МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2018 № 39

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 14.03.2018. Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 15. Тираж 300 экз.
Заказ № 3060. Цена свободная. Дата выхода в свет 30.03.2018.

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

СОДЕРЖАНИЕ

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

УДК 519.157

ОБ ЭКСТРЕМАЛЬНЫХ ВЕРОЯТНОСТЯХ ОСУЩЕСТВЛЕНИЯ k ИЗ n СОБЫТИЙ

Ю. А. Зуев

Московский государственный технический университет имени Н. Э. Баумана, г. Москва, Россия

В произвольном вероятностном пространстве заданы n событий, каждое из которых имеет вероятность p. На корреляционные связи между событиями не накладывается каких-либо ограничений. Из множества n событий выбираются всевозможные подмножества из k событий, и для каждого подмножества рассматривается вероятность их совместного осуществления. Выбираются подмножество с минимальной вероятностью и подмножество с максимальной вероятностью. Получены точные границы, в которых лежит каждая из этих вероятностей.

Ключевые слова: *событие, вероятность, линейное программирование, оптимальный базис.*

DOI 10.17223/20710410/39/1

ON EXTREME JOINT PROBABILITIES OF k EVENTS CHOSEN FROM n EVENTS

Yu. A. Zuev

Bauman Moscow State Technical University, Moscow, Russia

E-mail: 79851965730@yandex.ru

An arbitrary probability space with n events is considered. All events have the same probability p. No restrictions on correlations between the events are imposed and the events are considered simply as arbitrary subsets of measure p in the probability space. From the set of n events, all C_n^k subsets X consisting of k events are chosen, and for each such subset X the probability $\mathsf{P}(X)$ of joint implementation of its k events is considered. The subset with the minimum probability $\min\limits_{X:|X|=k} \mathsf{P}(X)$ and the subset with the maximum probability $\max\limits_{X:|X|=k} \mathsf{P}(X)$ are selected. In the paper, exact boundaries for both probabilities are obtained. For minimum probability:

$$\text{if } kp \leqslant k-1, \text{ then } \quad 0 \leqslant \min_{X:|X|=k} \mathsf{P}(X) \leqslant p;$$

$$\text{if } kp > k-1, \text{ then } \quad kp-k+1 \leqslant \min_{X:|X|=k} \mathsf{P}(X) \leqslant p.$$

For maximum probability:

$$\text{if } np < k-1, \text{ then } \quad 0 \leqslant \max_{X:|X|=k} \mathsf{P}(X) \leqslant p;$$

$$\text{if } k-1 \leqslant np < k, \text{ then } \quad \frac{np - \lfloor np \rfloor}{C_n^k} \leqslant \max_{X:|X|=k} \mathsf{P}(X) \leqslant p;$$

$$\text{if } k \leqslant np, \text{ then } \quad \frac{(\lfloor np \rfloor + 1 - np)C_{\lfloor np \rfloor}^k + (np - \lfloor np \rfloor)C_{\lfloor np \rfloor + 1}^k}{C_n^k} \leqslant \max_{X:|X|=k} \mathsf{P}(X) \leqslant p.$$

Keywords: *event, probability, linear programming, optimum base.*

Введение

С самого своего возникновения теория вероятностей была тесно связана с комбинаторикой. Впоследствии, когда теория вероятностей занялась изучением непрерывных процессов, эта связь ослабела. Сейчас, однако, она вновь окрепла. Развившийся в последние десятилетия *вероятностный метод* стал важным инструментом дискретной математики [1]. Если при становлении теории вероятностей элементарная комбинаторика была основным методом вычисления вероятностей, то теперь методами теории вероятностей в ряде случаев удаётся доказывать существование заданных комбинаторных объектов. Суть метода сводится к тому, что на множестве изучаемых объектов вводится некоторое вероятностное распределение, в простейшем случае равномерное, а затем доказывается, что вероятность объекта с заданными свойствами отлична от нуля. Это и является доказательством существования требуемого объекта, конструктивное построение которого может быть связано со значительными трудностями.

Вероятностный метод стимулировал новый интерес к конечным вероятностным пространствам. Вычисление вероятности одновременного осуществления k из n заданных на одном вероятностном пространстве событий с момента возникновения теории вероятностей было одной из её главных задач. В простейшем случае, когда события независимы и имеют одну и ту же вероятность p, вероятность осуществления ровно k событий задаётся формулой Бернулли $P(k) = C_n^k p^k (1-p)^{n-k}$ [2]. В случае наличия корреляций между событиями вероятности $P(k)$, $k = 0, 1, 2, \ldots, n$, могут быть найдены методом включения и исключения или оценены с помощью неравенств Бонферрони [3]. Для использования этих методов необходимо знание вероятностей совместного осуществления групп событий. Значительное число работ посвящено получению оценок, уточняющих неравенства Бонферрони. С обзорами этих исследований, дополненными собственными результатами авторов в этой области, можно ознакомиться по работам [4, 5].

Стоит также заметить, что ввиду того, что при использовании вероятностного метода в комбинаторных исследованиях особенную важность приобретает оценка $P(0)$ — вероятности того, что не осуществится ни одно из n событий, — здесь помимо уже перечисленных методов важную роль играет локальная лемма Ловаса, согласно которой $P(0) > 0$, если каждое из n событий зависит от не более чем d остальных событий и $pd < e^{-1}$ [1].

Задачи, рассматриваемые в настоящей работе, примыкают к перечисленным выше исследованиям и находятся в пограничной области между теорией вероятностей и дискретной математикой.

Пусть в произвольном вероятностном пространстве заданы n событий, каждое из которых имеет одну и ту же вероятность p. В случае независимости событий вероят-

ность осуществления любых k из них одинакова и равна p^k. Постулат независимости является фундаментальным в теории вероятностей и кладётся в основу большинства вероятностных моделей [6]. Однако в большинстве практических задач, использующих вероятностные модели, условия независимости в лучшем случае могут выполняться лишь приближённо. Если требование независимости не выполняется и между событиями допускаются корреляционные связи, то вероятности совместного осуществления для различных групп из k событий могут быть различны. В этом случае можно ставить задачу выбора k событий с максимальной вероятностью совместного осуществления и задачу выбора k событий с минимальной вероятностью совместного осуществления. При этом возникает вопрос о том, в каком диапазоне могут лежать данные вероятности. Ответ на этот вопрос зависит от имеющихся между событиями корреляционных связей.

Без требования независимости и без наложения каких-либо фиксированных корреляционных связей события могут рассматриваться как произвольные подмножества меры p в множестве единичной меры, которым является всё вероятностное пространство. Если в таких условиях поставить задачу выбора группы из k событий с максимальной вероятностью совместного осуществления или из k событий с минимальной вероятностью совместного осуществления, то в каких пределах могут лежать эти вероятности? При этом рассматриваемая мощность групп k может быть любым числом от 2 до n. Такая постановка задачи представляется достаточно естественной и заслуживает внимания.

Получение интервалов, в которых лежат данные экстремальные вероятности, является темой настоящей работы. В рассматриваемой модели, в которой события являются произвольными подмножествами меры p, используемые в настоящей работе методы не принадлежат собственно теории вероятностей, а являются, по-существу, комбинаторно-мощностными методами дискретной математики.

Если X — произвольное подмножество из k событий, то через $\mathsf{P}(X)$ будем обозначать вероятность того, что все события из X осуществятся. Тогда рассматриваемые экстремальные вероятности для групп мощности k выразятся как $\max\limits_{X:|X|=k} \mathsf{P}(X)$ и $\min\limits_{X:|X|=k} \mathsf{P}(X)$.

1. Границы для минимальной вероятности

Рассмотрим сначала вопрос о минимальной вероятности $\min\limits_{X:|X|=k} \mathsf{P}(X)$, который решается наиболее просто. Заметим, прежде всего, что верхняя граница для неё равна p. Она достигается, когда все n событий совпадают. Полностью же границы для $\min\limits_{X:|X|=k} \mathsf{P}(X)$ задаются следующей теоремой.

Теорема 1. Для множества из n событий, каждое из которых имеет вероятность p, минимальная по всем подмножествам X из k событий ($2 \leqslant k \leqslant n$) вероятность $\min\limits_{X:|X|=k} \mathsf{P}(X)$ совместного осуществления всех событий подмножества X лежит внутри следующих интервалов:

1) если $kp \leqslant k-1$, то $0 \leqslant \min\limits_{X:|X|=k} \mathsf{P}(X) \leqslant p$;
2) если $kp > k-1$, то $kp-k+1 \leqslant \min\limits_{X:|X|=k} \mathsf{P}(X) \leqslant p$.

Доказательство. Нижняя оценка, равная P, достигается в том случае, когда среди n событий имеется группа из k событий, таких, что вероятность их совместного осуществления есть P, а во всех остальных точках вероятностного пространства

осуществляется ровно $k-1$ из этих событий. Тогда для вероятности P имеет место соотношение $(k-1)(1-P)+kP=kp$. Отсюда

$$P=kp-k+1.$$

При $kp-k+1>0$ эта величина является нижней границей для $\min\limits_{X:|X|=k}\mathsf{P}(X)$. Если $kp-k+1\leqslant 0$, то становится возможным случай, когда во всех точках вероятностного пространства осуществляется не более $k-1$ событий, и нижняя граница для $\min\limits_{X:|X|=k}\mathsf{P}(X)$ оказывается равной нулю. ■

Отметим, что интервалы, задаваемые теоремой 1, являются универсальными, т. е. для любого подмножества из k событий вероятность их совместного осуществления лежит в этих интервалах.

2. Границы для максимальной вероятности

Рассмотрим теперь вопрос о максимальной вероятности $\max\limits_{X:|X|=k}\mathsf{P}(X)$. Ясно, что верхняя граница для неё всегда равна p. Она достигается, когда имеется группа из k совпадающих событий. Однако нижние границы интервалов для неё поднимаются по сравнению с границами для $\min\limits_{X:|X|=k}\mathsf{P}(X)$. Полностью границы для $\max\limits_{X:|X|=k}\mathsf{P}(X)$ описываются следующей теоремой.

Теорема 2. Для множества из n событий, каждое из которых имеет вероятность p, максимальная по всем подмножествам X из k событий $(2\leqslant k\leqslant n)$ вероятность $\max\limits_{X:|X|=k}\mathsf{P}(X)$ совместного осуществления всех событий подмножества X лежит внутри следующих интервалов:

1) если $np<k-1$, то $0\leqslant\max\limits_{X:|X|=k}\mathsf{P}(X)\leqslant p$;

2) если $k-1\leqslant np<k$, то $\dfrac{np-\lfloor np\rfloor}{C_n^k}\leqslant\max\limits_{X:|X|=k}\mathsf{P}(X)\leqslant p$;

3) если $k\leqslant np$, то $\dfrac{(\lfloor np\rfloor+1-np)C^k_{\lfloor np\rfloor}+(np-\lfloor np\rfloor)C^k_{\lfloor np\rfloor+1}}{C_n^k}\leqslant\max\limits_{X:|X|=k}\mathsf{P}(X)\leqslant p.$

Доказательство.

1) Пусть $np\leqslant k-1$. Здесь возможен случай, когда в каждой точке вероятностного пространства осуществляется не более $k-1$ событий. Поэтому нижняя граница для $\max\limits_{X:|X|=k}\mathsf{P}(X)$ равна нулю.

2) Пусть $k-1\leqslant np<k$. Обозначим через $P_0,P_1,P_2,\ldots,P_n$ вероятности соответственно того, что не осуществится ни одного события, осуществится ровно одно событие, ровно два события и, наконец, ровно n событий. Тогда величины P_i должны удовлетворять следующим очевидным соотношениям:

$$\begin{cases}P_0+P_1+P_2+P_3+\ldots+P_n=1,\\ P_1+2P_2+3P_3+\ldots+nP_n=np,\\ 0\leqslant P_i,\quad i=0,1,2,\ldots,n.\end{cases}\tag{1}$$

Необходимое условие $P_i\leqslant 1$, $i=0,1,2,\ldots,n$, является следствием соотношений (1).

Сумма величин $\mathsf{P}(X)$ по всем подмножествам событий X мощности k может быть найдена как

$$\sum_{X:|X|=k}\mathsf{P}(X)=C_k^kP_k+C_{k+1}^kP_{k+1}+C_{k+2}^kP_{k+2}+\ldots+C_n^kP_n.\tag{2}$$

Теперь, если найти минимальное по всем возможным распределениям значение для величины $\sum\limits_{X:|X|=k} \mathsf{P}(X)$, то для величины $\max\limits_{X:|X|=k} \mathsf{P}(X)$ будет справедливо

$$\max_{X:|X|=k} \mathsf{P}(X) \geqslant \frac{\min\left(\sum\limits_{X:|X|=k} \mathsf{P}(X)\right)}{C_n^k}, \tag{3}$$

причём равенство достигается, когда значения вероятностей $\mathsf{P}(X)$ одинаковы для всех подмножеств X мощности k.

Таким образом, задача нахождения $\max\limits_{X:|X|=k} \mathsf{P}(X)$ свелась к задаче нахождения минимума функционала (2) при ограничениях (1). Но эта последняя является типичной задачей линейного программирования, в которой соотношения (1) являются линейными ограничениями, а выражение (2) — минимизируемым линейным функционалом. В соответствии с основным положением линейного программирования минимальное значение этого функционала достигается на некотором базисном решении, когда не более двух из $n+1$ переменных $P_0, P_1, P_2, \ldots, P_n$ отличны от нуля [7].

Пусть $m = \lfloor np \rfloor$, т. е. m — такое натуральное число, что $m \leqslant np < m+1$. Покажем, что оптимальный базис образуют переменные P_m и P_{m+1}. Для этого выразим эти переменные из двух первых уравнений в (1) через остальные переменные. Имеем

$$\begin{cases} P_m &= m+1-np-(m+1)P_0 - mP_1 - (m-1)P_2 - \ldots \\ &\quad - 2P_{m-1} + P_{m+2} + 2P_{m+3} + \ldots + (n-m-1)P_n, \\ P_{m+1} &= np - m + mP_0 + (m-1)P_1 + (m-2)P_2 + \ldots \\ &\quad + P_{m-1} - 2P_{m+2} - 3P_{m+3} - \ldots - (n-m)P_n. \end{cases} \tag{4}$$

Подставив выражения (4) для переменных P_m и P_{m+1} в линейный функционал (2), выразим этот функционал через остальные (небазисные) переменные. Заметив, что из $k-1 \leqslant np < k$ следует $k = m+1$, получаем

$$\begin{gathered} \sum_{X:|X|=k} \mathsf{P}(X) = C_k^k P_k + C_{k+1}^k P_{k+1} + \ldots + C_n^k P_n = C_{m+1}^{m+1} P_{m+1} + C_{m+2}^{m+1} P_{m+2} + \ldots + C_n^{m+1} P_n = \\ = np - m + mP_0 + (m-1)P_1 + (m-2)P_2 + \ldots + P_{m-1} - \\ -2P_{m+2} - 3P_{m+3} - \ldots - (n-m)P_n + C_{m+2}^{m+1} P_{m+2} + \ldots + C_n^{m+1} P_n = \\ = (np-m) + mP_0 + (m-1)P_1 + (m-2)P_2 + \ldots + P_{m-1} + \\ +(C_{m+2}^{m+1} - 2)P_{m+2} + (C_{m+3}^{m+1} - 3)P_{m+3} + \ldots + \left(C_n^{m+1} - (n-m)\right) P_n = \\ = np - m + \sum_{i=0}^{m-1} (m-i)P_i + \sum_{i=m+2}^{n} \left(C_i^{m+1} - (i-m)\right) P_i. \end{gathered}$$

Покажем, что все стоящие в заключительном выражении перед величинами $P_0, P_1, P_2, \ldots, P_{m-1}, P_{m+2}, \ldots, P_n$ коэффициенты положительны. Тем самым будет показано, что переменные P_m и P_{m+1} образуют оптимальный базис, минимизирующий значение функционала (2).

Коэффициент при P_i, где $0 \leqslant i \leqslant m-1$, равен $(m-i) > 0$.

Коэффициент при P_i, где $m+2 \leqslant i \leqslant n$, равен $C_i^{m+1} - (i-m) > 0$, так как $C_i^{m+1} \geqslant i$ при $i \geqslant m+2$.

Поэтому минимальное значение функционала достигается, когда все они равны нулю. При этом базисные переменные $P_m = m+1-np$, $P_{m+1} = np-m$. Этим показано,

что переменные $P_m = m + 1 - np$ и $P_{m+1} = np - m$ образуют оптимальный базис, а минимальное значение функционала (2) равно $np - m = np - \lfloor np \rfloor$. Теперь из (3) вытекает нижняя оценка теоремы в п. 2.

3) Пусть, наконец, $k \leqslant np$. По-прежнему минимизируем линейный функционал (2) при ограничениях (1). Но теперь $k \leqslant \lfloor np \rfloor = m$. Выразив переменные P_m и P_{m+1} с помощью (4) через остальные переменные и подставив их в линейный функционал (2), получаем

$$\begin{gathered}
\sum_{X:|X|=k} \mathsf{P}(X) = C_k^k P_k + C_{k+1}^k P_{k+1} + \ldots + C_{m-1}^k P_{m-1} + \\
+ C_m^k P_m + C_{m+1}^k P_{m+1} + C_{m+2}^k P_{m+2} + \ldots + C_n^k P_n = \\
= C_k^k P_k + C_{k+1}^k P_{k+1} + \ldots + C_{m-1}^k P_{m-1} + C_m^k (m + 1 - np - (m+1)P_0 - mP_1 - \\
- (m-1)P_2 - \cdots - 2P_{m-1} + P_{m+2} + 2P_{m+3} + \ldots + (n-m-1)P_n) + \\
+ C_{m+1}^k (np - m + mP_0 + (m-1)P_1 + (m-2)P_2 + \ldots + P_{m-1} - \\
- 2P_{m+2} - 3P_{m+3} - \ldots - (n-m)P_n) + \ldots + C_n^k P_n = \\
= (m + 1 - np)C_m^k + (np - m)C_{m+1}^k + (mC_{m+1}^k - (m+1)C_m^k)P_0 + \\
+ ((m-1)C_{m+1}^k - mC_m^k)P_1 + \ldots + ((m-k+1)C_{m+1}^k - (m-k)C_m^k))P_{k-1} + \\
+ (C_k^k + (m-k)C_{m+1}^k - (m-k+1)C_m^k)P_k + \ldots + (C_{m-1}^k + C_{m+1}^k - 2C_m^k)P_{m-1} + \\
+ (C_{m+2}^k - 2C_{m+1}^k + C_m^k)P_{m+2} + \ldots + (C_n^k - (n-m)C_{m+1}^k + (n-m-1)C_m^k)P_n = \\
= (m + 1 - np)C_m^k - (np - m)C_{m+1}^k + \sum_{i=0}^{k-1} \left((m-i)C_{m+1}^k - (m-i+1)C_m^k\right) P_i + \\
+ \sum_{i=k}^{m-1} \left(C_i^k + (m-i)C_{m+1}^k - (m-i+1)C_m^k\right) P_i + \sum_{i=m+2}^{n} \left(C_i^k - (i-m)C_{m+1}^k + (i-m-1)C_m^k\right) P_i.
\end{gathered}$$

Покажем, что все стоящие в заключительном выражении перед величинами P_0, P_1, $P_2, \ldots, P_{m-1}, P_{m+2}, \ldots, P_n$ коэффициенты положительны. Тем самым будет показано, что переменные $P_m = m + 1 - np$ и $P_{m+1} = np - m$ образуют оптимальный базис, минимизирующий значение функционала (2).

Коэффициент при P_i, где $0 \leqslant i \leqslant k-1$, равен

$$\begin{gathered}
(m-i)C_{m+1}^k - (m-i+1)C_m^k = (m-i)(C_{m+1}^k - C_m^k) - C_m^k = \\
= (m-i)C_m^{k-1} - C_m^k = C_m^k \left(k\frac{m-i}{m-k+1} - 1\right) > 0,
\end{gathered}$$

так как $m - i \geqslant m - k + 1$.

Коэффициент при P_i, где $k \leqslant i \leqslant m-1$, равен

$$\begin{gathered}
C_i^k + (m-i)C_{m+1}^k - (m-i+1)C_m^k = C_i^k + (m-i)(C_{m+1}^k - C_m^k) - C_m^k = \\
= C_i^k + (m-i)C_m^{k-1} - C_m^k > 0.
\end{gathered}$$

Чтобы убедиться в справедливости последнего неравенства, рассмотрим систему из двух множеств $B \subset A$, где $|A| = m$, $|B| = i$. Подмножества множества A мощности k, которых C_m^k, или целиком состоят из элементов множества B, или содержат хотя бы один элемент из $A \setminus B$. Но последних подмножеств заведомо меньше чем $(m-i)C_m^{k-1}$, откуда и вытекает данное неравенство.

Коэффициент при P_i, где $m+2 \leqslant i \leqslant n$, равен

$$C_i^k - (i-m)C_{m+1}^k + (i-m-1)C_m^k = C_i^k - (i-m)(C_{m+1}^k - C_m^k) - C_m^k = \\ = C_i^k - (i-m)C_m^{k-1} - C_m^k > 0.$$

Чтобы убедиться в справедливости последнего неравенства, рассмотрим систему из двух множеств $B \subset A$, где $|A| = i$, $|B| = m$. Тогда $(i-m)C_m^{k-1} + C_m^k$ — это число k-элементных подмножеств множества A, имеющих не более одного элемента из множества $A \setminus B$, а C_i^k — полное число k-элементных подмножеств множества A.

Минимальное значение функционала (2), равное $(m+1-np)C_m^k + (np-m)C_{m+1}^k$, достигается, когда все переменные $P_0, P_1, P_2, \ldots, P_{m-1}, P_{m+2}, \ldots, P_n$ равны нулю. Теперь из (3) вытекает нижняя оценка в п. 3. ■

Заключение

Указанные в теоремах 1 и 2 интервалы возможных значений для $\max\limits_{X:|X|=k} \mathsf{P}(X)$ и $\min\limits_{X:|X|=k} \mathsf{P}(X)$ справедливы для любых распределений, дискретных или непрерывных. При этом в классе непрерывных распределений $\max\limits_{X:|X|=k} \mathsf{P}(X)$ и $\min\limits_{X:|X|=k} \mathsf{P}(X)$ могут принимать любое значение внутри этих интервалов.

Рассмотрим, например, случай 3 теоремы 2. Для множества событий $\{A_1, A_2, \ldots, A_n\}$ положим, что в каждой точке вероятностного пространства осуществляется либо ровно m событий, либо ровно $m+1$ событий, где $m = \lfloor np \rfloor$. При этом пусть вероятность осуществления любых m событий и неосуществления остальных $n-m$ равна $(m+1-np)/C_n^m$, а вероятность осуществления любых $m+1$ событий и неосуществления остальных $n-m-1$ равна $(np-m)/C_n^{m+1}$. Тогда вероятность осуществления каждого отдельного события оказывается, как и требуется, равной p, так как

$$C_{n-1}^{m-1}\frac{m+1-np}{C_n^m} + C_{n-1}^m\frac{np-m}{C_n^{m+1}} = \frac{(m+1-np)m}{n} + \frac{(np-m)(m+1)}{n} = p,$$

а вероятность осуществления любых k из n событий равна

$$C_{n-k}^{m-k}\frac{m+1-np}{C_n^m} + C_{n-k}^{m-k+1}\frac{np-m}{C_n^{m+1}} = \frac{(m+1-np)C_m^k + (np-m)C_{m+1}^k}{C_n^k},$$

т. е. величина $\max\limits_{X:|X|=k} \mathsf{P}(X)$ достигает левой границы указанного в теореме интервала. Преобразуя вероятностное пространство с помощью непрерывного преобразования к виду, когда все n событий совпадают, можно получить для $\max\limits_{X:|X|=k} \mathsf{P}(X)$ все значения в интервале от $\left((m+1-np)C_m^k + (np-m)C_{m+1}^k\right)/C_n^k$ до p.

Приведённые оценки для экстремальных вероятностей осуществления k событий могут оказаться полезными в прикладных задачах, использующих вероятностные модели, в которых возможны любые корреляционные связи между событиями. Применённый при доказательстве теоремы 2 технический приём, основанный на сведении задачи к задаче линейного программирования, может быть использован в других подобного рода задачах.

ЛИТЕРАТУРА

1. *Алон Н., Спенсер Дж.* Вероятностный метод. М.: БИНОМ. Лаборатория знаний, 2007. 313 с.

2. *Гнеденко Б. В.* Курс теории вероятностей. 11-е изд. М.: URSS, 2015. 448 с.
3. *Феллер В.* Введение в теорию вероятностей и её приложения. Т. 1. М.: Мир, 1984. 527 с.
4. *Зубков А. М.* Неравенства для распределения числа одновременно происходящих событий // Обозрение прикладной и промышленной математики. 1994. Т. 1. Вып. 4. С. 638–666.
5. *Фролов А. Н.* Об оценивании вероятностей объединений событий с приложениями к лемме Бореля — Кантелли // Вестник СПбГУ. Сер. 1. 2015. Т. 2(60). Вып. 3. С. 387–392.
6. *Колмогоров А. Н.* Основные понятия теории вероятностей. 5-е изд. М.: URSS, 2016. 120 с.
7. *Данциг Дж.* Линейное программирование, его применения и обобщения. М.: Прогресс, 1966. 600 с.

REFERENCES

1. *Alon N. and Spencer J. H.* The Probabilistic Method. N.Y., John Wiley and Sons, 2000. 313 p.
2. *Gnedenko B. V.* Kurs teorii veroyatnostey [The Course of Probability Theory]. Moscow, URSS, 2015. 448 p. (in Russian)
3. *Feller W.* An Introduction to Probability Theory and its Applications, vol. 1. N.Y., John Wiley and Sons, 1970. 527 p.
4. *Zubkov A. M.* Neravenstva dlya raspredeleniya chisla odnovremenno proiskhodyashchikh sobytiy [Inequalities for the distribution of the numer of simultaneosly occuring events]. Obozrenie Prikladnoi i Promyshlennoi Matematiki, 1994, vol. 1, iss. 4, pp. 638–666. (in Russian)
5. *Frolov A. N.* Ob otsenivanii veroyatnostey ob"edineniy sobytiy s prilozheniyami k lemme Borelya — Kantelli [On the evaluation of the probabilities of the unions of events with application to the Borel — Cantelli lemma]. Vestnik SPbGU, 2015, ser. 1, vol. 2(60), iss. 3, pp. 387–392. (in Russian)
6. *Kolmogorov A. N.* Osnovnye ponyatiya teorii veroyatnostey [The Fundamental Conceptions of Probability Theory]. Moscow, URSS, 2016. 120 p. (in Russian)
7. *Dantzig G. B.* Linear programming and extensions. Princeton, Princeton University Press, 1963.

УДК 519.728

ОБ ОДНОМ ИНВАРИАНТЕ В ЗАДАЧЕ РАЗЛОЖЕНИЯ НЕДООПРЕДЕЛЁННЫХ ДАННЫХ

Л. А. Шоломов

ФИЦ «Информатика и управление» РАН, г. Москва, Россия

Рассматривается задача разложения произвольного недоопределённого источника в произведение недоопределённых двоичных источников. Ранее автором было установлено, что точное разложение существует не всегда, но всегда имеется в определённом смысле лучшее аппроксимирующее разложение. Исследуются построение и минимизация аппроксимирующих разложений. Развивается новая техника работы с разложениями на основе связанного с ними графа, названного характеристическим. Доказано, что этот граф является инвариантом равносильных преобразований разложений и полностью определяет класс равносильности. Установлено, что каждому конкретному разложению из этого класса соответствует покрытие характеристического графа системой полных двудольных подграфов, причём это соответствие взаимно однозначно с точностью до некоторой стандартизации разложений. Введённый инвариант, позволяя вместо отдельных разложений иметь дело со всем классом равносильности, значительно расширяет возможности конструктивного исследования разложений. В частности, он даёт подход к решению задачи минимизации разложений, недоступной для предшествующих методов.

Ключевые слова: *недоопределённый источник, декомпозиция, аппроксимация, инвариант равносильных преобразований, характеристический граф разложения, биклика, NP-полнота.*

DOI 10.17223/20710410/39/2

ON AN INVARIANT FOR THE PROBLEM OF UNDERDETERMINED DATA DECOMPOSING

L. A. Sholomov

Federal Research Center "Computer Science and Control" of RAS, Moscow, Russia

E-mail: levshol@mail.ru

The subject of this article is the problem of the best decomposing an arbitrary underdetermined source into a product of underdetermined binary sources. Let $A_0 = \{a_i : i \in M\}$ be a finite alphabet of basic symbols, $\mathcal{T} \subseteq 2^M$, and $A = \{a_T : T \in \mathcal{T}\}$ be an alphabet of underdetermined symbols. Any symbol a_i, $i \in T$, is considered to be a specification of the symbol a_T. The symbol a_M, denoted by $*$, is called indefinite. An underdetermined source X generates symbols $a_T \in A$ independently with probabilities p_T. The entropy of the source X is the quantity

$$\mathcal{H}(X) = \min_Q \left(- \sum_{T \in \mathcal{T}} p_T \log_2 \sum_{i \in T} q_i \right)$$

where the minimum is taken over the set of probability vectors $Q = (q_i,\ i \in M)$. Underdetermined sources X and Y are equivalent ($X \approx Y$) if $\mathcal{H}(XZ) = \mathcal{H}(YZ)$ for

any source Z. The source X is not weaker than Y ($X \succsim Y$) if $XY \eqsim X$. For the underdetermined source X and a natural s, we assign to each $T \in \mathcal{T}$ some vector $\lambda_T \in \in \{0, 1, *\}^s$. If the component i ($i = 1, \ldots, s$) of the vector is considered as the output value of some source X_i, then the product $X_1 \ldots X_s$ of underdetermined binary sources arises. It's called decomposition associated with the source X. The decomposition $X_1 \ldots X_s$ is considered to be exact if $X_1 \ldots X_s \eqsim X$. The decomposition is called (lower) approximation of the source X if $X_1 \ldots X_s \precsim X$ and, for any connected with X decomposition $Z_1 \ldots Z_t$ such that $Z_1 \ldots Z_t \precsim X$, the relation $X_1 \ldots X_s \succsim Z_1 \ldots Z_t$ is fulfilled. It is known that the exact decomposition of a source may not exist, but there is always an approximation that is unique up to equivalence. If the exact decomposition exists, then the approximation coincides with it. The aim of this paper is to construct approximations of minimal complexity (i.e., with the smallest number of factors).

The decomposition $X_1 \ldots X_s$ is represented by a matrix Λ consisting of the columns λ_T, $T \in \mathcal{T}$. For the matrix Λ, we construct a graph $G_\chi(\Lambda)$ with the set of vertices $\mathcal{T}$. The graph is called the characteristic graph of Λ. In it, the vertices T and T' are connected by the edge (T, T') if and only if the vectors λ_T and $\lambda_{T'}$ are inconsistent, i.e. do not allow a common specification. It is proved that decompositions are equivalent if and only if the characteristic graphs of their matrices coincide. It means that the characteristic graph is the invariant with respect to equivalent transformations of decompositions and completely determines a equivalence class of the decompositions. For the equivalence class formed by approximations of the given source X, the characteristic graph $G_\chi^{\text{app}}(X)$ is also defined. The graph $G_\chi^{\text{app}}(X)$ is formed by all edges (T, T') such that $T \cap T' = \varnothing$. All decompositions belonging to a given equivalence class are described too. It is proved that there exists a one-to-one correspondence between the decompositions (in some standard form) and coverings of the characteristic graph G_χ with complete bipartite subgraphs. In particular, the approximate decompositions of the least complexity correspond to minimal coverings of $G_\chi^{\text{app}}(X)$. The construction problem for approximating decompositions with minimal complexity turned out to be NP-complete. The article describes a rather economical way to solve this problem for tasks having not too large dimension.

Keywords: *underdetermined source, decomposition, approximation, invariant of equivalent transformations, characteristic graph of approximation, biclique, NP-completeness.*

Введение

Данная работа имеет дело с недоопределёнными данными — последовательностями недоопределённых символов. Каждому такому символу соответствует некоторое множество основных (полностью определённых) символов, любым из которых он может быть замещен (доопределён). Считается, что недоопределённые последовательности порождаются источником, генерирующим недоопределённые символы независимо с некоторыми вероятностями. При использовании недоопределённых данных часто бывает достаточно вместо самих данных иметь их доопределения. Эти ослабленные требования к данным предоставляют дополнительные возможности, одной из которых является возможность нетривиальных равносильных преобразований данных (подробнее см. в [1]).

В работе изучается задача разложения недоопределённых источников общего вида в произведение недоопределённых двоичных источников. Эта задача рассматривалась в работе [2], где установлено, что точное разложение, при котором произведение

равносильно исходному источнику, существует не всегда, но всегда имеется лучшее (в некотором заданном смысле) аппроксимирующее разложение. Для источников, допускающих точное разложение, аппроксимирующее разложение является точным. В [2] описан эффективный (полиномиальный) алгоритм построения аппроксимирующего разложения источника и представлен ряд эффективных методов, относящихся к проверке равносильности и упрощению разложений. Но подход, развитый в [2], позволяет производить лишь некоторые локальные упрощения разложений и не обеспечивает возможности построения равносильных разложений, минимальных по сложности.

Предлагаемая работа устраняет этот недостаток. В ней найден и исследован некоторый инвариант равносильных преобразований разложений. Он представляет собой граф, который в явном виде (и очень просто) выписывается по разложению. Этот граф, названный характеристическим графом разложения, обладает тем свойством, что разложения равносильны тогда и только тогда, когда их характеристические графы совпадают. Таким образом, характеристический граф описывает не отдельное разложение, а весь класс равносильных разложений и его можно считать характеристическим графом класса.

Установлено, что каждое конкретное разложение из класса равносильности соответствует некоторому покрытию характеристического графа системой полных двудольных подграфов (биклик), причём соответствие между разложениями источника и покрытиями характеристического графа бикликами взаимно однозначно с точностью до некой стандартизации разложений. В частности, разложение, имеющее наименьшую сложность, соответствует минимальному покрытию характеристического графа. Это даёт принципиальную возможность построения разложений минимальной сложности посредством решения задачи наилучшего покрытия характеристического графа бикликами. Задача построения наилучшего по сложности равносильного разложения оказалась NP-полной, что исключает возможность её эффективного решения (при условии NP≠P). Описан достаточно экономный метод решения этой задачи, применимый к задачам небольшой размерности.

Всё сказанное выше относится к произвольным классам равносильности разложений. Но основной интерес представляет класс аппроксимирующих разложений заданного источника. В работе найден простой способ построения характеристического графа этого класса и все задачи, связанные с нахождением и минимизацией аппроксимирующих разложений источника, решаются посредством операций над этим графом.

Использование введённого инварианта упростило решение задач, связанных с разложением недоопределённых данных и заметно расширило возможности методов, развитых в [2]. Кроме того, в терминах этого инварианта получили явную формулировку некоторые важные свойства разложений, для которых в [2] установлена лишь возможность полиномиальной проверки.

1. Недоопределённые источники. Равносильность

Считаем, что задан конечный алфавит $A_0 = \{a_i : i \in M\}$ *основных* символов. Каждому непустому $T \subseteq M$ поставлен в соответствие символ a_T, называемый *недоопределённым*. *Доопределением символа* a_T считается всякий основной символ a_i, $i \in T$. Символ a_M, доопределимый любым основным символом, называется *неопределённым* и обозначается $*$. Будем говорить, что символ a_T *чётче* символа $a_{T'}$ ($a_{T'}$ *размытее* символа a_T), если $T \subseteq T'$.

Пусть выделена система $\mathcal{T} \subseteq 2^M$ некоторых непустых подмножеств T множества M и ей сопоставлен *недоопределённый алфавит* $A = A_{\mathcal{T}} = \{a_T : T \in \mathcal{T}\}$. Источник X,

порождающий символы $a_T \in A$ независимо с вероятностями p_T, $\sum\limits_{T\in\mathcal{T}} p_T = 1$, будем обозначать (A, P), где $P = (p_T, T \in \mathcal{T})$. Дальше будем считать, что все символы $a_T \in A$ имеют строго положительные вероятности, ибо при $p_T = 0$ символ a_T можно удалить из A (а множество T — из $\mathcal{T}$). Источник (A, P) будем называть *недоопределённым*, а в случае $A = A_0$ — *полностью определённым*.

Задавшись некоторым набором вероятностей $Q = (q_i,\ i \in M)$ символов $a_i \in A_0$, введём функцию

$$\mathcal{H}(P, Q) = -\sum_{T\in\mathcal{T}} p_T \log \sum_{i\in T} q_i$$

(здесь и дальше логарифмы двоичные). *Энтропией источника* X назовём величину

$$\mathcal{H}(X) = \min_Q \mathcal{H}(P, Q).$$

В задаче кодирования источников [3] эта величина играет для недоопределённых источников роль, подобную той, какую для полностью определённых играет энтропия Шеннона (подробнее см. в [4]). Распространение понятия энтропии на случай стационарных недоопределённых источников имеется в [5].

Недоопределённые источники X и Y будем называть *равносильными* [2, 6] (и записывать $X \approx Y$), если для любого источника Z выполнено[1]

$$\mathcal{H}(XZ) = \mathcal{H}(YZ).$$

Другие способы введения того же понятия равносильности предложены в [1]. Одной из интерпретаций этого понятия является следующая [7]. Если рассматривать кодирования, которые по коду недоопределённой последовательности позволяют восстановить какое-либо её доопределение, то недоопределённые источники равносильны тогда и только тогда, когда всякое кодирование одного из них является также кодированием второго (при другом способе декодирования). Отсюда следует, что оптимальное кодирование одного источника оптимально и для второго.

Введём также отношение $X \gtrsim Y$ (источник X *не слабее* Y), положив

$$X \gtrsim Y \iff XY \approx X. \tag{1}$$

В [2] установлено, что отношения $\approx$ и $\gtrsim$ являются соответственно эквивалентностью и частичным порядком (нестрогим) и что они связаны соотношением

$$X \approx Y \iff (X \gtrsim Y) \wedge (Y \gtrsim X). \tag{2}$$

Источники $X = (A, P)$ и $Y = (B, P)$ будем называть *идентичными*, если они различаются лишь обозначениями основных символов. Более точно, если основным и недоопределённым алфавитами источника X являются $A_0 = \{a_i : i \in M\}$ и $A = \{a_T : T \in \mathcal{T}\}$, то соответствующие алфавиты источника Y имеют вид $B_0 = \{b_i : i \in M\}$ и $B = \{b_T : T \in \mathcal{T}\}$, причём символы a_T и b_T порождаются с одинаковыми вероятностями p_T и имеют совместное распределение $p(a_T, b_T) = p_T$ (а потому $p(a_T, b_{T'}) = 0$ при $T \neq T'$).

[1] Когда говорится о взаимоотношении нескольких источников, считается, что задано их совместное распределение. В частности, в данном определении предполагается заданным распределение для XYZ.

Скажем, что символ a_i *мажорирует* в источнике $X = (A, P)$ символ a_j, если для любого $a_T \in A$ вхождение $j \in T$ влечёт $i \in T$. Операция *исключения мажорируемого символа* a_j производит замену символов a_T на $a_{T\setminus\{j\}}$. В результате возникает источник $X' = (A', P')$, для которого $\mathcal{T}' = \{T' : T' = T \setminus \{j\}, T \in \mathcal{T}\}$, $A' = \{a_{T'} : T' \in \mathcal{T}'\}$, $P' = (p'_{T'}, T' \in \mathcal{T}')$, $p'_{T'} = p_{T'} + p_{T'\cup\{j\}}$ (в случае $a_{T'} \notin A$ считается $p_{T'} = 0$; для символа $a_{T'\cup\{j\}}$ аналогично).

Источник, в котором отсутствуют мажорируемые символы, называется *приведённым*. По всякому источнику путём последовательного исключения (в произвольном порядке) мажорируемых символов может быть построен приведённый источник. Дальше понадобятся следующие факты.

Утверждение 1 [2, 6].

1. В результате исключения из источника мажорируемого символа возникает равносильный источник.
2. Разные приведённые источники, образованные из заданного последовательным исключением мажорируемых символов, идентичны.
3. Источники X и Y равносильны тогда и только тогда, когда соответствующие им приведённые источники X' и Y' идентичны.

2. Разложения

Пусть имеется недоопределённый источник $X = (A, P)$, $A = A_{\mathcal{T}}$. Задавшись натуральным числом s, сопоставим каждому символу $a_T \in A$ некоторый набор $\lambda_T = (\lambda_T(1), \ldots, \lambda_T(s)) \in \{0, 1, *\}^s$. Если рассматривать $\lambda_T(i)$, $T \in \mathcal{T}$, как значения выхода некоторого источника X_i, $i = 1, \ldots, s$, то возникнет произведение $X_1 \ldots X_s$ недоопределённых двоичных источников (т. е. источников в алфавите $\{0, 1, *\}$), которое будем называть *разложением*, связанным с источником X. В этом разложении набор λ_T значений выходов источников $X_1, \ldots, X_s$ имеет вероятность p_T. На вид разложений каких-либо ограничений не накладывается. В частности, различным T могут соответствовать одинаковые λ_T.

Выделим некоторые специальные типы разложений, связанных с источником X. Разложение $X_1 \ldots X_s$ назовём *точным*, если $X_1 \ldots X_s \eqsim X$. Точные разложения существуют не для всех источников, поэтому будем рассматривать также некоторые приближения. Разложение $X_1 \ldots X_s$ назовём *нижней аппроксимацией* источника X, если $X_1 \ldots X_s \precsim X$ и для всякого связанного с X разложения $Z_1 \ldots Z_t$, такого, что $Z_1 \ldots Z_t \precsim X$, выполнено $X_1 \ldots X_s \succsim Z_1 \ldots Z_t$. В [2] доказано, что для любого источника существует нижняя аппроксимация. Все нижние аппроксимации одного источника равносильны, ибо если $X_1 \ldots X_s$ и $Y_1 \ldots Y_t$ — две нижние аппроксимации, то $X_1 \ldots X_s \succsim Y_1 \ldots Y_t$ и $Y_1 \ldots Y_t \succsim X_1 \ldots X_s$, а потому в силу (2) выполнено $X_1 \ldots X_s \eqsim Y_1 \ldots Y_t$.

Двойственно (заменой соотношений $\precsim$ на $\succsim$ и наоборот) можно определить *верхнюю аппроксимацию*. Верхние аппроксимации в классе разложений существуют не для всех источников, поэтому дальше будем вести речь только о нижних аппроксимациях, которые будем называть просто *аппроксимациями*. Соответствующие разложения будем называть *аппроксимирующими*. Отметим, что если у источника существует точное разложение, то всякое аппроксимирующее разложение является точным.

Будем использовать представление разложений с помощью матриц. Положим $n = |\mathcal{T}|$ и $N = \{1, \ldots, n\}$. Произвольно занумеруем числами $j \in N$ множества T системы $\mathcal{T}$. Вместо λ_T и $\lambda_T(i)$ будем использовать обозначения λ_j и $\lambda_j(i)$, где j — номер множества $T = T_j$. Разложению $X_1 \ldots X_s$ сопоставим $(s \times n)$-матрицу $\Lambda = \|\lambda_j(i)\|$

с s строками $\lambda(i) = (\lambda_j(i), j \in N)$, соответствующими источникам X_i, и n столбцами λ_j (точнее, столбцами являются транспонированные наборы λ_j). Чтобы превратить матрицу в разложение, следует приписать каждому столбцу λ_j вероятность $p_{T_j} > 0$.

Разложение $X_1 \dots X_s$ представляет собой источник, недоопределённый алфавит которого образован столбцами матрицы Λ, а основной алфавит — всеми доопределениями столбцов. Вероятность недоопределённого символа равна сумме вероятностей, приписанных столбцам, с которыми он совпадает.

Процедура приведения источников фактически не зависит от значений приписанных его символам ненулевых вероятностей, а потому в силу п. 3 утверждения 1 от них не зависит понятие равносильности источников и можно говорить о равносильности алфавитов (подробнее см. в [1]). То же относится к разложениям; их равносильность определяется только матрицами разложений и не зависит от значений ненулевых вероятностей, приписанных столбцам. *Матрицы* назовём *равносильными*, если соответствующие им разложения равносильны. Для обозначения равносильности матриц Λ и Λ' будем применять запись $\Lambda \eqsim \Lambda'$. Отношение $\gtrsim$ (не слабее), определённое посредством (1), может быть переформулировано применительно к матрицам в виде

$$\Lambda \gtrsim \Lambda' \Longleftrightarrow \begin{bmatrix} \Lambda \\ \Lambda' \end{bmatrix} \eqsim \Lambda, \tag{3}$$

где через $\begin{bmatrix} \Lambda \\ \Lambda' \end{bmatrix}$ обозначена матрица, полученная записыванием друг под другом строк матриц Λ и Λ'. Свойства отношений $\eqsim$ и $\gtrsim$, доказанные для источников, переносятся на матрицы.

Будем рассматривать преобразования матриц, выполняемые посредством операций над их строками. Преобразование назовём *равносильным*, если в применении к любой матрице разложения оно даёт равносильную матрицу. Систему равносильных преобразований назовём *полной*, если для любых двух равносильных матриц Λ и Λ' в ней имеется последовательность преобразований, переводящих Λ в Λ'.

Распространим булевы операции конъюнкции, дизъюнкции и инверсии (отрицания) на множество $\{0, 1, *\}$, дополнив их соотношениями

$$\begin{gathered} 0 \wedge * = * \wedge 0 = 0, \quad 1 \wedge * = * \wedge 1 = *, \quad * \wedge * = *, \\ 1 \vee * = * \vee 1 = 1, \quad 0 \vee * = * \vee 0 = *, \quad * \vee * = *, \\ \overline{*} = *. \end{gathered}$$

Введём операции $\lambda(i) \vee \lambda(u)$, $\lambda(i) \wedge \lambda(u)$ и $\bar{\lambda}(i)$ над строками, при выполнении которых соответствующие операции производятся над разрядами строк. Кроме того, введём операцию размытия строки $\lambda(i)$, осуществляемую путём замены некоторых из её символов $\lambda_j(i)$ более размытыми.

Утверждение 2 [2]. Следующие преобразования образуют полную систему равносильных преобразований матриц разложений.

1°) Добавление либо удаление строки, составленной только из нулей или из единиц.
2°) Добавление либо удаление инверсии некоторой строки.
3°) Добавление либо удаление конъюнкции двух строк.
4°) Добавление либо удаление дизъюнкции двух строк.
5°) Добавление либо удаление строки, полученной размытием некоторой строки матрицы.

Приведём некоторые производные от 1°–5° равносильные преобразования матриц, которые понадобятся дальше.

Инвертирование строки, т. е. замена строки $\lambda(i)$ на $\bar{\lambda}(i)$. Эта операция может быть выполнена двойным применением правила 2°: сначала добавляется строка $\bar{\lambda}(i)$, а затем удаляется $\lambda(i)$ как инверсия строки $\bar{\lambda}(i)$.

Добавление либо удаление конъюнкции или дизъюнкции нескольких строк. Обоснуем, например, допустимость операции удаления дизъюнкции трёх строк. Пусть в матрице присутствуют строки $\lambda(i)$, $\lambda(u)$, $\lambda(v)$ и $\lambda(i) \vee \lambda(u) \vee \lambda(v)$. Тогда можно в соответствии с правилом 4° добавить $\lambda(i) \vee \lambda(u)$, затем удалить $\lambda(i) \vee \lambda(u) \vee \lambda(v)$, после чего удалить $\lambda(i) \vee \lambda(u)$.

Операции 1-расщепления и 0-расщепления. Обозначим через $J_\sigma(i)$, $\sigma \in \{0, 1\}$, множество тех j, для которых $\lambda_j(i) = \sigma$. Со строкой $\lambda(i)$ свяжем два семейства строк $\{\lambda_{\bar{\sigma},j}(i) : j \in J_\sigma(i)\}$, $\sigma = 0, 1$, где $\lambda_{\bar{\sigma},j}(i)$ означает строку, в которой все значения $\bar{\sigma}$ расположены в тех же позициях, что и в строке $\lambda(i)$, в позиции j находится значение σ, остальные разряды содержат $*$. Операция σ-расщепления производит замену строки $\lambda(i)$ семейством $\{\lambda_{\bar{\sigma},j}(i) : j \in J_\sigma(i)\}$. Допустимость операции 1-расщепления вытекает из того, что она может быть выполнена добавлением к матрице строк $\lambda_{0,j}(i)$, $j \in J_1(i)$, получающихся размытием строки $\lambda(i)$, и последующим удалением строки $\lambda(i)$ как дизъюнкции добавленных строк. Операция 0-расщепления обосновывается аналогично, но роль дизъюнкции строк играет конъюнкция.

3. Характеристический граф разложений

Как и раньше, рассматриваем матрицы Λ с n столбцами, где n — мощность недоопределённого алфавита. Столбцы λ_u и λ_v назовём *несовместными*, если они не могут быть доопределены до одного и того же столбца, т. е. в некотором разряде i значения $\lambda_u(i)$ и $\lambda_v(i)$ определены и различны. *Характеристическим графом* матрицы Λ назовем обыкновенный (т. е. неориентированный, без петель и кратных рёбер) граф с n вершинами, нумерованными числами $1, \ldots, n$, в котором вершины u и v соединены ребром (u, v) тогда и только тогда, когда столбцы λ_u и λ_v несовместны. Случай пустого графа (т. е. графа с n вершинами и пустым множеством рёбер) не исключается. Характеристический граф матрицы Λ будем обозначать через $G_\chi(\Lambda)$.

Следующее утверждение показывает, что характеристический граф инвариантен относительно равносильных преобразований матриц (и разложений).

Теорема 1. Матрицы разложений Λ и Λ' равносильны тогда и только тогда, когда их характеристические графы $G_\chi(\Lambda)$ и $G_\chi(\Lambda')$ совпадают.

Доказательство теоремы использует ряд лемм.

Лемма 1. Характеристические графы равносильных матриц совпадают.

Доказательство. Достаточно убедиться, что преобразования 1°–5° из утверждения 2 сохраняют характеристический граф.

1. При добавлении или удалении строки-константы во всех столбцах матрицы в одной и той же позиции возникает либо исчезает одна и та же константа. Это не влияет на совместность столбцов и не изменяет характеристический граф.

2. Инвертированная строка дублирует вклад исходной строки в множество несовместностей. Поэтому при добавлении либо удалении инвертированной строки характеристический граф сохраняется.

3. Пусть дизъюнкция $\lambda(i) \vee \lambda(j)$ обеспечивает несовместность столбцов λ_u и λ_v и $\lambda_u(i) \vee \lambda_u(j) = 0$, $\lambda_v(i) \vee \lambda_v(j) = 1$. Тогда $\lambda_u(i) = \lambda_u(j) = 0$ и какое-либо из значений $\lambda_v(i)$

и $\lambda_v(j)$ равно 1. Это означает, что дизъюнкция $\lambda(i) \vee \lambda(j)$ дублирует несовместность, даваемую одной из строк $\lambda(i)$ и $\lambda(j)$, и при добавлении либо удалении дизъюнкции строк характеристический граф не изменяется.

4. Случай добавления либо удаления конъюнкции строк рассматривается аналогично предыдущему.

5. Если строка является результатом размытия некоторой строки $\lambda(i)$, то все связанные с ней несовместности содержатся среди тех, которые обеспечиваются строкой $\lambda(i)$. Поэтому добавление или удаление более размытой строки не оказывает влияния на характеристический граф. ■

Будем различать матрицы разложений с точностью до перестановки строк. Разложение назовём *каноническим*, если его матрица имеет один из следующих видов:

(a) состоит из единственной строки $00\ldots0$;

(b) составлена из попарно различных строк, каждая из которых содержит по одному разу элементы 0 и 1 (остальными являются $*$), причём 0 расположен в строке раньше 1.

Лемма 2. Произвольному графу G (на n вершинах) соответствует единственная матрица канонического разложения, характеристический граф которой совпадает с G.

Доказательство. Пустому графу G соответствует единственное каноническое разложение, имеющее тип (a), ибо характеристические графы всех канонических разложений типа (b) непусты.

Если характеристический граф некоторого канонического разложения не пуст, то разложение имеет тип (b) и существует взаимно однозначное соответствие между рёбрами этого графа и строками матрицы разложения: ребру (u, v), $u < v$, соответствует строка

$$*\ldots*0*\ldots*1*\ldots*,$$

где 0 расположен в позиции u, а 1 — в позиции v. Поэтому по произвольному графу G, принятому в качестве характеристического, каноническое разложение находится однозначно. ■

Лемма 3. Для всякой матрицы разложения существует единственная равносильная ей матрица канонического разложения.

Доказательство.

1. Покажем, что для всякой матрицы существует равносильная матрица канонического разложения.

(a) Рассмотрим сначала случай, когда в каждой строке матрицы присутствуют булевы элементы не более одного типа.

Проинвертировав строки, содержащие элементы 1, получим матрицу, образованную элементами 0 и $*$. По правилу 1° добавим строку $00\ldots0$ и с помощью 5° удалим все другие строки. В результате получим матрицу с единственной строкой $00\ldots0$.

(b) Рассмотрим теперь случай, когда в матрице разложения имеется строка, в которой одновременно присутствуют 0 и 1.

Сначала устраним все строки, содержащие булевы элементы одного типа. Для этого, как в п. (a), преобразуем совокупность всех таких строк в строку $00\ldots0$ и удалим последнюю по правилу 1°.

Затем к каждой из оставшихся строк применим операцию 1-расщепления. Получим множество строк, любая из которых содержит некоторое количество 0 и единственное значение 1. Далее, выполнив для каждой из них операцию 0-расщепления, образуем

матрицу, во всех строках которой значения 0 и 1 встречаются по одному разу. Проинвертировав строки, в которых 1 предшествует значению 0, и удалив с помощью 5° повторяющиеся строки, придём к матрице канонического разложения, эквивалентной исходной матрице.

2. Матрица канонического разложения, равносильная заданной матрице, единственна, ибо матрицы канонических разложений, равносильные заданной матрице, имеют по лемме 1 одинаковые характеристические графы и в соответствии с леммой 2 обязаны совпасть. ■

Лемма 4. Если матрицы разложений имеют одинаковые характеристические графы, то они равносильны.

