МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2015 **№ 3(29)**

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

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Адрес редакции: 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 15.09.2015.
Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 12,8. Уч.-изд. л. 14,2. Тираж 300 экз. Заказ № 1282.

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 519.7

О НИЖНИХ ОЦЕНКАХ СЛОЖНОСТИ ФУНКЦИЙ МНОГОЗНАЧНОЙ ЛОГИКИ НАД БЕСКОНЕЧНЫМИ БАЗИСАМИ[1]

А. А. Андреев

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

Рассматривается задача о сложности и глубине реализации функций многозначной логики формулами и схемами из функциональных элементов над бесконечными неполными базисами. Приводятся примеры бесконечных базисов, допускающих высокие (в том числе сверхэкспоненциальные) нижние оценки сложности.

Ключевые слова: *функции многозначной логики, бесконечные базисы, неполные базисы, сверхэкспоненциальные оценки сложности формул, экспоненциальные оценки глубины.*

DOI 10.17223/20710410/29/1

ON LOWER BOUNDS FOR COMPLEXITY OVER INFINITE BASISES FOR FUNCTIONS OF MULTI-VALUED LOGIC

A. A. Andreev

Lomonosov Moscow State University, Moscow, Russia

E-mail: sanchez_14@mail.ru

The complexity and the depth of multi-valued logic functions realization by formulas and by circuits of functional gates over infinite incomplete basises are estimated. Some examples of infinite basises allowing high (including overexponential) lower bounds for complexity are presented.

Keywords: *functions of multi-valued logic, infinite basises, incomplete basises, overexponential complexity bounds, exponential depth bounds.*

В работе исследуется задача получения высоких нижних оценок различных мер сложности реализации функций многозначной логики из замкнутых классов над бесконечными базисами, порождающими эти классы. Под базисом понимается любая порождающая система, не обязательно полная и не обязательно минимальная по включению. Бесконечным базисом называется базис, содержащий функции, существенно зависящие от сколь угодно большого числа переменных.

[1]Работа выполнена при поддержке РФФИ (проект № 14-01-00598-а) и Программы фундаментальных исследований ОМН РАН «Алгебраические и комбинаторные методы математической кибернетики и информационные системы нового поколения» (проект «Задачи оптимального синтеза управляющих систем»).

В качестве основного модельного класса управляющих систем рассматриваются формулы [1], однако в некоторых случаях рассуждения проводятся и для схем из функциональных элементов [2]. В качестве мер сложности рассматриваются глубина и собственно сложность. Переменные и константы для удобства также будем считать формулами (будем называть их тривиальными). Под сложностью формулы понимается количество символов переменных и констант, входящих в формулу, под сложностью схемы из функциональных элементов (далее просто схемы) — число функциональных элементов в ней. Понятие глубины $D(F)$ формулы F определим индуктивно. Если формула F тривиальная, то $D(F) = 0$, а если $F = G(F_1, \ldots, F_m)$, где G — некоторая функция, то $D(F) = \max D(F_i) + 1$, где максимум берётся по всем $i = 1, \ldots, m$. Сложностью $L^{\Phi}_{\mathfrak{B}}(f)$ функции f при реализации формулами над базисом $\mathfrak{B}$ называется минимальная сложность формулы над базисом $\mathfrak{B}$, реализующей эту функцию. Аналогично определяются сложность $L^{\text{СФЭ}}_{\mathfrak{B}}(f)$ функции f при реализации схемами над базисом $\mathfrak{B}$ и глубина функции $D_{\mathfrak{B}}(f)$ (понятие глубины функции не отличается для случаев реализации схемами и формулами). Функцией Шеннона $L^{\Phi}_{\mathfrak{B}}(n)$ назовём максимальную сложность реализации формулами над базисом $\mathfrak{B}$ функций от n переменных из замыкания $[\mathfrak{B}]$ базиса $\mathfrak{B}$. Аналогично определим функцию Шеннона $L^{\text{СФЭ}}_{\mathfrak{B}}(n)$ сложности реализации функций из класса $[\mathfrak{B}]$ схемами над базисом $\mathfrak{B}$ и функцию Шеннона глубины $D_{\mathfrak{B}}(n)$ реализации функций из класса $[\mathfrak{B}]$ над базисом $\mathfrak{B}$.

Асимптотика роста функций Шеннона $L^{\text{СФЭ}}_{\mathfrak{B}}(n)$, $L^{\Phi}_{\mathfrak{B}}(n)$ и $D_{\mathfrak{B}}(n)$ над произвольным полным конечным базисом булевых функций установлена О. Б. Лупановым [3–5]. Для всякого конечного полного базиса $\mathfrak{B}$ функция Шеннона $L^{\text{СФЭ}}_{\mathfrak{B}}(n)$ сложности реализации булевых функций схемами над этим базисом при $n \to \infty$ растёт по порядку как $2^n/n$, сложности реализации формулами $L^{\Phi}_{\mathfrak{B}}(n)$ — как $2^n/\log n$, а функция Шеннона глубины $D_{\mathfrak{B}}(n)$ — как n.

При переходе от конечного полного базиса к бесконечному полному базису булевых функций порядки роста введённых функций Шеннона понижаются. Из результатов О. Б. Лупанова непосредственно следует, что для любых полных базисов $\mathfrak{B}_1$ и $\mathfrak{B}_2$, где $\mathfrak{B}_1$ — конечный, а $\mathfrak{B}_2$ — бесконечный, справедливы соотношения $L^{\text{СФЭ}}_{\mathfrak{B}_2}(n) = o(L^{\text{СФЭ}}_{\mathfrak{B}_1}(n))$, $L^{\Phi}_{\mathfrak{B}_2}(n) = o(L^{\Phi}_{\mathfrak{B}_1}(n))$, $D_{\mathfrak{B}_2}(n) = o(D_{\mathfrak{B}_1}(n))$. На самом деле, О. М. Касим-Заде установлено [6, 7], что при переходе от конечных к бесконечным полным базисам булевых функций порядок роста функций Шеннона $L^{\text{СФЭ}}_{\mathfrak{B}}(n)$ и $D_{\mathfrak{B}}(n)$ меняется качественно: в случае бесконечных базисов $L^{\text{СФЭ}}_{\mathfrak{B}}(n)$ растёт по порядку как $2^{n/2}$ или медленнее, а $D_{\mathfrak{B}}(n)$ имеет порядок роста не больше $\log n$. В случае реализации функций k-значной логики при $k \geqslant 3$ для функции Шеннона глубины при переходе от конечных полных к бесконечным полным базисам имеет место аналогичное понижение порядка роста. Для любого конечного базиса $\mathfrak{B}$ функций k-значной логики ($k \geqslant 3$) функция Шеннона глубины $D_{\mathfrak{B}}(n)$ при $n \to \infty$ растёт по порядку как линейная функция (для этого случая также известна асимптотика [8]). Для бесконечных базисов многозначной логики известно [9], что функция Шеннона глубины $D_{\mathfrak{B}}(n)$ растёт по порядку не быстрее, чем $\log n$. Стоит отметить, что при переходе от конечных базисов, порождающих замкнутые классы булевых функций, к бесконечным может также иметь место аналогичный эффект понижения порядков роста соответствующих функций Шеннона (см., например, [10]). Однако, к примеру, для класса линейных функций и в случае конечного, и в случае бесконечного базиса, порождающего этот класс, функция Шеннона сложности реализации формулами растёт по порядку как n.

В связи с этим возникает вопрос о возможности получения высоких нижних оценок сложности (для всех трёх мер) над бесконечными базисами, аналогичных известным [11–13] высоким нижним оценкам сложности в конечных неполных базисах функций k-значной логики, $k \geqslant 3$.

Стоит отметить, что в случае булевых функций все замкнутые классы конечно-порождённые. Это значит, что любой бесконечный базис будет избыточным и из него можно выделить конечную подсистему, порождающую тот же замкнутый класс. Следовательно, порядок роста при переходе от конечного базиса булевых функций к бесконечному не может увеличиться. В случае же функций многозначной логики ситуация иная. Существуют примеры замкнутых классов, не имеющих конечного базиса [14], что даёт дополнительные возможности для получения высоких нижних оценок в бесконечных базисах по сравнению с оценками в конечных базисах.

Покажем, как для произвольного конечного множества $\mathfrak{A}$ функций k-значной логики и произвольной последовательности функций ζ_n из класса $[\mathfrak{A}]$ построить бесконечный базис $\mathfrak{B}$ функций $(k+2)$-значной логики и предъявить последовательность функций $\{\eta_n\}$ из класса $[\mathfrak{B}]$, для которых справедливы равенства $L_{\mathfrak{A}}^{\text{СФЭ}}(\zeta_n) = L_{\mathfrak{B}}^{\text{СФЭ}}(\eta_n)$, $L_{\mathfrak{A}}^{\Phi}(\zeta_n) = L_{\mathfrak{B}}^{\Phi}(\eta_n)$, $D_{\mathfrak{A}}(\zeta_n) = D_{\mathfrak{B}}(\eta_n)$.

Пусть $E_k = \{0, \ldots, k-1\}$; $\mathfrak{A} = \{\varphi_1(x_1, \ldots, x_{i_1}), \ldots, \varphi_m(x_1, \ldots, x_{i_m})\}$ — конечная система функций k-значной логики, а $\zeta_1(x_1), \zeta_2(x_1, x_2), \ldots, \zeta_n(x_1, \ldots, x_n), \ldots$ — последовательность функций k-значной логики. Рассмотрим следующие функции $(k+2)$-значной логики:

$$\psi_i(x_1, \ldots, x_{l_i}) = \begin{cases} \varphi_i(x_1, \ldots, x_{l_i}), & \text{если } x_1, \ldots, x_{l_i} \in E_k, \\ k & \text{в остальных случаях;} \end{cases}$$

$$\eta_n(x_1, \ldots, x_n) = \begin{cases} \zeta_n(x_1, \ldots, x_n), & \text{если } x_1, \ldots, x_n \in E_k, \\ k & \text{в остальных случаях;} \end{cases}$$

$$\mu_n(x_1, \ldots, x_n) = \begin{cases} k, & \text{если } x_1 = k+1, \ldots, x_n = k+1, \\ k+1 & \text{в остальных случаях,} \end{cases}$$

где $i \in \{1, \ldots, m\}$; $n \in \mathbb{N}$.

Пусть $\mathfrak{B} = \{\psi_1, \ldots, \psi_m, \mu_1, \mu_2, \ldots\}$, $\mathfrak{B}' = \{\psi_1, \ldots, \psi_m\}$. Покажем, что все рассматриваемые меры сложности реализации функций ζ_n над базисом $\mathfrak{A}$ равны соответствующим мерам сложности реализации функций η_n над базисом $\mathfrak{B}$.

Если в некоторой формуле над системой $\mathfrak{B}$ для некоторого n есть хотя бы один элемент вида μ_n, то очевидно, что такая формула на всех наборах переменных может принимать только значения k и $k+1$. В то же время функции η_n при некоторых значениях переменных принимают значения, отличные от этих двух. Значит, сложность реализации функций η_n над системой $\mathfrak{B}$ не изменится, если из этой системы убрать функции μ_n (тем самым получив систему $\mathfrak{B}'$).

Возьмем формулу над системой $\mathfrak{A}$, реализующую функцию $\zeta_n(x_1, \ldots, x_n)$, и для всех $i \in \{1, \ldots, m\}$ заменим в ней все вхождения функций $\varphi_i(A_1, \ldots, A_{l_i})$ на функции $\psi_i(A_1, \ldots, A_{l_i})$ ($A_1, \ldots, A_{l_i}$ — формулы). Очевидно, что мы получим формулу над системой $\mathfrak{B}'$, реализующую функцию $\eta_n(x_1, \ldots, x_n)$. И наоборот, если заменить все функции $\psi_1, \ldots, \psi_m$ на функции $\varphi_1, \ldots, \varphi_m$, то из формулы, реализующей функцию η_n, получим формулу, реализующую функцию ζ_n. Таким образом, доказаны следующие соотношения: $L_{\mathfrak{A}}^{\Phi}(\zeta_n) = L_{\mathfrak{B}}^{\Phi}(\eta_n)$, $D_{\mathfrak{A}}(\zeta_n) = D_{\mathfrak{B}}(\eta_n)$. Аналогичным рассуждением получаем, что $L_{\mathfrak{A}}^{\text{СФЭ}}(\zeta_n) = L_{\mathfrak{B}}^{\text{СФЭ}}(\eta_n)$.

Стоит отметить, что класс, порождённый построенным базисом $\mathfrak{B}$, является конечно-порождённым. Однако не представляет особой трудности привести аналогичный пример класса, не имеющего конечного базиса (возможно, с увеличением значности логики ещё на единицу). Для этого достаточно в качестве функций μ_n взять аналог функций из примера в [14].

Если в качестве $\mathfrak{A}$ взять систему функций k-значной логики из [13], то приведённым способом будет построен бесконечный базис, сложность реализации некоторой последовательности функций над которым растёт сверхэкспоненциальным образом. Более точно, для соответствующей последовательности $\{\eta_n\}$ функций $(k+2)$-значной логики справедливо равенство $L_{\mathfrak{B}}^{\Phi}(\eta_n) = (n+1)\cdot 2^{n((k-3)^n-(k-4)^n)} - n$, где функция ζ_n зависит от $n+1$ переменной. В этом же примере для глубины верна следующая формула: $D_{\mathfrak{B}}(\eta_n) = (k-3)^n - (k-4)^n + n - 1$. Если же взять в качестве $\mathfrak{A}$ систему функций трёхзначной логики из [11], то для сложности соответствующей функции пятизначной логики η_n в бесконечном базисе $\mathfrak{B}$ справедливо равенство $L_{\mathfrak{B}}^{\text{СФЭ}}(\eta_n) = 2C_n^{[n/2]} - 1$.

Недостаток приведённого метода в том, что порождающую систему можно разделить на две практически не связанные части: одна доставляет необходимую оценку, а вторая обеспечивает бесконечность системы. Это делает систему сильно искусственной. Построим пример бесконечной системы функций с высокими нижними оценками сложности, лишённый этого и некоторых других недостатков. Для этого определим два базиса функций k-значной логики ($k \geqslant 5$): конечный $\mathfrak{A}$ и бесконечный $\mathfrak{B}$, удовлетворяющие следующим условиям:

1) $[\mathfrak{A}] = [\mathfrak{B}]$ (и, следовательно, класс $[\mathfrak{B}]$ конечно порождён);
2) функции Шеннона глубины и сложности реализации формулами в этих базисах асимптотически равны (и растут соответственно экспоненциально и сверхэкспоненциально);
3) каждая функция базиса $\mathfrak{B}$ используется хотя бы в одной минимальной формуле, реализующей функцию, на которой достигается значение функций Шеннона.

Для описания такого примера сначала рассмотрим пример из работы [13]. Обозначим через E_k^n ($n \geqslant 1$) множество всех наборов $(\alpha_1, \ldots, \alpha_n)$, таких, что $\alpha_1, \ldots, \alpha_n \in E_k$, а через Q_n множество всех наборов из E_k^n, состоящих только из символов $3, \ldots, k-1$, причём хотя бы один символ 3 должен быть обязательно. Определим функции k-значной логики $\lambda(x, y)$, $\mu(x, y, z)$, $\varphi_m(x, y)$ и $\zeta_n(y, x_1, \ldots, x_n)$, где $m \in \{3, \ldots, k-1\}$, $n \in \mathbb{N}$, следующим образом:

$$\lambda(x,y) = \begin{cases} 0, & \text{если } x = 0,\ y = 2, \\ 1, & \text{если } x = 0,\ y = 3 \text{ или если } x = 1, \\ 2 & \text{в остальных случаях;} \end{cases}$$

$$\mu(x,y,z) = \begin{cases} \lambda(x,z), & \text{если } x = y, \\ 2 & \text{в противном случае;} \end{cases}$$

$$\varphi_m(x,y) = \begin{cases} 3, & \text{если } x = 3,\ y = m, \\ 2 & \text{в остальных случаях;} \end{cases}$$

$$\zeta_n(y, x_1, \ldots, x_n) = \begin{cases} 0, & \text{если } y = 0,\ (x_1, \ldots, x_n) \notin Q_n, \\ 1, & \text{если } y = 0,\ (x_1, \ldots, x_n) \in Q_n \text{ или если } y = 1, \\ 2 & \text{в остальных случаях.} \end{cases}$$

Справедливо следующее утверждение.

Теорема 1 [13]. Пусть $\mathfrak{B} = \{\mu, \varphi_3, \ldots, \varphi_{k-1}, 2\}$, $n \geqslant 1$, $k \geqslant 4$. Тогда для функции k-значной логики ζ_n справедливо равенство

$$L^{\Phi}_{\mathfrak{B}}(\zeta_n) = (n+1) \cdot 2^{n((k-3)^n - (k-4)^n)} - n.$$

При доказательстве этой теоремы в [13] установлено, что любая формула F, реализующая функцию ζ_n, устроена строго определённым образом, а именно: формула F получается из некоторой формулы

$$G = \lambda(\lambda(\ldots\lambda(\lambda(y, Z_1), Z_2), \ldots), Z_N), \tag{1}$$

где $Z_1, \ldots, Z_N$ — формулы над $\mathfrak{B}$; N — натуральное, заменой всех подформул вида $\lambda(A, B)$ формулами $\mu(A, A, B)$. Во всех минимальных формулах над базисом $\mathfrak{B}$, реализующих функцию ζ_n, подформулы Z_i также устроены строго определённым образом: они все имеют вид $\varphi_{m_1}(\ldots\varphi_{m_s}(H_{s+1}, H_s), \ldots, H_1)$, где $H_1, \ldots, H_{s+1}$ — различные переменные из множества $\{x_1, \ldots, x_n\}$; каждая переменная из этого множества встречается в формуле Z_i как минимум один раз. Кроме того, установлено, что количество N таких подформул Z_i в формуле, реализующей функцию ζ_n, удовлетворяет соотношению $N \geqslant (k-3)^n - (k-4)^n$.

Особенности строения формул, описанные выше, поясним, приведя ряд свойств формул над базисами $\mathfrak{B}$ и $\mathfrak{A}$ из работы [13], где $\mathfrak{A} = \mathfrak{B} \cup \{\lambda(x, y)\}$. Для этого введём следующие определения и обозначения.

Пусть $F(y, x_1, \ldots, x_n)$ — некоторая формула над $\mathfrak{A}$. Поставим в соответствие формуле F дерево T с корнем v_*, в котором висячим вершинам приписаны символы из множества $\{x_1, \ldots, x_n, y, 2\}$. Между вершинами дерева T и подформулами формулы F естественным образом устанавливается взаимно однозначное соответствие, в частности, корневой вершине v_* соответствует формула F, главным подформулам формулы F — вершины, смежные с корневой, и так далее, висячим вершинам — выражения вида x_i, y или 2, $1 \leqslant i \leqslant n$. Если A — некоторая подформула формулы F, а v — некоторая вершина дерева T, то через v_A будем обозначать вершину дерева T, соответствующую подформуле A, а через A_v — подформулу формулы F, соответствующую вершине v.

Раскрасим рёбра и вершины дерева T в белый, чёрный и зелёный цвета следующим образом. Пусть A — некоторая нетривиальная подформула формулы F. Если она имеет вид $A = \lambda(B, C)$, где B, C — формулы над $\mathfrak{A}$, то раскрасим ребро (v_A, v_B) в белый цвет, а ребро (v_A, v_C) в чёрный. Если A — формула вида $A = \mu(B, C, D)$, где B, C, D — формулы, то раскрасим рёбра $(v_A, v_B), (v_A, v_C)$ в белый цвет, а ребро (v_A, v_D) в чёрный. И наконец, если A — формула вида $\varphi_m(B, C)$, где B, C — формулы, $m \in \{3, \ldots, k-1\}$, то раскрасим ребро (v_A, v_B) в чёрный цвет, а ребро (v_A, v_C) в зелёный. Корень v_* дерева T раскрашивается в белый цвет. Вершина v, отличная от корня, раскрашивается в белый цвет, если все рёбра цепи, соединяющей корень v_* дерева T с вершиной v, раскрашены в белый цвет. Вершина v раскрашивается в чёрный цвет, если все рёбра цепи, соединяющей корень v_* дерева T с вершиной v, раскрашены в белый или чёрный цвет, причем чёрные есть обязательно. Все остальные вершины раскрашиваются в зелёный цвет. Будем говорить, что подформула формулы F белого, чёрного или зелёного цвета, если сопоставленная ей вершина дерева T раскрашена в белый, чёрный или зелёный цвет соответственно. Обозначим через $V(F)$ множество всех висячих вершин дерева T, раскрашенных в белый цвет, а через $W(F)$ — множество символов, соответствующих вершинам из $V(F)$. Очевидно, что имеет место соотношение $W(F) \subseteq \{x_1, \ldots, x_n, y, 2\}$.

Нетрудно получить (аналогично [15]) следующие свойства формулы F:

1. Для любой белой подформулы A формулы F и любых наборов $\alpha, \beta \in E^{n+1}$, таких, что $F(\alpha) = 0$, $A(\beta) = 2$, выполняются соотношения $A(\alpha) = 0$, $F(\beta) = 2$.
2. Пусть формула F реализует функцию $\zeta_n(y, x_1, \ldots, x_n)$, A — произвольная белая подформула формулы F, $\beta = (\beta_0, \ldots, \beta_n)$ — произвольный набор из E^{n+1}. Тогда выполняются следующие соотношения:
   а) $W(F) = \{y\}$;
   б) если $A(\alpha) = 0$, то $\alpha_0 = 0$;
   в) если $\beta_0 = 1$, то $A(\beta) = 1$;
   г) формула A имеет вид $\lambda(B, C)$ или $\mu(B, C, D)$, где B, C, D — формулы.

Теперь зафиксируем некоторое k и построим базис $\mathfrak{C} \subseteq P_k$ (где P_k — множество функций k-значной логики). Для этого возьмём базис $\mathfrak{B} \subseteq P_k$ и для всех натуральных n для всех минимальных формул над $\mathfrak{B}$ для функций ζ_n для всех подформул Z_i из (1) добавим в базис функции, реализуемые формулами Z_i (то есть все функции вида $\varphi_{m_1}(\ldots \varphi_{m_s}(H_{s+1}, H_s), \ldots, H_1)$, где $H_1, \ldots, H_{s+1}$ — различные переменные из множества $x_1, \ldots, x_n$). Получается, что бесконечный базис $\mathfrak{C}$ порождает тот же замкнутый класс, что и базис $\mathfrak{B}$, и при реализации функций ζ_n каждая функция базиса используется хотя бы в одной минимальной формуле (минимальные формулы в базисе $\mathfrak{C}$ — это те же самые формулы, которые являются минимальными в базисе $\mathfrak{B}$, с возможной заменой подформул над $\{\varphi_3, \ldots, \varphi_{k-1}\}$ имеющейся в базисе функцией их суперпозиции), при этом формулы всё так же имеют описанные выше особенности строения. Это значит, что справедливо следующее утверждение.

Теорема 2. Пусть $\mathfrak{C}$ — описанный выше бесконечный базис, $n \geqslant 1$, $k \geqslant 4$. Тогда для функции k-значной логики ζ_n справедливо равенство

$$L^{\Phi}_{\mathfrak{C}}(\zeta_n) = (n+1) \cdot 2^{n((k-3)^n - (k-4)^n)} - n.$$

Отметим, что для глубины рассматриваемых функций можно записать соответствующие соотношения:

$$D_{\mathfrak{B}}(\zeta_n) = (k-3)^n - (k-4)^n + n - 1,$$
$$D_{\mathfrak{C}}(\zeta_n) = (k-3)^n - (k-4)^n + 1.$$

Кроме того, в замкнутом классе $[\mathfrak{B}] = [\mathfrak{C}]$ функции Шеннона глубины и сложности реализации формулами над базисом $\mathfrak{B}$ асимптотически равны соответствующим функциям Шеннона для формул над базисом $\mathfrak{C}$. Поясним этот факт.

Сначала отметим несколько очевидных свойств формул над базисом $\mathfrak{B}$:

1) если в некоторой формуле функция μ подаётся на вход функции φ_m для некоторого m, то при замене этой функции μ константой 2 реализуемая формулой функция не изменится;
2) если в некоторой формуле функция φ_m для некоторого m подаётся на вход функции μ не в качестве последнего аргумента, то при замене этой функции φ_m константой 2 реализуемая формулой функция не изменится;
3) для произвольных формул A и B справедливы равенства

$$\varphi_m(A, 2) = \varphi_m(2, A) = 2, \quad \mu(2, A, B) = \mu(A, 2, B) = 2.$$

Таким образом, для всех функций из класса $[\mathfrak{B}]$ минимальные формулы могут иметь один из следующих видов: константа 2; формула над системой $\{\varphi_3, \ldots, \varphi_{k-1}\}$;

формула, в которой все чёрные подформулы имеют один из двух предыдущих видов, а каждая белая подформула либо имеет вид $\mu(A, B, C)$, где A, B, C — формулы, либо является переменной.

Аналогичные свойства справедливы и для базиса $\mathfrak{C}$ (мы добавили в базис только комбинации уже имеющихся функций), и минимальные формулы также имеют один из вышеперечисленных видов (с поправкой на то, что наряду с функциями $\{\varphi_3, \ldots, \varphi_{k-1}\}$ присутствуют и их суперпозиции). Это значит, что для любой функции из класса $[\mathfrak{B}]$ по любой её минимальной формуле над базисом $\mathfrak{C}$ очевидным образом получается минимальная формула для этой же функции над базисом $\mathfrak{B}$. Поскольку с точки зрения функции Шеннона нас, очевидно, интересуют функции третьего вида, можно сделать вывод, что асимптотика функций Шеннона в базисах $\mathfrak{B}$ и $\mathfrak{C}$ одинакова.

Таким образом, приведён пример конечно-порождённого класса, а также конечного и бесконечного базисов, его порождающих, для которых совпадают асимптотики роста функций Шеннона глубины и сложности реализации формулами.

Теперь построим замкнутые классы функций многозначной логики, не являющиеся конечно-порождёнными, в которых для всех типов рассматриваемых функций Шеннона при соответствующем выборе базиса (очевидно, бесконечного) справедливы нижние оценки с более высоким порядком роста, чем у известных [11 – 13] соответствующих нижних оценок в случае конечно-порождённых классов.

Пусть $\varphi(x)$ — функция трёхзначной логики, принимающая значение 1 при $x = 2$ и значение 0 в остальных случаях; $\Omega(\tilde{x})$ — отображение из E_3^n в E_2^n, которое набору $(x_1, \ldots, x_n)$ сопоставляет набор $(\varphi(x_1), \ldots, \varphi(x_n))$; $\tilde{\alpha}$ — набор из E_2^n. Определим функцию $\lambda_{\tilde{\alpha}}^3(y, x_1, \ldots, x_n)$ из P_3 следующим образом:

$$\lambda_{\tilde{\alpha}}^3(y, x_1, \ldots, x_n) = \begin{cases} 0, & \text{если } y = 0 \text{ и } \Omega(\tilde{x}) \neq \tilde{\alpha}, \text{ или если } y = 2, \\ 1, & \text{если } y = 1, \text{ или если } y = 0 \text{ и } \Omega(\tilde{x}) = \tilde{\alpha}. \end{cases}$$

Покажем, что для формул специального вида справедливы определённые свойства.

1. Пусть формула F выглядит следующим образом: $F = \lambda_{\tilde{\alpha}}^3(A_0, \ldots, A_n)$, где $A_0, \ldots, A_n$ — формулы, и для некоторого $i \in \{1, \ldots, n\}$ формула A_i имеет вид $A_i = \lambda_{\tilde{\beta}}^3(B_0, \ldots, B_l)$. Тогда при замене формулы A_i на константу 0 функция, реализуемая формулой F, не изменится.

Действительно, формула A_i принимает значения 0 и 1. При этом равенство $\varphi(x_i) = \alpha_i$ выполняется при $x_i = 1$ тогда и только тогда, когда оно выполняется и при $x_i = 0$. Значит, при замене формулы A_i константой 0 значение формулы F не изменится.

2. Пусть формула $F = \lambda_{\tilde{\alpha}}^3(0, x_{i_1}, \ldots, x_{i_m})$, где $i_1, \ldots, i_m \in \{1, \ldots, n\}$, реализует константу 0. Тогда верно равенство $\lambda_{\tilde{\alpha}}^3(\lambda_{\tilde{\beta}}^3(A_0, \ldots, A_l), x_{i_1}, \ldots, x_{i_m}) = \lambda_{\tilde{\beta}}^3(A_0, \ldots, A_l)$, где $A_0, \ldots, A_l$ — формулы.

Действительно, если на некотором наборе формула $\lambda_{\tilde{\beta}}^3(A_0, \ldots, A_l)$ принимает значение 1, то и формула $\lambda_{\tilde{\alpha}}^3(\lambda_{\tilde{\beta}}^3(A_0, \ldots, A_l), x_{i_1}, \ldots, x_{i_m})$ также принимает значение 1. Если на этом наборе формула $\lambda_{\tilde{\beta}}^3(A_0, \ldots, A_l)$ равна нулю, то в силу тождества $\lambda_{\tilde{\alpha}}^3(0, x_{i_1}, \ldots, x_{i_m}) \equiv 0$ имеем $\lambda_{\tilde{\alpha}}^3(\lambda_{\tilde{\beta}}^3(A_0, \ldots, A_l), x_{i_1}, \ldots, x_{i_m}) = 0$ на этом наборе. Значение 2 ни одна из этих формул принимать не может.

Пусть $R_2 = \bigcup\limits_{n \in \mathbb{N}} E_2^n$. Из свойства 1 следует, что в семействе формул над базисом, состоящим из константы 0 и функций $\lambda_{\tilde{\alpha}}^3$, где $\tilde{\alpha} \in R_2$, в минимальных по сложности фомулах все подформулы могут иметь нетривиальные подформулы только в качестве первого аргумента, то есть в таких базисах все минимальные формулы «вытянуты в цепочку».

Определим функции трёхзначной логики $\zeta_n^3(x_1, \ldots, x_n)$, $n \geqslant 1$, следующим образом. Функция ζ_n^3 принимает значение 0, если среди её аргументов чётное количество двоек, и значение 1, если нечётное. Рассмотрим неполный бесконечный базис $\mathfrak{A} = \{0\} \cup \{\lambda_{\tilde{\alpha}}^3 : \tilde{\alpha} \in R_2\}$. Докажем следующее утверждение.

Теорема 3. Для функции трёхзначной логики ζ_n^3, $n \geqslant 1$, выполняется равенство

$$D_{\mathfrak{A}}(\zeta_n^3) = 2^{n-1}.$$

Доказательство. Сначала отметим, что $\zeta_n^3 \in [\mathfrak{A}]$. В процессе доказательства формула над $\mathfrak{A}$, реализующая указанную функцию, будет естественным образом получена.

Пусть формула F реализует функцию ζ_n^3. Если у некоторой её подформулы есть нетривиальные подформулы не в качестве первого аргумента, то заменим их на константу 0. Согласно свойству 1, реализуемая формулой функция от этого не поменяется, а глубина и сложность не увеличатся. В дальнейшем считаем, что у всех подформул все аргументы, кроме первого, тривиальны.

Итак, формула F имеет вид

$$\begin{aligned} F &= \lambda_{\tilde{\alpha}_1}^3(A_{1,0}, A_{1,1}, \ldots, A_{1,m_1}), \\ A_{1,0} &= \lambda_{\tilde{\alpha}_2}^3(A_{2,0}, A_{2,1}, \ldots, A_{2,m_2}), \\ &\ldots \\ A_{d-1,0} &= \lambda_{\tilde{\alpha}_d}^3(A_{d,0}, A_{d,1}, \ldots, A_{d,m_d}), \end{aligned} \tag{2}$$

где $A_{1,1}, \ldots, A_{1,m_1}, \ldots, A_{d,1}, \ldots, A_{d,m_d}$ и $A_{d,0}$ — тривиальные формулы.

Из определения функций $\lambda_{\tilde{\alpha}}^3$ следует, что если на некотором наборе формула $A_{i,0}$ $(i \in \{1, \ldots, d\})$ принимает значение 1, то и формула F на нём также принимает значение 1. Из определения также следует, что если на некотором наборе формула F принимает значение 0, то и для всех $i \in \{1, \ldots, d-1\}$ формулы $A_{i,0}$ также принимают значение 0.

Пусть формула $A_{d,0}$ имеет вид x_i для некоторого i. Тогда на наборе $(1, 1, \ldots, 1)$ функция $\zeta_n^3(x_1, \ldots, x_n)$ принимает значение 0, а формула F — значение 1. Следовательно, $A_{d,0}$ представляет из себя константу 0.

Рассмотрим формулы $B_1 = \lambda_{\tilde{\alpha}_1}^3(0, A_{1,1}, \ldots, A_{1,m_1}), \ldots, B_d = \lambda_{\tilde{\alpha}_d}^3(0, A_{d,1}, \ldots, A_{d,m_d})$. Если какая-нибудь из них реализует константу 0, то, согласно свойству 2, можно удалить соответствующую ей подформулу из «цепочки» (2). При этом реализуемая функция не изменится, глубина формулы не увеличится, а сложность уменьшится. Проделаем такую операцию со всеми подформулами, реализующими ноль. Далее будем считать, что все формулы B_i не реализуют константу 0, а значит, равны единице на некотором наборе (значения 2 они принимать не могут). Для формулы B_i обозначим такой набор $\tilde{\sigma}_i$. Заметим, что в силу показанных ранее соотношений формула F на всех наборах $\tilde{\sigma}_i$ $(i = 1, \ldots, d)$ принимает значение 1.

Пусть найдутся такие числа i и j, что в формулу B_i не входит переменная x_j $(i \in \{1, \ldots, d\}$, $j \in \{1, \ldots, n\})$. Тогда в наборе $\tilde{\sigma}_i$ заменим j-ю компоненту: если она равнялась двум, то заменим её нулём, а если нет, то заменим её двойкой. В результате получим набор $\tilde{\sigma}_i'$. С одной стороны, у этого набора и у набора $\tilde{\sigma}_i$ разная чётность количества двоек, и, следовательно, функция ζ_n^3 принимает на них разные значения.

С другой стороны, эти наборы отличаются только в j-й компоненте. А так как переменная x_j не входит в формулу B_i, то $B_i(\tilde{\sigma}_i) = B_i(\tilde{\sigma}'_i)$. Значит, $B_i(\tilde{\sigma}'_i) = 1$ и $F(\tilde{\sigma}'_i) = 1$. Полученное противоречие показывает, что во всех формулах B_i присутствуют все переменные $x_1, \ldots, x_n$ в качестве подформул.

Согласно определению функции $\lambda^3_{\tilde{\alpha}}$, все наборы, на которых B_i принимает значение 1, отображением Ω переводятся в один и тот же набор из P_2 (если переменные упорядочены и встречаются по одному разу, то этот набор — $\tilde{\alpha}$). Рассмотрим все наборы, на которых функция ζ^3_n принимает значение 1. При отображении Ω эти наборы перейдут в 2^{n-1} наборов из P_2. Принимая во внимание то, что для равенства формулы F единице на некотором наборе необходимо, чтобы на этом наборе хотя бы одна из формул B_i принимала значение 1, получаем, что количество d формул B_i удовлетворяет неравенству $d \geqslant 2^{n-1}$.

В качестве примера формулы глубины 2^{n-1}, реализующей функцию ζ^3_n, можно привести следующую: $m_1 = \ldots = m_d = n$, $A_{i,j} = x_j$ для всех $i \in \{1, 2, \ldots, d\}$, $j \in \{1, 2, \ldots, n\}$, $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_d$ — это все наборы длины n с чётным количеством единиц. ■

Заметим также, что сложность реализации функции схемами над некоторым базисом не меньше глубины этой функции над этим базисом. Поэтому приведённая выше формула доставляет пример минимальной схемы над базисом $\mathfrak{A}$ для функции ζ^3_n, т. е.

$$L^{\text{СФЭ}}_{\mathfrak{A}}(\zeta^3_n) = 2^{n-1},$$

причём полученное значение сложности превосходит максимальные нижние оценки, известные [11] для случая конечных базисов.

Покажем теперь, как на основе этого примера построить бесконечный базис в P_4 и последовательность функций, сложность реализации которых формулами над этим базисом растёт сверхэкспоненциально (и быстрее известных рекордных высоких оценок для случая конечно-порождённых классов P_4).

Пусть $\varphi(x)$ — функция четырёхзначной логики, принимающая значение 1 при $x = 2$ и значение 0 в остальных случаях; $\Omega(\tilde{x})$ — отображение из E^n_4 в E^n_2, которое набору $(x_1, \ldots, x_n)$ сопоставляет набор $(\varphi(x_1), \ldots, \varphi(x_n))$; $\tilde{\alpha}$ — набор из E^n_2. Определим функции $\lambda^4_{\tilde{\alpha}}(y, x_1, \ldots, x_n)$ и $\mu_{\tilde{\alpha}}(x_1, \ldots, x_{n+2})$ из P_4 следующим образом:

$$\lambda^4_{\tilde{\alpha}}(y, x_1, \ldots, x_n) = \begin{cases} \lambda^3_{\tilde{\alpha}}(y, x_1, \ldots, x_n), & \text{если } (y, x_1, \ldots, x_n) \in E^n_3, \\ 3 & \text{иначе;} \end{cases}$$

$$\mu_{\tilde{\alpha}}(x_1, \ldots, x_{n+2}) = \begin{cases} 3, & \text{если } x_i = 3 \text{ для некоторого} \\ & i \in \{1, \ldots, n+2\}, \text{ или если } x_1 \neq x_2, \\ \lambda^4_{\tilde{\alpha}}(x_2, \ldots, x_{n+2}) & \text{иначе.} \end{cases}$$

Определим последовательность функций четырёхзначной логики $\zeta^4_n(x_1, \ldots, x_n)$, $n \geqslant 1$, следующим образом. Если среди аргументов функции есть хотя бы одна тройка, то она принимает значение 3. Если же нет, то функция ζ^4_n принимает значение 0, если среди её аргументов чётное количество двоек, и значение 1, если нечётное. Рассмотрим неполный бесконечный базис $\mathfrak{A} = \{0\} \cup \{\mu_{\tilde{\alpha}} : \tilde{\alpha} \in R_2\}$. Докажем следующее утверждение.

Теорема 4. *Для функции четырёхзначной логики ζ^4_n, $n \geqslant 1$, верно равенство*

$$L^{\Phi}_{\mathfrak{A}}(\zeta^4_n) = 2^{2^{n-1}} - 1.$$

Доказательство. Отметим, что функция ζ_n^4 лежит в замыкании системы $\mathfrak{A}$, и соответствующая формула для неё будет построена в процессе доказательства.

Очевидно, что если каждая переменная участвует в построении формулы, то условие равенства формулы тройке на всех наборах, содержащих тройку, выполняется всегда. Заметим также, что если какая-то подформула на некотором наборе принимает значение 3, то и вся формула на этом наборе также принимает значение 3. Это следует из определения функций $\mu_{\tilde{\alpha}}$.

Пусть F — формула над $\mathfrak{A}$, реализующая функцию $\zeta_n^4(x_1, \ldots, x_n)$. Преобразуем формулу F без увеличения её сложности и глубины и одновременно построим по формуле F формулу G специального вида. Изначально возьмём $G = F$. Пусть формула F имеет вид $A = \mu_{\tilde{\alpha}}(A_1, \ldots, A_{n+2})$ (для некоторого $\tilde{\alpha}$). На всех наборах, не содержащих троек, формулы A_1 и A_2 принимают одинаковые значения. Пусть без ограничения общности $L_{\mathfrak{A}}^{\Phi}(A_1) \geqslant L_{\mathfrak{A}}^{\Phi}(A_2)$. Тогда в формуле F заменим A выражением $\mu_{\tilde{\alpha}}(A_2, A_2, \ldots, A_{n+2})$, а в формуле G — выражением $\lambda_{\tilde{\alpha}}^4(A_2, \ldots, A_{n+2})$. На наборах, не содержащих троек, значения формул от такой замены не поменяются. Далее проведём подобную замену для всех подформул формулы F (и соответственно G), затем для всех их подформул, и так далее, пока формула G не перестанет содержать функции вида $\mu_{\tilde{\alpha}}$ (для всех возможных $\tilde{\alpha}$).

Теперь заметим, что поскольку получившиеся формулы на всех наборах, не содержащих троек, принимают значения, равные значениям ζ_n^4 на этих же наборах, и значения на этих наборах зависят от всех переменных, то в формулы F и G каждая переменная входит в качестве подформулы. Это значит, что и на всех наборах, содержащих тройки, значения получившихся формул и функции ζ_n^4 также равны. При ограничении на наборы без троек формула G реализует ранее определённую функцию трёхзначной логики и, следовательно, её глубина не меньше чем 2^{n-1}. Очевидно, что у формулы F глубина такая же, а сложность удовлетворяет соотношению $L_{\mathfrak{A}}^{\Phi}(F) = 2^{D_{\mathfrak{A}}(F)} - 1$.

Пример формулы G, для которой полученная оценка достигается, такой же, как и в теореме 3. Формула F получается из неё тривиальной заменой $\lambda_{\tilde{\alpha}}^4(A_1, \ldots, A_{n+1})$ на $\mu_{\tilde{\alpha}}(A_1, A_1, A_2, \ldots, A_{n+1})$. Легко видеть, что построенная таким образом формула реализует функцию ζ_n^4 ■

Отметим, что установленный в теореме 4 рост сложности реализации последовательности функций ζ_n^4 формулами над бесконечным базисом $\mathfrak{A}$, порождающий замкнутый класс в P_4, не имеющий конечного базиса, превосходит известную [12] рекордную оценку сложности реализации последовательности функций над конечным базисом в P_4, имеющую вид

$$2^{C_n^{[n/2]}}([n/2] + 1) - [n/2].$$

Конструкции из доказательств теоремы 3 (для функций трёхзначной логики) и теоремы 4 (для функций четырёхзначной логики) могут быть обобщены на случай функций k-значной логики для бо́льших значений k. При $k > 3$, $n \geqslant 1$ можно привести пример замкнутого класса, не имеющего конечного базиса, и последовательности лежащих в нём функций от n переменных, глубина (и сложность реализации схемами) которых над некоторым бесконечным базисом, порождающим этот класс, равна $(k-1)^{n-1}$. При $k > 4$, $n \geqslant 1$ можно привести пример замкнутого класса, не имеющего конечного базиса, и последовательности лежащих в нём функций от n переменных, сложность реализации формулами которых над некоторым бесконечным базисом, порождающим этот класс, равна $2^{(k-2)^{n-1}} - 1$.

ЛИТЕРАТУРА

1. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986. 384 с.
2. *Лупанов О. Б.* Асимптотические оценки сложности управляющих систем. М.: Изд-во МГУ, 1984. 136 с.
3. *Лупанов О. Б.* Об одном методе синтеза схем // Изв. вузов. Радиофизика. 1958. Т. 1. № 1. С. 120–140.
4. *Лупанов О. Б.* О сложности реализации функций алгебры логики формулами // Проблемы кибернетики. Вып. 3. М.: Физматгиз, 1960. С. 61–80.
5. *Лупанов О. Б.* О схемах из функциональных элементов с задержками // Проблемы кибернетики. Вып. 23. М.: Наука, 1970. С. 43–81.
6. *Касим-Заде О. М.* Общая верхняя оценка сложности схем в произвольном бесконечном полном базисе // Вестник Моск. ун-та. Сер. 1. Математика. Механика. 1997. № 4. С. 59–61.
7. *Касим-Заде О. М.* О глубине булевых функций при реализации схемами над произвольным базисом // Вестник Моск. ун-та. Сер. 1. Математика. Механика. 2007. № 1. С. 18–21.
8. *Кочергин А. В.* О глубине функций k-значной логики в конечных базисах // Вестник Моск. ун-та. Сер. 1. Математика. Механика. 2013. № 1. С. 56–59.
9. *Кочергин А. В.* О глубине функций k-значной логики в бесконечных базисах // Вестник Моск. ун-та. Сер. 1. Математика. Механика. 2011. № 1. С. 22–26.
10. *Угольников А. Б.* Синтез схем и формул в неполных базисах // Препринт ИПМ АН СССР. М., 1980. № 112.
11. *Ткачёв Г. А.* О сложности реализации одной последовательности функций k-значной логики // Вестник Моск. ун-та. Сер. 15. Вычислительная математика и кибернетика. 1977. № 1. С. 45–47.
12. *Угольников А. Б.* О сложности реализации формулами одной последовательности функций 4-значной логики // Вестник Моск. ун-та. Сер. 1. Математика. Механика. 2004. № 3. С. 52–55.
13. *Андреев А. А.* О нижних оценках сложности для некоторых последовательностей функций многозначной логики // Вестник Моск. ун-та. Сер. 1. Математика. Механика. 2013. № 6. С. 25–30.
14. *Янов Ю. И., Мучник А. А.* О существовании k-значных замкнутых классов, не имеющих конечного базиса // ДАН СССР. 1959. Т. 127. № 1. С. 44–46.
15. *Андреев А. А.* Об одной последовательности функций многозначной логики // Вестник Моск. ун-та. Сер. 1. Математика. Механика. 2011. № 6. С. 52–57.

REFERENCES

1. *Yablonskiy S. V.* Vvedenie v diskretnuyu matematiku [Introduction to Discrete Mathematics]. Moscow, Nauka Publ., 1986. 384 p. (in Russian)
2. *Lupanov O. B.* Asimptoticheskie otsenki slozhnosti upravlyayushchikh sistem [Asymptotic Bounds for the Complexity of Control Systems]. Moscow, MSU Publ., 1984. 136 p. (in Russian)
3. *Lupanov O. B.* Ob odnom metode sinteza skhem [A method of circuits synthesis]. Izvestiya vuzov. Radiofizika, 1958, vol. 1, no. 1, pp. 120–140. (in Russian)
4. *Lupanov O. B.* O slozhnosti realizatsii funktsiy algebry logiki formulami [On the complexity of Boolean functions realization by formulas.] Problemy kibernetiki, iss. 3. Moscow, Fizmatgiz Publ., 1960, pp. 61–80. (in Russian)
5. *Lupanov O. B.* O skhemakh iz funktsional'nykh elementov s zaderzhkami [On circuits of functional elements with delays]. Problemy kibernetiki, iss. 23. Moscow, Nauka Publ., 1970, pp. 43–81. (in Russian)

6. *Kasim-Zade O. M.* Obshchaya verkhnyaya otsenka slozhnosti skhem v proizvol'nom beskonechnom polnom bazise [Total upper bound for the circuits complexity in any infinite complete basis]. Vestnik Mosk. un-ta. Ser. 1. Matematika. Mekhanika, 1997, no. 4, pp. 59–61. (in Russian)
7. *Kasim-Zade O. M.* O glubine bulevykh funktsiy pri realizatsii skhemami nad proizvol'nym bazisom [The depth of Boolean functions realized by circuits over an arbitrary basis]. Vestnik Mosk. un-ta. Ser. 1. Matematika. Mekhanika, 2007, no. 1, pp. 18–21. (in Russian)
8. *Kochergin A. V.* O glubine funktsiy k-znachnoy logiki v konechnykh bazisakh [Depth of k-valued logic functions in finite bases]. Vestnik Mosk. un-ta. Ser. 1. Matematika. Mekhanika, 2013, no. 1, pp. 56–59. (in Russian)
9. *Kochergin A. V.* O glubine funktsiy k-znachnoy logiki v beskonechnykh bazisakh [Depth of k-valued logic functions in infinite bases]. Vestnik Mosk. un-ta. Ser. 1. Matematika. Mekhanika, 2011, no. 1, pp. 22–26. (in Russian)
10. *Ugol'nikov A. B.* Sintez skhem i formul v nepolnykh bazisakh [The circuits and formulas synthesis in incomplete bases]. Preprint Keldysh Institute of Applied Mathematics, Moscow, 1980, no. 112. (in Russian)
11. *Tkachev G. A.* O slozhnosti realizatsii odnoy posledovatel'nosti funktsiy k-znachnoy logiki [On the complexity of a sequence of k-ary functions realization]. Vestnik Mosk. un-ta. Ser. 15. Vychislitel'naya matematika i kibernetika, 1977, no. 1, pp. 45–47. (in Russian)
12. *Ugol'nikov A. B.* O slozhnosti realizatsii formulami odnoy posledovatel'nosti funktsiy 4-znachnoy logiki [On the complexity of one sequence of 4-ary functions realization by formulas]. Vestnik Mosk. un-ta. Ser. 1. Matematika. Mekhanika, 2004, no. 3, pp. 52–55. (in Russian)
13. *Andreev A. A.* O nizhnikh otsenkakh slozhnosti dlya nekotorykh posledovatel'nostey funktsiy mnogoznachnoy logiki [Lower complexity estimates for some sequences of functions of multivalued logic]. Vestnik Mosk. un-ta. Ser. 1. Matematika. Mekhanika, 2013, no. 6, pp. 25–30. (in Russian)
14. *Yanov Yu. I., Muchnik A. A.* O sushchestvovanii k-znachnykh zamknutykh klassov, ne imeyushchikh konechnogo bazisa [On the existence of k-valued closed classes without a finite basis]. Reports of the USSR Academy of Sciences, 1959, vol. 127, no. 1, pp. 44–46. (in Russian)
15. *Andreev A. A.* Ob odnoy posledovatel'nosti funktsiy mnogoznachnoy logiki [A sequence of functions of the multi-valued logic]. Vestnik Mosk. un-ta. Ser. 1. Matematika. Mekhanika, 2011, no. 6, pp. 52–57. (in Russian)

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

2015 Теоретические основы прикладной дискретной математики № 3(29)

УДК 621.391.1:004.7

ЧАСТОТНЫЕ ХАРАКТЕРИСТИКИ ЦИКЛОВ ВЫХОДНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ КОМБИНИРУЮЩИХ ГЕНЕРАТОРОВ НАД ПОЛЕМ ИЗ ДВУХ ЭЛЕМЕНТОВ

И. Б. Биляк, О. В. Камловский

г. Москва, Россия

Приводятся формулы для подсчёта числа элементов на циклах выходных последовательностей комбинирующих генераторов над полем из двух элементов. Из этих формул выводятся некоторые оценки рассматриваемых частот. Получены формулы для вычисления автокорреляционных функций выходных последовательностей и расстояний Хемминга между отрезками последовательностей.

Ключевые слова: *комбинирующий генератор, линейные рекуррентные последовательности, статистические свойства рекуррент, псевдослучайные последовательности.*

DOI 10.17223/20710410/29/2

FREQUENCY CHARACTERISTICS OF CYCLES IN OUTPUT SEQUENCES GENERATED BY COMBINING GENERATORS OVER THE FIELD OF TWO ELEMENTS

I. B. Bilyak, O. V. Kamlovskii

Moscow, Russia

E-mail: bil-ib@mail.ru, ov-kam@yandex.ru

Some formulas are given for counting the number of elements in the cycles of output sequences generated by combining generators over the field of two elements. From these formulas, some estimates of the considered frequencies appear. Also, formulas for calculation of the autocorrelation functions and Hamming distances between the line segments of these sequences are obtained.

Keywords: *combining generator, linear recurrent sequences, distribution properties of recurrent, a pseudo-random sequence.*

Введение

Пусть $P = \mathrm{GF}(2)$, $F_1(x), \ldots, F_k(x)$ — многочлены над полем P, имеющие степени $m_1, \ldots, m_k$ соответственно. Для каждой булевой функции $\varphi(x_1, \ldots, x_k)$ от k переменных комбинирующий генератор (см. [1, с. 272; 2, с. 311; 3, с. 92]) вырабатывает выходную последовательность v, элементы которой имеют вид

$$v(i) = \varphi(u_1(i), u_2(i), \ldots, u_k(i)), \quad i \geqslant 0, \tag{1}$$

где u_j — ненулевая линейная рекуррентная последовательность (ЛРП) над полем P с характеристическим многочленом $F_j(x)$ для всех $j \in \{1, 2, \ldots, k\}$.

Наибольший интерес к исследованию комбинирующих генераторов отмечался с 1966 по 1986 г. Подробный обзор зарубежных публикаций за этот период представлен в работе [3, с. 92–93]. Там же приводятся все основные результаты о рангах (линейной сложности) последовательностей v, построенных по правилу (1) и основы корреляционной атаки на начальные отрезки ЛРП, приводящей к необходимости выбора корреляционно-иммунных функций усложнения φ [3, с. 93–141].

Вопрос о частотных характеристиках последовательностей v, построенных по правилу (1), оказался менее изученным. Авторам не известны результаты о распределении элементов в последовательностях v. Исключение составляет работа [2], где В. М. Фомичев для случая $P = \mathrm{GF}(2)$ получил точные формулы для вычисления частот появлений элементов на циклах последовательностей v и привёл оценки рассматриваемых частот.

Данная работа продолжает эти исследования. Предлагается другая формула для вычисления частот появлений элементов на циклах, основанная на знании коэффициентов Уолша — Адамара булевой функции φ. С использованием этой формулы приводятся оценки рассматриваемых частот в ситуации, когда φ — корреляционно-иммунная булева функция. Кроме того, приводится формула для вычисления автокорреляционной функции последовательности v, что позволяет находить расстояния Хемминга между отрезками выходных последовательностей комбинирующих генераторов, полученных на различных ключах.

1. Периоды и ранги выходных последовательностей

Последовательность v, определённая равенством (1), является периодической. Её минимально возможный период $T(v)$ делит наименьшее общее кратное чисел $T(u_1), T(u_2), \ldots, T(u_k)$. В частности, v является ЛРП над полем P [4, утверждение 10, с. 319]. Приведём результат [1, теорема 4, с. 273; 3, corollary 5.15], позволяющий достаточно просто находить ранг последовательности v, который определяется как наименьшая из всех возможных степеней характеристических многочленов последовательности v.

Теорема 1. Пусть $P = \mathrm{GF}(2)$, $F_1(x), \ldots, F_k(x)$ — неприводимые многочлены над полем P попарно взаимно простых степеней $m_1, \ldots, m_k$ соответственно. Тогда если многочлен Жегалкина функции φ имеет вид

$$\varphi(x_1, \ldots, x_k) = \sum_{s=1}^{k} \sum_{1 \leqslant i_1 < i_2 < \ldots < i_s \leqslant k} c_{i_1 i_2 \ldots i_s} x_{i_1} x_{i_2} \ldots x_{i_s},$$

то

$$\operatorname{rank} v = \sum_{s=1}^{k} \sum_{1 \leqslant i_1 < i_2 < \ldots < i_s \leqslant k} c_{i_1 i_2 \ldots i_s} m_{i_1} m_{i_2} \ldots m_{i_s}.$$

В процессе доказательства этой теоремы получено следующее утверждение.

Утверждение 1 [3, lemma 5.12]. В условиях теоремы 1 для всех чисел i_1, i_2, $\ldots$, i_s, таких, что $1 \leqslant i_1 < i_2 < \ldots < i_s \leqslant k$, последовательность $u_{i_1} u_{i_2} \ldots u_{i_s}$ с элементами

$$u_{i_1} u_{i_2} \ldots u_{i_s}(i) = u_{i_1}(i) u_{i_2}(i) \ldots u_{i_s}(i), \quad i \geqslant 0,$$

является ЛРП над полем P с неприводимым характеристическим многочленом степени $m_{i_1} m_{i_2} \cdots m_{i_s}$.

Утверждение 1 позволяет найти период $T(v)$ последовательности v.

Теорема 2. Пусть в условиях теоремы 1 функция φ существенно зависит от всех своих переменных. Тогда $T(v) = T(u_1)T(u_2)\cdots T(u_k)$.

Доказательство. Последовательность v задается следующим образом:

$$v = \sum_{s=1}^{k} \sum_{1\leqslant i_1<i_2<\ldots<i_s\leqslant k} c_{i_1 i_2\ldots i_s} u_{i_1} u_{i_2}\ldots u_{i_s}.$$

Согласно утверждению 1, все ненулевые ЛРП, входящие в рассматриваемую сумму, имеют неприводимые характеристические многочлены различных степеней. В работе [4, утверждение 11, с. 321] показано, что для ЛРП ω_1, ω_2, …, ω_l, у которых указаны взаимно простые характеристические многочлены, выполнено равенство

$$T(\omega_1+\omega_2+\ldots+\omega_l) = [T(\omega_1), T(\omega_2), \ldots, T(\omega_l)].$$

Поэтому $T(v)$ равен наименьшему общему кратному чисел $T(u_{i_1}u_{i_2}\ldots u_{i_s})$, рассматриваемых для тех наборов i_1, i_2, …, i_s, при которых $c_{i_1 i_2\ldots i_s} = 1$.

В работе [5, теорема 8.70] показано, что для ненулевых ЛРП ω_1, ω_2, …, ω_l, имеющих попарно взаимно простые периоды, выполнено равенство

$$T(\omega_1\omega_2\cdots\omega_l) = T(\omega_1)T(\omega_2)\cdots T(\omega_l).$$

Числа $T(u_1), \ldots, T(u_k)$ являются делителями чисел $2^{m_1}-1, \ldots, 2^{m_k}-1$ соответственно. В силу попарной взаимной простоты чисел $m_1, \ldots, m_k$ для всех $i \neq j$, $i, j \in \{1, 2, \ldots, k\}$, получим равенство $(2^{m_i}-1, 2^{m_j}-1) = 1$, а значит, $(T(u_i), T(u_j)) = 1$. Таким образом,

$$T(u_{i_1}u_{i_2}\ldots u_{i_s}) = T(u_{i_1})T(u_{i_2})\cdots T(u_{i_s})$$

и $T(v)$ равен наименьшему общему кратному чисел $T(u_{i_1}), T(u_{i_2}), \ldots, T(u_{i_s})$, рассматриваемых для тех наборов $i_1, i_2, \ldots, i_s$, при которых $c_{i_1 i_2\ldots i_s} = 1$. Учитывая, что булева функция φ существенно зависит от всех своих переменных, получим $T(v) = T(u_1)T(u_2)\cdots T(u_k)$. ■

Отметим, что ранее теорема 2 была доказана в частном случае, когда все многочлены $F_1(x)$, …, $F_k(x)$ являются примитивными многочленами, т. е. многочленами максимально возможных периодов $2^{m_1}-1$, …, $2^{m_k}-1$ соответственно [2, теорема 18.2].

В условиях теоремы 1 последовательность v является чисто периодической, так как все последовательности $u_1, \ldots, u_k$ являются чисто периодическими.

Некоторые нижние оценки периода последовательности v (не обязательно двоичной), полученной в результате усложнения последовательностей u_1, …, u_k с использованием функции φ, приводятся в работе [6, теорема 1].

2. Формулы В. М. Фомичева

Изучим частотные характеристики последовательности v, определённой равенством (1). Исследуем величину $N(z, v)$, равную количеству появлений элемента $z \in P$ среди элементов $v(0), v(1), \ldots, v(T(v)-1)$. Приведём вспомогательный результат из работы [2], представляющий и самостоятельный интерес. Этот результат формулируется как очевидный (см. [2, доказательство теоремы 18.2]). Для полноты изложения приведём его доказательство.

Утверждение 2. Пусть $P = \mathrm{GF}(2)$, $F_1(x), \ldots, F_k(x) \in P[x]$ — примитивные многочлены попарно взаимно простых степеней $m_1, \ldots, m_k$ соответственно, $z_j \in P$, u_j —

ЛРП максимального периода с характеристическим многочленом $F_j(x)$, $j \in \{1, 2, \ldots, k\}$, $T = (2^{m_1} - 1) \cdots (2^{m_k} - 1)$. Тогда для числа $N(z_1, \ldots, z_k, u_1, \ldots, u_k)$ решений системы уравнений

$$\begin{cases} u_1(i) = z_1, \\ u_2(i) = z_2, \\ \quad \ldots \\ u_k(i) = z_k \end{cases}$$

относительно неизвестного $i \in \{0, 1, \ldots, T-1\}$ справедливо равенство

$$N(z_1, \ldots, z_k, u_1, \ldots, u_k) = (2^{m_1 - 1} - 1 + z_1)(2^{m_2 - 1} - 1 + z_2) \cdots (2^{m_k - 1} - 1 + z_k).$$

Доказательство. Рассмотрим упорядоченные наборы

$$\alpha_i = (u_1(i), \ldots, u_1(i + m_1 - 1), u_2(i), \ldots, u_2(i + m_2 - 1), \ldots, u_k(i), \ldots, u_k(i + m_k - 1)),$$

где $i \in \{0, 1, \ldots, T-1\}$. Заметим, что равенство $\alpha_i = \alpha_j$ влечёт соотношения $T(u_1) \,|\, i-j$, $\ldots$, $T(u_k) \,|\, i-j$, а значит, $T(u_1) \cdots T(u_k) = T \,|\, i-j$. Отсюда следует, что все рассматриваемые векторы α_i попарно различны. Тогда

$$\{\alpha_i : i = 0, 1, \ldots, T-1\} = (P^{m_1} \backslash \{\mathbf{0}\}) \times \ldots \times (P^{m_k} \backslash \{\mathbf{0}\})$$

и число $N(z_1, \ldots, z_k, u_1, \ldots, u_k)$ равно количеству наборов из множества $(P^{m_1} \backslash \{\mathbf{0}\}) \times \ldots \times (P^{m_k} \backslash \{\mathbf{0}\})$, имеющих следующий вид:

$$(z_1, \underbrace{*, \ldots, *}_{m_1 - 1}, z_2, \underbrace{*, \ldots, *}_{m_2 - 1}, \ldots, z_k, \underbrace{*, \ldots, *}_{m_k - 1}),$$

где на местах, обозначенных символом $*$, стоят произвольные элементы поля P. Число таких наборов равно $(2^{m_1 - 1} - 1 + z_1)(2^{m_2 - 1} - 1 + z_2) \cdots (2^{m_k - 1} - 1 + z_k)$. ■

Заметим, что число решений не зависит от выбора ненулевых ЛРП $u_1, \ldots, u_k$.

Утверждение 2 описывает строение графа переходов автоматной модели комбинирующего генератора. При выборе произвольного начального состояния α_0 из множества $(P^{m_1} \backslash \{\mathbf{0}\}) \times \ldots \times (P^{m_k} \backslash \{\mathbf{0}\})$ в моменты времени $i = 0, 1, \ldots, T-1$ автомат проходит последовательность состояний $\alpha_0, \ldots, \alpha_{T-1}$ и все элементы множества $(P^{m_1} \backslash \{\mathbf{0}\}) \times \ldots \times (P^{m_k} \backslash \{\mathbf{0}\})$ расположены на одном цикле длины T.

Заметим, что, согласно теореме 2, в условиях утверждения 2 имеет место равенство $T(v) = T(u_1) \cdots T(u_k) = (2^{m_1} - 1) \cdots (2^{m_k} - 1) = T$ для каждой функции φ, существенно зависящей от всех своих переменных. Таким образом, справедлива

Теорема 3 [2]. Пусть в условиях утверждения 2 функция φ существенно зависит от всех своих переменных. Тогда

$$N(z, v) = \sum_{\mathbf{z} = (z_1, \ldots, z_k) \in P^k, \varphi(\mathbf{z}) = z} (2^{m_1 - 1} - 1 + z_1)(2^{m_2 - 1} - 1 + z_2) \cdots (2^{m_k - 1} - 1 + z_k).$$

Таким образом, в наиболее интересном с практической точки зрения случае теорема 3 позволяет найти точное значение частот $N(z, v)$.

3. Спектральный подход к исследованию частотных характеристик

Предложим альтернативный подход к исследованию частот $N(z,v)$. Пусть функция $\varphi(x_1,\ldots,x_k)$ имеет коэффициенты Уолша — Адамара [7, с. 76]

$$W_\varphi(\mathbf{a}) = \sum_{x_1,\ldots,x_k\in P} (-1)^{\varphi(x_1,\ldots,x_k)\oplus a_1x_1\oplus\ldots\oplus a_kx_k},$$

где $\mathbf{a}=(a_1,\ldots,a_k)$. Тогда

$$(-1)^{\varphi(x_1,\ldots,x_k)} = \frac{1}{2^k}\sum_{\mathbf{a}\in P^k} W_\varphi(\mathbf{a})(-1)^{a_1x_1\oplus\ldots\oplus a_kx_k},$$

и из равенства (1) получим

$$(-1)^{v(i)} = (-1)^{\varphi(u_1(i),\ldots,u_k(i))} = \frac{1}{2^k}\sum_{\mathbf{a}\in P^k} W_\varphi(\mathbf{a})(-1)^{a_1u_1(i)\oplus\ldots\oplus a_ku_k(i)}. \tag{2}$$

Следующее утверждение сводит исследование частот $N(z,v)$ к исследованию коэффициентов Уолша — Адамара функции φ и некоторой суммы, зависящей только от знаков ЛРП.

Утверждение 3. Для последовательности v, построенной по правилу (1), справедливы равенства

$$N(0,v) = T(v)\left(1-\frac{\|\varphi\|}{2^k}\right) + \frac{1}{2^{k+1}}\sum_{\mathbf{a}\in P^k\setminus\{\mathbf{0}\}} W_\varphi(\mathbf{a})\sigma(\mathbf{a},u_1,\ldots,u_k),$$

$$N(1,v) = \frac{T(v)\|\varphi\|}{2^k} - \frac{1}{2^{k+1}}\sum_{\mathbf{a}\in P^k\setminus\{\mathbf{0}\}} W_\varphi(\mathbf{a})\sigma(\mathbf{a},u_1,\ldots,u_k),$$

где $\|\varphi\|$ — вес функции φ, а величина $\sigma(\mathbf{a},u_1,\ldots,u_k)$ определена равенством

$$\sigma(\mathbf{a},u_1,\ldots,u_k) = \sum_{i=0}^{T(v)-1} (-1)^{a_1u_1(i)\oplus\ldots\oplus a_ku_k(i)}.$$

Доказательство. Заметим, что справедливы соотношения

$$N(z,v) = \frac{1}{2}\sum_{i=0}^{T(v)}\left(1+(-1)^{v(i)\oplus z}\right) = \frac{T(v)}{2} + \frac{1}{2}(-1)^z\sum_{i=0}^{T(v)-1}(-1)^{v(i)}.$$

Подставляя равенство (2), будем иметь

$$N(z,v) = \frac{T(v)}{2} + \frac{1}{2}(-1)^z\sum_{i=0}^{T(v)-1}\frac{1}{2^k}\sum_{\mathbf{a}\in P^k} W_\varphi(\mathbf{a})(-1)^{a_1u_1(i)\oplus\ldots\oplus a_ku_k(i)} =$$

$$= \frac{T(v)}{2} + \frac{1}{2^{k+1}}(-1)^z\sum_{\mathbf{a}\in P^k} W_\varphi(\mathbf{a})\sum_{i=0}^{T(v)-1}(-1)^{a_1u_1(i)\oplus\ldots\oplus a_ku_k(i)} =$$

$$= \frac{T(v)}{2} + \frac{1}{2^{k+1}}(-1)^z W_\varphi(\mathbf{0})T(v) + \frac{1}{2^{k+1}}(-1)^z\sum_{\mathbf{a}\in P^k\setminus\{\mathbf{0}\}} W_\varphi(\mathbf{a})\sum_{i=0}^{T(v)-1}(-1)^{a_1u_1(i)\oplus\ldots\oplus a_ku_k(i)}.$$

Для завершения доказательства остаётся воспользоваться равенством $W_\varphi(\mathbf{0}) = 2^k - 2\|\varphi\|$. ■

Значения величины $\sigma(\mathbf{a},u_1,\ldots,u_k)$ в некоторых случаях удаётся точно подсчитать.

Утверждение 4. Пусть $P = \mathrm{GF}(2)$, $\mathbf{a} = (a_1, \ldots, a_k) \in P^k$, u_j — ЛРП максимального периода $2^{m_j} - 1$ с примитивным характеристическим многочленом $F_j(x)$, где $j \in \{1, 2, \ldots, k\}$; $T = (2^{m_1} - 1) \cdots (2^{m_k} - 1)$. Тогда если числа $m_1, \ldots, m_k$ попарно взаимно просты, то

$$\sigma(\mathbf{a}, u_1, \ldots, u_k) = \frac{(-1)^{\|\mathbf{a}\|} T}{(2^{m_1} - 1)^{a_1} (2^{m_2} - 1)^{a_2} \ldots (2^{m_k} - 1)^{a_k}},$$

где $\|\mathbf{a}\|$ — вес вектора $\mathbf{a}$.

Доказательство. Если вектор $\mathbf{a}$ нулевой, то равенство выполнено. Пусть $\|\mathbf{a}\| = s$, $s \in \mathbb{N}$, и $a_{j_1}, \ldots, a_{j_s}$ — все ненулевые координаты вектора $\mathbf{a}$, где $1 \leqslant j_1 < \ldots < j_s \leqslant k$. Тогда

$$\sigma(\mathbf{a}, u_1, \ldots, u_k) = \sum_{i=0}^{T-1} (-1)^{u_{j_1}(i) \oplus \ldots \oplus u_{j_s}(i)}.$$

Введём обозначение $v_t = u_{j_t}$, где $t \in \{1, 2, \ldots, s\}$. Тогда

$$\sigma(\mathbf{a}, u_1, \ldots, u_k) = \sum_{i=0}^{T-1} (-1)^{v_1(i)} \ldots (-1)^{v_s(i)}. \tag{3}$$

Представим каждую ЛРП v_t с использованием функции след:

$$v_t(i) = \mathrm{tr}_P^{Q_t}(a_t \alpha_t^i), \quad i \geqslant 0,$$

где $Q_t = \mathrm{GF}(2^{m_{j_t}})$; α_t — корень многочлена $F_{j_t}(x)$ в поле Q_t; a_t — ненулевой элемент поля Q_t. С использованием [5, равенство (5.17)] получим

$$(-1)^{v_t(i)} = (-1)^{\mathrm{tr}_P^{Q_t}(a_t \alpha_t^i)} = \chi_t(a_t \alpha_t^i) = \frac{1}{2^{m_{j_t}} - 1} \sum_{\psi_t} G(\bar{\psi}_t, \chi_t) \psi_t(a_t \alpha_t^i),$$

где χ_t — аддитивный характер поля Q_t, определённый равенством $\chi_t(x) = (-1)^{\mathrm{tr}_P^{Q_t}(x)}$, $x \in Q_t$, суммирование осуществляется по всем мультипликативным характерам ψ_t поля Q_t; $G(\bar{\psi}_t, \chi_t)$ — сумма Гаусса. Тогда из равенства (3) будем иметь

$$\sigma(\mathbf{a}, u_1, \ldots, u_k) = \frac{1}{(2^{m_{j_1}} - 1) \cdots (2^{m_{j_s}} - 1)} \sum_{\psi_1, \ldots, \psi_s} \prod_{t=1}^{s} G(\bar{\psi}_t, \chi_t) \psi_t(a_t) \sum_{i=0}^{T-1} (\psi_1(\alpha_1) \cdots \psi_s(\alpha_s))^i.$$

Заметим, что

$$\sum_{i=0}^{T-1} (\psi_1(\alpha_1) \cdots \psi_s(\alpha_s))^i = \begin{cases} \dfrac{(\psi_1(\alpha_1) \cdots \psi_s(\alpha_s))^T - 1}{\psi_1(\alpha_1) \ldots \psi_s(\alpha_s) - 1}, & \text{если } \psi_1(\alpha_1) \cdots \psi_s(\alpha_s) \neq 1, \\ T, & \text{если } \psi_1(\alpha_1) \cdots \psi_s(\alpha_s) = 1, \end{cases}$$

а так как $(\psi_1(\alpha_1) \cdots \psi_s(\alpha_s))^T = \psi_1(\alpha_1^T) \cdots \psi_s(\alpha_s^T) = \psi_1(e) \cdots \psi_s(e) = 1$, то

$$\sum_{i=0}^{T-1} (\psi_1(\alpha_1) \ldots \psi_s(\alpha_s))^i = \begin{cases} 0, & \text{если } \psi_1(\alpha_1) \cdots \psi_s(\alpha_s) \neq 1, \\ T, & \text{если } \psi_1(\alpha_1) \cdots \psi_s(\alpha_s) = 1. \end{cases}$$

В силу того, что $\psi_t(\alpha_t)$ является корнем степени $2^{m_{j_t}} - 1$ из единицы для всех $t \in \{1, 2, \ldots, s\}$, а числа $2^{m_{j_1}} - 1, \ldots, 2^{m_{j_s}} - 1$ попарно взаимно просты, соотношение $\psi_1(\alpha_1) \cdots \psi_s(\alpha_s) = 1$ выполнено только в случае, когда $\psi_1(\alpha_1) = \ldots = \psi_s(\alpha_s) = 1$.

Другими словами, каждый из характеров $\psi_1, \dots, \psi_s$ должен быть тривиальным. Значит,

$$\sigma(\mathbf{a}, u_1, \dots, u_k) = \frac{T}{(2^{m_{j_1}} - 1) \cdots (2^{m_{j_s}} - 1)} G(\psi_0^{(1)}, \chi_1) \cdots G(\psi_0^{(s)}, \chi_s),$$

где $\psi_0^{(1)}, \dots, \psi_0^{(s)}$ — тривиальные мультипликативные характеры полей $Q_1, \dots, Q_s$ соответственно. Согласно [5, равенство (5.14)], для всех $j \in \{1, 2, \dots, k\}$ выполнено $G(\psi_0^{(j)}, \chi_1) = -1$. Таким образом, получим

$$\sigma(\mathbf{a}, u_1, \dots, u_k) = \frac{(-1)^s T}{(2^{m_{j_1}} - 1)(2^{m_{j_2}} - 1) \dots (2^{m_{j_s}} - 1)} = \frac{(-1)^{\|\mathbf{a}\|} T}{(2^{m_1} - 1)^{a_1} (2^{m_2} - 1)^{a_2} \dots (2^{m_k} - 1)^{a_k}}.$$

Утверждение доказано. ■

Непосредственно из утверждений 3 и 4 получим основной результат этого пункта.

Теорема 4. Пусть $P = \mathrm{GF}(2)$, $u_1, \dots, u_k$ — ЛРП максимального периода над полем P, имеющие периоды $2^{m_1} - 1, \dots, 2^{m_k} - 1$ соответственно, причём числа $m_1, \dots, m_k$ попарно взаимно просты, $T = (2^{m_1} - 1) \cdots (2^{m_k} - 1)$. Тогда для последовательности v, определённой равенством (1), где функция φ существенно зависит от всех своих переменных, справедливы следующие соотношения:

$$N(0, v) = T - \frac{T\|\varphi\|}{2^k} + \frac{T}{2^{k+1}} \sum_{\mathbf{a} \in P^k \setminus \{\mathbf{0}\}} \frac{(-1)^{\|\mathbf{a}\|} W_\varphi(\mathbf{a})}{(2^{m_1} - 1)^{a_1} (2^{m_2} - 1)^{a_2} \cdots (2^{m_k} - 1)^{a_k}},$$

$$N(1, v) = \frac{T\|\varphi\|}{2^k} - \frac{T}{2^{k+1}} \sum_{\mathbf{a} \in P^k \setminus \{\mathbf{0}\}} \frac{(-1)^{\|\mathbf{a}\|} W_\varphi(\mathbf{a})}{(2^{m_1} - 1)^{a_1} (2^{m_2} - 1)^{a_2} \cdots (2^{m_k} - 1)^{a_k}}.$$

Отметим, что по столбцу значений булевой функции $\varphi(x_1, \dots, x_k)$ достаточно просто находится набор $W_\varphi(\mathbf{a})$, $\mathbf{a} \in P^k \setminus \{\mathbf{0}\}$, её коэффициентов Уолша — Адамара, а также набор $\tilde{W}_\varphi(\mathbf{a}) = -W_\varphi(\mathbf{a})/2$, $\mathbf{a} \in P^k \setminus \{\mathbf{0}\}$, её коэффициентов Фурье. Таким образом, теорема 4 позволяет в наиболее интересном случае вычислить частоты $N(z, v)$.

4. Примеры

Рассмотрим некоторые примеры использования теорем 3 и 4. Во всех примерах функция φ существенно зависит от всех своих переменных, поэтому, согласно теореме 2, период $T(v)$ последовательности v, построенной по правилу (1), равен $T = (2^{m_1} - 1) \cdots (2^{m_k} - 1)$.

Утверждение 5. Пусть $\varphi(x_1, \dots, x_k) = x_1 \oplus \dots \oplus x_k \oplus \varepsilon$, где $\varepsilon \in \{0, 1\}$. Тогда

$$N(0, v) = \frac{T + (-1)^{k+\varepsilon}}{2}, \quad N(1, v) = \frac{T - (-1)^{k+\varepsilon}}{2}, \quad \operatorname{rank} v = m_1 + \dots + m_k.$$

Доказательство. Для булевой функции $\varphi(x_1, \dots, x_k) = x_1 \oplus \dots \oplus x_k \oplus \varepsilon$ имеем

$$W_\varphi(1, \dots, 1) = (-1)^\varepsilon 2^k, \quad W_\varphi(\mathbf{a}) = 0 \quad \text{для всех} \quad \mathbf{a} \neq (1, \dots, 1).$$

Тогда с использованием теоремы 4 получим

$$N(0, v) = T\left(1 - \frac{\|\varphi\|}{2^k}\right) + \frac{(-1)^{k+\varepsilon}}{2} = \frac{T + (-1)^{k+\varepsilon}}{2},$$

$$N(1, v) = \frac{T\|\varphi\|}{2^k} - \frac{(-1)^{k+\varepsilon}}{2} = \frac{T - (-1)^{k+\varepsilon}}{2}.$$

Равенство для $\operatorname{rank} v$ непосредственно следует из теоремы 1. ■

Отметим, что формулы для частот из утверждения 5 были ранее получены в работе [8, с. 133].

Утверждение 6. Пусть $\varphi(x_1, \ldots, x_k) = x_1^{\alpha_1} \cdots x_k^{\alpha_k}$, где $\alpha_1, \ldots, \alpha_k \in \{0, 1\}$ (считаем, что $x_i^0 = \bar{x}_i$, $x_i^1 = x_i$, $i = 1, \ldots, k$). Тогда

$$
\begin{gathered}
N(0, v) = T - (2^{m_1 - 1} - 1 + \alpha_1) \cdots (2^{m_k - 1} - 1 + \alpha_k), \\
N(1, v) = (2^{m_1 - 1} - 1 + \alpha_1) \cdots (2^{m_k - 1} - 1 + \alpha_k), \\
\operatorname{rank} v = (m_1 + \bar{\alpha}_1) \cdots (m_k + \bar{\alpha}_k).
\end{gathered}
$$

Доказательство. С использованием утверждения 2 получим

$$
\begin{gathered}
N(1, v) = N(\alpha_1, \ldots, \alpha_k, u_1, \ldots, u_k) = (2^{m_1 - 1} - 1 + \alpha_1) \cdots (2^{m_k - 1} - 1 + \alpha_k), \\
N(0, v) = T - N(1, v) = T - (2^{m_1 - 1} - 1 + \alpha_1) \cdots (2^{m_k - 1} - 1 + \alpha_k).
\end{gathered}
$$

Равенство для $\operatorname{rank} v$ непосредственно следует из теоремы 1 и равносильности формул $x_i^{\alpha_i} \equiv x_i \oplus \bar{\alpha}_i$, $i = 1, 2, \ldots, k$. ■

Утверждение 7. Пусть $\varphi(x_1, \ldots, x_k) = x_1 \oplus x_2 x_3 \cdots x_k$. Тогда

$$
\begin{gathered}
N(0, v) = (2^{m_1 - 1} - 1)(2^{m_2} - 1) \cdots (2^{m_k} - 1) + 2^{m_2 + \ldots + m_k - k + 1}, \\
N(1, v) = 2^{m_1 - 1}(2^{m_2} - 1) \cdots (2^{m_k} - 1) - 2^{m_2 + \ldots + m_k - k + 1}, \\
\operatorname{rank} v = m_1 + m_2 m_3 \cdots m_k.
\end{gathered}
$$

Доказательство. Обозначим через v' последовательность $u_2 \cdots u_k$. С использованием теоремы 3 и утверждения 6 получим

$$
N(1, v) = \sum_{\substack{\mathbf{z} = (z_1, \ldots, z_k) \in P^k, \\ \varphi(\mathbf{z}) = 1}} (2^{m_1 - 1} - 1 + z_1)(2^{m_2 - 1} - 1 + z_2) \cdots (2^{m_k - 1} - 1 + z_k) =
$$

$$
= 2^{m_1 - 1} \sum_{(z_2, \ldots, z_k) \neq (1, \ldots, 1)} (2^{m_2 - 1} - 1 + z_2) \cdots (2^{m_k - 1} - 1 + z_k) + (2^{m_1 - 1} - 1) 2^{m_2 + \cdots + m_k - k + 1} =
$$

$$
\begin{gathered}
= 2^{m_1 - 1} N(0, v') + (2^{m_1 - 1} - 1) 2^{m_2 + \cdots + m_k - k + 1} = \\
= 2^{m_1 - 1}((2^{m_2} - 1) \cdots (2^{m_k} - 1) - 2^{m_2 + \cdots + m_k - k + 1}) + (2^{m_1 - 1} - 1) 2^{m_2 + \cdots + m_k - k + 1} = \\
= 2^{m_1 - 1}(2^{m_2} - 1) \cdots (2^{m_k} - 1) - 2^{m_2 + \ldots + m_k - k + 1}, \\
N(0, v) = T - N(1, v) = (2^{m_1 - 1} - 1)(2^{m_2} - 1) \cdots (2^{m_k} - 1) + 2^{m_2 + \ldots + m_k - k + 1}.
\end{gathered}
$$

Равенство для $\operatorname{rank} v$ непосредственно следует из теоремы 1. ■

Отметим, что в утверждении 7 для подсчёта частот можно воспользоваться теоремой 4, если учесть, что справедливы равенства

$$
W_\varphi(\mathbf{a}) = \begin{cases} 0, & \text{если } \mathbf{a} = (0, a_2, \ldots, a_k), (a_2, \ldots, a_k) \in P^{k-1}, \\ 2^k - 4, & \text{если } \mathbf{a} = (1, 0, \ldots, 0), \\ 4(-1)^{\|\mathbf{a}\|}, & \text{если } \mathbf{a} = (1, a_2, \ldots, a_k), (a_2, \ldots, a_k) \neq \mathbf{0}. \end{cases}
$$

Утверждение 8. Пусть $\varphi(x_1, \ldots, x_k) = x_1 \oplus \cdots \oplus x_{k-2} \oplus x_{k-1} x_k$. Тогда для каждого $z \in \{0, 1\}$

$$
\begin{gathered}
N(z, v) = \frac{T}{2} + \frac{(-1)^{k+z}}{2}(2^{m_{k-1} + m_k - 1} - 2^{m_{k-1}} - 2^{m_k} + 1), \\
\operatorname{rank} v = m_1 + \cdots + m_{k-2} + m_{k-1} m_k.
\end{gathered}
$$

Доказательство. Заметим, что справедливы равенства

$$W_\varphi(\mathbf{a}) = \begin{cases} 0, & \text{если } (a_1, \dots, a_{k-2}) \neq (1, \dots, 1), \\ 2^{k-1}, & \text{если } (a_1, \dots, a_{k-2}) = (1, \dots, 1), (a_{k-1}, a_k) \neq (1, 1), \\ -2^{k-1}, & \text{если } \mathbf{a} = (1, 1, \dots, 1). \end{cases}$$

Тогда, согласно теореме 4,

$$N(z, v) = \frac{T}{2} + \frac{(-1)^{k+z}}{4}((2^{m_{k-1}} - 1)(2^{m_k} - 1) - (2^{m_k} - 1) - (2^{m_{k-1}} - 1) - 1) =$$
$$= \frac{T}{2} + \frac{(-1)^{k+z}}{2}(2^{m_{k-1}+m_k-1} - 2^{m_{k-1}} - 2^{m_k} + 1).$$

Равенство для $\operatorname{rank} v$ непосредственно следует из теоремы 1. ■

5. Оценки частот

Использование формул из теорем 3 и 4 не всегда удобно ввиду большого числа слагаемых в них. Представляет интерес получение оценок частот $N(z, v)$. Приведём оценку, полученную по аналогии с доказательством из работы [2, теорема 18.2].

Теорема 5. Пусть $P = \mathrm{GF}(2)$, $z \in P$, $u_1, \dots, u_k$ — ЛРП максимального периода над полем P, имеющие периоды $2^{m_1} - 1, \dots, 2^{m_k} - 1$ соответственно, причём числа $m_1, \dots, m_k$ попарно взаимно просты и $m_1 < m_2 < \ldots < m_k$. Тогда для последовательности v, определённой равенством (1) с функцией φ, существенно зависящей от всех своих переменных, справедливы неравенства

$$2^{m_1+\ldots+m_k-k}(1 - 2^{2-m_1} + 2^{2-m_1-k})|\varphi^{-1}(z)| \leqslant N(z, v) \leqslant 2^{m_1+\ldots+m_k-k}|\varphi^{-1}(z)|,$$

где $\varphi^{-1}(z) = \{\mathbf{a} \in P^k : \varphi(\mathbf{a}) = z\}$.

Доказательство. Верхняя оценка вытекает из теоремы 3 и неравенства

$$(2^{m_1-1} - 1 + z_1)(2^{m_2-1} - 1 + z_2) \cdots (2^{m_k-1} - 1 + z_k) \leqslant 2^{m_1+\ldots+m_k-k},$$

справедливого для всех $z_1, z_2, \dots, z_k \in P$. Докажем справедливость нижней оценки. Методом математической индукции по параметру k нетрудно показать, что для всех неотрицательных действительных чисел $a_1, \dots, a_k$ верно неравенство $(1 - a_1) \cdots \times \times (1 - a_k) \geqslant 1 - a_1 - \ldots - a_k$. Тогда с использованием теоремы 3 получаем

$$N(z, v) = \sum_{\mathbf{z} \in \varphi^{-1}(z)} 2^{m_1+\ldots+m_k-k} \left(1 - \frac{1 - z_1}{2^{m_1-1}}\right) \ldots \left(1 - \frac{1 - z_k}{2^{m_k-1}}\right) \geqslant$$
$$\geqslant 2^{m_1+\ldots+m_k-k}|\varphi^{-1}(z)| \left(1 - \frac{1}{2^{m_1-1}}\right) \ldots \left(1 - \frac{1}{2^{m_k-1}}\right) \geqslant$$
$$\geqslant 2^{m_1+\ldots+m_k-k}|\varphi^{-1}(z)| \left(1 - \frac{1}{2^{m_1-1}} - \ldots - \frac{1}{2^{m_k-1}}\right) \geqslant$$
$$\geqslant 2^{m_1+\ldots+m_k-k}|\varphi^{-1}(z)| \left(1 - \frac{1}{2^{m_1-1}} - \frac{1}{2^{m_1}} - \ldots - \frac{1}{2^{m_1+k-2}}\right) =$$
$$= 2^{m_1+\ldots+m_k-k}|\varphi^{-1}(z)|(1 - 2^{2-m_1} + 2^{2-m_1-k}).$$

Теорема доказана. ■

Верхняя оценка из теоремы 5 доказана в [2, теорема 18.2]. В этой же работе приводится нижняя оценка, которая в условиях теоремы 5 имеет следующий вид:

$$N(1,v) > 2^{m_1+\ldots+m_k-k}(1-2^{1-m_1})|\varphi^{-1}(z)|.$$

Доказательство этой оценки ошибочно и сама оценка не верна. Приведём соответствующий пример. Пусть $k \geqslant 2$, $\varphi(x_1,x_2,\ldots,x_k) = \bar{x}_1\bar{x}_2\cdots\bar{x}_k$. Тогда, согласно утверждению 6,

$$N(1,v) = (2^{m_1-1}-1)(2^{m_2-1}-1)\cdots(2^{m_k-1}-1) < (2^{m_1-1}-1)2^{m_2+\ldots+m_k-k+1},$$

а рассматриваемая оценка имеет вид

$$N(1,v) > 2^{m_1+\ldots+m_k-k} - 2^{m_2+\ldots+m_k-k+1},$$

что не верно.

Получение общих оценок с использованием теоремы 4 представляется затруднительным. Применим эти результаты для важного класса функций усложнения φ, называемых корреляционно-иммунными [3, § 5.3; 7, гл. 7].

Теорема 6. Пусть, в условиях теоремы 5, $T = (2^{m_1}-1)\cdots(2^{m_k}-1)$ и φ является корреляционно-иммунной порядка d функцией. Тогда

$$|N(z,v)-\delta(z)| \leqslant (2^{m_{d+2}}-1)\cdots(2^{m_k}-1)\frac{|M(\varphi)\backslash\{\mathbf{0}\}|^{1/2}}{2},$$

где $M(\varphi) = \{\mathbf{a}\in P^k : W_\varphi(\mathbf{a}) \neq 0\}$, а величина $\delta(z)$ определена равенствами

$$\delta(z) = \begin{cases} T\dfrac{\|\varphi\|}{2^k}, & \text{если } z = 1, \\ T\left(1-\dfrac{\|\varphi\|}{2^k}\right), & \text{если } z = 0. \end{cases}$$

Доказательство. Из теоремы 4 имеем

$$|N(z,v)-\delta(z)| = \frac{T}{2^{k+1}}\left|\sum_{\mathbf{a}\in P^k\backslash\{\mathbf{0}\}} \frac{(-1)^{\|\mathbf{a}\|}W_\varphi(\mathbf{a})}{(2^{m_1}-1)^{a_1}\cdots(2^{m_k}-1)^{a_k}}\right|.$$

Отсюда получим неравенство

$$|N(z,v)-\delta(z)| \leqslant \frac{T}{2^{k+1}}\sum_{\mathbf{a}\in P^k\backslash\{\mathbf{0}\}} \frac{|W_\varphi(\mathbf{a})|}{(2^{m_1}-1)^{a_1}\cdots(2^{m_k}-1)^{a_k}}.$$

Согласно [7, теорема 7.10], $W_\varphi(\mathbf{a}) = 0$ для всех векторов $\mathbf{a}\in P^k$ веса $1 \leqslant \|\mathbf{a}\| \leqslant d$. Следовательно,

$$|N(z,v)-\delta(z)| \leqslant \frac{T}{2^{k+1}(2^{m_1}-1)\ldots(2^{m_{d+1}}-1)}\sum_{\mathbf{a}\in P^k\backslash\{\mathbf{0}\}}|W_\varphi(\mathbf{a})|. \tag{4}$$

Воспользуемся неравенством Коши [4, с. 131] и равенством Парсеваля [7, с. 80] для получения оценки сверху суммы модулей коэффициентов Уолша — Адамара. Тогда

$$\sum_{\mathbf{a}\in P^k\setminus\{\mathbf{0}\}} |W_\varphi(\mathbf{a})| = \sum_{\mathbf{a}\in M(\varphi)\setminus\{\mathbf{0}\}} |W_\varphi(\mathbf{a})| \leqslant |M(\varphi)\setminus\{\mathbf{0}\}|^{1/2} \left(\sum_{\mathbf{a}\in M(\varphi)\setminus\{\mathbf{0}\}} |W_\varphi(\mathbf{a})|^2\right)^{1/2} \leqslant$$

$$\leqslant |M(\varphi)\setminus\{\mathbf{0}\}|^{1/2} \left(\sum_{\mathbf{a}\in M(\varphi)} |W_\varphi(\mathbf{a})|^2\right)^{1/2} = |M(\varphi)\setminus\{\mathbf{0}\}|^{1/2} 2^k.$$

Подставив полученную оценку в неравенство (4), получим

$$|N(z,v) - \delta(z)| \leqslant (2^{m_{d+2}} - 1)\cdots(2^{m_k} - 1)\frac{|M(\varphi)\setminus\{\mathbf{0}\}|^{1/2}}{2}.$$

Теорема доказана. ■

Для сбалансированных корреляционно-иммунных порядка d функций (d-устойчивых функций) получаем следующий результат.

Следствие 1. Пусть, в условиях теоремы 5, функция φ является d-устойчивой. Тогда

$$\left|N(z,v) - \frac{T}{2}\right| \leqslant (2^{m_{d+2}} - 1)\cdots(2^{m_k} - 1)\frac{|M(\varphi)\setminus\{\mathbf{0}\}|^{1/2}}{2}.$$

Приведём оценку, не зависящую от величины $M(\varphi)$.

Следствие 2. Пусть, в условиях теоремы 5, φ является корреляционно-иммунной порядка d функцией. Тогда

$$|N(z,v) - \delta(z)| \leqslant (2^{m_{d+2}} - 1)\cdots(2^{m_k} - 1)\frac{(2^k - s(k,d))^{1/2}}{2},$$

где

$$s(k,d) = \binom{k}{0} + \binom{k}{1} + \binom{k}{2} + \ldots + \binom{k}{d}.$$

Доказательство. Используя равенство $W_\varphi(\mathbf{a}) = 0$, справедливое для всех векторов $\mathbf{a} \in P^k$ веса $1 \leqslant \|\mathbf{a}\| \leqslant d$ [7, теорема 7.10], получим

$$|M(\varphi)| \leqslant 2^k - \binom{k}{1} - \binom{k}{2} - \ldots - \binom{k}{d}.$$

Если функция φ не является сбалансированной, то $W_\varphi(\mathbf{0}) \neq 0$ и $\mathbf{0} \in M(\varphi)$. Если функция φ является сбалансированной, то $W_\varphi(\mathbf{0}) = 0$ и $\mathbf{0} \notin M(\varphi)$. Таким образом,

$$|M(\varphi)\setminus\{\mathbf{0}\}| \leqslant 2^k - \binom{k}{0} - \binom{k}{1} - \binom{k}{2} - \ldots - \binom{k}{d} = 2^k - s(k,d),$$

и остаётся воспользоваться оценкой из теоремы 6. ■

Оценки из теоремы 6, следствий 1 и 2 обращаются в равенства для каждой из $(k-1)$-устойчивых функций $\varphi(x_1,\ldots,x_k) = x_1 \oplus \cdots \oplus x_k$ и $\varphi(x_1,\ldots,x_k) = x_1 \oplus \cdots \oplus x_k \oplus 1$ (см. утверждение 5).

6. Автокорреляционная функция и расстояние между отрезками выходных последовательностей

Рассмотрим случай, когда $F_1(x), \ldots, F_k(x)$ — примитивные многочлены над полем $P = \mathrm{GF}(2)$ попарно взаимно простых степеней $m_1, \ldots, m_k$ соответственно, φ — булева функция, существенно зависящая от всех своих переменных, а начальное состояние

$$\alpha_0 = (u_1(0), u_1(1), \ldots, u_1(m_1 - 1), \ldots, u_k(0), u_k(1), \ldots, u_k(m_k - 1))$$

комбинирующего генератора выбирается из множества $(P^{m_1} \backslash \{\mathbf{0}\}) \times \ldots \times (P^{m_k} \backslash \{\mathbf{0}\})$, т. е. все ЛРП $u_1, \ldots, u_k$ являются ненулевыми. Пусть $T = (2^{m_1} - 1) \cdots (2^{m_k} - 1)$. Согласно теореме 2, каждая выходная последовательность v комбинирующего генератора, построенная по правилу (1), имеет период $T(v) = T$. Как показано в утверждении 2, состояния

$$\alpha_i = (u_1(i), u_1(i+1), \ldots, u_1(i + m_1 - 1), \ldots, u_k(i), u_k(i+1), \ldots, u_k(i + m_k - 1))$$

комбинирующего генератора в моменты времени $i = 0, 1, \ldots, T - 1$ пробегают все множество $(P^{m_1} \backslash \{\mathbf{0}\}) \times \ldots \times (P^{m_k} \backslash \{\mathbf{0}\})$ и лежат на одном цикле.

Для каждого $r \in \mathbb{N}_0$ обозначим через $x^r v$ сдвиг последовательности v на r шагов, т. е. последовательность вида $x^r v = (v(r), v(r+1), \ldots)$. Заметим, что множество всех последовательностей над полем P образует модуль над кольцом многочленов $P[x]$ [4, с. 297], и введённое обозначение согласуется с внешней операцией умножения последовательности на многочлен. Последовательность $x^t v$, $t = 0, 1, \ldots, T - 1$, вырабатывается на начальном состоянии α_t, и в рассматриваемом случае все выходные последовательности генератора являются сдвигами друг друга. В силу равенства $T(v) = T$ последовательности $v, xv, \ldots, x^{T-1}v$ попарно различны.

Нас интересует вопрос о расстоянии Хэмминга $\rho(v, x^t v)$, $t = 0, 1, \ldots, T - 1$, между векторами $(v(0), v(1), \ldots, v(T-1))$ и $(v(t), v(t+1), \ldots, v(t+T-1))$, т. е. между циклами последовательностей v и $x^t v$.

Рассмотрим автокорреляционную функцию последовательности v [5; 9, с. 123], определённую равенством

$$C_v(t) = \sum_{i=0}^{T-1} (-1)^{v(i) \oplus v(i+t)}, \quad i = 0, 1, \ldots, T - 1. \tag{5}$$

Из равенства $C_v(t) = T - 2\rho(v, v_t)$ получим

$$\rho(v, x^t v) = (T - C_v(t))/2. \tag{6}$$

Всюду в дальнейшем изучается ситуация, когда для всех $j = 1, 2, \ldots, k$ выполнено

$$(u_j(0), u_j(1), \ldots, u_j(m_j - 1)) \neq (u_j(t), u_j(t+1), \ldots, u_j(t + m_j - 1)), \tag{7}$$

т. е. при выработке v и $x^t v$ все соответствующие начальные заполнения регистров сдвига различны.

Лемма 1. Условие (7) равносильно тому, что t не кратно каждому из чисел $2^{m_1} - 1, \ldots, 2^{m_k} - 1$.

Доказательство. При каждом $j = 1, 2, \ldots, k$ соотношение

$$(u_j(0), u_j(1), \ldots, u_j(m_j - 1)) = (u_j(t), u_j(t+1), \ldots, u_j(t + m_j - 1))$$

означает, что $u_j = x^t u_j$, т. е. $(x^t - 1)u_j = (0)$. Это равносильно тому, что $T(u_j) = 2^{m_j} - 1$ делит t. ■

Теорема 7. Пусть $P = \mathrm{GF}(2)$, $F_1(x), \ldots, F_k(x)$ — примитивные многочлены над полем P попарно взаимно простых степеней $m_1, \ldots, m_k$ соответственно, $u_1, \ldots, u_k$ — ненулевые ЛРП, $T = (2^{m_1} - 1) \cdots (2^{m_k} - 1)$, функция φ существенно зависит от всех своих переменных. Тогда для всех t, не кратных каждому из чисел $2^{m_1} - 1$, …, $2^{m_k} - 1$,

$$C_v(t) = T\left(1 - \frac{\|\varphi\|}{2^{k-1}}\right)^2 + \frac{T}{2^{k-1}}\left(1 - \frac{\|\varphi\|}{2^{k-1}}\right) \sum_{\mathbf{a} \in P^k \setminus \{\mathbf{0}\}} \frac{(-1)^{\|\mathbf{a}\|} W_\varphi(\mathbf{a})}{(2^{m_1} - 1)^{a_1} \ldots (2^{m_k} - 1)^{a_k}} +$$
$$+ \frac{T}{2^k} \sum_{\substack{\mathbf{a} = (a_1, \ldots, a_k), \\ \mathbf{b} = (b_1, \ldots, b_k) \in P^k \setminus \{\mathbf{0}\}}} \frac{(-1)^{\|(a_1 \vee b_1, \ldots, a_k \vee b_k)\|} W_\varphi(\mathbf{a}) W_\varphi(\mathbf{b})}{(2^{m_1} - 1)^{a_1 \vee b_1} \ldots (2^{m_k} - 1)^{a_k \vee b_k}}.$$

Доказательство. Из равенств (2) и (5) имеем

$$C_v(t) = \sum_{i=0}^{T-1} (-1)^{v(i)} (-1)^{v(i+t)} = \frac{1}{2^{2k}} \sum_{\mathbf{a}, \mathbf{b} \in P^k} W_\varphi(\mathbf{a}) W_\varphi(\mathbf{b}) \sum_{i=0}^{T-1} (-1)^{\omega_1(i) \oplus \cdots \oplus \omega_k(i)},$$

где для каждого $j = 1, 2, \ldots, k$ последовательность ω_j определена равенством $\omega_j = (b_j x^t + a_j) u_j$. Последовательность ω_j является ЛРП с характеристическим многочленом $F_j(x)$, причём она нулевая тогда и только тогда, когда $F_j(x)$ делит $b_j x^t + a_j$. По выбору числа t это выполнено только при условии $a_j = b_j = 0$. Отсюда с использованием утверждения 4 получаем следующее равенство:

$$\sum_{i=0}^{T-1} (-1)^{\omega_1(i) \oplus \cdots \oplus \omega_k(i)} = \frac{(-1)^{\|(a_1 \vee b_1, \ldots, a_k \vee b_k)\|} T}{(2^{m_1} - 1)^{a_1 \vee b_1} \ldots (2^{m_k} - 1)^{a_k \vee b_k}}.$$

Значит,

$$C_v(t) = \frac{T}{2^{2k}} \sum_{\mathbf{a}, \mathbf{b} \in P^k} W_\varphi(\mathbf{a}) W_\varphi(\mathbf{b}) \frac{(-1)^{\|(a_1 \vee b_1, \ldots, a_k \vee b_k)\|} T}{(2^{m_1} - 1)^{a_1 \vee b_1} \ldots (2^{m_k} - 1)^{a_k \vee b_k}}.$$

Выделяя отдельно случаи $\mathbf{a} = \mathbf{b} = \mathbf{0}$; $\mathbf{a} = \mathbf{0}$, $\mathbf{b} \in P^k \setminus \{\mathbf{0}\}$; $\mathbf{b} = \mathbf{0}$, $\mathbf{a} \in P^k \setminus \{\mathbf{0}\}$ и используя равенство $W_\varphi(\mathbf{0}) = 2^k - 2\|\varphi\|$, получим нужную формулу. ■

Теорема 7 и равенство (6) позволяют найти точные значения для расстояний Хэмминга $\rho(v, x^t v)$. Например, пусть, в условиях теоремы 7, $\varphi(x_1, \ldots, x_k) = x_1 \oplus \cdots \oplus x_k \oplus \varepsilon$, где $\varepsilon \in \{0, 1\}$. Тогда (см. утверждение 5)

$$C_v(t) = \frac{T(-1)^k W_\varphi(1, \ldots, 1)^2}{2^{2k}(2^{m_1} - 1) \cdots (2^{m_k} - 1)} = (-1)^k, \quad \rho(v, x^t v) = \frac{T + (-1)^{k-1}}{2}.$$

В этом случае рассматриваемые векторы отличаются примерно в половине координат.

Рассмотрим ещё один пример. Пусть, в условиях теоремы 7, $\varphi(x_1, \ldots, x_k) = x_1 \oplus \oplus \cdots \oplus x_{k-2} \oplus x_{k-1} x_k$. Тогда (см. утверждение 8)

$$C_v(t) = \frac{T}{2^{2k}} \sum_{\mathbf{a}, \mathbf{b} \in \{(1\ldots100), (1\ldots101), (1\ldots110), (1\ldots1)\}} \frac{(-1)^{(-1)^{\|(a_1 \vee b_1, \ldots, a_k \vee b_k)\|}} W_\varphi(\mathbf{a}) W_\varphi(\mathbf{b})}{(2^{m_1} - 1)^{a_1 \vee b_1} \ldots (2^{m_k} - 1)^{a_k \vee b_k}}.$$

Отсюда нетрудно получить равенства

$$C_v(t) = (-1)^k (2^{m_{k-1} + m_{k-2}} - 2^{m_{k-1}} - 2^{m_k} + 1),$$
$$\rho(v, x^t v) = (T + (-1)^{k-1} (2^{m_{k-1} + m_{k-2}} - 2^{m_{k-1}} - 2^{m_k} + 1))/2.$$

Установим оценку сверху величины $C_v(t)$ для корреляционно-иммунных функций усложнения.

Следствие 3. Пусть, в условиях теоремы 7, $m_1 < m_2 < \ldots < m_k$ и φ является корреляционно-иммунной порядка d функцией. Тогда

$$\left|C_v(t) - T\left(1 - \frac{\|\varphi\|}{2^{k-1}}\right)\right| \leqslant (2^{m_{d+2}}-1)\cdots(2^{m_k}-1)|M(\varphi)\backslash\{\mathbf{0}\}|^{1/2}\left(2 - \frac{\|\varphi\|}{2^{k-2}} + |M(\varphi)\backslash\{\mathbf{0}\}|^{1/2}\right).$$

Доказательство. Из теоремы 7 получим

$$\left|C_v(t) - T\left(1 - \frac{\|\varphi\|}{2^{k-1}}\right)^2\right| \leqslant \frac{T}{2^{k-1}}\left(1 - \frac{\|\varphi\|}{2^{k-1}}\right)\sum_{\mathbf{a}\in P^k\backslash\{\mathbf{0}\}}\frac{|W_\varphi(\mathbf{a})|}{(2^{m_1}-1)^{a_1}\ldots(2^{m_k}-1)^{a_k}} +$$
$$+\frac{T}{2^{2k}}\sum_{\substack{\mathbf{a}=(a_1,\ldots,a_k),\\ \mathbf{b}=(b_1,\ldots,b_k)\in P^k\backslash\{\mathbf{0}\}}}\frac{|W_\varphi(\mathbf{a})||W_\varphi(\mathbf{b})|}{(2^{m_1}-1)^{a_1\vee b_1}\ldots(2^{m_k}-1)^{a_k\vee b_k}}.$$

Так как $W_\varphi(\mathbf{a}) = 0$ для всех векторов $\mathbf{a}\in P^k$ веса $1 \leqslant \|\mathbf{a}\| \leqslant d$ [7, теорема 7.10], то

$$\left|C_v(t) - T\left(1 - \frac{\|\varphi\|}{2^{k-1}}\right)^2\right| \leqslant \frac{(2^{m_{d+2}}-1)\cdots(2^{m_k}-1)(2^k - 2\|\varphi\|)}{2^{2k-1}}\sum_{\mathbf{a}\in P^k\backslash\{\mathbf{0}\}}|W_\varphi(\mathbf{a})| +$$
$$+\frac{(2^{m_{d+2}}-1)\cdots(2^{m_k}-1)}{2^{2k}}\left(\sum_{\mathbf{a}\in P^k\backslash\{\mathbf{0}\}}|W_\varphi(\mathbf{a})|\right)^2.$$

Для завершения доказательства остаётся воспользоваться неравенством

$$\sum_{\mathbf{a}\in P^k\backslash\mathbf{0}}|W_\varphi(\mathbf{a})| \leqslant |M(\varphi)\backslash\{\mathbf{0}\}|^{1/2}2^k,$$

полученным при доказательстве теоремы 6. ■

Следствие 4. Пусть, в условиях теоремы 7, $m_1 < m_2 < \ldots < m_k$ и φ является d-устойчивой булевой функцией. Тогда

$$|C_v(t)| \leqslant (2^{m_{d+2}}-1)\cdots(2^{m_k}-1)|M(\varphi)\backslash\{\mathbf{0}\}| \leqslant (2^{m_{d+2}}-1)\cdots(2^{m_k}-1)(2^k - s(k,d)),$$
$$\left|\rho(v, x^t v) - \frac{T}{2}\right| \leqslant (2^{m_{d+2}}-1)\cdots(2^{m_k}-1)\frac{|M(\varphi)\backslash\{\mathbf{0}\}|}{2} \leqslant (2^{m_{d+2}}-1)\cdots(2^{m_k}-1)\frac{2^k - s(k,d)}{2}.$$

Доказательство. Достаточно воспользоваться оценкой $|M(\varphi)\backslash\{\mathbf{0}\}| \leqslant 2^k - s(k,d)$, полученной при доказательстве следствия 2, и равенством (6). ■

Оценки из следствий 3 и 4 достижимы для булевой функции $\varphi(x_1,\ldots,x_k) = x_1 \oplus x_2 \oplus \cdots \oplus x_k \oplus \varepsilon$, где $\varepsilon \in \{0,1\}$.

ЛИТЕРАТУРА

1. *Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В.* Основы криптографии. М.: Гелиос АРВ, 2001. 480 с.
2. *Фомичев В. М.* Методы дискретной математики в криптологии. М.: Диалог-МИФИ, 2010. 424 с.
3. *Rueppel R. A.* Analysis and Design of Stream Ciphers. Springer Verlag, 1986. 244 p.
4. *Глухов М. М., Елизаров В. П., Нечаев А. А.* Алгебра. Т. 2. М.: Гелиос АРВ, 2003. 416 с.
5. *Лидл Р., Нидеррайтер Г.* Конечные поля. М.: Мир, 1988. Т. 1, 2. 822 с.

6. *Фомичев В. М.* О периодах усложненных последовательностей // Математические вопросы кибернетики. Вып. 13. М.: Физматлит, 2004. С. 37–40.
7. *Логачев О. А., Сальников А. А., Смышляев С. В., Ященко В. В.* Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2012. 584 с.
8. *Niederreiter H.* Weights of cyclic codes // Information and Control. 1977. V. 34. P. 130–140.
9. *Golomb S. W. and Gong G.* Signal Design for Good Correlation. Cambridge, 2005. 438 p.

REFERENCES

1. *Alferov A. P., Zubov A. Yu., Kuz'min A. S., Cheremushkin A. V.* Osnovy kriptografii [Basics of Cryptography]. Moscow, Gelios ARV Publ., 2001. 480 p. (in Russian)
2. *Fomichev V. M.* Metody diskretnoy matematiki v kriptologii. Moscow, Dialog-MIFI Publ., 2010. 424 p. (in Russian)
3. *Rueppel R. A.* Analysis and Design of Stream Ciphers. Springer Verlag, 1986. 244 p.
4. *Glukhov M. M., Elizarov V. P., Nechaev A. A.* Algebra. V. 2. Moscow, Gelios ARV Publ., 2003. 416 p. (in Russian)
5. *Lidl R., Niderrayter G.* Konechnye polya [Finite Fields]. Moscow, Mir Publ., 1988, vol. 1, 2. 822 p. (in Russian)
6. *Fomichev V. M.* O periodakh uslozhnennykh posledovatel'nostey [On periods of complicated sequences]. Matematicheskie Voprosy Kibernetiki, iss. 13. Moscow, Fizmatlit Publ., 2004, pp. 37–40. (in Russian)
7. *Logachev O. A., Sal'nikov A. A., Smyshlyaev S. V., Yashchenko V. V.* Bulevy funktsii v teorii kodirovaniya i kriptologii [Boolean Functions in Coding Theory and Cryptology]. Moscow, MCCME Publ., 2012. 584 p. (in Russian)
8. *Niederreiter H.* Weights of cyclic codes. Information and Control, 1977, vol. 34, pp. 130–140.
9. *Golomb S. W. and Gong G.* Signal Design for Good Correlation. Cambridge, 2005. 438 p.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 003.26; 512.5

НОВАЯ СЕМАНТИЧЕСКИ СТОЙКАЯ СИСТЕМА ШИФРОВАНИЯ С ОТКРЫТЫМ КЛЮЧОМ НА БАЗЕ RSA[1]

В. А. Романьков

Омский государственный университет им. Ф. М. Достоевского, г. Омск, Россия

Представлена семантически стойкая система шифрования с открытым ключом на базе системы шифрования RSA. Кроме семантической стойкости, описаны другие преимущества данной системы по отношению к базовой системе RSA, в том числе возможность более широкого выбора ключей и возможность выбора ключа шифрования пользователем. Показано, что в предлагаемой системе, в отличие от RSA, для дешифрования недостаточно знания разложения модуля на множители.

Ключевые слова: *семантическая стойкость, система шифрования с открытым ключом, RSA, платформа шифрования, ключи шифрования и расшифрования, мультипликативная группа кольца вычетов, подгруппа квадратичных вычетов.*

DOI 10.17223/20710410/29/3

A SEMANTICALLY SECURE PUBLIC-KEY CRYPTOSYSTEM BASED ON RSA

V. A. Romankov

Omsk State University, Omsk, Russia

E-mail: romankov48@mail.ru

We present a semantically secure public-key cryptosystem based on the RSA cryptosystem. We describe possible preferences of the proposed cryptosystem with respect to the basic RSA cryptosystem. These preferences include a semantic security property, as well as more various choice of an encryption key, and the possibility to select this key by an ordinary user. It is shown that the knowledge of the modulus factorization does not allow to break the cryptosystem as it happens in the basic RSA.

Keywords: *semantic security, public-key cryptosystem, RSA cryptosystem, encryption platform, encryption and decryption keys, the multiplicative group of a residue ring, the subgroup of quadratic residues.*

Введение

Система RSA, введённая в обращение Роном Ривестом, Ади Шамиром и Леном Адлеманом в 1977 г. [1], является наиболее популярной криптографической системой

[1]Работа выполнена при финансовой поддержке РФФИ, проект № 15.41.04312.

с открытым ключом. Она широко используется во всём мире как для шифрования, так и для цифровой подписи. В настоящее время RSA входит во все учебники по криптографии. Ей посвящено немало статей, она выдержала множество атак. Не раз объявлялось, что система исчерпала себя, но по-прежнему подобные утверждения остаются безосновательными. Относительно системы RSA и её криптографической стойкости см. [2–4].

В то же время может создаться впечатление, что система RSA существует в законченном раз и навсегда зафиксированном виде. В отличие от систем, использующих дискретные логарифмы, для неё не придумано эллиптических или алгебраических аналогов. Возникает вопрос: почему так происходит? Есть ли что-то такое в самой системе RSA, что не позволяет рассматривать различные её версии, не даёт возможности использовать платформы шифрования, отличные от колец вычетов по модулю, являющемуся произведением двух (больших) различных простых чисел? Думается, что это не так. В настоящей работе предложены достаточно простые и естественные способы разнообразить формы использования системы RSA и дан краткий анализ новых возможностей. Мы оставляем в стороне различные технические вопросы, связанные с правильным выбором параметров системы, тонкостями её имплементации и т. п. Основное внимание будет сосредоточено на вопросах общего характера.

В дальнейшем $\mathbb{Z}_n$ обозначает кольцо вычетов по модулю n, который в данной работе является произведением двух (больших) различных простых чисел p и q. Через $\mathbb{Z}_n^*$ обозначается мультипликативная группа кольца $\mathbb{Z}_n$. Через $\mathbb{Z}^{(2)}$ обозначаем множество всех чисел, представляющихся в виде произведения двух различных простых нечётных чисел. Таким образом, $n \in \mathbb{Z}^{(2)}$.

1. Классическая версия RSA

Установка системы RSA

Платформой для системы RSA служит кольцо вычетов $\mathbb{Z}_n$, где модуль n есть произведение двух различных достаточно больших простых чисел p и q. Числа p и q являются секретными, модуль n — открытым. Из соображений секретности числа p и q выбираются случайным образом. Считается, что p и q должны быть примерно одинаковой битовой длины.

После выбора p и q, однозначно определяющих модуль n и платформу $\mathbb{Z}_n$, вычисляется функция Эйлера $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1)$. Её значение сохраняется в секрете.

Напомним, что $\varphi(n)$ равно числу натуральных чисел, не больших n и взаимно простых с n. Записываемые этими числами вычеты, и только они, являются обратимыми по модулю n, то есть обратимыми в кольце $\mathbb{Z}_n$. Значит, $\varphi(n) = |\mathbb{Z}_n^*|$ является порядком группы $\mathbb{Z}_n^*$ обратимых вычетов кольца $\mathbb{Z}_n$.

Далее выбирается ключ зашифрования e — целое число в пределах $2 \leqslant e \leqslant \varphi(n)-1$, такое, что e и $\varphi(n)$ взаимно простые. В дальнейшем e используется как открытый ключ зашифрования (экспонента зашифрования).

Однозначно вычисляется число d, такое, что $ed = 1 \pmod{\varphi(n)}$. Вычет $d = e^{-1} \bmod \varphi(n)$ является обратным элементом к e в группе $\mathbb{Z}_{\varphi(n)}^*$.

Число d является секретным и используется в дальнейшем в качестве ключа расшифрования (экспоненты расшифрования). Оно эффективно вычисляется по e и $\varphi(n)$ расширенным алгоритмом Евклида.

Открытые данные: модуль n и ключ зашифрования e.

Секретные данные: параметры p и q, значение функции Эйлера $\varphi(n)$, ключ расшифрования d.

Алгоритмы зашифрования и расшифрования

Для передачи своего сообщения m, представленного как элемент кольца $\mathbb{Z}_n$, корреспондент Алиса должна знать открытый ключ (n, e) корреспондента Боба, которому предназначено данное сообщение. Сообщение m записывается своим стандартным значением, то есть $0 \leqslant m \leqslant n-1$. Зашифрованное сообщение c вычисляется по правилу

$$c = m^e \bmod n. \tag{1}$$

При этом для вычета c, передаваемого по открытой сети, используется его стандартное значение, то есть $0 \leqslant c \leqslant n-1$.

Чтобы расшифровать сообщение, Боб использует свой секретный ключ расшифрования d, выполняя следующее действие:

$$c^d = m^{ed} = m \pmod{n}. \tag{2}$$

Справедливость равенства (2) в случае обратимого вычета m следует из формулы Эйлера $m^{\varphi(n)} = 1 \pmod{n}$. Если m необратим, равенство (2) также имеет место (см., например, [5]).

Криптостойкость системы RSA основывается на трудности решения задачи о разложении на множители больших чисел вида $n = pq$. Эта задача решается уже сотни лет, но эффективный алгоритм разложения в общем случае так и не найден. Если потенциальный взломщик или нелигитимный пользователь сумеет найти числа p и q, он сможет вычислить ключ d и читать все сообщения, предназначенные в данной системе Бобу. Известно также (см., например, [6, с. 94–95]), что по ключу d можно, используя метод Монте-Карло, вычислить параметры p и q, что полностью разоблачает и делает непригодной к использованию установленную Бобом систему.

Из приведённых рассуждений, однако, не следует, что криптостойкость системы RSA равносильна трудноразрешимости проблемы разложения чисел вида $n = pq$ на множители. Известны атаки, в которых использованы другие соображения (см., например, [7]). Возможно, что существует ещё не найденный универсальный способ прочтения сообщения m без вычисления секретных параметров p, q и ключа расшифрования d. Но это только предположение.

2. