные автоматы (в них функции переходов и выходов зависят только от состояния), и комбинационные автоматы (в них функция выходов не зависит от состояний и нет нужды ни в множестве последних, ни в функции переходов), и обратимые автоматы (в них входная последовательность однозначно определяется по выходной и, возможно, по начальному состоянию). По определению, каждая автоматная сеть является конечным автоматом, поэтому все понятия, введённые для автоматов, переносятся на автоматные сети и на криптоавтоматы, построенные на основе их классов. Так, в случае комбинационности, автономности или обратимости автомата сети сама сеть также называется комбинационной, автономной или обратимой соответственно. Можно говорить, следовательно, и о криптоавтоматах такого рода.

4. Примеры криптоавтоматов

Кроме приведённых выше примеров однокомпонентных криптоавтоматов, под определение криптоавтомата на основе класса автоматных сетей подпадают и роторные машины, включая Энигму, и уже упоминавшаяся пурпурная шифрсистема Purple, и конечно-автоматные криптосистемы с открытым ключом для шифрования и цифровой подписи семейства FAPKC [1, 2].

Комбинационные криптоавтоматы обычно состоят из комбинационных компонент и в таком составе часто могут применяться в роли блоков замены в тех блочных шифрах с аддитивными раундовыми ключами, которые являются листьями «дерева», выросшего из корня DES. Более того, сами шифры на этом дереве допускают описание комбинационными криптоавтоматами. Чтобы это понять, достаточно заметить, что сложение бит раундового ключа со значениями на входе блока замены превращает функции последнего в функции, зависящие ещё и от ключа шифра.

По определению, каждая автономная сеть является последовательным соединением автоматных сетей, в котором на месте первой компоненты (автомата A_1) выступает автономный автомат. Автономные сети порождают последовательности выходных символов, и основанные на них криптоавтоматы мы называем конечно-автоматными

криптографическими генераторами. Огромное множество примеров таких криптоавтоматов доставляют конечно-автоматные обобщения криптографических генераторов, построенных в виде схем из регистров сдвига с линейными обратными связями (LFSRs). Одно такое обобщение рассмотрено в [3, 4]. Далее, в п. 8, мы рассмотрим криптоавтоматное обобщение ещё одного генератора на LFSRs — криптографического генератора с альтернативным управлением [5], называемого в [6] the alternating step generator и в [7] — генератором с перемежающимся шагом.

5. Задачи криптоанализа

Предполагается, что криптоаналитику, исследующему криптоавтомат (C, I, K), известны все составляющие его множества C, I и K. Для автономного криптоавтомата $\Sigma = (C, I, K)$ есть только одна задача криптоанализа: по кратчайшему отрезку $\gamma = y(1)y(2)\dots y(l)$, $l \geqslant 1$, последовательности на выходе сети N_k в C узнать его ключ $k \in K$. Задача фактически распадается на три других: 1) сформулировать необходимые и достаточные условия единственности автоматной сети в C, порождающей γ; 2) разработать метод, позволяющий по любому такому отрезку γ построить каждую из сетей в C, которая может сгенерировать эту γ; 3) по любой сети N_k в C, порождающей γ, вычислить ключ k. Умея это делать, можно сформулированную задачу криптоанализа автономного криптоавтомата $\Sigma = (C, I, K)$ решать следующим образом: последовательность на выходе неизвестной сети N_k в C наблюдается до тех пор, пока не будет получен отрезок γ, для которого выполнены условия единственности в задаче 1, после чего методом в задаче 2 строится искомая сеть, а потом методом задачи 3 по этой сети вычисляется искомый ключ.

Для неавтономного криптоавтомата $\Sigma = (C, I, K)$ можно указать по меньшей мере две задачи криптоанализа: полное раскрытие — по заданным отрезкам $\alpha = x(1)x(2)\dots x(l)$ и $\gamma = y(1)y(2)\dots y(l)$, $l \geqslant 1$, соответствующих входной и выходной последовательностей неизвестной сети N_k в C вычислить ключ k, и раскрытие сообщения — по отрезку γ на выходе сети N_k вычислить отрезок α на её входе, который она преобразует в γ. Оптимизационный вариант каждой из этих задач формулируется аналогично задаче криптоанализа автономного криптоавтомата.

Раскрытие ключа в любом случае возможно тривиальным методом — так называемой атакой грубой силы (от англ. Brute-Force Attack (BFA)), состоящей в опробовании каждого ключа в K на соответствие определяемой им сети в C исходным данным: выходной последовательности — в автономном случае или выходной и входной последовательностей — в неавтономном случае. Вычислительная сложность этого метода, естественно, оценивается мощностью ключевого пространства криптоавтомата.

6. Доопределение частичной функции в заданном классе

Для произвольного класса F функций $f : V^n \to V$ над конечным множеством (алфавитом) V и для произвольной частичной функции $f' : D_{f'} \subset V^n \to V$ мы говорим, что f' *доопределима в классе* F, если существует функция $f \in F$, такая, что $f'(v) = f(v)$ для всех $v \in D_{f'}$; в этом случае говорят, что f' *доопределима до* f.

В криптоанализе криптографических систем с функциональными ключами, в том числе и криптоавтоматов, возникает проблема доопределения некоторых частичных функций до функций в заданном классе F, состоящая в выяснении существования и единственности такого доопределения и в построении всевозможных функций в классе, до которых доопределима заданная частичная функция. Решение этой проблемы для любого класса F возможно тривиальным методом — так называемым исчерпывающим

поиском (от англ. Exhaustive Search Method (ESM)), который заключается в проверке для каждой функции в F, доопределима ли до неё заданная частичная функция. Вычислительная сложность этого метода, естественно, оценивается мощностью класса F. Других общих методов решения данной проблемы (для всех классов F) неизвестно. В [8, 9] можно найти её нетривиальное решение для класса F булевых функций от n переменных, в котором булевы функции зависят существенно от ограниченного числа $k < n$ последних.

Здесь мы продемонстрируем эту проблему в криптоанализе уже упоминавшегося криптоавтоматного генератора с альтернативным управлением. Основным инструментом в этом криптоанализе служит метод DSS решения систем автоматных уравнений.

7. Метод DSS

7.1. Определение

Подмножество L переменных в некоторой системе уравнений E над конечным алфавитом называется *эффективным множеством*, если фиксирование любых возможных значений этих переменных превращает E в легко решаемую (например, за полиномиальное или меньшее время) систему (ЛРС). В частности, таковым является подмножество переменных, при любом фиксировании которых все уравнения в системе над конечным полем превращаются в линейные. Оно называется линеаризационным множеством переменных системы [5, 10].

Система уравнений E над k-элементным алфавитом с q-элементным эффективным множеством переменных L решается со сложностью k^q методом DS (Devide and Solve), или по-русски «разделяй и решай», состоящим в подстановке в систему E поочерёдно различных наборов значений переменных в L и в решении получаемой каждый раз ЛРС. В случае совместности последней её решение, взятое вместе с подставленным в E набором значений переменных в L, является решением системы E. Метод DS на основе линеаризационного множества переменных со значениями в конечном поле называется линеаризационной атакой [5, 10]. В нём решается частный случай ЛРС — система линейных уравнений.

Система уравнений E называется *рекурсивно легко решаемой системой* (РЛРС), если в ней существует непустая подсистема уравнений E' с небольшим эффективным подмножеством (ныне это не более 3–4 десятков) переменных, такая, что подсистема уравнений $E \setminus E'$ подстановкой в неё любого решения подсистемы E' преобразуется в РЛРС. По определению, такая система решается кратным применением метода DS к её подсистеме и подстановки полученных решений подсистемы в её дополнение. В [3] данный метод решения РЛРС назван DSS — Devide, Solve and Substitute («разделяй, решай и подставляй»).

Можно сказать, что метод DS является частным случаем метода DSS, а именно методом DSS с однократным применением метода DS. Таким образом, и линеаризационная атака на систему уравнений над конечным полем — это тоже частный случай метода DSS.

Далее мы продемонстрируем этот метод применительно к каноническим системам уравнений конечных автоматов и автоматных сетей над полем $\mathbb{F}_2$ из двух элементов. Переложение излагаемого материала на автоматные системы уравнений над произвольным алфавитом V возможно простой заменой $\mathbb{F}_2$ на V, которая не меняет существа метода и лишь увеличивает его вычислительную сложность.

7.2. Решение канонической системы уравнений автомата

Покажем, как методом DSS может быть решена каноническая система уравнений E произвольного конечного автомата $A = (\mathbb{F}_2, \mathbb{F}_2^m, \mathbb{F}_2, g, f, s(1))$ над $\mathbb{F}_2$ при известных функциях g и f, заданных начальном состоянии $s(1) \in \mathbb{F}_2^m$ и выходной последовательности $y(1)y(2)\dots y(l) \in \mathbb{F}_2^l$ и неизвестной входной последовательности $x(1)x(2)\dots x(l) \in \mathbb{F}_2^l$:

$$y(t) = f(x(t), s(t)),\ s(t+1) = g(x(t), s(t)),\ 1 \leqslant t \leqslant l.$$

В самом деле, при $t = 1$ имеем уравнение $y(1) = f(x(1), s(1))$, которое для $x(1)$ имеет либо два решения — 0 и 1, если $y(1) = f(0, s(1)) = f(1, s(1))$, либо одно решение — некоторое $b(1) \in \mathbb{F}_2$, если $y(1) = f(b(1), s(1)) \neq f(\neg b(1), s(1))$, либо ни одного решения, если $y(1) \neq f(0, s(1)) = f(1, s(1))$. В последнем случае система E несовместна. При $2 \leqslant t \leqslant l$ для каждого из найденных решений для $x(1)x(2)\dots x(t-1)$ вычисляем $s(t) = g(x(t-1), s(t-1))$ и получаем уравнение $y(t) = f(x(t), s(t))$, которое тоже может иметь одно ($b(t)$), два (0 и 1) или ни одного решения для $x(t)$. В последнем случае рассматриваемый вариант частичного решения $x(1)x(2)\dots x(t-1)$ отвергается, а в остальных случаях из него получаются соответственно один или два варианта расширенного частичного решения $x(1)x(2)\dots x(t-1)x(t)$ — соответственно с $x(t) = b(t)$ или с $x(t) = 0$ и с $x(t) = 1$. Если на некотором шаге с номером $t \leqslant l$ множество полученных частичных решений оказывается пустым, система E объявляется несовместной; в противном случае все её решения для $x(1)x(2)\dots x(l)$ находятся на l-м шаге. Алгоритмически метод представляет собой обход дерева решений высоты не более l со степенями исхода вершин не более 2. В нём на путях от корня к листьям, имеющих длину l, лежат все решения КСУ автомата — все его входные последовательности $x(1)x(2)\dots x(l)$, преобразуемые им в выходную последовательность $y(1)y(2)\dots y(l)$.

В случае, если A есть автомат Мура (в нём функция выходов зависит от входного символа фиктивно), то уравнение $y(t) = f(x(t), s(t))$ в его канонической системе имеет для $x(t)$ два решения (0 и 1) или ни одного, вследствие чего в её дереве решений из каждой неконцевой вершины исходят ровно две дуги.

Заметим также, что если A является автоматом некоторой автоматной сети, то метод DSS по выходной последовательности этой сети находит её возможные входные последовательности. Кроме того, если A есть некоторая компонента автоматной сети над $\mathbb{F}_2$, то метод DSS вычисляет входные последовательности этой компоненты по её выходной последовательности в сети. Далее мы покажем, как эти задачи решаются методом DSS непосредственно по канонической системе уравнений автоматной сети без построения её автомата. Ввиду индуктивности определения автоматных сетей достаточно показать это для базовых операций их построения — последовательного, параллельного и с обратной связью соединений автоматов над $\mathbb{F}_2$.

7.3. Решение канонической системы уравнений последовательной автоматной сети

Начнём с рассмотрения канонической системы уравнений последовательного соединения автоматов $A_1 \cdot A_2$ над F_2: КСУ$(A_1; x, s_1, y_1, s_1(1))$, КСУ$(A_2; y_1, s_2, y, s_2(1))$. В ней подсистема

$$y(t) = f_2(y_1(t), s_2(t)),\ s_2(t+1) = g_2(x_2(t), s_2(t)),\ 1 \leqslant t \leqslant l,$$

с начальным условием $s_2(1)$ является канонической системой уравнений автомата A_2 и может быть решена методом DSS, как только что описано, при известных его функ-

циях g_2 и f_2 и заданных его начальном состоянии $s_2(1) \in \mathbb{F}_2^{m_2}$ и выходной последовательности $y(1)y(2)\dots y(l) \in \mathbb{F}_2^l$. В результате будут найдены в $\mathbb{F}_2^l$ возможные входные последовательности $y_1(1)y_1(2)\dots y_1(l)$ автомата A_2, среди которых присутствуют и все те выходные последовательности автомата A_1 с неизвестными атрибутами $s_1(1), g_1, f_1$, которые он может выработать в ответ на входные последовательности сети, вызывающие на её выходе последовательность $y(1)y(2)\dots y(l)$.

При известных $s_1(1), g_1, f_1$ аналогичным образом по выходной последовательности $y_1(1)y_1(2)\dots y_1(l)$ автомата A_1 вычисляются его входные последовательности $x_1(1)x_1(2)\dots x_1(l)$, являющиеся одновременно входными последовательностями сети.

7.4. Решение канонической системы уравнений параллельной автоматой сети

Рассмотрим теперь каноническую систему уравнений $E = \text{КСУ}((A_1 \parallel A_2)h; x, s_1, s_2, y, s_1(1)s_2(1))$ параллельного соединения автоматов $(A_1 \parallel A_2)h$ над $\mathbb{F}_2$ с отображением $h : \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2$

$$\text{КСУ}(A_1; x, s_1, y_1, s_1(1)),\ \text{КСУ}(A_2; x, s_2, y_2, s_2(1)),\ y(t) = h(y_1(t), y_2(t)),\ 1 \leqslant t \leqslant l.$$

Пусть далее $M_h(t) = \{(y_1^1(t), y_2^1(t)), (y_1^2(t), y_2^2(t)), \dots, (y_1^{n_t}(t), y_2^{n_t}(t))\} = h^{-1}(y(t)) = \{(y_1(t), y_2(t)) \in \mathbb{F}_2^2 : h(y_1(t), y_2(t)) = y(t)\}$, $t \geqslant 1$. Здесь $n_t = |M_h(t)| \leqslant 4$. После подстановки в систему E вместо пар переменных $(y_1(t), y_2(t))$ их возможных значений $(y_1^j(t), y_2^j(t))$ для $j \in \{1, 2, \dots, n_t\}$ из $M_h(t)$ эта система принимает вид E^j:

$$\begin{aligned}
&y(t) = h(y_1^j(t), y_2^j(t)),\\
&y_1^j(t) = f_1(x(t), s_1(t)),\\
&y_2^j(t) = f_2(x(t), s_2(t)),\\
&s_1(t+1) = g_1(x(t), s_1(t)),\\
&s_2(t+1) = g_2(x(t), s_2(t)),\ 1 \leqslant t \leqslant l,\\
&s_1(1), s_2(1) \text{ — начальные условия.}
\end{aligned}$$

Последовательно для t со значениями 1, 2, …, l из второго и третьего уравнений находим для неизвестного $x(t)$ либо два решения — 0 и 1 (говорим, что имеет место случай 2), если

$$y_1^j(t) = f_1(0, s_1(t)) = f_1(1, s_1(t)) \text{ и } y_2^j(t) = f_2(0, s_2(t)) = f_2(1, s_2(t)),$$

либо одно решение — 0 или 1 (случай 1), если

$$(y_1^j(t) = f_1(0, s_1(t))) \wedge (y_2^j(t) = f_2(0, s_2(t))) \wedge [(y_1^j(t) \neq f_1(1, s_1(t))) \vee (y_2^j(t) \neq f_2(1, s_2(t)))] \vee$$
$$\vee (y_1^j(t) = f_1(1, s_1(t))) \wedge (y_2^j(t) = f_2(1, s_2(t))) \wedge [(y_1^j(t) \neq f_1(0, s_1(t))) \vee (y_2^j(t) \neq f_2(0, s_2(t)))],$$

либо ни одного решения (случай 0), если

$$[(y_1^j(t) \neq f_1(0, s_1(t))) \vee (y_2^j(t) \neq f_2(0, s_2(t)))] \wedge [(y_1^j(t) \neq f_1(1, s_1(t))) \vee (y_2^j(t) \neq f_2(1, s_2(t)))].$$

Найденное так множество решений для $x(t)$ в E^j обозначим $B_j(t)$ и положим $B(t) = B_1(t) \cup \dots \cup B_{n_t}(t)$. По построению $B(t) \subseteq \mathbb{F}_2$. В случае $B(t) = \varnothing$ система E несовместна. В противном случае для каждого из найденных решений $x(t)$ в $B(t)$ вычисляем $s_1(t+1) = g_1(x(t), s_1(t))$ и $s_2(t+1) = g_2(x(t), s_1(t))$ и аналогичным образом находим множество $B(t+1)$ решений в $\mathbb{F}_2$ для $x(t+1)$. После l таких шагов, т. е.

при $t = l$, получаем множество $\{x(1)x(2)\dots x(l) : x(t) \in B(t), t \in \{1, 2, \dots, l\}\}$ всех решений системы уравнений E параллельного соединения автоматов. Его элементы суть всевозможные входные последовательности сети $(A_1 \parallel A_2)h$, которые она с фиксированными компонентами и их начальными состояниями преобразует в выходную последовательность $y(1)y(2)\dots y(l)$.

7.5. Решение канонической системы уравнений автоматной сети с обратной связью

Рассмотрим наконец каноническую систему уравнений $E = \text{КСУ}(hA_1; x, s_1, y, s_1(1))$ автоматного соединения hA_1 над $\mathbb{F}_2$ с обратной связью $h : \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2$

$$\text{КСУ}(A_1; x_1, s_1, y_1),\ x_1(t) = h(x(t), y_1(t)),\ y(t) = y_1(t),\ 1 \leqslant t \leqslant l.$$

В ней подсистема уравнений компоненты A_1 с $1 \leqslant t \leqslant l$ решается методом DSS, как каноническая система уравнений автомата Мура (см. п. 7.2) при заданных $s_1(1), g_1, f_1$ и выходной последовательности сети $y(1)y(2)\dots y(l)$. Результатом решения являются входные последовательности $x_1(1)x_1(2)\dots x_1(l)$ автомата A_1. По каждой из них и выходной последовательности сети методом DSS решается система уравнений

$$x_1(t) = h(x(t), y(t)),\ 1 \leqslant t \leqslant l,$$

и находятся входные последовательности сети $x(1)x(2)\dots x(l)$, а именно: для каждого целого j, $1 \leqslant j < l$, и для каждого решения $x(1)x(2)\dots x(j-1)$ её подсистемы с $1 \leqslant t \leqslant j-1$ её подсистема с $1 \leqslant t \leqslant j$ имеет либо два решения $x(1)x(2)\dots x(j-1)x(j)$ с $x(j) \in \mathbb{F}_2$, если $x_1(j) = h(0, y(j)) = h(1, y(j))$, либо одно решение с $x(j) = b$, $b \in \mathbb{F}_2$, если $x_1(j) = h(b, y(j)) \neq h(\neg b, y(j))$, либо ни одного решения, если $x_1(j) \neq h(0, y(j)) = h(1, y(j))$.

8. Криптогенератор с альтернативным управлением

8.1. Определение

Рассмотрим *автоматные сети с альтернативным управлением*, представляющие собой последовательно-параллельные соединения трёх автоматов над полем $\mathbb{F}_2$. В каждой такой сети $N = A_1 \cdot (A_2 \parallel A_3)h$ один из автоматов, A_1, управляющий, он является автономным: $A_1 = (\mathbb{F}_2^{m_1}, \mathbb{F}_2, g_1, f_1, s_1(1))$, два других, A_2 и A_3, управляемые, они неавтономные: $A_i = (\mathbb{F}_2, \mathbb{F}_2^{m_i}, \mathbb{F}_2, g_i, f_i, s_i(1))$, $i \in \{2, 3\}$. В ней выход автомата A_1 соединён со входами автоматов A_2 и A_3 непосредственно, а выходы A_2 и A_3 соединены с выходом сети через отображение связи $h : \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2$, такое, что для любых y_2, y_3 в $\mathbb{F}_2$ имеет место $h(y_2, y_3) = y_2 \oplus y_3$, где $\oplus$ — операция сложения по mod 2. Альтернативность управления автоматами A_2 и A_3 со стороны автомата A_1 в N выражается в следующих ограничениях на их функции переходов, справедливых для любого их входного символа x и любых их состояний s_2 и s_3 соответственно и называемых далее *условиями альтернативности*: $g_2(x, s_2) = s_2 \Leftrightarrow g_3(x, s_3) \neq s_3$.

Автомат сети $N = A_1 \cdot (A_2 \parallel A_3)h$ с альтернативным управлением есть $A = (\mathbb{F}_2^{m_1} \times \times \mathbb{F}_2^{m_2} \times \mathbb{F}_2^{m_3}, \mathbb{F}_2, g, f, s_1(1)s_2(1)s_3(1))$, и в нём для любых $s_i \in \mathbb{F}_2^{m_i}$, $i \in \{1, 2, 3\}$, верно: $g(s_1s_2s_3) = (g_1(s_1), g_2(f_1(s_1), s_2), g_3(f_1(s_1), s_3))$ и $f(s_1s_2s_3) = f_2(f_1(s_1), s_2) \oplus f_3(f_1(s_1), s_3)$.

Каноническая система уравнений E этой сети записывается как

$$\begin{aligned}
&y_1(t) = f_1(s_1(t)),\\
&s_1(t+1) = g_1(s_1(t)),\\
&y_2(t) = f_2(y_1(t), s_2(t)),\\
&s_2(t+1) = g_2(y_1(t), s_2(t)),\\
&y_3(t) = f_3(y_1(t), s_3(t)),\\
&s_3(t+1) = g_3(y_1(t), s_3(t)),\\
&y(t) = y_2(t) \oplus y_3(t),\ t \geqslant 1,\\
&s_1(1)s_2(1)s_3(1) \text{ — начальное условие,}
\end{aligned}$$

а каноническая система уравнений её автомата A — как

$$\begin{aligned}
&y(t) = f_2(f_1(s_1(t)), s_2(t)) \oplus f_3(f_1(s_1(t)), s_3(t)),\\
&s_1(t+1)s_2(t+1)s_3(t+1) = (g_1(s_1(t)), g_2(f_1(s_1(t)), s_2(t)), g_3(f_1(s_1(t)), s_3(t))),\ t \geqslant 1,\\
&s_1(1)s_2(1)s_3(1) \text{ — начальное условие.}
\end{aligned}$$

В системе E подсистемы E_1 из первого и второго уравнений, E_2 из третьего и четвёртого уравнений и E_3 из пятого и шестого уравнений описывают работу автоматов A_1, A_2 и A_3 соответственно, седьмое уравнение описывет выход сети. Подсистема E' из уравнений с номерами от третьего до седьмого включительно и с начальными условиями $s_2(1)$ и $s_3(1)$ описывает работу параллельной подсети $(A_2 \parallel A_3)h$ сети N. Входные последовательности этой подсети можно вычислить по выходной последовательности сети применением к E' метода DSS, как это описано в п. 7.4 со следующими уточнениями, связанными со специфичностью отображения связи h и свойством альтернативного функционирования автоматов A_2 и A_3 в ней:

1) множество $M_h(t)$, равное $h^{-1}(y(t))$, состоит из двух пар в $\mathbb{F}_2^2$ — 00 и 11, если $y(t) = 0$, либо 01 и 10, если $y(t) = 1$;
2) система уравнений E несовместна не только когда $B(t) = \varnothing$, но и когда

$$(s_2(t+1) \neq s_2(t)) \wedge (s_3(t+1) \neq s_3(t)) \vee (s_2(t+1) = s_2(t)) \wedge (s_3(t+1) = s_3(t))$$

для некоторого t, $1 \leqslant t \leqslant l$.

Криптоавтомат $\Sigma = (C, I, K)$ называется *криптогенератором с альтернативным управлением*, если в нём $C = C(N)$ для некоторой автоматной сети N с альтернативным управлением. По определению, всякая сеть в C является автоматной сетью $A_1 \cdot (A_2 \parallel A_3)h$ с альтернативным управлением, в ней для каждого $i \in \{1, 2, 3\}$ состояния в i-й компоненте принадлежат множеству $S_i = \mathbb{F}_2^{m_i}$, а функции переходов и выходов — некоторым функциональным классам G_i и F_i соответственно. Элементы I_s, I_t и I_o носителя I ключа в Σ являются подмножествами в $\{1, 2, 3\}$, а декартовы множители K_s, K_t и K_o ключевого пространства K — декартовыми произведениями множеств S_i для $i \in I_\mathrm{s}$, G_i для $i \in I_\mathrm{t}$ и F_i для $i \in I_\mathrm{o}$ соответственно. Наконец, в ключе $k = k_\mathrm{s}k_\mathrm{t}k_\mathrm{o} \in K$ имеем: $k_\mathrm{s} \in K_\mathrm{s}$, $k_\mathrm{t} \in K_\mathrm{t}$ и $k_\mathrm{o} \in K_\mathrm{o}$.

8.2. Криптоанализ

Пусть далее $\Sigma = (C, I, K)$ есть произвольный криптогенератор с альтернативным управлением. Предполагается, что целью криптоанализа генератора является нахождение ключа по заданной его выходной последовательности $\gamma = y(1)y(2)\ldots y(l), l \geqslant 1$.

Допуская, разумеется, существование иных методов решения этой задачи, мы остановимся здесь, главным образом, на применениях метода DSS и задачи доопределения частичных булевых функций в атаках на генератор Σ с тем или иным носителем $I = \{I_{\rm s}, I_{\rm t}, I_{\rm o}\}$ ключевого пространства $K = K_{\rm s} \times K_{\rm t} \times K_{\rm o}$. Начнём с простейшего случая.

1. $I_{\rm s} = \{1\}$, $I_{\rm t} = I_{\rm o} = \varnothing$; $K_{\rm s} = S_1 = \mathbb{F}_2^{m_1}$, $K_{\rm t} = K_{\rm o} = \varnothing$; $K = K_{\rm s} = \mathbb{F}_2^{m_1}$; $k = s_1(1) \in K$.

А т а к а 1: 1) по заданной γ на выходе генератора методом DSS вычисляем входные последовательности параллельной подсети $N' = (A_2 \parallel A_3)h$ сети N, являющиеся одновременно выходными последовательностями автомата A_1; 2) для любой из таких последовательностей атакой грубой силы (BFA) находим начальное состояние $s_1(1)$ автомата A_1.

Вычислительная сложность атаки равна 2^{m_1}.

Примечание: если вычисление на шаге 2 выполнить для каждой последовательности, вычисленной на шаге 1, то будут получены все значения ключа k, при которых генератор Σ вырабатывает данную γ.

2. $I_{\rm s} = \{1, 2\}$, $I_{\rm t} = I_{\rm o} = \varnothing$; $K_{\rm s} = S_1 \times S_2 = \mathbb{F}_2^{m_1} \times \mathbb{F}_2^{m_2}$, $K_{\rm t} = K_{\rm o} = \varnothing$; $K = K_{\rm s} = \mathbb{F}_2^{m_1} \times \mathbb{F}_2^{m_2}$; $k = s_1(1)s_2(1) \in K$.

В этом случае атака относится к разряду meet-in-the-middle — встреча посредине. Предварительно, до атаки, для каждого значения a переменной $s_1(1)$ в S_1 вычисляются $s_1(t+1) = g_1(s_1(t))$ и $y_1(t) = f_1(s_1(t))$ для $t \in \{1, 2, \ldots, l\}$ и $s_1(1) = a$, и это a сохраняется по адресу $H(y_1(1)y_1(2)\ldots y_1(l))$, где $H : \mathbb{F}_2^l \to \mathbb{F}_2^{m_1}$ есть некоторая хеш-функция.

А т а к а 2: имея на выходе генератора последовательность γ, методом DSS вычисляем входные последовательности подсети N' при разных значениях переменной $s_2(1)$, выбираемых в S_2 до тех пор, пока при некотором её значении b на входе N' не будет получена некоторая последовательность β с некоторым непустым содержимым a по адресу $H(\beta)$, и тогда пару (a, b) принимаем за результат атаки — значение ключа k.

Вычислительная сложность атаки равна 2^{m_2}.

Примечание: атака остаётся в силе, если в ней автоматы A_2 и A_3 переставить местами.

3. $I_s = \{1, 2, 3\}$, $I_{\rm t} = I_{\rm o} = \varnothing$; $K_{\rm s} = S_1 \times S_2 \times S_3 = \mathbb{F}_2^{m_1} \times \mathbb{F}_2^{m_2} \times \mathbb{F}_2^{m_3}$, $K_{\rm t} = K_{\rm o} = \varnothing$; $K = K_{\rm s} = \mathbb{F}_2^{m_1} \times \mathbb{F}_2^{m_2} \times \mathbb{F}_2^{m_3}$; $k = s_1(1)s_2(1)s_3(1) \in K$, и переменные $y_1(1), y_1(2), \ldots, y_1(l)$ образуют линеаризационное множество системы уравнений E' подсети N' сети N.

А т а к а 3: для каждого $s_1(1)$ в S_1 1) вычисляем $s_1(t+1) = g_1(s_1(t))$ и $y_1(t) = f_1(s_1(t))$ для $t \in \{1, 2, \ldots, l\}$; 2) осуществляем линеаризационную атаку на E', а именно: найденные на шаге 1 значения $y_1(1), y_1(2), \ldots, y_1(l)$ подставляем в E' и полученную систему E'' линейных уравнений решаем (методом Гаусса, например) относительно булевых неизвестных в векторах $s_2(t)$ и $s_3(t)$, $t \in \{1, 2, \ldots, l\}$; 3) из каждого из тех решений системы E'', которые удовлетворяют условиям альтернативности для всех t, $1 \leqslant t \leqslant l$, выписываем значения $s_2(1)$ и $s_3(1)$ и тройку $s_1(1)s_2(1)s_3(1)$ принимаем за одно из значений ключа k.

Вычислительная сложность атаки равна 2^{m_1}.

Примечание: эффективным ключом автоматного криптогенератора с альтернативным управлением в этом случае является начальное состояние одного автомата — управляющего, его удлинение посредством начальных состояний управляемых автоматов не добавляет стойкости генератору. Для генератора с альтернативным управлением на регистрах сдвига с линейной обратной связью это показано в [5].

4. $I_{\rm s} = I_{\rm t} = \varnothing$, $I_{\rm o} = \{1\}$; $K_{\rm s} = K_{\rm t} = \varnothing$, $K_{\rm o} = F_1$; $K = K_{\rm o} = F_1$; $k = f_1 \in K$.

А т а к а 4: 1) вычисляем $s_1(t+1) = g_1(s_1(t)), t \in \{1, 2, \ldots, l-1\}$; 2) как в п. 1 атаки 1, методом DSS вычисляем входные последовательности подсети N' сети N; 3) по любой из них $y_1(1)y_1(2)\ldots y_1(l)$ и внутренней последовательности $s_1(1)s_1(2)\ldots s_1(l)$ автомата A_1 строим частичную функцию f_1' как $f_1'(s_1(t)) = y_1(t)$ для $t \in \{1, 2, \ldots, l\}$; 4) доопределяем f_1' до некоторой функции f_1 в классе F_1 и в случае успешности доопределения выдаём последнюю как одно из значений ключа k.

Примечание: если вычисление на шаге 3 выполнить для каждой последовательности, вычисленной на шаге 2, то будут получены все значения ключа k, при которых генератор Σ вырабатывает данную γ.

5. $I_\text{s} = I_\text{t} = \varnothing$, $I_\text{o} = \{2\}$; $K_\text{s} = K_\text{t} = \varnothing$, $K_\text{o} = F_2$; $K = K_\text{o} = F_2$; $k = f_2 \in K$.

А т а к а 5: 1) вычисляем $s_1(t+1) = g_1(s_1(t))$, $y_1(t) = f_1(s_1(t))$ в автомате A_1 и $s_3(t+1) = g_3(y_1(t), s_3(t))$, $y_3(t) = f_3(y_1, s_3(t))$ в автомате A_3 для $t \in \{1, 2, \ldots, l\}$; 2) строим частичную функцию f_2' как $f_2'(y_1(t), s_2(t)) = y(t) \oplus y_3(t)$ для $t \in \{1, 2, \ldots, l\}$; 3) доопределяем f_2' до некоторой функции f_2 в классе F_2 и в случае успешности доопределения выдаём последнюю как значение ключа k.

Примечание: атака остаётся в силе, если в ней автоматы A_2 и A_3 переставить местами.

6. $I_\text{s} = I_\text{t} = \varnothing$, $I_\text{o} = \{2, 3\}$; $K_\text{s} = K_\text{t} = \varnothing$, $K_\text{o} = F_2 \times F_3$; $K = K_\text{o} = F_2 \times F_3$; $k = f_2 f_3 \in K$.

А т а к а 6: 1) вычисляем $s_1(t+1) = g_1(s_1(t))$, $y_1(t) = f_1(s_1(t))$ в автомате A_1 для $t \in \{1, 2, \ldots, l\}$, $s_2(t+1) = g_2(y_1(t), s_2(t))$ в автомате A_2 и $s_3(t+1) = g_3(y_1(t), s_3(t))$ в автомате A_3 для $t \in \{1, 2, \ldots, l-1\}$; 2) вычисляем 2^l пар последовательностей $y_{2j}(1)y_{2j}(2)\ldots y_{2j}(l)$, $y_{3j}(1)y_{3j}(2)\ldots y_{3j}(l)$, $j \in \{1, 2, \ldots, l\}$, таких, что $y_{2j}(t) = y_{3j}(t) = 0$ или $y_{2j}(t) = y_{3j}(t) = 1$, если $y(t) = 0$, либо $y_{2j}(t) = 0, y_{3j}(t) = 1$ или $y_{2j}(t) = 1, y_{3j}(t) = 0$, если $y(t) = 1$; 3) для каждого $j \in \{1, 2, \ldots, l\}$ строим частичные булевы функции f_{2j} и f_{3j} как $f_{2j}(y_1(t), s_2(t)) = y_{2j}(t)$ и $f_{3j}(y_1(t), s_3(t)) = y_{3j}(t)$, $t \in \{1, 2, \ldots, l\}$, доопределяем их до некоторых функций f_2 и f_3 в F_2 и F_3 соответственно и в случае успешности доопределения выдаём $f_2 f_3$ как одно из значений ключа k.

Вычислительная сложность атаки равна 2^l.

Примечание: если на шаге 3 для каждого j хотя бы одна из функций f_{2j} или f_{3j} оказывается недоопределимой в соответствующем классе F_2 или F_3, то задача криптоанализа генератора Σ в данных условиях не имеет решения.

ЛИТЕРАТУРА

1. *Агибалов Г. П.* Конечные автоматы в криптографии // Прикладная дискретная математика. Приложение. 2009. № 2. С. 43–73.
2. *Tao R.* Finite Automata and Application to Cryptography. TSINGHUA University Press, 2009. 406 p.
3. *Агибалов Г. П., Панкратова И. А.* О двухкаскадных конечно-автоматных криптографических генераторах и методах их криптоанализа // Прикладная дискретная математика. 2017. № 35. С. 38–47.
4. *Агибалов Г. П., Панкратова И. А.* К криптоанализу двухкаскадных конечно-автоматных криптографических генераторов // Прикладная дискретная математика. Приложение. 2016. № 9. С. 41–43.
5. *Агибалов Г. П.* Логические уравнения в криптоанализе генераторов ключевого потока // Вестник Томского государственного университета. Приложение. 2003. № 6. С. 31–41.
6. *Menezes A., van Oorshot P., and Vanstone S.* Handbook of Applied Cryptography. CRC Press Inc., 1997. 661 p.

7. *Фомичёв В. М.* Дискретная математика и криптология. М.: ДИАЛОГ-МИФИ, 2003. 400 с.
8. *Агибалов Г. П.* О некоторых доопределениях частичной булевой функции // Труды Сибирского физико-технического института. 1970. Вып. 49. С. 12–19.
9. *Агибалов Г. П., Сунгурова О. Г.* Криптоанализ конечно-автоматного генератора ключевого потока с функцией выходов в качестве ключа // Вестник Томского государственного университета. Приложение. 2006. № 17. С. 104–108.
10. *Агибалов Г. П.* Методы решения систем полиномиальных уравнений над конечным полем // Вестник Томского государственного университета. Приложение. 2006. № 17. С. 4–9.

REFERENCES

1. *Agibalov G. P.* Konechnye avtomaty v kriptografii [Finite automata in cryptography]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2009, no. 2, pp. 43–73. (in Russian)
2. *Tao R.* Finite Automata and Application to Cryptography. TSINGHUA University Press, 2009. 406 p.
3. *Agibalov G. P. and Pankratova I. A.* O dvukhkaskadnykh konechno-avtomatnykh kriptograficheskikh generatorakh i metodakh ikh kriptoanaliza [About 2-cascade finite automata cryptographic generators and their cryptanalysis]. Prikladnaya Diskretnaya Matematika, 2017, no. 35, pp. 38–47. (in Russian)
4. *Agibalov G. P. and Pankratova I. A.* K kriptoanalizu dvukhkaskadnykh konechno-avtomatnykh kriptograficheskikh generatorov [To cryptanalysis of 2-cascade finite automata cryptographic generators]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2016, no. 9, pp. 41–43. (in Russian)
5. *Agibalov G. P.* Logicheskie uravneniya v kriptoanalize generatorov klyuchevogo potoka [Logical equations in cryptanalysis of key stream generators]. Vestnik TSU. Prilozhenie, 2003, no. 6, pp. 31–41. (in Russian)
6. *Menezes A., van Oorshot P., and Vanstone S.* Handbook of Applied Cryptography. CRC Press Inc., 1997. 661 p.
7. *Fomichev V. M.* Diskretnaya matematika i kriptologiya [Discrete Mathematics and Cryptology]. Moscow, DIALOG-MEPhI Publ., 2003. 400 p. (in Russian)
8. *Agibalov G. P.* O nekotorykh doopredeleniyakh chastichnoy bulevoy funktsii [Some completions of partial Boolean function]. Trudy SPhTI, 1970, iss. 49, pp. 12–19. (in Russian)
9. *Agibalov G. P. and Sungurova O. G.* Kriptoanaliz konechno-avtomatnogo generatora klyuchevogo potoka s funktsiey vykhodov v kachestve klyucha [Cryptanalysis of a finite-state keystream generator with an output function as a key]. Vestnik TSU. Prilozhenie, 2006, no. 17, pp. 104–108. (in Russian)
10. *Agibalov G. P.* Metody resheniya sistem polinomial'nykh uravneniy nad konechnym polem [Methods for solving systems of polynomial equations over a finite field]. Vestnik TSU. Prilozhenie, 2006, no. 17, pp. 4–9. (in Russian)

УДК 519.7

ФИНИТНЫЕ ФУНКЦИИ В АЛГОРИТМАХ КРИПТОГРАФИИ

А. В. Щуренко, В. Л. Леонтьев

Ульяновский государственный университет, г. Ульяновск, Россия

Предлагается применять ортогональные финитные функции (ОФФ) в алгоритмах криптографии в том случае, когда ОФФ не обладают свойством ортогональности. При этом последовательность сеточных наборов ОФФ, утративших свойство ортогональности, по-прежнему является базисом в пространстве Соболева. Потеря свойства ортогональности придает ОФФ новое свойство — множественность вариантов задания параметра ОФФ, что позволяет создавать дополнительные ключи в криптографических алгоритмах. Предложен алгоритм шифрования, основанный на использовании ОФФ, утративших свойство ортогональности. Использование ОФФ, не имеющих свойства ортогональности, в алгоритмах криптографии предложено В. Л. Леонтьевым. Реализация алгоритма выполнена в основном А. В. Щуренко.

Ключевые слова: *ОФФ, финитные функции, криптография, шифрование.*

DOI 10.17223/20710410/36/6

COMPACTLY SUPPORTED FUNCTIONS IN CRYPTOGRAPHY ALGORITHMS

A. V. Shchurenko, V. L. Leontiev

Ulyanovsk State University, Ulyanovsk, Russia

E-mail: tshurens@yandex.ru

In the paper, we propose a new symmetric block cipher based on the orthogonal finite functions (OFF) in the Sobolev's space. To encrypt a plaintext $a = a_1 a_2 \dots a_n$, we first convert a to a polynomial $a(x) = a_1 + a_2 x + \dots .. + a_n x^{n-1}$, then approximate $a(x)$ by a linear combination $F(x) = \sum\limits_{i=1}^{n} r_i f_i(x)$, where $f = (f_1, f_2, \dots, f_n)$ is an OFF-basis not having the orthogonality property, and finally compute the ciphertext $b = b_1 b_2 \dots b_n$, where $b_i = F(k_i)$ for some values k_i of x, $i = 1, 2, \dots, n$. The numbers $k_1, \dots, k_n$ and some parameters of functions in f form the key of the cipher. To decrypt the ciphertext b, we first, given $b_1, \dots, b_n$ and key parameters in f, compute the approximation coefficients $r_1, \dots, r_n$, next, given $k_1, \dots, k_n$, compute $x'_1, \dots, x'_n$ such that $a(x'_i) = r_i$ for $i = 1, 2, \dots, n$, then construct $a(x)$ by the Lagrange method, and finally convert $a(x)$ to a.

Keywords: *OFF, finite functions, cryptography, encryption.*

В работе предлагается алгоритм симметричного шифрования, основанный на применении ортогональных финитных функций. Теория ОФФ-базисов и её применение в алгоритмах численных методов изложены в [1 – 3].

Основная идея предлагаемого алгоритма заключается в аппроксимации многочлена, содержащего в себе исходное сообщение, с использованием функций ОФФ-базиса,

не обладающих свойствами ортогональности. После представления блока информации в виде многочлена проводится его аппроксимация с помощью ОФФ-базиса в выбранных узлах сетки, затем вычисляются значения ОФФ-аппроксимации с учётом произвольно заданного значения ключа, после чего результат шифрования представляется в виде блока ОФФ-аппроксимации. Информация о координатах точек, в которых проводилась аппроксимация, и о дополнительном ключе позволяет восстановить исходный многочлен и содержащееся в нём сообщение.

Операции по получению коэффициентов аппроксимации проводятся в кольце многочленов степени, не превосходящей n, над кольцом наименьших неотрицательных вычетов $\mathbb{Z}_N$, элементы которого образуют полную систему вычетов по модулю N. При этом n — чётное число, равное длине сообщения, а N — простое число. Сообщение длины, большей n, при шифровании следует разбить на блоки длины n.

Количество функций в используемой части ОФФ-базиса должно быть не меньше n. Для выполнения этого требования на равномерной сетке $x_1 < x_2 < \ldots < x_l$, $l \geqslant n$, с целыми неотрицательными x_i и шагом h задаётся набор $t = (t_1, t_2, \ldots, t_n)$, состоящий из n точек $t_i = (x_i, y_i)$, где все x_i, $i = 1, \ldots, n$, попарно различны. При этом все y_i принадлежат $\mathbb{Z}_N$, но их точные значения вычисляются уже в процессе шифрования. Далее задаётся соответствующий набор ОФФ $f = (f_1, f_2, \ldots, f_n)$. В качестве примера берутся ОФФ, являющиеся частным случаем [1, с. 11]. При использовании равномерной сетки с шагом h каждому узлу сетки x_i ставится в соответствие сеточная функция вида

$$f_i(x) = \begin{cases} (x - x_{i-1})/h, & x \in [x_{i-1}, x_{i-1} + h_1] \cup [x_{i-1} + h_2, x_i], \\ -\alpha + 2(\alpha h + h_1)(x_{i-1} + d_n - x)/(h(h_2 - h_1)), & x \in [x_{i-1} + h_1, x_{i-1} + d_n], \\ -\alpha + 2(\alpha h + h_2)(x - x_{i-1} - d_n)/(h(h_2 - h_1)), & x \in [x_{i-1} + d_n, x_{i-1} + h_2], \\ (x_{i+1} - x)/h, & x \in [x_i, x_i + h_1] \cup [x_i + h_2, x_{i+1}], \\ \beta + 2(\beta h + h_1 - h)(x - x_i - d_n)/(h(h_2 - h_1)), & x \in [x_i + h_1, x_i + d_n], \\ \beta + 2(\beta h + h_2 - h)(x_i + d_n - x)/(h(h_2 - h_1)), & x \in [x_i + d_n, x_i + h_2], \\ 0, & x \notin [x_{i-1}, x_{i+1}], \end{cases} \quad (1)$$

где $d_n = (h_1 + h_2)/2$; $h_1 = H_1 h$; $h_2 = H_2 h$; $0 \leqslant h_1 < h_2 \leqslant h$; H_1, H_2, α, β — некоторые константы; $\alpha > 0$, $\beta > 1$. Каждая f_i представляет собой сумму B-сплайна первой степени с конечным носителем $[x_{i-1}, x_{i+1}]$ и двух B-сплайнов первой степени, взятых с разными знаками, с более компактными по сравнению с $[x_{i-1}, x_{i+1}]$ конечными носителями. За счёт двух дополнительных B-сплайнов и с помощью выбора значений α, β, h_1, h_2 достигается выполнение условий ортогональности: $(f_i, f_j) = 0$ для всех $i \neq j$. Функции $f_i(x)$, $i = 1, \ldots, n$, линейно независимы, а их аппроксимирующие свойства представлены следующей теоремой.

Теорема 1 [1]. Если $u(x) \in W_2^1(\mathbb{R})$ и $H_1 + H_2 = 1$ ($h_1 + h_2 = h$), а $\alpha = \beta - 1$, то существует функция $u^h = \sum\limits_{i=1}^{n} \alpha_i f_i \in M_n$ (α_i — некоторые постоянные), такая, что

$$||u - u^h||_{\mathbb{L}_2(\mathbb{R})} \leqslant ch||u||_{W_2^1(\mathbb{R})}, \quad \sum_{i=1}^{n} |a_i|^2 \leqslant c_1 ||u||^2_{\mathbb{L}_2(\mathbb{R})},$$

где M_n — линейная оболочка $f_i(x)$; постоянные c и c_1 не зависят от u и h; W_2^l — функциональные пространства Соболева; $L_2 = W_2^0$.

Основная идея алгоритма состоит в том, чтобы, отказавшись от выполнения условия [1] ортогональности финитных функций, имеющего, например, в случае $h_1 = 0$, $h_2 = h$ вид

$$4\alpha\beta + \alpha - \beta = 0,$$

подбирать ОФФ, исходя только из условия $\alpha = \beta - 1$ теоремы. При этом числовое значение параметра β может быть любым вещественным числом, большим единицы, что является основой дополнительного шага шифрования исходного сообщения, связанного не только с деталями алгоритма шифрования, но и с дополнительным произвольно задаваемым значением параметра β. В общем случае значения β могут быть разными на различных участках выбранной сетки с узлами x_i, что порождает множество дополнительных ключей и поэтому значительно повышает уровень защищённости передаваемого сообщения. Отказ от ортогональности ОФФ значительно расширяет область применения ОФФ, которые, не обладая уже свойством ортогональности, представляют собой особую часть исходного множества ОФФ. Функции (1) в случае $h_1 = 0, h_2 = h$ принимают следующий вид:

$$f_i(x) = \begin{cases} 2\alpha(x_{i-1} - x)/h, & x \in [x_{i-1}, x_{i-1} + h/2], \\ 2(\alpha+1)(x - x_i)/h + 1, & x \in [x_{i-1} + h/2, x_i], \\ 2(\beta-1)(x - x_i)/h + 1, & x \in [x_i, x_i + h/2], \\ 2\beta(x_{i+1} - x)/h, & x \in [x_i + h/2, x_{i+1}], \\ 0, & x \notin [x_{i-1}, x_{i+1}]. \end{cases} \tag{2}$$

Пусть A — множество всех открытых текстов; K — множество выбранных случайным образом векторов k; B — множество всех шифртекстов. Алфавитом A является множество $\mathbb{Z}_L$ $(1 < L \leqslant N)$, алфавитом K — множество $\mathbb{Z}_{x_l+1}$, алфавитом B — множество целых чисел от $(-\lfloor \beta L \rceil - 1)$ до $(\lfloor \beta L \rceil + 1)$, $h \in \mathbb{Z}_{x_l}$, $\beta \in \mathbb{R}$ $(\beta > 1)$. Используемые здесь и далее усечённые квадратные скобки означают операцию округления до ближайшего целого числа.

Запишем исходное сообщение $a \in A$ в векторном виде:

$$a = (a_1, a_2, \ldots, a_n).$$

Пусть также выбран вектор $k \in K$, $k = (k_1, k_2, \ldots, k_n)$, и значения β, h и x_1, связанные с используемой сеткой. При этом все компоненты k попарно различны и для всех $i = 1, \ldots, n/2$ имеет место

$$\begin{cases} k_m \in [x_{j_i}, x_{j_{i+1}}], & m = 2i-1, 2i, \\ k_m \notin [x_{j_{i-1}}, x_{j_{i+2}}], & m \neq 2i-1, 2i, \end{cases}$$

где x_{j_i} и $x_{j_{i+1}}$ $(j_i, j_{i+1} \in \mathbb{Z}_l)$ — узлы сетки, между которыми лежат k_{2i-1} и k_{2i}. Другими словами, каждая компонента k_i является координатой на оси Ox точки, лежащей между двумя соседними узлами сетки x_{j_i} и $x_{j_{i+1}}$. Кроме того, если k_{2i-1} и k_{2i} лежат в $[x_{j_i}, x_{j_{i+1}}]$, то в данном интервале и в соседних $[x_{j_{i-1}}, x_{j_i}]$, $[x_{j_{i+1}}, x_{j_{i+2}}]$ не лежит более ни одна из компонент k_l, отличная от k_{2i-1} и k_{2i}.

Для большей определённости в предлагаемом алгоритме пусть $k_{2i} \in [x_{j_i} + h/2, x_{j_{i+1}}]$, $k_{2i-1} \in [x_{j_i}, x_{j_i} + h/2]$. Вместе с тем возможны другие варианты расположения точек с координатами k_{2i-1}, k_{2i} и связанные с этим другие варианты алгоритма.

При $k_{2i-1} \in [x_{j_i}, x_{j_i} + h/2]$, $k_{2i} \in [x_{j_i} + h/2, x_{j_{i+1}}]$, для исключения многозначности результатов расшифрования, пусть

$$(\beta - 1)(k_{2i-1} - x_{j_i}) > \beta(x_{j_{i+1}} - k_{2i}). \tag{3}$$

В силу ограничений, наложенных на элементы k, целесообразно проводить их выбор в три этапа. На первом шаге выбирается конкретный участок $[x_{j_i}, x_{j_{i+1}}]$, на втором шаге — k_{2i-1}, на третьем шаге — значение k_{2i} с учётом условия (3):

$$k_{2i} > x_{j_{i+1}} - \frac{(\beta - 1)(k_{2i-1} - x_{j_i})}{\beta}. \tag{4}$$

При невозможности выбора k_{2i} второй и третий шаги повторяются.

При этом следует отметить, что условия выбора элементов вектора k не препятствуют использованию для их генерации криптографически стойких генераторов случайных чисел. На всех трёх этапах, а также при генерации значений β, h и x_1 возможно использование таких алгоритмов, как алгоритм Блюм — Блюма — Шуба или криптографически стойкая хэш-функция. Это сводит задачу генерации ключей к передаче начальных значений генераторов, что упрощает задачу генерации, рассылки и хранения ключей без ущерба для криптостойкости алгоритма.

Обозначим x' вектор узлов сетки, используемых при аппроксимации многочлена. При этом $x'_{2i-1} = x_{j_i}$, $x'_{2i} = x_{j_{i+1}}$ $(i = 1, \ldots, n/2)$. В качестве f возьмем набор $f = (f_1, f_2, \ldots, f_n)$, где каждая f_i — ОФФ-функция, соответствующая узлу x'_i. Вектор (k, β, h, x_1) является ключом, вектор $b \in B$, $b = (b_1, b_2, \ldots, b_n)$, — шифртекстом.

Алгоритм шифрования состоит из следующих четырёх шагов.

Ш а г 1. Вектору a сопоставляется многочлен

$$a(x) = a_1 + a_2 x + a_3 x^2 + \ldots + a_n x^{n-1}. \tag{5}$$

Ш а г 2. Проводится аппроксимация многочлена $a(x)$ с помощью функций f_i. При этом аппроксимирующая функция $F(x)$ является суммой произведений ОФФ-функций f_i и соответствующих коэффициентов аппроксимации r_i:

$$F(x) = \sum_{i=1}^{n} r_i f_i(x).$$

Так как $f_j(x'_i) = \delta_{ij}$, где δ_{ij} — символ Кронекера $(i, j \in \{1, \ldots, n\})$, коэффициент i-й ОФФ-функции равен значению многочлена в точке с координатой x'_i. Значения этих коэффициентов — неотрицательные целые числа. Таким образом, значения вектора коэффициентов аппроксимации $r = (r_1, r_2, \ldots, r_n)$ находятся по формуле

$$r_i = a(x'_i) \bmod N.$$

Ш а г 3. Формируется вектор $b = (b_1, b_2, \ldots, b_n)$, где b_i равно значению аппроксимирующей функции в точке с координатой по оси Ox равной k_i, то есть $b_i = F(k_i)$, $i = 1, \ldots, n$.

Поскольку для всех $i \in \{1, \ldots, n/2\}$ точки k_{2i-1} и k_{2i} лежат между узлами x'_{2i-1} и x'_{2i}, значения аппроксимирующей функции в данных точках равны линейным комбинациям соответствующих значений f_{2i-1} и f_{2i} функций (2):

$$F(k_{2i-1}) = r_{2i-1} f_{2i-1}(k_{2i-1}) + r_{2i} f_{2i}(k_{2i-1}),$$
$$F(k_{2i}) = r_{2i-1} f_{2i-1}(k_{2i}) + r_{2i} f_{2i}(k_{2i}).$$

Согласно (2),

$$f_{2i-1}(k_{2i-1}) = 2(\beta-1)(k_{2i-1} - x'_{2i-1})/h + 1,$$
$$f_{2i}(k_{2i-1}) = 2\alpha(x'_{2i-1} - k_{2i-1})/h,$$
$$f_{2i-1}(k_{2i}) = 2\beta(x'_{2i} - k_{2i})/h,$$
$$f_{2i}(k_{2i}) = 2(\alpha+1)(k_{2i} - x'_{2i})/h + 1.$$

Таким образом,

$$\begin{aligned} F(k_{2i-1}) &= r_{2i-1}\left(\frac{2(\beta-1)(k_{2i-1}-x'_{2i-1})}{h}+1\right) + r_{2i}\frac{2(\beta-1)(x'_{2i-1}-k_{2i-1})}{h} = \\ &= \frac{2(\beta-1)(k_{2i-1}-x'_{2i-1})}{h}(r_{2i-1}-r_{2i}) + r_{2i-1}, \\ F(k_{2i}) &= r_{2i-1}\frac{2\beta(x'_{2i}-k_{2i})}{h} + r_{2i}\left(\frac{2\beta(k_{2i}-x'_{2i})}{h}+1\right) = \\ &= \frac{2\beta(x'_{2i}-k_{2i})}{h}(r_{2i-1}-r_{2i}) + r_{2i}. \end{aligned} \tag{6}$$

Ш а г 4. Вектор b, являющийся шифртекстом, записывается в виде

$$b = (\lfloor F(k_1)\rceil, \lfloor F(k_2)\rceil, \ldots, \lfloor F(k_n)\rceil).$$

Алгоритм расшифрования заключается в нахождении коэффициентов исходного многочлена $a(x)$ на основе вектора шифртекста b, известных значений k и β, а также параметров сетки h и x_1. Запишем шифртекст $b \in B$ в векторном виде

$$b = (b_1, b_2, \ldots, b_n),$$

где n — чётное натуральное число. Пусть известен вектор $k \in K$, $k = (k_1, k_2, \ldots, k_n)$, использованный при шифровании, а также значения β, h и x_1. При этом все k_i попарно различны и все k_{2i-1} и k_{2i} $(i = 1, \ldots, n/2)$ являются координатами на оси Ox точек, лежащих между двумя соседними узлами сетки x'_{2i-1} и x'_{2i}.

Ш а г 1. Вектору a исходного сообщения при шифровании был сопоставлен многочлен (5). При проведении аппроксимации $a(x)$ с помощью ОФФ-функций f в точках x' получился вектор коэффициентов аппроксимации $r = (r_1, r_2, \ldots, r_n)$, компоненты которого находятся по формуле

$$r_i = a(x_i) \bmod N.$$

Для нахождения значений вектора r при расшифровании значения вектора b берутся парами и первый элемент в паре складывается с элементом, обратным второму, то есть определяются значения $b'_i = b_{2i-1} - b_{2i}$ $(i = 1, \ldots, n/2)$:

$$b'_i = \left(\frac{2(\beta-1)(k_{2i-1}-x'_{2i-1})}{h} - \frac{2\beta(x'_{2i}-k_{2i})}{h} + 1\right)(r_{2i-1}-r_{2i}).$$

Далее находятся значения $(r_{2i-1} - r_{2i})$ по формуле

$$(r_{2i-1}-r_{2i}) = \left\lfloor \frac{b'_i}{2(\beta-1)(k_{2i-1}-x'_{2i-1})/h - 2\beta(x'_{2i}-k_{2i})/h + 1} \right\rceil.$$

Ш а г 2. Вычисляются значения вектора r. Для этого используется формула (6)

$$b_{2i} = \lfloor F(k_{2i}) \rceil = \left\lfloor \frac{2\beta(x'_{2i} - k_{2i})}{h}(r_{2i-1} - r_{2i}) \right\rceil + r_{2i},$$

где $i = 1, \ldots, n/2$. Отсюда следует, что

$$r_{2i} = b_{2i} - \left\lfloor \frac{2\beta(x'_{2i} - k_{2i})}{h}(r_{2i-1} - r_{2i}) \right\rceil, \quad r_{2i-1} = (r_{2i-1} - r_{2i}) + r_{2i}.$$

Ш а г 3. Вектор коэффициентов аппроксимации r теперь известен, по значениям вектора k определяются все x'_i ($k_{2i} \in [x'_{2i-1}, x'_{2i}]$, $i = 1, \ldots, n/2$). Решение системы уравнений

$$\begin{cases} a(x'_1) = r_1, \\ a(x'_2) = r_2, \\ \ldots \\ a(x'_n) = r_n \end{cases} \tag{7}$$

даёт коэффициенты многочлена $a(x)$. Найти целочисленное решение данной системы можно, применив сеточные полиномы Лагранжа для набора точек $t = ((x'_1, r_1), (x'_2, r_2), \ldots, (x'_n, r_n))$. Тогда $a(x)$ имеет вид

$$a(x) = \sum_{i=1}^{n} r_i l_i(x),$$

где $l_i(x) = \prod\limits_{j=1, j\neq i}^{n} \dfrac{x - x'_j}{x'_i - x'_j}$ есть многочлены Лагранжа, связанные с узлами x'_i сетки.

Ш а г 4. По значениям коэффициентов многочлена $a(x)$ строится вектор a, являющийся исходным сообщением.

Пример 1. Пусть дана равномерная сетка, определяемая значениями $h = 10$, $x_1 = 0$, и задано значение параметра ОФФ $\beta = 3{,}75$. При этом ОФФ-функции $f_i(x)$ имеют следующий вид:

$$\begin{cases} 5{,}5(x_{i-1} - x)/10, & x \in [x_{i-1}, x_{i-1} + 5], \\ 7{,}5(x - x_i)/10 + 1, & x \in [x_{i-1} + 5, x_i], \\ 5{,}5(x - x_i)/10 + 1, & x \in [x_i, x_i + 5], \\ 7{,}5(x_{i+1} - x)/10, & x \in [x_i + 5, x_{i+1}], \\ 0, & x \in [x_{i-1}, x_{i+1}]. \end{cases}$$

Пусть также $N = 257$, $L = 256$, $l = 26$. Тогда $x_l = 250$, алфавит $A = \mathbb{Z}_{256}$, алфавит $K = \mathbb{Z}_{251}$, алфавит B — множество целых чисел от -961 до 961.

Шифрование. Пусть исходное сообщение имеет вид $a = (20, 13, 2, 4, 5, 1)$, то есть $n = 6$. Пусть заданы значения вектора $k = (14, 19, 33, 39, 53, 58)$. При этом для всех компонент k справедливо следующее условие: если k_{2i-1} и k_{2i} лежат в $[x_{j_i}, x_{j_{i+1}}]$, то в данном интервале и в соседних интервалах $[x_{j_{i-1}}, x_{j_i}]$, $[x_{j_{i+1}}, x_{j_{i+2}}]$ не лежит более ни одна из компонент k, отличная от k_{2i-1} и k_{2i}.

Проверка условия (4)

$$19 > 20 - \frac{(3{,}75-1)(14-10)}{3{,}75} = 17{,}07,$$
$$39 > 40 - \frac{(3{,}75-1)(33-30)}{3{,}75} = 37{,}8,$$
$$58 > 60 - \frac{(3{,}75-1)(53-50)}{3{,}75} = 57{,}8$$

показывает его выполнение.

Случайные величины x_1 и h получили в результате генерации значения $x_1 = 0$, $h = 10$, поэтому вектор координат узлов сетки имеет вид $x' = (10, 20, 30, 40, 50, 60)$.

Ш а г 1. Вектору a сопоставляется многочлен $a(x) = 20+13x+2x^2+4x^3+5x^4+x^5$.

Ш а г 2. Проводится аппроксимация $a(x)$ с помощью ОФФ-функций f. При этом используется значение $\beta = 3{,}75$. В результате получается вектор коэффициентов аппроксимации $r = (r_1, r_2, r_3, r_4, r_5, r_6)$:

$$r_1 = a(10) = 154350 \equiv 150 \pmod{257},$$
$$r_2 = a(20) = 4033080 \equiv 236 \pmod{257},$$
$$r_3 = a(30) = 28460210 \equiv 30 \pmod{257},$$
$$r_4 = a(40) = 115459740 \equiv 177 \pmod{257},$$
$$r_5 = a(50) = 344255670 \equiv 58 \pmod{257},$$
$$r_6 = a(60) = 843272000 \equiv 2 \pmod{257}.$$

Получился вектор $r = (150, 236, 30, 177, 58, 2)$.

Ш а г 3. Строится вектор b:

$$b_1 = \lfloor F(k_1) \rceil = \left\lfloor \frac{2(3{,}75-1)(14-10)}{10}(150-236)+150 \right\rceil = -39,$$
$$b_2 = \lfloor F(k_2) \rceil = \left\lfloor \frac{2\cdot 3{,}75(20-19)}{10}(150-236)+236 \right\rceil = 172,$$
$$b_3 = \lfloor F(k_3) \rceil = \left\lfloor \frac{2(3{,}75-1)(33-30)}{10}(30-177)+30 \right\rceil = -213,$$
$$b_4 = \lfloor F(k_4) \rceil = \left\lfloor \frac{2\cdot 3{,}75(40-39)}{10}(30-177)+177 \right\rceil = 67,$$
$$b_5 = \lfloor F(k_5) \rceil = \left\lfloor \frac{2(3{,}75-1)(53-50)}{10}(58-2)+58 \right\rceil = 150,$$
$$b_6 = \lfloor F(k_6) \rceil = \left\lfloor \frac{2\cdot 3{,}75(60-58)}{10}(58-2)+2 \right\rceil = 86,$$
$$b = (-39, 172, -213, 67, 150, 86).$$

Вектор b является шифртекстом.

Расшифрование. При расшифровании шифртекста $b = (-39, 172, -213, 67, 150, 86)$ используются сформированные при шифровании векторы и величины $k = (14, 19, 33, 39, 53, 58)$, $\beta = 3{,}75$, $h = 10$, $x_1 = 0$, $x' = (10, 20, 30, 40, 50, 60)$, определяющие ключ.

Ш а г 1. Находятся значения выражений $(r_{2i-1} - r_{2i})$, $i = 1, \ldots, n/2$:

$$b'_1 = \left(\frac{2(\beta-1)(k_1 - x'_1)}{h} - \frac{2\beta(x'_2 - k_2)}{h} + 1\right)(r_1 - r_2) = -39 - 172 = -211,$$

$$r_1 - r_2 = \left\lfloor \frac{-211}{(2(\beta-1)(k_1 - x'_1)/h - 2\beta(x'_2 - k_2)/h + 1)} \right\rceil = \left\lfloor \frac{-211}{2{,}2 - 0{,}75 + 1} \right\rceil = -86,$$

$$b'_2 = \left(\frac{2(\beta-1)(k_3 - x'_3)}{h} - \frac{2\beta(x'_4 - k_4)}{h} + 1\right)(r_3 - r_4) = -213 - 67 = -280,$$

$$r_3 - r_4 = \left\lfloor \frac{-280}{2(\beta-1)(k_3 - x'_3)/h - 2\beta(x'_4 - k_4)/h + 1} \right\rceil = \left\lfloor \frac{-280}{1{,}65 - 0{,}75 + 1} \right\rceil = -147,$$

$$b'_3 = \left(\frac{2(\beta-1)(k_5 - x'_5)}{h} - \frac{2\beta(x'_6 - k_6)}{h} + 1\right)(r_5 - r_6) = 150 - 86 = 64,$$

$$r_5 - r_6 = \left\lfloor \frac{64}{2(\beta-1)(k_5 - x'_5)/h - 2\beta(x'_6 - k_6)/h + 1} \right\rceil = \left\lfloor \frac{64}{1{,}65 - 1{,}5 + 1} \right\rceil = 56.$$

Ш а г 2. Для определения значений компонент вектора r вычисляются

$$b_2 = \left\lfloor \frac{2 \cdot 3{,}75(20-19)}{10} \cdot (-86) \right\rceil + r_2 = -64 + r_2 = 172,$$

$$r_2 = 172 + 64 = 236, \quad r_1 = (r_1 - r_2) + r_2 = -86 + 236 = 150,$$

$$b_4 = \left\lfloor \frac{2 \cdot 3{,}75(40-39)}{10} \cdot (-147) \right\rceil + r_4 = -110 + r_4 = 67,$$

$$r_4 = 67 + 110 = 177, \quad r_3 = (r_3 - r_4) + r_4 = -147 + 177 = 30,$$

$$b_6 = \left\lfloor \frac{2 \cdot 3{,}75(60-58)}{10} \cdot 56 \right\rceil + r_6 = 84 + r_6 = 86,$$

$$r_6 = 86 - 84 = 2, \quad r_5 = (r_5 - r_6) + r_6 = 56 + 2 = 58.$$

Поэтому $r = (150, 236, 30, 177, 58, 2)$.

Ш а г 3. Решается система уравнений вида (7):

$$\begin{cases} a(10) = 150 \pmod{257}, \\ a(20) = 236 \pmod{257}, \\ a(30) = 30 \pmod{257}, \\ a(40) = 177 \pmod{257}, \\ a(50) = 58 \pmod{257}, \\ a(60) = 2 \pmod{257}. \end{cases}$$

В итоге получается многочлен $a(x) = 20 + 13x + 2x^2 + 4x^3 + 5x^4 + x^5$.

Ш а г 4. По значениям коэффициентов многочлена $a(x)$ строится вектор $a = (20, 13, 2, 4, 5, 1)$, являющийся исходным сообщением.

Замечание 1. Для обоснования выбора простого N запишем систему (7) в матричной форме:

$$\begin{pmatrix} 1 & x'_1 & \ldots & (x'_1)^{n-1} \\ \vdots & & \ddots & \vdots \\ 1 & x'_n & \ldots & (x'_n)^{n-1} \end{pmatrix} \begin{pmatrix} a_1 \\ \ldots \\ a_n \end{pmatrix} = \begin{pmatrix} r_1 \\ \ldots \\ r_n \end{pmatrix}.$$

Если N простое и все точки x'_i, $i = 1, \ldots, n$, различны, то основная матрица системы невырожденная и, следовательно, система имеет единственное решение.

Замечание 2. В рассмотренном примере показано использование алгоритма в режиме простой замены, однако в практических реализациях, за исключением случаев шифрования коротких сообщений, целесообразно использование иных режимов. При этом разницу в объёмах блоков открытого текста и шифртекста можно компенсировать, заменяя в процессе шифрования очередной блок открытого текста x на результат обратимой функции $g(x)$, по объёму равный блоку шифртекста.

Поскольку алгоритм оперирует с блоками текста фиксированной длины n, уместно его сравнение с алгоритмами блочного шифрования. К достоинствам предложенного алгоритма относится динамическая длина ключа, что упрощает переход на версии алгоритма с большей длиной n блока текста. Более того, значения N и L также не фиксированы, что позволяет алгоритму шифрования на основе ОФФ оперировать с множествами A, K и B, имеющими алфавит любой длины, а также задавать сетки с любым максимальным значением x_l.

Параметр β может принимать любые вещественные значения больше единицы, однако в практических реализациях алгоритма целесообразно взять некоторую конечную область значений β, разбить её на интервалы, например длины $1/N$, и выбрать значения β, лежащие на концах данных интервалов. При этом поскольку число возможных значений β на каждом участке $[i, i+1]$, $i = 0, \ldots, N-1$, зависит от N, при увеличении N расширяется множество допустимых значений β. Значительно увеличивается преимущество алгоритма в стойкости шифрования при использовании любых вещественных значений β по сравнению с целыми значениями. Увеличение максимального значения β расширяет множество допустимых значений ключа, однако вместе с тем значительно возрастает объём шифртекста по сравнению с открытым текстом. В связи с этим в практических реализациях алгоритма рекомендуется ограничивать максимальное значение β, находя баланс между возрастающим объёмом шифртекста и количеством допустимых значений ключа.

Сравним алгоритм с поточным шифром гаммирования. При многократном использовании ключа существенно повышается стойкость шифрования в предложенном алгоритме по сравнению с шифром гаммирования к таким видам криптоанализа, как атаки на основе шифртекста, известного открытого текста, выбранного открытого текста и выбранного шифртекста. При шифровании гаммированием знание открытого текста и шифртекста позволяет определить точное значение гаммы, что исключает многократное использование одного и того же ключа. В случае шифрования предложенным методом знание двух сообщений a и b не позволяет однозначно подобрать параметры (k, β, h, x_1). Изменение любого из них, при соответствующих изменениях других параметров, приведёт к тому, что данному открытому тексту a будет соответствовать тот же шифртекст b.

Уровень сложности дешифрования в предложенном алгоритме характеризуется необходимостью перебора возможных значений параметров h и x_1 и решения со всеми возможными парами (h, x_1) уравнений (6) с неизвестными β, r_{2i-1} и r_{2i}, а также необходимостью решения системы (7).

Увеличение n приводит к увеличению мощности множества допустимых значений ключа, вместе с тем растёт сложность шифрования и расшифрования, поскольку максимальная степень многочлена (5), а также количество уравнений (7) зависят от n.

Пусть, например, $N = 65537$, $h = 4$, $n = 30$, $x_l = 65536$, а максимальное допустимое значение β равно 256. Пусть также $x_1 = 0$. Тогда сетка разбивается на $65536/(2h) = 8192$ различных участка, каждому из которых могут принадлежать две компоненты вектора k. С учётом условия (4), каждому участку $[x'_{2i-1}, x'_{2i+1}]$ могут соответствовать

16 допустимых значений пар (k_{2i}, k_{2i-1}). Общее число допустимых значений компонент вектора k в таком случае равно

$$|K| = A_{8192}^{15} \cdot 8 = \frac{8192!}{(8192-15)!} \cdot 8 \approx 3{,}2 \cdot 10^{63}.$$

Разобьём область допустимых значений β ($\beta > 1$) на интервалы длины $1/N$. Таким образом, для каждого возможного значения вектора k существует $255 \cdot 65536 - 1 = 16711679$ допустимых значений β.

Множество допустимых значений ключа содержит $3{,}2 \cdot 10^{63} \cdot 16711679 \approx 5{,}3 \cdot 10^{70}$ элементов, что даже при многократном использовании ключа делает абсурдными задачи полного перебора либо записи всех подобранных сочетаний (k, β, h, x_1) с целью вскрытия следующего шифртекста перебором лишь записанных заранее значений. При этом следует учесть, что в практических реализациях алгоритма возможно разделение области допустимых значений β на интервалы длины меньше $1/N$, а также нефиксированное значение x_1, что дополнительно расширяет множество допустимых значений ключа.

Вместе с тем при уменьшении n распределение символов шифртекста становится все более схожим с распределением символов открытого текста, что значительно упрощает задачу дешифрования без знания ключа.

В дальнейшем предполагается более подробное исследование стойкости предложенных алгоритмов к различным атакам в зависимости от выбранных параметров и ключа.

Методы использования базисов в криптографических алгоритмах развиваются и в других работах [4, 5], новизна предложенного здесь алгоритма шифрования заключается в использовании принципиально иных базисных функций — ОФФ, не обладающих свойством ортогональности.

ЛИТЕРАТУРА

1. *Леонтьев В. Л.* Ортогональные финитные функции и численные методы. Ульяновск: Изд-во УлГУ, 2003. 178 с.
2. *Леонтьев В. Л., Лукашенец Н. Ч.* Сеточные базисы ортогональных финитных функций // Журн. вычисл. матем. и матем. физ. 1999. Т. 39. № 7. С. 1158–1168.
3. *Леонтьев В. Л.* Ортогональные сплайны и вариационно-сеточный метод // Математическое моделирование. 2002. Т. 14. № 3. С. 117–127.
4. *Лукомский Д. С., Лукомский С. Ф.* Всплесковые базисы и криптография // Математика. Механика. 2011. № 13. С. 55–58.
5. *Левина А. Б.* Сплайн-вэйвлеты и их некоторые применения: дис. ... канд. физ.-мат. наук. СПб., 2009. 214 с.

REFERENCES

1. *Leont'ev V. L.* Ortogonal'nye finitnye funktsii i chislennye metody [Orthogonal finite functions and numerical methods]. Ul'yanovsk, UlSU Publ., 2003. 178 p. (in Russian)
2. *Leont'ev V. L., Lukashenets N. Ch.* Setochnye bazisy ortogonal'nykh finitnykh funktsiy [Grid bases of orthogonal compactly supported functions]. Zh. Vychisl. Mat. Mat. Fiz., 1999, vol. 39, no. 7, pp. 1158–1168. (in Russian)
3. *Leont'ev V. L.* Ortogonal'nye splayny i variatsionno-setochnyy metod [Orthogonal splines and variational-grid method]. Matem. Mod., 2002, vol. 14, no. 3, pp. 117–127. (in Russian)

4. *Lukomskiy D. S., Lukomskiy S. F.* Vspleskovye bazisy i kriptografiya [Splash bases and cryptography]. Matematika. Mekhanika, 2011, no. 13, pp. 55–58. (in Russian)
5. *Levina A. B.* Splayn-veyvlety i ikh nekotorye primeneniya [Spline-Wavelets and some their Applications]. PhD Thesis. SPb., 2009. 214 p. (in Russian)

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 004.492.3

АНАЛИЗ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ МОДЕЛЕЙ ПРИВИЛЕГИЙ И API-ВЫЗОВОВ ВРЕДОНОСНЫХ ПРИЛОЖЕНИЙ

А. А. Сковорода, Д. Ю. Гамаюнов

Московский государственный университет им. М. В. Ломоносова, г. Москва, Россия

Предложен метод автоматической классификации мобильных приложений на основе статического анализа и сопоставления моделей, полученных по его результатам, с моделями ранее известных вредоносных приложений. Модели основаны на привилегиях и API-вызовах, используемых в приложении. Все шаги анализа, а также построение моделей полностью автоматизированы. Таким образом, метод адаптирован для автоматизированного использования магазинами мобильных приложений или другими заинтересованными организациями.

Используя предложенный метод, мы проанализировали коллекции вредоносных приложений Drebin и ISCX, а также более 40000 приложений из Google Play, собранных в период с 2013 по 2016 г. Результаты анализа показывают, что комбинация достаточно простых признаков, таких, как запрашиваемые привилегии и цепочки используемых API-вызовов, достаточна для проведения бинарной классификации приложений на вредоносные и легитимные, а также для определения семейства вредоносных приложений, при этом количество ложных срабатываний приемлемо (около 3 %). Результаты исследования коллекции вредоносных приложений показывают, что нынешние вредоносные Android-приложения редко используют техники обфускации или шифрования для затруднения статического анализа, что пока не соответствует наблюдаемому в области семейства платформ «Wintel». Представлено экспериментальное сравнение предложенного метода с другим эффективным методом анализа Android-приложений, реализованном в программном средстве adagio.

Ключевые слова: *статический анализ, вредоносные Android-приложения.*

DOI 10.17223/20710410/36/7

AUTOMATED STATIC ANALYSIS AND CLASSIFICATION OF ANDROID MALWARE USING PERMISSION AND API CALLS MODELS

A. A. Skovoroda, D. Y. Gamayunov

Lomonosov Moscow State University, Moscow, Russia

E-mail: nastya_jane@seclab.cs.msu.su

In this paper, we propose a heuristic approach to static analysis of Android applications based on matching suspicious applications with the predefined malware models.

Static models are built from Android capabilities and Android Framework API call chains used by the application. All of the analysis steps and model construction are fully automated. Therefore, the method can be easily deployed as one of the automated checks provided by mobile application marketplaces or other interested organizations.

Using the proposed method, we analyzed the Drebin and ISCX malware collections in order to find possible relationships and dependencies between samples in collections, and a large fraction of Google Play apps collected between 2013 and 2016 representing benign data. Analysis results show that a combination of relatively simple static features represented by permissions and API call chains is enough to perform binary classification between malware and benign apps, and even find the corresponding malware family, with an appropriate false positive rate of about 3 %. Malware collections exploration results show that modern Android malware rarely uses obfuscation or encryption techniques to make static analysis more difficult, which is quite the opposite of what we see in the case of the "Wintel" endpoint platform family.

We also provide the experiment-based comparison with the previously proposed state-of-the-art Android malware detection method *adagio*. This method outperforms our proposed method in resulting detection coverage (98 vs 91 % of malicious samples are covered) while at the same time causing a significant number of false alarms corresponding to 9.3 % of benign applications on average.

Keywords: *static analysis, Android malware.*

Введение

С появлением мобильных устройств, обладающих широким спектром предоставляемых возможностей, стала существенной и проблема мобильной безопасности. Многие пользователи хранят на устройствах личную и конфиденциальную информацию, включая данные о банковских картах и данные, необходимые для авторизации в различных системах. Получение пользовательских данных является целью большинства вредоносных мобильных приложений.

Для борьбы с вредоносными приложениями (ВП) предложено множество методов обнаружения и предотвращения [1]. Из методов обнаружения выделяют обнаружение аномалий и обнаружение злоупотреблений. Методы обнаружения злоупотреблений можно условно разделить на две группы: методы, использующие некоторые предположения о свойствах (функциях) вредоносных приложений, и методы, использующие для обнаружения сигнатуры/модели известных вредоносных образцов. К первой группе можно отнести, например, taint-анализ, нацеленный на обнаружение утечек данных с устройства [2, 3]. Такие методы ограничены теми классами ВП, функции которых в них рассматриваются. Однако задание алгоритма обнаружения для произвольной вредоносной функциональности весьма затруднительно, и такие методы, как правило, ограничены узким классом обнаруживаемых приложений. Другая группа методов также имеет недостаток: такие методы существенно зависят от используемых сигнатур/моделей и неприменимы для обнаружения неизвестных ВП (zero-days). Методы обнаружения аномалий [4, 5] требуют задания моделей легитимной функциональности, что является ещё более затруднительным, так как требуется гораздо более широкое покрытие функциональности по сравнению с вредоносной.

По основополагающим техникам методы обнаружения ВП можно разделить на две крупных группы: статические и динамические, в зависимости от использования эмуляции/исполнения анализируемого приложения. Более детальная классификация

методов обнаружения включает методы машинного обучения, анализ привилегий, мониторинг уровня заряда аккумулятора, облачные вычисления [1].

В работе описывается метод статического анализа приложений, использующий модели ВП. Модели приложений отражают их поведение и строятся, исходя из используемых приложением системных привилегий и API-вызовов — вызовов программного интерфейса фреймворка Android или некоторых сторонних библиотек. Построение моделей ВП не требует ручной работы специалиста, в отличие от многих других методов статического анализа [6, 7]. Результаты анализа приложения представляют собой информацию о том, была ли найдена подобная модель среди вредоносных, и список подобных моделей. Данный метод может быть использован магазинами мобильных приложений для анализа новых приложений перед их публикацией, антивирусными приложениями как составная часть анализа на сервере, корпорациями с политикой BYOD (Bring Your Own Device), а также отдельными исследователями в области информационной безопасности.

Мы применили данный метод для бинарной классификации на множестве приложений из коллекций вредоносных приложений Drebin и ISCX, а также части собранной нами коллекции легитимных приложений. В этом эксперименте метод правильно распознал как вредоносные 91 % приложений, неверно классифицировав 2 % легитимных приложений. Проведение того же эксперимента с программным средством для анализа Android-приложений adagio выявило более широкую полноту покрытия adagio, но при этом и значительно большее число ложных срабатываний.

Мы также применили предложенный метод для исследования коллекций Drebin и ISCX. Вредоносные приложения, для которых анализатором обнаружено сходство, были сгруппированы в кластеры семейств ВП. Полученные кластеры не полностью соответствуют заранее заданному разделению коллекций по семействам, которое должно было быть получено исследователями в процессе ручного анализа каждого из образцов. Нужно отметить, что приложения в таких предопределённых семействах очень сильно разнятся, и для некоторых из них действительно сложно обнаружить сходство даже вручную. Исходя из этого, мы считаем предложенный метод подходящим для определения сходства между приложениями, относящимися к одному и тому же семейству ВП. В работе представлены некоторые особенности и характеристики ВП, полученные в процессе ручного и автоматического анализа исследованных коллекций.

Тестирование метода на коллекции из более 40000 легитимных приложений выявило долю ложных срабатываний 3,2 %, приемлемую для статических методов, но слишком высокую для полностью автоматизированного использования, например в качестве самостоятельного антивирусного средства. Тем не менее при использовании реализации предложенного метода исследователями в области безопасности информация о найденных общих цепочках API-вызовов позволяет без затруднений отсеять возникшие ложные срабатывания. Таким образом, метод может быть очень полезен для упрощения и автоматизации работы вирусных аналитиков. Для полностью автоматизированного использования метод может быть дополнен другими подходами к обнаружению для исключения/уменьшения числа ложных срабатываний.

1. Краткий обзор существующих исследований

С момента появления первых угроз, вызванных мобильными ВП, исследователями в области информационной безопасности предложено множество эффективных решений. В [8] представлен всесторонний обзор методов обнаружения ВП, наряду с эволюцией мобильных ВП и их способов противостоять анализу.

Сигнатурные методы обнаружения были использованы одними из первых [9], однако они неустойчивы к модификациям ВП. Для телефонов с небольшими вычислительными ресурсами показали свою эффективность методы мониторинга уровня заряда аккумулятора [5, 10]. Как показано в [11], такие методы больше неприменимы к современному состоянию мобильной среды.

Ряд методов обнаружения ориентируется на некоторые признаки вредоносной активности (такие, как передача по сети конфиденциальной информации без подтверждения пользователем) в статическом представлении приложения [2, 6] или в наблюдаемом поведении [3]. Класс обнаруживаемых ВП для таких методов строго ограничен соответствующими признаками. Так, например, методы, нацеленные на обнаружение утечек данных, не могут быть применены для обнаружения ВП, предназначенных для вымогательства. Кроме того, программные средства с динамическим анализом, предназначенные для выявления утечек данных, нуждаются в дополнении средствами для генерации цепочек событий, приводящих к утечкам [12]. Другая группа методов использует модели/сигнатуры некоторых известных ВП для обнаружения схожей вредоносной активности [13, 9]. Метод, предложенный в данной работе, относится именно к этой группе. В [13, 9] используется преимущественно сигнатурный подход, о недостатках которого упомянуто выше, в работе [13] он дополнен динамическим анализом.

Для обнаружения вредоносных мобильных приложений также широко применяются алгоритмы машинного обучения. Для задач классификации и кластеризации Android-приложений используются как признаки на уровне приложений [14 – 16], так и на уровне операционной системы [17] и генерируемой сетевой активности. В [18] описана кластеризация вредоносных мобильных приложений и генерация их сигнатур на основе наблюдаемого HTTP-трафика. Как и предлагаемый метод, подобные методы существенно зависят от данных, используемых для обучения. Однако, в отличие от нашего метода, в методах, использующих машинное обучение, необходима представительная выборка легитимных приложений для обучения, составление которой довольно затруднительно с учётом большого разнообразия таких приложений. Кроме того, методы, использующие машинное обучение, предоставляют в качестве результата, как правило, бинарную оценку либо вероятность отнесения анализируемого приложения к вредоносным без пояснения причин такой классификации.

Использование API-вызовов и привилегий Android в качестве основы анализа не является отличительной чертой только предлагаемого метода. Многие ранее предложенные в данной области методы машинного обучения используют множество признаков, преимущественно состоящее из такого рода характеристик приложения [19, 20, 16, 14]. Однако во всех перечисленных методах модели приложений (векторы признаков) формируются на основе значительно более поверхностного анализа приложений по сравнению с предложенным в данной работе, и эффективность использования данных методов обоснована прежде всего хорошей приспособленностью алгоритмов машинного обучения к работе на больших выборках приложений. Кроме того, хотя данные методы и предоставляют некоторое разъяснение результатов классификации, оно является гораздо менее детализированным и наглядным по сравнению с пояснением результатов анализа, выдаваемых предлагаемым методом. Экспериментальное сравнение с одним из таких методов проводится в п. 6.

Многие из методов нацелены не на бинарную классификацию приложений, а на создание некоторых структур для дальнейшего анализа либо информации о приложении, предназначенной для пользователей [21, 22]. Метод, предложенный в данной работе, выдаёт в качестве основного результата бинарную оценку вредоносности ана-

лизируемого приложения и список подобных ему вредоносных моделей. Помимо основного результата, предоставляются данные о найденных схожих цепочках API-вызовов, которые могут быть очень полезны для аналитиков мобильных приложений.

Поскольку ресурсы мобильных устройств весьма ограничены, многие схемы анализа приложений не могут быть запущены непосредственно на них. Предложены эффективные схемы организации анализа, уменьшающие нагрузку на мобильные устройства [23, 24]. В данной работе вопрос внедрения не рассматривается подробно, основным вариантом предполагается использование в качестве одной из автоматических проверок магазинами мобильных приложений и антивирусными приложениями.

2. Построение моделей

Анализ приложений заключается в построении моделей анализируемого приложения и их сравнении с предварительно построенными моделями ВП. Используется три вида моделей, каждый из которых отражает поведение приложения на определённой степени детализации. Самый общий вид — это модель привилегий. Модели API-вызовов Android Framework и цепочек API-вызовов являются последовательными детализациями первой модели.

2.1. Модель привилегий

Операционная система Android использует систему привилегий (permissions) для разграничения доступа к программным и аппаратным ресурсам устройства [25]. Все привилегии, необходимые приложению для корректной работы, должны быть объявлены в манифесте приложения (файле AndroidManifest.xml), который содержится в архиве apk-файла приложения. На рис. 1 приведён фрагмент манифеста приложения с описанием запрашиваемых привилегий.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest android:versionCode="10023" android:versionName="1.0.2"
  package="com.wia.ucgepcdvlsl"
  xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
</manifest>
```

Рис. 1. Запрашиваемые привилегии в файле AndroidManifest.xml

Запрашиваемые привилегии могут относиться как к системным ресурсам, так и к ресурсам сторонних приложений, установленных на устройство. Проанализировав привилегии, запрашиваемые вредоносными приложениями, мы составили список из 101 системной привилегии (т. е. привилегии, относящейся к ресурсам фреймворка). Поскольку целью анализа является определение сходства с какими-либо ВП, мы не расширяли список системными привилегиями, запрашиваемыми приложениями из коллекции легитимных.

Моделью привилегий для приложения является вектор $v \in \{0,1\}^{101}$, каждый из компонентов которого соответствует одной из привилегий в составленном списке (привилегии упорядочены в некотором заданном порядке):

$$v_p = \begin{cases} 1, \text{ если привилегия } p \text{ запрашивается,} \\ 0 \text{ в противном случае.} \end{cases}$$

2.2. Модель API-вызовов Android Framework

Более точное представление о поведении приложения можно получить, исходя из API-вызовов функций фреймворка. Для построения таких моделей необходима декомпиляция приложения, которая осуществляется с помощью средства Androguard [26].

В [27] проанализирована система привилегий Android и для каждой из системных привилегий составлен список API-вызовов, требующих её наличия у приложения. Будем называть такие API-вызовы защищёнными. Пользуясь этими результатами, мы проанализировали API-вызовы, используемые приложениями из вредоносной коллекции, и составили список тех из них, которые являются защищёнными. Поскольку некоторые ВП не использовали ни одного вызова из списка защищённых, мы расширили список вызовов некоторыми незащищёнными вызовами так, чтобы у каждого из ВП использовалось по крайней мере 20 вызовов из итогового списка. Всего в списке получилось 384 API-вызова.

Моделью API-вызовов Android Framework для приложения является вектор $v \in \in \{0,1\}^{384}$, каждый из компонентов которого соответствует одному из API-вызовов в списке (API-вызовы упорядочены в некотором заданном порядке):

$$v_f = \begin{cases} 1, \text{ если API-вызов } f \text{ используется,} \\ 0 \text{ в противном случае.} \end{cases}$$

2.3. Модель цепочек API-вызовов

Модель цепочек API-вызовов является наиболее подробной моделью, используемой в предлагаемом анализе. Она представляет собой список последовательностей (цепочек) API-вызовов, в которых вызовы упорядочены в порядке их возможного следования при запуске приложения. Каждая цепочка соответствует *точке входа* в приложение — функции, вызываемой фреймворком при некотором событии, сгенерированном пользователем или системой. Простейшим примером точки входа может служить метод onCreate в стартовой активности (Activity) приложения. Модели цепочек API-вызовов также строятся по декомпилированному представлению приложения.

Построение модели

Нахождение возможных точек входа в приложение

Для нахождения возможных точек входа в приложение используется алгоритм, описанный в [28]. Он заключается в следующем.

1) Строится первичный набор точек входа: методы основных компонентов приложения (Activity, Service, Broadcast Receiver, Content Provider), которые вызываются фреймворком.
2) По заданным точкам входа строится граф достижимых функциональных вызовов.
3) Все методы, перегружающие методы классов Android Framework и недостижимые из заданных точек входа, добавляются в точки входа, если в графе достижимых функциональных вызовов содержится конструктор их класса. Шаги 2 и 3 повторяются до тех пор, пока на очередной итерации множество точек входа не останется неизменным.

Построение графа функциональных вызовов для каждой точки входа

Выделяя вершины, соответствующие точкам входа в общем графе достижимых функциональных вызовов, а также все вершины, достижимые из этих вершин, и рёбра, инцидентные им, получаем графы функциональных вызовов для каждой из точек входа. В полученных графах для любой начальной вершины рёбра, инцидентные ей, упорядочены в порядке следования функциональных вызовов внутри метода, соответствующего данной начальной вершине.

Формирование цепочек API-вызовов

API-вызовы Android Framework являются листьями в построенных графах функциональных вызовов. Кроме них, интерес для анализа также представляют API-вызовы библиотек, вкомпилированных в приложение. Потомки таких вершин исключаются из обхода, поскольку могут привести к совпадениям в цепочках для приложений, совершенно не схожих по функциональности, а лишь использующих одинаковые библиотеки. Таким образом, для каждого функционального графа приложения в порядке обхода в глубину формируется цепочка из API-вызовов Android Framework и библиотечных API-вызовов.

На рис. 2 приведён пример графа функциональных вызовов для точки входа onCreate одного из сервисов приложения. Заданный порядок обхода смежных вершин на рисунке соответствует обходу слева направо и сверху вниз. Соответствующая графу цепочка API-вызовов:

Landroid/app/Service;->onCreate,
Landroid/os/PowerManager;->newWakeLock,
Landroid/os/PowerManager$WakeLock;->acquire,
Ljava/lang/Object;-><init>,
Landroid/database/sqlite/SQLiteOpenHelper;-><init>,
Landroid/os/Handler;->postDelayed

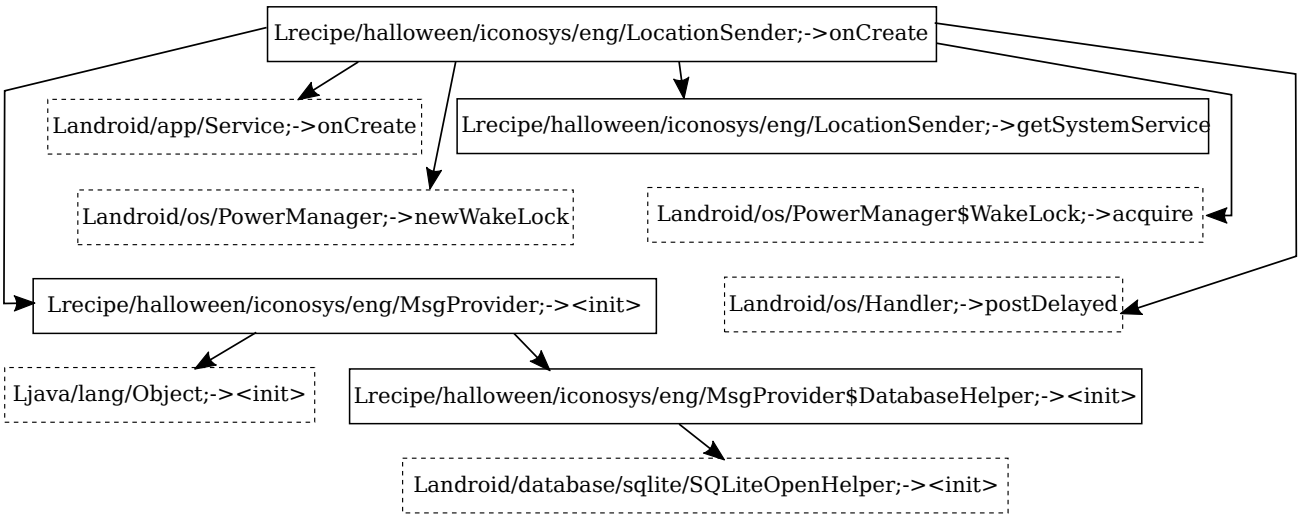

Рис. 2. Граф функциональных вызовов. Вершины, соответствующие API-вызовам, выделены пунктиром

В моделях цепочек API-вызовов для ВП содержатся и легитимные цепочки, которые могут быть встречены в невредоносных приложениях. Однако более длинные совпадения даже таких цепочек уже подозрительны и встречаются довольно редко. Больший интерес для анализа представляют цепочки моделей, содержащие ключевую

вредоносную функциональность модели. Приведём подцепочку такой цепочки в модели троянского приложения Mobiletx, которое собирает пользовательские данные (номер IMSI) и отправляет их в SMS-сообщении злоумышленникам:

Landroid/content/Context;->getSystemService,
Landroid/telephony/TelephonyManager;-> getSubscriberId,
Ljava/lang/StringBuilder;->append,
Ljava/lang/StringBuilder;->toString,
Landroid/telephony/SmsManager;->sendTextMessage

3. Анализ приложений

Анализ мобильных приложений состоит из построения моделей анализируемого приложения и сравнения их с предварительно построенными моделями ВП. Построение моделей приложения описано в п. 2. Сравнение полученных моделей с моделями ВП осуществляется в три этапа, на каждом из которых сравниваются между собой соответствующие модели. На первом этапе осуществляется сравнение с моделями привилегий всех ВП. Результатом является список ВП, модели привилегий которых оказались схожими с моделью привилегий анализируемого приложения, будем называть их *подобными по привилегиям приложениями*. На втором этапе сравнивается модель API-вызовов Android Framework с аналогичными моделями ВП, определённых как подобные по привилегиям на предыдущем шаге. В результате получается список ВП, подобных анализируемому по API-вызовам Android Framework. И наконец, на третьем этапе сравнивается модель цепочек API-вызовов с аналогичными моделями ВП, определённых как подобные на предыдущем этапе. Итоговым результатом анализа является список ВП, схожих с исследуемым по цепочкам API-вызовов. Соответственно анализируемое приложение считается вредоносным, если найдено хотя бы одно ВП, схожее с ним по цепочкам API-вызовов.

Вложенность результатов анализа обеспечивается за счёт вложенности используемых моделей. Действительно, приложения, имеющие схожие цепочки API-вызовов, имеют и схожие множества API-вызовов Android Framework. А так как возможность использования в приложении большинства API-вызовов Android Framework, рассматриваемых при построении этой модели, напрямую зависит от наличия соответствующей привилегии у приложения, то приложения, подобные по API-вызовам, в большинстве своем подобны и по привилегиям. Необходимо отметить, что результаты описываемого трёхэтапного анализа для большинства приложений будут совпадать с результатами анализа при использовании сравнения только модели цепочек API-вызовов с аналогичными моделями всех вредоносных приложений. Однако такой анализ был бы существенно более затратным по времени. К тому же результаты, полученные на первых двух этапах сравнения, могут представлять интерес для аналитиков мобильных приложений.

3.1. Сравнение моделей привилегий

Модели привилегий для приложений представляют собой двоичные векторы длины 101, значения которых соответствуют привилегиям из выбранного нами множества. Обозначим это множество привилегий как P. Введём веса для привилегий: обозначим вес привилегии p как w_p. Будем использовать большие веса для выделения наиболее опасных привилегий, таких, как android.permission.SEND_SMS; для большинства привилегий значение w_p равно 1. Более высокие веса опасных привилегий повышают значение функции сходства для анализируемых приложений, запрашивающих такие привилегии. Это не всегда соответствует большему сходству между сравниваемыми

приложениями, но, как правило, указывает на подозрительность приложения и необходимость его дальнейшего детализированного анализа.

Для сравнения модели привилегий v анализируемого приложения и модели привилегий u некоторого ВП вычисляется значение функции их сходства:

$$S_P(v,u) = \begin{cases} \dfrac{\sum\limits_{p\in P} w_p \cdot 1_{v_p=u_p \& u_p=1}}{\sum\limits_{p\in P} 1_{u_p=1}}, \text{ если } \sum\limits_{p\in P} 1_{u_p=1} \neq 0, \\ 1 \text{ в противном случае.} \end{cases} \quad (1)$$

Приложения считаются подобными по привилегиям, если $S_P(v,u)$ превышает наперед заданный порог $\text{Threshold}_{p_{\text{sim}}}$. В соответствии с (1) мы считаем все приложения подобными по привилегиям приложениям, не запрашивающим никаких системных привилегий. В качестве примера ВП, не запрашивающих никаких системных привилегий, можно привести ВП, эксплуатирующие уязвимости операционной системы для получения системных привилегий (root exploits).

3.2. Сравнение моделей API-вызовов Android Framework

Сравнение моделей API-вызовов Android Framework проводится аналогично сравнению моделей привилегий. Обозначим множество выбранных API-вызовов Android Framework как A. Для сравнения модели API-вызовов Android Framework v анализируемого приложения и модели API-вызовов Android Framework u некоторого ВП вычисляется значение функции их сходства:

$$S_{\text{API}}(v,u) = \frac{\sum\limits_{f\in A} 1_{v_f=u_f \& u_f=1}}{\sum\limits_{f\in A} 1_{u_f=1}}.$$

Приложения считаются подобными по API-вызовам Android Framework, если $S_{\text{API}}(v,u)$ превышает наперёд заданный порог $\text{Threshold}_{\text{API}_{\text{sim}}}$.

3.3. Сравнение моделей цепочек API-вызовов

Сравнение моделей цепочек API-вызовов основано на поиске наибольшей общей подпоследовательности между парами цепочек. Обозначим множество цепочек в модели приложения u за $\text{chains}(u)$. Для каждой пары цепочек c_v и c_u считается значение $\text{lcs}(c_v, c_u)$, равное количеству элементов в цепочке, являющейся их наибольшей общей подпоследовательностью. Саму наибольшую общую подпоследовательность для c_v и c_u обозначим $\text{cs}(c_v, c_u)$. Будем считать, что цепочки c_v и c_u имеют *общую подцепочку*, если их наибольшая общая подпоследовательность достаточно длинна, то есть относительная величина $\dfrac{\text{lcs}(c_v,c_u)}{|c_u|}$ превосходит порог $\text{Threshold}_{\text{common}}$ и абсолютное значение $\text{lcs}(c_v, c_u)$ не меньше $\text{Threshold}_{\text{common}_{\text{abs}}}$ API-вызовов.

Цепочки в модели анализируемого приложения рассматриваются в порядке уменьшения длины. Для каждой из цепочек c_v в модели анализируемого приложения, имеющих общие подцепочки с цепочками в модели ВП, выбираем цепочку $c_{u_{\text{similar}}}$, для которой общая подцепочка имеет наибольшую длину; в случае, если она не единственна, рассматриваются все такие цепочки. Общая подцепочка разбивается цепочкой c_v на *компоненты* — слитные участки. Если общих подцепочек наибольшей длины несколько, то из них выбирается та, которая разбивается на минимальное количество компонентов. На рис. 3 цепочки изображены в виде последовательностей символов, верхняя соответствует одной из цепочек анализируемого приложения, нижняя — одна из

цепочек ВП. Градиентной заливкой выделена их общая подцепочка с минимальным количеством компонентов, в данном случае их два.

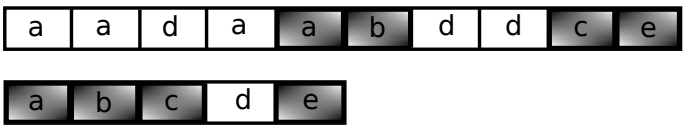

Рис. 3. Разбиение цепочки на компоненты

Обозначим количество компонентов, на которые разбивается общая подцепочка цепочек c_u и c_v как $\text{components}(c_v, c_u)$. Чем меньше компонентов в разбиении общей подцепочки, тем больше сходство цепочки c_v с цепочкой c_u, при этом слишком строгое ограничение на количество компонентов может привести к тому, что анализатором будут игнорироваться схожие цепочки даже с минимальными изменениями (например, добавление API-вызовов, не связанных с основной функциональностью цепочки). Поэтому в анализе мы учитывали только такие цепочки $c_{u_{\text{match}}}$ из $c_{u_{\text{similar}}}$, для которых выполнено следующее соотношение:

$$\text{components}(c_v, c_{u_{\text{match}}}) < \left\lceil \frac{1}{3} \operatorname{lcs}(c_v, c_{u_{\text{match}}}) \right\rceil. \tag{2}$$

Данное соотношение означает, что в общей подцепочке должен быть слитный участок из как минимум трёх звеньев (3 — величина, обратная коэффициенту в правой части соотношения). Если соотношение (2) выполнено для нескольких цепочек $c_{u_{\text{similar}}}$ из модели ВП, из них выбирается цепочка, содержащая защищённые API-вызовы (если таких нет, то любая цепочка). Таким образом, для каждой из цепочек c_v может быть выбрано не более одной цепочки в качестве наилучшего совпадения $c_{u_{\text{match}}}$ среди цепочек в модели ВП. Соответственно цепочки $\text{cs}(c_v, c_{u_{\text{match}}})$ формируют множество общих цепочек для сравниваемых моделей. Результатом сравнения являются четыре значения:

1) a — количество общих цепочек;
2) b — суммарная длина общих цепочек;
3) c — количество *длинных* общих цепочек (длинной считается цепочка с количеством элементов не меньше заданного порога $\text{Threshold}_{\text{long}_{\text{chain}}}$);
4) d — количество общих цепочек, содержащих защищённые API-вызовы.

Чем выше каждое из этих значений, тем больше сходство сравниваемых моделей. Мы считаем сравниваемые приложения подобными по цепочкам API-вызовов, если выполнено хотя бы одно из следующих условий:

— $a \geqslant \text{Threshold}_{\text{amount}}$ и $b \geqslant \text{Threshold}_{\text{length}}$ — наличие нескольких общих подцепочек с достаточно большой суммарной длиной;
— $c \geqslant 2$ — у приложений есть как минимум две длинные общие подцепочки;
— $c \geqslant 1$ и $d \geqslant 1$ — у приложений есть как минимум она длинная общая подцепочка и как минимум одна общая подцепочка, включающая защищённые API-вызовы;
— $d \geqslant 1$ и $b \geqslant \text{Threshold}_{\text{length}}$ — общие подцепочки приложений имеют достаточно большую суммарную длину и как минимум одна из них содержит защищённые API-вызовы;
— $\dfrac{a}{|\text{chains}(u)|} \geqslant 0{,}95$ и $\dfrac{b}{\sum\limits_{c \in \text{chains}(u)} \text{length}(c)} \geqslant 0{,}95$.

Последнее условие использовалось для обнаружения сходства с моделями, в которых количество цепочек и/или их суммарная длина невелики. Кроме этого, мы проверяем цепочки анализируемых приложений на полное совпадение независимо от их длины, считая приложения с одинаковыми цепочками подобными по цепочкам API-вызовов. Это сделано для обеспечения свойства рефлексивности отношения моделей, другими словами, каждое приложение подобно по цепочкам API-вызовов самому себе.

В приведённых условиях $|\text{chains}(u)|$ — количество цепочек API-вызовов в модели приложения u; $\text{length}(c)$ — длина (количество API-вызовов) цепочки c.

4. Подбор пороговых значений

Все пороговые значения, а также коэффициент в правой части соотношения (2) определены экспериментальным путём на части вредоносных и легитимных приложений из коллекций, подробно описанных в п. 5. В табл. 1 приведены пороговые значения, использованные в экспериментах. Эти значения могут быть перенастроены пользователями программного средства самостоятельно, хотя их изменение в ту или иную сторону может привести к изменению полноты обнаружения и количества ложных срабатываний (увеличение первого приводит к увеличению второго, и наоборот).

Таблица 1
Используемые пороговые значения

Наименование	Значение
$\text{Threshold}_{p_{\text{sim}}}$	0,9
$\text{Threshold}_{\text{API}_{\text{sim}}}$	0,7
$\text{Threshold}_{\text{common}}$	0,85
$\text{Threshold}_{\text{common}_{\text{abs}}}$	3
$\text{Threshold}_{\text{long}_{\text{chain}}}$	30
$\text{Threshold}_{\text{amount}}$	7
$\text{Threshold}_{\text{length}}$	50

Поскольку предложенный метод зависит от большого числа пороговых значений, практически невозможно исследовать зависимость его результатов от всех заданных пороговых значений одновременно. Поэтому мы изменяли каждое из пороговых значений по очереди при фиксированных остальных и выбирали значение, дающее наилучшие результаты.

Пороговые значения $\text{Threshold}_{p_{\text{sim}}}$ и $\text{Threshold}_{\text{API}_{\text{sim}}}$ должны быть выбраны так, чтобы итоговая доля ошибок 1-го рода была как можно ближе к нулю. При этом нужно минимизировать число подобных по привилегиям и API-вызовам Android Framework моделей, с которыми проводится сравнение на последнем (наиболее затратном по времени) шаге анализа. При подборе этих пороговых значений была использована половина коллекции *Drebin* для формирования моделей и ещё 300 приложений из этой коллекции для оценки результата работы предложенного метода (все эти приложения выбирались случайным образом). Подбирая $\text{Threshold}_{p_{\text{sim}}}$, мы пропустили второй шаг анализа, сразу сравнивая модели цепочек API-вызовов для приложений, подобных по привилегиям. Установлено, что доля ошибок 1-го рода одинакова для пороговых значений $\text{Threshold}_{p_{\text{sim}}} \in [0{,}1, 0{,}9]$ и равна 4,6 %; при значении $\text{Threshold}_{p_{\text{sim}}} = 0{,}95$ доля ошибок 1-го рода повышается до 5 %. В то же время среднее число подобных по привилегиям моделей падает с 1929 ($\text{Threshold}_{p_{\text{sim}}} = 0{,}1$) до 412 ($\text{Threshold}_{p_{\text{sim}}} = 0{,}9$). Поэтому для дальнейших экспериментов было выбрано пороговое значение $\text{Threshold}_{p_{\text{sim}}} = 0{,}9$.

В процессе подбора $\text{Threshold}_{\text{API}_{\text{sim}}}$ использован тот же критерий: максимально возможное сокращение числа моделей, подобных по API-вызовам Android Framework, при минимально возможной итоговой доле ошибок 1-го рода. На графике рис. 4 видно наилучшую точку с долей ошибок 1-го рода, равной 4,6 %, и средним числом подобных по API-вызовам Android Framework моделей, равным 142. Эта точка соответствует значению $\text{Threshold}_{\text{API}_{\text{sim}}} = 0{,}7$.

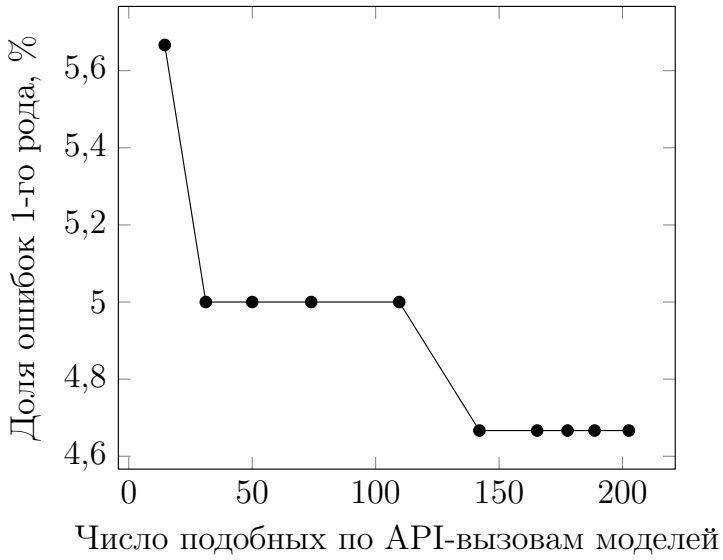

Рис. 4. Зависимость ошибки 1-го рода и среднего числа подобных по API-вызовам Android Framework моделей от $\text{Threshold}_{\text{API}_{\text{sim}}}$

Пороговые значения для сравнения цепочек API-вызовов были получены в процессе ручного анализа при сравнении некоторого числа приложений, относящихся к одному и тому же семейству ВП. Эти значения были проверены на части коллекции легитимных приложений, чтобы убедиться в их эффективности относительно ложных срабатываний.

5. Эксперименты

Далее описаны эксперименты, проведенные с использованием реализованного метода. Сначала метод был применен для решения задачи бинарной классификации, оценены доли ошибок 1-го и 2-го рода. Затем были исследованы коллекции ВП, показана способность метода обнаруживать похожие приложения внутри семейств ВП. Результаты этого же исследования были применены для выбора множества моделей ВП с минимальным количеством элементов таким образом, чтобы они покрывали все ВП в использованных коллекциях. Измерена доля ложных срабатываний метода на большой коллекции легитимных приложений.

5.1. Бинарная классификация

Для экспериментальной оценки метода были использованы две коллекции ВП: коллекция Drebin [16] и коллекция ISCX, предоставленная центром ISCX университа в Нью-Брансуик [29], в частности, её часть Android botnet. Общее число приложений в обеих коллекциях составляет 7489, однако с использованием предложенного метода не удалось построить модели 40 приложений, ещё для части приложений были построены нефункциональные модели, не содержащие ни одной цепочки API-вызовов. Более подробная информация об этом представлена далее. Таким образом, общее количество

проанализированных ВП составило 7449. В качестве выборки легитимных приложений использовано 3000 приложений, загруженных из Google Play. Предложенным методом не удалось построить модели семи из этих приложений. Сделана случайная выборка 2/3 ВП для формирования моделей ВП и протестирован метод на остальной части коллекции ВП, а также на всех легитимных приложениях. Этот эксперимент был проведён трижды, каждый раз со случайным разбиением общей коллекции ВП на модели и приложения для тестирования.

Предложенный метод показал в среднем долю ошибок 1-го рода менее 10 % и долю ложных срабатываний менее 2 % (табл. 2). Результаты промежуточных шагов анализа не предоставляются, поскольку они не существенны для оценки итоговой полноты покрытия метода, а также ложных срабатываний.

Таблица 2

Результаты анализа приложений (бинарная классификация)

Число моделей ВП	Срабатывания на вредоносной коллекции	Срабатывания на легитимной коллекции	Доля ошибок 1-го рода, %	Доля ошибок 2-го рода, %
4787	2254 / 2483	63 / 2993	9,3	2,1
4787	2240 / 2483	51 / 2993	9,7	1,7
4787	2262 / 2483	46 / 2993	9,1	1,5

5.2. Анализ коллекций вредоносных приложений

Предложенный метод был использован для исследования структуры двух вышеупомянутых коллекций ВП: Drebin и ISCX. Под исследованием здесь подразумевается обнаружение сходства и зависимостей между приложениями в коллекции. Основной целью было получение множества приложений из каждой коллекции, такого, что будучи взятым для создания моделей ВП, оно покроет всю оставшуюся часть коллекции, то есть каждое приложение из вредоносной коллекции будет подобным по цепочкам API-вызовов одной из моделей ВП.

Коллекция Drebin

Коллекция состоит из 5560 приложений, относящихся к 179 семействам. Все вредоносные образцы были собраны в период с августа 2010 по октябрь 2012 г. Из этих приложений 10 не было декомпилировано используемой библиотекой Androguard из-за программных ошибок в библиотеке или в приложениях (6 образцов не были распознаны как zip-архив утилитой file). Для 28 из оставшихся 5550 приложений не удалось построить ни одной цепочки API-вызовов. Эти приложения проанализированы вручную и выявлено, что 25 из них не запускаются эмулятором Android в среде динамического анализа Droidbox [30]. В ходе статического анализа в большинстве из них был обнаружен некорректный smali-код: он либо не содержал никаких инструкций методов, либо имена, объявленные в манифесте, не соответствовали действительным именам классов в приложении. Для трёх приложений не удалось построить цепочки, поскольку имена используемых в них пакетов совпадали с именами известных библиотечных пакетов, а такой код не учитывается при построении моделей. Про этот недостаток метода, а также способы борьбы с ним подробнее описано далее в п. 5.4 и 7. Таким образом, мы исключили 38 приложений из дальнейшего анализа.

На основе обнаруженного подобия по цепочкам API-вызовов среди приложений можно выделить 422 приложения так, что они покроют 5131 приложение в коллекции

(включая эти 422). Для 381 приложения не было найдено ни одного подобного по цепочкам приложения в коллекции. Далее были проанализированы кластеры подобных приложений, сформированные этими 422 приложениями. Каждому кластеру поставлена в соответствие метка семейства ВП (коллекции изначально были предоставлены размеченными) таким образом, чтобы метка кластера соответствовала меткам большинства приложений, образующих его. Затем было посчитано число приложений, у которых метка семейства не соответствовала метке кластера: 3,1 % (159 из 5131) приложений были неправильно кластеризованы. Таким образом, для формирования требуемого множества моделей ВП использовано $422+(5560-38-5131) = 813$ приложений из коллекции Drebin.

Кроме кластеризации, был проведён базовый анализ коллекции с целью получить представление об используемых приложениями практиках программирования и техниках обфускации. Результаты показывают, что 37 % приложений в коллекциях содержат встроенную рекламу; приложения преимущественно небольшого размера — 50 % приложений содержат меньше 50 классов. Анализ наиболее длинных цепочек API-вызовов выявил, что для 50 % приложений их длина меньше 70, в то время как для 92 % приложений длина таких цепочек меньше 200. Рефлексию используют 60,5 % приложений; 38 % используют API, связанные с выполнением нативного кода; 32 % используют криптографические API-вызовы (учитывался только пакет javax.crypto). Хотя предложенный метод не приспособлен для обнаружения использования нативного кода или рефлексии, с его помощью тем не менее можно обнаружить сходство между такими приложениями, основываясь на встреченных в цепочках API-вызовах рефлексии/исполнения нативного кода (Ljava/lang/Class;->forName, Ljava/lang/reflect/Method;->invoke, Ljava/lang/Runtime;->exec ...) и других частях приложений, не задействующих подобные техники. Мы также проанализировали 200 случайно выбранных приложений, чтобы получить представление об использовании обфускации имён в коллекции: 44 % не используют обфускации имён вовсе; в 51,5 % приложений обфусцирована часть названий классов, большинство из которых находятся внутри статически скомпилированных вместе с приложением библиотек; и только 4,5 % приложений используют обфускацию всех имён. В целом, ручной анализ приложений из коллекции выявил, что большинство приложений может быть достаточно легко подвергнуто обратной разработке и не использует сложной обфускации.

Коллекция ISCX

Коллекция ISCX была проанализирована по аналогии с коллекцией Drebin. Она состоит из 1929 приложений, принадлежащих к 14 семействам ВП, и содержит более новые образцы ВП, собранные с 2010 по 2014 г. Из них 30 приложений не были декомпилированы библиотекой Androguard, но в данном случае проблема была исключительно на стороне программного кода библиотеки. Предложенным методом не удалось построить цепочки API-вызовов для 66 образцов, из которых 19 содержали пакеты, имена которых совпадали с именами известных библиотек. Анализ оставшейся части приложений, для которых не удалось построить модели, выявил ещё один случай некорректных приложений: в некоторых приложениях все имена методов (включая такие методы, как onCreate, onStart) были модифицированы добавлением суффикса «abc123», что вызывало ошибки при попытке запуска компонентов. Суммарно из дальнейшего анализа были исключены 96 приложений коллекции.

На оставшейся части коллекции с помощью предлагаемого метода было сформировано 120 кластеров (состоящих из по меньшей мере двух приложений каждый), покрывающих 1723 приложения в коллекции. Другие приложения в коллекции не были кластеризованы и для 114 приложений не было обнаружено ни одного подобного по цепочкам API-вызовов приложения в коллекции. Из них 3,9 % (68 из 1723) были кластеризованы неверно, т. е. метка семейства ВП для них не соответствовала метке большинства приложений в кластере. Таким образом, мы сформировали требуемое множество моделей из $120 + (1929 - 96 - 1723) = 230$ приложений в коллекции.

Базовый анализ коллекции ICSX выявил, что приложения, представленные в ней, в целом крупнее и обладают более сложной структурой по сравнению с приложениями из коллекции Drebin: только 30 % приложений содержат менее 50 классов, почти 50 % содержат более 100 классов; 18 % приложений включают встроенную рекламу, 76 % используют рефлексию, 62 % содержат нативный код и более 53 % используют криптографические API-вызовы. Вручную проанализировано 100 приложений из коллекции, выбранных случайным образом, и выявлено, что 29 % не используют обфускации имён, 70 % обфусцируют имена части классов, относящихся к рекламным пакетам или другим библиотекам, и только в одном приложении все имена обфусцированы.

5.3. Анализ крупной коллекции легитимных приложений

Предложенный метод анализа мобильных приложений был протестирован на коллекции из 43819 мобильных приложений, загруженных из Google Play в период с 2013 по 2016 г., с использованием 850 приложений из коллекций Drebin и ISCX в качестве моделей ВП. Приложения для создания моделей получены объединением моделей для каждой коллекции с исключением дублирующихся и похожих приложений — коллекции частично пересекаются. В этом эксперименте пороговое значение $\text{Threshold}_{p_{\text{sim}}}$ было установлено в 0,7 вместо 0,9, чтобы выявить больше ложных срабатываний, вызванных именно подобием по цепочкам API-вызовов. Табл. 3 содержит результаты проведённого анализа.

Таблица 3

Результаты анализа коллекции легитимных приложений

Шаг анализа	Количество срабатываний	Доля ошибок 2-го рода, %
Привилегии (1)	43357 / 43819	98,9
API-вызовы Android Framework (2)	39451 / 43819	90,0
Цепочки API-вызовов (3)	1388 / 43819	3,2

Таким образом, доля ложных срабатываний (ошибок 2-го рода) составила 3,2 %. Эта оценка немного выше полученной в эксперименте с бинарной классификацией, поскольку использовано другое значение $\text{Threshold}_{p_{\text{sim}}}$. Тем не менее она лучше отражает устойчивость предложенного метода к ложным срабатываниям, поскольку основной частью анализа является сопоставление цепочек API-вызовов. Результаты промежуточных шагов (Привилегии (1) и API-вызовы Android Framework (2)) показывают количество анализированных приложений, для которых найдены подобные модели на соответствующем шаге. Как можно заметить из этих промежуточных результатов, только множества запрашиваемых привилегий и API-вызовов недостаточно для точного определения поведения приложения и характеристики его как вредоносного или легитимного. Первый и второй шаги анализа используются в основном для сокраще-

ния числа моделей ВП, используемых в сравнении на последнем и наиболее затратном по времени шаге анализа.

5.4. Анализ ошибок 1-го и 2-го рода

Ошибки 1-го рода в эксперименте с бинарной классификацией в основном связаны с отсутствием необходимых приложений в множестве моделей из-за их случайного выбора. Для обеспечения наилучшей полноты обнаружения предложенного метода необходим тщательный отбор моделей ВП.

Часть легитимных приложений из Google Play, для которых анализатором сгенерированы ложные срабатывания, была проанализирована вручную. Большинство ложных срабатываний возникло из-за наличия общего кода в пакетах, имена которых были изменены при обфускации приложений, а потому не были восприняты анализатором как библиотечный код. Как отмечено в п. 2, мы учитывали API-вызовы известных библиотек в построении цепочек, но исключали внутреннюю часть кода библиотек из дальнейшего анализа. Для реализации такого отсеивания был сформирован список известных библиотечных пакетов и отфильтровыван по именам. В программном средстве была реализована также фильтрация библиотечных пакетов по хеш-суммам их методов. Однако подходящий фильтр, содержащий все необходимые хеш-суммы, сформировать не удалось, это нетривиальная задача, требующая анализа большого числа легитимных приложений. Тем не менее у пользователей предложенного метода есть возможность задействовать свои фильтры Блума с хеш-суммами библиотечных методов и, таким образом, избавиться от этого недостатка.

Некоторые легитимные приложения имеют значительную часть общего кода с вредоносной моделью, в этих случаях, скорее всего, соответствующее ВП создавалось на основе ранней версии данного легитимного приложения (было переупаковано с добавлением функциональности). Ещё часть приложений содержала методы, схожие с некоторыми методами ВП, не включающими вредоносную активность (например, десериализация и сериализация XML-моделей).

6. Сравнение с adagio

Adagio — это программное средство, реализующее метод статического анализа Android-приложений, предложенный в [20]. Этот метод выбран для сравнения из-за сходства его основополагающих идей с предложенным методом, а также из-за доступности его исходного кода [31].

Adagio реализует алгоритм машинного обучения SVM, используя статические признаки приложений для формирования множества векторов признаков. Векторы признаков строятся по графам функциональных вызовов приложений по аналогии с моделями цепочек API-вызовов в предложенном методе. Однако графы функциональных вызовов интерпретируются по-другому: учитываются инструкции всех методов в графе вызовов, а также зависимости между ними для формирования множества хеш-сумм, впоследствии встраиваемого в пространство признаков. В предложенном методе учитываются API-вызовы Android Framework и известных библиотек, соответствующие параметрам только инструкций методов вида invoke*.

Мы провели трижды тот же эксперимент с бинарной классификацией на коллекции из 7449 вредоносных приложений из Drebin и ISCX и 3000 легитимных приложений, выделив 2/3 из них как обучающую выборку и 1/3 как проверочную. К сожалению, не удалось построить модели 509 вредоносных и 11 легитимных приложений. Табл. 4 содержит результаты эксперимента: число вредоносных и легитимных моделей немного разнится между экспериментами, поскольку 2/3 моделей для обучающей выборки

были выбраны случайным образом из объединения векторов признаков вредоносных и легитимных приложений.

Таблица 4

Результаты анализа приложений с использованием программного средства adagio

Вредоносные модели	Легитимные модели	Ошибки 1-го рода	Ошибки 2-го рода
4572	1983	45 / 2368 (1,9 %)	93 / 1006 (9,2 %)
4571	1984	51 / 2369 (2,2 %)	97 / 1005 (9,7 %)
4579	1976	57 / 2361 (2,4 %)	90 / 1013 (8,9 %)

Программное средство adagio с использованием конфигурационных значений по умолчанию показало среднюю долю ошибок 1-го рода 2,2 % и 2-го рода 9,3 %. Таким образом, предложенный метод имеет меньшую полноту покрытия со случайно выбранными вредоносными моделями, но в то же время является значительно более надёжным по ложным срабатываниям.

Касательно разъяснений классификации, для каждого предположительно вредоносного приложения adagio перечисляет методы в графе функциональных вызовов, которые внесли наибольший вклад в функцию потерь (desicion function). Таким образом, пояснение представляет собой список методов приложения, которые программное средство считает подозрительными (предположительно содержащими вредоносную активность), без объяснения, что в действительности является подозрительным в данных методах. Предложенный метод, напротив, даёт достаточно подробные разъяснения, представленные всеми совпавшими цепочками API-вызовов в анализируемом приложении и конкретной вредоносной модели.

Программное средство adagio способно анализировать приложения с гораздо более высокой скоростью по сравнению с реализацией предложенного метода. Время анализа составляет около 1 с для adagio, в то время как для предложенного метода оно составляет 2,3 с в случае ВП и около 23,8 с в случае легитимного приложения, поданного на вход. В то же время предложенный метод тратит меньше времени на генерацию моделей: 1,5 с по сравнению с 4,4 с для adagio в случае ВП и 11,2 с по сравнению с 20,5 с в случае легитимного приложения. Измерения производительности проводились на ПК с процессором Intel(R) Xeon(R) CPU E31225 @ 3.10GHz с использованием только одного ядра процессора.

7. Ограничения метода и возможные улучшения

Подводя итоги, нужно отметить, что предлагаемый метод проводит глубинный статический анализ приложений, давая при этом детальное объяснение результатов. Однако он требует тщательного выбора вредоносных моделей для поддержания высокого уровня обнаружения и приемлемой производительности, поскольку полнота покрытия метода существенно зависит от ВП, использованных для генерации моделей. На самом деле, это общая проблема для почти всех методов обнаружения ВП — они требуют постоянного переобучения/обновления множества моделей. В то же время множество моделей должно быть как можно более компактным, поскольку время, затрачиваемое на анализ, растёт линейно с ростом числа моделей.

Улучшение метода, которое можно предложить для снижения доли ложных срабатываний, — это тщательная фильтрация библиотечных пакетов, статически скомпилированных в приложении. Это может быть достигнуто с использованием фильтра по

хеш-суммам методов, однако такой фильтр должен быть получен на основе анализа большого числа легитимных приложений, чтобы включать в себя хеш-суммы всех известных библиотечных методов (кроме того, должны быть представлены все версии каждой библиотеки).

Возможно улучшение метода за счёт изменения алгоритмов построения цепочек API-вызовов или их сравнения. Например, в предлагаемом методе в графе функциональных вызовов приложения вершины, соответствующие методам классов, уникальны и в построенной цепочке API-вызовов соответствуют единственному участку, даже если вызываются из нескольких других методов. Повторение таких участков в цепочках для каждого из мест вызова существенно увеличит их длину (и время работы анализатора), но, возможно, увеличит и точность метода. Предложенный метод не учитывает взаимодействия между компонентами посредством передачи объектов класса Intent. Их учёт добавит в модели цепочек API-вызовов связи между цепочками, что также может увеличить точность метода. Для обнаружения обфусцированных образцов используемых вредоносных моделей метод может быть дополнен динамическим анализом, учитывающим фактические наблюдаемые при выполнении цепочки API-вызовов.

Заключение

В работе представлен метод анализа мобильных приложений для платформы Android, основанный на используемых в приложении системных привилегиях и API-вызовах. Результаты экспериментов на коллекциях вредоносных приложений показывают, что метод обнаруживает подобные приложения для 90–94 % приложений в них. Тестирование метода на коллекции приложений из Google Play выявило долю ложных срабатываний около 3,2 %. Метод может быть использован как часть многокомпонентного анализа на стороне магазинов мобильных приложений или аналитиками мобильных приложений как вспомогательное средство, предназначенное для сокращения времени, затрачиваемого на ручной анализ.

Анализ использованных коллекций вредоносных приложений показал, что приложения в них редко используют техники шифрования или обфускации для усложнения статического анализа. Это наблюдение противоположно наблюдаемому в семействе платформ «Wintel», и, вероятно, следует ожидать изменения ситуации в ближайшем будущем.

Представлены результаты сравнения предложенного метода с методом, показавшим свою эффективность в задаче обнаружения вредоносных Android-приложений и реализованным в программном средстве adagio. Предлагаемый метод имеет меньшую полноту покрытия при случайном выборе вредоносных приложений для генерации моделей, но показывает значительно лучшие результаты в отношении ошибок 2-го рода.

ЛИТЕРАТУРА

1. *Skovoroda A. and Gamayunov D.* Securing mobile devices: malware mitigation methods // J. Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications. 2015. No. 6(2). P. 78–97.
2. *Egele M., Kruegel C., Kirda E., and Vigna G.* PiOS: Detecting privacy leaks in iOS applications // Proc. 18th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 2011. `http://www.eurecom.fr/publication/3282`

3. *Enck W., Gilbert P., Chun B.-G., et al.* TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones // Proc. 9th USENIX Conf. Design and Implementation OSDI'10, Vancouver, BC, Canada. USENIX Association, 2010. P. 1–6.
4. *Shabtai A., Tenenboim-Chekina L., Mimran D., et al.* Mobile malware detection through analysis of deviations in application network behavior // Computers & Security. 2014. V. 43. P. 1–18.
5. *Kim H., Smith J., and Shin K. G.* Detecting energy-greedy anomalies and mobile malware variants // Proc. 6th Intern. Conf. Applications, and Services MobiSys'08, Breckenridge, CO, USA. ACM, 2008. P. 239–252.
6. *Grace M., Zhou Y., Zhang Q., et al.* RiskRanker: scalable and accurate zero-day android malware detection // Proc. 10th Intern. Conf. Applications, and Services MobiSys'12, Low Wood Bay, Lake District, UK. ACM, 2012. P. 281–294.
7. *Feng Y., Anand S., Dillig I., and Aiken A.* Apposcopy: Semantics-based detection of android malware through static analysis // Proc. 22nd ACM SIGSOFT Intern. Symp. FSE 2014, New York, NY, USA. ACM, 2014. P. 576–587.
8. *Tam K., Feizollah A., Anuar N. B., et al.* The evolution of android malware and android analysis techniques // ACM Comput. Surv. 2017. V. 49. Iss. 4. Article No. 76.
9. *Yang L., Ganapathy V., and Iftode L.* Enhancing mobile malware detection with social collaboration // Proc. 2011 IEEE Third Intern. Conf. Social Computing (SocialCom), Boston, MA. IEEE, 2011. P. 572–576.
10. *Liu L., Yan G., Zhang X., and Chen S.* VirusMeter: preventing your cellphone from spies // LNCS. 2009. V. 5758. P. 244–264.
11. *Hoffmann J., Neumann S., and Holz T.* Mobile malware detection based on energy fingerprints — a dead end? // LNCS. 2013. V. 8145. P. 348–368.
12. *Wong M. Y. and Lie D.* Intellidroid: a targeted input generator for the dynamic analysis of android malware // NDSS. 2016. http://dx.doi.org/10.14722/ndss.2016.23118
13. *Zhou Y., Wang Z., Zhou W., and Jiang X.* Hey, you, get off of my market: detecting malicious apps in official and alternative android markets // Proc. 19th Annual NDDS Symp., San Diego, CA, USA. The Internet Society, 2012. https://www.internetsociety.org/sites/default/files/07_5.pdf
14. *Aafer Y., Du W., and Yin H.* DroidAPIMiner: Mining API-level features for robust malware detection in android // Security and Privacy in Communication Networks / eds. T. Zia, A. Zomaya, V. Varadharajan, and M. Mao. Springer International Publishing, 2013. P. 86–103.
15. *Zhang M., Duan Y., Yin H., and Zhao Z.* Semantics-aware android malware classification using weighted contextual API dependency graphs // Proc. ACM SIGSAC Conf. CCS'14, New York, NY, USA. ACM, 2014. P. 1105–1116.
16. *Arp D., Spreitzenbarth M., Huebner M., et al.* Drebin: efficient and explainable detection of android malware in your pocket // Proc. NDSS Symp., San Diego, CA, USA. The Internet Society, 2014. https://www.internetsociety.org/sites/default/files/11_3_1.pdf
17. *Damshenas M., Dehghantanha A., Choo K., and Mahmud R.* M0Droid: an android behavioral-based malware detection model // J. Inform. Privacy Security. 2015. V. 11. Iss. 3. P. 141–157.
18. *Aresu M., Ariu D., Ahmadi M., et al.* Clustering android malware families by http traffic // Proc. MALCON'2015, Fajardo, Puerto Rico, USA, 2015. https://pralab.diee.unica.it/sites/default/files/MALCON_0.pdf
19. *Mariconti E., Onwuzurike L., Andriotis P., et al.* Mamadroid: Detecting android malware by building markov chains of behavioral models. 2016. https://arxiv.org/abs/1612.04433

20. *Gascon H., Yamaguchi F., Arp D., and Rieck K.* Structural detection of android malware using embedded call graphs // Proc. AISec'13, New York, NY, USA. ACM, 2013. P. 45–54.
21. *Yang Z., Yang M., Zhang Y., et al.* AppIntent: analyzing sensitive data transmission in android for privacy leakage detection // Proc. CCS'13, Berlin, Germany. ACM, 2013. P. 1043–1054.
22. *Rosen S., Qian Z., and Mao Z. M.* AppProfiler: a flexible method of exposing privacy-related behavior in android applications to end users // Proc. Third ACM Conf. CODASPY '13, San Antonio, Texas, USA. ACM, 2013. P. 221–232.
23. *Portokalidis G., Homburg P., Anagnostakis K., and Bos H.* Paranoid android: versatile protection for smartphones // Proc. 26th Annual Conf. ACSAC'10, Austin, Texas. ACM, 2010. P. 347–356.
24. *Burguera I., Zurutuza U., and Nadjm-Tehrani S.* Crowdroid: behavior-based malware detection system for android // Proc. 1st ACM Workshop SPSM '11, Chicago, Illinois, USA. ACM, 2011. P. 15–26.
25. `http://developer.android.com/intl/ru/guide/topics/security/permissions.html` — Permissions, 2017.
26. `https://github.com/androguard/androguard` — Androguard, 2017.
27. *Au K. W. Y., Zhou Y. F., Huang Z., and Lie D.* PScout: analyzing the android permission specification // Proc. ACM Conf. CCS'12, New York, NY, USA. ACM, 2012. P. 217–228.
28. *Lu L., Li Z., Wu Z., et al.* CHEX: statically vetting android apps for component hijacking vulnerabilities // Proc. ACM Conf. CCS'12, New York, NY, USA. ACM, 2012. P. 229–240.
29. `http://www.unb.ca/research/iscx/dataset/iscx-android-botnet-dataset.html` — Коллекция вредоносных приложений UNB ISCX Android Botnet, 2017.
30. `http://code.google.com/p/droidbox/` — DroidBox: Сэндбоксинг Android-приложений, 2017.
31. `https://github.com/hgascon/adagio` — Adagio, 2017.

REFERENCES

1. *Skovoroda A. and Gamayunov D.* Securing mobile devices: malware mitigation methods. J. Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 2015, no. 6(2), pp. 78–97.
2. *Egele M., Kruegel C., Kirda E., and Vigna G.* PiOS: Detecting privacy leaks in iOS applications. Proc. 18th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 2011. `http://www.eurecom.fr/publication/3282`
3. *Enck W., Gilbert P., Chun B.-G., et al.* TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. Proc. 9th USENIX Conf. Design and Implementation OSDI'10, Vancouver, BC, Canada. USENIX Association, 2010, pp. 1–6.
4. *Shabtai A., Tenenboim-Chekina L., Mimran D., et al.* Mobile malware detection through analysis of deviations in application network behavior. Computers & Security, 2014, vol. 43, pp. 1–18.
5. *Kim H., Smith J., and Shin K. G.* Detecting energy-greedy anomalies and mobile malware variants. Proc. 6th Intern. Conf. Applications, and Services MobiSys'08, Breckenridge, CO, USA. ACM, 2008, pp. 239–252.
6. *Grace M., Zhou Y., Zhang Q., et al.* RiskRanker: scalable and accurate zero-day android malware detection. Proc. 10th Intern. Conf. Applications, and Services MobiSys'12, Low Wood Bay, Lake District, UK. ACM, 2012, pp. 281–294.

7. *Feng Y., Anand S., Dillig I., and Aiken A.* Apposcopy: Semantics-based detection of android malware through static analysis. Proc. 22nd ACM SIGSOFT Intern. Symp. FSE 2014, New York, NY, USA. ACM, 2014, pp. 576–587.
8. *Tam K., Feizollah A., Anuar N. B., et al.* The evolution of android malware and android analysis techniques. ACM Comput. Surv., 2017, vol. 49, iss. 4, article no. 76.
9. *Yang L., Ganapathy V., and Iftode L.* Enhancing mobile malware detection with social collaboration. Proc. 2011 IEEE Third Intern. Conf. Social Computing (SocialCom), Boston, MA. IEEE, 2011, pp. 572–576.
10. *Liu L., Yan G., Zhang X., and Chen S.* VirusMeter: preventing your cellphone from spies. LNCS, 2009, vol. 5758, pp. 244–264.
11. *Hoffmann J., Neumann S., and Holz T.* Mobile malware detection based on energy fingerprints — a dead end? LNCS, 2013, vol. 8145, pp. 348–368.
12. *Wong M. Y. and Lie D.* Intellidroid: a targeted input generator for the dynamic analysis of android malware. NDSS, 2016. http://dx.doi.org/10.14722/ndss.2016.23118
13. *Zhou Y., Wang Z., Zhou W., and Jiang X.* Hey, you, get off of my market: detecting malicious apps in official and alternative android markets. Proc. 19th Annual NDDS Symp., San Diego, CA, USA. The Internet Society, 2012. https://www.internetsociety.org/sites/default/files/07_5.pdf
14. *Aafer Y., Du W., and Yin H.* DroidAPIMiner: Mining API-level features for robust malware detection in android. Security and Privacy in Communication Networks / eds. T. Zia, A. Zomaya, V. Varadharajan, and M. Mao. Springer International Publishing, 2013, pp. 86–103.
15. *Zhang M., Duan Y., Yin H., and Zhao Z.* Semantics-aware android malware classification using weighted contextual API dependency graphs. Proc. ACM SIGSAC Conf. CCS'14, New York, NY, USA. ACM, 2014, pp. 1105–1116.
16. *Arp D., Spreitzenbarth M., Huebner M., et al.* Drebin: efficient and explainable detection of android malware in your pocket. Proc. NDSS Symp., San Diego, California, USA. The Internet Society, 2014. https://www.internetsociety.org/sites/default/files/11_3_1.pdf
17. *Damshenas M., Dehghantanha A., Choo K., and Mahmud R.* M0Droid: an android behavioral-based malware detection model. J. Inform. Privacy Security, 2015, vol. 11, iss. 3, pp. 141–157.
18. *Aresu M., Ariu D., Ahmadi M., et al.* Clustering android malware families by http traffic. Proc. MALCON'2015, Fajardo, Puerto Rico, USA, 2015. https://pralab.diee.unica.it/sites/default/files/MALCON_0.pdf
19. *Mariconti E., Onwuzurike L., Andriotis P., et al.* Mamadroid: Detecting android malware by building markov chains of behavioral models. 2016. https://arxiv.org/abs/1612.04433
20. *Gascon H., Yamaguchi F., Arp D., and Rieck K.* Structural detection of android malware using embedded call graphs. Proc. AISec'13, New York, NY, USA. ACM, 2013, pp. 45–54.
21. *Yang Z., Yang M., Zhang Y., et al.* AppIntent: analyzing sensitive data transmission in android for privacy leakage detection. Proc. CCS'13, Berlin, Germany. ACM, 2013, pp. 1043–1054.
22. *Rosen S., Qian Z., and Mao Z. M.* AppProfiler: a flexible method of exposing privacy-related behavior in android applications to end users. Proc. Third ACM Conf. CODASPY '13, San Antonio, Texas, USA. ACM, 2013, pp. 221–232.
23. *Portokalidis G., Homburg P., Anagnostakis K., and Bos H.* Paranoid android: versatile protection for smartphones. Proc. 26th Annual Conf. ACSAC'10, Austin, Texas. ACM, 2010, pp. 347–356.

24. *Burguera I., Zurutuza U., and Nadjm-Tehrani S.* Crowdroid: behavior-based malware detection system for android. Proc. 1st ACM Workshop SPSM '11, Chicago, Illinois, USA. ACM, 2011, pp. 15–26.
25. `http://developer.android.com/intl/ru/guide/topics/security/permissions.html` — Permissions, 2017.
26. `https://github.com/androguard/androguard` — Androguard, 2017.
27. *Au K. W. Y., Zhou Y. F., Huang Z., and Lie D.* PScout: analyzing the android permission specification. Proc. ACM Conf. CCS'12, New York, NY, USA. ACM, 2012, pp. 217–228.
28. *Lu L., Li Z., Wu Z., et al.* CHEX: statically vetting android apps for component hijacking vulnerabilities. Proc. ACM Conf. CCS'12, New York, NY, USA. ACM, 2012, pp. 229–240.
29. `http://www.unb.ca/research/iscx/dataset/iscx-android-botnet-dataset.html` — UNB ISCX Android Botnet, 2017.
30. `http://code.google.com/p/droidbox/` — DroidBox, 2017.
31. `https://github.com/hgascon/adagio` — Adagio, 2017.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

УДК 510.52

О ГЕНЕРИЧЕСКОЙ NP-ПОЛНОТЕ ПРОБЛЕМЫ ВЫПОЛНИМОСТИ БУЛЕВЫХ ФОРМУЛ[1]

А. Н. Рыбалов

Омский государственный технический университет, г. Омск, Россия

Вводится понятие полиномиальной генерической сводимости алгоритмических проблем, которое сохраняет свойство разрешимости проблемы для почти всех входов и обладает свойством транзитивности и рефлексивности. Доказывается, что классическая проблема выполнимости булевых формул является полной относительно этой сводимости в генерическом аналоге класса NP.

Ключевые слова: *генерическая сложность, проблема выполнимости, NP-полнота.*

DOI 10.17223/20710410/36/8

ON GENERIC NP-COMPLETENESS OF THE BOOLEAN SATISFIABILITY PROBLEM

A. N. Rybalov

Omsk State Technical University, Omsk, Russia

E-mail: alexander.rybalov@gmail.com

Generic-case approach to algorithmic problems was suggested by Miasnikov, Kapovich, Schupp and Shpilrain in 2003. This approach studies behavior of an algorithm on typical (almost all) inputs and ignores the rest of inputs. Many classical undecidable or hard algorithmic problems become feasible in the generic case. But there are generically hard problems. In this paper we introduce a notion of generic polynomial reducibility algorithmic problems, which preserve the property of polynomial decidability of the problem for almost all inputs and has the property of transitivity. It is proved that the classical satisfiability problem of Boolean formulas is complete with respect to this generic reducibility in the generic analogue of class NP.

Keywords: *generic complexity, Boolean satisfiability problem, NP-completeness.*

Введение

Важнейшим понятием классической теории сложности вычислений является понятие полиномиальной сводимости алгоритмических проблем. С его помощью можно сравнивать проблемы по вычислительной сложности и развивать богатую теорию NP-

[1]Работа поддержана грантом РНФ № 17-11-01117.

полноты [1]. В работе [2] введено понятие полиномиальной сводимости и NP-полноты в среднем. В [3] приведены примеры алгоритмических проблем, которые являются NP-полными в среднем. При анализе вычислительной сложности проблемы в среднем изучается математическое ожидание времени работы алгоритма для всех входов данного размера. В рамках генерического подхода [4] алгоритмическая проблема рассматривается не на всём множестве входов, а на некотором подмножестве «почти всех» входов. Такие входы образуют так называемое генерическое множество. Понятие «почти все» формализуется введением естественной меры на множестве входных данных. С точки зрения практики алгоритмы, быстро решающие проблему на генерическом множестве, так же хороши, как и быстрые алгоритмы для всех входов. Классическим примером такого алгоритма является симплекс-метод — он за полиномиальное время решает задачу линейного программирования для большинства входных данных, но имеет экспоненциальную сложность в худшем случае. Более того, может так оказаться, что проблема трудноразрешима или вообще неразрешима в классическом смысле, но легкоразрешима на генерическом множестве. Для многих классических NP-полных проблем существуют полиномиальные генерические алгоритмы [5].

В данной работе вводится понятие генерической полиномиальной сводимости алгоритмических проблем, которое сохраняет свойство разрешимости проблемы за полиномиальное время для почти всех входов при условии равенства классов $\mathrm{P} = \mathrm{BPP}$, где класс BPP состоит из проблем, разрешимых за полиномиальное время на вероятностных машинах Тьюринга. В работе [6] доказано, что это равенство следует из весьма правдоподобных гипотез о вычислительной сложности некоторых трудных проблем. Далее в работе доказывается, что проблема выполнимости булевых формул [7] является полной относительно генерической полиномиальной сводимости в генерическом аналоге класса NP.

1. Генерические алгоритмы

Пусть I — некоторое множество. Функция $\mathrm{size} : I \to \mathbb{N}$ называется *функцией размера*, если для любого $n \in \mathbb{N}$ множество $I_n = \{x \in I : \mathrm{size}(x) = n\}$ конечно. Например, если $I = \Sigma^*$ — множество слов над конечным алфавитом Σ, то функцией размера будет функция, определённая для любого слова w как его длина $|w|$. Для множества натуральных чисел $\mathbb{N}$ функция размера сопоставляет любому натуральному числу длину его двоичной записи. Как обычно делается в теории вычислимости, будем под алгоритмическими проблемами понимать проблемы распознавания подмножеств из некоторого множества входов с определённой на нем функцией длины. Для подмножества $S \subseteq I$ определим последовательность

$$\rho_n(S) = \frac{|S_n|}{|I_n|},\ n = 1, 2, 3, \ldots,$$

где $S_n = S \cap I_n$ — множество входов из S размера n. Заметим, что $\rho_n(S)$ — это вероятность попасть в S при случайной и равновероятной генерации входов из I_n. *Асимптотической плотностью* S назовем предел (если он существует)

$$\rho(S) = \lim_{n\to\infty} \rho_n(S).$$

Множество S называется *генерическим*, если $\rho(S) = 1$, и *пренебрежимым*, если $\rho(S) = 0$. Очевидно, что S генерическое тогда и только тогда, когда его дополнение $I \setminus S$ пренебрежимо.

Следуя [4], назовём множество S *строго пренебрежимым*, если последовательность $\rho_n(S)$ экспоненциально быстро сходится к нулю, т. е. существуют константы σ, $0 < \sigma < 1$, и $C > 0$, такие, что для любого n

$$\rho_n(S) < C\sigma^n.$$

Теперь S называется *строго генерическим*, если его дополнение $I \setminus S$ строго пренебрежимо.

Алгоритм $\mathcal{A}$ с множеством входов I и множеством выходов $J \cup \{?\}$ $(? \notin J)$ называется (*строго*) *генерическим*, если

1) $\mathcal{A}$ останавливается на всех входах из I;
2) множество $\{x \in I : \mathcal{A}(x) \neq ?\}$ является (строго) генерическим.

Аналогично определяется понятие вероятностного генерического алгоритма. Генерический алгоритм $\mathcal{A}$ вычисляет функцию $f : I \to J$, если $(\mathcal{A}(x) = y \in J) \Rightarrow (f(x) = y)$ для всех $x \in I$. Ситуация $\mathcal{A}(x) = ?$ означает, что $\mathcal{A}$ не может вычислить функцию f на аргументе x. Но условие 2 гарантирует, что $\mathcal{A}$ корректно вычисляет f на почти всех входах (входах из генерического множества). Множество $S \subseteq I$ называется (*строго*) *генерически разрешимым за полиномиальное время*, если существует (строго) генерический полиномиальный алгоритм, вычисляющий его характеристическую функцию. Различие между генерически разрешимыми проблемами и строго генерически разрешимыми проблемами поясняется в работе [8].

2. Генерическая полиномиальная сводимость

Пусть I, J — некоторые множества входов с определёнными на них функциями размера, $P(J)$ — множество всех подмножеств J. Множество $A \subseteq I$ *генерически полиномиально сводится* к множеству $B \subseteq J$ (обозначается $A \leqslant_{\mathrm{Gen}P} B$), если существуют вероятностный полиномиальный алгоритм $\mathcal{R} : I \times \mathbb{N} \to P(J) \cup \{?, !\}$, полином $p(n)$, полином $q(n)$ степени больше 2 и константа $C > 0$, такие, что:

1) для всех $x \in I$ либо $(\forall n\, \mathcal{R}(x, n) = ?)$, либо для всех $n \geqslant q(k)$, где $k = \mathrm{size}(x)$, выполнены следующие условия:
   а) $\forall y \in \mathcal{R}(x, n)$ $(y \neq ! \;\Rightarrow\; \mathrm{size}(y) = n)$;
   б) все элементы в $\mathcal{R}(x, n) \setminus \{!\}$ выдаются алгоритмом $\mathcal{R}$ равновероятно;
   в) вероятность получить ответ ! в $\mathcal{R}(x, n)$ не больше 2^{-Ck};
   г) $\dfrac{|\mathcal{R}(x, n)|}{|J_n|} > \dfrac{1}{(p(n))^k}$;
   д) $x \in A \Rightarrow \mathcal{R}(x, n) \subseteq B$;
   е) $x \notin A \Rightarrow \mathcal{R}(x, n) \subseteq J \setminus B$;
2) множество $\{x \in I : \forall n\ (\mathcal{R}(x, n) = ?)\}$ строго пренебрежимо.

Рассмотрим подробнее каждое из свойств данного определения. Помимо элементов множества J, алгоритм $\mathcal{R}$ может выдавать два особых выхода: ? — неопределённый ответ и ! — неудачная генерация. Вероятностный полиномиальный алгоритм $\mathcal{R}$ (если его выход не равен ?) каждому элементу x множества входов I проблемы A и каждому размеру $n \geqslant q(\mathrm{size}(x))$ сопоставляет достаточно большое подмножество (свойство 1, *г*) $\mathcal{R}(x, n)$ входов J проблемы B размера n (свойство 1, *а*). При этом каждый элемент из $\mathcal{R}(x, n)$, за исключением элемента !, генерируется равновероятно (свойство 1, *б*). Вероятность получить неудачно сгенерированный выход ! экспоненциально мала (свойство 1, *в*). Свойства 1, *д* и 1, *е* гарантируют, что это подмножество полностью попадает в B, если $x \in A$, и полностью лежит вне B для $x \notin A$. Свойство 2 говорит о том, что

алгоритм $\mathcal{R}$ лишь для пренебрежимого множества входов из I может всегда выдавать неопределённый ответ.

Теорема 1. Если $A \leqslant_{\mathrm{Gen}P} B$ и B строго генерически разрешимо за полиномиальное время, то для A существует полиномиальный вероятностный алгоритм, распознающий A на некотором полиномиальном строго генерическом множестве. Таким образом, при условии $\mathrm{P} = \mathrm{BPP}$ множество A строго генерически разрешимо за полиномиальное время.

Доказательство. Пусть $\mathcal{R}$ — полиномиальный вероятностный алгоритм, осуществляющий генерическую полиномиальную сводимость $A \leqslant_{\mathrm{Gen}P} B$; $\mathcal{B}$ — строго генерический полиномиальный алгоритм, вычисляющий характеристическую функцию множества B; $q(n)$ — многочлен из определения сводимости $A \leqslant_{\mathrm{Gen}P} B$. Вероятностный генерический алгоритм $\mathcal{A}$, вычисляющий характеристическую функцию множества A, работает на входе x размера n следующим образом:

1) запустить алгоритм $\mathcal{R}(x, q(n))$, который выдаёт ответ y;
2) если $y = ?$, выдать ?;
3) если $y = !$, выдать НЕТ;
4) иначе запустить алгоритм $\mathcal{B}$ на $y \in J$;
5) если $\mathcal{B}(y) = ?$, выдать НЕТ;
6) иначе выдать ответ $\mathcal{B}(y)$.

Очевидно, что этот вероятностный алгоритм полиномиален. Кроме того, неопределённый ответ он выдаёт на строго генерическом множестве — это следует из свойства 2 определения генерической полиномиальной сводимости. Заметим, что неправильный ответ алгоритм может выдать на шагах 3 и 5. Докажем, что вероятность этого меньше $1/2$. Для шага 3 вероятность меньше 2^{-Cn} по свойству 1, *в* из определения сводимости — меньше $1/4$ при достаточно большом n. Для шага 5 она не больше

$$\frac{|\{y \in J : \mathcal{B}(y) = ?\}_{q(n)}|}{|\mathcal{R}(x, q(n))|} = \frac{|\{y \in J : \mathcal{B}(y) = ?\}_{q(n)}|}{|J_{q(n)}|} \frac{|J_{q(n)}|}{|\mathcal{R}(x, q(n))|}.$$

Напомним, что индекс $q(n)$ внизу означает, что мы рассматриваем в соответствующих множествах только элементы размера $q(n)$. Так как алгоритм $\mathcal{B}$ строго генерический, то существует константа $\alpha > 0$, такая, что

$$\frac{|\{y \in J : \mathcal{B}(y) = ?\}_{q(n)}|}{|J|_{q(n)}} < \frac{1}{2^{\alpha q(n)}}.$$

Из свойства 1, *г* определения сводимости следует, что существует полином $p(n)$, такой, что для любого n

$$\frac{|J|_{q(n)}}{|\mathcal{R}(x, q(n))|} < p(q(n))^n = 2^{n \log(p(q(n)))}.$$

Получаем оценку сверху для вероятности выдачи ответа на шаге 5

$$\frac{2^{n \log(p(q(n)))}}{2^{\alpha q(n)}} = 2^{n \log(p(q(n))) - \alpha q(n)} < 1/4$$

для достаточно больших n, так как многочлен $q(n)$ имеет степень не меньше 2. Итого, получаем, что вероятность неправильного ответа меньше $1/4 + 1/4 = 1/2$. Таким отбразом, вероятностный алгоритм $\mathcal{A}$ работает корректно. ■

Докажем транзитивность полиномиальной генерической сводимости.

Теорема 2. Если $A \leqslant_{\mathrm{Gen}P} B$ и $B \leqslant_{\mathrm{Gen}P} C$, то $A \leqslant_{\mathrm{Gen}P} C$.

Доказательство. Пусть $\mathcal{R}_1$, $\mathcal{R}_2$ — полиномиальные вероятностные алгоритмы, осуществляющие генерические полиномиальные сводимости $A \leqslant_{\mathrm{Gen}P} B$ и $B \leqslant_{\mathrm{Gen}P} C$. Пусть $p_1(n), p_2(n), q_1(n), q_2(n)$ — соответствующие полиномы и C_1, C_2 — соответствующие константы. Вероятностный алгоритм $\mathcal{R}_3$, осуществляющий сводимость $A \leqslant_{\mathrm{Gen}P} C$, работает на входе (x, n), где $n > q_2(q_1(k))$, $k = \mathrm{size}(x)$, следующим образом:

1) запустить алгоритм $\mathcal{R}_1(x, q_1(k))$, который выдаст ответ y;
2) если $y = ?$, выдать ?;
3) если $y = !$, выдать !;
4) иначе запустить алгоритм $\mathcal{R}_2(y, n)$, который выдаст ответ z;
5) если $z = ?$, выдать !;
6) если $z = !$, выдать !;
7) иначе выдать z.

Очевидно, что этот вероятностный алгоритм является полиномиальным. Необходимо проверить, что выполнены все свойства полиномиальной генерической сводимости. Свойство 2 выполнено, так как алгоритм выдаёт неопределённый ответ только на тех элементах x, на которых это делает алгоритм $\mathcal{R}_1$. Свойства $1, а$ и $1, б$ выполняются, так как они выпонены для алгоритма $\mathcal{R}_2$. Свойства $1, д$ и $1, е$ выполняются, так как они верны для алгоритмов $\mathcal{R}_1$ и $\mathcal{R}_2$. Свойство $1, г$ также следует из того, что оно выполнено для алгоритма $\mathcal{R}_2$, при этом соответсвующий полином $p_3(n) = p_2(n)$. Докажем свойство $1, в$. Ответ ! может быть выдан на шагах 3, 5 и 6. Оценим вероятность для каждого шага. Для шага 3 вероятность меньше $2^{-C_1 k}$. На шаге 5 вероятность оценивается как в доказательстве теоремы 1: она не больше

$$2^{k \log(p_1(q_1(k))) - \alpha q_1(k)} < 2^{-C_1 k}$$

для достаточно больших k. Для шага 6 эта вероятность меньше $2^{-C_2 \mathrm{size}(y)} < 2^{-C_2 k}$ по свойству $1, в$ для сводимости $B \leqslant_{\mathrm{Gen}P} C$. Итого получаем оценку сверху

$$2^{-C_1 k} + 2^{-C_1 k} + 2^{-C_2 k} < 2^{-C_3 k}$$

для некоторой константы $C_3 > 0$. ■

3. Генерическая NP-полнота проблемы выполнимости булевых формул

В работе [8] введено представление булевых формул с помощью бинарных деревьев. Это представление более компактно и практично, чем классическое представление булевых формул с помощью таблиц истинности, когда размер таблицы растёт экспоненциально с ростом числа переменных. Представление формул с помощью бинарных деревьев часто используется в программировании различных приложений, связанных с символьными вычислениями. Кроме того, оно удобно для различного рода подсчётов.

Напомним вкратце, в чём заключается этот способ представления. Булевой формуле ϕ в базисе $\{\vee, \wedge, \neg\}$ сопоставляется бинарное дерево T_ϕ, внутренние вершины которого помечены символами $\vee$ и $\wedge$, а листья — переменными или их отрицаниями. С бинарным деревом T_ϕ, имеющим n листьев, связывается множество переменных $\{x_1, \ldots, x_n\}$ так, что каждому листу T_ϕ может быть приписана, вообще говоря, произвольная переменная из множества $\{x_1, \ldots, x_n\}$ либо её отрицание. Под размером $s(\phi)$ формулы ϕ будем понимать число листьев n дерева T_ϕ. В дальнейшем будем отождествлять каждую булеву формулу ϕ с деревом T_ϕ. Обозначим через $\mathcal{F}$ множество всех булевых формул, а через $\mathcal{F}_n$ — множество всех формул в $\mathcal{F}$ размера n.

Лемма 1 [8]. $|\mathcal{F}_n| = 2^{n-1}(2n)^n C_{n-1}$, где $C_n = \dfrac{1}{n+1}\binom{2n}{n}$ — n-е число Каталана.

Для любой формулы ϕ определим множество

$$Eq(\phi) = \{\phi \vee ((x_1 \wedge \neg x_1) \wedge \psi),\ \psi \text{ — произвольная формула}\}.$$

Очевидно, что ϕ выполнима тогда и только тогда, когда любая формула из $Eq(\phi)$ выполнима.

Лемма 2. Для любой формулы ϕ имеет место

$$\frac{|Eq(\phi) \cap \mathcal{F}_n|}{|\mathcal{F}_n|} > \frac{1}{(16(n-2))^k}$$

для любого $n > k+2$, где k — размер формулы ϕ.

Доказательство. Пусть формула ϕ имеет размер k. Тогда для любой формулы $\phi \vee ((x_1 \wedge \neg x_1) \wedge \psi)$ из множества $Eq(\phi) \cap \mathcal{F}_{n+2}$ формула ψ должна иметь размер $n-k$. Кроме того, в этой формуле может участвовать любая из n переменных. Поэтому, аналогично тому, как это делалось в доказательстве леммы 1, можно подсчитать

$$|Eq(\phi) \cap \mathcal{F}_{n+2}| = 2^{n-k-1}(2n)^{n-k} C_{n-k-1}.$$

Отсюда

$$\frac{|Eq(\phi) \cap \mathcal{F}_{n+2}|}{|\mathcal{F}_{n+2}|} = \frac{2^{n-k-1}(2n)^{n-k}C_{n-k-1}}{2^{n-1}(2n)^n C_{n-1}} = \frac{1}{(4n)^k}\frac{C_{n-k-1}}{C_{n-1}} = \frac{1}{(4n)^k}\frac{n}{n-k}\frac{\binom{2(n-k-1)}{n-k-1}}{\binom{2(n-1)}{n-1}} >$$

$$> \frac{1}{(4n)^k}\frac{(n-1)!}{(n-k-1)!}\frac{2(n-k-1)\dots(n-k)}{2(n-1)\dots n} = \frac{1}{(4n)^k}\frac{((n-1)\dots(n-k))^2}{2(n-1)\dots(2n-2k-1)} >$$

$$> \frac{1}{(4n)^k}\Big(\frac{(n-1)\dots(n-k)}{2(n-1)\dots(2n-2k-1)}\Big)^2 > \frac{1}{(4n)^k}\frac{1}{2^{2k}} = \frac{1}{(16n)^k}.$$

Таким образом, имеем $\dfrac{|Eq(\phi)_{n+2}|}{|\mathcal{F}_{n+2}|} > \dfrac{1}{(16n)^k}$, откуда $\dfrac{|Eq(\phi)_n|}{|\mathcal{F}_n|} > \dfrac{1}{(16(n-2))^k}$. ■

Определим генерический аналог класса NP. Множество $S \subseteq I$ принадлежит классу $\operatorname{sg}\mathrm{NP}$, если существует полиномиальное строго генерическое множество $G \subseteq I$, такое, что $S \cap G \in \mathrm{NP}$. Множество $S \in \operatorname{sg}\mathrm{NP}$ называется *генерически NP-полным*, если для любого $A \in \operatorname{sg}\mathrm{NP}$ имеет место $A \leqslant_{\mathrm{Gen}P} S$.

Теорема 3. Проблема выполнимости булевых формул генерически NP-полна.

Доказательство. Пусть $A \in \operatorname{sg}\mathrm{NP}$. Тогда существует полиномиальное строго генерическое множество G, такое, что $A \cap G \in \mathrm{NP}$. Отсюда, по теореме Кука, следует, что существует классическая полиномиальная сводимость f множества $A \cap G$ к проблеме выполнимости. Полиномиальный вероятностный алгоритм $\mathcal{R}$, генерически сводящий проблему A к проблеме выполнимости, работает на входе (x, n) следующим образом:

1) проверяет, принадлежит ли x множеству G;
2) если $x \notin G$, выдаёт ?;
3) если $x \in G$, то строит формулу $\phi = f(x)$, такую, что ϕ выполнима тогда и только тогда, когда $x \in A$;

4) случайно и равновероятно генерирует формулу из множества $Eq(\phi) \cap \mathcal{F}_n$. Как это делается за полиномиальное время, описано в [8].

Проверим свойства полиномиальной сводимости. Свойство 2 следует из того, что множество G строго генерическое. Свойства $1, а, б$ следуют из описания шага 4. Свойство $1, в$ очевидно выполняется: алгоритм вообще не выдаёт ответ !. Свойства $1, д, е$ также выполняются по построению формулы ϕ. Наконец, свойство $1, г$ следует из леммы 2. ■

Автор выражает благодарность рецензенту за полезные замечания и предложения по улучшению текста статьи.

ЛИТЕРАТУРА

1. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 419 с.
2. *Levin L.* Average case complete problems // SIAM J. Computing. 1987. V. 15. P. 285–286.
3. *Gurevich Y.* Average case completeness // J. Computer System Sciences. 1991. V. 42. P. 346–398.
4. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks // J. Algebra. 2003. V. 264. No. 2. P. 665–694.
5. *Gilman R., Miasnikov A. G., Myasnikov A. D., and Ushakov A.* Report on generic case complexity // Herald of Omsk University. 2007. Special Issue. P. 103–110.
6. *Impagliazzo R. and Wigderson A.* P = BPP unless E has subexponential circuits: Derandomizing the XOR Lemma // Proc. 29th STOC. El Paso: ACM, 1997. P. 220–229.
7. *Cook S. A.* The complexity of theorem proving procedures // Proc. 3d Annual ACM Symposium on Theory of Computing. N. Y., USA, 1971. P. 151–158.
8. *Рыбалов А.* О генерической сложности проблемы общезначимости булевых формул // Прикладная дискретная математика. 2016. № 2(32). С. 119–126.

REFERENCES

1. *Garey M. and Johnson D.* Computers and Intractability. N. Y., Freeman & Co, 1979. 340 p.
2. *Levin L.* Average case complete problems. SIAM J. Computing, 1987, vol. 15, pp. 285–286.
3. *Gurevich Y.* Average case completeness. J. Computer System Sciences, 1991, vol. 42, pp. 346–398.
4. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks. J. Algebra, 2003, vol. 264, no. 2, pp. 665–694.
5. *Gilman R., Miasnikov A. G., Myasnikov A. D., and Ushakov A.* Report on generic case complexity. Herald of Omsk University, 2007, Special Issue, pp. 103–110.
6. *Impagliazzo R. and Wigderson A.* P = BPP unless E has subexponential circuits: Derandomizing the XOR Lemma. Proc. 29th STOC, El Paso, ACM, 1997, pp. 220–229.
7. *Cook S. A.* The complexity of theorem proving procedures. Proc. 3d Annual ACM Symposium on Theory of Computing, N. Y., USA, 1971, pp. 151–158.
8. *Rybalov A.* O genericheskoy slozhnosti problemy obshcheznachimosti bulevykh formul [On generic complexity of the validity problem for Boolean formulas]. Prikladnaya Diskretnaya Matematika, 2016, no. 2(32), pp. 119–126. (in Russian)

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

УДК 519.7

О НЕИЗБЫТОЧНОМ ПРЕДСТАВЛЕНИИ МИНИМАКСНОГО БАЗИСА СТРОГИХ АССОЦИАТИВНЫХ ПРАВИЛ

В. В. Быкова, А. В. Катаева

Сибирский федеральный университет, г. Красноярск, Россия

Ассоциативные правила — тип зависимостей между данными, которые отражают, какие признаки или события встречаются совместно и насколько часто это происходит. Строгие ассоциативные правила представляют интерес для тех приложений, где требуется высокая степень уверенности в установленных зависимостях между данными, например, в информационной безопасности, анализе компьютерных сетей и медицине. Чрезмерно большое число выявленных правил существенно усложняет их экспертизу и применение. Для решения этой проблемы предложен алгоритм MClose, расширяющий возможности известного алгоритма Close. Алгоритм Close формирует минимаксный базис, в котором каждое строгое ассоциативное правило имеет минимальную посылку и максимальное следствие. Однако в минимаксном базисе остаются избыточные строгие ассоциативные правила. Алгоритм MClose в процессе построения минимаксного базиса распознаёт избыточные строгие ассоциативные правила и устраняет их. Предложенный алгоритм основан на свойствах замкнутых множеств. Доказаны выводимости, аргументирующие корректность алгоритма MClose.

Ключевые слова: *соответствия Галуа, замкнутые множества, строгие ассоциативные правила, неизбыточность, минимаксный базис.*

DOI 10.17223/20710410/36/9

ON THE NON-REDUNDANT REPRESENTATION OF THE MINIMAX BASIS OF STRONG ASSOCIATIONS

V. V. Bykova, A. V. Kataeva

Siberian Federal University, Krasnoyarsk, Russia

E-mail: bykvalen@mail.ru, kataeva_av@mail.ru

Associative rules are the type of dependencies between data that reflect which features or events occur together and how often this happens. Strong associative rules are of interest for those applications where a high degree of confidence of dependencies is required. For example, they are used in information security, computer network analysis and medicine. Excessively large number of identified rules significantly complicates their expert analysis and application. To reduce the severity of this problem, we propose the MClose algorithm, which extends the capabilities of the well-known algorithm Close. The Close algorithm forms a minimax basis in which each strong associative rule has a minimum premise and a maximal consequence. However, in the

minimax basis, some redundant strong associative rules remain. The MClose algorithm recognizes and eliminates them in the process of constructing a minimax basis. The proposed algorithm is based on the properties of closed sets. Its correctness is proved by proving the reflexivity, additivity, projectivity, and transitivity properties of strong associative rules.

Keywords: *Galois connection, closed sets, strong association rules, non-redundant, minimax basis.*

Введение

При поиске ассоциативных правил анализируемое множество данных обычно описывается бинарным контекстом — матрицей, строки которой отвечают объектам предметной области, а столбцы — признакам этих объектов. Единичное значение элемента матрицы трактуется как наличие у объекта соответствующего признака, а нулевое — как его отсутствие. Бинарное представление данных существенно расширяет математический аппарат для их исследования. Современные методы поиска ассоциативных правил базируются преимущественно на анализе формальных понятий и теории вероятностей [1 – 3]. Анализ формальных понятий, как прикладная ветвь алгебраической теории решёток, является удобным математическим аппаратом описания методов поиска ассоциативных правил [4 – 6].

Главная проблема при поиске ассоциативных правил — это огромное число правил, возникающих при анализе больших контекстов. Для решения этой проблемы используются различные меры значимости правил. С их помощью правила фильтруются и для анализа предъявляются только те, для которых значения мер превышают заданные пороговые значения. Подобная фильтрация, конечно, уменьшает число правил, но не решает проблему размерности полностью. Часто после фильтрации всё равно остаётся значительное число правил, при этом многие из них избыточные. Ассоциативное правило считается избыточным, если его удаление из множества правил не приводит к потере информации о связях между данными в рассматриваемой предметной области.

Множество ассоциативных правил, не содержащее избыточных (в некотором смысле) правил, принято называть базисом. Существуют различные формальные определения избыточных ассоциативных правил и методы их устранения [7]. Наиболее развиты методы устранения избыточности для строгих ассоциативных правил. Для таких ассоциативных правил имеется плотная параллель с функциональными зависимостями из теории реляционных баз данных [8]. Строгие ассоциативные правила имеют достоверность, равную единице, и представляют интерес для приложений, где требуется высокая степень уверенности в установленных зависимостях между данными, например, в информационной безопасности, анализе компьютерных сетей и медицине [9 – 11].

В настоящее время известен ряд алгоритмов, позволяющих строить различные базисы для строгих ассоциативных правил. Наиболее значимыми базисами являются канонический и минимаксный. Канонический базис (или базис Дюкена — Гига) состоит из минимального числа строгих ассоциативных правил, рекуррентно описываемых в терминах псевдосодержаний [12]. Канонический базис математически глубоко исследован, однако все предложенные на сегодняшний день алгоритмы его построения в большей степени представляют теоретический, чем практический интерес [13]. Минимаксный базис состоит из строгих ассоциативных правил, имеющих минимальную посылку и максимальное следствие [14]. Для минимаксного базиса имеются хо-

рошо апробированные алгоритмы, к которым относится алгоритм Close [7, 15]. Исследования показали, что в построенных канонических и минимаксных базисах остаётся некоторая избыточность, которая может быть устранена на основе выводимостей, подобных аксиомам Амстронга, известным в теории реляционных баз данных для функциональных зависимостей.

В данной работе для строгих ассоциативных правил предложен алгоритм MClose, расширяющий возможности алгоритма Close. Алгоритм MClose позволяет в процессе построения минимаксного базиса устранять избыточность с сохранением поддержки и достоверности без дополнительного обращения к исходному бинарному контексту. Доказаны выводимости, обосновывающие корректность алгоритма MClose. Доказательство этих выводимостей — основной результат работы.

1. Основные термины и обозначения анализа формальных понятий

Приведём термины и обозначения, традиционно применяемые в анализе формальных понятий [4 – 6].

Пусть для предметной области определены два непустых конечных множества G и M объектов и признаков соответственно (от немецких слов *Gegenstände* — объект, *Merkmale* — признак). Предполагаем, что все объекты в G и признаки в M различны. Пусть задано отношение $I \subseteq G \times M$ инцидентности между G и M. Тройку $K = (G, M, I)$ принято называть формальным контекстом (или просто контекстом) предметной области. Считаем, что существование в I пары (g, m) означает, что объект g имеет признак m, и наоборот — признак m присущ объекту g.

Выберем в $K = (G, M, I)$ два произвольных элемента $g \in G$ и $m \in M$. Определим для них два отображения ϕ и ψ:

$$\phi(g) = \{m \in M : (g, m) \in I\}, \quad \psi(m) = \{g \in G : (g, m) \in I\},$$

где $\phi(g)$ — множество признаков, присущих объекту g, а $\psi(m)$ — множество объектов, обладающих признаком m.

Отображения ϕ и ψ обобщаются на $A \subseteq G$ и $B \subseteq M$ следующим образом:

$$\phi(A) = \bigcap_{g \in A} \phi(g) = \{m \in M : \forall g \in A \ ((g, m) \in I)\},$$
$$\psi(B) = \bigcap_{m \in B} \psi(m) = \{g \in G : \forall m \in B \ ((g, m) \in I)\}.$$

Отсюда $\phi(A)$ — множество признаков, общих для всех объектов из A, а $\psi(B)$ — множество объектов, которые обладают всеми признаками из B. Отображения ϕ и ψ определены так, что если $A_1, A_2 \subseteq G$ и $B_1, B_2 \subseteq M$, то

$$\phi(A_1 \cup A_2) = \phi(A_1) \cap \phi(A_2), \quad \psi(B_1 \cup B_2) = \psi(B_1) \cap \psi(B_2).$$

Целесообразно положить, что $\phi(\varnothing) = M$ и $\psi(\varnothing) = G$: пустому множеству объектов присущи все признаки из M и каждый объект рассматриваемого контекста K обладает пустым множеством признаков. Если для отображений ϕ и ψ применить единое обозначение $(\cdot)'$, то формулы для $\phi(A), \psi(B), \phi(A_1 \cup A_2)$ и $\psi(B_1 \cup B_2)$ записываются так:

$$A' = \bigcap_{g \in A} g' = \{m \in M : \forall g \in A \ ((g, m) \in I)\}; \tag{1}$$
$$B' = \bigcap_{m \in B} m' = \{g \in G : \forall m \in B \ ((g, m) \in I)\}; \tag{2}$$

$$(A_1 \cup A_2)' = {A_1}' \cap {A_2}'; \tag{3}$$

$$(B_1 \cup B_2)' = {B_1}' \cap {B_2}'. \tag{4}$$

Если $g \in G$ и $m \in M$, то обозначения g' и m' традиционно служат сокращённой формой записи множеств $\phi(g) = \{g\}'$ и $\psi(m) = \{m\}'$ соответственно.

Из определения отображений «$'$» вытекают свойства, которые формально можно выразить в виде следующих утверждений.

Утверждение 1. Для всякого контекста $K = (G, M, I)$ и любых $B_1, B_2 \subseteq M$ верны следующие свойства:

— *антимонотонность*: если $B_1 \subseteq B_2$, то ${B_2}' \subseteq {B_1}'$;

— *экстенсивность*: $B_1 \subseteq {B_1}''$, где ${B_1}'' = ((B_1)')' \subseteq M$.

Утверждение 2. Для всякого контекста $K = (G, M, I)$ и любых $A_1, A_2 \subseteq G$ верны следующие свойства:

— *антимонотонность*: если $A_1 \subseteq A_2$, то ${A_2}' \subseteq {A_1}'$;

— *экстенсивность*: $A_1 \subseteq {A_1}''$, где ${A_1}'' = ((A_1)')' \subseteq G$.

В силу утверждений 1 и 2, отображения ϕ и ψ составляют пару соответствий Галуа между 2^G и 2^M — системами всех подмножеств множеств G и M соответственно, частично упорядоченными по теоретико-множественному включению [4, 5]. Известно, что для соответствий Галуа ϕ и ψ справедливы равенства [5]

$$\phi(\psi(\phi(A))) = \phi(A), \quad \psi(\phi(\psi(B))) = \psi(B)$$

или, то же самое, в единых обозначениях

$$((A')')' = (A'')' = A', \quad ((B')')' = (B'')' = B'. \tag{5}$$

Двойное применение отображения «$'$» определяет оператор замыкания на 2^M в алгебраическом смысле [4]. Ему свойственны:

— *рефлексивность*: для любого $B \subseteq M$ всегда $B \subseteq B''$;

— *монотонность*: если $B_1 \subseteq B_2 \subseteq M$, то ${B_1}'' \subseteq {B_2}'' \subseteq M$;

— *идемпотентность*: для любого $B \subseteq M$ всегда $(B'')'' = B''$.

Справедливость этих свойств вытекает из утверждений 1 и 2.

Если $B = B''$, то множество признаков $B \subseteq M$ называется замкнутым относительно оператора «$''$» в контексте K. Множество $B'' = \phi(\psi(B))$ можно трактовать как набор признаков, которые всегда появляются в объектах контекста K вместе с признаками из B, причём это множество является наибольшим по включению в пределах K. Очевидно, что $(\varnothing)'' = \phi(\psi(\varnothing)) = G'$, где G' — множество признаков, свойственных всем объектам контекста K. Если $B' = \varnothing$, то всегда $B'' = \phi(\psi(B)) = \phi(\varnothing) = M$. Если $B' \neq \varnothing$, то, исходя из (1)–(4), замыкание для $B \subseteq M$ можно вычислить за один просмотр контекста K по формуле

$$B'' = \bigcap_{g \in G} \{g' : B \subseteq g'\}. \tag{6}$$

В анализе формальных понятий пара множеств (A, B), $A \subseteq G$, $B \subseteq M$, таких, что $A' = B$ и $B' = A$, называется формальным понятием контекста $K = (G, M, I)$ с объёмом A и содержанием B [6]. В формальном понятии (A, B) множества A и B всегда замкнуты относительно «$''$» в этом контексте: $A = A''$ и $B = B''$. С помощью формальных понятий возможно концептуальное (или понятийное) моделирование различных предметных областей [3, 6]. Замкнутые множества также нашли широкое применение в поиске ассоциативных правил [7, 14, 15].

2. Ассоциативные правила и основные меры их значимости

Ассоциативным правилом на множестве признаков контекста $K = (G, M, I)$ называется упорядоченная пара множеств $r = (X, Y)$, $X, Y \subseteq M$. Принято ассоциативное правило $r = (X, Y)$ записывать в виде $X \Rightarrow Y$, здесь множества X и Y называют посылкой (или причиной) и заключением (или следствием) соответственно [16]. В анализе ассоциативных правил часто полагают, что посылка и заключение — непустые непересекающиеся множества. С формальных позиций эти ограничения несущественны. Применительно к заданному контексту K всякое ассоциативное правило $X \Rightarrow Y$ количественно характеризуется с помощью поддержки $\delta(X \Rightarrow Y)$ и достоверности $\gamma(X \Rightarrow Y)$ [17]. Эти числовые функции определяются через понятие поддержки множества признаков.

Поддержка $\delta(X)$ множества признаков $X \subseteq M$ в контексте $K = (G, M, I)$ — отношение числа объектов, которым присущи признаки X, к общему числу объектов, представленных в этом контексте:

$$\delta(X) = |X'|/|G|. \tag{7}$$

Таким образом, $\delta(X)$ — частота встречаемости в контексте K объектов, имеющих признаки X. Из формулы (7) следует, что для любого $X \subseteq M$ значение $\delta(X)$ неизменно находится в естественных границах

$$0 \leqslant \delta(X) \leqslant 1. \tag{8}$$

Чем ближе значение $\delta(X)$ к 1, тем большее число объектов рассматриваемого контекста обладает всеми признаками из X. В силу антимонотонности отображения «$'$» поддержка множества признаков также удовлетворяет свойству антимонотонности: для всякого контекста $K = (G, M, I)$ и любых $X, Y \subseteq M$ при $X \subseteq Y$ верно неравенство

$$\delta(Y) \leqslant \delta(X). \tag{9}$$

Согласно (9), поддержка множества признаков не может превышать поддержки любого из его подмножеств. Так, для произвольного $X \subseteq M$ всегда

$$0 \leqslant \delta(M) \leqslant \delta(X) \leqslant \delta(\varnothing) = 1.$$

Множество признаков $X \subseteq M$ называется частым в контексте $K = (G, M, I)$, если его поддержка больше или равна заданному пороговому значению $\delta_0 \in [0, 1]$. Если $\delta(X) \geqslant \delta_0$ и $X = X''$, то X называется частым замкнутым множеством признаков в K. Частые множества и частые замкнутые множества признаков традиционно служат основой для поиска ассоциативных правил в заданном контексте. Следует отметить, что в худшем случае число частых замкнутых множеств признаков контекста K совпадает с числом частых множеств признаков и экспоненциально зависит от $|M|$. Однако на практике обычно число частых замкнутых множеств значительно меньше числа частых множеств. Примеры контекстов с полиномиальным относительно $|M|$ числом частых замкнутых множеств можно найти в [15]. Использование следующего утверждения позволяет при поиске ассоциативных правил вместо частых множеств признаков применять частые замкнутые множества и тем самым сокращать пространство поиска ассоциативных правил.

Утверждение 3. Для всякого контекста $K = (G, M, I)$ и любого множества $X \subseteq M$ поддержка X'' совпадает с поддержкой множества X:

$$\delta(X'') = \delta(X).$$

Доказательство. Пусть $X \subseteq M$ — произвольное множество признаков контекста $K = (G, M, I)$. Тогда, исходя из (5) и (7), имеем равенство

$$\delta(X'') = |(X'')'|/|G| = |X'|/|G| = \delta(X).$$

Утверждение доказано. ■

Таким образом, если $\delta(X) \geqslant \delta_0$, то $\delta(X'') \geqslant \delta_0$, т. е. замыкание частого множества признаков также является частым.

Поддержкой $\delta(X \Rightarrow Y)$ ассоциативного правила $X \Rightarrow Y$ относительно контекста $K = (G, M, I)$ называется величина

$$\delta(X \Rightarrow Y) = \delta(X \cup Y) = |(X \cup Y)'|/|G|, \tag{10}$$

указывающая, какая доля объектов этого контекста имеет признаки $X \cup Y$. Достоверность $\gamma(X \Rightarrow Y)$ ассоциативного правила $X \Rightarrow Y$ относительно контекста K определяется как отношение числа объектов, обладающих всеми признаками из $X \cup Y$, к числу объектов, которым свойственны только признаки X:

$$\gamma(X \Rightarrow Y) = |(X \cup Y)'|/|X'|.$$

Достоверность ассоциативного правила через функцию поддержки выражается формулой

$$\gamma(X \Rightarrow Y) = \delta(X \Rightarrow Y)/\delta(X) = \delta(X \cup Y)/\delta(X). \tag{11}$$

Заметим, что достоверность определяется формулой (11) только для тех ассоциативных правил $X \Rightarrow Y$, для которых $\delta(X) \neq 0$. Если $\delta(X) = 0$ (в контексте нет ни одного объекта, обладающего признаками X), то, согласно (8) и (9), $\delta(X \cup Y) = 0$. В этом особом случае полагают $\gamma(X \Rightarrow Y) = 1$. Исходя из (7)–(11), достоверность ассоциативного правила $X \Rightarrow Y$ при произвольных $X, Y \subseteq M$ всегда находится в границах

$$0 \leqslant \gamma(X \Rightarrow Y) \leqslant 1.$$

Чем ближе значение $\gamma(X \Rightarrow Y)$ к 1, тем с большей уверенностью можно сказать, что признаки Y появляются в объектах рассматриваемого контекста вместе с признаками X.

Ассоциативное правило $X \Rightarrow Y$ называется минимаксным в контексте $K = (G, M, I)$, если для K не существует другого ассоциативного правила $X^* \Rightarrow Y^*$, такого, что $X^* \subseteq X$ и $Y \subseteq Y^*$ и

$$\delta(X^* \Rightarrow Y^*) = \delta(X \Rightarrow Y), \quad \gamma(X^* \Rightarrow Y^*) = \gamma(X \Rightarrow Y).$$

Пусть заданы контекст $K = (G, M, I)$ и δ_0, γ_0 — вещественные числа из $[0, 1]$. Будем говорить, что $X \Rightarrow Y$ является (δ_0, γ_0)-*ассоциативным правилом в* K, если выполняются два условия:

$$\delta_0 \leqslant \delta(X \Rightarrow Y) \leqslant 1; \tag{12}$$

$$\gamma_0 \leqslant \gamma(X \Rightarrow Y) \leqslant 1. \tag{13}$$

Величины δ_0 и γ_0 играют роль пороговых значений для поддержки и достоверности соответственно. При $\delta_0 = 0$ условие (12) отражает естественные границы поддержки. Данная ситуация свидетельствует о том, что нет ограничений на частоту появления

признаков $X \cup Y$ в K. При $\gamma_0 = 1$ условие (13) приводит к равенству $\gamma(X \Rightarrow Y) = 1$. В этом случае имеем $(\delta_0, 1)$-ассоциативное правило, которое будем называть *строгим ассоциативным правилом*. Таким образом, строгое ассоциативное правило — правило с достоверностью 1 и любой ненулевой поддержкой. Заметим, что ассоциативные правила с нулевой поддержкой и нулевой достоверностью не имеют практической ценности и поэтому не рассматриваются.

3. Свойства строгих ассоциативных правил

Известен критерий наличия в контексте строгого ассоциативного правила [6]. Приведём его с доказательством.

Утверждение 4. Достоверность ассоциативного правила $X \Rightarrow Y$ относительно контекста $K = (G, M, I)$ равна 1 тогда и только тогда, когда $X' \subseteq Y'$ (или $Y \subseteq X''$).

Доказательство. Согласно формуле (11), равенство $\gamma(X \Rightarrow Y) = 1$ верно тогда и только тогда, когда $\delta(X \cup Y) = \delta(X)$, или, то же самое, когда $(X \cup Y)' = X'$. В силу (4) имеем $(X \cup Y)' = X' \cap Y'$. Равенство $X' \cap Y' = X'$ возможно тогда и только тогда, когда $X' \subseteq Y'$.

Предположим, что $X' \subseteq Y'$. По утверждениям 1 и 2 всегда $Y \subseteq Y''$, а при $X' \subseteq Y'$ верно включение $Y'' \subseteq X''$. Отсюда $Y \subseteq X''$. Тогда в силу антимонотонности отображения «$'$» имеем $(X'')' \subseteq Y'$. С учётом (5) верно $X' \subseteq Y'$. ■

Заметим, что утверждение 4 тривиальным образом выполняется для $X \Rightarrow X''$ во всяком контексте K и при любом $X \subseteq M$. Рассмотрим некоторые частные случаи утверждения 4, важные с точки зрения устранения избыточности в множестве строгих ассоциативных правил.

С л у ч а й 1: ассоциативные правила вида $X \Rightarrow Y$ при любых $Y \subseteq X \subseteq M$.

В силу антимонотонности отображения «$'$» при $Y \subseteq X$ справедливо включение $X' \subseteq Y'$. Условие утверждения 4 выполняется, поэтому $\gamma(X \Rightarrow Y) = 1$. Таким образом, если в ассоциативном правиле заключение является подмножеством посылки, то такое правило имеет достоверность 1 в любом контексте K с поддержкой $\delta(X)$. Подобные строгие ассоциативные правила не несут в себе информации о существенных отношениях между множествами признаков X и Y, кроме естественного отношения «целое и часть целого», поэтому их следует считать тривиальными и не принимать во внимание. В частности, ассоциативные правила вида $\varnothing \Rightarrow \varnothing$, $M \Rightarrow Y$ и $X \Rightarrow \varnothing$ при любых $X, Y \subseteq M$ относятся к тривиальным строгим правилам.

С л у ч а й 2: ассоциативные правила вида $\varnothing \Rightarrow Y$ при $\varnothing \neq Y \subseteq M$.

Для всякого контекста $K = (G, M, I)$ и $Y \subseteq M$ всегда $Y' \subseteq G$. Кроме того, $\psi(\varnothing) = \varnothing' = G$ и $\delta(\varnothing) = 1$. Поэтому для правила $\varnothing \Rightarrow Y$ имеем

$$\gamma(\varnothing \Rightarrow Y) = \delta(\varnothing \cup Y)/\delta(\varnothing) = \delta(Y).$$

Исходя из утверждения 4, равенство $\gamma(\varnothing \Rightarrow Y) = 1$ имеет место тогда и только тогда, когда $\varnothing' \subseteq Y'$. Поскольку $\psi(\varnothing) = \varnothing' = G$, то $\varnothing' \subseteq Y'$ верно лишь при $G = Y'$. Таким образом, строгое ассоциативное правило $\varnothing \Rightarrow Y$ при $Y \neq \varnothing$ имеет поддержку $\delta(\varnothing) = 1$ и отражает наличие жёсткого ограничения на контекст K: все объекты, представленные в этом контексте, обязательно обладают множеством признаков Y.

Рассмотрим строгое ассоциативное правило $X \Rightarrow Y$. В силу (11) его поддержка всегда совпадает с поддержкой его посылки: $\delta(X \Rightarrow Y) = \delta(X)$. Если $\delta(X) \geqslant \delta_0$, то также $\delta(X \Rightarrow Y) \geqslant \delta_0$. Если после какого-либо изменения правила $X \Rightarrow Y$ результирующее правило имеет поддержку не менее $\delta(X)$, то говорят, что такое изменение

сохраняет поддержку исходного правила. Докажем свойства строгих ассоциативных правил, которые позволяют из одних строгих ассоциативных правил вывести другие строгие ассоциативные правила (с сохранением или без сохранения поддержки).

Лемма 1. Пусть в контексте $K = (G, M, I)$ множество $X \subseteq M$ имеет поддержку $\delta(X) \geqslant \delta_0$. Тогда для контекста K при любом $Y \subseteq X$ всегда справедливо строгое ассоциативное правило $X \Rightarrow Y$ с поддержкой $\delta(X \Rightarrow Y) \geqslant \delta_0$.

Доказательство. При $Y \subseteq X$ в силу антимонотонности отображения «$'$» всегда $X' \subseteq Y'$. Тогда по утверждению 4 ассоциативное правило $X \Rightarrow Y$ является строгим. В силу (11) для него $\delta(X \Rightarrow Y) = \delta(X) \geqslant \delta_0$. ■

Лемма 2. Если для контекста $K = (G, M, I)$ справедливо строгое ассоциативное правило $X \Rightarrow Y$ с поддержкой $\delta(X)$, то при любом $Z \subseteq M$ для этого контекста также верно строгое ассоциативное правило $X \cup Z \Rightarrow Y$ с поддержкой $\delta(X \cup Z) \leqslant \delta(X)$.

Доказательство. Воспользуемся утверждением 4 и свойством монотонности оператора замыкания. Так как $X \Rightarrow Y$ является строгим ассоциативным правилом в K, то $Y \subseteq X''$ и $\delta(X \cup Y) = \delta(X)$. При $X \subseteq X \cup Z$ верно включение $X'' \subseteq (X \cup Z)''$. Следовательно, $Y \subseteq (X \cup Z)''$. Это означает, что для K справедливо строгое ассоциативное правило $X \cup Z \Rightarrow Y$ и для него $\delta(X \cup Z \cup Y) = \delta(X \cup Z)$. Отсюда с учётом антимонотонности поддержки имеем

$$\delta(X \cup Z) = \delta(X \cup Z \cup Y) \leqslant \delta(X \cup Y) = \delta(X).$$

Лемма доказана. ■

Лемма 2 отражает возможность пополнения посылки для строгого ассоциативного правила, но без гарантии сохранения поддержки. Особо следует отметить случай, когда расширение посылки строгого ассоциативного правила сохраняет поддержку этого правила.

Следствие 1. Если для контекста $K = (G, M, I)$ справедливо строгое ассоциативное правило $X \Rightarrow Y$ с поддержкой $\delta(X)$, то при любом $Z \subseteq Y$ для этого контекста также справедливо строгое ассоциативное правило $X \cup Z \Rightarrow Y$ с поддержкой $\delta(X)$.

Доказательство. Если $\gamma(X \Rightarrow Y) = 1$, то по лемме 2 также $\gamma(X \cup Z \Rightarrow Y) = 1$. Значит, верны равенства

$$\delta(X \Rightarrow Y) = \delta(X \cup Y) = \delta(X), \quad \delta(X \cup Z \Rightarrow Y) = \delta(X \cup Z \cup Y) = \delta(X \cup Z).$$

Отсюда при $Z \subseteq Y$ имеем $\delta(X \cup Z \Rightarrow Y) = \delta(X \cup Z \cup Y) = \delta(X \cup Y) = \delta(X)$. ■

Лемма 3. Пусть в контексте $K = (G, M, I)$ множество $X \subseteq M$ имеет поддержку $\delta(X) \geqslant \delta_0$. Если для K справедливы строгие ассоциативные правила $X \Rightarrow Y$ и $X \Rightarrow Z$, то для этого контекста также справедливо строгое ассоциативное правило $X \Rightarrow Y \cup Z$ с поддержкой $\delta(X) \geqslant \delta_0$.

Доказательство. По утверждению 4, поскольку $X \Rightarrow Y$ и $X \Rightarrow Z$ являются строгими ассоциативными правилами в K, выполняются включения $Y \subseteq X''$ и $Z \subseteq X''$. Значит, $Y \cup Z \subseteq X''$, что достаточно для выполнимости строгого ассоциативного правила $X \Rightarrow Y \cup Z$ в заданном контексте. В данном случае $\delta(X \Rightarrow Y \cup Z) = \delta(X) \geqslant \delta_0$, т. е. свойство аддитивности сохраняет поддержку исходных правил. ■

Легко убедиться, что свойства рефлексивности и пополнения невыполнимы для произвольных (δ_0, γ_0)-ассоциативных правил. Однако следующее свойство, называемое *проективностью*, выполняется для любых (δ_0, γ_0)-ассоциативных правил.

Лемма 4. Если для $K = (G, M, I)$ справедливо (δ_0, γ_0)-ассоциативное правило $X \Rightarrow Y$, то при любых $Z \subseteq Y$ и $Y \neq \varnothing$ для этого контекста также верно (δ_0, γ_0)-ассоциативное правило $X \Rightarrow Z$.

Доказательство. Поскольку $Z \subseteq Y$ и $Y \neq \varnothing$, то Y можно представить в виде $Y = Z \cup (Y \backslash Z)$. Тогда, согласно условию (12) и антимонотонности функции поддержки, имеем

$$\delta_0 \leqslant \delta(X \cup Y) = \delta(X \cup (Z \cup (Y \backslash Z))) = \delta((X \cup Z) \cup (Y \backslash Z)) \leqslant \delta(X \cup Z).$$

Значит, $\delta(X \Rightarrow Z)$ удовлетворяет условию (12). Аналогично, полагая, что $\delta(X) \neq 0$, получаем

$$\gamma_0 \leqslant \gamma(X \Rightarrow Y) = \delta(X \cup Y)/\delta(X) \leqslant \delta(X \cup Z)/\delta(X) = \gamma(X \Rightarrow Z).$$

Следовательно, $\gamma(X \Rightarrow Z)$ удовлетворяет условию (13). При $\delta(X) = 0$ всегда $\gamma(X \Rightarrow \Rightarrow Y) = 1$ при любом Y, в том числе и $Z \subseteq Y$.

Применительно к строгим ассоциативным правилам лемма доказывается тривиальным образом. Если для контекста K справедливо строгое ассоциативное правило $X \Rightarrow Y$, то $Y \subseteq X''$. Значит, при любых $Z \subseteq Y$ и $Y \neq \varnothing$ справедливо включение $Z \subseteq X''$. Следовательно, $X \Rightarrow Z$ является строгим ассоциативным правилом в K. Кроме того, $\delta(X \Rightarrow Y) = \delta(X)$, $\delta(X \Rightarrow Z) = \delta(X)$. Если $\delta(X) \geqslant \delta_0$, то $\delta(X \Rightarrow Z) \geqslant \delta_0$. Лемма доказана. ■

Лемма 4 отражает тот факт, что правую часть всякого (δ_0, γ_0)-ассоциативного правила можно «расщепить» до отдельного признака, сохраняя при этом поддержку и достоверность в заданных границах. Для строгих ассоциативных правил леммы 3 и 4 констатируют равноценность различных эквивалентных форм записи этих правил.

Следствие 2. Представление строгого ассоциативного правила $X \Rightarrow Y \cup Z$ эквивалентно его представлению в виде двух строгих ассоциативных правил $X \Rightarrow Y$ и $X \Rightarrow Z$, при этом $\delta(X \Rightarrow Y \cup Z) = \delta(X \Rightarrow Y) = \delta(X \Rightarrow Z) = \delta(X)$.

Лемма 5. Если для контекста $K = (G, M, I)$ справедливы строгие ассоциативные правила $X \Rightarrow Y$ и $Y \Rightarrow W$ и $\delta(X) \geqslant \delta_0$, то какими бы ни были $X, Y, W \subseteq M$, для этого контекста также справедливо строгое ассоциативное правило $X \Rightarrow W$ с поддержкой $\delta(X) \geqslant \delta_0$.

Доказательство. Воспользуемся утверждением 4. Если для контекста K верны строгие ассоциативные правила $X \Rightarrow Y$ и $Y \Rightarrow W$, то верны включения $X' \subseteq Y'$ и $Y' \subseteq W'$. Следовательно, $X' \subseteq W'$. По утверждению 4 это условие является достаточным для выполнимости строгого ассоциативного правила $X \Rightarrow W$. Для него $\delta(X \Rightarrow W) = \delta(X) \geqslant \delta_0$. Таким образом, поддержка результирующего правила $X \Rightarrow W$ совпадает с поддержкой правила $X \Rightarrow Y$, играющего роль начала транзитивной цепочки строгих ассоциативных правил. ■

Следующая лемма обобщает лемму 5 и определяет свойство, называемое *псевдотранзитивностью* строгих ассоциативных правил.

Лемма 6. Если для контекста $K = (G, M, I)$ справедливы строгие ассоциативные правила $X \Rightarrow Y$ и $Y \cup Z \Rightarrow W$, то какими бы ни были $X, Y, Z, W \subseteq M$, для контекста K также справедливо строгое ассоциативное правило $X \cup Z \Rightarrow W$ с поддержкой $\delta(X \cup \cup Z) \leqslant \delta(X)$.

Доказательство. Если для контекста K верны строгие ассоциативные правила $X \Rightarrow Y$ и $Y \cup Z \Rightarrow W$, то $X' \subseteq Y'$ и $(Y \cup Z)' \subseteq W'$. Требуется доказать, что $(X \cup Z)' \subseteq \subseteq W'$.

По лемме 2 из строгого ассоциативного правила $X \Rightarrow Y$ вытекает справедливость строгого ассоциативного правила $(X \cup Z) \Rightarrow Y$. Тогда $(X \cup Z)' \subseteq Y'$. В силу антимонотонности отображения «$'$» всегда $(X \cup Z)' \subseteq Z'$. Отсюда если $(X \cup Z)' \subseteq Y'$ и $(X \cup Z)' \subseteq Z'$, то верно включение $(X \cup Z)' \subseteq Y' \cap Z'$. По формуле (4) имеем $(Y \cup Z)' = Y' \cap Z'$. Учитывая, что $(Y \cup Z)' \subseteq W'$, окончательно получаем

$$(X \cup Z)' \subseteq Y' \cap Z' = (Y \cup Z)' \subseteq W'.$$

К сожалению, свойство псевдотранзитивности не гарантирует для результирующего правила сохранение поддержки. Для него $\delta(X \cup Z \Rightarrow W) = \delta(X \cup Z)$, но ввиду антимонотонности поддержки $\delta(X \cup Z) \leqslant \delta(X)$. ■

Леммы 1–6 позволяют сформулировать следующую теорему.

Теорема 1. Для всякого контекста $K = (G, M, I)$ и любых $X, Y, Z, W \subseteq M$ справедливы следующие свойства строгих ассоциативных правил:

D_1. *Рефлексивность*: $X \Rightarrow X$.

D_2. *Пополнение*: если $X \Rightarrow Y$, то $X \cup Z \Rightarrow Y$.

D_3. *Аддитивность*: если $X \Rightarrow Y$ и $X \Rightarrow Z$, то $X \Rightarrow Y \cup Z$.

D_4. *Проективность*: если $X \Rightarrow Y$ и $Z \subseteq Y$, то $X \Rightarrow Z$.

D_5. *Транзитивность*: если $X \Rightarrow Y$ и $Y \Rightarrow W$, то $X \Rightarrow W$.

D_6. *Псевдотранзитивность*: если $X \Rightarrow Y$ и $Y \cup Z \Rightarrow W$, то $X \cup Z \Rightarrow W$.

Указанные в теореме 1 свойства (или выводимости) D_1–D_6 позволяют из некоторого множества строгих ассоциативных правил вывести многие другие строгие ассоциативные правила без дополнительного сканирования контекста. Выводимости, подобные D_1–D_6, справедливы и для функциональных зависимостей, имеющих место в теории реляционных баз данных, где их принято называть аксиомами Амстронга. Из доказанных лемм следует, что выводимости D_1, D_3, D_4, D_5 гарантируют сохранение поддержки: результатом применения их к строгим ассоциативным правилам с поддержкой не менее чем δ_0 всегда являются строгие ассоциативные правила с таким же порогом поддержки.

4. Алгоритмы Close и MClose

Известно большое число алгоритмов поиска ассоциативных правил. Основополагающими являются алгоритмы Apriori и Close. Алгоритм Apriori основан на свойстве антимонотонности функции поддержки. Он извлекает из заданного контекста все (δ_0, γ_0)-ассоциативные правила при любых заданных допустимых значениях δ_0 и γ_0 [11]. Алгоритм Close использует свойства частых замкнутых множеств и извлекает только строгие ассоциативные правила с заданным порогом поддержки δ_0 [7, 15].

Суть классического алгоритма Close заключается в пошаговом извлечении генераторов и частых замкнутых множеств признаков [14]. Множество $\rho \subseteq M$ называется *генератором* замкнутого множества признаков $X \subseteq M$, $X = X''$, если $\rho'' = X$ и не существует другого множества $\tau \subseteq M$, такого, что $\tau \subset \rho$ и $\tau'' = X$. Другими словами, генератор замкнутого множества признаков X — наименьшее по мощности множество признаков, имеющее замыкание X. Число признаков, входящих в генератор ρ, называется *мощностью* этого генератора. Если $|\rho| = k$, то ρ является k-генератором.

На вход алгоритма Close подается исходный контекст $K = (G, M, I)$ и пороговое значение δ_0. На выходе алгоритм Close формирует минимаксный базис и записывает его в множество AR. Изначально AR считается пустым и $k = 1$. На первом шаге в качестве k-генераторов рассматриваются все одноэлементные подмножества множества M. Замыкание ${\rho_k}''$ для генератора ρ_k вычисляется по формуле (6). Поддержка для ${\rho_k}''$ находится по формуле (7). Если $\delta({\rho_k}'') \geqslant \delta_0$, то по частому замкнутому множеству ${\rho_k}''$ строится минимаксное строгое ассоциативное правило

$$\rho_k \Rightarrow {\rho_k}''\backslash\rho_k \tag{14}$$

и сохраняется в AR. Согласно (10) и утверждению 3, для него

$$\delta(\rho_k \Rightarrow {\rho_k}''\backslash\rho_k) = \delta({\rho_k}'') \geqslant \delta_0, \quad \gamma(\rho_k \Rightarrow {\rho_k}''\backslash\rho_k) = 1.$$

Тот факт, что ассоциативное правило (14) является минимаксным, следует из определения генератора.

После генерации ассоциативного правила по ${\rho_k}''$ создаются кандидаты в $(k+1)$-генераторы для следующей итерации. Каждый такой кандидат формируется путём объединения двух k-генераторов, обладающих одинаковыми первыми $k-1$ признаками; предполагается, что все признаки в k-генераторах линейно упорядочены, например лексикографически. Далее выполняется проверка, вложен ли найденный кандидат в ${\rho_k}''$. Если вложен, то он исключается из рассмотрения. После нахождения всех $(k+1)$-генераторов осуществляется переход к следующей итерации. Алгоритм Close завершает работу, когда исчерпаны все генераторы.

Множество ассоциативных правил, полученных в результате работы алгоритма Close, образует минимаксный базис строгих ассоциативных правил контекста K. Корректность алгоритма Close доказана в [14]. К сожалению, минимаксный базис может содержать избыточные строгие ассоциативные правила. Алгоритм MClose с помощью выводимостей D_1, D_3, D_4, D_5 расширяет возможности алгоритма Close, он в процессе построения минимаксного базиса распознаёт избыточные строгие ассоциативные правила и устраняет их.

Дадим формальное определение избыточного строгого ассоциативного правила. Пусть AR — множество строгих ассоциативных правил, каждое из которых справедливо для контекста K. Будем говорить, что строгое ассоциативное правило $X \Rightarrow Y$ *логически следует* из множества AR, если оно может быть выведено из AR с помощью выводимостей D_1, D_3, D_4, D_5. Будем обозначать этот факт так: $AR \models X \Rightarrow Y$.

Строгое ассоциативное правило $X \Rightarrow Y$ назовём избыточным в AR, если

$$AR\backslash\{X \Rightarrow Y\} \models X \Rightarrow Y. \tag{15}$$

Множество строгих ассоциативных правил неизбыточное, если оно не содержит избыточных строгих ассоциативных правил. Обозначим через CSB (Concise Strong Basis) неизбыточное множество минимаксных строгих ассоциативных правил. Множество CSB можно построить путём генерации минимаксных строгих ассоциативных правил (например, с помощью алгоритма Close) и устранения среди них избыточных.

Распознавание избыточного строгого ассоциативного правила в AR основано на проверке логического следования (15). Алгоритм такой проверки использует понятие замыкания множества признаков относительно множества AR и является полиномиальным относительно $|M|$ и $|AR|$. *Замыканием* множества $X \subseteq M$ относительно AR

(обозначается X^+) называется множество всех признаков $m \in M$, таких, что верно логическое следование $AR \models X \Rightarrow m$. Заметим, что неизменно $X^+ \subseteq M$. Из выводимостей D_1, D_3, D_4 вытекает справедливость следующего критерия: логическое следование $AR \models X \Rightarrow Y$ имеет место тогда и только тогда, когда $Y \subseteq X^+$. Отсюда

$$AR \models X \Rightarrow X^+, \quad AR \models X \Rightarrow X^+ \backslash X.$$

Чтобы убедится в справедливости (15), достаточно вычислить X^+ относительно $AR \backslash \{X \Rightarrow Y\}$ и проверить включение $Y \subseteq X^+$. Если $Y \subseteq X^+$, то строгое ассоциативное правило $X \Rightarrow Y$ избыточно в AR, иначе оно не является избыточным.

Алгоритм вычисления X^+ целиком базируется на выводимостях D_1, D_3, D_4, D_5 и сводится к выполнению следующих действий. Сначала полагается $X^+ = X$. Далее осуществляется просмотр правил из AR и пополнение X^+ по следующему принципу: если для правила $Y \Rightarrow Z \in AR$ верно включение $Y \subseteq X^+$, то множество Z добавляется к X^+. Этот процесс повторяется до тех пор, пока изменяется X^+. Поскольку множества M и AR конечные, то процесс вычисления X^+ конечен.

Заметим, что процесс исключения избыточных строгих ассоциативных правил не требует доступа к контексту K, и поэтому время его выполнения незначительно по сравнению со временем получения частых замкнутых множеств признаков. Для того чтобы исключить добавление заведомо избыточного строгого ассоциативного правила в AR, необходимо всякий раз после построения ${\rho_k}''$ выполнять следующие действия. Если посылка ρ_k найденного правила (14) не равна ${\rho_k}''$, то найти замыкание ρ_k^+ относительно вычисленного множества AR. Если $\rho_k^+ = {\rho_k}''$, то это минимаксное ассоциативное правило является избыточным, иначе оно включается в AR.

После завершения генерации минимаксных строгих ассоциативных правил необходим дополнительный просмотр результирующего множества AR с целью обнаружения оставшихся избыточных правил. Такие правила вполне возможны: они неизбыточные по отношению к ранее выявленным правилам, однако после пополнения AR новыми правилами могут оказаться избыточными. Таким образом, по построению результирующее множество AR состоит из минимаксных строгих ассоциативных правил и является неизбыточным. Заметим, что оперативное удаление избыточных правил сдерживает рост мощности AR и снижает время выполнения алгоритма.

Алгоритмы Apriori, Close и MClose сравнивались по числу сгенерированных строгих ассоциативных правил и времени работы. Эксперименты осуществлялись на компьютере с процессором Intel® Core™ i5 CPU & 2.30 ГГц и ОЗУ размером 4 Гбайт. Эксперименты выполнялись на контекстах, сгенерированных случайным образом.

Результаты экспериментов представлены в таблице. Для всякого анализируемого контекста $K = (G, M, I)$ указаны $|G|$ — число объектов, $|M|$ — число признаков, $\sigma = n/(|G| \cdot |M|)$ — плотность контекста, где n задаёт число единичных элементов матрицы инцидентности I. Контекст из 10000 объектов сформирован многократным копированием контекста, состоящего из 500 объектов.

Из таблицы видно, что алгоритмы Close и MClose эффективнее алгоритма Apriori как по числу извлечённых строгих ассоциативных правил, так и по времени работы. Алгоритм MClose по времени работы сопоставим с алгоритмом Close. Однако алгоритм MClose более чем в 2 раза уменьшает мощность минимаксного базиса, формируемого алгоритмом Close. Перспективны исследования, направленные на устранение избыточности для ассоциативных правил с любыми поддержками и достоверностями.

Результаты экспериментов

Характеристика контекста $K = (G, M, I)$			Число извлечённых строгих ассоциативных правил / время, мс		
$\|G\|$	$\|M\|$	σ	Apriori	Close	MClose
20	10	0,38	1797 / 17562	45 / 250	22 / 297
30	10	0,39	2029 / 18347	46 / 374	19 / 412
30	10	0,55	15438 / 187202	69 / 390	20 / 484
50	10	0,53	27769 / 375178	46 / 78	13 / 124
500	10	0,53	27769 /376154	42 / 124	13 / 168
10000	10	0,53	27769 / 378400	42 / 671	13 / 872

ЛИТЕРАТУРА

1. *Витяев Е. Е., Демин А. В., Пономарев Д. К.* Вероятностное обобщение формальных понятий // Программирование. 2012. № 5. С. 18–34.
2. *Городецкий В. И., Самойлов В. В.* Ассоциативный и причинный анализ и ассоциативные байесовские сети // Тр. СПИИРАН. 2009. Вып. 9. С. 13–65.
3. *Кузнецов С. О.* Автоматическое обучение на основе анализа формальных понятий // Автоматика и телемеханика. 2001. № 10. С. 3–27.
4. *Биркгоф Г., Барти Т.* Современная прикладная алгебра. СПб.: Лань, 2005. 400 с.
5. *Гуров С. И.* Булевы алгебры, упорядоченные множества, решетки: определения, свойства, примеры. М.: Книжный дом «ЛИБРОКОМ», 2013. 352 с.
6. *Ganter B. and Wille R.* Formal Concept Analyses: Mathematical Foundations. Springer Science and Business Media, 2012. 314 p.
7. *Pasquier N., Bastide Y., Taoui R., and Lakhal L.* Generating a condensed representation for association rules // J. Intelligent Inform. Systems. 2005. V. 24. No. 1. P. 29–60.
8. *Майер Д.* Теория реляционных баз данных. М.: Мир, 1987. 608 с.
9. *Батура Т. В.* Модели и методы анализа компьютерных социальных сетей // Программные продукты и системы. 2013. № 3. С. 130–137.
10. *Платонов В. В., Семенов П. О.* Методы сокращения размерности в системах обнаружения сетевых атак // Проблемы информационной безопасности. Компьютерные системы. 2012. № 3. С. 40–45.
11. *Ilayaraja M. and Meyyappan T.* Mining medical data to identify frequent diseases using Apriori algorithm // Pattern Recognition, Informatics and Mobile Engineering (PRIME), IEEE, 2013. P. 194–199.
12. *Duquenne V. and Obiedkov S. A.* Attribute-incremental construction of the canonical implication basis // Ann. Math. Artif. Intelligence. 2007. V. 49. No. 1. P. 77–99.
13. *Rudolph S.* Some notes on pseudo-closed sets // LNCS. 2007. V. 4390. P. 151–165.
14. *Zaki M. J. and Hsiao C.-J.* Efficient algorithms for mining closed itemsets and their lattice structure // IEEE Trans. Knowledge Data Eng. 2005. V. 17. No. 4. P. 462–478.
15. *Uno T., Asai T., Uchida Y., and Arimura H.* An efficient algorithm for enumerating closed patterns in transaction databases // LNCS. 2004. V. 3245. P. 16–31.
16. *Zhang C. and Zhang S.* Association Rules Mining. Springer, 2002. 240 p.
17. *Geng L. and Hamilton H. J.* Interestingness measures for data mining: a survey // ACM Computing Surveys. 2006. V. 38. No. 3. Article 9.

REFERENCES

1. *Vityaev E., Demin A. V., and Ponomaryov D. K.* Probabilistic generalization of formal concepts. Programming and Computer Software, 2012, no. 5, pp. 219–230.

2. *Gorodetskiy V. I., Samoylov V. V.* Assotsiativnyy i prichinnyy analiz i assotsiativnye bayesovskie seti [Association and Casual rule mining using associative Bayesian networks]. Tr. SPIIRAN, 2009, iss. 9, pp. 13–65. (in Russian)
3. *Kuznetsov S. O.* Machine learning on the basis of formal concept analysis. Automation and Remote Control, 2001, vol. 6, iss. 10, pp. 1543–1564.
4. *Birkhoff G. and Bartee T. C.* Modern Applied Algebra. N. Y., McGraw-Hill, 1970. 431 p.
5. *Gurov S. I.* Bulevy algebry, uporyadochennye mnozhestva, reshetki: opredeleniya, svoystva, primery [Boolean Algebras, Ordered Sets, Lattices: Definitions, Properties, Examples]. Moscow, LIBROKOM Publ., 2013. 352 p. (in Russian)
6. *Ganter B. and Wille R.* Formal Concept Analyses: Mathematical Foundations. Springer Science and Business Media, 2012. 314 p.
7. *Pasquier N., Bastide Y., Taoui R., and Lakhal L.* Generating a condensed representation for association rules. J. Intelligent Inform. Systems, 2005, vol. 24, no. 1, pp. 29–60.
8. *Maier D.* Theory of Relational Databases. Computer Science Pr., 1983. 640 p.
9. *Batura T. V.* Modeli i metody analiza komp'yuternykh sotsial'nykh setey [Social networks analysis models and methods]. Programmnye Produkty i Sistemy, 2013, no. 3, pp. 130–137. (in Russian)
10. *Platonov V. V. and Semenov P. O.* Metody sokrashcheniya razmernosti v sistemakh obnaruzheniya setevykh atak [Dimension reduction in network attacks detection systems]. Information Security Problems. Computer Systems, 2012, no. 3, pp. 40–45. (in Russian)
11. *Ilayaraja M. and Meyyappan T.* Mining medical data to identify frequent diseases using Apriori algorithm. Pattern Recognition, Informatics and Mobile Engineering (PRIME), IEEE, 2013, pp. 194–199.
12. *Duquenne V. and Obiedkov S. A.* Attribute-incremental construction of the canonical implication basis. Ann. Math. Artif. Intelligence, 2007, vol. 49, no. 1, pp. 77–99.
13. *Rudolph S.* Some notes on pseudo-closed sets. LNCS, 2007, vol. 4390, pp. 151–165.
14. *Zaki M. J. and Hsiao C.-J.* Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Trans. Knowledge Data Eng., 2005, vol. 17, no. 4, pp. 462–478.
15. *Uno T., Asai T., Uchida Y., and Arimura H.* An efficient algorithm for enumerating closed patterns in transaction databases. LNCS, 2004, vol. 3245, pp. 16–31.
16. *Zhang C. and Zhang S.* Association Rules Mining. Springer, 2002. 240 p.
17. *Geng L. and Hamilton H. J.* Interestingness measures for data mining: a survey. ACM Computing Surveys, 2006, vol. 38, no. 3, article 9.

СВЕДЕНИЯ ОБ АВТОРАХ

АГИБАЛОВ Геннадий Петрович — доктор технических наук, профессор, профессор кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск. E-mail: **agibalov@isc.tsu.ru**

БЫКОВА Валентина Владимировна — доктор физико-математических наук, профессор Института математики и фундаментальной информатики Сибирского федерального университета, г. Красноярск. E-mail: **bykvalen@mail.ru**

ВАХРАМЕЕВ Михаил Анатольевич — аспирант Института математики им. С. Л. Соболева СО РАН, г. Омск. E-mail: **vahrmih@yandex.ru**

ВОЛГИН Артем Владимирович — преподаватель Московского государственного университета информационных технологий, радиотехники и электроники, г. Москва. E-mail: **artem.volgin@bk.ru**

ГАМАЮНОВ Денис Юрьевич — кандидат физико-математических наук, старший научный сотрудник, и.о. заведующего ЛБИС ВМК Московского государственного университета имени М. В. Ломоносова, г. Москва. E-mail: **gamajun@seclab.cs.msu.su**

ГЛУХОВ Михаил Михайлович — кандидат физико-математических наук, доцент кафедры ГТУ МИРЭА, г. Москва. E-mail: **mmgthl@mail.ru**

КАМЛОВСКИЙ Олег Витальевич — кандидат физико-математических наук, доцент, ООО «Центр сертификационных исследований», г. Москва.
E-mail: **ov-kam@yandex.ru**

КАТАЕВА Алина Владимировна — аспирантка Института математики и фундаментальной информатики Сибирского федерального университета, г. Красноярск. E-mail: **kataeva_av@mail.ru**

ЛЕОНТЬЕВ Виктор Леонтьевич — доктор физико-математических наук, профессор Ульяновского государственного университета, г. Ульяновск.
E-mail: **LeontievVL@ulsu.ru**

РОМАНЬКОВ Виталий Анатольевич — доктор физико-математических наук, профессор Омского государственного университета им. Ф. М. Достоевского, г. Омск. E-mail: **romankov48@mail.ru**

РЫБАЛОВ Александр Николаевич — кандидат физико-математических наук, научный сотрудник научно-исследовательской лаборатории «Информационная безопасность» кафедры комплексной защиты информации Омского государственного технического университета, г. Омск, Россия. E-mail: **alexander.rybalov@gmail.com**

СКОВОРОДА Анастасия Алексеевна — соискатель, программист ЛБИС ВМК Московского государственного университета имени М. В. Ломоносова, г. Москва.
E-mail: **nastya_jane@seclab.cs.msu.su**

ЩУРЕНКО Александр Васильевич — аспирант Ульяновского государственного университета, г. Ульяновск. E-mail: **tshurens@yandex.ru**

Журнал «Прикладная дискретная математика» включен в перечень ВАК рецензируемых российских журналов, в которых должны быть опубликованы основные результаты диссертаций, представляемых на соискание учёной степени кандидата и доктора наук, в базы данных Web of Science (Russian Science Citation Index — RSCI) и Scopus, а также в перечень журналов, рекомендованных ФУМО ВО ИБ в качестве учебной литературы по специальности «Компьютерная безопасность».

Журнал «Прикладная дискретная математика» распространяется по подписке; его подписной индекс 38696 в объединённом каталоге «Пресса России». Полнотекстовые электронные версии вышедших номеров журнала доступны на его сайте journals.tsu.ru/pdm и на Общероссийском математическом портале www.mathnet.ru. На сайте журнала можно найти также правила подготовки рукописей статей в журнал.

Тематика публикаций журнала:

- *Теоретические основы прикладной дискретной математики*
- *Математические методы криптографии*
- *Математические методы стеганографии*
- *Математические основы компьютерной безопасности*
- *Математические основы надёжности вычислительных и управляющих систем*
- *Прикладная теория кодирования*
- *Прикладная теория автоматов*
- *Прикладная теория графов*
- *Логическое проектирование дискретных автоматов*
- *Математические основы информатики и программирования*
- *Вычислительные методы в дискретной математике*
- *Дискретные модели реальных процессов*
- *Математические основы интеллектуальных систем*
- *Исторические очерки по дискретной математике и её приложениям*