Доказательство. Пусть матрицы Λ и Λ' имеют одинаковые характеристические графы. Для матриц Λ и Λ' рассмотрим единственные соответствующие им канонические разложения (лемма 2). По лемме 1 они имеют те же характеристические графы и потому по лемме 2 совпадают. Матрицы Λ и Λ', будучи равносильными одной и той же матрице (канонического) разложения, равносильны. ■

Теорема 1 вытекает из лемм 1 и 4.

Отношение равносильности матриц является эквивалентностью и потому множество матриц разложения разбивается на классы равносильности. Согласно теореме 1, каждому классу равносильности можно сопоставить граф, обладающий тем свойством, что матрица принадлежит классу, если и только если её характеристический граф совпадает с этим графом. Будем называть его *характеристическим графом класса.*

Следствие 1. Существует взаимно однозначное соответствие между классами равносильности матриц разложения с n столбцами и графами на n вершинах.

Доказательство. В силу сказанного выше, достаточно убедиться, что всякому графу соответствует некоторый класс равносильности матриц. Но это так, поскольку по лемме 2 каждый граф является характеристическим графом некоторого (в частности, канонического) разложения. ■

В терминах характеристических графов может быть выражено отношение $\succsim$ (не слабее) для матриц и разложений. В теоретико-множественных соотношениях с участием графов под G понимается множество рёбер графа G.

Следствие 2. Отношение $\Lambda \succsim \Lambda'$ имеет место тогда и только тогда, когда $G_\chi(\Lambda) \supseteq G_\chi(\Lambda')$.

Доказательство. С учётом равенства $G_\chi\left(\begin{bmatrix}\Lambda\\ \Lambda'\end{bmatrix}\right) = G_\chi(\Lambda) \cup G_\chi(\Lambda')$ и теоремы 1 соотношение 3 может быть переписано в виде $\Lambda \succsim \Lambda' \iff G_\chi(\Lambda) \cup G_\chi(\Lambda') = G_\chi(\Lambda)$. Отсюда следует, что необходимым и достаточным условием для $\Lambda \succsim \Lambda'$ является $G_\chi(\Lambda) \supseteq G_\chi(\Lambda')$. ■

Характеристический граф оказывается полезным в задаче упрощения разложений, где под упрощением понимается сокращение числа сомножителей в разложениях, а в терминах матриц — сокращение числа строк.

Для $I \subseteq \{1, \ldots, s\}$ обозначим через $\Lambda(I)$ множество строк $\lambda(i)$, $i \in I$, а через $\Lambda(\bar{I})$ — множество оставшихся строк. Поскольку мы различаем матрицы с точностью до перестановок строк, те же обозначения будем использовать и для матриц, составленных из этих строк. Множество строк $\Lambda(I)$ назовём *устранимым* из Λ, если $\Lambda(\bar{I}) \approx \Lambda$.

Следствие 3. Множество строк $\Lambda(I)$ устранимо из Λ тогда и только тогда, когда $G_\chi(\Lambda(I)) \supseteq G_\chi(\Lambda(\bar{I}))$.

Доказательство. Вытекает из следствия 2 и того, что условие устранимости множества строк $\Lambda(I)$ совпадает с условием $\Lambda(\bar{I}) \succsim \Lambda(I)$ для матриц. ■

Теорема 1 и следствия 2 и 3 усиливают соответствующие результаты работы [2], поскольку дают явную формулировку условий, для которых в [2] установлена лишь возможность полиномиальнй проверки.

4. Характеристический граф аппроксимаций

Пусть требуется найти аппроксимирующее разложение заданного недоопределённого источника X (оно является точным, если точное разложение существует). Множество разложений, аппроксимирующих источник X, образует класс равносильности, характеристический граф которого обозначим через $G_\chi^{\mathrm{app}}(X)$. Цель данного пункта — построение графа $G_\chi^{\mathrm{app}}(X)$. Способ нахождения по графу самих аппроксимирующих разложений изложен в п. 5.

В работе [2] предложен эффективный (полиномиальный) алгоритм построения матрицы аппроксимирующего разложения заданного недоопределённого источника

$$X = (A, P), \quad A = A_{\mathcal{T}}, \quad \mathcal{T} = \{T_1, \ldots, T_n\}. \tag{4}$$

Утверждение 3 [2]. Аппроксимирующее разложение источника (4) задаётся матрицей $\Lambda = \|\lambda_j(i)\|$ со строками $\lambda(i)$ и столбцами λ_j, $i, j \in N$, где

$$\lambda_j(i) = \begin{cases} 1, & \text{если } T_j \subseteq T_i, \\ 0, & \text{если } T_j \cap T_i = \varnothing, \\ * & \text{в остальных случаях.} \end{cases}$$

Отсюда и из утверждения 3 получаем явное описание графа $G_\chi^{\mathrm{app}}(X)$.

Теорема 2. Характеристический граф $G_\chi^{\mathrm{app}}(X)$ аппроксимирующих разложений источника (4) образован всеми такими рёбрами (u, v), $u, v \in N$, что $T_u \cap T_v = \varnothing$.

Доказательство. Граф $G_\chi^{\mathrm{app}}(X)$ совпадает с графом $G_\chi(\Lambda)$ матрицы Λ, представленной в утверждении 3. Ребро (u, v) принадлежит графу $G_\chi(\Lambda)$ тогда и только тогда, когда столбцы λ_u и λ_v несовместны, т. е. когда найдётся строка $\lambda(i)$, элементы $\lambda_u(i)$ и $\lambda_v(i)$ которой являются булевыми и различными. Пусть, для определённости, $\lambda_u(i) = 1$, $\lambda_v(i) = 0$. Тогда $T_i \supseteq T_u$ и $T_i \cap T_v = \varnothing$, откуда $T_u \cap T_v = \varnothing$.

Обратно, пусть $T_u \cap T_v = \varnothing$. Рассмотрим строку $\lambda(u)$. В ней $\lambda_u(u) = 1$, $\lambda_v(u) = 0$, а потому столбцы λ_u, λ_v несовместны и $(u, v) \in G_\chi(\Lambda)$. ■

Напомним некоторые сведения из теории графов [8]. Всякий граф G изоморфен *графу пересечений*, т. е. обладает следующим свойством. Если всеми вершинами графа G являются $1, \ldots, n$, то существует система множеств $\mathcal{T} = \{T_1, \ldots, T_n\}$, такая, что граф G содержит ребро (u, v) тогда и только тогда, когда $T_u \cap T_v \neq \varnothing$. В качестве T_i $(i = 1, \ldots, n)$ можно использовать множество, образованное вершиной i и всеми рёбрами, инцидентными i. Если $n \geqslant 3$ и граф G связен, то при его реализации в виде графа пересечений достаточно в качестве T_i $(i = 1, \ldots, n)$ взять множество всех рёбер, инцидентных вершине i.

Покажем, что в качестве характеристических графов при аппроксимациях источников с n-буквенным недоопределённым алфавитом могут возникнуть любые графы на n вершинах.

Теорема 3. Любой граф G является характеристическим графом $G_\chi^{\text{app}}(X)$ аппроксимирующих разложений некоторого источника X.

Доказательство. Возьмём дополнение $\bar{G}$ графа G и реализуем $\bar{G}$ в виде графа пересечений. Пусть при этом возникает система множеств $\mathcal{T} = \{T_1, \ldots, T_n\}$. Обозначим через X источник (4), в котором в качестве $\mathcal{T}$ использована построенная система. Поскольку выполнено

$$(u, v) \in G \Leftrightarrow (u, v) \notin \bar{G} \Leftrightarrow T_u \cap T_v = \varnothing,$$

то граф G в силу теоремы 2 совпадает с $G_\chi^{\text{app}}(X)$. ■

Пример 1. Пусть требуется указать недоопределённый источник X, для которого $G_\chi^{\text{app}}(X)$ совпадает с графом G, представленным на рис. 1,*а*. Обозначим через G' граф на пяти вершинах, полученный из G удалением изолированной вершины 4, а через $\bar{G}'$ — дополнение последнего. Граф $\bar{G}'$ (рис. 1,*б*) связен. Занумеровав $e_1 = (1, 2)$, $e_2 = (1, 3)$, $e_3 = (1, 6)$, $e_4 = (2, 6)$, $e_5 = (3, 5)$ рёбра графа $\bar{G}'$ и используя в множествах T_i вместо рёбер их номера, получим систему множеств $T_1 = \{1, 2, 3\}$, $T_2 = \{1, 4\}$, $T_3 = \{2, 5\}$, $T_5 = \{5\}$, $T_6 = \{3, 4\}$, которая, наряду с $T_4 = \{1, 2, 3, 4, 5\}$, задаёт алфавит источника X. Вероятности $p_{T_i} > 0$ могут быть назначены произвольно. Заметим, что удаление изолированной вершины 4 не предусмотрено конструкцией теоремы 3 и не является обязательным, но позволяет упростить результат.

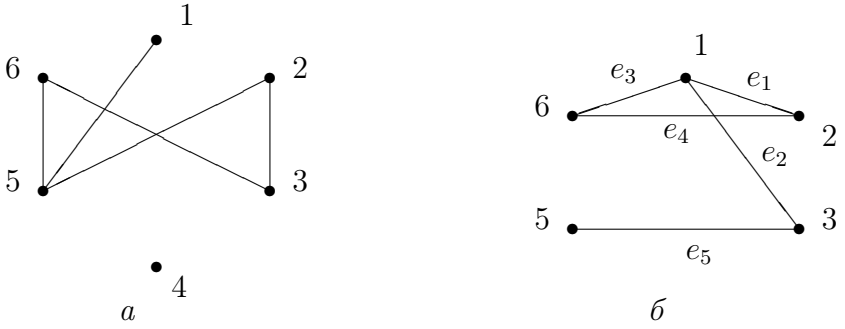


Рис. 1. Графы G (*а*) и $\bar{G}'$ (*б*)

5. Построение разложений на основе характеристического графа

Характеристический граф инвариантен относительно равносильных преобразований разложений и полностью определяет класс равносильности. Рассмотрим построение разложений, имеющих заданный характеристический граф.

Строку матрицы разложений будем считать тривиальной, если в неё входят булевы элементы не более одного из типов 0 и 1. Нетривиальную *строку* назовем *стандартной*, если самым левым булевым элементом строки является 0. Матрицу разложения будем считать нетривиальной, если в ней имеется хотя бы одна нетривиальная строка. *Матрицу* назовем *стандартной*, если все её строки стандартны и различны. Всякая нетривиальная матрица может быть эффективно преобразована в эквивалентную стандартную матрицу. Для этого сначала следует, как в лемме 3, исключить тривиальные строки, затем проинвертировать строки, в которых самым левым булевым элементом является 1, после чего удалить с помощью 5° повторяющиеся строки.

Тривиальные матрицы разложений, и только они, обладают пустым характеристическим графом. Дальше будем считать граф непустым и матрицу нетривиальной. Более того, поскольку при стандартизации матриц число строк не увеличивается (а

мы будем стремиться к уменьшению числа строк), ограничимся рассмотрением стандартных матриц.

Напомним некоторые понятия из теории графов. Граф называется *двудольным*, если множество его вершин можно разбить на два непустых подмножества (доли) так, что каждое ребро графа соединяет вершины из разных долей. *Двудольный граф* называется *полным*, если каждая пара вершин, принадлежащих разным долям, соединена ребром. Всякий полный двудольный подграф графа называется его *бикликой*.

Введём отображение F, сопоставляющее всякому полному двудольному графу H с не более чем n вершинами, которым приписаны числа (номера), образующие некоторое подмножество множества $N = \{1, \ldots, n\}$, стандартную строку $F(H) = \lambda = (\lambda_1, \ldots, \lambda_n)$. Для этого обозначим доли вершин двудольного графа H через U и V так, чтобы вершина с наименьшим номером принадлежала доле U, и положим $\lambda_i = 0$ для $i \in U$, $\lambda_i = 1$ для $i \in V$, $\lambda_i = *$ для $i \in N \setminus (U \cup V)$ (дальше вершины и их номера не различаем). Очевидно, что F взаимно однозначно отображает множество всех двудольных графов (с не более чем n вершинами, нумерованными числами из N) на множество стандартных строк (длины n).

Системе $\mathbf{H} = \{H_1, \ldots, H_s\}$ различных двудольных графов поставим в соответствие стандартную матрицу $\Lambda = F(\mathbf{H})$, образованную строками $\lambda(i) = F(H_i)$, $i = 1, \ldots, s$. Напомним, что мы различаем матрицы с точностью до перестановок строк.

Теорема 4. Пусть K_G — класс равносильности матриц разложения, ассоциированный с характеристическим графом G. Имеют место следующие утверждения:

1) Если некоторая система $\mathbf{H}$ биклик графа G образует его покрытие, то стандартная матрица $\Lambda = F(\mathbf{H})$ принадлежит классу K_G.
2) Отображение F задает для любого G взаимно однозначное соответствие между покрытиями графа G бикликами и стандартными матрицами класса K_G.

Доказательство.

1. Пусть $\mathbf{H} = \{H_1, \ldots, H_s\}$. Поскольку $\mathbf{H}$ — покрытие графа G, выполнено $H_1 \cup \cup \ldots \cup H_s = G$. Стандартная матрица $\Lambda = F(\mathbf{H})$ образована строками $\lambda(i) = F(H_i)$, $i = 1, \ldots, s$. Рассматривая строку $\lambda(i)$ как матрицу, свяжем с ней характеристический граф $G_\chi(\lambda(i))$. Он представляет собой полный двудольный граф, доли которого образованы номерами нулевых и номерами единичных разрядов строки $\lambda(i)$, а потому $G_\chi(\lambda(i)) = H_i$. С учётом этого имеем $G_\chi(\Lambda) = \bigcup\limits_{i=1}^{s} G_\chi(\lambda(i)) = \bigcup\limits_{i=1}^{s} H_i = G$. Это означает $\Lambda \in K_G$.

2. Докажем взаимную однозначность отображения F при любом G.

Из того, что в применении к полным двудольным графам H отображение $F(H)$ взаимно однозначно, следует, что для разных систем $\mathbf{H}$ таких графов стандартные матрицы $\Lambda = F(\mathbf{H})$ различны.

Осталось установить, что всякая стандартная матрица $\Lambda \in K_G$ является образом $F(\mathbf{H})$ некоторого покрытия $\mathbf{H}$ графа G бикликами. Пусть матрица Λ образована строками $\lambda(i)$, $i = 1, \ldots, s$. Введём полные двудольные графы $H_i = F^{-1}(\lambda(i))$; для них $\lambda(i) = F(H_i)$. При доказательстве п. 1 теоремы показано, что тогда $G_\chi(\lambda(i)) = H_i$. С учётом этого и $\Lambda \in K_G$ имеем $G = G_\chi(\Lambda) = \bigcup\limits_{i=1}^{s} G_\chi(\lambda(i)) = \bigcup\limits_{i=1}^{s} H_i$. Это означает, что система $\mathbf{H}$ образует покрытие графа G. Остаётся заметить, что $\Lambda = F(\mathbf{H})$. ■

Метод построения матрицы разложения по характеристическому графу проиллюстрируем примером.

Пример 2. Построим одну из стандартных матриц, соответствующих графу G, представленному на рис. 1, *а*. В качестве покрытия графа G бикликами возьмём, например,

$$G = (\{2,6\} \times \{3,5\}) \cup (\{5\} \times \{1,2,6\}),$$

где $U \times V = \{(u,v) : u \in U, v \in V\}$ — полный двудольный граф с долями U и V. Стандартная матрица Λ', сопоставленная этому покрытию по теореме 4, имеет вид

1	2	3	4	5	6
$*$	0	1	$*$	1	0
0	0	$*$	$*$	1	0

.

Укажем источник X', соответствующий разложению, представленному матрицей Λ', и имеющий тот же основной алфавит, что и X. Метод его построения, обоснование которого мы опускаем, состоит из следующих этапов:

1. Каждой строке $\lambda'(i)$ матрицы Λ' сопоставляется множество $U(i)$ — объединение множеств T_j, соответствующих единичным разрядам $\lambda'_j(i)$ этой строки, где множества T_j относятся к источнику X. В рассматриваемом случае $U(1) = T_3 \cup T_5 = \{2,5\}$, $U(2) = T_5 = \{5\}$.

2. С каждым множеством $U(i)$ связывается его характеристическая строка (имеющая 1 в разрядах, принадлежащих $U(i)$) и из этих строк образуется матрица Λ'_0. Она является матрицей разложения символов основного алфавита, согласованной с матрицей разложения Λ'. В данном случае матрица Λ'_0 имеет вид

1	2	3	4	5
0	1	0	0	1
0	0	0	0	1

.

3. Строится система множеств T'_j, задающая алфавит аппроксимирующего источника X'. Множество T'_j образуется номерами тех столбцов матрицы Λ'_0, которые доопределяют столбец λ'_j. В рассматриваемом случае возникает система множеств $T'_1 = \{1,2,3,4\}$, $T'_2 = \{1,3,4\}$, $T'_3 = \{2,5\}$, $T'_4 = \{1,2,3,4,5\}$, $T'_5 = \{5\}$, $T'_6 = \{1,3,4\}$. Непосредственно видно, что она связана с системой множеств для источника X соотношениями $T_j \subseteq T'_j$, означающими, что источник X' слабее X. Поскольку X' реализует (нижнюю) аппроксимацию источника X, он является сильнейшим из разделимых источников, система множеств которых обладает указанным свойством.

6. Минимизация разложений

Под *сложностью разложения* будем понимать число s двоичных источников, входящих в разложение. *Задача минимизации разложений* состоит в том, чтобы по заданному разложению и числу s узнать, существует ли равносильное разложение в произведение не более s источников. *Задача о минимальной аппроксимации* ставится как задача выяснения по недоопределённому источнику X и числу s, существует ли для X аппроксимирующее разложение, сложность которого не выше s.

В этих постановках ответ не зависит от значений ненулевых вероятностей. Поэтому, исключив вероятности из рассмотрения, будем вместо разложений говорить о матрицах разложений, а вместо источников — об их алфавите (либо системе множеств, задающих алфавит).

Теорема 5. Задача минимизации разложений и задача о минимальной аппроксимации недоопределённых источников NP-полны.

Доказательство. NP-полноту этих задач установим путём сведения к ним NP-полной задачи о покрытии графа бикликами [9, 10]. Она состоит в том, чтобы по графу G и числу s узнать, существует ли для G покрытие, образованное не более s бикликами.

Сведение к задаче минимизации разложений осуществляется путём покрытия графа G какой-либо системой биклик (например, биклик S_i, образованных всеми рёбрами, инцидентными вершинам i) и построению по этому покрытию стандартной матрицы Λ методом теоремы 4. Из теоремы 4 следует, что граф G может быть покрыт не более s бикликами тогда и только тогда, когда существует матрица с не более s строками, равносильная матрице Λ.

Для сведения к задаче о минимальной аппроксимации достаточно построить методом теоремы 3 источник X, для которого $G_\chi^{\mathrm{app}}(X) = G$. Из теорем 3 и 4 следует, что граф G может быть покрыт не более s бикликами тогда и только тогда, когда существует аппроксимирующее разложение источника X, имеющее сложность не выше s. Теорема доказана. ■

Ясно, что задачи минимизации разложений и о минимальной аппроксимации источника также полиномиально сводятся к задаче покрытия графа бикликами, а потому полиномиально эквивалентны ей. Дальше будем вести речь только о покрытии графа бикликами. В [11] рассматривается вопрос о возможности решения с полиномиальной сложностью задачи о покрытии графа не более s бикликами при малых значениях s. Методами параметризованной сложности установлено, что при $s = \mathrm{const}$ эта задача допускает решение со сложностью $\mathrm{O}(f(s) + n^3)$. Но полученное в работе значение $f(s) = 3^{2s^2+3s}$ исключает реальное применение этого результата в приложениях.

В связи с NP-полнотой представляют интерес и неполиномиальные достаточно экономные методы решения задачи покрытия бикликами. При построении наилучших покрытий можно ограничиться максимальными (по включению) бикликами. Поэтому задача покрытия будет решаться в два этапа, на первом из которых будут найдены все максимальные биклики графа, а на втором — осуществлено покрытию ими графа.

Опишем первый этап. Пусть задан граф G. Ребру этого графа, соединяющему вершины u и v, сопоставим двумерную точку (u, v), $u < v$, и множество этих точек обозначим через $\tilde{G}$. Для полного двудольного графа H с долями U и V будем наряду с $\tilde{H}$ использовать задание посредством множества точек $\hat{H} = U \times V = \{(u, v) : u \in U, v \in V\}$, которое назовём *прямоугольником*. В вырожденных случаях, когда какое-либо из множеств U и V одноэлементно, также будем говорить о прямоугольниках. Прямоугольник $\hat{H}$ *допустим*, если $\tilde{H} \subseteq \tilde{G}$.

Прямоугольники $U \times V$ и $V \times U$ будем называть *сопряжёнными*. Скажем, что прямоугольник $\hat{H}_1 = U_1 \times V_1$ *поглощает* прямоугольник $\hat{H}_2 = U_2 \times V_2$, если $U_1 \supseteq U_2$ и $V_1 \supseteq V_2$, и что $\hat{H}_1$ *сопряжённо поглощает* $\hat{H}_2$, если $U_1 \supseteq V_2$ и $V_1 \supseteq U_2$. Допустимый прямоугольник $\hat{H}$ назовём *максимальным*, если любой допустимый прямоугольник, поглощающий $\hat{H}$, совпадает с ним. Отметим, что для пары сопряжённых прямоугольников $\hat{H}_1$ и $\hat{H}_2$ множества $\tilde{H}_1$ и $\tilde{H}_2$ совпадают, а потому, если один из них допустим (либо максимален), то и второй прямоугольник допустим (максимален).

Некоторое множество максимальных прямоугольников назовём *представительным*, если всякий максимальный прямоугольник либо сам содержится в этом мно-

жестве, либо в нём содержится сопряжённый ему прямоугольник (но не оба). Совокупность $H_1, \ldots, H_s$ биклик образует покрытие графа G, если $\tilde{H}_1 \cup \ldots \cup \tilde{H}_s = \tilde{G}$. Покрытие с наименьшим возможным значением s называется *минимальным*. Ясно, что при построении минимального покрытия можно ограничиться прямоугольниками, входящими в любое представительное множество.

Алгоритм построения представительного множества максимальных прямоугольников использует операцию $\hat{H}_1 \circ \hat{H}_2$ *композиции* прямоугольников, которая по $\hat{H}_1 = U_1 \times V_1$ и $\hat{H}_2 = U_2 \times V_2$ строит пряоугольник

$$\hat{H}_1 \circ \hat{H}_2 = (U_1 \cup U_2) \times (V_1 \cap V_2).$$

Композиция допустимых прямоугольников даёт допустимый прямоугольник, ибо если прямоугольники $U_1 \times V_1$ и $U_2 \times V_2$ допустимы, то допустимыми являются $U_1 \times (V_1 \cap V_2)$ и $U_2 \times (V_1 \cap V_2)$, а потому прямоугольник $(U_1 \cup U_2) \times (V_1 \cap V_2)$ также допустим. С каждой вершиной i $(1 \leqslant i \leqslant n)$ графа G свяжем допустимый прямоугольник $\hat{S}_i = \{i\} \times W_i$, где W_i — множество номеров всех вершин, смежных с i.

Алгоритм состоит из n основных этапов и заключительного. Результатом этапа i $(1 \leqslant i \leqslant n)$ является некоторое множество прямоугольников, обозначаемое Π_i.

Этап 1. Образуем множество Π_1, состоящее из единственного прямоугольника $\hat{S}_1$.

Этап i $(2 \leqslant i \leqslant n)$. К множеству Π_{i-1} добавляем прямоугольник $\hat{S}_i$. Затем последовательно образуем (в каком-либо порядке) прямоугольники $\hat{H} \circ \hat{S}_i = U \times V$, $\hat{H} \in \Pi_{i-1}$, и присоединяем к множеству те из них, в которых $|V| \geqslant 2$, где $|\cdot|$ — мощность множества, и для которых нет поглощающего прямоугольника среди включённых в множество к данному моменту. Множество Π_i является результатом этой процедуры и последующего удаления поглощаемых прямоугольников.

Заключительный этап. Путём удаления из Π_n всех сопряжённо поглощаемых прямоугольников получаем множество Π, считающееся результатом алгоритма.

Утверждение 4. Результатом работы алгоритма является представительное множество максимальных прямоугольников.

Доказательство. Сначала индукцией по i покажем, что для любого допустимого прямоугольника $\hat{H} = U \times V$, у которого все элементы множества U не превосходят i и $|V| \geqslant 2$, в множестве Π_i имеется прямоугольник, поглощающий $\hat{H}$.

Основание индукции $i = 1$. Если допустимый прямоугольник $\hat{H} = U \times V$ удовлетворяет указанным условиям при $i = 1$, то $U = \{1\}$ и $V \subseteq W_1$, а потому $\hat{H}$ поглощается прямоугольником $\hat{S}_1 \in \Pi_1$.

Индукционный шаг $i \geqslant 2$. Предположим, что множество Π_{i-1} обладает требуемым свойством. Рассмотрим произвольный допустимый прямоугольник $\hat{H} = U \times V$, для которого максимальным элементом множества U является i и $|V| \geqslant 2$.

Если $U \setminus \{i\} = \varnothing$, то $\hat{H}$ поглощается прямоугольником $\hat{S}_i \in \Pi_i$ либо поглотившим его прямоугольником в случае, когда $\hat{S}_i$ был удалён из Π_i как поглощенный.

Пусть теперь $U \setminus \{i\} = U_1 \neq \varnothing$. Прямоугольник $\hat{H}_1 = U_1 \times V$ допустим и элементы множества U_1 не превосходят $i-1$. По предположению индукции он поглощается некоторым прямоугольником $\hat{H}_2 = U_2 \times V_2 \in \Pi_{i-1}$. В соответствии с алгоритмом образуем композицию $\hat{H}_3 = \hat{H}_2 \circ \hat{S}_i$. В силу $U_2 \cup \{i\} \supseteq U_1 \cup \{i\} = U$ и $V_2 \cap W_i \supseteq V \cap W_i = V$, прямоугольник $\hat{H}_3$ поглощает $\hat{H}$. В Π_i содержится либо $\hat{H}_3$, либо поглотивший его прямоугольник, который поглощает и $\hat{H}$. Индукция завершена.

При $i = n$ заключаем, что для любого допустимого прямоугольника $U \times V$ с $|V| \geqslant 2$ в Π_n имеется поглощающий его прямоугольник. В частности, там имеются все максимальные прямоугольники $U \times V$ с $|V| \geqslant 2$. Поскольку в Π_n отсутствуют поглощения,

немаксимальных прямоугольников там нет. Если существует максимальный прямоугольник вида $U \times \{i\}$, то сопряжённый прямоугольник $\{i\} \times U$ также максимален и в силу условия $U \subseteq W_i$ обязан совпасть с S_i. Последний, будучи максимальным, содержится в Π_n. Множество Π представительно, поскольку на заключительном этапе алгоритма устранены сопряжённые поглощения. ■

Пусть результатом работы алгоритма является система прямоугольников $\Pi = \{\hat{H}_1, \ldots, \hat{H}_k\}$. От прямоугольников $\hat{H}_i$ следует перейти к множествам $\tilde{H}_i$ и решить задачу минимального покрытия множества $\tilde{G}$ множествами $\tilde{H}_1, \ldots, \tilde{H}_k$. Для этого можно применять любые методы решения задач покрытия. Воспользуемся некоторой модификацией метода Яблонского [12] в сочетании с градиентной процедурой [13] (жадным алгоритмом). Опишем, в чём состоит эта модификация.

Напомним, что при решении задачи покрытия множества $A = \{a_1, \ldots, a_n\}$ наименьшим числом множеств B_j из заданной системы $\{B_1, \ldots, B_k\}$, $B_1 \cup \ldots \cup B_k = A$, методом Яблонского с каждым множеством B_j связывается булева переменная b_j и элементам $a_i \in A$ ставятся в соответствие дизъюнкции $D_i = b_{j_1} \vee \ldots \vee b_{j_p}$, где $B_{j_1}, \ldots, B_{j_p}$ — все множества системы, покрывающие a_i. Образуется функция $D_1 \ldots D_n$, которая затем путём раскрытия скобок с использованием булевых эквивалентностей преобразуется в выражение $F = \bigvee\limits_{u_1,\ldots,u_q} b_{u_1} \ldots b_{u_q}$, в котором отсутствуют поглощения конъюнкций. Каждой конъюнкции $b_{u_1} \ldots b_{u_q}$ этого выражения соответствует тупиковое (безызбыточное) покрытие $B_{u_1} \cup \ldots \cup B_{u_q} = A$, и наоборот [12]. Наилучшие покрытия соответствуют конъюнкциям наименьшей длины.

Задавшись натуральным параметром l, модифицируем эту процедуру так, чтобы она давала дизъюнкцию $F|_l$ всех тех конъюнкций из F, длина которых не превосходит l. Для этого при последовательном раскрытии скобок в произведении $D_1 \ldots D_n$ будем руководствоваться правилом $(D_1 \ldots D_i)|_l = (((D_1 \ldots D_{i-1})|_l D_i)|_l$, т. е. после каждого шага i будем отбрасывать конъюнкции, имеющие длину больше l. Если в качестве l взять какую-либо верхнюю оценку длины минимального покрытия, то самые короткие конъюнкции функции $F|_l$ будут соответствовать минимальным покрытиям.

Для получения оценки l длины покрытия можно воспользоваться градиентной процедурой. В терминах произведения $D_1 \ldots D_n$ она состоит в том, что на каждом шаге выбирается переменная, участвующая в наибольшем числе дизъюнкций, не исключённых на предыдущих шагах, и содержащие её дизъюнкции исключаются. Процедура завершается после исключения всех дизъюнкций; значение l равно числу выбранных в течение процедуры переменных. Градиентная процедура эффективна и обеспечивает достаточно хорошую оценку размера покрытия [13].

Пример 3. Проиллюстрируем все этапы построения наилучшего по сложности аппроксимирующего разложения заданного источника. Чтобы сравнить подход, развитый в данной работе, с методом из [2], рассмотрим тот же пример, что и в [2].

Пусть $M = \{1, 2, 3, 4, 5, 6\}$ и допустимы множества $T_1 = \{1, 2\}$, $T_2 = \{2, 3\}$, $T_3 = \{3, 4\}$, $T_4 = \{4, 5\}$, $T_5 = \{5, 6\}$ и $T_6 = \{1, 6\}$, т. е. алфавитом источника X является $A = \{a_{12}, a_{23}, a_{34}, a_{45}, a_{56}, a_{16}\}$. Значения вероятностей символов несущественны и их указывать не будем.

Методом теоремы 2 построим характеристический граф $G_\chi^{\text{app}}(X)$ аппроксимирующих разложений источника X (рис. 2).

Построение минимального аппроксимирующего разложения источника X в соответствии с теоремой 4 требует наилучшего покрытия графа $G = G_\chi^{\text{app}}(X)$ бикликами. Для решения этой задачи сначала методом утверждения 4 найдём представитель-

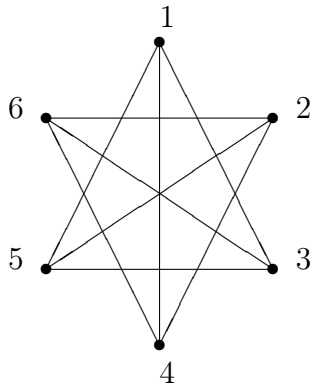


Рис. 2. Характеристический граф $G_\chi^{\mathrm{app}}(X)$

ное множество максимальных прямоугольников. Исходными для алгоритма являются прямоугольники

$$\hat{S}_1 = \{1\} \times \{3,4,5\}, \quad \hat{S}_2 = \{2\} \times \{4,5,6\}, \quad \hat{S}_3 = \{3\} \times \{1,5,6\},$$
$$\hat{S}_4 = \{4\} \times \{1,2,6\}, \quad \hat{S}_5 = \{5\} \times \{1,2,3\}, \quad \hat{S}_6 = \{6\} \times \{2,3,4\},$$

связанные с вершинами графа G. В результате этапов i, $i = 1, \ldots, 6$, работы алгоритма возникают системы прямоугольников Π_i, где

$\Pi_1 = \{\hat{S}_1\}$,
$\Pi_2 = \Pi_1 \cup \{\hat{S}_2, \hat{H}_1\}$, $\hat{H}_1 = \hat{S}_2 \circ \hat{S}_1 = \{1,2\} \times \{4,5\}$,
$\Pi_3 = \Pi_2 \cup \{\hat{S}_3, \hat{H}_2\}$, $\hat{H}_2 = \hat{S}_3 \circ \hat{S}_2 = \{2,3\} \times \{5,6\}$,
$\Pi_4 = \Pi_3 \cup \{\hat{S}_4, \hat{H}_3\}$, $\hat{H}_3 = \hat{S}_4 \circ \hat{S}_3 = \{3,4\} \times \{1,6\}$,
$\Pi_5 = \Pi_4 \cup \{\hat{S}_5, \hat{H}_4\}$, $\hat{H}_4 = \hat{S}_5 \circ \hat{S}_4 = \{4,5\} \times \{1,2\}$,
$\Pi_6 = \Pi_5 \cup \{\hat{S}_6, \hat{H}_5, \hat{H}_6\}$, $\hat{H}_5 = \hat{S}_6 \circ \hat{S}_1 = \{1,6\} \times \{3,4\}$, $\hat{H}_6 = \hat{S}_6 \circ \hat{S}_5 = \{5,6\} \times \{2,3\}$.

Устранив из результирующего множества $\Pi_6 = \{\hat{S}_1, \ldots, \hat{S}_6, \hat{H}_1, \ldots, \hat{H}_6\}$ сопряжённо покрываемые прямоугольники $\hat{H}_4, \hat{H}_5, \hat{H}_6$, придём к представительному множеству $\Pi = \{\hat{S}_1, \ldots, \hat{S}_6, \hat{H}_1, \hat{H}_2, \hat{H}_3\}$. Задача свелась к наилучшему покрытию множества точек $\tilde{G} = \{(1,3), (1,4), (1,5), (2,4), (2,5), (2,6), (3,5), (3,6), (4,6)\}$ множествами системы $\{\tilde{S}_1, \ldots, \tilde{S}_6, \tilde{H}_1, \tilde{H}_2, \tilde{H}_3\}$.

Множествам $\tilde{S}_i$ и $\tilde{H}_j$ сопоставим булевы переменные s_i и h_j и с каждой точкой $(u,v) \in \tilde{G}$ свяжем дизъюнкцию $D_{u,v}$ переменных, соответствующх множествам, покрывающим эту точку (так, например, $D_{1,3} = s_1 \vee s_3 \vee h_2$). Образуем произведение

$$\begin{gathered}(s_1 \vee s_3 \vee h_3)(s_1 \vee s_4 \vee h_1 \vee h_3)(s_1 \vee s_5 \vee h_1)(s_2 \vee s_4 \vee h_1)(s_2 \vee s_5 \vee h_1 \vee h_2) \wedge \\ \wedge (s_2 \vee s_6 \vee h_3)(s_3 \vee s_5 \vee h_2)(s_3 \vee s_6 \vee h_2 \vee h_3)(s_4 \vee s_6 \vee h_3)\end{gathered} \tag{5}$$

этих дизъюнкций.

Применением градиентной процедуры оценим сверху размер минимального покрытия. В данном примере одним из возможных результатов процедуры является последовательность переменных h_1, s_6, s_3, и, таким образом, можно в модифицированной процедуре взять $l = 3$.

Приведём результаты раскрытия скобок в (5) с удалением конъюнкций, длина которых превышает 3. В строке i, $i = 1, \ldots, 9$, помещён результат раскрытия первых i скобок:

1) $s_1 \vee s_3 \vee h_3$;
2) $s_1 \vee s_3 s_4 \vee s_3 h_1 \vee h_3$;
3) $s_1 \vee s_3 s_4 s_5 \vee s_5 h_3 \vee s_3 h_1 \vee h_1 h_3$;
4) $s_1 s_2 \vee s_2 s_5 h_3 \vee s_1 s_4 \vee s_3 s_4 s_5 \vee s_4 s_5 h_3 \vee s_1 h_1 \vee s_3 h_1 \vee h_1 h_3$;
5) $s_1 s_2 \vee s_2 s_5 h_3 \vee s_1 s_4 s_5 \vee s_3 s_4 s_5 \vee s_4 s_5 h_3 \vee s_1 h_1 \vee s_3 h_1 \vee h_1 h_3 \vee s_1 s_4 h_2$;
6) $s_1 s_2 \vee s_2 s_5 h_3 \vee s_2 s_3 h_1 \vee s_2 h_1 h_3 \vee s_1 s_6 h_1 \vee s_3 s_6 h_1 \vee s_6 h_1 h_3 \vee s_1 h_1 h_2 \vee s_3 h_1 h_2 \vee h_1 h_2 h_3 \vee s_1 s_4 h_2$;
7) $s_1 s_2 s_3 \vee s_2 s_3 h_1 \vee s_3 s_6 h_1 \vee s_1 s_2 s_5 \vee s_2 s_5 h_3 \vee s_1 s_2 h_2 \vee s_1 h_1 h_2 \vee s_3 h_1 h_2 \vee h_1 h_2 h_3 \vee s_1 s_4 h_2$;
8) $s_1 s_2 s_3 \vee s_2 s_3 h_1 \vee s_3 s_6 h_1 \vee s_2 s_5 h_3 \vee s_1 s_2 h_2 \vee s_1 h_1 h_2 \vee s_3 h_1 h_2 \vee h_1 h_2 h_3 \vee s_1 s_4 h_2$;
9) $s_1 s_4 h_2 \vee s_3 s_6 h_1 \vee s_2 s_5 h_3 \vee h_1 h_2 h_3$.

Из последней строки, содержащей выражение $F|_3$, видно, что имеются четыре минимальных покрытия (их размер равен 3). Отметим, что и градиентное покрытие оказалось в данном случае минимальным.

Для построения минимального аппроксимирующего разложения воспользуемся, например, покрытием

$$G = S_1 \cup S_4 \cup H_2 = (\{1\} \times \{3,4,5\}) \cup (\{4\} \times \{1,2,6\}) \cup (\{2,3\} \times \{5,6\}),$$

представленным первой конъюнкцией строки 9. Стандартная матрица, сопоставленная этому покрытию по теореме 4, имеет вид

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 0 & * & 1 & 1 & 1 & * \\ 0 & 0 & * & 1 & * & 0 \\ * & 0 & 0 & * & 1 & 1 \end{array}. \tag{6}$$

Она задаёт одно из минимальных аппроксимирующих разложений рассматриваемого источника.

Методом, описанным в примере 2, построим аппроксимирующий источник X', основной алфавит которого совпадает с основным алфавитом источника X. Построение включает следующие этапы:

1. Каждой строке i, $i = 1, 2, 3$, матрицы разложения (6) сопоставляется множество $U(i)$ — объединение множеств T_j, соответствующих единичным разрядам этой строки. В данном случае $U(1) = T_3 \cup T_4 \cup T_5 = \{3,4,5,6\}$, $U(2) = T_4 = \{4,5\}$, $U(3) = T_5 \cup T_6 = \{1,5,6\}$.

2. Строится матрица Λ'_0 кодирования символов основного алфавита, образованная характеристическими строками множеств $U(i)$. Она имеет вид

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{array}.$$

3. Находится система множеств T'_j, задающая алфавит аппроксимирующего источника X', где множество T'_j образуется номерами тех столбцов матрицы Λ'_0, которые доопределяют столбец j матрицы (6). В рассматриваемом случае возникает система множеств $T'_1 = \{1,2\}$, $T'_2 = \{2,3\}$, $T'_3 = \{3,4\}$, $T'_4 = \{4,5\}$, $T'_5 = \{5,6\}$, $T'_6 = \{1,6\}$. Эта система совпадает с системой множеств $T_1, \ldots, T_6$ для исходного источника X. Равенство $X' = X$ означает, что источник X разложим и матрица (6) задаёт его минимальное разложение.

Тот же пример рассматривался в работе [2], в которой путём преобразования матриц разложения было найдено одно из минимальных аппроксимирующих разложений. Но это потребовало применения эвристических соображений, интуиции, перебора. Развитая в данной работе техника, базирующаяся на характеристических графах, позволила регулярным методом найти все минимальные аппроксимирующие разложения.

Заключение

В работе рассмотрена задача разложения недоопределённых источников общего вида в произведение недоопределённых двоичных источников. Точное разложение существует не всегда, но всегда имеется лучшее аппроксимирующее разложение, которое для источников, допускающих точное разложение, является точным. Все аппроксимации заданного источника равносильны и образуют класс равносильности разложений. Ставится задача построения аппроксимирующего разложения, имеющего минимальную сложность, т. е. образованного минимальным числом сомножителей.

В [2] предложен полиномиальный алгоритм построения по источнику одного из его аппроксимирующих разложений, а также найдена полная система равносильных преобразований, позволяющая по одному разложению строить другие. Но это не даёт возможности направленного упрощения разложений и, тем более, их минимизации.

Эту проблему позволяет решить техника, развитая в данной работе. В ней с каждым разложением связывается граф, названный характеристическим. Установлено, что характеристический граф инвариантен к равносильным преобразованиям разложений и однозначно определяет класс равносильности. Зная какое-либо разложение из этого класса, можно, найдя по нему характеристический граф, получить доступ ко всему классу. Любому разложению из класса равносильности соответствует покрытие характеристического графа полными двудольными подграфами (бикликами), причём соответствие между разложениями и покрытиями взаимно однозначно с точностью до некоторой стандартизации разложений. Наилучшее покрытие характеристического графа бикликами решает задачу построения минимального разложения.

Но задача покрытия графа бикликами, а вместе с ней и задача минимизации разложений NP-полны. Покрытие графа бикликами сводится к покрытию двумерного дискретного точечного множества прямоугольниками и позволяет использовать известные приближённые либо точные (при малых размерностях) методы решения задач на покрытие. В работе описано применение для этой цели модификации метода Яблонского.

Отметим, что все построения, связанные с нахождением и минимизацией аппроксимирующего разложения источника, реализуемы в терминах введённого инварианта — характеристического графа. Сам граф находится непосредственно по источнику и все дальнейшие действия выполняются посредством операций над графами. Разложения в виде матриц используются лишь для представления ответа.

ЛИТЕРАТУРА

1. *Шоломов Л. А.* О понятии равносильности недоопределённых алфавитов // Прикладная дискретная математика. 2014. № 3(25). С. 40–57.
2. *Шоломов Л. А.* Разложение недоопределенных данных // Дискретный анализ и исследование операций. 2012. Т. 19. № 6. С. 72–98.
3. *Галлагер Р.* Теория информации и надежная связь. М.: Сов. радио, 1974. 720 с.
4. *Шоломов Л. А.* Элементы теории недоопределённой информации // Прикладная дискретная математика. Приложение. 2009. № 2. С. 18–42.

5. *Потапов В. Н.* Введение в теорию информации. Ижевск: РХД, 2014. 152 с.
6. *Шоломов Л. А.* Преобразование нечетких данных с сохранением информационных свойств // Дискретный анализ и исследование операций. Сер 1. 2005. Т. 12. № 3. С. 85–104.
7. *Шоломов Л. А.* О сводимости алфавитов в задачах кодирования недоопределенных данных // Доклады Академии наук. 2015. Т. 460. № 1. С. 26–29.
8. *Харари Ф.* Теория графов. М.: Мир, 1973. 300 с.
9. *Orlin J.* Contentment in graph theory: Coverings graphs with cliques // Nederlandce Akademie Wetenschappen Proc. Ser. A 80. Indiagationes Mathematcae. 1977. V. 39. No. 5. P. 406–424.
10. *Muller J.* On edge perfectness and classes of bipartite graphs // Discrete Mathematics. 1996. V. 149. No. 1–3. P. 159–187.
11. *Flischner H, Majuni E, Paulusma D., and Szeider S.* Covering graphs with few complete bipartite subgraphs // Theor. Computer Sci. 2009. V. 410. P. 2045–2053.
12. *Чегис И. А., Яблонский С. В.* Логические способы контроля электрических схем // Труды МИАН им. В. А. Стеклова. 1958. Т. 51. С. 270–360.
13. *Нигматуллин Р. Г.* Сложность булевых функций. М.: Наука, 1991. 240 с.

REFERENCES

1. *Sholomov L. A.* O ponyatii ravnosilnosti nedoopredelennyh alfavitov [On the concept of underdetermined alphabets of equal strength]. Prikladnaya Diskretnaya Matematika, 2014, no. 3(25), pp. 40–57. (in Russian)
2. *Sholomov L. A.* Decomposition of underdetermined data. J. Appl. Industr. Math., 2013, vol. 7, no. 1, pp. 100–116.
3. *Gallager R. G.* Information Theory and Reliable Communication. New York, Wiley, 1968. 608 p.
4. *Sholomov L. A.* Elementy teorii nedoopredelennoy informatsii [Elements of the underdetermined information theory]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2009, no. 2, pp. 18–42. (in Russian)
5. *Potapov V. N.* Vvedenie v teoriyu informatsii [Introduction to Information Theory]. Izhevsk, RHD, 2014. 152 p. (in Russian)
6. *Sholomov L. A.* Preobrazovanie nechetkih dannyh s sohraneniem informatsionnih svoystv [Transformations of fuzzy data preserving information properties]. Diskretniy Analiz i Issledovanie Operatsiy, ser. 1, 2005, vol. 12, no. 3, pp. 85–104. (in Russian)
7. *Sholomov L. A.* On the reducibility of alfabets for problems of underdetermined data coding. Doklady Mathematics, 2015, vol. 91, no. 1, pp. 117–120.
8. *Harary F.* Graph Theory. London, Addison-Wesley, 1969. 274 p.
9. *Orlin J.* Contentment in graph theory: Coverings graphs with cliques. Nederlandce Akademie Wetenschappen Proc., ser. A 80, Indiagationes Mathematcae, 1977, vol. 39, no. 5, pp. 406–424.
10. *Muller J.* On edge perfectness and classes of bipartite graphs. Discrete Mathematics, 1996, vol. 149, no. 1–3, pp. 159–187.
11. *Flischner H, Majuni E, Paulusma D., and Szeider S.* Covering graphs with few complete bipartite subgraphs. Theor. Computer Sci., 2009, vol. 410, pp. 2045–2053.
12. *Chegis I. A. and Yablonski S. V.* Logicheskie spocobi kontrolya elektricheskih skhem [Logic methods for the electrical networks testing]. Trudy MIAN im. V. A. Steklova, 1958, vol. 51, pp. 270–360. (in Russian)
13. *Nigmatullin R. G.* Slozhnost bulevyh funktsiy [A Complexity of Boolean Functions]. Moscow, Nauka, 1991. 240 p. (in Russian)

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

UDC 519.7

EHE: NONCE MISUSE-RESISTANT MESSAGE AUTHENTICATION

S. V. Agievich

Belarusian State University, Minsk, Belarus

We propose a nonce misuse-resistant message authentication scheme called EHE (Encrypt-Hash-Encrypt). In EHE, a message-dependent polynomial is evaluated at the point which is an encrypted nonce. The resulting polynomial hash value is encrypted again and becomes an authentication tag. We prove the prf-security of the EHE scheme and extend it to two authenticated encryption modes which follow the "encrypt-then-authenticate" paradigm.

Keywords: *message authentication, authenticated encryption, polynomial hashing, prf-security.*

DOI 10.17223/20710410/39/3

Introduction

Let $\mathbb{F}$ be a finite field of $N \gg 1$ elements. Polynomial hashing over $\mathbb{F}$ is defined as follows. A message X to be hashed is transformed into a polynomial $f_X(\lambda) \in \mathbb{F}[\lambda]$, this polynomial is evaluated at some point $H \in \mathbb{F}$, the result of evaluation becomes a hash value of X. Further we suppose that the polynomial f_X has positive degree, its constant term equals 0, different messages are transformed into different polynomials. Usually the message X is divided into blocks which determine the coefficients of f_X. The shorter X, the lower the degree of f_X. Let messages be rather short and $\deg f_X \leqslant d \ll N$.

If H is chosen uniformly at random from $\mathbb{F}$, then the hash values of different messages X and X' coincide with only small probability:

$$\mathsf{P}\{f_X(H) = f_{X'}(H)\} = \mathsf{P}\{H \text{ is a root of } f_X - f_{X'}\} \leqslant \frac{d}{N}. \tag{1}$$

This simple fact supports security of message authentication schemes based on polynomial hashing. Possibly the most well-known scheme of this type was proposed by M. Wegman and J. Carter in [1] and refined by V. Shoup in [2]. Following [3], we call it WCS, by the first letters of the authors' names.

The WCS scheme was successfully used in GCM, a widely deployed authenticated encryption (AE) mode. GCM was introduced in [4] and standardized in [5]. Recall that an AE mode augments an authentication scheme with encryption one.

Describe WCS with inessential simplifications. The point H becomes a random secret key. The additional key is a random permutation π acting on $\mathbb{F}$. An authentication tag T (a key-dependent hash value) of X is calculated using a unique nonce $S \in \mathbb{F}$ as follows:

$$T = f_X(H) + \pi(S).$$

To instantiate WCS, π is usually chosen as an encryption permutation of some block cipher and H is usually a result of encryption of a fixed element $c \in \mathbb{F}$ using π. This is how WCS is instantiated in GCM.

The uniqueness of nonces is essential. Indeed, if the tags $T = f_X(H) + \pi(S)$ and $T' = f_{X'}(H) + \pi(S')$ are calculated with $X \neq X'$ but $S = S'$, then an adversary can effectively determine H as one of the roots of the polynomial equation $f_X(H) - f_{X'}(H) = T - T'$. After determining H, the adversary finds $\pi(S) = T - f_X(H)$ and then can calculate the tag $T'' = f_{X''}(H) + \pi(S)$ of any X''.

The described situation, the significant loss of security after some event, we call *the security collapse*. Message authentication schemes collapse in different ways. For example, in the schemes of type CBC-MAC (see, for example, [6]) an internal collision, which occurs after processing about $\sqrt{N}$ message blocks, allows to perform a selective forgery, that is, to forge tags of special messages. For comparison, WCS collapses much more seriously: universal forgery after only a single nonce repetition.

In the mentioned standard [5] nonce repetition is considered as misuse of GCM. The standard proposes to solve the misuse problem at the cryptoengineering layer. But it is preferable to solve such problems cryptographically, by designing authentication schemes or AE modes which security does not collapse so much after nonce repetition. Such schemes and modes are called *nonce misuse-resistant*.

The GCM mode follows the "encrypt-then-authenticate" paradigm. In this paradigm the nonce misuse-sensitive scheme WCS cannot provide misuse-resistance of the whole mode. Resistance appears if we turn the paradigm into "authenticate-then-encrypt" keeping WCS. This approach was successfully implemented in the GCM-SIV mode [7]. Unfortunately, due to the paradigm shift, GCM-SIV requires an additional pass over the protected data.

In this paper we propose another approach: strengthening the basic message authentication scheme. In Section 1 we introduce a nonce misuse-resistant scheme called EHE. We justify its security and then, in Section 2, discuss details of its instantiation based on a block cipher. In Section 3 we extend EHE to AE modes which preserve the "encrypt-then-authenticate" paradigm. We denote these modes as AE[EHE]. Note that the first mode was standardized in [8] under the name `belt-datawrap`. We accompanied it with a rather cumbersome proof of security. In this paper the proof is drastically simplified.

1. The EHE scheme

In the proposed EHE scheme, a key is a pair of permutations π_1 and π_2 acting on $\mathbb{F}$. A message X to be authenticated is represented by a polynomial f_X which satisfies the previous restrictions. A tag T is calculated using a nonce $S \in \mathbb{F}$ as a value of the following function:

$$\varphi[\pi_1, \pi_2](X, S) = \pi_2(f_X(\pi_1(S))).$$

In this function we start with the permutation π_1, continue with polynomial hashing and finish with the permutation π_2. The permutations mean block encryption, so we deal with the Encrypt-Hash-Encrypt cascade or EHE in short.

In contrast to WCS, the polynomials f_X are evaluated not in a fixed point $H = \pi(c)$ but generally in different points $H = \pi_1(S)$. It is well known (see, for example, [9, Theorem 6.13]) that the polynomial $g(\lambda, \lambda') = f_X(\lambda) - f_{X'}(\lambda') \in \mathbb{F}[\lambda, \lambda']$ has at most $\deg g \cdot N = \max(\deg f_X, \deg f_{X'})N$ roots in $\mathbb{F}^2$. Therefore, for independent random H, H', each uniformly distributed over $\mathbb{F}$, and for arbitrary messages X, X' it holds that

$$\mathsf{P}\{f_X(H) = f_{X'}(H')\} \leqslant \frac{\max(\deg f_X, \deg f_{X'})N}{N^2} \leqslant \frac{d}{N}.$$

This bound forms the basis of our security proofs of EHE. More precisely, we use the slightly stronger bound:

$$\Pr\{f_X(H) = f_{X'}(H') \mid H \neq H'\} \leqslant \frac{d}{N}. \tag{2}$$

It followed from the fact that the polynomial $g(\lambda, \lambda')$ has at most $d(N-1)$ roots (H, H') such that $H \neq H'$. Indeed, the substitution $\lambda' = \lambda + \mu$ transforms g into a polynomial $g'(\lambda, \mu)$. A number of the suitable roots of g is the number of roots of g' with a nonzero last coordinate. This coordinate can be chosen in $N - 1$ ways. Each choice $\mu = c$ yields the univariate polynomial $g'(\lambda, c)$ which has at most d roots.

Let us justify the prf-security of EHE, that is, the indistinguishability of $\varphi[\pi_1, \pi_2]$ from a truly random (ideal) message authentication function. The indistinguishability means that $\varphi[\pi_1, \pi_2]$ is pseudorandom (it "looks" like random) or prf in short. Let an adversary (a probabilistic algorithm) have access to a message authentication oracle G which on a query (X, S) gives a response T. The oracle implements either the function $\varphi[\pi_1, \pi_2]$ (a real implementation) or a truly random function ρ (an ideal implementation). In the real implementation, the permutations π_1, π_2 are chosen independently uniformly at random from the set of all permutations on $\mathbb{F}$. In the ideal implementation, the oracle, given a new query, chooses a response T uniformly at random from $\mathbb{F}$ independently of previous responses. The adversary can make arbitrary queries, can collect and analyze the corresponding responses. Its task is to determine which function G implements. The adversary returns 1 if it is $\varphi[\pi_1, \pi_2]$, or 0 if it is ρ. Let A^G be the output of A.

The quality of A's distinguishing capabilities is characterized by the advantage

$$\mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}}(A) = |\,\mathsf{P}\{A^{\varphi[\pi_1,\pi_2]} = 1\} - \mathsf{P}\{A^{\rho} = 1\}|.$$

The probabilities here are over the random tape of A and over the random choice of π_1, π_2 and ρ. If $\mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}}(A)$ is small then the adversary is hard to distinguish $\varphi[\pi_1, \pi_2]$ from ρ.

Theorem 1. Let EHE be built over a field of N elements. Let an adversary A make at most q queries (X, S) and messages X in these queries be such that $\deg f_X \leqslant d$. Then

$$\mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}}(A) \leqslant \frac{q(q-1)d}{2N}.$$

Proof. Let $(X_1, S_1), \ldots, (X_q, S_q)$ be different queries and $T_1, \ldots, T_q$ be different elements of $\mathbb{F}$ (potential responses). It is sufficient to prove that

$$p = \mathsf{P}\{\varphi[\pi_1, \pi_2](X_i, S_i) = T_i \colon i = 1, \ldots, q\} \geqslant \frac{1}{N^q}(1 - \varepsilon),\ \varepsilon = \frac{q(q-1)(d-1)}{2N}.$$

Indeed, then using the H-coefficients technique [10] or, more precisely, Theorem 1 from [11], we obtain

$$\mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}}(A) \leqslant \frac{q(q-1)}{2N} + \varepsilon = \frac{q(q-1)d}{2N}.$$

Let $H_i = \pi_1(S_i)$ and $Y_i = f_{X_i}(H_i)$, $i = 1, \ldots, q$. Introduce the event $\mathcal{D}_1$ that all Y_i are distinct and the event $\mathcal{D}_2$ that $\pi_2(Y_i) = T_i$ for each i. Let us estimate the probabilities $\mathsf{P}\{\mathcal{D}_1\}$ and $\mathsf{P}\{\mathcal{D}_2 \mid \mathcal{D}_1\}$. They are correspondingly determined by the random choice of π_1 and π_2.

The estimates (1), (2) imply

$$\mathsf{P}\{Y_i = Y_j\} = \mathsf{P}\{f_{X_i}(\pi_1(S_i)) = f_{X_j}(\pi_1(S_j))\} = \mathsf{P}\{f_{X_i}(H_i) = f_{X_j}(H_j)\} \leqslant \frac{d}{N}$$

regardless of whether S_i and S_j coincide or not. Indeed, if $S_i = S_j$ then $H_i = H_j$ is uniformly distrubuted over $\mathbb{F}$ and (1) works. If $S_i \neq S_j$ then (H_i, H_j) is uniformly distributed over $\mathbb{F}^2 \setminus \{(a,a)\colon a \in \mathbb{F}\}$ and (2) works. In whole,

$$\mathsf{P}\{\mathcal{D}_1\} \geqslant 1 - \sum_{1 \leqslant i < j \leqslant q} \mathsf{P}\{Y_i = Y_j\} \geqslant 1 - \frac{q(q-1)d}{2N}.$$

Denote by $N^{[q]}$ the qth factorial power of N:

$$N^{[q]} = N(N-1)\dots(N-q+1) = N^q \prod_{i=0}^{q-1} \left(1 - \frac{i}{N}\right) \leqslant N^q \left(1 - \frac{q(q-1)}{2N}\right).$$

We have

$$\mathsf{P}\{\mathcal{D}_2 \mid \mathcal{D}_1\} = \frac{1}{N^{[q]}} \geqslant \frac{1}{N^q}\left(1 + \frac{q(q-1)}{2N}\right)$$

and

$$p \geqslant \mathsf{P}\{\mathcal{D}_2 \mid \mathcal{D}_1\}\,\mathsf{P}\{\mathcal{D}_1\} \geqslant \frac{1}{N^q}\left(1 + \frac{q(q-1)}{2N}\right)\left(1 - \frac{q(q-1)d}{2N}\right) \geqslant \frac{1}{N^q}(1-\varepsilon)$$

which was to be proved. ■

The theorem implies that the EHE authentication remains prf-secure as long as the number of messages processed by a single key is well below $\sqrt{N/d}$. The prf-security is the strongest property of the message authentication schemes. In particular, it implies the security against forgery attacks. In these attacks an adversary interacts with $G = \varphi[\pi_1, \pi_2]$ making arbitrary queries and getting corresponding responses. The adversary's aim is to predict a response to a query that has not been made yet.

Both permutations π_1 and π_2 in EHE are necessary. Confirm this fact in the context of the forgery attacks. If π_1 is omitted, then an adversary can effectively find different messages X and X' with the same hash values $f_X(S)$ and $f_{X'}(S)$. Then using a tag $T = \pi_2(f_X(S))$ of X the adversary determines the tag $T' = T$ of X' without a query. If π_2 is omitted, then an adversary finds $H = \pi_1(S)$ from $T = f_X(H)$ and determines the tag $T' = f_{X'}(H)$ of arbitrary X', again without a query.

The theorem means that EHE preserves security even if nonces repeat. In principle, EHE can be used with a fixed nonce $S = c$. But in this case the security is collapsed in the following sense. As soon as an adversary finds a collision of tags T and T' of different messages X and X', it obtains the polynomial equation $f_X(H) = f_{X'}(H)$ in $H = \pi_1(c)$. After determining H, the adversary constructs a new message X'' such that $f_{X''}(H) = f_X(H)$ and determines its tag $T'' = T$ without a query. Note that the collision $T = T'$ is expected to occur and EHE is expected to collapse after about $\sqrt{N}$ queries to the authentication oracle. This fact does not contradict to the bound of the theorem.

To determine H the adversary first finds roots of the polynomial $f_X(\lambda) - f_{X'}(\lambda)$ (let us ignore the time required) and then checks each of them to localize the right one. To check a root it is necessary to make a special query, for example, (X'', S). Let M be the number of different roots. If M is large, then the number of check queries is large too. But if M is small, then the collision probability M/N is small too. Therefore, regardless of M, check-time to collision-probability ratio for the pair (X, X') is of order N.

The situation changes drastically if nonces do not repeat. In this case the collision $T = T'$ means that (H, H') is a root of the bivariate polynomial $g(\lambda, \lambda') = f_X(\lambda) - f_X(\lambda')$. Let g

have M different roots. There are at least $\dfrac{M}{2d}$ different coordinates of these roots and time to check is of lower order $\dfrac{M}{d}$. The collision occurs with the probability $\dfrac{M}{N(N-1)}$ and check-time to collision-probability ratio of the pair (X, X') is of lower order $\dfrac{N^2}{d}$, that is, it dramatically increases comparing to the previous situation.

2. Instantiation

Instead of two secret permutations it is convenient to use only one, say π, and derive π_1 and π_2 from it. We are interested in two variants (*instantiation templates*) of such deriving: $(\pi_1, \pi_2) = (\pi, \pi)$ and $(\pi_1, \pi_2) = (\pi^2, \pi)$. The first template is clearly natural, the second one will be used in the following section while extending EHE to AE[EHE]. Let, as usual, π be chosen uniformly at random from the set of all permutations on $\mathbb{F}$.

Theorem 2. Let EHE be built over a field of N elements with $(\pi_1, \pi_2) = (\pi, \pi)$. Let an adversary A make at most q queries (X, S) and messages X in these queries be such that $\deg f_X \leqslant d$. Then

$$\mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}}(A) \leqslant \frac{q(3q-1)d}{2N}.$$

Proof. Modify the previous proof. Let the event $\mathcal{D}_1$ suppress not only the collisions $Y_i = Y_j$ but also the collisions $Y_i = S_j$ and $T_i = H_j$. Additional restrictions mean that every pair (Y_i, T_i) is *fresh*, that is, π_2 can map Y_i to T_i despite the facts that $H_j = \pi_1(S_j)$ and $\pi_1 = \pi_2$.

There are q^2 additional collisions of each type, their probabilities are

$$\mathsf{P}\{Y_i = S_j\} = \mathsf{P}\{f_X(\pi_1(S_i)) = S_j\} = \mathsf{P}\{\pi_1(S_i) \text{ is a root of } f_X - S_j\} \leqslant \frac{d}{N},$$
$$\mathsf{P}\{T_i = H_j\} = \mathsf{P}\{\pi_1(S_j) = T_i\} = \frac{1}{N}.$$

In whole,

$$\mathsf{P}\{\mathcal{D}_1\} \geqslant 1 - \frac{q(q-1)d}{2N} - \frac{q^2 d}{N} - \frac{q^2}{N} = 1 - \frac{q(3q-1)d + 2q^2}{2N}.$$

The event $\mathcal{D}_1$ fixes no more than q different pairs (a preimage S_i, an image H_i) of $\pi_1 = \pi$. So there are at least $(N-q)^{[q]}$ ways to determine images of $\pi_2 = \pi$ for the q additional preimages $Y_1, \ldots, Y_q$ and only one of these ways is suitable, that is, $T_1, \ldots, T_q$. Repeating the estimation technique of the previous proof, we obtain

$$\mathsf{P}\{\mathcal{D}_2 \mid \mathcal{D}_1\} \geqslant \frac{1}{(N-q)^{[q]}} \geqslant \frac{1}{N^q}\left(1 + \frac{q(3q-1)}{2N}\right).$$

Combining the bounds on $\mathsf{P}\{\mathcal{D}_1\}$ and $\mathsf{P}\{\mathcal{D}_2 \mid \mathcal{D}_1\}$ completes the proof. ■

The permutation π can be interpreted as an ideal implementation of a block encryption oracle E. It is an internal oracle of EHE to which A does not have a direct access. A real implementation of E is a permutation F_K uniformly at random chosen from a family F of permutations acting on $\mathbb{F}$. This family is called a block cipher. The index K above is a random key of this cipher. Let EHE[F] be the EHE scheme with the described instantiation of E.

The advantage of A against $\mathrm{EHE}[F]$ is defined and estimated in the following way:

$$\mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}[F]}(A) = |\,\mathsf{P}\{A^{\varphi[F_K,F_K]}=1\} - \mathsf{P}\{A^{\rho}=1\}| \leqslant$$
$$\leqslant \mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}}(A) + |\,\mathsf{P}\{A^{\varphi[F_K,F_K]}=1\} - \mathsf{P}\{A^{\varphi[\pi,\pi]}=1\}|.$$

The last summand characterizes the quality of distinguishing of E, that is, differentiating between its real implementation F_K and its ideal implementation π. The advantage of an adversary B which distinguish E is defined similar to the advantage of A:

$$\mathbf{Adv}^{\mathtt{prp}}_{F}(B) = \left|\mathsf{P}\{B^{F_K}=1\} - \mathsf{P}\{B^{\pi}=1\}\right|.$$

Let $\mathbf{Adv}^{\mathtt{prp}}_{F}(t,q)$ be the maximum of $\mathbf{Adv}^{\mathtt{prp}}_{F}(B)$ over all B which run in time at most t and make at most q queries to E.

The adversary B can use A to distinguish E. To do this, B simulates the oracle $G=\varphi[E,E]$ which responses $E(f_X(E(S)))$ to (X,S) the adversary determines making two queries to E and one polynomial hashing. The adversary B grants A access to the simulated oracle, waits for the output from A and returns this output as its own. The simulation of G needs $q^*=2q$ queries to E and time $t^*=\mathrm{O}(qd)$.

If A runs in time t, then

$$|\,\mathsf{P}\{A^{\varphi[F_K,F_K]}=1\} - \mathsf{P}\{A^{\varphi[\pi,\pi]}=1\}| = \mathbf{Adv}^{\mathtt{prp}}_{F}(B) \leqslant \mathbf{Adv}^{\mathtt{prp}}_{F}(q^*,t+t^*)$$

and, in whole,

$$\mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}[F]}(A) \leqslant \mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}}(A) + \mathbf{Adv}^{\mathtt{prp}}_{F}(q^*,t+t^*).$$

The arguments above are standard in provable security. The last estimate can be used to continue all our further theorems. In such a continuation one should only refine q^* (the total number of A's indirect queries to the internal oracle E) and t^* (time to simulate G over E).

3. The AE[EHE] modes

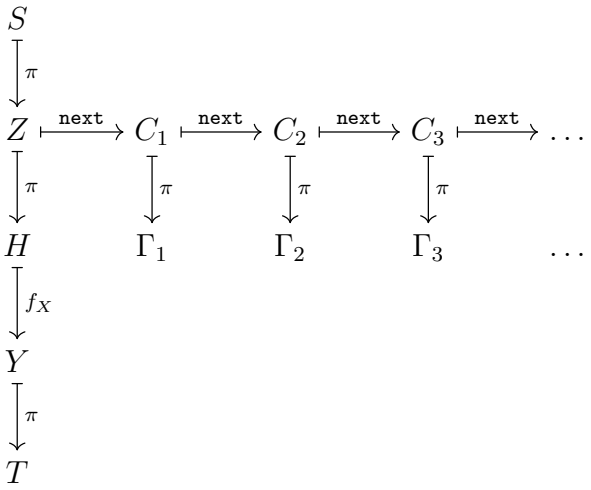
The permutation π can be used not only to instantiate EHE, but also to manage encryption, that is, to extend EHE to AE[EHE]. In this section we provide two modes of authenticated encryption based on EHE. In both modes plaintexts and ciphertexts are considered as words in the alphabet $\mathbb{F}$.

A plaintext is encrypted in the counter mode using a full-cycle permutation $\mathtt{next}$ acting on $\mathbb{F}$. A nonce S is used to calculate $H=\pi(S)$ and then the sequence $C_1=\mathtt{next}(H)$, $C_2=\mathtt{next}(C_1)=\mathtt{next}^2(H),\dots$ of counters. The encrypted counters $\Gamma_k=\pi(C_k)$ are added to the plaintext symbols during encryption or subtracted from the ciphertext symbols during decryption. An adversary can get $\pi(C_k)$ (it can subtract a known plaintext from an intersected ciphertext) but not C_k.

The obtained ciphertext and arbitrary additional data form a message X which is authenticated using EHE. Since $\deg f_X \leqslant d$, the length of the plaintext cannot exceed d and at most d symbols Γ_k are sufficient for encryption.

We cannot justify the security of AE[EHE] in the general case if the template $(\pi_1,\pi_2)=(\pi,\pi)$ is used. It is due to possible similarities between f_X and $\mathtt{next}^k$. For example, if f_X and $\mathtt{next}^k$ act identically, then an adversary can predict $T=\pi(f_X(\pi(S)))$ using $\Gamma_k=\pi(\mathtt{next}^k(\pi(S)))=T$. We should either impose restrictions on $\mathtt{next}$ or change the template.

Start with the second option. The simplest suitable template is $(\pi_1, \pi_2) = (\pi^2, \pi)$. It separates a preimage $\pi^2(S)$ of f_X from a preimage $\pi(S)$ of $\mathtt{next}^k$ and makes similarities between f_X and $\mathtt{next}^k$ ineffective. The whole AE[EHE] mode with the new template can be depicted as follows:



Theorem 3. Let EHE be built over a field of N elements with $(\pi_1, \pi_2) = (\pi^2, \pi)$. Let an adversary A make at most q queries (X, S) and messages X in these queries be such that $\deg f_X \leqslant d$. Let the adversary in addition to each response T receive at most d first elements of the sequence

$$\Gamma_1 = \pi(\mathtt{next}(\pi(S))),\ \Gamma_2 = \pi(\mathtt{next}^2(\pi(S))), \ldots$$

and let r be the total number of such elements. Then

$$\mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}}(A) \leqslant \frac{q(5q + 2r - 1)d}{2N}.$$

Proof. Again modify the previous proof. Let $Z_i = \pi(S_i)$, $H_i = \pi(Z_i)$, $C_{i,k} = \mathtt{next}^k(Z_i)$, $\Gamma_{i,k} = \pi(C_{i,k})$. In the event $\mathcal{D}_1$ suppress the following collisions:

Collisions	Quantity	Probability (upper bound)
$Y_i = Y_j$	$q(q-1)/2$	d/N
$Y_i = S_j$	q^2	d/N
$Y_i = Z_j$	q^2	d/N
$Y_i = C_{j,k}$	qr	d/N
$T_i = Z_j$	q^2	$1/N$
$T_i = H_j$	q^2	$1/N$
$T_i = \Gamma_{j,k}$	qr	$1/N$

These restrictions guarantee the freshness of the pairs (Y_i, T_i). Processing the last two columns of the table, we obtain

$$\mathsf{P}\{\mathcal{D}_1\} \geqslant 1 - \frac{q(5q + 2r - 1)d + 4q^2 + 2qr}{2N}.$$

The event $\mathcal{D}_1$ fixes at most $2q+r$ different images of π. Hence there are at least $(N-2q-r)^{[q]}$ ways to determine q additional images which correspond to the preimages $Y_1,\dots,Y_q$ and only one of these ways is suitable. In result,

$$\mathsf{P}\{\mathcal{D}_2 \mid \mathcal{D}_1\} \geqslant \frac{1}{(N-2q-r)^{[q]}} \geqslant \frac{1}{N^q}\left(1+\frac{q(5q+2r-1)}{2N}\right).$$

Repeating the estimation technique of the previous proof, we get the result required. ■

The instantiation template $(\pi_1,\pi_2)=(\pi,\pi)$ is preferable than $(\pi_1,\pi_2)=(\pi^2,\pi)$ because it requires one less encryption. As we said before, to securely use the template $(\pi_1,\pi_2)=(\pi,\pi)$, it is necessary to impose restrictions on $\mathtt{next}$. These restrictions should impede the collisions of the form $f_X(H)=\mathtt{next}^k(H')$ or even the form $f_X(H)=\mathtt{next}^k(H')$, $H\neq H'$.

In this connection, call the permutation $\mathtt{next}$ (d,δ)-*uniform*, if for any suitable f_X with $\deg f_X\leqslant d$ and each $k=1,\dots,d$ it holds that

$$\mathsf{P}\{f_X(H)=\mathtt{next}^k(H)\},\ \mathsf{P}\{f_X(H)=\mathtt{next}^k(H')\mid H\neq H'\}\leqslant\frac{d\delta}{N}.$$

Here H and H' are independent random, each uniformly distributed over $\mathbb{F}$.

Example. Consider an affine permutation $\mathtt{aff}\colon H\mapsto\alpha H+\beta$, $\alpha,\beta\in\mathbb{F}\setminus\{0\}$. If α is the multiplicative unit of $\mathbb{F}$, then $\mathtt{aff}$ is $(d,1)$-uniform, the best we can get, but it is a full-cycle only if $\mathbb{F}$ is prime. For arbitrary $\mathbb{F}$ the permutation $\mathtt{aff}$ turns into an almost-full-cycle if α is primitive. Indeed, in this case $\mathtt{aff}$ decomposes into a cycle of length $N-1$ and a fixed point $\beta/(1-\alpha)$. The probability to fall into the sole fixed point during encryption is negligible and $\mathtt{aff}$ can be used in the counter mode without meaningful loss of security.

Theorem 4. Let EHE be built over a field of N elements with $(\pi_1,\pi_2)=(\pi,\pi)$. Let an adversary A make at most q queries (X,S) and messages X in these queries be such that $\deg f_X\leqslant d$. Let the adversary in addition to each response T receive at most d first elements of the sequence $\Gamma_1=\pi(\mathtt{next}(\pi(S)))$, $\Gamma_2=\pi(\mathtt{next}^2(\pi(S)))$, … and let r be the total number of such elements. Let $\mathtt{next}$ be (d,δ)-uniform. Then

$$\mathbf{Adv}^{\mathtt{prf}}_{\mathrm{EHE}}(A)\leqslant\frac{q(3q+2r\delta-1)d}{2N}.$$

Proof. Modify the proof of Theorem 2. Let $C_{i,k}=\mathtt{next}^k(H_i)$, $\Gamma_{i,k}=\pi(C_{i,k})$. In the event $\mathcal{D}_1$ suppress the following collisions:

Collisions	Quantity	Probability (upper bound)
$Y_i=Y_j$	$q(q-1)/2$	d/N
$Y_i=S_j$	q^2	d/N
$Y_i=C_{j,k}$	qr	$d\delta/N$
$T_i=H_j$	q^2	$1/N$
$T_i=\Gamma_{j,k}$	qr	$1/N$

The result required follows from the estimates:

$$\mathsf{P}\{\mathcal{D}_1\}\geqslant 1-\frac{q(3q+2r\delta-1)d+2q^2+2qr}{2N},$$
$$\mathsf{P}\{\mathcal{D}_2\mid\mathcal{D}_1\}\geqslant\frac{1}{(N-q-r)^{[q]}}\geqslant\frac{1}{N^q}\left(1+\frac{q(3q+2r-1)}{2N}\right).$$

The theorem is proved. ■

To fully justify the security of the proposed AE[EHE] modes we need to show that it is hard to distinguish from random not only the tags T but also the symbols Γ_k (provided that the nonces S do not repeat). Technically, it can be quite easily done by rebuilding the proofs of Theorems 3 and 4. We leave such rebuilding outside the scope of this paper.

REFERENCES

1. *Wegman M. and Carter J.* New hash functions and their use in authentication and set equality. J. Comp. and System Sci., 1981, vol. 22, pp. 265–279.
2. *Shoup V.* On fast and provably secure message authentication based on universal hashing. CRYPTO'2006, LNCS, 1996, vol. 1109, pp. 313–328.
3. *Bernstein D.* Stronger security bounds for Wegman — Carter — Shoup authenticators. EUROCRYPT'2005, LNCS, 2005, vol. 3494, pp. 164–180.
4. *McGrew D. A. and Viega J.* The security and performance of the Galois / Counter Mode (GCM) of operation. INDOCRYPT'2004, LNCS, 2004, vol. 3348, pp. 343–355.
5. *Dworkin M.* Recommendation for Block Cipher Modes of Operation: Galois-Counter Mode (GCM) for Confidentiality and Authentication. NIST Special Publication 800-38D, 2007. `http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf`.
6. *Rogaway R.* Evaluation of Some Blockcipher Modes of Operation, Cryptography Research and Evaluation Committees (CRYPTREC), 2011. `http://www.cryptrec.go.jp/estimation/techrep_id2012_2.pdf`.
7. *Gueron S. and Lindell Y.* GCM-SIV: full nonce misuse-resistant authenticated encryption at under one cycle per byte. Proc. CCS'15, Denver, CO, USA, 2015, pp. 109–119.
8. STB 34.101.31-2011. Informatsionnye Tekhnologii i Bezopasnost'. Zashchita Informatsii. Kriptograficheskie algoritmy shifrovaniya i kontrolya tselostnosti [Information Technology and Security. Data Encryption and Integrity Algorithms]. Standard of Belarus, 2011. `http://apmi.bsu.by/assets/files/std/belt-spec27.pdf` (in Russian)
9. *Lidl R. and Niederraiter H.* Finite Fields. Cambridge University Press, 1997.
10. *Patarin J.* Etude des Gènèrateurs de Permutations Basès sur le Sch'ema du D.E.S., Ph.D. Thesis, University of Paris, 1991. (in French)
11. *Nandi M.* Improved security analysis for OMAC as a pseudorandom function, J. Math. Cryptol., 2009, vol. 3, pp. 133–148.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 004.94

МНОГОУРОВНЕВОЕ ТЕМАТИКО-ИЕРАРХИЧЕСКОЕ УПРАВЛЕНИЕ ДОСТУПОМ (MLTHS-СИСТЕМА)

Н. А. Гайдамакин

Уральский федеральный университет имени первого Президента России Б. Н. Ельцина, г. Екатеринбург, Россия

Управление доступом строится на объединении доверительно-мандатного и тематического принципа. Субъектам и объектам доступа присваиваются составные метки безопасности, содержащие уровень безопасности (гриф секретности для объектов, уровень допуска для субъектов) и тематический индекс (тематика объектов, тематические полномочия субъектов). В отличие от известной MLS-модели, использующей неиерархические, т. е. неупорядоченные, тематические категории в виде набора тематических рубрик, тематические индексы объектов и тематические полномочия субъектов отображаются элементами иерархических тематических классификаторов, которые широко применяются в технологиях организации документальных хранилищ. Математически метки безопасности представляют собой элементы произведения линейной решётки уровней безопасности, применяемой в модели Белла — ЛаПадулы, и специально построенной решётки мультирубрик на иерархических классификаторах. Построены специальные отношения доминирования («шире — уже»), операции взятия наименьшей верхней и наибольшей нижней границ мультирубрик, которые не могут быть выражены обычными теоретико-множественными отношениями включения, операциями объединения и пересечения. В MLTHS-системе определены процедуры присвоения пользователям и инициируемым ими субъектам меток безопасности. Установлены правила санкционирования монитором безопасности доступов субъектов к объектам на чтение, запись, выполнение, в том числе определены процедуры присвоения меток безопасности вновь создаваемым объектам. Установлены правила санкционирования множественных доступов (одного субъекта одновременно к нескольким объектам, нескольких субъектов одновременно к одному объекту). Доказывается безопасность функционирования MLTHS-системы по критерию отсутствия потоков «сверху вниз» и между несравнимыми по меткам безопасности сущностями (субъектами или объектами доступа). MLTHS-система позволяет соединить технологии текстового поиска в документальных хранилищах и технологии разграничения доступа, создавая на этой основе защищённые документально-поисковые системы без ограничения их функциональности.

Ключевые слова: *управление доступом, модель безопасности, иерархический тематический классификатор, мультирубрика, решётка мультирубрик, документальные информационно-поисковые системы, тематическое индексирование, MLS-модель.*

DOI 10.17223/20710410/39/4

MULTILEVEL THEMATIC-HIERARCHICAL ACCESS CONTROL (MLTHS-SYSTEM)

N. A. Gaydamakin

Ural Federal University, Ekaterinburg, Russia

E-mail: haid2@bk.ru

Access control in computer systems is based on the combination of confidence-mandatory and thematic principles. Composite security labels (tags) containing a security level (classification grade for objects and access level for subjects) and a thematic index (object themes and thematic permissions for subjects) are assigned to the access objects and subjects. In contrast to the known MLS-model that uses so called non-hierarchical (i.e. unordered) thematic categories in the form of thematic rubrics, our model (MLTS-system) uses thematic object indexes and thematic subject permissions which appear as hierarchical thematic classifier elements widely used in document storage technologies. Mathematically, the security labels are elements of the product of the security level algebraic lattice used in Bell — LaPadula model and of a special multirubric lattice based on hierarchical classifiers. Special dominance relations (wider — narrower) and binary operations (greatest lower and least upper multirubric bounds) that cannot be expressed by using ordinary set-theoretic inclusion relation and union and intersection operations are introduced. In MLTHS-system, for assigning security tags to users and to user-initiated subjects, some specific procedures are defined. Authorization rules to subject-to-object read, write and execute access are defined for security monitor as well as security tag assignment procedures for newly created objects. Multiple access (a single subject to many objects and many subjects to a single object) authorization rules are established. It is proven that MLTHS-system is secure by criteria of flow absence between security tag-incomparable entities (objects or subjects) and of top down flow absence. MLTHS-system allows combining access control and document storage text search technologies to create secure search engines with no functional limitations.

Keywords: *access control (management), security model, hierarchical thematic classifier, multirubric, multirubric lattice, documentary information retrieval systems, thematic indexing, MLS-model.*

1. Исходные положения

Широко известным подходом к управлению доступом к информационным объектам в компьютерных системах (КС) является применение мандатной политики [1 – 3]. Реализующая эту политику модель Белла — ЛаПадулы [4] основывается на линейной решётке неких абстрактных уровней безопасности

$$\Lambda_{\mathsf{L}}(L, <, \circ, \otimes),$$

где L — множество уровней безопасности; $<$ — отношение доминирования, определяющее строгий и полный порядок на множестве L; $\circ$ — оператор, определяющий для любой пары $l_1, l_2 \in L$ точную (наименьшую) верхнюю границу $l = l_1 \circ l_2 = \max(l_1, l_2)$; $\otimes$ — оператор, определяющий для любой пары $l_1, l_2 \in L$ точную (наибольшую) нижнюю границу $l = l_1 \otimes l_2 = \min(l_1, l_2)$.

Множество сущностей системы управления доступом (множество субъектов доступа $s \in S$ и множество объектов доступа $o \in O$) отображаются на решётку уровней безопасности Λ_{L}. Результатом отображения (классификации) является присвоение каждому субъекту или объекту уровня безопасности, который для субъектов именуется уровнем допуска, а для объектов — грифом секретности.

Критерием безопасности является запрет информационных потоков «сверху вниз», т. е. от сущностей с более высоким уровнем безопасности к сущностям с более низким уровнем безопасности.

Вместе с тем «подсмотренная» в сфере традиционного «бумажного» секретного делопроизводства политика мандатного управления доступом на основе грифов секретности и уровней допуска не является единственной. В специальных библиотеках, в архивах многих организаций и предприятий управление доступом осуществляется по тематическому принципу, т. е. в зависимости от тематики сведений, находящихся в информационных объектах. Наиболее простой подход основан на решётке множества подмножеств тематических категорий (тематических рубрик):

$$\Lambda_{\mathsf{TD}}(\boldsymbol{P}_{\mathsf{TD}}, \subseteq, \cup, \cap),$$

где $\boldsymbol{P}_{\mathsf{TD}}$ — множество подмножеств P_{TD} из базового множества тематических категорий $T_{\mathsf{TD}} = \{\tau_1, \tau_2, \dots\}$; τ_i — i-я тематическая категория (рубрика); $P_{\mathsf{TD}} \in \boldsymbol{P}_{\mathsf{TD}}$; $P_{\mathsf{TD}} \subseteq T_{\mathsf{TD}}$; $\subseteq$ — отношение теоретико-множественного включения, определяющее на множестве $\boldsymbol{P}_{\mathsf{TD}}$ отношение нестрого частичного порядка; $\cup$ — операция теоретико-множественного объединения, определяющая для любой пары $P_{\mathsf{TD}1}, P_{\mathsf{TD}2} \in \boldsymbol{P}_{\mathsf{TD}}$ наименьшую верхнюю границу $P_{\mathsf{TD}\cup} = P_{\mathsf{TD}1} \cup P_{\mathsf{TD}2}$; $\cap$ — операция теоретико-множественного пересечения, определяющая для любой пары $P_{\mathsf{TD}1}, P_{\mathsf{TD}2} \in \boldsymbol{P}_{\mathsf{TD}}$ наибольшую нижнюю границу $P_{\mathsf{TD}\cap} = P_{\mathsf{TD}1} \cap P_{\mathsf{TD}2}$.

Критерием безопасности является недопустимость информационных потоков от сущностей с более широкой тематикой к сущностям с более узкой тематикой, а также между несравнимыми по отношению включения сущностями.

Поскольку в некоторых областях, в частности в области обработки текстов, набор элементов, характеризующих тематику информационных объектов (набор ключевых слов, набор тематических рубрик), называют дескрипторами, то $\Lambda_{\mathsf{TD}}(\boldsymbol{P}_{\mathsf{TD}}, \subseteq, \cup, \cap)$ будем называть *дескрипторной тематической решёткой*.

На рис. 1 представлены диаграмма Хассе решётки $\Lambda_{\mathsf{TD}}(\boldsymbol{P}_{\mathsf{TD}}, \subseteq, \cup, \cap)$ на множестве из трёх тематических рубрик $T_{\mathsf{TD}} = \{\tau_1, \tau_2, \tau_3\}$ и граф допустимых по критерию «снизу вверх» потоков. Символ $\varnothing$ соответствует «пустой» рубрике, т. е., в контексте управления доступом, такой тематике сведений, доступ к которым неограничен.

Пунктирные стрелки на рис. 1, *б* задают допустимые информационные потоки между сущностями с соответствующими тематическими категориями. Все остальные потоки являются недопустимыми как потоки от сущностей с более широкой тематикой к сущностям с более узкой тематикой или как потоки между несравнимыми по тематике сущностями.

При объединении рассмотренных подходов субъекты и объекты доступа отображаются в произведение решёток $\Lambda_{\mathsf{L}}(L, <, \circ, \otimes)$ и $\Lambda_{\mathsf{TD}}(\boldsymbol{P}_{\mathsf{TD}}, \subseteq, \cup, \cap)$. Каждый субъект или объект доступа помечается двойной меткой: уровнем секретности и тематическим индексом в виде набора тематических («неиерархических») категорий.

На рис. 2 представлен результат произведения линейной решётки из двух уровней безопасности $(l_1, l_2; l_1 < l_2)$ и тематической решётки из двух рубрик $\{\tau_1, \tau_2\}$.

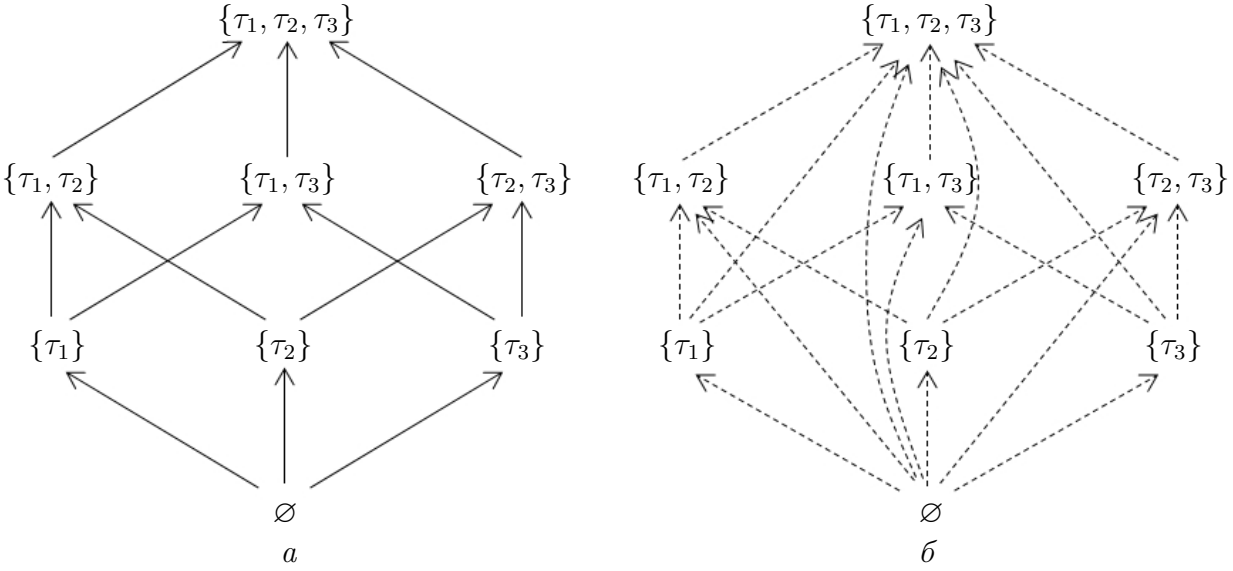


Рис. 1. Диаграмма Хассе решётки подмножеств на множестве из трёх тематических рубрик $\{\tau_1, \tau_2, \tau_3\}$ (*а*) и граф допустимых потоков в системе управления доступом (*б*)

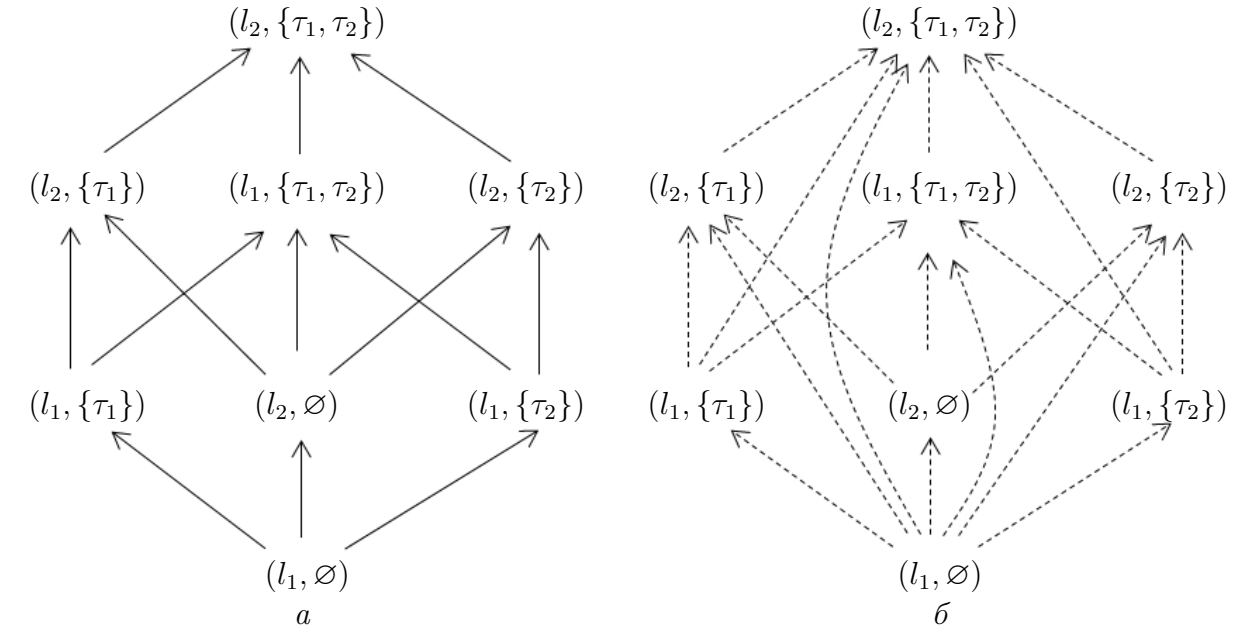


Рис. 2. Диаграмма Хассе произведения линейной решётки из двух уровней безопасности $(l_1, l_2; l_1 < l_2)$ и тематической решётки из двух рубрик $\{\tau_1, \tau_2\}$ (*а*) и граф допустимых потоков (*б*)

Объединённый подход, именуемый в некоторых источниках как *MLS-модель*[1], позволяет строить системы управления доступом, отражающие существенно более разнообразные технологии работы с конфиденциальной информацией, более адекватно настраиваемые на особенности информационно-технологической сферы конкретных организаций.

[1]Multi-Level Security. В некоторых источниках под термином MLS понимают в целом модели мандатной политики, в том числе модель Белла — ЛаПадулы, характеризующиеся наличием множества уровней безопасности.

Вместе с тем в MLS-модели, как и в базовой модели Белла — ЛаПадулы, не определены такие важные в практической реализации аспекты, как присвоение/изменение уровней безопасности и тематических меток субъектам и объектам доступа, процедуры совместного доступа субъектов к какому-либо объекту, одновременного доступа субъекта к нескольким объектам.

Кроме того, в технологиях текстового поиска при организации хранилищ документов широко применяется индексирование текстовых документов не просто наборами тематических рубрик (ключевых слов), а элементами иерархически-организованных классификационных структур (иерархические тематические классификаторы, тезаурусы, иерархические онтологии [5]), использование которых не может быть отражено с помощью простой дескрипторной тематической решётки $\Lambda_{\mathsf{TD}}(\boldsymbol{P}_{\mathsf{TD}}, \subseteq, \cup, \cap)$.

В [6] представлена модель тематического разграничения доступа при иерархической структуре классификатора, введено понятие мультирубрики и построена решётка мультирубрик. Приведём в кратком изложении основные положения алгебры мультирубрик.

2. Мультирубрики и решётка мультирубрик на иерархическом тематическом классификаторе

Иерархические тематические классификаторы, применяемые для тематического индексирования информационных объектов (книг в библиотеках, документов в делопроизводстве), строятся по таксономическому (родовидовому) принципу. Вся тематика предметной области разбивается на рубрики, которые, в свою очередь, могут разбиваться на подрубрики и т. д. Таким образом, если сведения, содержащиеся в информационном объекте, отнесены к рубрике τ_i, то в нём содержатся сведения по всем подрубрикам τ_j, составляющим рубрику τ_i, т. е. подчинённым рубрике τ_i.

В теоретико-множественной интерпретации иерархический тематический классификатор представляет собой частично упорядоченное множество тематических категорий (рубрик) $T_{\mathsf{TH}} = \{\tau_1, \tau_2, \ldots, \tau_K\}$, образующих корневое дерево (см., например, рис. 3).

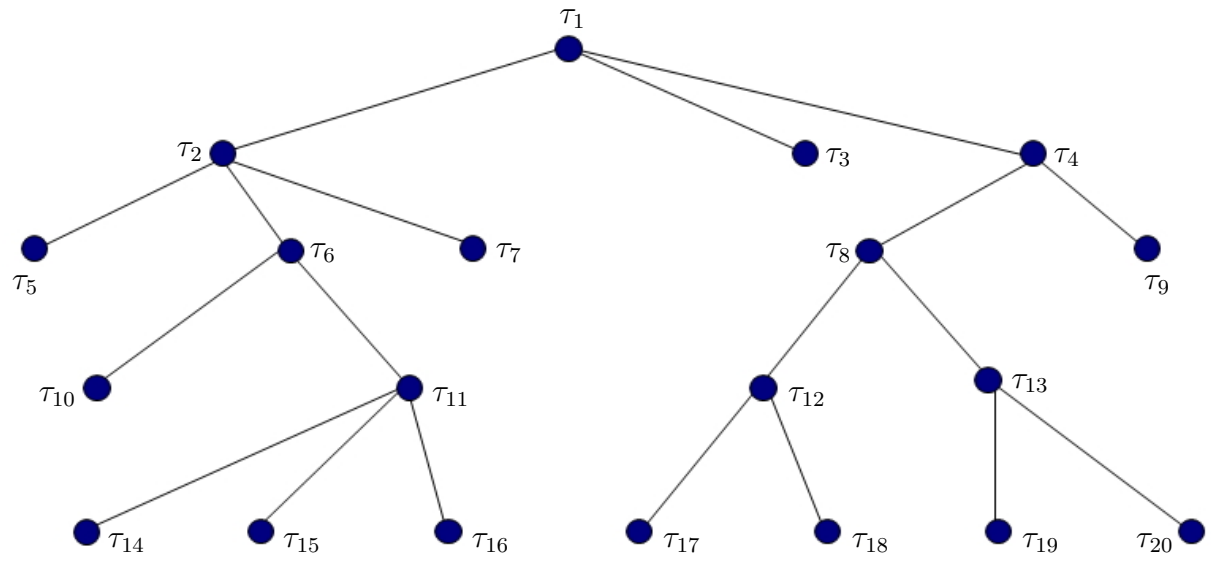


Рис. 3. Пример иерархического тематического классификатора в виде корневого графа (τ_1 — корень дерева, обозначающий всю тематику предметной области)

Информационный объект может содержать сведения, относящиеся сразу к нескольким рубрикам. С учётом иерархической сущности классификатора для тематического индексирования информационных объектов вводится понятие «*мультирубрика*».

Определение 1. Мультирубрикой $\mathcal{T}_i^{\mathsf{M}}$ называется любое подмножество $\{\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_L}\}$ $(L \leqslant K)$ множества $T_{\mathsf{TH}} = \{\tau_1, \tau_2, \ldots, \tau_K\}$ рубрик-вершин иерархического тематического классификатора, такое, что:

а) любая вершина τ_{i_k} не находится в отношениях «родитель — потомок» с любой другой вершиной τ_{i_m} того же подмножества: $\tau_{i_k} \lessgtr \tau_{i_m}$, где $k \neq m$; $\lessgtr$ — знак несравнимости, т. е. неподчинённости по корневому дереву иерархического классификатора (если объект помечен рубрикой родителя, то он содержит тематику и всех подчинённых рубрик-потомков, так что указывать их в его тематическом индексе нет необходимости);

б) в подмножестве $\{\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_L}\}$ не содержится полного набора сыновей ни одной из вершин корневого дерева иерархического классификатора (если тематика объекта содержит полный набор сыновей какой-либо рубрики, то в тематический индекс объекта вместо этого набора сыновей необходимо и достаточно внести родительскую рубрику).

На рис. 4 наборы рубрик $\mathcal{T}_1^{\mathsf{M}} = \{\tau_2, \tau_{12}, \tau_{19}\}$ и $\mathcal{T}_2^{\mathsf{M}} = \{\tau_7, \tau_9, \tau_{13}, \tau_{15}, \tau_{16}\}$ являются мультирубриками, наборы рубрик $\mathcal{T}_1 = \{\tau_6, \tau_{14}\}$ и $\mathcal{T}_2 = \{\tau_{13}, \tau_{17}, \tau_{18}\}$ требованиям для мультирубрики не удовлетворяют, поскольку τ_6 и τ_{14} связаны отношением «родитель — потомок», а τ_{17} и τ_{18} образуют полный набор сыновей рубрики τ_{12}.

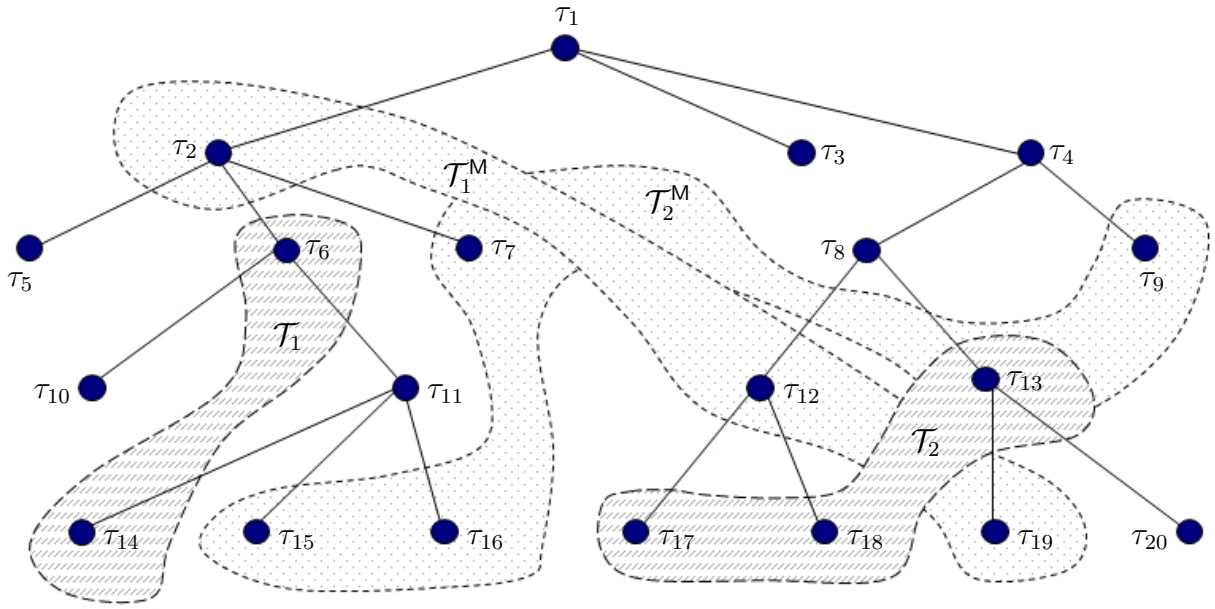


Рис. 4. Примеры наборов рубрик, являющихся мультирубриками ($\mathcal{T}_1^{\mathsf{M}}, \mathcal{T}_2^{\mathsf{M}}$) и не являющихся мультирубриками ($\mathcal{T}_1, \mathcal{T}_2$)

Множество всех мультирубрик на корневом дереве классификатора будем обозначать T^{M} ($\mathcal{T}_i^{\mathsf{M}} \in T^{\mathsf{M}}$).

Отметим, что по определению 1 мультирубриками являются также набор из одной рубрики (вершины классификатора) и пустой набор рубрик-вершин $\varnothing \in T^{\mathsf{M}}$.

Поскольку рубрики τ_k, входящие в различные мультирубрики, могут находиться в отношениях подчинённости («родитель — потомок»), то отношения «шире — уже»,

т. е. доминирования одной мультирубрики над другой, не могут адекватно воспроизводиться отношением теоретико-множественного включения.

Определение 2. Мультирубрика $\mathcal{T}_i^{\mathsf{M}} = \{\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_I}\}$ доминирует над мультирубрикой $\mathcal{T}_j^{\mathsf{M}} = \{\tau_{j_1}, \tau_{j_2}, \ldots, \tau_{j_J}\}$ (шире или равна по тематике; для обозначения доминирования мультирубрик будем использовать знак $\leqslant_{\mathsf{M}}$) в том и только в том случае, когда для любого $m = 1, \ldots, J$ существует $n = 1, \ldots, I$, такое, что $\tau_{j_m} \leqslant \tau_{i_n}$ (вершина $\tau_{j_m} \in \mathcal{T}_j^{\mathsf{M}}$ подчинена по корневому дереву иерархического классификатора вершине $\tau_{i_n} \in \mathcal{T}_i^{\mathsf{M}}$ или τ_{j_m} и τ_{i_n} совпадают):

$$\mathcal{T}_j^{\mathsf{M}} \leqslant_{\mathsf{M}} \mathcal{T}_i^{\mathsf{M}} \quad \Leftrightarrow \quad \forall \tau_{j_m} \in \mathcal{T}_j^{\mathsf{M}}\, \exists \tau_{i_n} \in \mathcal{T}_i^{\mathsf{M}}\, (\tau_{j_m} \leqslant \tau_{i_n}).$$

Заметим, что если ни $\mathcal{T}_i^{\mathsf{M}}$ не доминирует над $\mathcal{T}_j^{\mathsf{M}}$, ни $\mathcal{T}_j^{\mathsf{M}}$ не доминирует над $\mathcal{T}_i^{\mathsf{M}}$, то мультирубрики $\mathcal{T}_i^{\mathsf{M}}$ и $\mathcal{T}_j^{\mathsf{M}}$ являются *несравнимыми*; для обозначения этого будем использовать знак $\leqslant_{\mathsf{M}}\geqslant$. При этом часть рубрик-вершин $\tau_{i_n} \in \mathcal{T}_i^{\mathsf{M}}$ и $\tau_{j_m} \in \mathcal{T}_j^{\mathsf{M}}$ могут совпадать или находиться в отношении «родитель — потомок».

На рис. 5 приведены примеры доминирования и несравнимости мультирубрик.

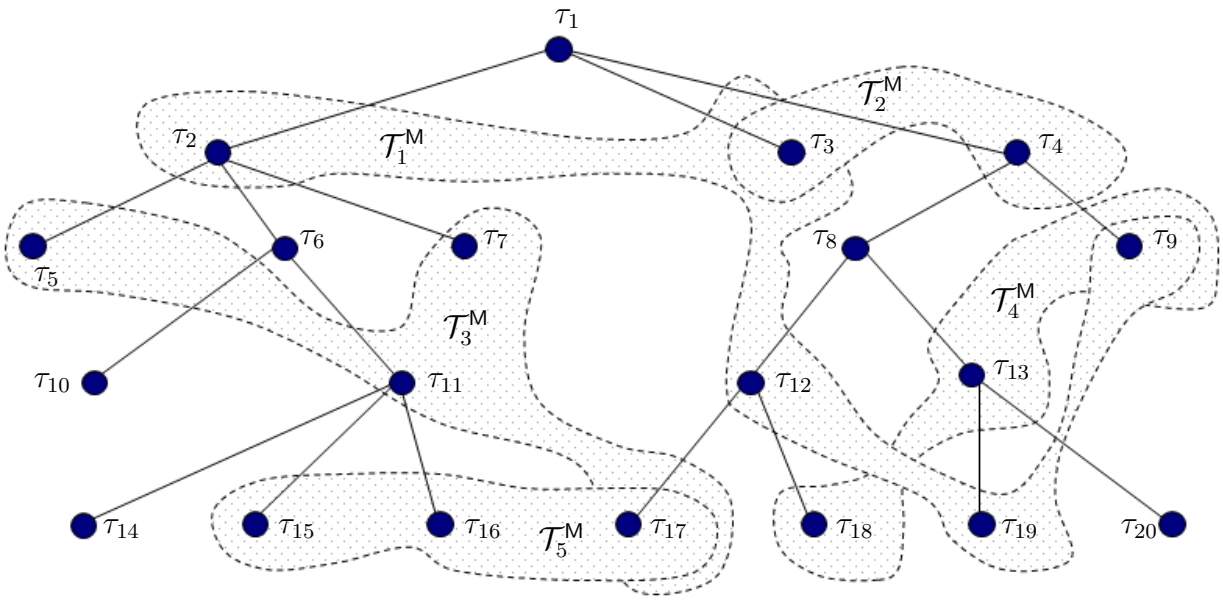


Рис. 5. Примеры доминирования и несравнимости мультирубрик: $\mathcal{T}_5^{\mathsf{M}} \leqslant_{\mathsf{M}} \mathcal{T}_3^{\mathsf{M}} \leqslant_{\mathsf{M}} \mathcal{T}_1^{\mathsf{M}}$, $\mathcal{T}_4^{\mathsf{M}} \leqslant_{\mathsf{M}} \mathcal{T}_2^{\mathsf{M}}$, $\mathcal{T}_1^{\mathsf{M}} \leqslant_{\mathsf{M}}\geqslant \mathcal{T}_2^{\mathsf{M}}$, $\mathcal{T}_2^{\mathsf{M}} \leqslant_{\mathsf{M}}\geqslant \mathcal{T}_5^{\mathsf{M}}$, $\mathcal{T}_3^{\mathsf{M}} \leqslant_{\mathsf{M}}\geqslant \mathcal{T}_2^{\mathsf{M}}$

Таким образом, определение 2 устанавливает на множестве мультирубрик T^{M} отношение частичного порядка.

Для построения решётки мультирубрик необходимо определить операции получения точных границ (наименьшей верхней и наибольшей нижней) для любых пар мультирубрик. Как и в случае с отношением доминирования, операции теоретико-множественного объединения и пересечения в отношении наборов рубрик, составляющих мультирубрики, не подходят.

Определение 3. Иерархическим сжатием $\sqcup_{\mathsf{H}}$ множества элементов $\{\tau_{k_1}, \tau_{k_2}, \ldots, \tau_{k_L}\}$ $(L < K)$, являющихся вершинами корневого дерева, задающего частичный порядок на множестве рубрик иерархического классификатора $T_{\mathsf{H}} = \{\tau_1, \tau_2, \ldots, \tau_K\}$, будем называть итеративную операцию замены вершинами-родителями любых подмножеств элементов, являющихся в множестве $\{\tau_{k_1}, \tau_{k_2}, \ldots, \tau_{k_L}\}$ полными наборами сыновей соответствующих вершин-родителей.

К примеру, иерархическое сжатие набора вершин $\mathcal{T}_2 = \{\tau_{13}, \tau_{17}, \tau_{18}\}$ на рис. 4 даёт набор из одной вершины-рубрики $\{\tau_8\}$:

$$\sqcup_{\mathsf{H}}(\mathcal{T}_2) = \sqcup_{\mathsf{H}}\{\tau_{13}, \tau_{17}, \tau_{18}\} = \sqcup_{\mathsf{H}}\{\tau_{13}, \tau_{12}\} = \{\tau_8\}.$$

Отметим, что иерархическое сжатие по набору рубрик какой-либо мультирубрики является в соответствии с определением 1 той же мультирубрикой:

$$\sqcup_{\mathsf{H}}(\mathcal{T}_i^{\mathsf{M}}) = \sqcup_{\mathsf{H}}\{\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_I}\} = \{\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_I}\} = \mathcal{T}_i^{\mathsf{M}}.$$

Введём понятия объединения и пересечения мультирубрик.

Определение 4. Объединением $\cup_{\mathsf{M}}$ мультирубрик $\mathcal{T}_i^{\mathsf{M}} = \{\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_I}\}$ и $\mathcal{T}_j^{\mathsf{M}} = \{\tau_{j_1}, \tau_{j_2}, \ldots, \tau_{j_J}\}$ называется операция формирования множества вершин иерархического рубрикатора $\mathcal{T}^{\mathsf{M}\cup} = \mathcal{T}_i^{\mathsf{M}} \cup_{\mathsf{M}} \mathcal{T}_j^{\mathsf{M}}$ на основе следующего алгоритма:

а) формируется теоретико-множественное объединение $\mathcal{T}^{\cup}$ множеств вершин, составляющих мультирубрики:

$$\mathcal{T}^{\cup} = \{\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_I}\} \cup \{\tau_{j_1}, \tau_{j_2}, \ldots, \tau_{j_J}\};$$

б) формируется набор вершин $\mathcal{T}^{\cup'}$ путем исключения из набора $\mathcal{T}^{\cup}$ тех вершин τ_k, для которых хотя бы одна вершина из того же набора $\mathcal{T}^{\cup}$ является предком:

$$(\tau_k \in \mathcal{T}^{\cup} \wedge \tau_k \notin \mathcal{T}^{\cup'}) \quad \Leftrightarrow \quad \exists \tau_m \in \mathcal{T}^{\cup}\,(\tau_m < \tau_k);$$

в) формируется итоговый набор вершин путём иерархического сжатия набора вершин $\mathcal{T}^{\cup'}$:

$$\mathcal{T}^{\mathsf{M}\cup} = \sqcup_{\mathsf{H}}(\mathcal{T}^{\cup'}).$$

Определение 5. Пересечением $\cap_{\mathsf{M}}$ мультирубрик $\mathcal{T}_i^{\mathsf{M}} = \{\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_I}\}$ и $\mathcal{T}_j^{\mathsf{M}} = \{\tau_{j_1}, \tau_{j_2}, \ldots, \tau_{j_J}\}$ называется операция формирования множества вершин иерархического классификатора $\mathcal{T}^{\mathsf{M}\cap} = \mathcal{T}_i^{\mathsf{M}} \cap_{\mathsf{M}} \mathcal{T}_j^{\mathsf{M}}$ на основе следующего алгоритма:

а) из множества вершин мультирубрики $\mathcal{T}_i^{\mathsf{M}}$ формируется множество вершин $\mathcal{T}_i^{\mathsf{M}'}$, для которых в наборе вершин мультирубрики $\mathcal{T}_j^{\mathsf{M}}$ найдётся совпадающая вершина или вершина-предок:

$$(\tau_k \in \mathcal{T}_i^{\mathsf{M}} \wedge \tau_k \in \mathcal{T}_i^{\mathsf{M}'}) \quad \Leftrightarrow \quad \exists \tau_m \in \mathcal{T}_j^{\mathsf{M}}\,(\tau_k \leqslant \tau_m);$$

б) из множества вершин мультирубрики $\mathcal{T}_j^{\mathsf{M}}$ формируется множество вершин $\mathcal{T}_j^{\mathsf{M}'}$, для которых в наборе вершин мультирубрики $\mathcal{T}_i^{\mathsf{M}}$ найдётся совпадающая вершина или вершина-предок:

$$(\tau_k \in \mathcal{T}_j^{\mathsf{M}} \wedge \tau_k \in \mathcal{T}_j^{\mathsf{M}'}) \quad \Leftrightarrow \quad \exists \tau_m \in \mathcal{T}_i^{\mathsf{M}}\,(\tau_k \leqslant \tau_m);$$

в) осуществляется теоретико-множественное объединение множеств $\mathcal{T}_i^{\mathsf{M}'}$ и $\mathcal{T}_j^{\mathsf{M}'}$:

$$\mathcal{T}^{\mathsf{M}\cap} = \mathcal{T}_i^{\mathsf{M}'} \cup_{\mathsf{M}} \mathcal{T}_j^{\mathsf{M}'}.$$

Отметим, что по алгоритму определения 5, если мультирубрики $\mathcal{T}_i^{\mathsf{M}}$ и $\mathcal{T}_j^{\mathsf{M}}$ содержат только несравнимые вершины, то результатом их пересечения будет пустое множество.

Нетрудно доказать [6], что множества $\mathcal{T}^{\mathsf{M}\cup} = \mathcal{T}_i^{\mathsf{M}} \cup_{\mathsf{M}} \mathcal{T}_j^{\mathsf{M}}$ и $\mathcal{T}^{\mathsf{M}\cap} = \mathcal{T}_i^{\mathsf{M}} \cap_{\mathsf{M}} \cap_{\mathsf{M}} \mathcal{T}_j^{\mathsf{M}}$ являются а) мультирубриками и б) наименьшей верхней границей и наибольшей нижней границей для мультирубрик $\mathcal{T}_i^{\mathsf{M}}$ и $\mathcal{T}_j^{\mathsf{M}}$ соответственно.

На рис. 6 приведён пример объединения и пересечения мультирубрик, иллюстрирующий, насколько данные операции отличаются от обычных операций объединения и пересечения множеств.

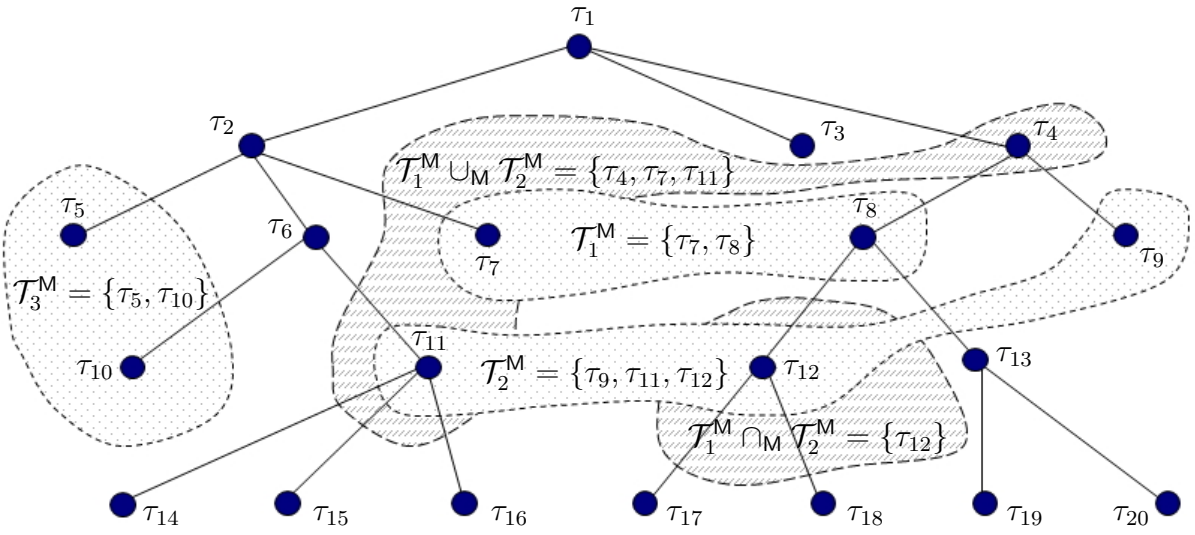


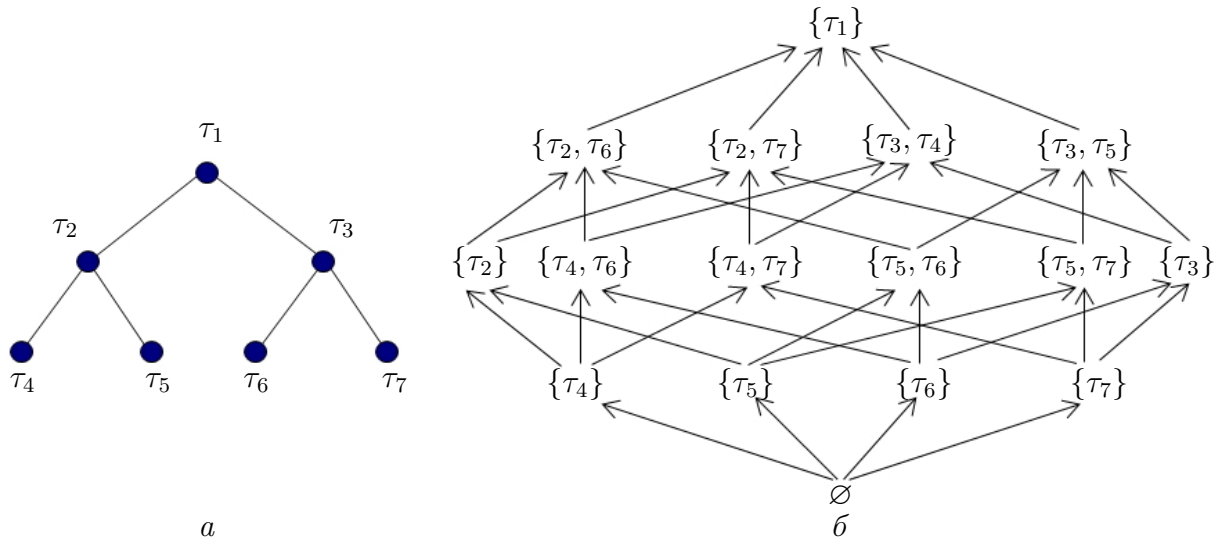
Рис. 6. Пример объединения и пересечения мультирубрик: $\mathcal{T}_1^{\mathsf{M}} \cup_{\mathsf{M}} \mathcal{T}_2^{\mathsf{M}} = \{\tau_7, \tau_8\} \cup_{\mathsf{M}} \cup_{\mathsf{M}} \{\tau_9, \tau_{11}, \tau_{12}\} = \{\tau_4, \tau_7, \tau_{11}\}$; $\mathcal{T}_1^{\mathsf{M}} \cap_{\mathsf{M}} \mathcal{T}_2^{\mathsf{M}} = \{\tau_7, \tau_8\} \cap_{\mathsf{M}} \{\tau_9, \tau_{11}, \tau_{12}\} = \{\tau_{12}\}$; $\mathcal{T}_1^{\mathsf{M}} \cap_{\mathsf{M}} \mathcal{T}_3^{\mathsf{M}} = \varnothing$; $\mathcal{T}_2^{\mathsf{M}} \cap_{\mathsf{M}} \mathcal{T}_3^{\mathsf{M}} = \varnothing$

Заметим, что объединение и пересечение мультирубрик, одна из которых доминирует над другой, дает доминирующую или доминируемую мультирубрику соответственно:

$$\mathcal{T}_j^{\mathsf{M}} \leqslant_{\mathsf{M}} \mathcal{T}_i^{\mathsf{M}} \Rightarrow \mathcal{T}_j^{\mathsf{M}} \cup_{\mathsf{M}} \mathcal{T}_i^{\mathsf{M}} = \mathcal{T}_i^{\mathsf{M}}, \quad \mathcal{T}_j^{\mathsf{M}} \cap_{\mathsf{M}} \mathcal{T}_i^{\mathsf{M}} = \mathcal{T}_j^{\mathsf{M}}.$$

В результате имеем решётку мультирубрик $\Lambda_{\mathsf{TH}}(T^{\mathsf{M}}, \leqslant_{\mathsf{M}}, \cup_{\mathsf{M}}, \cap_{\mathsf{M}})$.

На рис. 7 для простейшего иерархического рубрикатора приведена диаграмма Хассе решётки мультирубрик.



а *б*

Рис. 7. Иерархический рубрикатор (*а*); диаграмма Хассе решётки мультирубрик для него (*б*)

Используя линейную решётку уровней безопасности $\Lambda_{\mathsf{L}}(L, <, \circ, \otimes)$ и решётку мультирубрик $\Lambda_{\mathsf{TH}}(T^{\mathsf{M}}, \leqslant_{\mathsf{M}}, \cup_{\mathsf{M}}, \cap_{\mathsf{M}})$, можно построить модель безопасности, основанную

на объединении модели Белла — ЛаПадулы и тематико-иерархического разграничения доступа.

3. MLTHS-система

Будем строить MLTHS-систему на базе аппарата *субъектно-объектных моделей конечных состояний* [1, 3, 7], основные положения которых сводятся к следующему:

1. В контексте безопасности компьютерная система представляется совокупностью *объектов* доступа $o \in O$ и управляемых пользователями *субъектов* доступа $s \in S$.

2. Объекты доступа $o \in O$ рассматриваются как слова (совокупность слов, запись), характеризующие относительно некоторого языка **Я** состояния логических (файлы, каталоги, таблицы и т. д.) или физических элементов (порты, принтеры, приводы и т. д.).

3. Субъекты доступа $s \in S$ рассматриваются как активизированные состояния специальных объектов (*объектов-источников*), описывающих относительно некоторого языка *преобразования информации* (отображение одного слова в другое), при котором инициализированным, т. е. действующим преобразованиям выделены определённые вычислительные ресурсы (*домен преобразования*) и передано управление компьютерной системой. В некоторых источниках вместо понятия «домен субъекта доступа» используют понятие «функционально или параметрически ассоциированных с субъектом объектов доступа».

4. Субъекты осуществляют *доступы* к объектам (рис. 8), в результате которых возникают *информационные потоки*, заключающиеся в изменении слов, характеризующих или объект доступа (доступ *на запись* в объект — `w`), или *домен* субъекта доступа (доступ *на чтение* из объекта — `r`), или инициализирующих активизацию объектов-источников с выделением домена и передачей управления (доступ *на выполнение* объекта-источника — `e`).

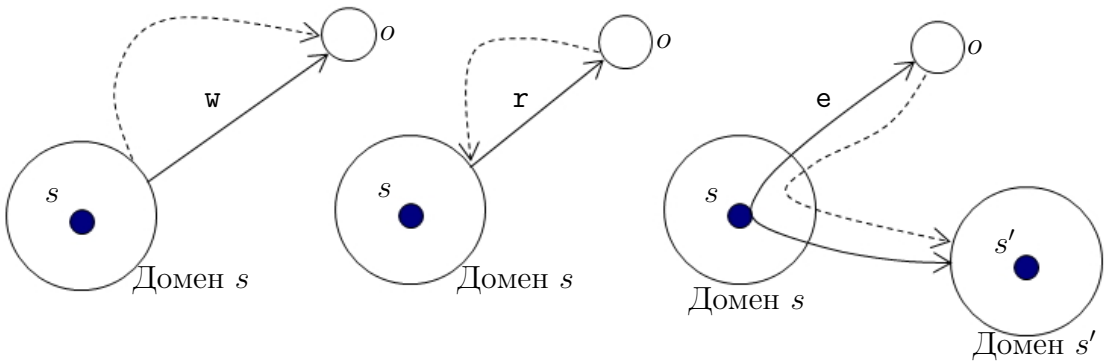


Рис. 8. Доступы на запись (`w`), чтение (`r`), выполнение (`e`) и соответствующие им информационные потоки (штриховые линии со стрелками)

5. В компьютерной системе действует дискретное время, в каждый момент которого t_k состояние системы V_k характеризуется определённой декомпозицией на множество субъектов S, осуществляющих доступы на множестве объектов O, в результате чего формируются информационные потоки, переводящие в момент времени t_{k+1} систему в новое состояние V_{k+1}.

6. Помимо субъектов доступа, управляемых пользователями, в системе изначально (в том числе в момент времени t_0) присутствуют *системные субъекты*, используя сервис которых, пользователи, начиная работу в системе, инициализируют свои первичные субъекты.

7. *Информационные потоки*, вызываемые доступами субъектов к объектам, по правилам (критериям) политики безопасности разделяются на неопасные (*допустимые*) и опасные (*недопустимые*), результатом которых может быть нарушение конфиденциальности и/или целостности и/или правомерной доступности объектов (информации).

8. В подмножестве системных субъектов действует *монитор безопасности*, санкционирующий по правилам (критериям) политики безопасности запросы субъектов на доступ к объектам.

9. Проблема безопасности, которую должна разрешать модель безопасности, заключается в формальном доказательстве того факта, что при условии отсутствия опасных (недопустимых) информационных потоков в начальном состоянии V_0 и санкционировании монитором безопасности только тех запросов на доступ, которые удовлетворяют правилам (критериям) безопасности, при переходе в состояние V_k не возникло недопустимых информационных потоков.

Основные положения MLTHS-системы сводятся к следующему.

Положение 1. Тематика информации, обрабатываемой в компьютерной системе, отображается иерархическим тематическим классификатором, представляющим собой конечное множество тематических рубрик $T_{\mathsf{H}} = \{\tau_1, \tau_2, \ldots, \tau_K\}$, на котором установлен частичный порядок, задаваемый корневым деревом, и множество мультирубрик которого представляет решётку $\Lambda_{\mathsf{TH}}(T^{\mathsf{M}}, \leqslant_{\mathsf{M}}, \cup_{\mathsf{M}}, \cap_{\mathsf{M}})$.

Положение 2. Конфиденциальность (ценность) информации измеряется в порядковой шкале L, конечное множество уровней которой является линейной решёткой уровней безопасности $\Lambda_{\mathsf{L}}(L, <, \circ, \otimes)$.

Положение 3. Множество сущностей системы (субъектов S и объектов доступа O) отображается в произведение линейной решётки уровней безопасности $\Lambda_{\mathsf{L}}(L, <, \circ, \otimes)$ и решётки мультирубрик $\Lambda_{\mathsf{TH}}(T^{\mathsf{M}}, \leqslant_{\mathsf{M}}, \cup_{\mathsf{M}}, \cap_{\mathsf{M}})$.

На рис. 9 представлена диаграмма Хассе произведения простейшей решётки из двух уровней безопасности ($l_1 > l_2$) и решётки мультирубрик (рис. 7, *б*) для простейшего иерархического тематического классификатора.

Произведение решёток $\Lambda_{\mathsf{L}}(L, <, \circ, \otimes)$ и $\Lambda_{\mathsf{TH}}(T^{\mathsf{M}}, \leqslant_{\mathsf{M}}, \cup_{\mathsf{M}}, \cap_{\mathsf{M}})$ будем называть *мандатной тематико-иерархической решёткой*.

В результате каждый экземпляр сущности $x \in (S \cup O)$ характеризуется меткой безопасности $\mathcal{F}_{\mathsf{LTH}}(x)$, состоящей из уровня безопасности $l \in L$ и мультирубрики $\mathcal{T}^{\mathsf{M}} \in T^{\mathsf{M}}$: $\mathcal{F}_{\mathsf{LTH}}(x) = \{l_x, \mathcal{T}_x^{\mathsf{M}}\}$.

Заметим, что из $\mathcal{F}_{\mathsf{LTH}}(x_1) \leqslant \mathcal{F}_{\mathsf{LTH}}(x_2)$ следует $(l_{x_1} \leqslant l_{x_2}) \wedge (\mathcal{T}_{x_1}^{\mathsf{M}} \leqslant_{\mathsf{M}} \mathcal{T}_{x_1}^{\mathsf{M}})$. В противном случае $\mathcal{F}_{\mathsf{LTH}}(x_1)$ и $\mathcal{F}_{\mathsf{LTH}}(x_2)$ *несравнимы* (рис. 9). Таким образом, на множестве меток безопасности $\mathcal{F}_{\mathsf{LTH}}(x)$ имеется частичный порядок. Далее, если $\mathcal{F}_{\mathsf{LTH}}(x_1) \leqslant \mathcal{F}_{\mathsf{LTH}}(x_2)$, то будем говорить, что метка $\mathcal{F}_{\mathsf{LTH}}(x_2)$ выше (или равна) метки $\mathcal{F}_{\mathsf{LTH}}(x_1)$.

Положение 4. Доверительно-тематические *полномочия пользователей* $\mathcal{F}_{\mathsf{LTH}}(s) = \{l_s, \mathcal{T}_s^{\mathsf{M}}\}$ задаются внешним по отношению к компьютерной системе фактором посредством присвоения их регистрационным записям меток безопасности. Таким образом, вопросы безопасности, связанные с изменением уровней допуска и/или тематических полномочий пользователей, моделью MLTHS-системы не охватываются.

Положение 5. Все системные субъекты, кроме монитора безопасности, имеют метку безопасности $(l_{\min}, \varnothing)$. Монитор безопасности имеет метку $(l_{\max}, \{\tau_1\})$.

Положение 6. Пользователи, начиная работу в системе, посредством сервиса системных субъектов инициализируют свои первичные субъекты, которым монитор безопасности присваивает метки безопасности их регистрационных записей.

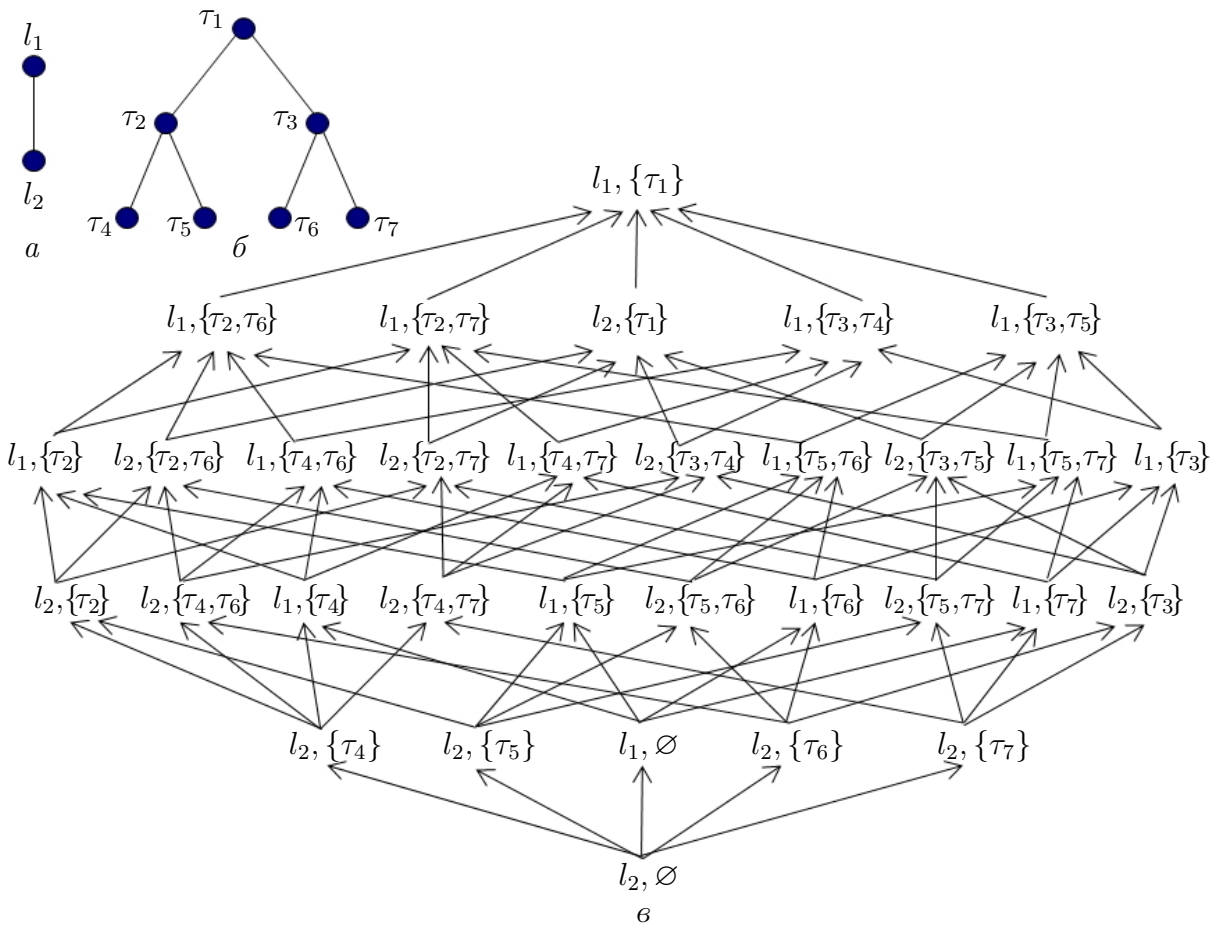


Рис. 9. Решётка уровней безопасности ($а$); иерархический тематический классификатор ($б$); диаграмма Хассе произведения решётки уровней безопасности и решётки мультирубрик ($в$)

Определение 6 (критерий безопасности). MLTHS-система безопасна тогда и только тогда, когда в ней отсутствуют информационные потоки следующих видов: от сущностей с более высокой меткой безопасности к сущностям с более низкой меткой безопасности; между несравнимыми по меткам безопасности сущностями.

Положение 7. Переходы системы из одного состояния V_k в другое состояние V_{k+1}, обусловленные осуществлением доступов существующих субъектов к существующим объектам, санкционируются монитором безопасности на основе следующих правил, вытекающих из критерия, задаваемого определением 6:

П р а в и л о 1. Доступ субъекта s к объекту o, вызывающий поток по чтению (`r`), неопасен и может быть разрешён монитором безопасности тогда и только тогда, когда метка безопасности субъекта доминирует над меткой безопасности объекта:

$$\mathcal{F}_{\mathsf{LTH}}[s] \geqslant \mathcal{F}_{\mathsf{LTH}}[o].$$

П р а в и л о 2. Доступ субъекта s к объекту o, вызывающий поток по записи (`w`), неопасен и может быть разрешён монитором безопасности тогда и только тогда, когда метка безопасности объекта доминирует над меткой безопасности субъекта:

$$\mathcal{F}_{\mathsf{LTH}}[o] \geqslant \mathcal{F}_{\mathsf{LTH}}[s].$$

Положение 8. Переходы системы из одного состояния V_k в другое состояние V_{k+1}, связанные с созданием новых объектов, санкционируются монитором безопасности на основе следующего правила:

П р а в и л о 3. Создание субъектом s нового объекта o', в том числе за счёт чтения из другого объекта o, вызывает неопасный поток и может быть разрешено монитором безопасности тогда и только тогда, когда метка безопасности субъекта доминирует над меткой объекта o, при этом монитор безопасности присваивает новому объекту o' мультирубрику, доминирующую над мультирубрикой субъекта:

$$\mathcal{F}_{\mathsf{LTH}}[o] \leqslant \mathcal{F}_{\mathsf{LTH}}[s] \leqslant \mathcal{F}_{\mathsf{LTH}}[o'].$$

Присвоение новому объекту метки безопасности, строго большей, чем метка безопасности субъекта ($\mathcal{F}_{\mathsf{LTH}}[s] < \mathcal{F}_{\mathsf{LTH}}[o']$), осуществляется монитором безопасности по специальному запросу к субъекту. В противном случае новому объекту присваивается метка безопасности субъекта ($\mathcal{F}_{\mathsf{LTH}}[s] \equiv \mathcal{F}_{\mathsf{LTH}}[o']$).

С учётом положений 4 и 6, при инициализации новых субъектов действует следующее правило:

П р а в и л о 4. Инициализация субъектом s нового субъекта s' посредством воздействия на объект-источник o (доступ $\mathtt{e}$) вызывает неопасный поток и может быть разрешена монитором безопасности тогда и только тогда, когда метка безопасности субъекта доминирует над меткой безопасности объекта-источника, при этом монитор безопасности присваивает новому субъекту метку безопасности, тождественную метке безопасности инициализирующего субъекта:

$$\mathcal{F}_{\mathsf{LTH}}[o] \leqslant \mathcal{F}_{\mathsf{LTH}}[s] \equiv \mathcal{F}_{\mathsf{LTH}}[s'].$$

Положение 9. Переходы системы из одного состояния V_k в другое состояние V_{k+1}, связанные с одновременными множественными доступами с учётом транзитивности информационных потоков и отношений доминирования меток безопасности, осуществляются на основе следующего правила:

П р а в и л о 5. Одновременный доступ субъекта s к объектам $o_1, o_2, \ldots$ или субъектов $s_1, s_2, \ldots$ к объекту o может быть разрешён (неопасен) тогда и только тогда, когда каждый одиночный доступ из запрашиваемой совокупности доступов удовлетворяет правилам 1–4.

В технологическом отношении реализация правила 5 может осуществляться на основе операций взятия наименьшей верхней границы или наибольшей нижней границы меток безопасности объектов $o_1, o_2, \ldots$ или субъектов $s_1, s_2, \ldots$ (рис. 10).

Условия доступа на чтение ($\mathtt{r}$) субъекта s одновременно к нескольким объектам $o_1, o_2, \ldots$:

$$(l_s \geqslant \max[l_{o_1}, l_{o_2}, \ldots]) \wedge (\mathcal{T}_s^{\mathsf{M}} \geqslant_{\mathsf{M}} (\mathcal{T}_{o_1}^{\mathsf{M}} \cup_{\mathsf{M}} \mathcal{T}_{o_2}^{\mathsf{M}} \cup_{\mathsf{M}} \ldots)).$$

Условия доступа на запись ($\mathtt{w}$) субъекта s одновременно к нескольким объектам $o_1, o_2, \ldots$:

$$(l_s \leqslant \min[l_{o_1}, l_{o_2}, \ldots]) \wedge (\mathcal{T}_s^{\mathsf{M}} \leqslant_{\mathsf{M}} (\mathcal{T}_{o_1}^{\mathsf{M}} \cap_{\mathsf{M}} \mathcal{T}_{o_2}^{\mathsf{M}} \cap_{\mathsf{M}} \ldots)).$$

Условия доступа на чтение ($\mathtt{r}$) одновременно нескольких субъектов $s_1, s_2, \ldots$ к одному объекту o:

$$(l_o \leqslant \min[l_{s_1}, l_{s_2}, \ldots]) \wedge (\mathcal{T}_o^{\mathsf{M}} \leqslant_{\mathsf{M}} (\mathcal{T}_{s_1}^{\mathsf{M}} \cap_{\mathsf{M}} \mathcal{T}_{s_2}^{\mathsf{M}} \cap_{\mathsf{M}} \ldots)).$$

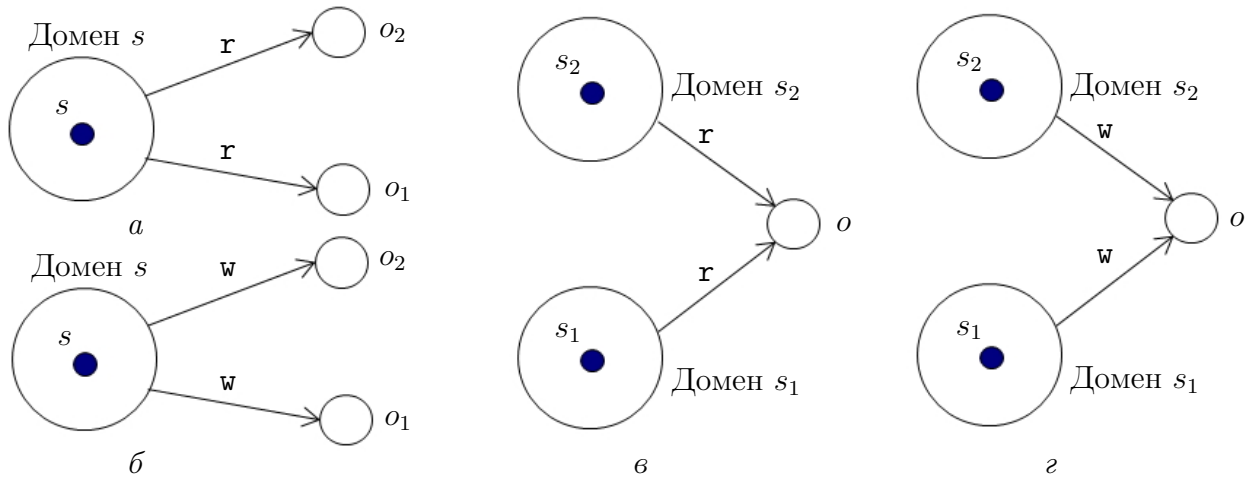


Рис. 10. Множественные доступы: a — одного субъекта по чтению одновременно к двум объектам; $б$ — одного субъекта по записи одновременно к двум объектам; $в$ — по чтению к одному объекту одновременно двух субъектов; $г$ — по записи к одному объекту одновременно двух субъектов

Условия доступа на запись (w) одновременно нескольких субъектов $s_1, s_2, \ldots$ к одному объекту o:

$$(l_o \geqslant \max[l_{s_1}, l_{s_2}, \ldots]) \wedge (\mathcal{T}_o^{\mathsf{M}} \geqslant_{\mathsf{M}} (\mathcal{T}_{s_1}^{\mathsf{M}} \cup_{\mathsf{M}} \mathcal{T}_{s_2}^{\mathsf{M}} \cup_{\mathsf{M}} \ldots)).$$

Справедливо следующее утверждение:

Утверждение 1. В MLTHS-системе реализуется множество только таких потоков, которые удовлетворяют критерию безопасности по определению 6.

Доказательство. В соответствии с положениями субъектно-объектных моделей конечных состояний в начальном состоянии V_0 потоки от сущностей с более высокой (доминирующей) меткой безопасности к сущностям с более низкой меткой безопасности, а также потоки между несравнимыми по меткам безопасности сущностями отсутствуют. Необходимо доказать, что при последовательном переходе системы в состояние V_k на основе доступов, которые санкционируются по правилам 1–5, таких потоков также не появится.

Утверждение очевидно для переходов, реализуемых на основе одиночных доступов, которые санкционируются по правилам 1–4, в том числе и для потоков, связанных с созданием новых объектов и инициализацией новых субъектов доступа.

В случае переходов, вызываемых одновременными множественными доступами только по чтению или только по записи (одним субъектом одновременно к множеству объектов или множеством субъектов одновременно к одному объекту) также очевидно, что опасные потоки не могут реализоваться в соответствии с правилом 5.

Рассмотрим доступы одного субъекта одновременно к части объектов по чтению, а к части объектов по записи и, аналогично, доступы к одному объекту частью субъектов по чтению, а частью субъектов по записи. По свойству транзитивности такие доступы могут реализовывать информационные потоки от одного объекта к другому объекту или от одного субъекта к другому субъекту (рис. 11).

По условиям утверждения 1 при санкционировании доступа, вызывающего поток $o_1 \to o_2$, имеем

$$(\mathcal{F}_{\mathsf{LTH}}[s] \geqslant \mathcal{F}_{\mathsf{LTH}}[o_1]) \wedge (\mathcal{F}_{\mathsf{LTH}}[o_2] \geqslant \mathcal{F}_{\mathsf{LTH}}[s]).$$

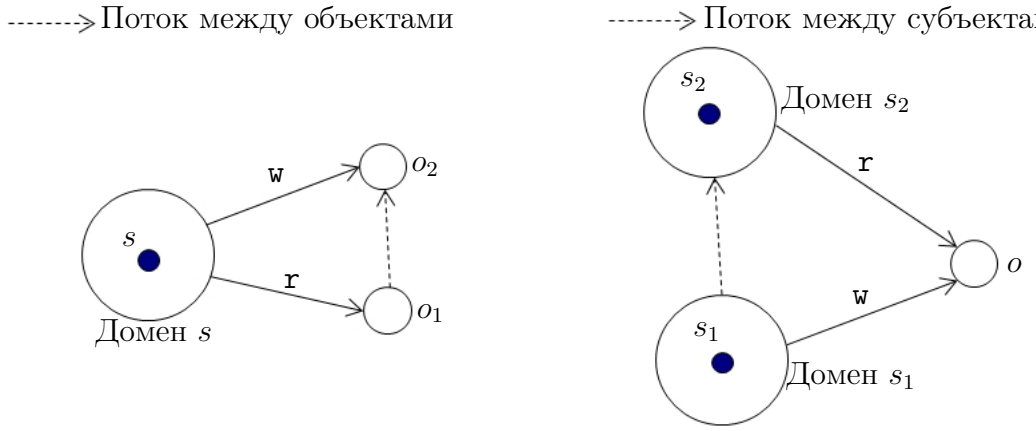


Рис. 11. Множественные доступы, вызывающие по свойству транзитивности потоки между объектами и между субъектами

Отсюда следует, что

$$\mathcal{F}_{\mathsf{LTH}}[o_2] \geqslant \mathcal{F}_{\mathsf{LTH}}[o_1].$$

Таким образом, осуществляется неопасный поток «снизу вверх» между сравнимыми по меткам безопасности сущностями.

Аналогично по условиям утверждения 1 при санкционировании потока $s_1 \to s_2$ имеем

$$(\mathcal{F}_{\mathsf{LTH}}[s_1] \leqslant \mathcal{F}_{\mathsf{LTH}}[o]) \wedge (\mathcal{F}_{\mathsf{LTH}}[s_2] \geqslant \mathcal{F}_{\mathsf{LTH}}[o]).$$

Отсюда следует, что

$$\mathcal{F}_{\mathsf{LTH}}[s_2] \geqslant \mathcal{F}_{\mathsf{LTH}}[s_1].$$

Таким образом, и в этом случае осуществляется неопасный поток «снизу вверх».

Нетрудно видеть, что любые другие множественные потоки являются комбинацией двух рассмотренных потоков $o_1 \to o_2$ и $s_1 \to s_2$, из чего по свойству транзитивности вытекает наличие в системе только неопасных потоков, т. е. потоков, удовлетворяющих критерию безопасности по определению 6. ■

Заключение

MLTHS-система, как и MLS-система с неиерархическими рубриками-категориями, реализуя одновременно доверительно-мандатный и тематический принципы, позволяет гораздо детальнее обеспечить разграничение доступа к информации. В классической модели Белла — ЛаПадулы для детализации доступа приходится вводить матрицу доступа, что соответственно привносит проблемы безопасности, присущие дискреционным моделям, в частности, проблему троянских программ в рамках разграничения доступа к объектам одного уровня безопасности.

При этом, в отличие от простейшего дескрипторно-тематического подхода в MLS-системе, тематический доступ в MLTHS-системе основывается на иерархических классификаторах, повсеместно применяемых в организации информационно-поисковых хранилищ документов. В результате имеется возможность соединить технологии текстового поиска в документальных хранилищах и технологии разграничения доступа, создавая на этой основе защищённые документально-поисковые системы без ограничения их функциональности.

ЛИТЕРАТУРА

1. *Грушо А. А., Применко Е. А., Тимонина Е. Е.* Теоретические основы компьютерной безопасности. М.: Издательский центр «Академия», 2009. 272 с.
2. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками: учеб. пособие для вузов. М.: Горячая линия — Телеком, 2011. 320 с.
3. *Гайдамакин Н. А.* Разграничение доступа к информации в компьютерных системах. Екатеринбург: Изд-во Урал. ун-та, 2003. 328 с.
4. *Bell D. E. and LaPadula L. J.* Secure Computers Systems: Unified Exposition and Multics Interpretation. Bedford, Mass.: MITRE Corp., 1976.
5. *Крюков К. В., Панкова Л. А., Пронина В. А. и др.* Меры семантической близости в онтологии // Пробл. управл. 2010. № 5. С. 2–14.
6. *Гайдамакин Н. А.* Модель тематического разграничения доступа к информации при иерархической структуре классификатора в автоматизированных системах управления // Автоматика и телемеханика. 2003. № 3. С. 177–189.
7. *Щербаков А. Ю.* Современная компьютерная безопасность. Теоретические основы. Практические аспекты. М.: Книжный мир, 2009. 352 с.

REFERENCES

1. *Grusho A. A., Primenko E. A., and Timonina E. E.* Teoreticheskie osnovy komp'yuternoy bezopasnosti [Theoretical Foundations of Computer Security]. Moscow, Akademiya Publ., 2009. 272 p. (in Russian)
2. *Devyanin P. N.* Modeli bezopasnosti komp'yuternykh sistem. Upravlenie dostupom i informatsionnymi potokami [Models of Computer Systems Security. Control of Access and Information Flows]. Moscow, Goryachaya liniya — Telekom, 2011. 320 p. (in Russian)
3. *Gaydamakin N. A.* Razgranichenie dostupa k informatsii v komp'yuternykh sistemakh [Differentiation of Access to Information in Computer Systems]. Ekaterinburg, USU Publ., 2003. 328 p. (in Russian)
4. *Bell D. E. and LaPadula L. J.* Secure Computers Systems: Unified Exposition and Multics Interpretation. Bedford, Mass., MITRE Corp., 1976.
5. *Kryukov K. V., Pankova L. A., Pronina V. A., et al.* Mery semanticheskoy blizosti v ontologii [Semantic similarity measures in ontology]. Control Science, 2010, no. 5, pp. 2–14. (in Russian)
6. *Gaidamakin N. A.* A model of thematic differentiation of access to information for the hierarchical classifier in automatic control systems. Automaton and Remote Control, 2003, vol. 64, iss. 3, pp. 505–516.
7. *Shcherbakov A. Yu.* Sovremennaya komp'yuternaya bezopasnost'. Teoreticheskie osnovy. Prakticheskie aspekty [Modern Computer Security. Theoretical Basis. Practical Aspects]. Moscow, Knizhnyy mir, 2009. 352 p. (in Russian)

УДК 004.94

УРОВЕНЬ ЗАПРЕЩАЮЩИХ РОЛЕЙ ИЕРАРХИЧЕСКОГО ПРЕДСТАВЛЕНИЯ МРОСЛ ДП-МОДЕЛИ

П. Н. Девянин

Федеральное учебно-методическое объединение в сфере высшего образования по УГСН 10.00.00 Информационная безопасность, г. Москва, Россия

Рассматривается построенный на основе базового уровня иерархического представления мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в операционных системах семейства Linux (МРОСЛ ДП-модели) уровень запрещающих ролей. В рамках этого уровня при реализации ролевого управления доступом кроме ролей и административных ролей определяется порядок использования запрещающих ролей, права доступа которых не разрешают, а наоборот, запрещают получение субъект-сессиями соответствующих доступов к сущностям. Приводятся примеры изменений в условиях и результатах применения де-юре правил преобразования состояний заданной в рамках этого уровня МРОСЛ ДП-модели абстрактной системы и формулируется утверждение о корректности этих правил.

Ключевые слова: *компьютерная безопасность, ролевое управление доступом, запрещающая роль.*

DOI 10.17223/20710410/39/5

THE LEVEL OF NEGATIVE ROLES OF THE HIERARCHICAL REPRESENTATION OF MROSL DP-MODEL

P. N. Devyanin

Federal Educational and Methodological Association in Information Security, Moscow, Russia

E-mail: devyanin.peter@yandex.ru

MROSL DP-model is widely used as a mandatory entity-role model of access and information flows security control in Linux-type OS. To make the model to be more adequate for a number of special security features of the Russian OS Astra Linux Special Edition, it has been decided to extend MROSL DP-model by adding to it so called negative roles. In contrast to the ordinary roles, these ones contain access rights which prohibit entities or subject-sessions from getting some access. In this paper, an order of using negative roles in MROSL DP-model is defined, the corresponding changes of conditions and application results for state transformation de-jure rules in MROSL DP-model with negative roles are described, and the correctness of these modified rules are stated, namely: let G and G' be some states of MROSL DP-model with negative roles, G' be a result of transformation de-jure rules application to G, and G be satisfying all the conditions for mandatory role access control; then G' also satisfies all these conditions.

Keywords: *computer security, role-based access control, negative role.*

Введение

Ролевое управление доступом RBAC [1, 2] изначально являлось основой мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в операционных системах семейства Linux (МРОСЛ ДП-модели) [2, 3]. При переходе к иерархическому представлению этой модели [4] именно ролевое управление доступом (без мандатного управления доступом и мандатного контроля целостности) задано на первом базовом уровне модели. На этом уровне по сложившейся традиции определены роли и административные роли, позволяющие субъект-сессиям соответственно получать через них права доступа к сущностям и другим субъект-сессиям и административные права доступа к ролям и административным ролям.

Однако практика использования МРОСЛ ДП-модели при реализации механизма управления доступом в отечественной защищенной операционной системе специального назначения (ОССН) Astra Linux Special Edition [5, 6] показала, что в ряде случаев применение только ролей, обладающих правами доступа к сущностям, разрешающим субъект-сессиям получение соответствующих доступов к ним, является недостаточным и создаёт неудобство при администрировании ОССН. В работе [7] описываются такие ситуации, когда, например, только одну учётную запись пользователя системы необходимо лишить права доступа, предоставляемого ей через роль, которой обладают все учётные записи пользователей системы, при этом все другие права доступа этой роли должны быть оставлены без изменений. Чтобы упростить администрирование ролевого управления доступом в подобных ситуациях, в этой же работе, а также в [8, 9] было предложено использование запрещающих ролей. Запрещающие роли, в отличие от «обычных», содержат права доступа к сущностям или субъект-сессиям, которые не разрешают, а наоборот, запрещают получение соответствующих доступов.

В то же время для анализа ролевого управления доступом с запрещающими ролями в [7] построена самостоятельная ДП-модель (мандатная сущностно-ролевая ДП-модель управления доступом в компьютерных сетях — МРОКС ДП-модель), элементы которой несколько отличаются от используемых в рамках МРОСЛ ДП-модели. Кроме того, в МРОКС ДП-модели запрещающие роли назначаются учётным записям пользователей через административные права доступа на чтение административных ролей. Такой подход может оказаться недостаточно удобным в реальной ОССН, так как, например, он не позволяет назначать явно запрещающие роли, соответствующие не только административным, но и «обычным» ролям.

В связи с этим в иерархическое представление МРОСЛ ДП-модели был добавлен описываемый в статье уровень запрещающих ролей. Этот уровень основан на базовом уровне модели, на нём не реализованы мандатное управление доступом и мандатный контроль целостности (это сделано автором на последующих уровнях модели), а для задания запрещающих ролей, в отличие от МРОКС ДП-модели, используется описанный ещё в ролевых моделях семейства RBAC механизм ограничений.

В п. 1 приводится описание элементов состояния рассматриваемой на уровне запрещающих ролей МРОСЛ ДП-модели абстрактной системы. В п. 2 задаётся порядок использования запрещающих ролей при реализации ролевого управления доступом. В п. 3 приводятся примеры изменений в условиях и результатах применения де-юре правил преобразования состояний системы и формулируется утверждение о корректности этих правил.

1. Состояние системы

С базового уровня МРОСЛ ДП-модели (его элементы и обозначения в целом соответствуют заданным в неиерархическом представлении МРОСЛ ДП-модели [2, 3]) на её уровне запрещающих ролей без изменений использованы следующие множества: U — учётных записей пользователей, S — субъект-сессий, O — объектов, C — контейнеров, E — сущностей, $NAMES$ — имён сущностей, R — ролей, AR — административных ролей, SAR — специальных административных ролей, R_r — видов прав доступа, R_a — видов доступа, P — прав доступа к сущностям и субъект-сессиям, A — доступов субъект-сессий к сущностям, G^* — всех возможных состояний системы, OP — правил преобразования состояний системы, U_ADMIN, U_ROLES — индивидуальных административных ролей и индивидуальных ролей учётных записей пользователей соответственно, $COMMON\_ROLES$ — общих ролей; функции: $user : S \to U$ — принадлежности субъект-сессии учётной записи пользователя, $entity\_name : C \times E \to 2^{NAMES}$ — имён сущностей, $H_E : E \to 2^E$ — иерархии сущностей, $H_S : S \to 2^S$ — иерархии субъект-сессий, $APA : AR \to 2^{AP}$ — административных прав доступа, $direct : E \cup R \cup AR \to \{\texttt{true}, \texttt{false}\}$ — вида метки; обозначения: $users\_admin\_role$, $entities\_admin\_role$, $subjects\_admin\_role$, $roles\_admin\_role$, $admin\_roles\_admin\_role$ — специальные административные роли, позволяющие администрировать учётные записи пользователей, сущности, субъект-сессии, роли или административные роли соответственно, u_admin, u_c, $common\_role$ — индивидуальные административные роли, роли и общие роли учётных записей пользователей, $G \vdash_{op} G'$ — переход системы из состояния G в состояние G' в результате применения правила op, $\Sigma(G^*, OP)$ — система, $\Sigma(G^*, OP, G_0)$ — система с начальным состоянием G_0.

Используем следующие дополнительные обозначения или переопределим некоторые обозначения базового уровня МРОСЛ ДП-модели:

- NR — множество запрещающих ролей, при этом по определению $AR \cap NR = R \cap NR = \varnothing$;
- $AP \subset (R \cup NR \cup AR) \times R_r$ — множество административных прав доступа к ролям или административным ролям (в данное множество добавлены права доступа к запрещающим ролям);
- $AA \subset S \times (R \cup NR \cup AR) \times R_a$ — множество доступов субъект-сессий к ролям, запрещающим ролям или административным ролям;
- $PA : R \cup NR \cup AR \to 2^P$ — функция прав доступа к сущностям ролей, запрещающих ролей и административных ролей, при этом для каждого права доступа $p \in P$ существует роль $r \in R \cup NR \cup AR$, такая, что выполняется условие $p \in PA(r)$;
- $shared\_container : C \cup R \cup NR \cup AR \to \{\texttt{true}, \texttt{false}\}$ — функция разделяемых контейнеров, такая, что сущность-контейнер, роль, запрещающая роль или административная роль $c \in C \cup R \cup NR \cup AR$ является разделяемой, когда $shared\_container(c) = \texttt{true}$, в противном случае $shared\_container(c) = \texttt{false}$;
- $role\_name : (R \cup NR \cup AR) \times (R \cup NR \cup AR) \to NAMES$ — функция имён ролей, запрещающих ролей и административных ролей в составе роли, запрещающей роли или административной роли (как контейнера);
- $H_R : R \cup NR \cup AR \to 2^R \cup 2^{NR} \cup 2^{AR}$ — функция иерархии ролей, запрещающих ролей или административных ролей, для задания которой по аналогии с базовым уровнем модели используется отношение частичного порядка $\leqslant$ на множестве $R \cup NR \cup AR$;
- $execute\_container : S \times C \times E \to \{\texttt{true}, \texttt{false}\}$ — функция доступа субъект-сессии к сущностям в контейнерах, такая, что по определению для субъект-

сессии $s \in S$, контейнера $c \in C$ и сущности $e \in E$ справедливо равенство $execute\_container(s, c, e) = \texttt{true}$ тогда и только тогда, когда существует последовательность сущностей $e_1, \ldots, e_n \in E$, где либо $n = 1$ и $c = e = e_1$, либо $n \geqslant 2$, $c = e_{n-1}$, $e = e_n$ и выполняются следующие условия:

У с л о в и е 1. Не существует сущности-контейнера $e_0 \in E$, такой, что $e_1 \in H_E(e_0)$ (без изменений по сравнению с базовым уровнем).

У с л о в и е 2. Верно $e_i \in H_E(e_{i-1})$, где $1 < i \leqslant n$ (без изменений по сравнению с базовым уровнем).

У с л о в и е 3. Существует роль или административная роль $r_i \in R \cup AR$, такая, что $(s, r_i, read_a) \in AA$ и $(e_i, execute_r) \in PA(r_i)$, где $1 \leqslant i < n$ (без изменений по сравнению с базовым уровнем).

У с л о в и е 4. Не существует запрещающей роли $nr \in NR$ и i, $1 \leqslant i < n$, таких, что $(s, nr, read_a) \in AA$ и $(e_i, execute_r) \in PA(nr)$.

Для обеспечения применения запрещающих ролей с использованием модели RBAC [1, 2] зададим механизм ограничений (*Constraints*) необходимого обладания запрещающей ролью с помощью следующей функции:

— $constraint_{NR} : R \cup AR \to 2^{NR}$ — функция необходимого обладания запрещающей ролью, задающая для каждой роли или административной роли множество запрещающих ролей, которые должна иметь субъект-сессия как текущие в случае обладания ею как текущей соответствующей ролью или административной ролью.

Пусть определена функция $constraint_{NR}$ необходимого обладания запрещающей ролью. Переопределим $G = (APA, PA, user, A, AA, H_R, H_E, H_S, constraint_{NR})$ — состояние системы.

2. Дополнительные условия реализации ролевого управления доступом с запрещающими ролями

Зададим порядок предоставления прав доступа и доступов к запрещающим ролям, основанный на порядке реализации ролевого управления доступом, определённом на базовом уровне МРОСЛ ДП-модели.

Предположение 1. В системе на уровне запрещающих ролей МРОСЛ ДП-модели выполняются следующие условия:

У с л о в и е 1 (создание, удаление, переименование запрещающих ролей, административные права доступа, иерархия и администрирование запрещающих ролей):

— все запрещающие роли являются «разделяемыми контейнерами»: для каждой запрещающей роли $nr \in NR$ справедливо равенство $shared\_container(nr) = \texttt{true}$;

— у каждой административной роли есть права доступа $execute_r$ ко всем запрещающим ролям: для каждой административной роли $ar \in AR$ и запрещающей роли $nr \in NR$ выполняется условие $(nr, execute_r) \in APA(ar)$;

— существует единственная специальная административная роль $negative\_roles\_admin\_role$, обладающая к каждой запрещающей роли административным правом доступа владения: для каждой запрещающей роли $nr \in NR$ выполняется условие $\{ar' \in AR : (nr, own_r) \in APA(ar')\} = \{negative\_roles\_admin\_role\}$, где $negative\_roles\_admin\_role \in SAR$;

— для создания, переименования или удаления запрещающей роли или «жёсткой» ссылки на неё в запрещающей роли субъект-сессии необходимо иметь к последней доступ на запись, текущую административную роль, обладающую к последней правом доступа на выполнение $execute_r$, и иметь административные досту-

пы на чтение и запись (кроме случая переименования) к административной роли $negative\_roles\_admin\_role$;

— для управления доступом к запрещающей роли, получения её параметров, числа «жёстких» ссылок к ней, множества административных ролей, обладающих к ней правами доступа, субъект-сессия должна иметь административный доступ на чтение к административной роли $negative\_roles\_admin\_role$, за исключением случаев, когда создаваемым административным ролям назначаются административные права доступа $execute_r$ ко всем запрещающим ролям.

У с л о в и е 2 (администрирование функции необходимого обладания запрещающей ролью, получение её значения):

— для специальных административных ролей из множества SAR не задаются запрещающие роли: для $ar \in SAR$ верно $constraint_{NR}(ar) = \varnothing$;

— для изменения значения функции необходимого обладания запрещающей ролью $constraint_{NR}$ субъект-сессия должна иметь административный доступ на чтение к административной роли $negative\_roles\_admin\_role$, а также в зависимости от того, является параметр функции ролью или административной ролью, субъект-сессия должна иметь административный доступ на чтение к административной роли $roles\_admin\_role$ или $admin\_roles\_admin\_role$ соответственно;

— при добавлении для некоторой роли или административной роли запрещающих ролей ни одна субъект-сессия не должна иметь к ней административных доступов;

— для получения субъект-сессией запрещающих ролей, заданных с помощью функции $constraint_{NR}$ для роли или административной роли, или, наоборот, ролей или административных ролей, для которых задана запрещающая роль, субъект-сессия должна иметь административный доступ на чтение к административной роли $negative\_roles\_admin\_role$.

У с л о в и е 3 (доступы и права доступа запрещающих ролей):

— к запрещающим ролям субъект-сессии могут иметь любые виды административных доступов из множества R_a;

— в случае, когда для некоторой роли или административной роли задано непустое множество запрещающих ролей, административный доступ на чтение субъект-сессии к такой роли или административной роли предоставляется только при получении ею административного доступа на чтение ко всем соответствующим запрещающим ролям: если существуют субъект-сессия $s \in S$, роль или административная роль $r \in R \cup AR$, запрещающая роль $nr \in NR$, такие, что $nr \in constraint_{NR}(r)$, $(s, r, read_a) \in AA$, то $(s, nr, read_a) \in AA$;

— запрещающие роли могут обладать к субъект-сессиям только правом доступа владения: для запрещающей роли $nr \in NR$ и субъект-сессии $s \in S$ если $(s, \alpha_r) \in \in PA(nr)$, то $\alpha_r = own_r$;

— запрещающие роли могут иметь к сущностям любые права доступа из множества R_r, только административные роли могут иметь к запрещающим ролям права доступа из множества R_r;

— для изменения прав доступа запрещающей роли субъект-сессии необходимо иметь к ней доступ на запись, за исключением случаев, когда удаляются сущности или субъект-сессии и соответственно удаляются имеющиеся к ним у запрещающих ролей права доступа.

У с л о в и е 4 (доступ к сущностям в иерархии сущностей):

— для получения субъект-сессией любого доступа к сущности, создания «жёсткой» ссылки на неё, получения или изменения параметров, прав доступа к ней, активизации из неё субъект-сессии требуется существование последовательности непосредственно вложенных друг в друга сущностей-контейнеров, начинающейся с некоторой сущности-«корневой контейнер» (например, корневой контейнер «/» в ОССН) и заканчивающейся сущностью-контейнером, в состав которой непосредственно входит сама сущность, и наличие у субъект-сессии текущих ролей или административных ролей, обладающих в совокупности правами доступа $execute_r$ ко всем сущностям-контейнерам этой последовательности, а также отсутствие у субъект-сессии запрещающей роли, обладающей правом доступа $execute_r$ хотя бы к одной сущности-контейнеру этой последовательности: для состояний системы G и G', правила преобразования состояний системы $op \in OP$, таких, что $G \vdash_{op} G'$, если $s \in S$, $e \in E$, $(s, e, \alpha_a) \notin A$ и $(s, e, \alpha_a) \in A'$, где $\alpha_a \in R_a$, то существует контейнер $c \in C$ и $execute\_container(s, c, e) = \texttt{true}$.

У с л о в и е 5 (создание, переименование, удаление сущности или «жёсткой» ссылки на неё, получение её параметров):

— при управлении доступом к сущности субъект-сессия не должна иметь текущую запрещающую роль, обладающую правом доступа владения к этой сущности;
— при создании, переименовании или удалении сущности или «жёсткой» ссылки на неё в сущности-контейнере субъект-сессия не должна иметь текущую запрещающую роль, обладающую правом доступа на выполнение к этой сущности-контейнеру;
— при изменении метки разделяемости сущности-контейнера в случае, когда субъект-сессия не имеет доступа на чтение к административной роли $entities\_admin\_role$, она не должна иметь текущую запрещающую роль, обладающую правом доступа владения к этой сущности-контейнеру;
— при получении субъект-сессией данных о ролях, запрещающих ролях или административных ролях, обладающих правами доступа к сущности, в случае, когда субъект-сессия не имеет доступа на чтение к административной роли $entities\_admin\_role$, она не должна иметь текущую запрещающую роль, обладающую правом доступа владения к этой сущности;
— при получении субъект-сессией данных о ролях или административных ролях, обладающих правом доступа владения к другой субъект-сессии или имеющихся у этой субъект-сессии текущих ролях или административных ролях, в случае, когда субъект-сессия не имеет доступа на чтение к административной роли $subjects\_admin\_role$, она не должна иметь текущую запрещающую роль, обладающую правом доступа владения ко второй субъект-сессии.

У с л о в и е 6 (создание и удаление субъект-сессии):

— при активизации субъект-сессией новой субъект-сессии из сущности первая субъект-сессия не должна иметь текущую запрещающую роль, обладающую правом доступа на выполнение к этой сущности;
— при удалении субъект-сессией субъект-сессии первая субъект-сессия не должна иметь текущую запрещающую роль, обладающую правом доступа владения ко второй субъект-сессии.

У с л о в и е 7 (вид метки):

— переопределяется функция $direct : E \cup R \cup NR \cup AR \to \{\texttt{true}, \texttt{false}\}$ и для каждой запрещающей роли вид её метки задаётся прямой: для $nr \in NR$ верно $direct(nr) = \texttt{true}$;

— для каждой сущности с косвенной меткой существует единственная старшая её в иерархии сущность-контейнер с прямой меткой, для которой все сущности, находящиеся ниже её в иерархии, имеют косвенные метки, а права доступа всех запрещающих ролей к ним равны правам доступа к этой сущности-контейнеру: для каждой сущности $e \in E$, такой, что $direct(e) = \texttt{false}$, существует единственная сущность-контейнер $c \in C$, такая, что $direct(c) = \texttt{true}$, $e < C$ и для любой сущности $e' \in E$, такой, что $e' < c$, верно $direct(e') = \texttt{false}$ и выполняется $(e', \alpha_r) \in PA(nr)$ тогда и только тогда, когда $(c, \alpha_r) \in PA(nr)$, где $nr \in NR$.

У с л о в и е 8 (доступы субъект-сессий к запрещающим ролям, соответствующим индивидуальным административным ролям, индивидуальным ролям или общим ролям):

— если для индивидуальной административной роли или индивидуальной роли учётной записи пользователя заданы запрещающие роли, то индивидуальная административная роль должна обладать административными правами доступа на чтение ко всем таким запрещающим ролям и при создании от имени этой учётной записи пользователя субъект-сессии она получает доступ на чтение ко всем соответствующим запрещающим ролям: если для учётной записи пользователя $u \in U$ выполняется $nr \in constraint_{NR}(u\_admin) \cup \cup constraint_{NR}(u\_c)$, то $(nr, read_r) \in APA(u\_admin)$; если для субъект-сессии $s \in S$ верно $nr \in constraint_{NR}(user(s)\_admin) \cup constraint_{NR}(user(s)\_c)$, то $(s, nr, read_a) \in \in AA$;

— если для общей роли заданы запрещающие роли, то индивидуальная административная роль каждой учётной записи пользователя должна обладать административными правами доступа на чтение ко всем таким запрещающим ролям и при создании любой субъект-сессии она получает доступ на чтение ко всем соответствующим запрещающим ролям: если выполняется $nr \in constraint_{NR}(common\_role)$, то для всех учётных записей пользователей $u \in U$ верно $(nr, read_r) \in APA(u\_admin)$ и для всех субъект-сессий $s \in S$ верно $(s, nr, read_a) \in AA$;

— при удалении субъект-сессии удаляются все её административные доступы к запрещающим ролям.

Рассмотрим особенности реализации в реальной защищённой ОССН условий сформулированного предположения.

Условие 1 содержит требования к запрещающим ролям и их иерархии, аналогичные требованиям к «обычным» ролям. Для создания, удаления, переименования, управления правами доступа запрещающих ролей, получения их параметров задаётся специальная административная роль $negative\_roles\_admin\_role$.

Условие 2 определяет порядок администрирования функции необходимого обладания запрещающей ролью $constraint_{NR}$, а также получения значений этой функции. При этом указывается на отсутствие запрещающих ролей, соответствующих специальным административным ролям, так как обратное едва ли потребуется в ОССН и может сильно затруднить её администрирование.

В условии 3 основным является описание порядка использования функции необходимого обладания запрещающей ролью $constraint_{NR}$, заключающегося в обязатель-

ном получении субъект-сессией запрещающей роли как текущей в случае, когда эта субъект-сессия получает как текущую соответствующую роль или административную роль.

В условии 4 дополнительно при переопределении функции доступа к сущностям в иерархии сущностей $execute\_container$ учитывается требование отсутствия даже одной текущей запрещающей роли, обладающей правом доступа на выполнение к сущности-контейнеру, через которую соответствующая субъект-сессия делает попытку получить какой-либо доступ к сущности.

Условия 5 и 6 раскрывают содержание ограничений, которые накладываются правами доступа («запрещающими» правами доступа) запрещающих ролей при попытке осуществления субъект-сессиями соответствующих доступов к сущностям, а также при создании или удалении субъект-сессий. При этом учитывается, что наличие запрещающих ролей не накладывает ограничений на использование специальных административных ролей $entities\_admin\_role$ и $subjects\_admin\_role$.

В условии 7 по аналогии с ролями и административными ролями определяется порядок задания прав доступа запрещающих ролей к сущностям, обладающим косвенной меткой.

Выполнение условия 8 обеспечивает возможность получения субъект-сессией заданной запрещающей роли как текущей в случае, если эта роль соответствует индивидуальной роли или индивидуальной административной роли учётной записи пользователя этой субъект-сессии. Правом доступа на чтение к такой запрещающей роли должна обладать индивидуальная административная роль учётной записи пользователя. В этом случае при создании субъект-сессии она сразу получает соответствующую запрещающую роль. Аналогично в случае, когда запрещающая роль задана для общей роли, то правом доступа на чтение к запрещающей роли должны обладать все учётные записи пользователей. Таким образом, условие обеспечивает согласованность задания значений функции необходимого обладания запрещающей ролью $constraint_{NR}$ с правами доступа индивидуальных административных ролей учётных записей пользователей. Кроме того, условие не указывает, какой административной роли должно быть дано право доступа на чтение к запрещающей роли, заданной для какой-то роли или административной роли. Это означает, что для того, чтобы получить эту роль или административную роль как текущую, субъект-сессии необходимо иметь право доступа на чтение к запрещающей роли через индивидуальную административную роль учётной записи пользователя субъект-сессии или через любую другую административную роль, которую как текущую должна предварительно получить субъект сессия. В противном случае субъект-сессия не сможет получить роль или административную роль как текущую.

В условии 8 описан способ задания запрещающей роли для индивидуальной административной роли или индивидуальной роли некоторой учётной записи пользователя, делающий такую запрещающую роль текущей для всех субъект-сессий, функционирующих от имени этой учётной записи пользователя. Возможны другие более «гибкие» способы применения запрещающих ролей. Например, в некоторых случаях в ОССН требуется ограничить права доступа субъект-сессии (процесса), активизированного из конкретной сущности (исполняемого файла). Тогда роли, которая обладает правом доступа на выполнение $execute_r$ к исполняемому файлу, можно с помощью функции $constraint_{NR}$ назначить соответствующую запрещающую роль. После этого родительский процесс должен сначала получить как текущую эту роль вместе с запрещающей

ролью и активизировать из исполняемого файла новый процесс, который «наследует» от родительского и запрещающую роль.

3. Де-юре правила преобразования состояний системы и их корректность

С учётом добавления в модель по сравнению с базовым уровнем запрещающих ролей модифицируются её де-юре правила преобразования состояний. В таблице приведены примеры таких правил, при этом для каждого правила в первой строке указаны его параметры, условия и результаты применения, наследованные с базового уровня без изменений, а во второй строке — внесённые в них дополнения и изменения, соответствующие уровню запрещающих ролей МРОСЛ ДП-модели.

Таблица

Примеры де-юре правил преобразования состояний уровня запрещающих ролей МРОСЛ ДП-модели

Параметры	Исходное состояние G	Результирующее состояние G'
$create\_user(x, u)$		
x, u	$x \in S$, $u \notin U$, $(x, users\_admin\_role, read_a) \in AA$, $(x, roles\_admin\_role, \alpha_a) \in AA$, $(x, admin\_roles\_admin\_role, \alpha_a) \in AA$, где $\alpha_a \in \{read_a, write_a\}$	$U' = U \cup \{u\}$, $AR' = AR \cup \{u\_admin\}$, $PA'(u\_admin) = \varnothing$, $direct'(u\_admin) =$ $= shared\_container'(u\_admin) =$ `true`, $role\_name'(u\_admin) =$ "u_admin", $H'_R(u\_admin) = \varnothing$, $R' = R \cup \{u\_c\}$, $PA'(u\_c) = \varnothing$, $direct'(u\_c) =$ $= shared\_container'(u\_c) =$ `true`, $role\_name'(u\_c) =$ "u_c", $H'_R(u\_c) = \varnothing$, $APA'(admin\_roles\_admin\_role) =$ $= APA(admin\_roles\_admin\_role) \cup$ $\cup \{(u\_admin, own_r)\}$, $APA'(roles\_admin\_role) =$ $= APA(roles\_admin\_role) \cup \{(u\_c, own_r)\}$, для $ar \in AR$ выполняется $APA'(ar) = APA(ar) \cup$ $\cup \{(u\_admin, execute_r), (u\_c, execute_r)\}$
—	—	$APA'(u\_admin) = \{(u\_admin, \alpha_r) :$ $\alpha_r \in \{read_r, write_r, execute_r\}\} \cup$ $\cup \{(u\_c, \alpha_r), (common\_role, \alpha_r) :$ $\alpha_r \in \{read_r, write_r, execute_r\}\} \cup$ $\cup \{(r, execute_r) : r \in R' \cup NR \cup AR'\} \cup$ $\cup \{(nr, read_r) :$ $nr \in constraint_{NR}(common\_role)\}\}$
$access\_write(x, y)$		
x, y	$x \in S$, существует $r \in R \cup AR :$ $(x, r, read_a) \in AA$, [если $y \in E$, то $(y, write_r) \in PA(r)$, существует контейнер $c \in C$, такой, что $execute\_container(x, c, y) =$ `true`], [если $y \in R \cup AR$, то $(y, write_r) \in APA(r)$]	если $y \in E$, то $A' = A \cup \{(x, y, write_a)\}$, $AA' = AA$, если $y \in R \cup AR$, то $AA' = AA \cup \{(x, y, write_a)\}$, $A' = A$
—	$y \in E \cup R \cup NR \cup AR$, [если $y \in E$, то не существует запрещающей роли $nr \in NR$, такой, что $(x, nr, read_a) \in AA$ и $(y, write_r) \in PA(nr)$], [если $y \in NR$, то $(y, write_r) \in APA(r)$]	—

Продолжение таблицы

Параметры	Исходное состояние G	Результирующее состояние G'
$add\_negative\_owner(x, nr, y)$		
—	—	—
x, nr, y	$x \in S$, $y \in E \cup S$, $(x, nr, write_a) \in AA$, [существует $r' \in R \cup AR$: $(x, r', read_a) \in AA$, $(y, own_r) \in PA(r')$], [не существует запрещающей роли $nr' \in NR$, такой, что $(x, nr', read_a) \in \in AA$ и $(y, own_r) \in PA(nr')$], [если $y \in E$, то $direct(y) =$ `true`, существует контейнер $c \in C$, такой, что $execute\_container(x, c, y) =$ `true`]	$PA'(nr) = PA(nr) \cup \{(y, own_r)\}$, если $y \in E$ и $H_E(y) = \{y' \in H_E(y) :$ $direct(y') =$ `false`$\}$, то для всех $y' < y$ верно $PA'(r) = PA(r) \cup \{(y', own_r)\}$
$add\_negative\_role(x, r, nr)$		
—	—	—
x, r, nr	$x \in S$, $r \in (R \cup AR) \setminus SAR$, $nr \in NR$, [не существует $s \in S$, такого, что $(s, r, read_a) \in AA$], $(x, negative\_roles\_admin\_role, read_a) \in AA$, [если $r \in R$, то $(x, roles\_admin\_role, read_a) \in AA$, если $r \in AR$, то $(x, admin\_roles\_admin\_role, read_a) \in AA$], [если существует $u \in U$, такая, что $r \in \{u\_admin, u\_c\}$, то $(nr, read_r) \in APA(u\_admin)$], [если $r = common\_role$, то для всех $u \in U$ $((nr, read_r) \in APA(u\_admin))$]	$constraint'_{NR}(r) = constraint_{NR}(r) \cup \{nr\}$
$create\_hard\_link(x, y, name, z)$		
$x, y, name, z$	$x \in S$, $y \in O$, $z \in C$, [существует $c \in C$: $execute\_container(x, c, y) =$ `true`], [$(x, z, write_a) \in A$, существует $r \in R \cup AR$, такая, что $(x, r, read_a) \in \in AA$ и $(z, execute_r) \in PA(r)$], $name \in \in NAME \setminus (\{\text{“”}\} \cup \{entity\_name(z, y'):$ $y' \in H_E(z)\})$, $H_E(z) = \{y' \in H_E(z):$ $direct(y') = direct(y)\}$, [если $direct(y) =$ `true`, то $direct(z) =$ `true`], [если $direct(y) =$ `false`, то существует $z' \in C$, такая, что $direct(z) =$ `true`, $y < z'$, $z \leqslant z'$ и для всех $y' < z'$ верно $direct(y') =$ `false`]	$entity\_name'(z, y) = entity\_name(z, y) \cup$ $\cup \{name\}$, $H'_E(z) = H_E(z) \cup \{y\}$
—	не существует запрещающей роли $nr \in \in NR$, такой, что $(x, nr, read_a) \in AA$ и $(z, execute_r) \in PA(nr)$	—
$rename\_role(x, ry, name)$		
$x, ry, name$	$x \in S$, $name \in NAME \setminus (\{\text{“”}\} \cup$ $\cup \{role\_name(rz, ry') : ry', rz \in R \cup AR,$ $ry' \in H_R(rz)\})$, [если $ry \in R$, то $(x, roles\_admin\_role, read_a) \in AA$, если $ry \in AR$, то $(x,$ $admin\_roles\_admin\_role, read_a) \in AA$]	для $rz \in R \cup AR$, таких, что $ry \in H_R(rz)$, выполняется $role\_name'(rz, ry) = name$
—	$ry \in (R \cup NR \cup AR) \setminus (U\_ADMIN \cup$ $\cup U\_ROLES \cup COMMON\_ROLES \cup$ $\cup SAR)$, $name \notin \{role\_name(nr', nr'') :$ $nr', nr'' \in NR\}$, [если $ry \in NR$, то $(x,$ $negative\_roles\_admin\_role, read_a) \in AA$], [для $rz \in R \cup NR \cup AR : ry \in H_R(rz)$, выполняется $(x, rz, write_a) \in AA$]	для $rz \in NR$, таких, что $ry \in H_R(rz)$, выполняется $role\_name'(rz, ry) = name$

Окончание таблицы

Параметры	Исходное состояние G	Результирующее состояние G'
$create\_first\_session(x, u, y, z)$		
x, u, y, z	$x \in S$, $u \in U$, $y \in E$, $z \notin S$, существует $r \in R \cup AR$, такая, что $(x, r, read_a) \in$ $\in AA$ и $(y, execute_r) \in PA(r)$, существует контейнер $c \in C$, такой, что $execute\_container(x, c, y) =$ `true`	$S' = S \cup \{z\}$, $user'(z) = u$, $H'_S(z) = \varnothing$, $PA'(u\_c) = PA(u\_c) \cup \{(z, own_r)\}$
—	[не существует запрещающей роли $nr \in NR$: $(x, nr, read_a) \in AA$ и $(y, execute_r) \in PA(nr)$], [для всех $nr \in constraint_{NR}(u\_admin) \cup$ $\cup\, constraint_{NR}(u\_c) \cup$ $\cup\, constraint_{NR}(common\_role)$ верно $(nr, read_r) \in APA(u\_admin)$]	$AA' = AA \cup \{(z, u\_admin, read_a)$, $(z, u\_c, write_a)$, $(z, common\_role, write_a)$, $(z, u\_c, read_a)$, $(z, common\_role, read_a)\} \cup$ $\cup \{(z, nr, read_a)$: $nr \in constraint_{NR}(u\_admin) \cup$ $\cup\, constraint_{NR}(u\_c) \cup$ $\cup\, constraint_{NR}(common\_role)\}$

Всего в рамках уровня запрещающих ролей иерархического представления МРОСЛ ДП-модели задано 35 де-юре правил преобразования состояний, из которых четыре правила вида $add\_negative\_owner(x, nr, y)$, $remove\_negative\_owner(x, nr, y)$, $add\_negative\_role(x, r, nr)$ и $remove\_negative\_role(x, r, nr)$ отсутствовали на базовом уровне, так как предназначены для добавления или удаления запрещающим ролям права доступа владения к сущностям или субъект-сессиям, а также для управления множествами запрещающих ролей и административных ролей, задаваемых функцией $constraint_{NR}$. При этом так же, как на базовом уровне, на текущем уровне не анализируются информационные потоки, поэтому на нём также не задаются де-факто правила преобразования состояний.

Рассмотрим приведённые в таблице примеры де-юре правил преобразования состояний.

В де-юре правило вида $create\_user(x, u)$ добавляется только требование, чтобы индивидуальной административной роли создаваемой учётной записи пользователя были даны административные права доступа на чтение ко всем запрещающих ролям, заданным для общей роли.

В де-юре правило вида $access\_write(x, y)$ включены дополнительные условия, позволяющие субъект-сессиям получать доступ на запись к запрещающим ролям. Кроме того, в этом правиле, как во многих других де-юре правилах, используется переопределённая функция доступа субъект-сессии к сущностям в контейнерах $execute\_container$, учитывающая необходимость проверки отсутствия у субъект-сессии x запрещающих ролей, обладающих правами доступа на выполнение к соответствующим сущностям-контейнерам, в которых содержится сущность y. Требуется также отсутствие у субъект-сессии x текущей запрещающей роли, обладающей правом доступа на запись.

Де-юре правило вида $add\_negative\_owner(x, nr, y)$ является новым и во многом аналогичным правилам вида $grant\_rights(x, r, \{(y, \alpha_{rj}) : 1 \leqslant j \leqslant k\})$, $set\_entity\_owner(x, r, r', y)$ и $set\_subject\_owner(x, r, r', y)$. Оно позволяет субъект-сессии x, обладающей ролью или административной ролью, имеющей право доступа владения к сущности или субъект-сессии y, а также не обладающей запрещающей ролью, имеющей это право доступа к y, добавить запрещающей роли nr (к которой субъект-сессия x должна иметь административный доступ на запись) право доступа владения к y. При этом запрещающая роль не становится ролью-«владельцем» или ролью-«антивладельцем» сущности или субъект-сессии y, допускается, что запрещаю-

щих ролей, обладающих правом доступа владения к сущности или субъект-сессии, может быть несколько. С этим связано добавление этого правила на рассматриваемом уровне модели и неиспользование для этого правил вида $set\_entity\_owner(x, r, r', y)$ и $set\_subject\_owner(x, r, r', y)$.

Де-юре правило вида $add\_negative\_role(x, r, nr)$ также является новым и позволяет субъект-сессии x добавить для роли или административной роли r запрещающую роль nr. Для этого субъект-сессия x должна обладать текущей специальной административной ролью $negative\_roles\_admin\_role$, а также соответственно либо специальной административной ролью $roles\_admin\_role$, либо $admin\_roles\_admin\_role$. При этом нельзя задавать запрещающие роли для любых специальных административных ролей из множества SAR и для упрощения администрирования системы ни одна субъект-сессия не должна иметь к роли или административной роли r административных доступов на чтение.

В де-юре правиле вида $create\_hard\_link(x, y, name, z)$ добавлена проверка отсутствия у субъект-сессии x текущей запрещающей роли, обладающей правом доступа на выполнение к сущности-контейнеру z, в которой создаётся «жёсткая» ссылка на сущность y.

Переименование запрещающих ролей осуществляется аналогично «обычным» ролям, поэтому дополнительным условием де-юре правила вида $rename\_role(x, ry, name)$ является наличие у субъект-сессии x административного доступа на чтение к специальной административной роли $negative\_roles\_admin\_role$. Требуется также исключение совпадения нового имени $name$ с именем какой-либо запрещающей роли.

В де-юре правиле вида $create\_first\_session(x, u, y, z)$ дополнительно требуется отсутствие у субъект-сессии x текущей запрещающей роли, обладающей правом доступа на выполнение к сущности y, из которой активизируется новая субъект-сессия. Кроме того, чтобы создаваемая субъект-сессия z могла получить в качестве текущих все запрещающие роли для индивидуальной административной роли, индивидуальной роли её учётной записи пользователя и для общей роли, проверяется наличие права доступа на чтение ко всем таким запрещающим ролям у индивидуальной административной роли учётной записи пользователя субъект-сессии z. После применения правила созданная субъект-сессия z получает административный доступ на чтение ко всем этим запрещающим ролям.

На уровне запрещающих ролей МРОСЛ ДП-модели верно утверждение о том, что условия и результаты правил преобразования состояний системы заданы корректно, т. е. соответствуют предположениям базового и текущего (предположение 1) уровней модели.

Утверждение 1. Пусть G_0 — начальное состояние системы $\Sigma(G^*, OP, G_0)$, удовлетворяющее условиям предположений, задающих порядок реализации ролевого управления доступом на базовом уровне и уровне запрещающих ролей (предположение 1) МРОСЛ ДП-модели. Тогда на любой траектории $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$, где $N \geqslant 1$, состояние G_N и переход $G_{N-1} \vdash_{op_N} G_N$ удовлетворяют условиям этих предположений.

Заключение

Таким образом, в МРОСЛ ДП-модель добавлены запрещающие роли, применение которых в ОССН позволит в перспективе упростить настройку и администрирование реализуемого в ней ролевого управления доступом. Кроме того, вместе с уровнем мандатного контроля целостности с невырожденной решёткой уровней целостности [10]

текущий уровень модели стал основой для её дальнейшего развития и построения нескольких новых уровней МРОСЛ ДП-модели, включающих также мандатное управление доступом с контролем информационных потоков по памяти и по времени.

ЛИТЕРАТУРА

1. *Sandhu R.* Role-Based Access Control. Advanced in Computers. Academic Press, 1998. V. 46.
2. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками: учеб. пособие для вузов. 2-е изд., испр. и доп. М.: Горячая линия — Телеком, 2013. 338 с.
3. *Буренин П. В., Девянин П. Н., Лебеденко Е. В. и др.* Безопасность операционной системы специального назначения Astra Linux Special Edition: учеб. пособие для вузов / под ред. П. Н. Девянина. 2-е изд., стереотип. М.: Горячая линия — Телеком, 2016. 312 с.
4. *Девянин П. Н.* О результатах формирования иерархического представления МРОСЛ ДП-модели // Прикладная дискретная математика. Приложение. 2016. № 9. С. 83–87.
5. `www.astralinux.com` — Операционные системы Astra Linux. 2017.
6. `ru.wikipedia.org/wiki/Astra_Linux` — Astra Linux. 2017.
7. *Тележников В. Ю.* Правила преобразования состояний системы в рамках ДП-модели управления доступом в компьютерных сетях, построенных на основе ОС семейства Linux // Прикладная дискретная математика. 2016. № 1(31). С. 67–85.
8. *Armando A. and Ranise S.* Automated and efficient analysis of role-based access control with attributes // LNCS. 2012. V. 7371. P. 25–40.
9. *Kuijper W. and Ermolaev V.* Sorting out role based access control // Proc. SACMAT'14. N. Y.: ACM, 2014. P. 63–74.
10. *Девянин П. Н.* Реализация невырожденной решётки уровней целостности в рамках иерархического представления МРОСЛ ДП-модели // Прикладная дискретная математика. Приложение. 2017. № 10. С. 111–114.

REFERENCES

1. *Sandhu R.* Role-Based Access Control. Advanced in Computers, Academic Press, 1998, vol. 46.
2. *Devyanin P. N.* Modeli bezopasnosti kompyuternyh sistem. Upravlenie dostupom i informacionnymi potokami [Security Models of Computer Systems. Access Control and Information Flows]. Moscow, Goryachaya liniya — Telekom, 2013. 338 p. (in Russian)
3. *Burenin P. V., Devyanin P. N., Lebedenko E. V., et al.* Bezopasnost' operacionnoy sistemy special'nogo naznacheniya Astra Linux Special Edition [Security of Operating System Astra Linux Special Edition]. Moscow, Goryachaya liniya — Telekom, 2016. 312 p. (in Russian)
4. *Devyanin P. N.* O rezultatah formirovaniya ierarhiceskogo predstavleniya MROSL DP-modeli [About results of design hierarchical representation of MROSL DP-model]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2016, no. 9, pp. 83–87. (in Russian)
5. `www.astralinux.com` — Operating System Astra Linux, 2017.
6. `ru.wikipedia.org/wiki/Astra_Linux` — Astra Linux, 2017.
7. *Telezhnikov V. Y.* Pravila preobrazovaniya sostoyaniy sistemy v ramkah DP-modeli upravleniya dostupom v komp'yuternyh setyah, postroennyh na osnove OS semeystva Linux [System state transformation rules in DP-model of access control in computer networks based on operating systems of Linux]. Prikladnaya Diskretnaya Matematika, 2016, no. 1(31), pp. 67–85. (in Russian)
8. *Armando A. and Ranise S.* Automated and efficient analysis of role-based access control with attributes. LNCS, 2012, vol. 7371, pp. 25–40.

9. *Kuijper W. and Ermolaev V.* Sorting out role based access control. Proc. SACMAT'14, New York, ACM, 2014, pp. 63–74.
10. *Devyanin P. N.* Realizaciya nevyrozhdennoy reshyotki urovney celostnosti v ramkah ierarhiceskogo predstavleniya MROSL DP-modeli [Implementation of a non-degenerate lattice integrity levels within the hierarchical representation of MROSL DP-model]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2017, no. 10, pp. 111–114. (in Russian)

ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ

УДК 519.725

СПИСОЧНОЕ ДЕКОДИРОВАНИЕ БИОРТОГОНАЛЬНЫХ ВЕЙВЛЕТ-КОДОВ С ЗАДАННЫМ КОДОВЫМ РАССТОЯНИЕМ В ПОЛЕ НЕЧЁТНОЙ ХАРАКТЕРИСТИКИ

Д. В. Литичевский

Челябинский государственный университет, г. Челябинск, Россия

Представлено теоретическое обоснование возможности списочного декодирования для биортогональных вейвлет-кодов $W[n, n/2, d]$ с заданным кодовым расстоянием над полями нечётной характеристики. Для входного сообщения длины n задача списочного декодирования заключается в нахождении всех кодовых слов, расстояние Хэмминга до которых не превосходит заданного значения. Для кода $W[n, n/2, d]$ эта задача сводится к задаче списочного декодирования для кода Рида — Соломона $\mathrm{RS}[n, n-d+1]$ посредством преобразования входящего сообщения и последующего применения к его результатам улучшенной версии алгоритма Гурусвами — Судана. Результаты декодирования для кода $W[n, n/2, d]$ находятся путём решения системы линейных уравнений относительно коэффициентов информационного многочлена, полученной из преобразования Фурье кодового слова вейвлет-кода, для каждого найденного информационного слова кода $\mathrm{RS}[n, n-d+1]$, являющегося в решаемой системе столбцом свободных членов. Приведены примеры результатов списочного декодирования для кода $W[26, 13, 12]$, на которых длина результирующего списка равна 2.

Ключевые слова: *вейвлет-коды, коды с заданным кодовым расстоянием, декодирование списком.*

DOI 10.17223/20710410/39/6

LIST DECODING OF THE BIORTHOGONAL WAVELET CODE WITH PREDETERMINED CODE DISTANCE ON A FIELD OF ODD CHARACTERISTIC

D. V. Litichevskiy

Chelyabinsk State University, Chelyabinsk, Russia

E-mail: litichevskiydv@gmail.com

In the article, a list decoding algorithm for the biorthogonal wavelet codes $W[n, n/2, d]$ with a predetermined code distance on a field of odd characteristic is presented. The "list decoding" problem the algorithm solves is the following: given an input message of the length n, compute all the codewords the Hamming distance to which does not exceed the given value. The list decoding algorithm for the code $W[n, n/2, d]$ is based on the transformation of the list decoding problem for $W[n, n/2, d]$ to the list decoding problem for the Reed — Solomon code $\mathrm{RS}[n, n-d+1]$ by proper converting

the incoming messages and on the subsequent solution of the second problem by the improved Guruswami — Sudan algorithm. Decoding results for the code $W[n, n/2, d]$ are found by solving a system of linear equations with respect to the coefficients of the information polynomial. The system is obtained from the Fourier transform of the code word of the wavelet code for each found information word of the code $\mathrm{RS}[n, n-d+1]$. In the system, the symbols of this word are constant terms. Examples of the list decoding for the code $W[26, 13, 12]$ are given. The algorithm has been implemented in the form of a computer program for which an author's certificate has been received.

Keywords: *wavelet codes, code with predetermined code distance, list decoding.*

Введение

В классической задаче декодирования требуется, чтобы исправление ошибки в принятом кодовом слове осуществлялось однозначно. В задаче декодирования списком допускается на выходе декодера иметь до L вариантов декодирования при некотором фиксированном L.

Будем говорить, что код допускает декодирование списком длины L с исправлением e ошибок, если любой шар радиуса e содержит не более L кодовых слов. Для кодов Рида — Соломона задача полиномиального декодирования списком решена В. Гурусвами и М. Суданом и [1].

В настоящей работе рассматривается возможность использования алгоритма списочного декодирования Гурусвами — Судана для построенных ранее вейвлет-кодов с заданным кодовым расстоянием [2].

1. Алгоритм списочного декодирования

Введём следующие обозначения: $\mathrm{GF}(q)$ — конечное поле из q элементов; n — длина кодовых слов; k — длина информационных слов; e — число возможных ошибок; d $(d \leqslant n-k+1)$ — кодовое расстояние (равное $n-k+1$ для кодов Рида — Соломона).

Информационному слову $v_0, v_1, \ldots, v_{k-1}$ процедура кодирования кода Рида — Соломона ставит в соответствие набор значений многочлена $v(x) = \sum\limits_{j=0}^{k-1} v_j x^j$ в n точках поля. Пусть $x_1, x_2, \ldots, x_n$ ($x_i \neq x_j$ при $i \neq j$) и $y_1, y_2, \ldots, y_n$ — произвольные элементы поля $\mathrm{GF}(q)$. Построение списка кодовых слов алгоритм декодирования осуществляет в два этапа: интерполяционный и факторизационный.

На интерполяционном этапе выполняется поиск многочлена $Q(x, y) \in \mathrm{GF}(q)[x, y]$, такого, что $Q(x, y)$ обращается в нуль во всех точках (x_i, y_i), $i = 1, \ldots, n$.

Для натурального числа $a > \sqrt{2(k-1)n}$ доказано, что если многочлен $p(x)$ имеет степень меньше k и $p(x_i) = y_i$ в s точках (x_i, y_i), причём $s > a$, то $Q(x, p(x)) = 0$ и многочлен $Q(x, y)$ делится на $y - p(x)$. Число делителей вида $y - p(x)$ у многочлена $Q(x, y)$ меньше числа $a/(k-1)$. Неизвестными являются коэффициенты $a_{u\nu}$ многочлена

$$Q(x, y) = \sum_{u+(k-1)\nu<a} a_{u\nu} x^u y^\nu,$$

которые находятся из системы линейных уравнений

$$\sum_{u+(k-1)\nu<a} a_{u\nu} x_i^u y_i^\nu = 0, \ i = 1, \ldots, n,$$

при небольших значениях n, например методом Гаусса.

На факторизационном шаге выполняется поиск всех многочленов $p(x) \in \mathrm{GF}(q)[x]$, таких, что $y - p(x)$ делит $Q(x, y)$ и степень $p(x)$ меньше k. Найденные многочлены формируют искомый список кодовых слов. Эффективный алгоритм факторизации предложен в [3]. Подробное изложение алгоритма приводится в [4].

2. Схема полифазного кодирования

В поле $\mathrm{GF}(q)$, где $q = p^m$, m — натуральное и $p \neq 2$ — простое число, введём биортогональные вейвлет-коды длины n, $n = q - 1$, с информационными словами длины $n/2$.

Пусть

$$H = \mathrm{cir}_2(h_0, h_1, \ldots, h_{n-1}), \quad G = \mathrm{cir}_2(g_0, g_1, \ldots, g_{n-1}),$$
$$\widetilde{H} = \mathrm{cir}_2(\widetilde{h}_0, \widetilde{h}_1, \ldots, \widetilde{h}_{n-1}), \quad \widetilde{G} = \mathrm{cir}_2(\widetilde{g}_0, \widetilde{g}_1, \ldots, \widetilde{g}_{n-1})$$

— 2-циркулянтные матрицы, определяющие процедуры разложения и восстановления сигнала биортогонального кратномасштабного анализа и удовлетворяющие условию точного восстановления. 2-Циркулянтная матрица задаётся первой строкой, последующая строка получается из предыдущей циклическим сдвигом на две позиции вправо.

Последовательность $\{h_k\}_{k=0}^{n-1}$ будем называть фильтром, многочлен $h(x) = \sum\limits_{k=0}^{n-1} x^k$ — весовым многочленом. Представим $h(x)$ в виде суммы полифазных компонент

$$h(x) = h_e(x^2) + x h_o(x^2),$$

где $h_e(x) = \sum\limits_{k=0}^{n/2-1} h_{2k} x^k$ и $h_o(x) = \sum\limits_{k=0}^{n/2-1} h_{2k+1} x^k$. Аналогично поступим с весовыми многочленами $g(x)$, $\widetilde{h}(x)$, $\widetilde{g}(x)$, которые соответствуют фильтрам $\{g_k\}_{k=0}^{n-1}$, $\{\widetilde{h}_k\}_{k=0}^{n-1}$ и $\{\widetilde{g}_k\}_{k=0}^{n-1}$.

Введём полифазные матрицы для пар весовых функций $(h(x), g(x))$ и $(\widetilde{h}(x), \widetilde{g}(x))$:

$$P(x) = \begin{bmatrix} h_e(x) & g_e(x) \\ h_o(x) & g_o(x) \end{bmatrix} \quad \text{и} \quad \widetilde{P}(x) = \begin{bmatrix} \widetilde{h}_e(x) & \widetilde{g}_e(x) \\ \widetilde{h}_o(x) & \widetilde{g}_o(x) \end{bmatrix}.$$

Пара матриц $P(x)$ и $\widetilde{P}(x)$ удовлетворяет условию точного восстановления тогда и только тогда, когда

$$P(x)\, \widetilde{P}(x^{n/2-1}) = I_{2\times 2}$$

(см. [2, 5]). Многочлены $h(x)$ и $g(x)$ называются комплементарными, если полифазная матрица имеет единичный определитель. Для многочлена $h(x)$, $\deg h(x) \leqslant n - 1$, комплементарный многочлен можно построить с помощью алгоритма Евклида нахождения наибольшего общего делителя и операции лифтинга. Для многочлена $s(x)$, $\deg s(x) < n/2$, определим операцию лифтинга как умножение полифазной матрицы $P(x)$ на треугольную матрицу вида

$$\begin{bmatrix} 1 & s(x) \\ 0 & 1 \end{bmatrix}.$$

Из условия точного восстановления следует, что $P(x)^{-1} = \widetilde{P}(x^{n/2-1})$ и

$$g_o(x) = \widetilde{h}_e(x^{n/2-1}), \quad g_e(x) = -\widetilde{h}_o(x^{n/2-1}),$$
$$h_o(x) = -\widetilde{g}_e(x^{n/2-1}), \quad h_e(x) = -\widetilde{g}_o(x^{n/2-1}).$$

Процедура кодирования определяется с помощью полифазных составляющих

$$\begin{bmatrix} c_e(x) \\ c_o(x) \end{bmatrix} = \begin{bmatrix} h_e(x) & g_e(x) \\ h_o(x) & g_o(x) \end{bmatrix} \begin{bmatrix} v(x) \\ av(x) \end{bmatrix},$$

где $v(x) = \sum\limits_{j=0}^{n/2-1} v_j x^j$ — информационный многочлен, $a \in \mathrm{GF}(q)$. Кодовый многочлен $c(x)$ имеет вид

$$c(x) = c_e(x^2) + xc_o(x^2) = (h(x) + ax^2 g(x))v(x^2) \bmod (x^n - 1).$$

Построенный вейвлет-код является 2-циркулянтным. Многочлен $F(x) = h(x) + x^2 g(x)$ будем называть *порождающим*.

Информационное слово восстанавливается по кодовому слову с помощью матрицы

$$\begin{bmatrix} \widetilde{h}_e(x^{n/2-1}) & \widetilde{g}_e(x^{n/2-1}) \end{bmatrix}.$$

Выберем d так, чтобы $0 < d < n/2$. Будем считать, что многочлены $h(x)$ и $g(x)$ степени $\leqslant n-1$ построены так (см. [2]), что имеют место равенства

$$h(\alpha^j) + \alpha^{2j} g(\alpha^j) = 0 \ \text{ при } \ j = 0, \ldots, d.$$

Здесь α — примитивный элемент поля $\mathrm{GF}(q)$. Тогда преобразование Фурье кодового многочлена $c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$ запишется в виде

$$\big(\underbrace{0, \ldots, 0}_{d+1}, C_{d+1}, \ldots, C_{n-1}\big).$$

В [2] доказано, что для любого d, $0 < d < n/2$, среди построенных вейвлет-кодов над полем $\mathrm{GF}(q)$ найдётся код с заданным кодовым расстоянием $d+2$. Для $C(x) = C_{d+1}x^{d+1} + C_{d+2}x^{d+2} + \cdots + C_{n-1}x^{n-1}$ обратным преобразованием Фурье находятся коэффициенты кодового многочлена: $c_j = n^{-1}C(\alpha^{-j})$, $j = 0, \ldots, n-1$, где n^{-1} — элемент, обратный элементу $n \bmod p$ в поле $\mathrm{GF}(q)$.

Так как α^{-1}, как и α, — примитивный элемент поля $\mathrm{GF}(q)$, то обратное преобразование Фурье последовательности $\{C_i\}_{i=0}^{n-1}$ является кодовой последовательностью кода Рида — Соломона с параметрами $(n, n-d-1)$. Поэтому построенный 2-циркулярный вейвлет-код является подпространством кода Рида — Соломона.

Из результатов работы [1], в частности, следует, что длина списка вейвлет-кодов не превышает двух, чему соответствуют результаты проведённых численных экспериментов.

3. Декодирование списком вейвлет-кода

Пусть α — примитивный элемент поля $\mathrm{GF}(q)$. Обозначим через $W[n, n/2, d+2]$ вейвлет-код с заданным кодовым расстоянием $d+2$, $0 < d < n/2$. Обозначим через $v(x)$ информационный многочлен и $c(x)$ — кодовый многочлен вида

$$c(x) = v(x^2)F(x) \bmod (x^n - 1), \tag{1}$$

где $F(x)$ — порождающий многочлен кода. Согласно свойствам порождающего многочлена $F(x)$, последовательность

$$F(\alpha^0), F(\alpha^1), \ldots, F(\alpha^{n-1})$$

имеет непрерывный участок длины $d+1$, состоящий из нулевых элементов. Тогда спектральный многочлен

$$C(y) = \sum_{j=0}^{n-1} C_j y^j,$$

где $C_j = c(\alpha^j)$, $j = 0, \ldots, n-1$, — коэффициенты преобразования Фурье для кодового слова, имеет не более $n-d-1$ ненулевых коэффициентов и может быть представлен в виде

$$C(y) = y^{j^*} \sum_{j=0}^{n-d-2} C_{j+j^*} y^j,$$

где j^* — номер первого ненулевого C_j. Тогда представим $c_i = n^{-1}C(\alpha^{-i})$ в виде

$$c_i = n^{-1}C(\alpha^{-i}) = \alpha^{-ij^*} n^{-1} \sum_{j=0}^{n-d-2} C_{j+j^*} \alpha^{-ij},\ i = 0, \ldots, n-1,$$

или

$$\sum_{j=0}^{n-d-2} C_{j+j^*} \alpha^{-ij} = c_i n \alpha^{ij^*}. \tag{2}$$

Полученное в (2) преобразование соответствует коду Рида — Соломона $\mathrm{RS}(n, n-d-1)$, в котором кодовое слово s получается из информационного слова β по формулам

$$s_i = \sum_{j=0}^{n-d-2} \beta_j \alpha^{-ij},\ i = 0, \ldots, n-1, \tag{3}$$

что допустимо, поскольку α^{-1} является примитивным элементом поля $\mathrm{GF}(q)$. Таким образом, для получения списка возможных спектральных многочленов $C(y)$ можем воспользоваться декодером Гурусвами — Судана, применённым к коду $\mathrm{RS}(n, n-d-1)$ с процедурой кодирования (3), на вход которого подаётся кодовое слово $s_i = c_i n \alpha^{ij^*}$, $i = 0, \ldots, n-1$.

Согласно процедуре кодирования (1) вейвлет-кода $W[n, n/2, d+2]$, ненулевые значения C_j, $j = j^*, \ldots, j^* + n - d - 2$, могут быть найдены как

$$v(\alpha^{2j}) F(\alpha^j) = C_j.$$

Значит, должны выполняться соотношения

$$v(\alpha^{2(j+j^*)}) F(\alpha^{j+j^*}) = \beta_j,\ j = 0, \ldots, n-d-2. \tag{4}$$

Равенства (4) задают систему линейных уравнений относительно коэффициентов информационного многочлена вейвлет-кода, содержащую $n/2$ неизвестных и $n-d-1 > n/2$ уравнений. Поэтому рассматриваемый вейвлет-код $W[n, n/2, d+2]$ является подпространством кода $\mathrm{RS}[n, n-d-1]$ и не всякому информационному слову β из списка, возвращённого декодером Гурусвами — Судана, будет соответствовать информационное слово v вейвлет-кода, которое находится из системы (4).

4. Результаты численных экспериментов

Описанный подход проверялся в эксперименте с вейвлет-кодом $W[26, 13, 12]$ и соответствующим ему кодом $\mathrm{RS}[26, 15]$ над полем $\mathrm{GF}(27)$ с неприводимым многочленом $2 + 2x + x^3$ и порождающим элементом 2α, где α — корень выбранного неприводимого многочлена. В качестве фильтра $h(x)$ был произвольным образом выбран многочлен

$h(x) = 22 + x^3 + 2x^4 + 3x^5 + 4x^6 + x^7 + 6x^8 + 7x^9 + 8x^{10} + 9x^{11} + x^{12} + 10x^{13} + x^{14} + 12x^{15} + + x^{16} + 14x^{17} + x^{18} + 17x^{19} + x^{20} + 19x^{21} + 20x^{22} + x^{23} + x^{24} + x^{25}$, на основании которого был построен порождающий многочлен кода $W[26, 13, 12]$.

Для эксперимента рассматривалось нулевое информационное слово. В соответствующее ему нулевое кодовое слово вводились 6 случайных ошибок, правильными считались 20 принятых значений. В большинстве случаев список во временной области содержит единственное кодовое слово, например, для зашумленных слов

$$(1, 1, 8, 14, 15, 16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 1, 0, 0, 0, 1, 0, 8, 0, 0, 0, 0, 0, 0, 14, 2, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0)$$

список содержит только нулевое кодовое слово. Вместе с тем для зашумленного кодового слова $(0, 0, 1, 0, 0, 0, 1, 0, 12, 0, 0, 0, 0, 0, 0, 0, 12, 4, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0)$ список во временной области содержит два информационных слова:

$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ и $(25, 22, 2, 6, 4, 11, 20, 25, 12, 8, 13, 22, 14)$.

Заключение

В работе описана процедура списочного декодирования биортогональных вейвлет-кодов с заданным кодовым расстоянием на примере вейвлет-кода $W[26, 13, 12]$. Предложенная процедура списочного декодирования может быть применена к любому коду, являющемуся подпространством кода Рида — Соломона. Алгоритм программно реализован (свидетельство № 2017619148).

Работа выполнена под руководством проф. А. А. Соловьева.

ЛИТЕРАТУРА

1. *Guruswami V. and Sudan M.* Improved Decoding of Reed — Solomon and Algebraic — Geometric Codes. Electronic Colloquium on Computational Complexity. 1998. Report No. 43.
2. *Соловьев А. А., Черников Д. В.* Биортогональные вейвлет-коды с заданным кодовым расстоянием // Дискретная математика. 2017. Т. 29. № 2. С. 96–108.
3. *Ruth R. and Ruckenstein G.* Efficient decoding of Reed — Solomon codes beyond half the minimum distance // IEEE Trans. Inform. Theory. 2000. V. 46. No. 1. P. 246–257.
4. *McEliece R. J.* The Guruswami — Sudan Decoding Algorithm for Reed — Solomon Codes. IPN Progress Report 42-153. May 15, 2003.
5. *Черников Д. В.* Полифазная схема помехоустойчивого кодирования над полями нечетной характеристики // Челябинский физико-математический журнал. 2016. Т. 1. № 3. С. 77–85.

REFERENCES

1. *Guruswami V. and Sudan M.* Improved Decoding of Reed — Solomon and Algebraic — Geometric Codes. Electronic Colloquium on Computational Complexity, 1998, Report no. 43.
2. *Soloviev A. A. and Chernikov D. V.* Biortogonal'nye veyvlet-kody s zadannym kodovym rasstoyaniem [Biorthogonal wavelet codes with predetermined code distance]. Diskr. Mat., 2017, vol. 29, no. 2, pp. 96–108. (in Russian)
3. *Ruth R. and Ruckenstein G.* Efficient decoding of Reed — Solomon codes beyond half the minimum distance. IEEE Trans. Inform. Theory, 2000, vol. 46, no. 1, pp. 246–257.
4. *McEliece R. J.* The Guruswami — Sudan Decoding Algorithm for Reed — Solomon Codes. IPN Progress Report 42-153. May 15, 2003.
5. *Chernikov D. V.* Polifaznaya skhema pomekhoustoychivogo kodirovaniya nad polyami nechetnoy kharakteristiki [Polyphase scheme of noise-immune encoding over fields of odd characteristic]. Chelyab. Fiz.-Mat. Zh., 2016, vol. 1, no. 3, pp. 77–85. (in Russian)

ПРИКЛАДНАЯ ТЕОРИЯ АВТОМАТОВ

УДК 519.716.35

О ЛОКАЛЬНОЙ ОБРАТИМОСТИ КОНЕЧНЫХ АВТОМАТОВ БЕЗ ПОТЕРИ ИНФОРМАЦИИ[1]

О. А. Логачев

Московский государственный университет имени М. В. Ломоносова, г. Москва, Россия

Рассматриваются вопросы восстановления фрагментов входных слов конечных автоматов без потери информации по известным выходным словам (локальное обращение). Показана связь локального обращения автомата из этого класса со свойством синхронизируемости ассоциированного с ним автомата без выхода. Найдены новые классы регистров сдвига с фильтрующими булевыми функциями, допускающих локальное обращение.

Ключевые слова: *конечный автомат, автомат без потери информации, локальная обратимость, регистр сдвига, булева функция.*

DOI 10.17223/20710410/39/7

ON THE LOCAL INVERTIBILITY OF FINITE STATE INFORMATION LOSSLESS AUTOMATA

O. A. Logachev

Lomonosov Moscow State University, Moscow, Russia

E-mail: logol@iisi.msu.ru

A Mealy finite automaton $\mathcal{M} = (A, Q, A, \varphi, \psi)$ with input and output alphabets A, state set Q, and transition and output functions $\varphi : Q\times A\to Q$ and $\psi : Q\times A\to A$ respectively is said to be an information lossless automaton (ILA) if a map $\psi_q\colon A \to A$ defined as $\psi_q(a) = \psi(q, a)$ is a permutation on A for any q in Q. The ILA $\mathcal{M}$ is called locally invertible if there exist $u \in A^n$ and $n \in \mathbb{N}$ such that, for any $x^1 = x^1_1x^1_2\ldots$, $x^2 = x^2_1x^2_2\ldots \in A^\infty$, $q^1, q^2 \in Q$, $w \in A^m$, $m \in \mathbb{N}$, and $y \in A^\infty$, the equality $\psi(q^1, x^1) = \psi(q^2, x^2) = wuy$ implies $(x^1_{m+n+1}, x^1_{m+n+2}, \ldots) = (x^2_{m+n+1}, x^2_{m+n+2}, \ldots)$. For ILA $\mathcal{M}$, we define an automaton without outputs $\mathcal{B}(\mathcal{M}) = (A, Q, \delta)$ where $\delta(q, b) = \varphi(q, \psi_q^{-1}(b))$, $q \in Q$, and $b \in A$. The automaton $\mathcal{B}(\mathcal{M})$ is called synchronizable if there exist $v \in A^l$, $l \in \mathbb{N}$, and $q_0 \in Q$ such that $\delta(q, v) = q_0$ for any $q \in Q$. Our main results are the following: 1) we have proved that ILA $\mathcal{M}$ is locally invertible iff the automation $\mathcal{B}(\mathcal{M})$ is synchronizable, 2) we have constructed some new classes of locally invertible binary shift registers with output functions being some monotonic Boolean functions, namely the nondecreasing (nonincreasing) functions for which the weights of all the minimal (respectively maximal) elements in support are less or more than the half of the number of variables.

Keywords: *finite state automaton, information lossless automaton, local invertibility, shift register.*

[1]Работа поддержана грантами РФФИ № 16-01-00226А и 16-01-00470А.

1. Необходимые понятия и определения

Пусть A — конечное множество (алфавит), A^* — множество слов конечной длины в этом алфавите (включая пустое слово λ), A^∞ — множество бесконечных вправо последовательностей в алфавите A.

Для последовательности из A^∞ определим оператор редуцирования. Пусть $i, j \in \mathbb{N}$, $1 \leqslant i < j$. Тогда для любой последовательности $x \in A^\infty$ положим

$$x_{[i,j]} = x_i x_{i+1} \dots x_j \quad \text{и} \quad x_{[i,j)} = x_i x_{i+1} \dots x_{j-1}.$$

Если $i = j$, то $x_{[i,i]} = x_{[i]} = x_i$. Кроме того, положим $x_{[i,\infty)} = x_i x_{i+1} \dots$ Аналогично оператор редуцирования определим для элементов множества A^*.

Пусть

$$\mathcal{M} = (A, Q, B, \varphi, \psi)$$

— конечный автомат Мили, где конечные множества A и B — входной и выходной алфавиты соответственно; Q — конечное множество состояний; $\varphi : Q \times A \to Q$ — функция переходов; $\psi : Q \times A \to B$ — функция выходов автомата.

Естественным образом распространим действие функций φ и ψ на $Q \times A^*$ [1]. Пусть $x = x_1 x_2 \dots x_k \in A^*, q \in Q$. Если

$$q_1 = \varphi(q, x_1),\ q_2 = \varphi(q_1, x_2),\ \dots,\ q_k = \varphi(q_{k-1}, x_k),$$
$$y_1 = \psi(q, x_1),\ y_2 = \psi(q_1, x_2),\ \dots,\ y_k = \psi(q_{k-1}, x_k),$$

где $q_1, \dots, q_k \in Q, y_1, \dots, y_k \in B$, то будем полагать

$$\varphi(q, x) = q_k,\ \psi(q, x) = y_1 y_2 \dots y_k.$$

Кроме того, для любого q из Q будем полагать $\varphi(q, \lambda) = q$, $\psi(q, \lambda) = \lambda$. Аналогичным образом распространим действие функций на множество $Q \times A^\infty$.

Определение 1 [2]. Состояния q_1 и q_2 автомата $\mathcal{M}$ называются эквивалентными, если при любых $x \in A^*$ выполнено $\psi(q_1, x) = \psi(q_2, x)$. Автомат $\mathcal{M}$ называется приведённым, если у него нет эквивалентных состояний.

Определение 2 [2]. Автомат Мили $\mathcal{M} = (A, Q, B, \varphi, \psi)$ называется автоматом без потери информации (БПИ-автоматом), если $\#A = \#B$ ($\#A$ — мощность конечного множества A) и для любого состояния q из Q отображение $\psi_q = \psi(q, \cdot) : A \to B$ является взаимно однозначным.

Замечание 1. Очевидно, что любая последовательность из A^∞ может быть получена на выходе БПИ-автомата $\mathcal{M}$. Более того, при этом в качестве начального состояния может быть выбрано произвольное состояние из Q. Следовательно, для любой выходной последовательности (слова) автомата $\mathcal{M}$ существует ровно $\#Q$ различных пар начальное состояние–входная последовательность (входное слово), перерабатываемых автоматом $\mathcal{M}$ в данную выходную последовательность (слово).

Пусть $\mathcal{M} = (A, Q, B, \varphi, \psi)$ — конечный автомат Мили, удовлетворяющий условию $\#A = \#B$. Для удобства (не теряя общности) будем считать, что $A = B$. Вместе с тем для рассматриваемых далее конструкций важно различать входные и выходные слова (последовательности) автомата $\mathcal{M}$. Будем обозначать входные слова (последовательности) автомата $x = x_1 x_2 \dots x_r, a = a_1 a_2 \dots a_r \in A^r$, $r \in \mathbb{N}$ ($x = x_1 x_2 \dots, a = a_1 a_2 \dots \in \in A^\infty$), а его выходные слова (последовательности) — $y = y_1 y_2 \dots y_s, b = b_1 b_2 \dots b_s \in A^*$ ($y = y_1 y_2 \dots, b = b_1 b_2 \dots \in A^\infty$). Кроме того, если для пары последовательностей

$x, y \in A^\infty$ и состояния $q \in Q$ равенство $\psi(q, x_{[1,j]}) = y_{[1,j]}$ выполняется для всех $j \geqslant 1$, то будем писать $\psi(q, x) = y$.

Будем обозначать через $\mathcal{B} = (A, Q, \delta)$ конечный автомат без выхода, где A — входной алфавит; Q — множество состояний; $\delta : Q \times A \to Q$ — функция переходов. Естественным образом (как ранее для автомата Мили) распространим действие функции δ на множества $Q \times A^k$, $k \in \mathbb{N}$ и $Q \times A^\infty$. Если $S \subseteq Q$ и $y \in A^*$, то будем полагать $\delta(S, y) = \bigcup\limits_{q \in S} \{\delta(q, y)\}$.

Пусть $A = \mathbb{F}_2 = \{0, 1\}$ — поле Галуа. Для произвольного натурального n будем рассматривать $\mathbb{F}_2^n$ — векторное пространство наборов (векторов) длины n с компонентами из $\mathbb{F}_2$. Операции сложения в $\mathbb{F}_2$ и $\mathbb{F}_2^n$ (покомпонентно) будем обозначать «$\oplus$», а операцию умножения в $\mathbb{F}_2$ — «$\cdot$» (обычно эта операция опускается в алгебраических выражениях). Зафиксируем обозначения для двух векторов из $\mathbb{F}_2^n$: $0^n = (0, \ldots, 0)$ и $1^n = (1, \ldots, 1)$, а также канонический базис для $\mathbb{F}_2^n$:

$$e^1 = (1, 0, \ldots, 0),\ e^2 = (0, 1, \ldots, 0),\ \ldots,\ e^n = (0, 0, \ldots, 1).$$

Тогда вектор x из $\mathbb{F}_2^n$ может быть представлен в виде $x = x_1 e^1 \oplus \ldots \oplus x_n e^n$, где коэффициенты x_i $(1 \leqslant i \leqslant n)$ из $\mathbb{F}_2$ называются координатами вектора (компонентами набора) x относительно канонического базиса.

Булевой функцией от n переменных называется отображение $f : \mathbb{F}_2^n \to \mathbb{F}_2$. Множество всех булевых функций от n переменных будем обозначать $\mathcal{F}_n$. Для записи значений функции f на наборе x будем использовать следующие выражения:

$$f(x) = f(x_1 e^1 \oplus \ldots \oplus x_n e^n) = f(x_1, \ldots, x_n).$$

Координаты $x_1, \ldots, x_n$ будем называть переменными функции f.

Будем говорить, что функция f существенно зависит от переменной x_i $(1 \leqslant i \leqslant n)$, если выполнено условие $f(x) \oplus f(x \oplus e^i) \not\equiv 0$. Булева функция f линейно зависит от переменной x_i, если $f(x) \oplus f(x \oplus e^i) \equiv 1$. Каждая булева функция $f \in \mathcal{F}_n$ может быть единственным образом представлена в виде полинома Жегалкина (или алгебраической нормальной формы — АНФ):

$$f\,(x_1, \ldots, x_n) = \bigoplus_{k=0}^{n} \bigoplus_{1 \leqslant i_1 < \ldots < i_k \leqslant n} a_{i_1 \ldots i_k} x_{i_1} \cdot \ldots \cdot x_{i_k},\ a_{i_1 \ldots i_k} \in \mathbb{F}_2.$$

Максимальная длина монома в АНФ функции f называется алгебраической степенью этой функции — $\deg(f)$. Через $\mathrm{wt}(f)$ будем обозначать вес булевой функции f: $\mathrm{wt}(f) = \sum\limits_{x \in \mathbb{F}_2^n} f(x)$.

2. Локальная обратимость конечных автоматов без потери информации

Частичное обращение дискретных ограниченно-детерминированных функций имеет характерную особенность [3]. Локализация (т. е. положение) однозначно восстанавливаемых фрагментов прообраза не может быть задана детерминированно и имеет характер случайного процесса, параметры которого определяются свойствами частично обратного автомата. Однако для автоматов без потери информации может быть рассмотрен вариант частичного обращения, для которого позиция однозначно восстанавливаемого фрагмента в прообразе точно определяется появлением на выходе автомата специальных слов — индикаторов. Этот вариант частичного обращения в [4] назван локальным обращением.

Определение 3. Автомат Мили $\mathcal{M} = (A, Q, B, \varphi, \psi)$ без потери информации обладает свойством локальной обратимости, если существует слово $u \in B^l$, $l \in \mathbb{N}$, для которого выполнено следующее условие: для любых $x^1, x^2 \in A^\infty$, $q^1, q^2 \in Q$, $i \in \mathbb{N}$, таких, что

$$\psi(q^1, x^1) = \psi(q^2, x^2) \ \text{ и } \ \psi(q^1, x^1)_{[i,i+l-1]} = \psi(q^2, x^2)_{[i,i+l-1]} = u,$$

справедливо равенство $x^1_{[i+l,\infty]} = x^2_{[i+l,\infty]}$. Слово u будем называть индикатором локального обращения для автомата $\mathcal{M}$.

3. Синхронизируемость конечных автоматов без выхода

В теории конечных автоматов значительное внимание уделяется изучению понятия «синхронизируемость автомата», имеющего важные практические приложения. Впервые это понятие было формализовано в работе Я. Черны [5], хотя ранее неявно оно использовалось в научных исследованиях по крайней мере с 1956 г. Поскольку статья Я. Черны была опубликована на словацком языке, то она длительное время оставалась неизвестной (см., например, [6 – 8]). Необходимо также отметить тесную связь этого понятия с теорией автоматов без потери информации конечного порядка, теорией префиксных кодов и символической динамикой.

Определение 4. Конечный автомат без выхода $\mathcal{B} = (A, Q, \delta)$ называется синхронизируемым, если существует слово $y \in A^*$, такое, что $\#\delta(Q, y) = 1$.

Замечание 2 (гипотеза Черны). Пусть $r \in \mathbb{N}$ и $C(r)$ — максимальная длина кратчайших синхронизирующих слов синхронизируемых автоматов с r состояниями (функция Черны). Для этой функции справедливы следующие оценки [5, 9]:

$$(r-1)^2 \leqslant C(r) \leqslant \frac{r^3 - r}{6}.$$

В частности, в [5] построена серия автоматов с кратчайшими синхронизирующими словами длины $(r-1)^2$. Под гипотезой Я. Черны понимают его предположение о том, что описанная в [5] серия автоматов реализует наихудший в смысле скорости синхронизации случай, т. е. любой синхронизируемый автомат с r состояниями обладает синхронизирующим словом длины не более $(r-1)^2$ (подробнее см. [10]).

Лемма 1 (критерий синхронизируемости). Автомат без выхода $\mathcal{B} = (A, Q, \delta)$ является синхронизируемым тогда и только тогда, когда для любой пары состояний $q, q' \in Q$ существует слово $y = y(q, q') \in A^*$, такое, что $\delta(q, y) = \delta(q', y)$.

Доказательство. Пусть автомат без выхода $\mathcal{B}$ синхронизируемый. Тогда существует такое слово $y \in A^*$, что $\#\delta(Q, y) = 1$. Следовательно, необходимость очевидна.

Докажем достаточность. Предположим, что условие леммы выполнено. Пусть $q, \tilde{q} \in Q$, $q \neq \tilde{q}$. Тогда существует слово $y^1 \in A^*$, такое, что $\delta(q, y^1) = \delta(\tilde{q}, y^1)$ и для множества состояний $S^1 = \delta(Q, y^1)$ выполнено неравенство $\#S^1 \leqslant \#Q - 1$. Если $\#S^1 \geqslant 2$, то выберем пару состояний $q^1, \tilde{q}^1 \in S^1$, $q^1 \neq \tilde{q}^1$. По условию леммы для этих состояний существует слово $y^2 \in A^*$, такое, что $\delta(q^1, y^2) = \delta(\tilde{q}^1, y^2)$. Следовательно, для множества состояний $S^2 = \delta(S^1, y^2)$ выполнено неравенство $\#S^2 \leqslant \#Q - 2$. Если $\#S^2 \geqslant 2$, то, продолжив аналогичные рассуждения, определим последовательность множеств состояний $S^3, S^4, \ldots$ Для них выполнены неравенства $\#S^i \leqslant \#Q - i$, $i = 3, 4, \ldots$ Поскольку Q — конечное множество, найдётся $t \leqslant \#Q - 1$, такое, что $\#S^t = 1$. Тогда для слова $y = y^1 y^2 \ldots y^t \in A^*$ выполнено условие $\#\delta(Q, y) = 1$, т. е. автомат без выхода $\mathcal{B}$ синхронизируемый. ■

4. Локальная обратимость и синхронизируемость

Для дальнейшего изучения свойства локальной обратимости введём понятие автомата без выхода, ассоциированного с автоматом без потери информации.

Определение 5. Пусть $\mathcal{M} = (A, Q, A, \varphi, \psi)$ — произвольный конечный автомат без потери информации. Будем называть автоматом без выхода, ассоциированным с $\mathcal{M}$, автомат $\mathcal{B}(\mathcal{M})$, задаваемый следующим образом:

— A — входной алфавит автомата $\mathcal{B}(\mathcal{M})$;
— Q — множество состояний автомата $\mathcal{B}(\mathcal{M})$;
— δ — функция переходов автомата $\mathcal{B}(\mathcal{M})$, определяемая соотношением

$$\delta(q, b) = \varphi(q, \psi_q^{-1}(b))$$

для любых $q \in Q$, $b \in A$.

Теперь можно сформулировать утверждение, связывающее свойство локальной обратимости автомата $\mathcal{M}$ с синхронизируемостью автомата $\mathcal{B}(\mathcal{M})$.

Замечание 3. В ходе доказательства данного утверждения будем пользоваться известным свойством автоматов без потери информации. При произвольном фиксированном состоянии $q \in Q$ и любом $r \in \mathbb{N}$ отображение из A^r в A^r вида $\psi_q : z \to \psi_q(z) = \psi(q, z)$, $z \in A^r$б является взаимно однозначным.

Теорема 1. Приведённый автомат без потери информации $\mathcal{M} = (A, Q, A, \varphi, \psi)$ обладает свойством локальной обратимости тогда и только тогда, когда ассоциированный с ним автомат без выхода $\mathcal{B}(\mathcal{M})$ синхронизируем.

Доказательство. Предположим, что автомат $\mathcal{B}(\mathcal{M})$ синхронизируем. Тогда существует синхронизирующее слово $u = u_1 u_2 \dots u_l$ и состояние $q_0 \in Q$, такие, что $\delta(q, u) = q^0$ для любого $q \in Q$. Пусть $x^1, x^2 \in A^\infty$, $q^1, q^2 \in Q$, $i \in \mathbb{N}$ такие, что

$$\psi\left(q^1, x^1\right) = \psi\left(q^2, x^2\right) = y \in A^\infty,\ y_{[i,i+l-1]} = u. \tag{1}$$

Покажем, что $\varphi\left(q^1, x^1_{[1,i+l-1]}\right) = \varphi\left(q^2, x^2_{[1,i+l-1]}\right) = q^0$. Действительно, если

$$q' = \varphi\left(q^1, x^1_{[1,i-1]}\right),\ q'' = \varphi\left(q^2, x^2_{[1,i-1]}\right),$$

то, воспользовавшись соотношениями (1) и тем, что u — синхронизирующее слово для $\mathcal{B}(\mathcal{M})$, получаем

$$\varphi\left(q', x^1_{[i,i+l-1]}\right) = \delta\left(q', y_{[i,i+l-1]}\right) = \delta(q', u) = q^0,$$
$$\varphi\left(q'', x^2_{[i,i+l-1]}\right) = \delta\left(q'', y_{[i,i+l-1]}\right) = \delta(q'', u) = q^0.$$

Поскольку в соответствии с (1) $\psi\left(q^0, x^1_{[i+l,\infty)}\right) = \psi\left(q^0, x^2_{[i+l,\infty)}\right) = y_{[i+l,\infty)}$ и частичная функция выходов (при фиксированном q) $\psi_q(\cdot) = \psi(q, \cdot)$ является перестановкой элементов A, то $x^1_{[i+l,\infty)} = x^2_{[i+l,\infty)}$. Следовательно, БПИ-автомат $\mathcal{M}$ обладает свойством локальной обратимости и слово u — его индикатор.

Обратно. Пусть приведённый БПИ-автомат $\mathcal{M}$ обладает свойством локальной обратимости. Тогда существует слово-индикатор $u \in A^l$, для которого выполняется условие: для любых $x^1, x^2 \in A^\infty$, $q^1, q^2 \in Q$, $i \in \mathbb{N}$, таких, что $\psi(q^1, x^1) = \psi(q^2, x^2)$ и $\psi(q^1, x^1)_{[i,i+l-1]} = \psi(q^2, x^2)_{[i,i+l-1]} = u$, выполнено равенство $x^1_{[i+l,\infty)} = x^2_{[i+l,\infty)}$.

Будем доказывать от противного. Предположим, что автомат $\mathcal{B}(\mathcal{M})$ не является синхронизируемым. Тогда существует пара состояний $q', q'' \in Q$, $q' \neq q''$, для которых выполнено $\delta(q', z) \neq \delta(q'', z)$ при любом слове $z \in A^*$. Следовательно, и для слова-индикатора u выполнено $\delta(q', u) \neq \delta(q'', u)$.

Так как $\mathcal{M}$ — БПИ-автомат, для пар (q', u) и (q'', u) однозначно определяются входные слова этого автомата v' и v'' соответственно такие, что $\psi(q', v') = \psi(q'', v'') = u$ и $\varphi(q', v') = \delta(q', u) \neq \delta(q'', u) = \varphi(q'', v'')$.

Пусть y — произвольная последовательность из A^∞. Тогда для состояний $\varphi(q', v')$ и $\varphi(q'', v'')$ однозначно определяются последовательности $x', x'' \in A^\infty$, такие, что $\psi(\varphi(q', v'), x') = \psi(\varphi(q'', v''), x'') = y$ и

$$\psi(q', v'x') = \psi(q'', v''x'') = uy. \tag{2}$$

Так как БПИ-автомат $\mathcal{M}$ обладает свойством локальной обратимости и u — индикатор, из соотношения (2) вытекает равенство $x' = x'' = x$. Следовательно, для любой последовательности $y \in A^\infty$ существует $x \in A^\infty$, что

$$\psi(\varphi(q', v'), x) = \psi(\varphi(q'', v''), x) = y. \tag{3}$$

Для произвольного $r \in \mathbb{N}$ рассмотрим семейство из $t = (\#A)^r$ последовательностей $y^1, y^2, \ldots, y^t$ из A^∞, таких, что

$$\left\{y^i_{[1,r]} : \ i = 1, \ldots, t\right\} = A^r. \tag{4}$$

Для соответствующих (см. (3)) входных последовательностей $x^1, x^2, \ldots, x^t$ из A^∞ имеем

$$\psi\left(\varphi(q', v'), x^i\right) = \psi\left(\varphi(q'', v''), x^i\right) = y^i, \ i = 1, 2, \ldots, t.$$

Поскольку $\mathcal{M}$ — БПИ-автомат и выполнено (4), то $\left\{x^i_{[1,r]} : i = 1, \ldots, t\right\} = A^r$ и по замечанию 3

$$\psi\left(\varphi(q', v'), x^i_{[1,r]}\right) = \psi\left(\varphi(q'', v''), x^i_{[1,r]}\right) = y^i_{[1,r]}, \ i = 1, 2, \ldots, t. \tag{5}$$

Так как соотношения (5) выполняются при любом $r \in \mathbb{N}$, то состояния $\varphi(q', v')$ и $\varphi(q'', v'')$ эквивалентны. Это противоречит приведённости автомата $\mathcal{M}$. Следовательно, автомат $\mathcal{B}(\mathcal{M})$ синхронизируем. ■

5. Локальное обращение неавтономных регистров сдвига с фильтрующими функциями

Рассмотрим вопрос о локальном обращении для одного вида БПИ-автоматов. Интересующий нас класс автоматов — неавтономные регистры сдвига с фильтрующими функциями — используется, в частности, при синтезе генераторов псевдослучайных последовательностей. Свойство «без потери информации» в данном случае означает, что фильтрующая функция линейна по последней переменной.

Для наших целей указанный выше автомат удобно представить как автомат Мили вида

$$\mathcal{M}_n(f) = (\mathbb{F}_2, \mathbb{F}_2^n, \mathbb{F}_2, \varphi_n, \psi_n),$$

где f — булева функция от n переменных,

$$\varphi_n((x_1, \ldots, x_n), x_{n+1}) = (x_2, \ldots, x_n, x_{n+1}),$$
$$\psi_n((x_1, \ldots, x_n), x_{n+1}) = f(x_1, \ldots, x_n) \oplus x_{n+1}.$$

Утверждение 1. Автомат, ассоциированный с $\mathcal{M}_n(f)$, имеет вид

$$\mathcal{B}(\mathcal{M}_n(f)) = (\mathbb{F}_2^n, \mathbb{F}_2, \delta_n^f),$$

где $\delta_n((s_1, \ldots, s_n), y) = (s_2, \ldots, s_n, f(s_1, \ldots, s_n) \oplus y)$, $(s_1, \ldots, s_n) \in \mathbb{F}_2^n$, $y \in \mathbb{F}_2$.

Доказательство. Проводится непосредственной проверкой выполнения условий определения 5. ■

Замечание 4. Автомат $\mathcal{B}(\mathcal{M}_n(f))$ в работах по теории кодирования и криптологии иногда называют неавтономным регистром сдвига с обратной связью f — $NFSR(f)$.

Формулировку аналога теоремы 1 для данного класса автоматов необходимо предварить некоторым вспомогательным утверждением.

Утверждение 2. Существуют функции f из $\mathcal{F}_n$, такие, что автомат $\mathcal{M}_n(f)$ не является приведённым.

Доказательство. Пусть для функции f из $\mathcal{F}_n$ выполняется условие: существует фиксированный набор $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$, такой, что

$$f(x_1 \oplus 1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n).$$

Тогда состояния $x = (x_1, x_2, \ldots, x_n)$ и $x' = (x_1 \oplus 1, x_2, \ldots, x_n)$ будут эквивалентными для автомата $\mathcal{M}_n(f)$. ■

Следовательно, для класса автоматов $\mathcal{M}_n(f)$ не выполняется условие теоремы 1. Вместе с тем утверждение этой теоремы для любого автомата $\mathcal{M}_n(f)$ и ассоциированного с ним автомата без выхода $NFSR(f)$ справедливо.

Кроме того, особенности этого класса автоматов таковы, что параметры локальной обратимости для них отличны от приведённых в определении 3.

Определение 6. Автомат $\mathcal{M}_n(f) = (\mathbb{F}_2, \mathbb{F}_2^n, \mathbb{F}_2, \varphi_n, \psi_n)$ обладает свойством локальной обратимости, если существует слово $y \in \mathbb{F}_2^l$, $l \geqslant n$, для которого выполнены следующие условия: для любых последовательностей $x^1, x^2 \in \mathbb{F}_2^\infty$, состояний $s^1, s^2 \in \mathbb{F}_2^n$ и $i \in \mathbb{N}$, таких, что $\psi_n(s^1, x^1) = \psi_n(s^2, x^2)$ и $\psi_n(s^1, x^1)_{[i,i+l-1]} = \psi_n(s^2, x^2)_{[i,i+l-1]} = y$, справедливо равенство $x^1_{[i+l-n,\infty)} = x^2_{[i+l-n,\infty)}$.

Замечание 5. Очевидно, что свойство локальной обратимости автомата $\mathcal{M}_n(f)$ остаётся содержательным и в случае, когда мы рассматриваем конечные выходные (входные) наборы, содержащие подслово-индикатор y.

Теорема 2. Автомат Мили $\mathcal{M}_n(f)$ обладает свойством локальной обратимости тогда и только тогда, когда ассоциированный с ним автомат без выхода $NFSR(f)$ является синхронизируемым.

Доказательство.

Необходимость. Предположим, что автомат $\mathcal{M}_n(f)$ обладает свойством локальной обратимости. Тогда (согласно определению 6) существует слово $y \in \mathbb{F}_2^l$, $l \geqslant n$, такое, что для любых $x^1, x^2 \in \mathbb{F}_2^l$ и состояний $s^1, s^2 \in \mathbb{F}_2^n$, таких, что

$$\psi_n(s^1, x^1) = \psi_n(s^2, x^2) = y, \tag{6}$$

выполнено равенство

$$x^1_{[l-n+1,l]} = x^2_{[l-n+1,l]}. \tag{7}$$

Рассмотрим систему уравнений (6) подробнее:

$$\begin{cases} y_1 & = f(s_1^1,\dots,s_n^1)\oplus x_1^1 = f(s_1^2,\dots,s_n^2)\oplus x_1^2,\\ & \vdots\\ y_{l-n} & = f(\dots,x_{l-n-1}^1)\oplus x_{l-n}^1 = f(\dots,x_{l-n-1}^2)\oplus x_{l-n}^2,\\ y_{l-n+1} & = f(\dots,x_{l-n}^1)\oplus x_{l-n+1}^1 = f(\dots,x_{l-n}^2)\oplus x_{l-n+1}^2,\\ & \vdots\\ y_l & = f(\dots,x_{l-1}^1)\oplus x_l^1 = f(\dots,x_{l-1}^2)\oplus x_l^2. \end{cases} \tag{8}$$

Преобразуем систему уравнений (8) в систему функционирования автомата $\mathcal{B}(\mathcal{M}_n(f)) = NFSR(f)$. Пусть имеем два экземпляра этого автомата с начальными состояниями s^1 и s^2 соответственно. На их входы подается одна и та же последовательность y. Тогда (в соответствии с (8)) получаем

$$\begin{cases} \delta_n^f((s_1^1,\dots,s_n^1),y_1) & = (s_2^1,\dots,s_n^1,x_1^1),\\ \delta_n^f((s_2^1,\dots,x_1^1),y_2) & = (s_3^1,\dots,x_1^1,x_2^1),\\ & \vdots\\ \delta_n^f((\dots,x_{l-n+1}^1),y_{l-n}) & = (\dots,x_{l-n-1}^1,x_{l-n}^1),\\ \delta_n^f((\dots,x_{l-n}^1),y_{l-n+1}) & = (\dots,x_{l-n}^1,x_{l-n+1}^1),\\ & \vdots\\ \delta_n^f((\dots,x_{l-1}^1),y_l) & = (x_{l-n+1}^1,\dots,x_l^1); \end{cases} \tag{9}$$

$$\begin{cases} \delta_n^f((s_1^2,\dots,s_n^2),y_1) & = (s_2^2,\dots,s_n^2,x_1^2),\\ \delta_n^f((s_2^2,\dots,x_1^2),y_2) & = (s_3^2,\dots,x_1^2,x_2^2),\\ & \vdots\\ \delta_n^f((\dots,x_{l-n+1}^2),y_{l-n}) & = (\dots,x_{l-n-1}^2,x_{l-n}^2),\\ \delta_n^f((\dots,x_{l-n}^2),y_{l-n+1}) & = (\dots,x_{l-n}^2,x_{l-n+1}^2),\\ & \vdots\\ \delta_n^f((\dots,x_{l-1}^2),y_l) & = (x_{l-n+1}^2,\dots,x_l^2). \end{cases} \tag{10}$$

Поскольку выполнено равенство (7), то

$$\delta_n^f(s^1,y) = \delta_n^f(s^2,y). \tag{11}$$

Так как $\mathcal{M}_n(f)$ — БПИ-автомат, для любого состояния $s\in\mathbb{F}_2^n$ существует единственное входное слово $x\in\mathbb{F}_2^l$, что $\psi_n(s,x)=y$. Следовательно, существует всего 2^n пар (s^1,x^1), (s^2,x^2), …, (s^{2^n},x^{2^n}), таких, что $\psi_n(s^i,x^i)=y$, $i=1,2,\dots,2^n$. Кроме того, имеем $\{s^1,s^2,\dots,s^{2^n}\}=\mathbb{F}_2^n$ — пространство внутренних состояний автомата $\mathcal{M}_n(f)$.

Соотношения (6)–(11) показывают, что $\delta_n^f(s^1,y)=\delta_n^f(s^2,y)=\ldots=\delta_n^f(s^{2^n},y)$, т. е. входное слово y является синхронизирующим для автомата $NFSR(f)$.

Д о с т а т о ч н о с т ь. Предположим, что $y\in\mathbb{F}_2^l$, $l\geqslant n$, является синхронизирующим словом для автомата $\mathcal{B}(\mathcal{M}_n(f))=NFSR(f)$. Тогда существует состояние $s^0\in\mathbb{F}_2^n$ автомата $NFSR(f)$, такое, что выполнены 2^n соотношений вида

$$\delta_n^f(s^1,y)=\delta_n^f(s^2,y)=\ldots=\delta_n^f(s^{2^n},y)=s^0. \tag{12}$$

Заметим, что соответствующие (12) булевы системы уравнений можно преобразовать в системы функционирования автомата $\mathcal{M}_n(f)$ (см. (7)–(11) в обратном порядке). Тогда получаем

$$\psi_n(s^1, x^1) = \psi_n(s^2, x^2) = \ldots = \psi_n(s^{2^n}, x^{2^n}) = y,$$
$$\varphi_n(s^1, x^1) = \varphi_n(s^2, x^2) = \ldots = \varphi_n(s^{2^n}, x^{2^n}) = s^0.$$

Напомним, что $\mathcal{M}_n(f)$ — БПИ-автомат и для любого $s^i \in \mathbb{F}_2^n$ такое входное слово x^i определяется однозначно. Кроме того, для любого x^i имеем $x^i_{[l-n+1,l]} = s^0$.

Таким образом, если в выходном слове автомата $\mathcal{M}_n(f)$ встретилось подслово y, то, начиная с такта, в котором на выход автомата поступает слово y_{l-n+1}, входные символы определяются однозначно. Следовательно, автомат $\mathcal{M}_n(f)$ обладает свойством локальной обратимости. ■

6. Булевы функции со свойством синхронизируемости

Рассмотрим теперь подробнее параметры и характеристики функций из $\mathcal{F}_n$, влияющие на наличие (или отсутствие) у автомата $NFSR(f)$ свойства синхронизируемости. Для краткости будем говорить, что $f \in \mathcal{F}_n$ обладает (не обладает) свойством синхронизируемости, если автомат $NFSR(f)$ синхронизируем (не синхронизируем).

Пример 1. Пусть $f \in \mathcal{F}_n$ — самодвойственная функция, т. е.

$$f(x \oplus 1^n) \oplus 1 = f(x). \tag{13}$$

Для произвольного слова $y \in \mathbb{F}_2^l$ и пары состояний $s \in \mathbb{F}_2^n$ и $s' = s \oplus 1^n$ с учётом свойства (13) для автомата $NFSR(f)$ имеем

$$\delta_n^f(s, y_1) = (s_2, \ldots, s_n, f(s_1, \ldots, s_n) \oplus y_1),$$
$$\delta_n^f(s \oplus 1^n, y_1) = (s_2 \oplus 1, \ldots, s_n \oplus 1, f(s_1 \oplus 1, \ldots, s_n \oplus 1) \oplus y_1) =$$
$$= (s_2 \oplus 1, \ldots, s_n \oplus 1, f(s_1, \ldots, s_n) \oplus 1 \oplus y_1),$$

то есть после первого такта имеем

$$\delta_n^f(s, y_1) \oplus \delta_n^f(s \oplus 1^n, y_1) = 1^n. \tag{14}$$

Аналогично после тактов с номерами $i = 2, 3, \ldots, l$ получаем

$$\begin{aligned}
&\delta_n^f(s, y_1 y_2) \oplus \delta_n^f(s \oplus 1^n, y_1 y_2) = 1^n,\\
&\vdots\\
&\delta_n^f(s, y_1 y_2 \ldots y_i) \oplus \delta_n^f(s \oplus 1^n, y_1 y_2 \ldots y_i) = 1^n,\\
&\vdots\\
&\delta_n^f(s, y) \oplus \delta_n^f(s \oplus 1^n, y) = 1^n.
\end{aligned} \tag{15}$$

Соотношения (14) и (15) показывают, что состояния s и $s \oplus 1^n$ не могут быть переведены автоматом $NFSR(f)$ в одно и то же состояние под действием произвольного входного слова y. Следовательно, любая самодвойственная функция не обладает синхронизирующим свойством.

Пример 2. Пусть $n \in \mathbb{N}$. Рассмотрим представителей класса монотонных функций [11], а именно функции голосования.

1) Если n — чётное, то функция

$$f(x_1,\ldots,x_n)=\begin{cases}0 & \text{при } \sum\limits_{i=1}^{n} x_i \leqslant n/2,\\ 1 & \text{при } \sum\limits_{i=1}^{n} x_i > n/2\end{cases}$$

обладает синхронизирующим свойством.

2) Если n — нечётное, то функция

$$g(x_1,\ldots,x_n)=\begin{cases}0 & \text{при } \sum\limits_{i=1}^{n} x_i \leqslant (n-1)/2,\\ 1 & \text{при } \sum\limits_{i=1}^{n} x_i > (n+1)/2\end{cases}$$

не обладает синхронизирующим свойством.

Пример 3. Рассмотрим представителей класса функций с бесповторной АНФ, т. е. функций, существенно зависящих от всех своих переменных, у которых каждая переменная входит в АНФ только один раз.

1) Функция $f(x_1,\ldots,x_n)=x_1\oplus x_2\cdot\ldots\cdot x_n$ не обладает синхронизирующим свойством.
2) Функция $f(x_1,\ldots,x_n)=x_1x_2\oplus\ldots\oplus x_{n-1}x_n$, где n — чётное, обладает синхронизирующим свойством. Данная функция является представителем еще одного интересного класса — бент-функций.

Пример 4. Пусть $f\in\mathcal{F}_n$ и $f(x\oplus e^1)\oplus f(x)=1$, т. е. функция f линейна по первой переменной. Тогда отображение $\delta_n^f(\cdot,\varepsilon):\mathbb{F}_2^n\to\mathbb{F}_2^n$ является взаимно однозначным, т. е. функция f не обладает синхронизирующим свойством.

Рассмотрим ряд утверждений, описывающих свойства классов функций относительно наличия или отсутствия свойства синхронизируемости.

Теорема 3. Пусть функция f из $\mathcal{F}_n$, существенно зависящая от всех переменных, удовлетворяет следующему условию: АНФ этой функции содержит лишь мономы алгебраической степени 2 вида x_ix_{i+1} для некоторых $i\in\{1,2,\ldots,n-1\}$. Тогда функция f обладает синхронизирующим свойством.

Доказательство. Предположим, что $s,s'\in\mathbb{F}_2^n$, $s\neq s'$, — произвольная пара несовпадающих состояний автомата $NFSR(f)$. Тогда существует входное слово $y^1\in\in\mathbb{F}_2^n$ этого автомата, что

$$\begin{aligned}\delta_n^f(s,y_1)&=(*,0,*,0,\ldots),\\ \delta_n^f(s',y_1)&=(0,*,0,*,\ldots).\end{aligned}\tag{16}$$

Звездочкой в (16) помечены компоненты векторов, конкретные значения которых в доказательстве не важны. Существование слова y^1 вытекает из линейной зависимости очередного состояния $NFSR(f)$ от входного символа. Нетрудно заметить, что для наборов вида $(*,0,*,0,\ldots)\in\mathbb{F}_2^n$ или $(0,*,0,*,\ldots)\in\mathbb{F}_2^n$ значения функции f равны 0. Выбрав $y^2=0^n$, получаем $\delta_n^f((*,0,*,0,\ldots),0^n)=\delta_n^f((0,*,0,*,\ldots),0^n)=0^n$. В результате для входного слова $y=y^1y^2\in\mathbb{F}_2^{2n}$ имеем $\delta_n^f(s,y)=\delta_n^f(s',y)$. Это свойство равносильно существованию для $NFSR(f)$ синхронизирующей последовательности. Следовательно, функция f обладает свойством синхронизируемости. ■

Множество функций из $\mathcal{F}_n$, обладающих синхронизирующим свойством и существенно зависящих от крайних переменных, будем обозначать $\mathcal{F}_n^{\mathrm{sync}}$. Очевидно, что $\mathcal{F}_n \setminus \mathcal{F}_n^{\mathrm{sync}} \neq \varnothing$ (см. примеры 1–3).

Обозначим через $\delta_{n,\varepsilon}^f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ частичную функцию переходов автомата $NFSR(f)$:

$$\delta_{n,\varepsilon}^f(x) = \delta_n^f(x, \varepsilon)$$

для любых $x \in \mathbb{F}_2^n$ и $\varepsilon \in \mathbb{F}_2$. Полугруппу, порождённую отображениями $\delta_{n,0}^f$ и $\delta_{n,1}^f$, называют полугруппой автомата $NFSR(f)$ [2]: $\mathrm{Sem}(NFSR(f)) = \langle \delta_{n,0}^f, \delta_{n,1}^f \rangle$.

Утверждение 3. Функция f из $\mathcal{F}_n$ обладает свойством синхронизируемости ($f \in \in \mathcal{F}_n^{\mathrm{sync}}$) тогда и только тогда, когда в полугруппе $\mathrm{Sem}(NFSR(F))$ имеется константное отображение.

Доказательство. Непосредственно вытекает из определения 4. ■

Исследуем некоторые особенности «монотонности» функций, обладающих свойством синхронизируемости.

Для двух векторов $x = (x_1, \ldots, x_n)$ и $y = (y_1, \ldots, y_n)$ из $\mathbb{F}_2^n$ выполнено отношение предшествования $x \preccurlyeq y$ (или $y \succcurlyeq x$) [11], если $x_1 \leqslant y_1, \ldots, x_n \leqslant y_n$. Если $x \preccurlyeq y$ и $y \preccurlyeq z$, то $x \preccurlyeq z$. Не все пары находятся в отношении предшествования. Таким образом, $\mathbb{F}_2^n$ с отношением $\preccurlyeq$ является частично упорядоченным множеством.

Определение 7. Функция f из $\mathcal{F}_n$ называется неубывающей (в [11] — монотонной), если для любых двух векторов x и y, таких, что $x \preccurlyeq y$, имеет место неравенство $f(x) \leqslant f(y)$. Будем обозначать через $\mathcal{FM}_n^+$ класс неубывающих функций из $\mathcal{F}_n$.

Функция f из $\mathcal{F}_n$ называется невозрастающей, если для любых двух векторов x и y, таких, что $x \preccurlyeq y$, имеет место неравенство $f(x) \geqslant f(y)$. Будем обозначать через $\mathcal{FM}_n^-$ класс невозрастающих функций из $\mathcal{F}_n$. Объединение этих классов будем обозначать

$$\mathcal{FM}_n = \mathcal{FM}_n^+ \cup \mathcal{FM}_n^- \subset \mathcal{F}_n.$$

Нетрудно заметить, что если $f \in \mathcal{FM}_n^+$, то функции $f'(x) = f(x) \oplus 1$ и $f''(x) = \\ = f(x \oplus 1^n)$ принадлежат классу $\mathcal{FM}_n^-$. С другой стороны, если $g \in \mathcal{FM}_n^-$, то функции $g'(x) = g(x) \oplus 1$ и $g''(x) = g(x \oplus 1^n)$ принадлежат классу $\mathcal{FM}_n^+$.

Если $f \in \mathcal{FM}_n^+$, то будем обозначать множество минимальных элементов носителя функции f как

$$\min \mathrm{supp}(f) = \{x \in \mathrm{supp}(f) : \forall y \prec x (f(y) = 0)\}.$$

Если $f \in \mathcal{FM}_n^-$, будем обозначать множество максимальных элементов носителя функции f как

$$\max \mathrm{supp}(f) = \{x \in \mathrm{supp}(f) : \forall y \succ x (f(y) = 0)\}.$$

Лемма 2. Следующие условия эквивалентны:

1) $f(x) \in \mathcal{F}_n^{\mathrm{sync}}$;
2) $g(x) = 1 \oplus f(x) \in \mathcal{F}_n^{\mathrm{sync}}$;
3) $h(x) = 1 \oplus f(x \oplus 1^n) \in \mathcal{F}_n^{\mathrm{sync}}$.

Доказательство.

1⇔2. Пусть функция f из $\mathcal{F}_n$ обладает свойством синхронизируемости. Тогда для автомата $NFSR(f)$ существует синхронизирующее слово $y = y_1 y_2 \dots y_l \in \mathbb{F}_2^l$. Следовательно, для любого $s \in \mathbb{F}_2^n$ выполнено

$$\delta_n^f(s, y) = s^0 = (s_1^0, \dots, s_n^0), \tag{17}$$

где s^0 — некоторое фиксированное состояние автомата $NFSR(f)$.

Рассмотрим соотношение (17) подробнее:

$$\begin{aligned}
&\delta_n^f((s_1, \dots, s_n), y_1) = (s_2, \dots, s_{n+1}), && s_{n+1} = f(s_1, \dots, s_n) \oplus y_1,\\
&\delta_n^f((s_2, \dots, s_{n+1}), y_2) = (s_3, \dots, s_{n+2}), && s_{n+2} = f(s_2, \dots, s_{n+1}) \oplus y_2,\\
&\vdots\\
&\delta_n^f((s_i, \dots, s_{i+n-1}), y_i) = (s_{i+1}, \dots, s_{i+n}), && s_{i+n} = f(s_i, \dots, s_{i+n-1}) \oplus y_i,\\
&\vdots\\
&\delta_n^f((s_l, \dots, s_{l+n-1}), y_l) = (s_{l+1}, \dots, s_{l+n}), && s_{l+n} = f(s_l, \dots, s_{l+n-1}) \oplus y_l,\\
&(s_{l+1}, \dots, s_{l+n}) = (s_1^0, \dots, s_n^0).
\end{aligned} \tag{18}$$

Нетрудно заметить, что i-е уравнение системы (18) можно представить в виде

$$\begin{aligned}
&\delta_n^f((s_i, \dots, s_{i+n-1}), y_i) = (s_{i+1}, \dots, s_{i+n-1}, f(s_i, \dots, s_{i+n-1}) \oplus y_i) =\\
&= (s_{i+1}, \dots, s_{i+n-1}, f(s_i, \dots, s_{i+n-1}) \oplus 1 \oplus (y_i \oplus 1)) =\\
&= \delta_n^{f\oplus 1}((s_i, \dots, s_{i+n-1}), y_i \oplus 1) = \delta_n^g((s_i, \dots, s_{i+n-1}), y_i \oplus 1).
\end{aligned} \tag{19}$$

Соотношения (19) для $i = 1, 2, \dots, l$ показывают, что для любого $s \in \mathbb{F}_2^n$ выполнено $\delta_n^g(s, y \oplus 1^l) = s^0$, т. е. $g(x) = 1 \oplus f(x) \in \mathcal{F}_n^{\text{sync}}$. Получаем: из условия 1 следует условие 2. Обратное утверждение доказывается аналогичными рассуждениями из предположения, что $g(x) = 1 \oplus f(x) \in \mathcal{F}_n^{\text{sync}}$.

1⇔3. Пусть функция f из $\mathcal{F}_n^{\text{sync}}$. Тогда для автомата $NFSR(f)$ существует синхронизирующее слово $y = y_1 y_2 \dots y_l$ и выполнено условие (17). Рассмотрим отображение $\delta_n^h(\cdot, y)$, реализуемое автоматом $NFSR(h)$, взяв в качестве начального состояния $s' = s \oplus 1^n$, где s — произвольное состояние из $\mathbb{F}_2^n$. Выпишем систему уравнений для $\delta_n^h(s \oplus 1^n, y)$:

$$\begin{cases}
\delta_n^h((s_1 \oplus 1, \dots, s_n \oplus 1), y_1) = (s_2 \oplus 1, \dots, s_{n+1} \oplus 1),\\
\vdots\\
\delta_n^h((s_i \oplus 1, \dots, s_{i+n-1} \oplus 1), y_i) = (s_{i+1} \oplus 1, \dots, s_{i+n} \oplus 1),\\
\vdots\\
\delta_n^h((s_l \oplus 1, \dots, s_{l+n-1} \oplus 1), y_l) = (s_{l+1} \oplus 1, \dots, s_{l+n} \oplus 1),
\end{cases}$$

где $s_i, i = 1, 2, \dots, l+n$ те же, что и в (18). Следовательно, $\delta_n^h(s \oplus 1^n, y) = s^0 \oplus 1^n$ для любых $s \in \mathbb{F}_2^n$. Очевидно, что тогда $\delta_n^h(s, y) = s^0 \oplus 1^n$, т. е. $h(x) \in \mathcal{F}_n^{\text{sync}}$.

Обратное утверждение очевидно, так как $h(x \oplus 1^n) \oplus 1 = f(x)$. ■

Следствие 1. Функция $f \in \mathcal{F}_n^{\text{sync}}$ тогда и только тогда, когда $h'(x) = f(x \oplus 1^n) \in \in \mathcal{F}_n^{\text{sync}}$.

Доказательство. Достаточно заметить, что $h'(x)$ получается из $f(x)$ комбинацией преобразований, описанных в условиях 2 и 3 леммы 2. ■

Для функций из $\mathcal{FM}_n^- \cup \mathcal{FM}_n^+$ справедливы следующие утверждения.

Теорема 4. Пусть n — чётное натуральное число.

1) Если $f \in \mathcal{FM}_n^+$ и для любого $x \in \min \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) > n/2$ (либо $\operatorname{wt}(x) < n/2$), то функция f обладает свойством синхронизируемости.
2) Если $f \in \mathcal{FM}_n^-$ и для любого $x \in \max \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) < n/2$ (либо $\operatorname{wt}(x) > n/2$), то функция f обладает свойством синхронизируемости.

Доказательство.

1) Пусть $f \in \mathcal{FM}_n^+$ и для любого $x \in \min \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) > n/2$. Рассмотрим произвольную пару состояний s^1 и s^2, $s^1 \neq s^2$, автомата $NFSR(f)$. Как и в теореме 2, воспользуемся линейной зависимостью очередного состояния $NFSR(f)$ от входного символа. Тогда существует входное слово $y^1 \in \mathbb{F}_2^n$, что $\delta_n^f(s^1, y^1) = 0^n$ и $\delta_n^f(s^2, y^1) = u = (u_1, \ldots, u_n)$ — некоторое фиксированное состояние. Подберём теперь такое входное слово $y^2 \in \mathbb{F}_2^{n/2}$, что

$$\delta_n^f(0^n, y^2) = (\underbrace{0, \ldots, 0}_{n/2}, v_1, \ldots, v_{n/2}), \quad \delta_n^f(u, y^2) = (u_{n/2+1}, \ldots, u_n, \underbrace{0, \ldots, 0}_{n/2}), \tag{20}$$

где $(v_1, \ldots, v_{n/2}) \in \mathbb{F}_2^{\frac{n}{2}}$ — некоторый набор. Поскольку для состояний из (20) выполнено

$$\operatorname{wt}((\underbrace{0, \ldots, 0}_{n/2}, v_1, \ldots, v_{n/2}) \leqslant n/2, \quad \operatorname{wt}(u_{n/2+1}, \ldots, u_n, \underbrace{0, \ldots, 0}_{n/2}) \leqslant n/2, \tag{21}$$

то

$$f(\underbrace{0, \ldots, 0}_{n/2}, v_1, \ldots, v_{n/2}) = f(u_{n/2+1}, \ldots, u_n, \underbrace{0, \ldots, 0}_{n/2}) = 0. \tag{22}$$

Для состояний (20) можно подобрать входное слово $y^3 \in \mathbb{F}_2^n$, такое, что

$$\delta_n^f((0, \ldots, 0, v_1, \ldots, v_{n/2}), y^3) = \delta_n^f((u_{n/2+1}, \ldots, u_n, 0, \ldots, 0), y^3) = 0^n,$$

а именно: учитывая (21) и (22), в каждом из n тактов будем выбирать входной символ автомата так, чтобы веса получаемых состояний не увеличивались.

В итоге получаем: для входного слова $y = y^1 y^2 y^3$ выполнено

$$\delta_n^f(s^1, y) = \delta_n^f(s^2, y) = 0^n,$$

т. е. любые два состояния автомата $NFSR(f)$ подходящим входным словом y переводятся в состояние 0^n. Следовательно, для данного автомата существует синхронизирующее слово и f обладает синхронизирующим свойством.

Предположим теперь, что $f \in \mathcal{FM}_n^+$ и для любого $x \in \min \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) < n/2$. Доказательство соответствующего утверждения теоремы проводится аналогично рассмотренному выше с заменой состояния 0^n на состояние 1^n.

2) Предположим, что $f \in \mathcal{FM}_n^-$ и для любого $x \in \max \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) < n/2$. Рассмотрим функцию $h'(x) = f(x \oplus 1^n)$. Функция $h'(x)$ принадлежит $\mathcal{FM}_n^+$. Очевидно, что если $h'(x) = 1$, то $\operatorname{wt}(x) > n/2$. Следовательно, для любого $x \in \min \operatorname{supp}(h')$ имеем $\operatorname{wt}(x) > n/2$. Согласно п. 1 теоремы, функция h' обладает синхронизирующим свойством. Тогда по следствию 1 функция $f(x) = h'(x \oplus 1^n)$ также обладает синхронизирующим свойством.

Пусть теперь $f \in \mathcal{FM}_n^-$ и для любого $x \in \max \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) > n/2$. Рассмотрим функцию $g(x) = 1 \oplus f(x)$. Ясно, что $g(x) \in \mathcal{FM}_n^+$. Если $g(x) = 1$, то $\operatorname{wt}(x) > n/2$. Следовательно, для любого $x \in \min \operatorname{supp}(g)$ имеем $\operatorname{wt}(x) > n/2$. Согласно п. 1 теоремы, функция g обладает синхронизирующим свойством. Тогда по п. 2 леммы 2 функция $f(x) = 1 \oplus g(x)$ также обладает синхронизирующим свойством. ■

Теорема 5. Пусть n — нечётное натуральное число.

1) Если $f \in \mathcal{FM}_n^+$ и для любого $x \in \min \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) > (n+1)/2$ ($\operatorname{wt}(x) < (n-1)/2$), то функция f обладает синхронизирующим свойством.
2) Если $f \in \mathcal{FM}_n^-$ и для любого $x \in \max \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) > (n+1)/2$ ($\operatorname{wt}(x) < (n-1)/2$), то функция f обладает синхронизирующим свойством.

Доказательство.

1) Пусть $f \in \mathcal{FM}_n^+$ и для любого $x \in \min \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) > (n+1)/2$. Рассмотрим произвольную пару состояний s^1 и s^2, $s^1 \neq s^2$, автомата $NFSR(f)$. Существует входное слово $y^1 \in \mathbb{F}_2^{(n+1)/2}$, что

$$\delta_n^f(s^1, y^1) = (s_{(n+3)/2}, \ldots, s_n, \underbrace{0, \ldots, 0}_{(n+1)/2}), \quad \delta_n^f(s^2, y^1) = u = (u_1, \ldots, u_n), \tag{23}$$

где $u = (u_1, \ldots, u_n)$ — некоторое фиксированное состояние. Подберём входное слово $y^2 \in \mathbb{F}_2^{(n-1)/2}$, что

$$\begin{gathered}\delta_n^f((s_{(n+3)/2}, \ldots, s_n, \underbrace{0, \ldots, 0}_{(n+1)/2}), y^2) = (\underbrace{0, \ldots, 0}_{(n+1)/2}, v_1, \ldots, v_{(n-1)/2}),\\ \delta_n^f(u, y^2) = (u_{(n+1)/2}, \ldots, u_n, \underbrace{0, \ldots, 0}_{(n-1)/2}),\end{gathered} \tag{24}$$

где $v = (v_1, \ldots, v_{(n-1)/2})$ — некоторый набор. Из соотношений (23) и (24) вытекает, что

$$\operatorname{wt}(\delta_n^f(s^1, y^1y^2)) \leqslant (n-1)/2, \quad \operatorname{wt}(\delta_n^f(s^2, y^1y^2)) \leqslant (n+1)/2.$$

Следовательно,

$$f(\delta_n^f(s^1, y^1y^2)) = f(\delta_n^f(s^2, y^1y^2)) = 0. \tag{25}$$

Для состояний $\delta_n^f(s^1, y^1y^2)$ и $\delta_n^f(s^2, y^1y^2)$ можно подобрать входное слово $y^3 = 0^n$ автомата $NFSR(f)$ так, чтобы

$$\delta_n^f(s^1, y^1y^2y^3) = \delta_n^f(\delta_n^f(s^1, y^1y^2), y^3) = \delta_n^f(s^2, y^1y^2y^3) = \delta_n^f(\delta_n^f(s^2, y^1y^2), y^3) = 0^n,$$

т. е. любые два состояния автомата $NFSR(f)$ подходящим входным словом переводятся в состояние 0^n. Следовательно, для данного автомата существует синхронизирующее слово и f обладает синхронизирующим свойством.

Предположим теперь, что $f \in \mathcal{FM}_n^+$ и для любого $x \in \min \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) < (n-1)/2$. Доказательство этого утверждения теоремы проводится аналогично рассмотренному выше с заменой состояния 0^n на 1^n.

2) Предположим, что $f \in \mathcal{FM}_n^-$ и для любого $x \in \max \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) < (n-1)/2$. Рассмотрим функцию $h'(x) = f(x \oplus 1^n)$. Очевидно, что $h' \in \mathcal{FM}_n^+$. Пусть для некоторого $x \in \mathbb{F}_2^n$ выполнено $h'(x) = 1$. Тогда $\operatorname{wt}(x) > (n+1)/2$. Действительно, если $\operatorname{wt}(x) \leqslant (n+1)/2$, то существует $y \in \mathbb{F}_2^n$, такой, что $\operatorname{wt}(y) \geqslant (n-1)/2$ и $f(y) = 1$.

Получаем противоречие. Следовательно, для любого $x \in \min \operatorname{supp}(h')$ имеем $\operatorname{wt}(x) > (n+1)/2$. Согласно п. 1 теоремы, h' обладает синхронизирующим свойством. Тогда по следствию 1 функция $f(x) = h'(x \oplus 1^n)$ тоже обладает синхронизирующим свойством.

Пусть теперь $f \in \mathcal{FM}_n^-$ и для любого $x \in \max \operatorname{supp}(f)$ выполнено $\operatorname{wt}(x) > (n+1)/2$. Рассмотрим функцию $g(x) = f(x) \oplus 1$. Ясно, что $g \in \mathcal{FM}_n^+$. Если $g(x) = 1$, то $\operatorname{wt}(x) > (n+1)/2$. Следовательно, для любых $x \in \min \operatorname{supp}(g)$ имеем $\operatorname{wt}(x) > (n+1)/2$. Согласно п. 1 теоремы, g обладает синхронизирующим свойством. Тогда по п. 2 леммы 2 функция $f(x) = 1 \oplus g(x)$ также обладает синхронизирующим свойством. ■

ЛИТЕРАТУРА

1. *Gecseq F. and Peak I.* Algebraic Theory of Automata. Budapest: Akademiai Kiado, 1972. 325 p.
2. *Кудрявцев В. Б., Алешин С. В., Подколзин А. В.* Введение в теорию автоматов. М.: Наука, 1985. 316 с.
3. *Логачев О. А., Проскурин Г. В., Ященко В. В.* Локальное обращение конечного автомата с помощью автоматов // Дискретная математика. 1995. Т. 7. Вып. 2. С. 19–33.
4. *Логачев О. А.* О локальной обратимости одного класса булевых отображений // Материалы IX Междунар. семинара «Дискретная математика и ее приложения». Москва, 18–23 июня 2007 г. М.: Изд-во мех.-мат. факультета МГУ, 2007. С. 440–442.
5. *Cerny J.* Poznamka k homogennym eksperimentoms konecnymi automatamy. Matematicko-Fizikalny Casopis Slovensk. Akad. Vied. 1964. V. 14. No. 3. P. 208–216. (in Slovak)
6. *Laemmel A. E. and Rudner B.* Study of the Applications of Coding Theory. Report PIBEP-69-034. Politechnic Inst. Brooklyn, N.Y., 1969. 94 p.
7. *Клосс Б. Б.* Некоторые свойства помехоустойчивых автоматов // Кибернетика. 1988. № 1. С. 10–15.
8. *Рысцов И. К.* Возвратные слова для разрешимых автоматов // Кибернетика и системный анализ. 1994. № 6. С. 21–26.
9. *Pin J.* On two combinatorial problems arising from automata theory // Ann. Discrete Math. 1983. V. 17. P. 535–548.
10. *Volkov M. V.* Synchronizing automata and the Cerny conjecture // LNCS. 2008. V. 5196. P. 11–27.
11. *Яблонский С. В.* Введение в дискретную математику. М.: Высшая школа, 2010. 384 с.

REFERENCES

1. *Gecseq F. and Peak I.* Algebraic Theory of Automata. Budapest, Akademiai Kiado, 1972. 325 p.
2. *Kudryavtsev V. B., Aleshin S. V., Podkolzin A. V.* Vvedenie v teoriyu avtomatov [Introduction to the Automata Theory]. Moscow, Nauka Publ., 1985. 316 p. (in Russian)
3. *Logachev O. A., Proskurin G. V., and Yashchenko V. V.* Lokal'noe obrashchenie konechnogo avtomata s pomoshch'yu avtomatov [Local inversion of a finite automaton by means of automata]. Diskr. Mat., 1995, vol. 7, iss. 2, pp. 19–33. (in Russian)
4. *Logachev O. A.* O lokal'noy obratimosti odnogo klassa bulevykh otobrazheniy [On the local invertibility of a class of Boolean maps]. Proc. IX Intern. Conf. "Discrete Mathematics and its Applications", Moscow, 2007, MSU Publ., pp. 440–442. (in Russian)
5. *Cerny J.* Poznamka k homogennym eksperimentoms konecnymi automatamy. Matematicko-Fizikalny Casopis Slovensk. Akad. Vied., 1964, vol. 14, no. 3, pp. 208–216. (in Slovak)

6. *Laemmel A. E. and Rudner B.* Study of the Applications of Coding Theory. Report PIBEP-69-034. Politechnic Inst. Brooklyn, N.Y., 1969. 94 p.
7. *Kloss B. B.* Nekotorye svoystva pomekhoustoychivykh avtomatov [Some properties of noise-immune automata]. Kibernetika, 1988, no. 1, pp. 10–15. (in Russian)
8. *Rystsov I. K.* Vozvratnye slova dlya razreshimykh avtomatov [Return words for solvable automata]. Kibernetika i Sistemnyy Analiz, 1994, no. 6, pp. 21–26. (in Russian)
9. *Pin J.* On two combinatorial problems arising from automata theory. Ann. Discrete Math., 1983, vol. 17, pp. 535–548.
10. *Volkov M. V.* Synchronizing automata and the Cerny conjecture. LNCS, 2008, vol. 5196, pp. 11–27.
11. *Yablonskiy S. V.* Vvedenie v diskretnuyu matematiku [Introduction to Discrete Mathematics]. Moscow, Vysshaya Shkola Publ., 2010. 384 p. (in Russian)

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

УДК 510.52

О СЛОЖНОСТИ ПРОБЛЕМЫ РАЗРЕШИМОСТИ СИСТЕМ УРАВНЕНИЙ НАД КОНЕЧНЫМИ ЧАСТИЧНЫМИ ПОРЯДКАМИ[1]

А. Ю. Никитин, А. Н. Рыбалов

Омский государственный технический университет, г. Омск, Россия

Классическая алгебраическая геометрия изучает множества решений алгебраических уравнений над полями вещественных и комплексных чисел. В рамках вычислительной алгебраической геометрии предложены алгоритмы (например, алгоритм Бухбергера) для решения систем полиномиальных уравнений над этими полями. Универсальная алгебраическая геометрия изучает системы уравнений над произвольными алгебраическими системами. Под уравнениями понимаются атомарные формулы языка алгебраической системы. В работе рассматривается проблема распознавания разрешимости систем уравнений над конечными частично упорядоченными множествами. Доказывается, что эта проблема является NP-полной в случае, когда проверяется существование решения, состоящего из попарно различных элементов частично упорядоченного множества.

Ключевые слова: *частично упорядоченное множество, системы уравнений, NP-полнота.*

DOI 10.17223/20710410/39/8

ON COMPLEXITY OF THE SATISFIABILITY PROBLEM OF SYSTEMS OVER FINITE POSETS

A. Y. Nikitin, A. N. Rybalov

Omsk State Technical University, Omsk, Russia

E-mail: nikitinlexey@gmail.com, alexander.rybalov@gmail.com

Classical algebraic geometry studies sets of solutions of algebraic equations over the fields of real and complex numbers. In the frameworks of computational algebraic geometry, several algorithms were proposed (for example, the Buchberger algorithm) for solving systems of polynomial equations over these fields. Universal algebraic geometry studies systems of equations over arbitrary algebraic structures. The equations are atomic formulas in the language of an algebraic structure. In this article, we consider the problem of recognizing the solvability of systems of equations over finite partially ordered sets. We prove that this problem is NP-complete in the case when we seek a solution consisting of pairwise distinct elements of a finite partially ordered set.

Keywords: *partially ordered sets, systems of equations, NP-completeness.*

[1]Работа поддержана грантом РНФ № 17-11-01117.

Введение

Решение уравнений и систем уравнений над вещественными, комплексными, рациональными, целыми числами является классической темой исследований в различных областях математики в течение уже нескольких тысяч лет. Классическая алгебраическая геометрия изучает множества решений алгебраических уравнений над полями вещественных и комплексных чисел. В рамках диофантовой геометрии и диофантова анализа изучаются решения алгебраических уравнений над целыми и рациональными числами. В XX веке большую роль начали играть вычислительные аспекты этих теорий. Изучение алгоритмических проблем, связанных с определением наличия решения у систем уравнений, а также с нахождением и описанием множества решений, является темой многочисленных теоретических и практических исследований. Как правило, эти алгоритмические проблемы оказываются либо неразрешимыми, либо разрешимыми, но вычислительно трудными. Например, проблема решения полиномиальных уравнений над кольцом целых чисел неразрешима [1]. Известные на сегодняшний день алгоритмы для решения систем полиномиальных уравнений над полями комплексных и вещественных чисел имеют дважды экспоненциальную сложность [2]. Проблема решения систем полиномиальных уравнений над конечными полями, эквивалентная проблеме выполнимости булевых формул, является NP-полной [3].

В последние 20 лет активно развивается так называемая универсальная алгебраическая геометрия (см., например, [4]), в которой изучаются системы уравнений над произвольными алгебраическими системами. Под уравнениями понимаются атомарные формулы языка алгебраической системы. Многие понятия и алгоритмы классической алгебраической геометрии переносятся на произвольные алгебраические системы. При этом возникают новые интересные вопросы. В данной работе в рамках универсальной алгебраической геометрии рассматривается проблема распознавания разрешимости систем уравнений над конечными частично упорядоченными множествами. Доказывается, что эта проблема является NP-полной в случае, когда проверяется существование решения, состоящего из попарно различных элементов частично упорядоченного множества.

1. Предварительные сведения

Напомним, что *частично упорядоченным множеством* (*частичным порядком*) называется алгебраическая система $\mathcal{P} = \langle P| \leqslant^{(2)}, A\rangle$, где $\leqslant$ — предикатный символ отношения порядка и A — множество константных символов, на которой выполнены следующие аксиомы:

1) $\forall p \in P\ (p \leqslant p)$;
2) $\forall p_1, p_2 \in P\ ((p_1 \leqslant p_2 \wedge p_2 \leqslant p_1) \Rightarrow p_1 = p_2)$;
3) $\forall p_1, p_2, p_3 \in P\ ((p_1 \leqslant p_2 \wedge p_2 \leqslant p_3) \Rightarrow p_1 \leqslant p_3)$.

Уравнение над частичным порядком $\mathcal{P}$ от переменных X — это выражение одного из следующих типов:

1) $a_i = a_j$, где $a_i, a_j \in \mathcal{P}$;
2) $a_i \leqslant a_j$, где $a_i, a_j \in \mathcal{P}$;
3) $x_i = a_j$, где $x_i \in X$, $a_j \in \mathcal{P}$;
4) $x_i = x_j$, где $x_i, x_j \in X$;
5) $a_i \leqslant x_j$, где $a_i \in \mathcal{P}$, $x_j \in X$;
6) $x_i \leqslant a_j$, где $x_i \in X$, $a_j \in \mathcal{P}$;
7) $x_i \leqslant x_j$, где $x_i, x_j \in X$.

Система уравнений $S(X)$ от переменных $X = \{x_1, \ldots, x_n\}$ — это любое множество уравнений от переменных X. Точка $p = (p_1, \ldots, p_n) \in P^n$ называется *решением* системы уравнений S, если для любого уравнения системы S при подстановке вместо переменных соответствующих значений получается истинное над $\mathcal{P}$ выражение (равенство или неравенство). Две системы уравнений S_1 и S_2 называются *эквивалентными* над $\mathcal{P}$, если их множества решений совпадают. В данной работе будем рассматривать только конечные системы уравнений, то есть системы, состоящие из конечного числа уравнений.

Легко видеть, что любая система уравнений над частичным порядком либо не имеет решения, либо для неё можно построить эквивалентную систему, в которой присутствуют только уравнения типов 5, 6, 7. Действительно, уравнения первых двух типов либо тождественно истинны над данным частичным порядком (тогда от них можно избавиться, упростив систему), либо какое-то из них ложно (тогда система не имеет решения). От уравнений третьего вида можно избавиться, заменяя для каждого такого уравнения $x_i = a_j$ каждое вхождение переменной x_i в системе на константу a_j. После всех таких замен могут возникнуть уравнения первых двух типов — от них нужно избавиться описанным выше способом.

Напомним некоторые сведения из теории графов. Граф G задается парой (V, E), где V — множество вершин; $E \subseteq V \times V$ — множество рёбер графа G. Для любого конечного множества A через $|A|$ будем обозначать его мощность. Два графа $G = (V_1, E_1)$ и $H = (V_2, E_2)$ называются *изоморфными* (обозначается $G \cong H$), если $|V_1| = |V_2|$, $|E_1| = |E_2|$ и существует взаимно однозначная функция $f : V_1 \to V_2$, такая, что $(u, v) \in E_1$ тогда и только тогда, когда $(f(u), f(v)) \in E_2$. Проблема изоморфизма подграфу состоит в следующем. Заданы два графа $G = (V, E)$ и $H = (V', E')$. Требуется определить, существует ли подграф $G_0 = (V_0, E_0)$, где $V_0 \subseteq V$ и $E_0 \subseteq E$, такой, что $G_0 \cong H$. Известно [5], что эта проблема является NP-полной. Более того, эта проблема остаётся NP-полной в классе ациклических ориентированных графов [6]. Напомним, что ориентированный граф называется *ациклическим*, если в нём отсутствуют направленные циклы — пути, начинающиеся и заканчивающиеся в одной и той же вершине. Далее нас будет интересовать эта проблема для ещё более узкого класса графов — транзитивно замкнутых. Ориентированный граф называется *транзитивно замкнутым*, если для любых его трёх вершин x, y, z наличие дуг из x в y и из y в z влечёт наличие дуги из x в z.

Лемма 1. Проблема изоморфизма подграфу в классе транзитивно замкнутых графов является NP-полной.

Доказательство. Пусть дан простой граф G. Опишем процедуру трансформации графа G в ориентированный ациклический граф G^t. На первом шаге присвоим каждому ребру (v_1, v_2) графа G уникальную метку $m(v_1, v_2)$. На втором шаге для каждого ребра (v_1, v_2) графа G делаем следующее: ребро (v_1, v_2) удаляем из графа G и вместо него добавляем вершину $m(v_1, v_2)$, затем проводим две дуги (v_1, m) и (v_2, m). Очевидно, что получившийся граф G^t является ациклическим, так как из всех старых (соответствующих вершинам графа G) вершин дуги выходят, а в новые вершины (соответствующие дугам в графе G) — входят. Кроме того, граф G^t является транзитивно замкнутым. Описанный алгоритм построения графа G^t по графу G является полиномиальным от размера графа G, так как число вершин графа G^t равно сумме числа рёбер и числа вершин графа G.

Пусть есть вход (G, H) проблемы изоморфизма подграфу для простых графов. По описанному алгоритму получим пару транзитивно замкнутых ациклических ориентированных графов (G^t, H^t). Легко видеть, что в графе G^t найдется подграф, изоморфный H^t, тогда и только тогда, когда в G найдется подграф, изоморфный H. Таким образом, по входу проблемы изоморфизма подграфу для простых графов строится вход для проблемы изоморфизма подграфу в классе ориентированных ациклических графов. Следовательно, и для класса транзитивно замкнутых ациклических ориентированных графов проблема изоморфизма подграфу NP-полна. ■

2. Основной результат

Естественным образом можно перейти от частичного порядка к ациклическому ориентированному графу [7]. Пусть дан частичный порядок $\mathcal{P}$, построим по нему граф $\Gamma_{\mathcal{P}}$. Множеством вершин графа $\Gamma_{\mathcal{P}}$ будет базовое множество частичного порядка $\mathcal{P}$. В графе $\Gamma_{\mathcal{P}}$ присутствует дуга от v_i к v_j, если над $\mathcal{P}$ верно $v_j \leqslant v_i$. При рассмотрении алгоритмических проблем вместо частичного порядка $\mathcal{P}$ будем подавать на вход алгоритма матрицу смежности графа $\Gamma_{\mathcal{P}}$.

Проблема разрешимости систем уравнений над конечными частичными порядками состоит в следующем. Заданы конечный частичный порядок $\mathcal{P}$ и конечная система уравнений $S(X)$ над $\mathcal{P}$. Требуется определить, существует ли решение $p = (p_1, \ldots, p_n)$ системы $S(X)$ над $\mathcal{P}$ с попарно различными p_i, $i = 1, \ldots, n$.

Теорема 1. Проблема разрешимости систем уравнений над конечными частичными порядками NP-полна.

Доказательство. Доказательство заключается в сведении проблемы изоморфизма подграфу в классе транзитивно замкнутых ориентированных графов к проблеме разрешимости систем уравнений над конечными частичными порядками.

Пусть заданы два транзитивно замкнутых ориентированных графа G и H и для этих графов нужно определить, существует ли в графе G подграф G_0, изоморфный графу H. Поскольку граф G транзитивно замкнут и ацикличен, ему можно сопоставить частичный порядок $\mathcal{P}$ таким образом, что вершины графа G будут элементами базового множества $\mathcal{P}$, а дуги задают отношение порядка $\leqslant$. Пусть также для каждого элемента $p \in \mathcal{P}$ выполняется $p \leqslant p$.

По графу H строится система $S(X)$ так, что переменным X соответствуют вершины графа H (положим, что их n штук), а уравнения определяются дугами, т. е. если в графе H есть дуга (x_i, x_j), то в системе $S(X)$ содержится уравнение $x_j \leqslant x_i$.

Докажем, что множество изоморфизмов между графом H и всевозможными подграфами G находится во взаимно однозначном соответствии с множеством решений системы уравнений $S(X)$ над $\mathcal{P}$ при условии, что $x_i \neq x_j$, $i, j \in \{1, \ldots, n\}$, $i \neq j$. Пусть задано вложение графов $\alpha : H \to G$. Рассмотрим пару вершин $h_1, h_2 \in H$. Тогда $\alpha(h_i) = g_{\alpha_i}$, $i = 1, 2$. Если в H есть ребро (h_1, h_2), то в G также есть ребро $(g_{\alpha_1}, g_{\alpha_2})$. При описанном сведении графов к системе уравнений и частичному порядку вершинам h_1 и h_2 соответствуют переменные x_1 и x_2, вершинам $g_{\alpha_1}, g_{\alpha_2}$ — элементы базового множества частичного порядка p_{α_1} и p_{α_2}. При этом для ребра (h_1, h_2) верно, что в системе $S(X)$ присутствует уравнение $x_2 \leqslant x_1$, а на частичном порядке $\mathcal{P}$ определено отношение $p_{\alpha_2} \leqslant p_{\alpha_1}$. Следовательно, для вложения α существует такое решение системы уравнений, что $x_i = p_{\alpha_i}$, $i = 1, 2$.

Обратно, пусть дано решение системы уравнений $S(X)$ над $\mathcal{P}$. Рассмотрим уравнение $x_1 \leqslant x_2$ из $S(X)$. Пусть в этом решении $x_i = p_{\alpha_i}$, $i = 1, 2$. Уравнению $x_1 \leqslant x_2$ соответствует ребро (h_2, h_1) в графе H для x_1, x_2 и соответствующих вершин h_1, h_2.

Отношению порядка на элементах базового множества $p_{\alpha_1} \leqslant p_{\alpha_2}$ соответствует ребро $(g_{\alpha_2}, g_{\alpha_1})$ в графе G.

Итак, проблема изоморфизма графа H подграфу G для класса транзитивно замкнутых ориентированных графов сведена к проблеме разрешимости системы уравнений $S(X)$ над частичным порядком $\mathcal{P}$. Сведение является полиномиальным по размеру матриц смежности графов G и H. Действительно, представление частичного порядка $\Gamma_{\mathcal{P}}$ совпадает с матрицей смежности графа G. Процедура построения системы $S(X)$ по графу H выполняется за линейное время, так как число уравнений в $S(X)$ равно числу рёбер графа H.

Таким образом, проблема разрешимости систем уравнений над конечными частичными порядками является NP-полной. ■

Авторы выражают благодарность рецензенту за полезные замечания и предложения по улучшению текста статьи.

ЛИТЕРАТУРА

1. *Матиясевич Ю. В.* Диофантовость перечислимых множеств // Доклады Академии наук СССР. 1970. Т. 191. № 2. С. 279–282.
2. *Mayr E. W. and Meyer A. R.* The complexity of the word problems for commutative semigroups and polynomial ideals // Adv. Math. 1982. V. 46 (3). P. 305–329.
3. *Cook S. A.* The complexity of theorem proving procedures // Proc. 3d Ann. ACM Symp. Theory of Computing. N.Y., USA, 1971. P. 151–158.
4. *Даниярова Э. Ю., Мясников А. Г., Ремесленников В. Н.* Алгебраическая геометрия над алгебраическими системами. Новосибирск: СО РАН, 2016. 288 с.
5. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 419 с.
6. *Werth T., Wörlein M., Dreweke A., et al.* DAG mining for code compaction / eds. L. Cao, P. S. Yu, C. Zhang, and H. Zhang. Data Mining for Business Applications. Boston, MA: Springer, 2009. P. 209–223.
7. *Оре О.* Теория графов. М.: Наука, 1980. 336 с.

REFERENCES

1. *Matiyasevich Yu. V.* Diofantovost' perechislimykh mnozhestv [Diophantineity of enumerable sets]. Doklady Akademii Nauk USSR, 1970, vol. 191, no. 2, pp. 279–282. (in Russian)
2. *Mayr E. W. and Meyer A. R.* The complexity of the word problems for commutative semigroups and polynomial ideals. Adv. Math., 1982, vol. 46 (3), pp. 305–329.
3. *Cook S. A.* The complexity of theorem proving procedures. Proc. 3d Ann. ACM Symp. Theory of Computing, N.Y., USA, 1971, pp. 151–158.
4. *Daniyarova E. Yu., Myasnikov A. G., and Remeslennikov V. N.* Algebraicheskaya geometriya nad algebraicheskimi sistemami [Algebraic Geometry over Algebraic Structures]. Novosibirsk, SB RAS, 2016. 288 p (in Russian).
5. *Garey M. and Johnson D.* Computers and Intractability. N. Y., Freeman & Co, 1979. 340 p.
6. *Werth T., Wörlein M., Dreweke A., et al.* DAG mining for code compaction. In: Cao L., Yu P. S., Zhang C., and Zhang H. (eds). Data Mining for Business Applications. Boston, MA, Springer, 2009, pp. 209–223.
7. *Ore O.* Theory of Graphs. Amer. Math. Soc., 1962. 270 p.

УДК 004.43+004.422.636

ИНТЕРВАЛЫ ИНДЕКСОВ В ЛЯПАСЕ[1]

А. А. Песняк, Д. А. Стефанцов

Национальный исследовательский Томский государственный университет, г. Томск, Россия

Разработаны новые типы данных для работы с частями комплекса. Добавлены обозначения для этих типов данных, операции над ними, расположение этих объектов в памяти программы. Приведены примеры использования реализации абстрактных типов данных в языке ЛЯПАС.

Ключевые слова: *язык программирования ЛЯПАС, операции над комплексами, абстрактные типы данных.*

DOI 10.17223/20710410/39/9

INDEX INTERVALS IN LYAPAS

A. A. Pesnyak, D. A. Stefantsov

National Research Tomsk State University, Tomsk, Russia

E-mail: a.pesnyak92@mail.ru, d.a.stephantsov@gmail.com

The new data type called index interval is proposed for LYaPAS along with the operations defined on members of that type. It enables isolated work on parts of members of another LYaPAS data type, namely the contiguous sequence of homogeneous elements called complex. The syntax for describing the members of the new data type is introduced along with that for the operations on its members. As the implementation detail, the memory layout is discussed. Finally, implementations of some well-known algorithms using index intervals in LYaPAS are given, such as Karatsuba multiplication, quick sort, and the lookup operation in a binary search tree.

Keywords: *LYaPAS programming language, operations on complexes, abstract data type.*

Введение

В языке ЛЯПАС используются только такие типы данных, как комплексы и переменные [1]. В существующей реализации языка ЛЯПАС проблематичным является процесс задания поддиапазонов комплекса. В некоторых типах задач оптимальной является возможность выделения только части комплекса и последующей работы с ней. Например, такими задачами являются сортировка Хоара, умножение чисел методом Карацубы. Вместо того чтобы подавать весь комплекс на вход функции, достаточно отдать только его часть, необходимую для сортировки или умножения на данной итерации. Эта проблема особенно остро возникает при реализации абстрактных типов данных, использующих указатели. Необходимость их использования связана с применением абстрактных типов данных, таких, как списки и деревья, в множестве существующих программ и форматов хранения данных, например в файловой системе

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

ext4. Чтение данных с томов ext4 является одной из задач на первых этапах развития ОС ЛЯПАС.

Альтернативой указателям являются индексы элементов комплекса, хранимые в элементах этого комплекса, что не отменяет некоторые проблемы. Так, например, проблемой безопасности является явная манипуляция индексами комплекса. Имея доступ на чтение и запись к любой ячейке комплекса, программист имеет возможность изменять индексы, тем самым получая доступ к другим участкам комплекса или даже меняя логику работы программы. Невозможно ограничить доступ к части комплекса и нецелесообразно использовать эту технику в символьных комплексах, так как размер их элементов составляет один байт, а следовательно, значение индекса не может быть больше 255, что слишком мало для практических задач.

В данной работе рассматривается создание новых типов данных языка ЛЯПАС — *интервалов* и *комплексов ссылок*. Описаны операции над ними, а также представление этих типов данных в памяти прикладной программы. Приведены примеры применения интервалов и комплексов ссылок.

1. Интервалы в ЛЯПАСе

1.1. Определения, правила, обозначения

Комплекс — структура данных в виде набора равных по размеру элементов, расположенных в памяти непосредственно друг за другом. Комплекс имеет потенциальный и текущий размеры, которые называются *ёмкостью* и *мощностью* соответственно. Обращение к элементу за пределами комплекса приводит к аварийному завершению программы.

Интервал — множество элементов логического или символьного комплекса, ограниченное двумя элементами комплекса — началом и концом интервала. У интервала, как и у комплекса, есть такие параметры, как ёмкость и мощность.

Таблица идентификаторов — служебная таблица, в которой хранится информация об интервалах заданного комплекса: индекс начала интервала, индекс конца интервала, мощность и ёмкость интервала, комплекс ссылок интервала.

Идентификатор (*ссылка*) — четырёхбайтовое значение, описывающее адрес некоторой строки таблицы идентификаторов.

Комплекс ссылок — комплекс, элементами которого являются идентификаторы.

Каждому интервалу соответствует свой комплекс ссылок, элементы этого комплекса являются идентификаторами интервалов, на которые ссылается данный интервал. К элементам комплекса ссылок запрещается применение операций присваивания и считывания.

Во всех приведённых ниже наименованиях символы i, j, k около комплексов или интервалов являются константами и составляют часть имени комплекса (интервала).

Для введённых определений используются следующие обозначения:

N_j — интервал с номером j;

W_j — мощность интервала N_j;

Y_j — ёмкость интервала N_j;

M_j — комплекс ссылок интервала N_j;

T_j — мощность комплекса ссылок M_j;

V_j — ёмкость комплекса ссылок M_j.

1.2. Операции над интервалами и комплексами ссылок

Создание интервала

`@+(x,y,Li/Nj)`

В результате выполнения данной операции в логическом комплексе L_i из элементов, начиная с элемента с индексом x и заканчивая элементом с индексом $y-1$, будет создан интервал N_j мощности и ёмкости $y-x$. В таблицу идентификаторов добавится соответствующая запись о созданном интервале.

Сохранение идентификатора

`@↓(i,Nj/Mk)`

Операция записывает адрес, соответствующий интервалу N_j в таблице идентификаторов, в элемент по индексу i комплекса ссылок M_k.

Восстановление по идентификатору

`@+(i,Mj/Nk)`

Операция восстанавливает сведения об интервале на основании идентификатора, находящегося в элементе с индексом i комплекса M_j, и ставит им в соответствие псевдоним N_k.

Обнуление идентфикатора

`OMi.j`

В результате выполнения данной команды в комплексе ссылок M_i в элементе с индексом j будет записано `null`-значение.

Переход по идентификатору

`↑(id?Mj.i)q`

Команда выполняет проверку элемента с индексом i комплекса ссылок M_j на соответствие идентификатору: если в таблице идентификаторов по этому адресу есть запись, то выполняется переход на параграф q.

Переход по пустому идентификатору

`↑(null?Mj.i)q`

Операция выполняет проверку элемента с индексом i комплекса ссылок M_j: если этот элемент равен `null` или в таблице идентификаторов по этому адресу отсутствует запись, то выполняется переход на параграф q.

Присваивание

`@+(Nj/Nk)`

В результате выполнения данной операции псевдониму N_k будет соответствовать интервал, соответствующий псевдониму N_j.

2. Использование и реализация

2.1. Примеры решения задач с помощью интервалов

Поддиапазоны

Интервалы позволяют выделять нужные части комплекса для последующей обработки. Это может потребоваться в реализации сортировки Хоара (листинг 1).

```
1  Hoare(N1/N1)
2       Oi  W1-1⇒j>1⇒r  N1r⇒x
```

```
3  §1  ↑(i>j)5
4  §2  ↑(N1i⩾x)3  Δ i  →2
5  §3  ↑(N1j⩽x)4  ▽ j  →3
6  §4  ↑(i⩾j)10  ⇔(N1ij)  →1
7  §5  ↑(i⩾W1)6  @+(i,W1,N1/N2)  *Hoare(N2/N2)
8  §6  j↪10  @+(0,j,N1/N2)  *Hoare(N2/N2)
9  §10  **
```

Листинг 1. Подпрограмма сортировки Хоара

На рис. 1 и в листинге 2 приведён пример реализации умножения больших чисел методом Карацубы [2]. Пусть N_{11}, N_{22} — множители одного размера ($W_{11} = W_{22}$), интервал N_{33} выделен на всей длине логического комплекса L_3 необходимого размера При мощности комплекса меньше 50 умножение выполняется с помощью стандартной операции языка ЛЯПАС-Т [3], так как для таких чисел умножение методом Карацубы работает медленнее. Комплекс L_4 является временным.

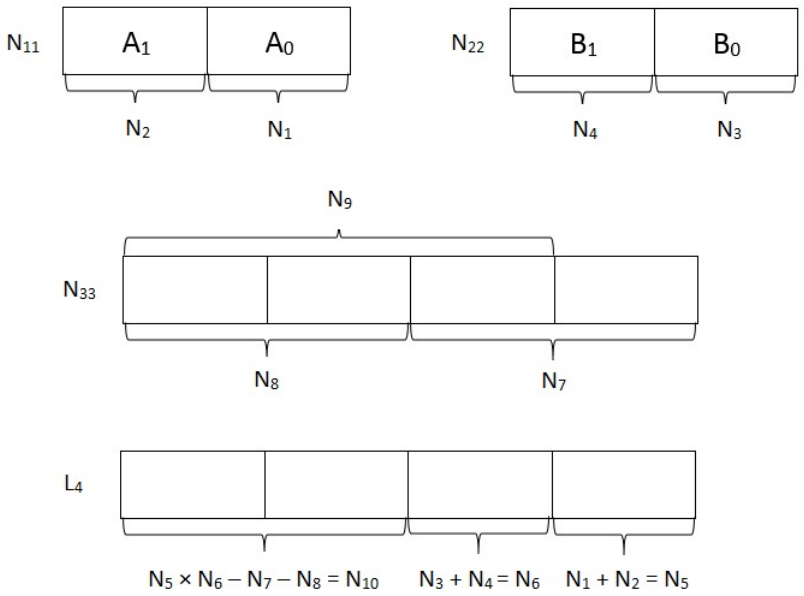


Рис. 1. Расположение интервалов в комплексах для перемножения Карацубы

```
1  Karatsuba(N11,N22/N33)
2      W11>1⇒n<1⇒t<1⇒f  ↑(n<25)1
3      @+(0,n,N11/N1)  @+(n,Q1,N11/N2)
4      @+(0,n,N22/N3)  @+(n,Q2,N22/N4)
5      @+(0,t,N33/N7)  @+(t,f,N33/N8)  @+(n,f,N33/N9)
6      f+4⇒s  @+L4(s)  ***  tmp
7      n+1⇒a<1⇒b
8      @+(0,a,L4/N5)  @+(a,b,L4/N6)  @+(b,s,L4/N10)
9      N1+N2⇒N5  N3+N4⇒N6
10     *Karatsuba(N1,N3/N7)
11     *Karatsuba(N2,N4/N8)
12     *Karatsuba(N5,N6/N10)
```

```
13        N10-N7-N8+N9⇒N9
14        @-L4  →2
15  §1    N11*N22⇒N33
16  §2    **
```

Листинг 2. Подпрограмма умножения Карацубы

Списки

С помощью интервалов можно реализовать списки. В качестве примера выбрано суммирование элементов списка, каждый из которых состоит из некоторого числового значения и ссылки на следующий элемент (рис. 2, листинг 3).

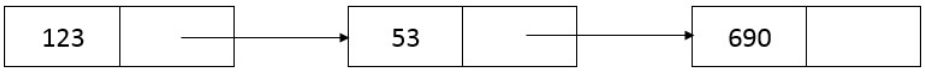


Рис. 2. Пример односвязного списка

```
1  sum_list(N1/s)
2       0s
3  §1   s+N1.0⇒s
4       ↑(null?M1.0)2
5       @+(0,M1/N1)  →1
6  §2  **
```

Листинг 3. Подпрограмма суммирования элементов списка

В данной реализации комплекс ссылок и интервал состоят из одного элемента — ссылки на следующий элемент списка и числового значения соответственно.

Деревья

Интервалы можно использовать для создания такой иерархической структуры данных, как деревья. Приведём пример программы поиска значения в двоичном дереве поиска; пример дерева представлен на рис. 3, его представление в комплексе — на рис. 4.

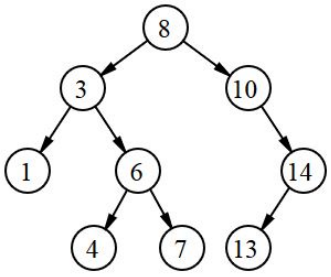


Рис. 3. Пример двоичного дерева поиска

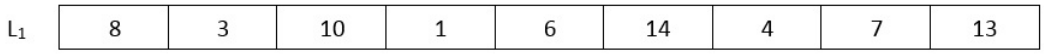


Рис. 4. Расположение дерева в комплексе

Каждому интервалу соответствует комплекс ссылок, содержащий два идентификатора — ссылки на левое и правое поддеревья. Связь комплексов между собой приведена на рис. 5, код программы — в листинге 4.

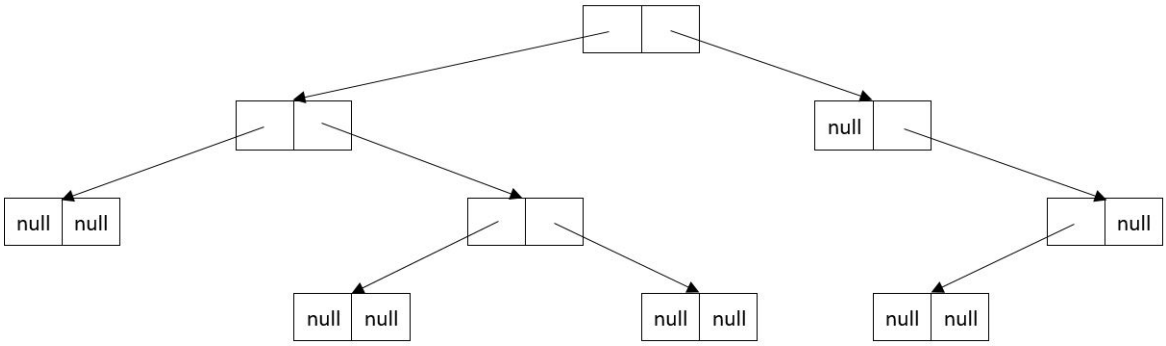


Рис. 5. Связь комплексов ссылок для двоичного дерева поиска

Входными параметрами подпрограммы являются ключ и интервал, соответствующий корню дерева. В случае успешного нахождения ключа в дереве поиска в нулевой элемент комплекса ссылок M_2, соответствующего выходному интервалу N_2, будет помещён идентификатор интервала, содержащего ключ; в противном случае — `null`-значение.

```
 1  search_BSTree(k,N1/N2)
 2      OM2.0
 3  §1  ↑(k<N1.0)2
 4      ↑(k>N1.0)3
 5      ↑(k=N1.0)4
 6  §2  ↑(null?M1.0)5
 7      @+(0,M1/N1) →1
 8  §3  ↑(null?M1.1)5
 9      @+(1,M1/N1) →1
10  §4  @↓(0,N1/M2)
11  §5  **
```

Листинг 4. Подпрограмма поиска ключа в бинарном дереве поиска

2.2. Представление в памяти

Комплексу, на котором определены интервалы, соответствует единственная таблица идентификаторов, имеющая формат табл. 1. Здесь и далее $a_1, a_2, a_3 \ldots$ — некоторые адреса таблицы интервалов при выполнении прикладной программы.

Таблица 1

Адрес	Начало интервала	Конец интервала	Мощность	Ёмкость	Комплекс ссылок
a_1	$i_{1,1}$	$i_{1,2}$	W_1	Y_1	M_1
a_2	$i_{2,1}$	$i_{2,2}$	W_2	Y_2	M_2

Подпрограмме листинга 4 соответствует следующая таблица интервалов (табл. 2):

Таблица 2

Адрес	Начало интервала	Конец интервала	Мощность	Ёмкость	Комплекс ссылок
a_1	0	1	1	1	M_1
a_2	1	2	1	1	M_2
a_3	2	3	1	1	M_3
a_4	3	4	1	1	M_4
a_5	4	5	1	1	M_5
a_6	5	6	1	1	M_6
a_7	6	7	1	1	M_7
a_8	7	8	1	1	M_8
a_9	8	9	1	1	M_9

Взаимосвязь комплексов ссылок для данного примера представлена на рис. 6.

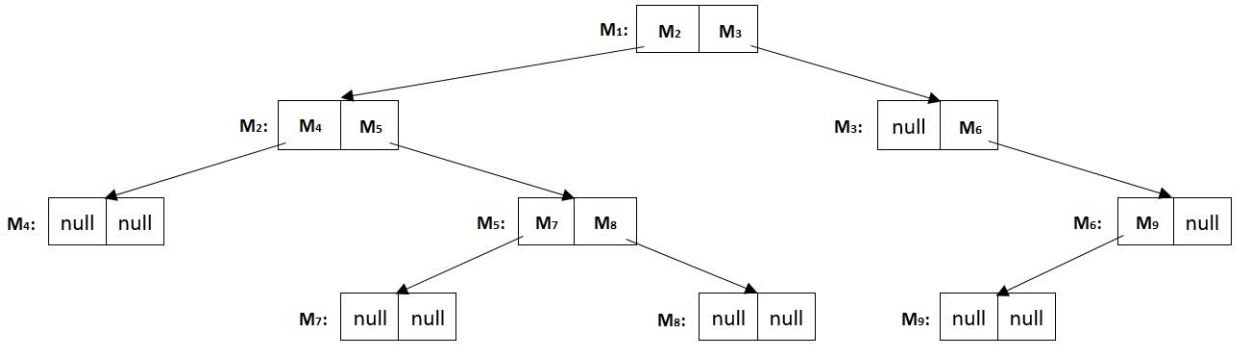


Рис. 6. Связь комплексов ссылок для двоичного дерева поиска

2.3. Дополнительные изменения в языке ЛЯПАС

В некоторых случаях программисту требуется работа со всем комплексом, как с интервалом. Для этого предложен механизм, позволяющий произвести данную операцию — интервал будет создан на всём комплексе и будет содержать все его элементы.

В целях удобства работы с интервалами предлагается ввести следующие дополнительные операции:

Распаковка в переменные

```
@+(i,Nk/x,y,z)
```

Операция производит запись элементов интервала N_k, начиная с элемента с индексом i, в переменные, указанные в качестве выходных параметров. Их количество произвольно и задаётся программистом путём перечисления переменных. В данном примере будут считаны три элемента из интервала N_k и последовательно записаны в переменные x, y, z.

Распаковка в аргументы подпрограммы

```
*func(x,y,@+(i,j,Lk/*)/a)
```

В данном примере созданный интервал над комплексом L_k будет распакован на отдельные входные переменные подпрограммы `func`.

Упаковка из переменных

```
@+(x,y,z/Nk)
```

В результате выполнения данной операции будет создан интервал N_k, содержащий значения всех входных параметров. Как и в операции распаковки в переменные, количество входных переменных произвольно.

Упаковка из выходных значений подпрограммы

```
*proc(x,y/@+(*/Ni))
```

Данная команда создаёт интервал N_i из выходных значений процедуры `proc`.

В целях стандартизации языковых конструкций предлагается заменить операцию создания комплекса `@+Li(n)` [4] на `@+(n/Li)`.

Заключение

В языке ЛЯПАС разработаны новые типы данных, позволяющие работать с частями комплекса — интервалы и комплексы ссылок. Описаны обозначения для этих типов данных, операции над ними, расположение этих объектов в памяти программы. Рассмотрены примеры использования интервалов для описания абстрактных типов данных.

ЛИТЕРАТУРА

1. *Торопов Н. Р.* Язык программирования ЛЯПАС // Прикладная дискретная математика. 2009. № 2. С. 9–25.
2. *Кнут Д.* Искусство программирования. Т. 2. Получисленные алгоритмы. 3-е изд. М.: Изд. дом «Вильямс», 2001. 832 с.
3. *Грибанов А. С., Сибирякова В. А.* Программная реализация операций над большими числами в языке ЛЯПАС-Т // Прикладная дискретная математика. Приложение. 2014. № 7. С. 146–148.
4. *Агибалов Г. П., Липский В. Б., Панкратова И. А.* О криптографическом расширении и его реализации для русского языка программирования // Прикладная дискретная математика. Приложение. 2013. № 3. С. 93–104.

REFERENCES

1. *Toropov N. R.* Yazik programmirovaniya LYaPAS [Programming language LYaPAS]. Prikladnaya Diskretnaya Matematika, 2009, no. 2, pp. 9–25. (in Russian)
2. *Knuth D. E.* The Art of Computer Programming, vol. 2. Seminumerical Algorithms, Third Ed. Reading, Massachusetts, Addison-Wesley, 1997.
3. *Gribanov A. S. and Sibiryakova V. A.* Programmnaya realizatsiya operatsiy nad bolshimi chislami v yazike LYaPAS-T [Software implementation of operations over large numbers in LYaPAS-T]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2014, no. 7, pp. 146–148. (in Russian)
4. *Agibalov G. P., Lipskiy V. B., and Pankratova I. A.* O kriptograficheskom rasshirenii i ego realizatsii dlya russkogo yazyka programmirovaniya [Cryptographic extension and its implementation for Russian programming language]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2013, no. 3, pp. 93–104. (in Russian)

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 512.55

О СХОДИМОСТИ НОВОГО АЛГОРИТМА ХАРАКТЕРИЗАЦИИ k-ЗНАЧНЫХ ПОРОГОВЫХ ФУНКЦИЙ

А. В. Бурделев

Белорусский государственный университет, г. Минск, Беларусь

Функция k-значной логики $f(x_1,\ldots,x_n)$, для которой существует линейная форма $L(x_1,\ldots,x_n)=a_1x_1+a_2x_2+\ldots+a_nx_n$, $x_i\in\mathbb{Z}_k$, с вещественными коэффициентами и набор вещественных порогов $b_0<b_1<\ldots<b_k$, такие, что для всех $i\in\in\{0,\ldots,k-1\}$ выполняется условие $f(x_1,\ldots,x_n)=i\Leftrightarrow b_i\leqslant L(x_1,\ldots,x_n)<b_{i+1}$, называется пороговой k-значной функцией. Под алгоритмом характеризации пороговой k-значной функции понимается процедура нахождения коэффициентов $a_1,a_2,\ldots,a_n$ линейной формы $L(x_1,\ldots,x_n)$ и множества порогов $b_0,b_1,\ldots,b_{k-1}$. В работе доказывается сходимость алгоритма нахождения коэффициентов линейной формы и порогов (характеризации) k-значных пороговых функций по столбцу значений. Основная идея алгоритма заключается в раздельном последовательном вычислении коэффициентов линейной формы и порогов. В качестве первичной аппроксимации линейной формы используются коэффициенты роста либо коэффициенты возрастания и итеративно осуществляется корректировка линейной формы. После нахождения коэффициентов линейной формы вычисляются разделяющие пороги.

Ключевые слова: *алгоритм характеризации, доказательство сходимости, пороговая функция.*

DOI 10.17223/20710410/39/10

CONVERGENCE OF AN ITERATIVE ALGORITHM FOR COMPUTING PARAMETERS OF MULTI-VALUED THRESHOLD FUNCTIONS

A. V. Burdelev

Belarus State University, Minsk, Belarus

E-mail: aburd2011@mail.ru

A k-valued threshold function is defined as $f(x_1,\ldots,x_n)=i\in\{0,1,\ldots,k-1\}\Leftrightarrow\Leftrightarrow b_i\leqslant L(x_1,\ldots,x_n)<b_{i+1}$ where $L(x_1,\ldots,x_n)=a_1x_1+a_2x_2+\ldots+a_nx_n$ is a linear form in variables $x_1,\ldots,x_n$ with the values in $\{0,1,\ldots,k-1\}$ and coefficients $a_1,\ldots,a_n$ in $\mathbb{R}$ and $b_0,\ldots,b_k$ are some thresholds for L in $\mathbb{R}$, $b_0<b_1<\ldots<b_k$. A. V. Burdelev and V. G. Nikonov have created and published in J. Computational Nanotechnology (2017, no. 1, pp. 7–14) an iterative algorithm for computing coefficients $a_1,\ldots,a_n$ and thresholds $b_0,\ldots,b_k$ for any k-valued threshold function $f(x_1,\ldots,x_n)$ given by its values $f(c_1,\ldots,c_n)$ for all $(c_1\ldots c_n)$ in $\{0,\ldots,k-1\}^n$.

In computer experiment they showed the convergence of this algorithm on many different examples. Here, we present a theoretical proof of this algorithm convergence on each k-valued threshold function for a finite number of steps (iterations). The proof is very much similar to the geometrical proof of perceptron convergence theorem by M. Minsky and S. Papert.

Keywords: *threshold functions, iterative algorithms, convergence.*

Введение

В целом ряде разделов дискретной математики возникает задача распознавания принадлежности функции к классу пороговых и восстановления (характеризации) неизвестной дискретной функции из этого класса с помощью последовательных вопросов о её значениях в точках.

Определение 1 [1]. Функция k-значной логики $f(x_1, \ldots, x_n)$, для которой существует линейная форма $L(x_1, \ldots, x_n) = a_1x_1 + a_2x_2 + \ldots + a_nx_n$, $x_i \in \mathbb{Z}_k$ с вещественными коэффициентами и набор вещественных порогов $b_0 < b_1 < \ldots < b_k$, такие, что для всех $i \in \{0, \ldots, k-1\}$ выполняется условие

$$f(x_1, \ldots, x_n) = i \Longleftrightarrow b_i \leqslant L(x_1, \ldots, x_n) < b_{i+1},$$

называется *пороговой* k-значной функцией.

Замечание 1. В силу неоднозначности задания пороговой функции будем полагать возможным использование порогов, удовлетворяющих нестрогому неравенству

$$b_0 \leqslant b_1 \leqslant \ldots \leqslant b_k.$$

В случае равенства порогов $b_i = b_{i+1}$ для некоторого $i \in \{0, \ldots, k-1\}$, очевидно, функция $f(x_1, \ldots, x_n)$ не принимает значения i. Если пороговая функция принимает значение i, то будем считать, что выполняется строгое двухстороннее неравенство

$$f(x_1, \ldots, x_n) = i \Leftrightarrow b_i < L(x_1, \ldots, x_n) < b_{i+1}.$$

Этого всегда можно добиться небольшим изменением порогов или весов.

Под алгоритмом характеризации пороговой k-значной функции понимается процедура нахождения какого-либо семейства параллельных гиперплоскостей, разделяющих множества различных значений данной функции, то есть нахождения коэффициентов $a_1, a_2, \ldots, a_n$ линейной формы $L(x_1, \ldots, x_n)$ и множества порогов $b_0, b_1, \ldots, b_{k-1}$.

Подобная задача рассматривалась в работе [2], где предложен итеративный алгоритм нахождения представления k-значной пороговой функции, который на практических примерах показал сходимость.

Далее описан новый алгоритм характеризации k-значных пороговых функций и доказана его сходимость. Как показывают эксперименты [1], трудоёмкость нового алгоритма на порядки меньше, чем у алгоритма в [2].

1. Основные понятия и характеристики пороговых функций

Определение 2. Будем говорить, что линейная форма $L(x_1, \ldots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n$ даёт *чистое разделение* областей значений функции $f(x_1, \ldots, x_n)$, принимающей все значения из $\mathbb{Z}_k$, если для любого $\alpha \in \{0, \ldots, k-2\}$ выполняется строгое неравенство

$$\max_{f(\varepsilon_1, \ldots, \varepsilon_n) = \alpha} \{a_1\varepsilon_1 + \cdots + a_n\varepsilon_n\} < \min_{f(\varepsilon_1, \ldots, \varepsilon_n) = \alpha+1} \{a_1\varepsilon_1 + \cdots + a_n\varepsilon_n\}.$$

Если это выполняется, то границы $b_0, b_1, \ldots, b_k$ можно определить, например, следующим способом:

$$b_\alpha = \min_{f(\varepsilon_1,\ldots,\varepsilon_n)=\alpha} \{a_1\varepsilon_1 + \ldots + a_n\varepsilon_n\}, \ \alpha \in \mathbb{Z}_k,$$
$$b_k = \max_{f(\varepsilon_1,\ldots,\varepsilon_n)=k-1} \{a_1\varepsilon_1 + \ldots + a_n\varepsilon_n\} + 1.$$

Замечание 2. В случае, когда функция $f(x_1, \ldots, x_n)$ не принимает некоторых значений из $\mathbb{Z}_k$, необходимо следующим образом убрать из рассмотрения соответствующие области значений: пусть функция $f(x_1, \ldots, x_n)$ принимает значения $0 \leqslant \alpha_0 < \alpha_1 < \ldots < \alpha_t < k$, $0 < t < k$. Тогда для всех $i \in \{0, \ldots, t-1\}$ необходимо проверить выполнение строгого неравенства

$$\max_{f(\varepsilon_1,\ldots,\varepsilon_n)=\alpha_i} \{a_1\varepsilon_1 + \ldots + a_n\varepsilon_n\} \ < \min_{f(\varepsilon_1,\ldots,\varepsilon_n)=\alpha_{i+1}} \{a_1\varepsilon_1 + \ldots + a_n\varepsilon_n\}.$$

Пороги в этом случае можно определить, например, следующим образом: для всех значений $\alpha_0, \ldots, \alpha_t$ присвоить значения

$$b_{\alpha_i} = \min_{f(\varepsilon_1,\ldots,\varepsilon_n)=\alpha_i} \{a_1\varepsilon_1 + \ldots + a_n\varepsilon_n\}; \tag{1}$$

далее, если функция принимает значение $k-1$, то положить

$$b_k = \max_{f(\varepsilon_1,\ldots,\varepsilon_n)=k-1} \{a_1\varepsilon_1 + \ldots + a_n\varepsilon_n\} + 1, \tag{2}$$

в противном случае положить $b_k = +\infty$; для оставшихся значений $j \in \mathbb{Z}_k \backslash \{\alpha_0, \ldots, \alpha_t\}$ (которые не являются значениями функции $f(x_1, \ldots, x_n)$), начиная со старшего, присвоить соответствующему порогу значение $b_j = b_{j+1}$.

В работе [1] представлен подробный обзор существующих методов характеризации k-значных пороговых функций и предложен новый алгоритм характеризации, использующий в качестве первичной аппроксимации коэффициентов линейной формы коэффициенты роста и коэффициенты возрастания.

Определение 3. Для функции k-значной логики $f(x_1, \ldots, x_n)$ *коэффициентом роста по переменной* x_i называется величина

$$\Delta_i = \sum_{(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)} (f(x_1, \ldots, x_{i-1}, k-1, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)).$$

Определение 4. Для функции k-значной логики $f(x_1, \ldots, x_n)$ *коэффициентом возрастания по переменной* x_i называется величина

$$\lambda_i = \sum_{(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)\in\mathbb{Z}_k^{n-1}} \sum_{l=0}^{k-2} \sum_{\varepsilon=l+1}^{k-1} (f(x_1, \ldots, x_{i-1}, \varepsilon, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, l, x_{i+1}, \ldots, x_n)).$$

2. Алгоритм характеризации k-значных пороговых функций

Приведём новый алгоритм (алгоритм 1) характеризации k-значных пороговых функций в формализованном виде. Для этого понадобятся следующие обозначения для произвольного $i \in \mathbb{Z}_k$:

$$F_i = \{(x_1, \ldots, x_n) \in \mathbb{Z}_k^n : f(x_1, \ldots, x_n) = i\};$$
$$\max(F_i) = \max_{(x_1,\ldots,x_n)\in F_i} \{L(x_1, \ldots, x_n)\}; \tag{3}$$
$$x_{\max}(F_i) = \{(x_1, \ldots, x_n) \in \mathbb{Z}_k^n : L(x_1, \ldots, x_n) = \max(F_i)\};$$
$$\min(F_i) = \min_{(x_1,\ldots,x_n)\in F_i} \{L(x_1, \ldots, x_n)\}; \tag{4}$$
$$x_{\min}(F_i) = \{(x_1, \ldots, x_n) \in \mathbb{Z}_k^n : L(x_1, \ldots, x_n) = \min(F_i)\}.$$

В случае если $F_i = \varnothing$, положим $\max(F_i) = +\infty$ и $\min(F_i) = -\infty$, $x_{\min}(F_i) = x_{\max}(F_i) = (-1, \ldots, -1)$.

Пусть пороговая функция $f(x_1, \ldots, x_n)$ принимает только значения $0 \leqslant \alpha_0 < \alpha_1 < \ldots < \alpha_t < k$, $0 < t < k$.

Алгоритм 1. Новый алгоритм характеризации пороговых k-значных функций

Вход: Δ_i, λ_i, $i = 1, \ldots, n$; F_{α_i}, α_i, $i = 0, \ldots, t$

// Инициализация коэффициентов линейной формы

Положить $(a_1, \ldots, a_n) := (\Delta_1, \ldots, \Delta_n)$ либо $(a_1, \ldots, a_n) := (\lambda_1, \ldots, \lambda_n)$

// Проверка чистого разделения областей значений функции

Для всех $i = 0, \ldots, t-1$

  вычислить $\min\left(F_{\alpha_{i+1}}\right)$ по (4);

  вычислить $\max(F_{\alpha_i})$ по (3).

  Если $\max(F_{\alpha_i}) \geqslant \min\left(F_{\alpha_{i+1}}\right)$, **то**

    // Блок «Коррекция»

    $x \xleftarrow{\mathcal{R}} x_{\max}(F_{\alpha_i})$,

    $y \xleftarrow{\mathcal{R}} x_{\min}\left(F_{\alpha_{i+1}}\right)$.

    Для всех $j = 1, \ldots, n$

      $a_j := a_j - x_j + y_j$. *// Коррекция коэффициентов линейной формы*

    Перейти на шаг 2.

// Блок «Вычисление порогов»

Вычислить $\min(F_{\alpha_0})$.

Для всех $i = 0, \ldots, t$

  положить $b_{\alpha_i} := \min(F_{\alpha_i})$ *// в соответствии с* (1).

Если $\alpha_t = k - 1$, **то**

  вычислить $\max(F_{k-1})$,

  положить $b_k := \max(F_{k-1}) + 1$, *// в соответствии с* (2)

иначе

  $b_k := +\infty$.

// Задание остальных порогов (см. замечание 2)

$Z := \mathbb{Z}_k \backslash \{\alpha_0, \ldots \alpha_t\}$.

Пока $Z \neq \varnothing$

  $j := \max Z$, $b_j = b_{j+1}$, $Z = Z \backslash \{j\}$.

Вывести $(a_1, \ldots, a_n)$, $(b_0, \ldots, b_k)$

На шаге 1 вектор коэффициентов линейной формы инициализируется коэффициентами роста либо коэффициентами возрастания. На шагах 6 и 7 производится случайный и равновероятный выбор точек x и y из множеств $x_{\max}(F_{\alpha_i})$ и $x_{\min}\left(F_{\alpha_{i+1}}\right)$ соответственно.

3. Доказательство сходимости

На практике алгоритм показал сходимость во всех рассмотренных случаях при изменении параметров в широких интервалах. Более того, в сравнении с известным алгоритмом Обрадовича [2] он демонстрирует значительное превосходство по трудоёмкости, а на параметрах n и k больше 10 — на несколько порядков. В связи с этим теоретическое доказательство его сходимости можно рассматривать как ожидаемое и чрезвычайно важное для данной области исследования. Предварительно отметим, что доказательство теоремы 1 во многом схоже с геометрическим доказательством теоремы о сходимости персептрона, изложенным в работе [3].

Теорема 1. Если f — k-значная пороговая функция, то алгоритм 1 сходится за конечное число шагов и даёт её характеризацию.

Доказательство. Пусть пороговая функция k-значной логики $f(x_1, \ldots, x_n)$ задаётся линейной формой $L(x_1, \ldots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n$, т. е. вектором $\mathbf{L} = (a_1, a_2, \ldots, a_n)$. Вокруг вектора $\mathbf{L}$ всегда можно построить конус допустимых решений Q, такой, что любой вектор, лежащий в данном конусе, даёт чистое разделение областей значений функции $f(x_1, \ldots, x_n)$ [3].

Конусом допустимых решений Q назовём множество векторов, для которых выполнены следующие свойства:

1) $\alpha \in Q \implies k\alpha \in Q$ для всех $k > 0$;
2) $\alpha \in Q,\ \beta \in Q \implies (\alpha + \beta) \in Q$;
3) для любого $\alpha \in Q$ линейная форма, построенная по коэффициентам вектора α, даёт чистое разделение областей значений функции $f(x_1, \ldots, x_n)$.

Обозначим угол раствора конуса Θ ($\Theta > 0$). Докажем, что за конечное число шагов алгоритма вектор коэффициентов линейной формы окажется внутри конуса допустимых решений.

Очевидно, что в силу абсолютной монотонности пороговой функции [4] векторы коэффициентов роста и коэффициентов возрастания лежат в одном ортанте с вектором $\mathbf{L}$. Таким образом, алгоритм начинает работу с вектора $\mathbf{A}_0$, лежащего в одном ортанте с вектором $\mathbf{L}$.

Рассмотрим операцию коррекции в алгоритме. Обозначим текущий вектор линейной формы $\mathbf{A}_i$, он поступает на вход блока «Коррекция», после коррекции переходит в вектор $\mathbf{A}_{i+1}$.

Алгоритм осуществляет коррекцию с помощью двух точек $\mathbf{u}_i \in x_{\max}\left(F_{\alpha_j}\right)$ и $\mathbf{v}_i \in x_{\min}\left(F_{\alpha_{j+1}}\right)$ тогда и только тогда, когда одновременно выполняются следующие два неравенства:

1) $\mathbf{A}_i\mathbf{v}_i \leqslant \mathbf{A}_i\mathbf{u}_i$, так как задействован блок «Коррекция» алгоритма;
2) при этом для неизвестного истинного вектора $\mathbf{L}$ выполнено противоположное неравенство $\mathbf{L}\mathbf{v}_i > \mathbf{L}\mathbf{u}_i$ в силу определения 1.

Эти неравенства равносильны системе

$$\begin{cases} \mathbf{L}(\mathbf{v}_i - \mathbf{u}_i) > 0, \\ \mathbf{A}_i(\mathbf{v}_i - \mathbf{u}_i) \leqslant 0. \end{cases} \tag{5}$$

Вектор коррекции $(\mathbf{v}_i - \mathbf{u}_i) \equiv \mathbf{C}_i$ прибавляется к текущему значению $\mathbf{A}_i$ и получается новое значение $\mathbf{A}_{i+1} = \mathbf{A}_i + \mathbf{C}_i$.

Рассмотрим плоскость, на которой лежат векторы $\mathbf{L}$ и $\mathbf{A}_i$ (рис. 1). Обозначим $\mathbf{C}_{LAi}$ проекцию вектора $\mathbf{C}_i$ на плоскость, образованную векторами $\mathbf{L}$ и $\mathbf{A}_i$. Перенесём начало этой проекции в конец вектора $\mathbf{A}_i$.

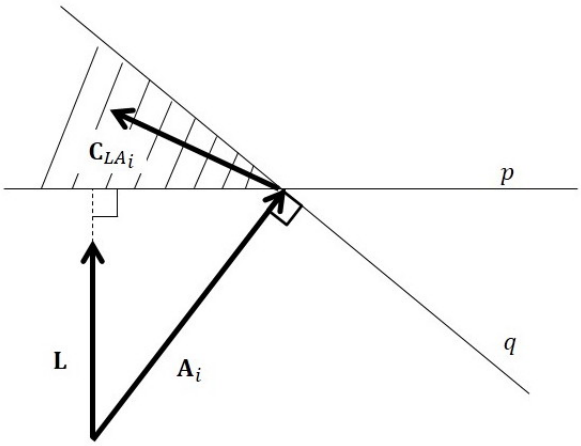


Рис. 1. Проекция вектора коррекции

В силу первого неравенства системы (5) конец вектора $\mathbf{C}_{LAi}$ должен быть ниже прямой q, в силу второго неравенства — выше прямой p, т. е. должен лежать в заштрихованной области на рис. 1. Таким образом, вектор $\mathbf{C}_{LAi}$ направлен к вектору $\mathbf{L}$.

Рассмотрим конус K_i, образованный вращением вектора $\mathbf{A}_i$ вокруг вектора $\mathbf{L}$. Видно, что вектор $\mathbf{C}_i$ заходит внутрь конуса K_i. Для каждого шага алгоритма возможны два случая: вектор $\mathbf{C}_i$ лежит внутри конуса K_i целиком или выходит за пределы конуса K_i, то есть «прокалывает» конус, как на рис. 2.

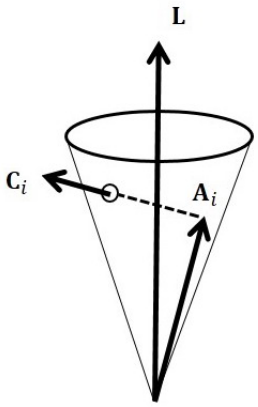


Рис. 2. «Прокол конуса»

Покажем, что «прокол конуса» может произойти конечное число раз. Так как по замечанию 1 для любых $\mathbf{v}_i, \mathbf{u}_i \in \mathbb{Z}_k^n$ выполняется неравенство $\mathbf{L}(\mathbf{v}_i - \mathbf{u}_i) > 0$, существует $r > 0$, такое, что $\mathbf{L}(\mathbf{v}_i - \mathbf{u}_i) > r$ для любых $\mathbf{v}_i, \mathbf{u}_i \in \mathbb{Z}_k^n$. Это означает, что часть заштрихованной на рис. 1 области, заведомо содержащая конец вектора $\mathbf{C}_{LA_i}$, лежит на $r/|L|$

выше, чем окончание вектора $\mathbf{A}_i$. Рис. 3 является модификацией рис. 1, учитывающей вышеизложенное. Если рассмотреть проекцию на плоскость основания конуса K_i, то расстояние от конца проекции на эту плоскость вектора $\mathbf{C}_i$ до касательной к окружности конуса K_i в точке окончания вектора $\mathbf{A}_i$ будет не менее чем d.

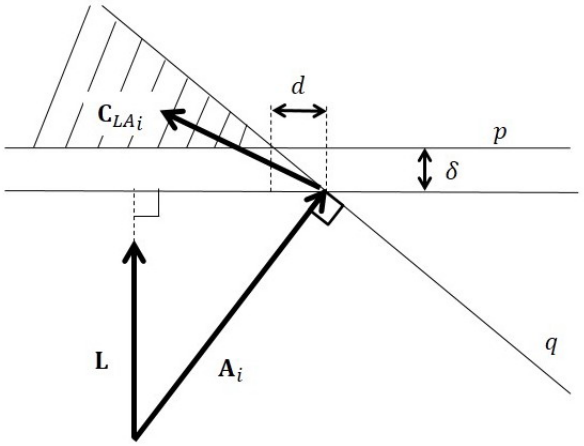


Рис. 3. Проекция вектора коррекции

Нижнюю границу для d определяет тот факт, что угол между векторами $\mathbf{A}_i$ и $\mathbf{L}$ составляет не меньше Θ (иначе алгоритм останавливает работу) и меньше $(\pi/2 - \Theta)$; таким образом, $d \geqslant (r/|L|)\operatorname{ctg}\Theta > 0$. Длина вектора $\mathbf{C}_i$ ограничена сверху величиной $(k-1)\sqrt{n}$.

Рассмотрим проекцию на плоскость основания конуса K_i. На рис. 4 изображены две окружности: окружность основания конуса K_i вокруг вектора $\mathbf{L}$ и окружность радиуса $(k-1)\sqrt{n}$ вокруг окончания вектора $\mathbf{A}_i$. К окружности основания конуса K_i проведена касательная в точке окончания вектора $\mathbf{A}_i$. На расстоянии d от касательной проведена линия.

Согласно вышеизложенному, окончание проекции вектора $\mathbf{C}_i$ на данную плоскость должно лежать на расстоянии не меньше d от касательной, не больше $(k-1)\sqrt{n}$ от окончания вектора $\mathbf{A}_i$ и по одну сторону с вектором $\mathbf{L}$ относительно касательной. Таким образом, окончание проекции вектора $\mathbf{C}_i$ лежит в заштрихованной на рис. 4 области.

Можно легко заметить, что заштрихованная область не целиком лежит внутри окружности основания конуса K_i. Когда окончание проекции вектора $\mathbf{C}_i$ будет попадать в заштрихованную область вне окружности основания конуса K_i, будет происходить «прокол конуса».

Покажем, что с ростом числа итераций i «прокол конуса» прекратится. Так как вектор $\mathbf{C}_i$ всегда содержит вертикальную составляющую не менее $r/|L|$, высота конуса K_i растет с каждой итерацией алгоритма. Угол между векторами $\mathbf{L}$ и $\mathbf{A}_i$ больше Θ (в противном случае алгоритм останавливается и решение найдено). Таким образом, радиус основания конуса K_i увеличивается неограниченно.

С увеличением радиуса основания конуса K_i вся заштрихованная область попадёт внутрь окружности основания конуса K_i, как показано на рис. 5, и «прокол конуса» прекратится. Таким образом, «прокол конуса» может произойти конечное число раз, после чего вектор коррекции $\mathbf{C}_i$ целиком будет попадать в конус K_i и оконча-

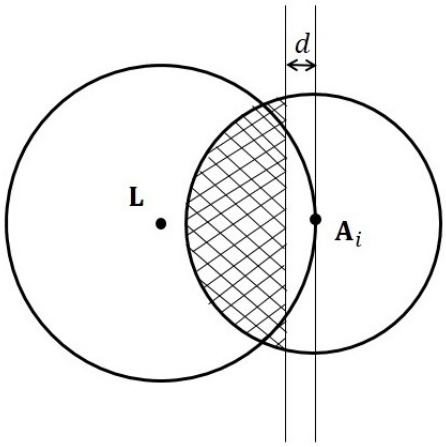


Рис. 4. Проекция на плоскость основания конуса K_i

ние вектора коррекции будет всегда попадать внутрь круга фиксированного радиуса. С ростом числа итераций высота конуса K_i растёт неограниченно минимум на $r/|L|$ с каждой итерацией (см. рис. 3). Вместе с высотой конуса растёт радиус основания конуса допустимых решений Q — минимум на величину $r\cos\Theta/|L|$.

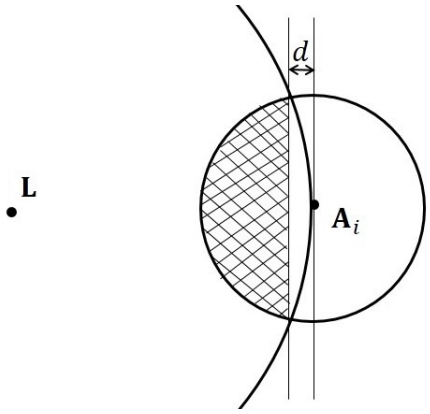


Рис. 5. Проекция на плоскость основания конуса

Итак, начиная с некоторого шага алгоритма, окончания всех векторов $\mathbf{A}_i$ будут лежать в вертикальном цилиндре, расположенном внутри конуса допустимых решений, имеющего своей осью вектор $\mathbf{L}$. ■

Заключение

Предложенный в работе [1] алгоритм сходится и находит за конечное число шагов решение проблемы характеризации k-значных пороговых функций.

Сходимость алгоритма 1, как следует из доказательства, выполняется при использовании и коэффициентов роста, и коэффициентов возрастания для первичной аппроксимации коэффициентов линейной формы. Вместе с тем эксперименты показывают, что получаемые в этих двух случаях результаты, а также сложности реализации алгоритмов отличаются и требуют дальнейшего изучения.

ЛИТЕРАТУРА

1. *Бурделёв А. В., Никонов В. Г.* О новом алгоритме характеризации k-значных пороговых функций // Computational Nanotechnology. 2017. Вып. 1. С. 7–14.
2. *Obradovic Z. and Parberry I.* Learning with discrete multi-valued neurons // Proc. 7th Intern. Conf. Machine Learning. University of Texas, Austin, Texas, June 21–23 1990. P. 392–399.
3. *Минский М., Пейперт С.* Персептроны. М.: Мир, 1971.
4. *Никонов В. Г., Никонов Н. В.* Особенности пороговых представлений k-значных функций // Труды по дискретной математике. 2008. Т. 11. Вып. 1. С. 60–85.

REFERENCES

1. *Burdelev A. V. and Nikonov V. G.* O novom algoritme harakterizacii k-znachnyh porogovyh funkcii [A new algorithm for recognition of k-valued threshold functions]. Computational Nanotechnology, 2017, vol. 1, pp.7–14. (in Russian)
2. *Obradovic Z. and Parberry I.* Learning with discrete multi-valued neurons. Proc. 7th Intern. Conf. Machine Learning, University of Texas, Austin, Texas, June 21–23 1990, pp. 392–399.
3. *Minsky M. and Papert S.* Perceptrons. Cambridge, MA, MIT Press, 1969.
4. *Nikonov V. G. and Nikonov N. V.* Osobennosti porogovyh predstavlenii k-znachnyh funkcii [Features of threshold representations of k-valued functions]. Tr. Diskr. Mat., 2008, vol. 11, iss. 1, pp. 60–85. (in Russian)

УДК 519.254

БЫСТРЫЙ АЛГОРИТМ КЛАСТЕРНОГО АНАЛИЗА k-MEDOIDS

И. Н. Дмитриев

Федеральное учебно-методическое объединение в сфере высшего образования по УГСН 10.00.00 Информационная безопасность, г. Москва, Россия

Рассмотрена новая реализация FKM известного алгоритма k-medoids, основанная на общеизвестной PAM-реализации и использующая новую эвристику выбора центров кластеров, методику CLARA, а также предварительное прореживание L-SPAR, что позволило перейти от квадратичной вычислительной сложности реализации к линейной и снизить временные затраты на обработку реальных данных в среднем в 16 раз.

Ключевые слова: *быстрый алгоритм кластерного анализа, PAM-реализация k-medoids, методика CLARA, прореживание L-SPAR.*

DOI 10.17223/20710410/39/11

FAST ALGORITHM OF CLUSTER ANALYSIS k-MEDOIDS

I. N. Dmitriev

Federal Educational and Methodological Association in Information Security, Moscow, Russia

E-mail: i.n.dmitriev@mail.ru

The algorithm k-medoids (KM) is widely used for clustering graphs. The following modifications of it by adding some procedures have been researched in a computer experiments: CKM = KM + CLARA (Cluster LARge Application), LSKM = KM + + L-SPAR (Local SPARsification), BKM = KM + CLARA + L-SPAR, NKM = = KM + NH (New Heuristic choosing cluster center), CNKM = NKM + CLARA, LSNKM = NKM + L-SPAR, FKM = NKM + CLARA + L-SPAR. The experiment results show that L-SPAR procedure allows to decrease the average running time for LSKM by 5.3 %, BKM by 6.1 %, LSNKM by 8.1 %, FKM by 8.3 % and to improve the clustering quality by 2 %, 2.1 %, 3 %, 3.3 % respectively. Besides, the number of iterations in NKM is twice more than in KM, but the running time of NKM is an order less than of KM, the running time of CKM is two thirds of the running time of KM, the computer complexity of CNKM linearly depends on the graph size, and the application of CLARA does not diminish it appreciably. NH allows 16 times decreasing the computer complexity of KM without loosing the clustering quality. The experiments were conducted on data arrays that represented graphs of social networks YouTube, Live Journal, and the shop Amazon.

Keywords: *fast algorithm of cluster analysis, PAM, CLARA, Local Graph Sparsification.*

Введение

Известный алгоритм k-medoids является модификацией классического алгоритма кластеризации k-means и предназначен для решения задач выделения групп объектов (кластеров) в случаях, когда проводится кластеризация объектов без использования свойств линейного пространства. Такие задачи возникают, когда используется

специфическая мера близости объектов (не расстояние), например при кластеризации неориентированных графов. В этом случае, в отличие от k-means, центром кластера может быть не любая точка признакового пространства (центроид), а только точка, принадлежащая кластеризуемой выборке — медоид.

Неориентированные графы используются для представления информации о фактах взаимодействия или взаимосвязи объектов. Из них легко получить количество взаимодействий с объектом, как с каждым в отдельности, так и совместных взаимодействий для каждой пары объектов. В частности, задачу выделения новых групп в данных сетевого трафика можно рассматривать как задачу выделения кластеров в графе взаимодействия узлов сети и дальнейшего анализа этих кластеров с использованием контентных характеристик.

На концептуальном уровне кластером неориентированного графа называется такая группа вершин, в которой внутригрупповые связи гораздо плотнее межгрупповых [1]. В настоящей работе используется предположение о структуре кластеров в графе, согласно которому каждая вершина графа входит в один и только один кластер. Отметим, что из предположения следует, что разбиение полностью покрывает граф. Тогда можно говорить о поиске кластеров в графе как о задаче поиска разбиения множества вершин на подмножества, которое минимизирует некоторый функционал.

Для решения задачи выделения кластеров в графах предназначена PAM-реализация (Partitioning Around Medoids) алгоритма k-medoids [2]. В работе рассмотрен быстрый алгоритм k-medoids (Fast K-Medoids, FKM) — модификация PAM, в которой предложена новая эвристика выбора следующего медоида, позволяющая снизить вычислительную сложность алгоритма, дополненная механизмом предварительного прореживания L-SPAR и методикой кластеризации больших массивов данных CLARA (CLustering LARge Applications). В результате экспериментальной оценки временных характеристик и показателей качества FKM относительно PAM-реализации алгоритма k-medoids, также дополненного CLARA и L-SPAR (далее — Basic K-Medoids, BKM), на реальных графах, построенных на базе сетевых ресурсов YouTube, Live Journal и Amazon, показано, что FKM не только не уступает в качестве разбиения BKM, но и позволяет сократить временные затраты на обработку массивов данных.

1. Быстрый алгоритм кластерного анализа k-medoids

Для описания PAM введем понятие модулярности, которая используется в качестве функции потерь при оценке «удачности» выбора нового медоида, а также при проверке критерия останова алгоритма.

Пусть $G = (V, E)$ — неориентированный граф с матрицей смежности $A = (a_{ij})$, где V — множество вершин, E — множество рёбер, и $C = \{C_1, C_2, \dots C_k\}$ — разбиение вершин графа на кластеры.

Определение 1. Модулярностью Гирвана — Ньюмана [3] для графа $G = (V, E)$ называется функционал вида

$$Q(C) = \frac{1}{2\,|E|}\sum_{i=1}^{|V|}\sum_{j=1}^{|V|}\left(a_{ij} - \gamma\frac{\deg(v_i)\deg(v_j)}{2\,|E|}\right)\sigma(h_C(v_i), h_C(v_j)),$$

где $\deg(v_i)$ — степень вершины графа v_i; $h_C(v_i)$ — номер кластера, в который попала вершина v_i при разбиении C; σ — символ Кронекера; $\gamma > 0$ — константа, позволяющая управлять размером кластеров. Выбор $\gamma < 1$ приводит к увеличению размера кластеров в разбиении, при $\gamma > 1$ размеры кластеров уменьшаются, а их количество возрастает.

Функционал $Q(C)$ не является непрерывным и задача его оптимизации дискретная. Для поиска глобального оптимума используют приближённые алгоритмы. Некоторые из них действительно оптимизируют значение функционала, другие по значению модулярности выбирают наилучшее решение из найденных, то есть не дают гарантий локальной оптимальности решения.

Для определения степени близости двух вершин в методике CLARA, способе предварительного прореживания L-SPAR, а также на этапе распределения вершин по кластерам в реализации PAM используется коэффициент Жаккарда. Применительно к неориентированному графу G для двух вершин v_i и v_j этот коэффициент имеет следующий вид: $\mathrm{sim}(v_i, v_j) = \dfrac{|\mathrm{Adj}\,(v_i) \cap \mathrm{Adj}\,(v_j)|}{|\mathrm{Adj}\,(v_i) \cup \mathrm{Adj}\,(v_j)|}$, где $\mathrm{Adj}(i)$ — множество соседей вершины i.

Будем понимать под медоидом объект, принадлежащий набору данных или кластеру, отличие которого от других объектов в нём минимально. Медоиды близки по смыслу центроидам, но, в отличие от них, являются объектами, принадлежащими кластеру, и, как правило, используются в тех случаях, когда невозможно вычислить средние координаты или центр масс кластера.

PAM является одной из реализаций k-medoids (алгоритм 1) и представляет собой «жадный» алгоритм с простейшей эвристикой (проход по всем вершинам, шаг 4 алгоритма).

Алгоритм 1. PAM-реализация алгоритма k-medoids

Вход: Граф G, заданное число кластеров k.
Выход: Оптимальное разбиение на k кластеров (оптимум модулярности).
Медоид для каждого кластера

1: ***Инициализация***
Случайным образом выбираем k вершин для множества медоидов M.
Для каждой вершины находим ближайший (по Жаккарду) медоид, формируя разбиение на кластеры $C = \{C_1, C_2, \ldots C_k\}$.
$minCost \leftarrow$ значение модулярности для исходного разбиения.
2: **Повторять**
3: **Для всех** $m \in M$
4: **Для всех** $v \in C_m$ и $v \neq m$
5: перемещаем медоид в v;
6: строим очередное разбиение с новым медоидом;
7: $cost \leftarrow$ модулярность для очередного разбиения.
8: **Если** $cost < minCost$, **то**
9: $m' \leftarrow m$, $minCost \leftarrow cost$.
10: $m \leftarrow m'$
11: **Пока** не перестанет изменяться $minCost$.

Оценим вычислительную сложность PAM-реализации алгоритма k-medoids. На каждой итерации перебирается $n - k$ вершин графа (здесь n — общее количество вершин), в которые перемещается соответствующий медоид. Для каждой такой замены нужно пересчитать расстояния от всех точек до медоидов. Если все попарные расстояния между точками помещаются в память, то этот этап займёт $(n - k)\,k$ действий. Оптимальная замена требует ещё $(n - k)\,k$ действий. Таким образом, сложность одной

итерации в худшем случае равна $(n-k+1)(n-k)k = \mathrm{O}(n^2k)$, то есть вычислительная сложность реализации $\mathrm{O}(n^2kI)$, где I — количество итераций.

Одним из слабых мест PAM является неэффективный поиск новых кандидатов на места медоидов. Перебор всех имеющихся вершин в графе нельзя назвать оптимальным, попытаемся сократить число кандидатов для перебора. Введём понятие множества замен кластера C_m с медоидом в m.

Определение 2. Множество $Z_m^s \subset C_m \setminus \{m\}$, состоящее из s вершин, будем называть множеством замен кластера C_m.

Для построения множества Z_m^s могут быть использованы следующие техники:

1) выбор s случайных вершин из C_m с равной вероятностью;
2) выбор s случайных вершин из C_m с вероятностью, пропорциональной степени вершины;
3) выбор $s/2$ пар вершин из C_m с наибольшим значение коэффициента Жаккарда;
4) выбор s вершин из «ближайших» соседей медоида m.

Для того чтобы однозначно сказать, какая из техник позволит сократить число итераций, повысить качество итогового разбиения или прийти к глобальному минимуму функционала модулярности, необходимо провести дополнительные эксперименты. В данной работе для построения множества Z_m^s использована самая простая техника: равновероятный выбор s случайных вершин из кластера.

Суть новой эвристики заключается в сокращении множества для выбора нового медоида со всего графа до множества замен кластера C_m. Это позволяет искать оптимальную замену не по всем вершинам графа, а по одному кластеру, перебирая не все вершины внутри этого кластера, а лишь s из них. В этом случае сложность одной итерации равна $(n-k)k(s+1) = \mathrm{O}(nks)$ — она линейна по числу вершин, а сложность всей реализации составляет $\mathrm{O}(nksI)$, где $s \ll n/k$.

Исследуем, как повлияет новая эвристика на качество результатов, получаемых при работе алгоритма, в зависимости от количества элементов s в множестве замен. В [4] приводится пример алгоритма, в котором имеется параметр с похожим на s смыслом. В работе доказывается, что снижение значения параметра s до 2 и даже до 1 не ухудшает различные метрики качества кластеров.

Для увеличения максимального размера графа, который может обработать реализация алгоритма k-medoids, применим методику обработки данных большого объёма, называемую CLARA [2]. Применительно к задаче кластеризации графа методика CLARA имеет следующий вид: из исходного графа выбирается подмножество вершин и кластеризуется подграф, образованный этими вершинами, затем, в предположении связности графа, оставшиеся вершины распределяются по ближайшим медоидам из подграфа.

Вся сложность модификации CLARA состоит в последовательном прогоне на разных подмножествах вершин и выборе оптимального из полученных результатов. За счёт этого предполагается компенсировать ущерб от исключения части информации при каждом отдельном прогоне, а также избежать «застревания» в локальном минимуме (в качестве меры оптимальности при выборе результата используется модулярность).

Существуют следующие техники получения подграфа:

1) выбор случайных вершин с равной вероятностью;
2) выбор случайных вершин с вероятностью, пропорциональной степени вершины в исходном графе;

3) выбор вершин с наибольшей степенью;
4) выбор пар вершин с наибольшим коэффициентом Жаккарда.

Для того чтобы установить степень влияния техники на конечный результат методики CLARA, необходимо провести дополнительный эксперимент. В рамках данной работы эксперимент не проводился, поэтому предпочтение отдано первой технике, как наиболее простой и очевидной.

Выбор объёма подграфа для CLARA представляет собой компромисс между качеством и скоростью. Чем больше кластеров присутствует в данных и чем меньше они по размеру, тем больше вероятность того, что некоторые кластеры могут не попасть в выборку. Если разница в размерах кластеров большая, то такой подход вообще противопоказан. Если же кластеры в данных сравнимы по размеру, то будем брать подграф, состоящий из не менее чем $n/2$ вершин. Данное решение принято на основании результатов экспериментов, проведенных над модификацией CLARA для PAM в [2].

CLARA изначально предназначалась для уменьшения времени выполнения алгоритмов кластеризации, чья вычислительная сложность в зависимости от числа вершин растёт быстрее, чем линейно, т. е. когда выполнить алгоритм несколько раз на подграфе быстрее, чем один раз на полном графе. Поэтому выигрыш модификации CLARA по времени заметен на PAM-реализации алгоритма k-medoids, а при использовании реализации с улучшенной эвристикой (имеющей линейную сложность) выигрыш минимальный.

Следующий шаг по улучшению PAM-реализации с новой эвристикой и методикой CLARA называется L-SPAR (Local SPARsification), он подробно описан в [5]. Этот метод предобработки графа перед кластеризацией «прореживает» граф путём удаления части рёбер (но не вершин), не разрушая, как правило, его связность.

Рассмотрим подробнее алгоритм L-SPAR. Сначала для каждой вершины i все рёбра (i, j) сортируются в порядке убывания $\mathrm{sim}(v_i, v_j)$, после чего хвост списка включается в множество для фильтрации. Каждая следующая вершина даёт своё множество рёбер для фильтрации, которое объединяется с уже существующим. Когда все вершины обработаны, все рёбра полученного множества удаляются из графа. Авторы метода предлагают включать в список «выживших» $\min\{1, d_i^e\}$ рёбер, где d_i — степень вершины i и $0 \leqslant e \leqslant 1$. Если ребро попало в список «выживших» хотя бы для одной из своих вершин, оно «выживает». Таким образом, не образуется новых изолированных вершин и, если исходный граф был связным, разреженный также останется связным.

Доказано, что при применении L-SPAR сохраняется степенной закон распределения степеней вершин в графе, приводящий к феномену «тесного мира» («small world»), когда кратчайший путь между двумя вершинами в среднем имеет очень маленькую длину. В то же время степень прорежения в L-SPAR зависит от плотности графа в непосредственной близости от вершины и сохраняет структуру как плотных, так и разреженных сообществ.

Данный метод предобработки графа можно использовать при любом методе кластеризации. Наибольший эффект может быть достигнут для алгоритмов, чья сложность зависит только от числа рёбер, но не вершин, так как сокращается именно число рёбер. В этом плане реализации алгоритма k-medoids не получат глобального прироста в производительности: его сложность зависит только от числа вершин, однако при более отчетливой структуре групп может уменьшиться число итераций, необходимых для сходимости.

2. Сравнение реализаций алгоритма *k*-medoids

Перейдем к результатам экспериментов, проведённых над описанными реализациями алгоритма k-medoids. Обозначения реализаций приведены в табл. 1.

Таблица 1

Обозначения рассмотренных реализаций

Код	Описание
KM	PAM-реализация алгоритма k-medoids
CKM	KM с методикой CLARA
LSKM	KM с предварительным прореживанием рёбер L-SPAR
BKM	KM с методикой CLARA и прореживанием рёбер L-SPAR
NKM	PAM-реализация алгоритма k-medoids с новой эвристикой
CNKM	NKM с методикой CLARA
LSNKM	NKM с предварительным прореживанием рёбер L-SPAR
FKM	NKM с методикой CLARA и прореживанием рёбер L-SPAR

Эксперименты проводились на массивах данных, представляющих собой размеченные графы социальных сетей YouTube, Live Journal, а также магазина Amazon. Рассмотрим подробнее наборы данных, используемых для эксперимента [6].

Amazon — граф, построенный на основе данных о товарах магазина Amazon.com, в качестве вершины выступает товар магазина, при этом две вершины связаны, если известен факт совместной покупки товаров одним покупателем.

Live Journal — граф социальной сети LiveJournal.com, в котором в качестве вершин выступают пользователи, а в качестве связи — взаимная дружба.

YouTube — граф социальной сети Youtube.com, в котором вершинами являются пользователи, связь между ними определяется наличием подписки одного на другого.

Данные массивы обрабатывались не целиком — для кластеризации выбиралось некоторое количество групп (группы определяются разметкой графа), чтобы число вершин объединения этих групп соответствовало наперёд заданному порядку, затем строился граф с вершинами и связями, входящими только в эти группы (табл. 2).

Таблица 2

Описание характеристик данных, использованных в экспериментах

Массив	Код	Порядок количества вершин 10^l	Количество кластеров k
Amazon	A3	$l = 3$	176
	A4	$l = 4$	1635
	A5	$l = 5$	15378
Life Journal	LJ3	$l = 3$	87
	LJ4	$l = 4$	765
	LJ5	$l = 5$	8104
YouTube	Y3	$l = 3$	3
	Y4	$l = 4$	22
	Y5	$l = 5$	176

Из табл. 2 видно, что в массиве Amazon преобладают кластеры небольшого объёма, а в массиве YouTube, наоборот, предпочтение отдано кластерам, имеющим большой объём. Такое различие в размере кластеров при проведении экспериментов позволит сделать вывод о том, как влияет объём кластера на количество итераций, скорость работы алгоритма, а также на качество полученной кластерной структуры.

Эксперименты проводились по трём направлениям:

1) сравнение количества итераций для KM, NKM, LSKM и LSNKM;
2) сравнение времени работы всех описанных реализаций алгоритма k-medoids;
3) сравнение качества получаемых разбиений графа на кластеры для всех описанных реализаций алгоритма k-medoids.

Для определения мер близости результатов кластеризаций рассмотрим множество вершин $V = \{v_1, \ldots, v_n\}$ из n объектов и два различных разбиения этого множества на подмножества (кластеры): $C^{(1)} = \{C_1^{(1)}, \ldots, C_k^{(1)}\}$ и $C^{(2)} = \{C_1^{(2)}, \ldots, C_l^{(2)}\}$, где $C_i^{(1)} \subset V$, $C_j^{(2)} \subset V$, $\bigcup\limits_{i=1}^{k} C_i^{(1)} = V = \bigcup\limits_{j=1}^{l} C_j^{(2)}$, $C_i^{(1)} \cap C_{i'}^{(1)} = \varnothing$ и $C_j^{(2)} \cap C_{j'}^{(2)} = \varnothing$ для $i, i' = 1, \ldots, k$ и $i \neq i'$, $j, j' = 1, \ldots, l$ и $j \neq j'$. Пересечения кластеров, входящих в указанные разбиения, можно представить с помощью таблицы сопряжённости $T_{k\times l}$ (табл. 3). Её элементы n_{ij} суть числа вершин, общих для $C_i^{(1)}$ и $C_j^{(2)}$, $i = 1, \ldots, k$ и $j = 1, \ldots, l$. Дополнительно в $T_{k\times l}$ вводятся столбец и строка сумм основных элементов таблицы по строкам и столбцам, которые соответствуют мощностям множеств $C_i^{(1)}$ и $C_j^{(2)}$ и представляются в виде $n_{i\circ} = \sum\limits_{j=1}^{l} n_{ij}$ и $n_{\circ j} = \sum\limits_{i=1}^{k} n_{ij}$ соответственно.

Таблица 3

Таблица сопряженности $T^{k\times l}$

Номера кластеров	1	2	…	l	Суммы
1	n_{11}	n_{12}	…	n_{1l}	$n_{1\circ}$
2	n_{21}	n_{22}	…	n_{2l}	$n_{2\circ}$
…	…	…	…	…	…
k	n_{k1}	n_{k2}	…	n_{kl}	$n_{k\circ}$
Суммы	$n_{\circ 1}$	$n_{\circ 2}$	…	$n_{\circ l}$	$n_{\circ\circ} \equiv n$

Для оценки конечного разбиения будем использовать F-меру. Эта мера пришла из области оценки качества информационного поиска. Будем трактовать некоторое подмножество (кластер) разбиения вершин графа $C^{(1)}$ как результат информационного запроса, а определённое подмножество (класс) разбиения $C^{(2)}$ — как совокупность вершин графа, соответствующих этому же запросу (эталон). Тогда для j-го кластера и i-го класса полнота R_{ij} и точность P_{ij} информационного поиска суть величины

$$R_{ij} = \frac{n_{ij}}{n_{i\circ}}, \quad P_{ij} = \frac{n_{ij}}{n_{\circ j}}.$$

Для j-го кластера и i-го класса F-мера характеризуется величиной

$$F_{ij} = \left(\frac{1}{2}\frac{1}{P_{ij}} + \frac{1}{2}\frac{1}{R_{ij}}\right)^{-1}.$$

Для всех кластеров F-мера определяется следующим образом:

$$F = \sum_i \tfrac{n_{i\circ}}{n} \max_j F_{ij}.$$

Из определения F_{ij} следует, что $F_{ij} \leqslant 1$ и F_{ij} принимает максимальное значение, когда есть полное соответствие между разбиениями $C^{(1)}$ и $C^{(2)}$.

Для каждого эксперимента реализация алгоритма запускалась несколько раз, затем результаты усреднялись по количеству запусков. Компьютер, на котором производились эксперименты, имеет следующие характеристики: Intel Core M, 4 Гб RAM,

256 Гб SSD, Windows 10 Professional. Алгоритмы реализованы на языке Java, версия JRM 1.8.0.111.

Оценим количество итераций для реализаций KM и NKM в случае графа с прореженными рёбрами и без прореживания. Для этого проведём эксперимент на выборках из массивов Amazon, Live Journal, YouTube с количеством вершин порядка 10^4. Результаты представлены в виде диаграммы, на которой высота столбца отражает количество итераций (рис. 1).

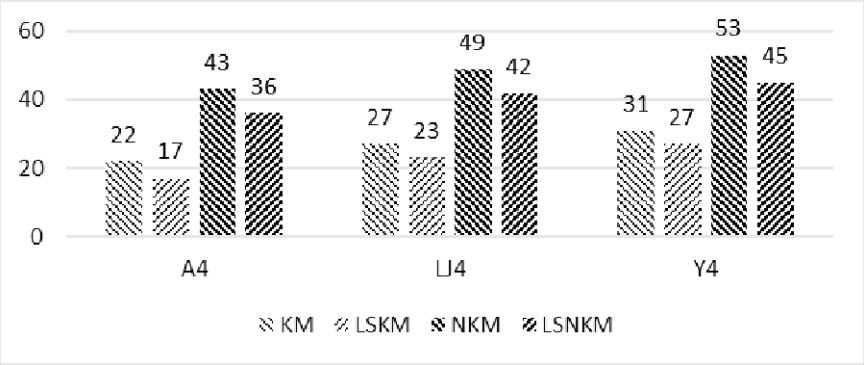


Рис. 1. Число итераций, необходимых для остановки алгоритма

Из эксперимента видно, что для реализации NKM необходимо почти в 2 раза больше итераций, чем для реализации KM. Особенно это проявляется при малом объёме кластеров в массиве Amazon. В массивах с большими кластерами эта разница менее выражена, но всё равно большая. Видно также, что применение предварительного прореживания массивов данных позволяет сократить число итераций, необходимых для успешного завершения работы. Это свидетельствует о том, что локальное прореживание позволяет избавиться от лишнего «шума» среди связей объектов и получить более ясную кластерную структуру.

Оценим временные показатели работы реализаций алгоритма на данных, не подвергавшихся никакой предварительной обработке. При проведении эксперимента время фиксировалось в секундах. Так как разброс значений времени работы имеет достаточно большой диапазон, на графике представлен десятичный логарифм от значений времени работы реализации в соответствующем эксперименте (рис. 2).

Несмотря на то, что реализация NKM требует большего числа итераций, чем базовая реализация KM, она показала время разбиения графа на кластеры на порядок меньше, чем KM. Данный результат связан с введением новой эвристики, позволяющей снизить вычислительную сложность алгоритма.

Реализация CKM позволила сократить время работы в среднем на треть, однако для реализации CNKM время работы на тех же данных сократилось менее чем на процент. Такой контраст вызван тем, что CKM имеет квадратичную сложность и многократная обработка выборок меньшего объема позволяет сократить общее время работы алгоритма, чего нельзя сказать о модификации CNKM: её сложность линейно зависит от числа элементов и применение методики CLARA не позволяет существенно снизить время обработки исходного графа.

Проведём аналогичный эксперимент с данными, которые предварительно прошли прореживание с использованием метода L-SPAR. Для наглядности результатов на графике вместо значения времени работы будем отображать долю от времени, которое

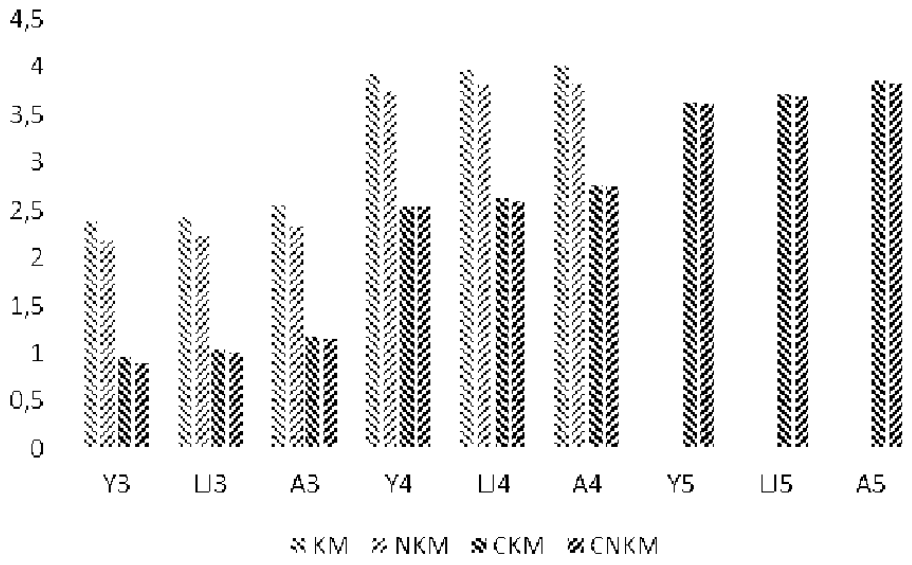


Рис. 2. Десятичный логарифм временных затрат реализаций на различных выборках

потребовалось на обработку данных без предварительного прореживания рёбер графа (рис. 3).

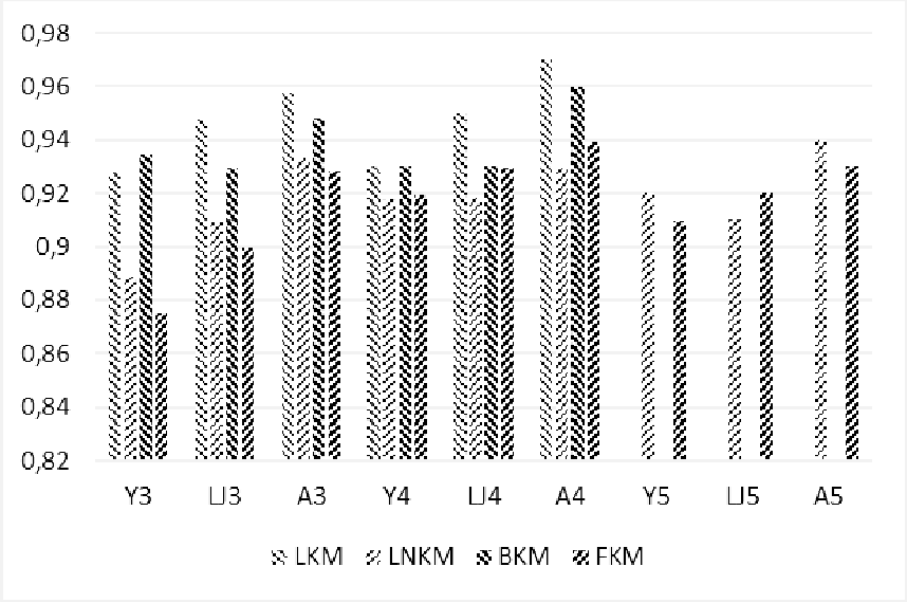


Рис. 3. Доля от времени работы реализаций с прореживанием данных и без него

Из результатов эксперимента следует, что среднее время работы сократилось для LSKM на 5,3 %, BKM на 6,1 %, LSNKM на 8,1 %, FKM на 8,3 %. Реализация LSNKM из-за прореживания получает больший процент сокращения времени обработки данных по сравнению с LSKM, так как граф без «шумных» рёбер имеет более чёткую кластерную структуру и сокращается число итераций для этой реализации. На данных, где объём кластеров больше, положительное действие прореживания рёбер графа усиливается из-за сокращения шума среди связей графа.

Последний аспект, по которому проводилось сравнение реализаций, — качество получаемого разбиения графа. Для оценки качества использована F-мера (рис. 4).

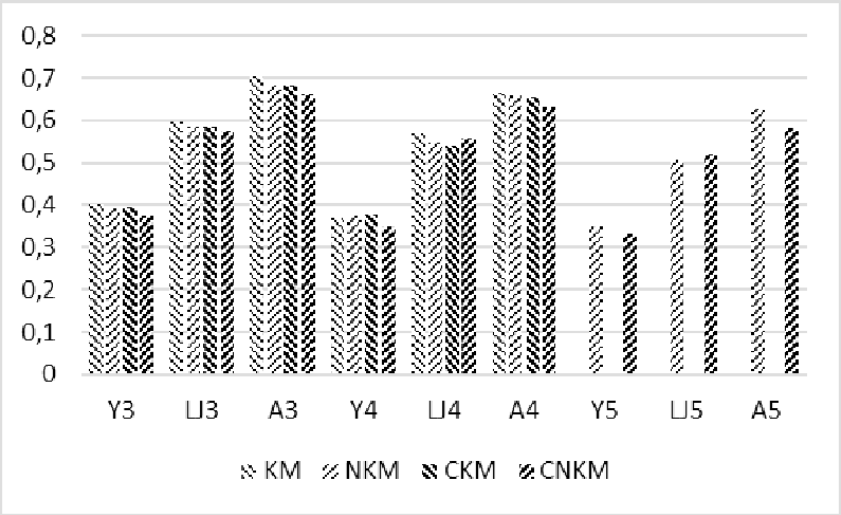


Рис. 4. Значение показателя качества при обработке массивов различными реализациями алгоритма k-medoids

Реализация с новой эвристикой NKM позволяет получить разбиение графа, по качеству не уступающее исходной реализации KM (в рамках статистической погрешности). Для реализаций с методикой CLARA характерно меньшее значение F-меры, чем для аналогичных модификаций без CLARA, в связи с тем, что алгоритмом обрабатывается только часть выборки, а остальные объекты распределяются с использованием коэффициента Жаккарда.

Рассмотрим аналогичный эксперимент с данными, предварительно обработанными методикой прореживания L-SPAR. Как и в предыдущем примере, на графике будем отображать не значение F-меры в конкретном эксперименте, а долю этого значения от показателя качества, полученного на данных без предварительного прореживания (рис. 5).

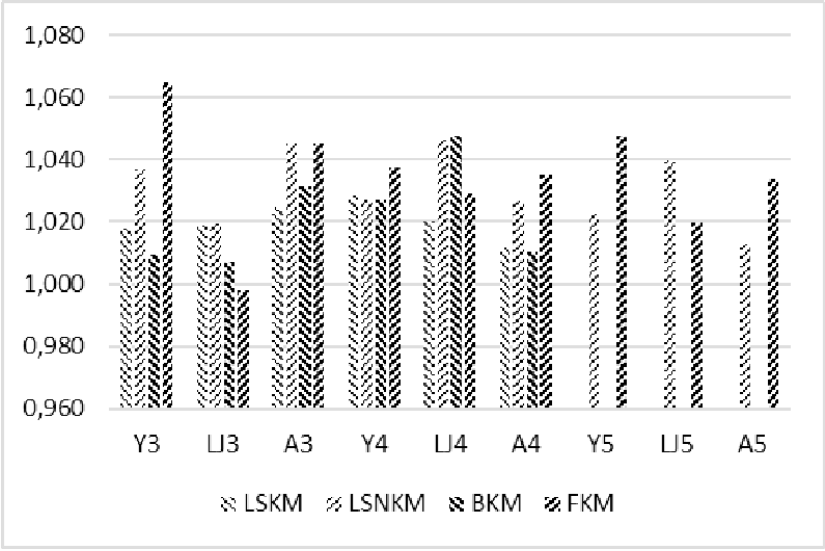


Рис. 5. Доля, на которую увеличилось качество разбиения графа после процедуры прореживания

Из результатов эксперимента следует, что среднее качество разбиения графа улучшилось для LSKM на 2 %, BKM на 2,1 %, LSNKM на 3 %, FKM на 3,3 %. Чёткая кластерная структура, полученная после прореживания «шумных» рёбер графа, позволяет реализациям повысить качество получаемых разбиений.

Заключение

Рассмотрен алгоритм кластеризации неориентированных графов на основе известного алгоритма k-medoids, а также его реализации с новой эвристикой и методикой CLARA. Кроме этого, исследован вопрос использования механизма предварительного прореживания рёбер графа, позволяющего избавиться от «шума» в множестве связей графа. В экспериментах показано, что новая эвристика позволяет существенно (в среднем в 16 раз) снизить вычислительную сложность алгоритма k-medoids и при этом сохранить качество разбиения графа.

Предложенная реализация алгоритма с использованием методики CLARA позволила на треть сократить время обработки данных для PAM-реализации алгоритма k-medoids, однако для реализации с новой эвристикой использование этой методики не дало ощутимого сокращения времени обработки графа. Сокращение времени работы привело к потере качества и в том и в другом случае.

Предварительное прореживание графа при помощи методики L-SPAR позволило сократить среднее время работы для LSKM на 5,3 %, BKM на 6,1 %, LSNKM на 8,1 %, FKM на 8,3 %. Кроме этого, использование механизма предварительного прореживания рёбер графа позволило повысить качество получаемого разбиения для LSKM на 2 %, BKM на 2,1 %, LSNKM на 3 %, FKM на 3,3 %.

Таким образом, предложенная новая эвристика для PAM-реализации алгоритма k-medoids позволяет существенно снизить вычислительную сложность алгоритма в зависимости от числа вершин и приблизить её к линейной. Основным направлением дальнейшего развития предложенной быстрой модификации алгоритма k-medoids является разработка его параллельной версии для работы на нескольких ядрах или процессорах.

ЛИТЕРАТУРА

1. *Fortunato S.* Community detection in graphs // Physics Reports. 2010. V. 486. Iss. 3–5. P. 75–174.
2. *Kaufman L. and Rousseeuw P. J.* Finding Groups in Data: An Introduction to Cluster Analysis. New Jersey: Hoboken, 2005.
3. *Newman M. J.* Modularity and community structure in networks // Proc. of the National Academy of Sciences of the USA. 2006. V. 103. No. 23. P. 8577–8582.
4. *Satuluri V. M.* Scalable Clustering of Modern Networks. Ohio State University Columbus, OH, USA, 2012.
5. *Ovelgönne M.* Scalable Algorithms for Community Detection in Very Large Graphs. Karlsruhe, 2011. 146 p.
6. `http://snap.stanford.edu/data/` — Stanford Large Network Dataset Collection. 2017.

REFERENCES

1. *Fortunato S.* Community detection in graphs. Physics Reports, 2010, vol. 486, iss. 3–5, pp. 75–174.
2. *Kaufman L. and Rousseeuw P. J.* Finding Groups in Data: An Introduction to Cluster Analysis. New Jersey, Hoboken, 2005.

3. *Newman M. J.* Modularity and community structure in networks. Proc. of the National Academy of Sciences of the USA, 2006, vol. 103, no. 23, pp. 8577–8582.
4. *Satuluri V. M.* Scalable Clustering of Modern Networks. Ohio State University Columbus, OH, USA, 2012.
5. *Ovelgönne M.* Scalable Algorithms for Community Detection in Very Large Graphs. Karlsruhe, 2011. 146 p.
6. `http://snap.stanford.edu/data/` — Stanford Large Network Dataset Collection. 2017.

СВЕДЕНИЯ ОБ АВТОРАХ

АГИЕВИЧ Сергей Валерьевич — кандидат физико-математических наук, заведующий НИЛ НИИ прикладных проблем математики и информатики Белорусского государственного университета, г. Минск, Беларусь. E-mail: **agievich@bsu.by**

БУРДЕЛЕВ Александр Владимирович — старший преподаватель кафедры математического моделирования и анализа данных факультета прикладной математики и информатики Белорусского государственного университета, г. Минск.
E-mail: **aburd2011@mail.ru**

ГАЙДАМАКИН Николай Александрович — доктор технических наук, профессор, профессор кафедры алгебры и фундаментальной информатики Уральского федерального университета имени первого Президента России Б. Н. Ельцина, г. Екатеринбург, Россия. E-mail: **haid2@bk.ru**

ДЕВЯНИН Петр Николаевич — доктор технических наук, член-корреспондент Академии криптографии Российской Федерации, доцент, председатель УМС Федерального учебно-методического объединения в сфере высшего образования по УГСН 10.00.00 Информационная безопасность, г. Москва.
E-mail: **devyanin.peter@yandex.ru**

ДМИТРИЕВ Игорь Николаевич — научный сотрудник Федерального учебно-методического объединения в сфере высшего образования по УГСН 10.00.00 Информационная безопасность, г. Москва. E-mail: **i.n.dmitriev@mail.ru**

ЗУЕВ Юрий Анатольевич — доктор физико-математических наук, доцент, профессор кафедры информационной безопасности МГТУ им. Н. Э. Баумана, г. Москва.
E-mail: **79851965730@yandex.ru**

ЛИТИЧЕВСКИЙ Дмитрий Владимирович — аспирант Челябинского государственного университета, г. Челябинск. E-mail: **litichevskiydv@gmail.com**

ЛОГАЧЕВ Олег Алексеевич — кандидат физико-математических наук, доцент, заведующий отделом Института проблем информационной безопасности МГУ имени М. В. Ломоносова, г. Москва. E-mail: **logol@iisi.msu.ru**

НИКИТИН Алексей Юрьевич — аспирант научно-исследовательской лаборатории «Информационная безопасность» кафедры комплексной защиты информации Омского государственного технического университета, г. Омск.
E-mail: **nikitinlexey@gmail.com**

ПЕСНЯК Андрей Алексеевич — студент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск. E-mail: **a.pesnyak92@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. E-mail: **alexander.rybalov@gmail.com**

СТЕФАНЦОВ Дмитрий Александрович — старший преподаватель кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск. E-mail: **d.a.stephantsov@gmail.com**

ШОЛОМОВ Лев Абрамович — доктор физико-математических наук, профессор, главный научный сотрудник лаборатории математических методов анализа и синтеза сложных систем Федерального исследовательского центра «Информатика и управление» РАН, г. Москва. E-mail: **levshol@mail.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 (Emerging Sources Citation Index — ESCI и Russian Science Citation Index — RSCI), Scopus и MathSciNet, 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*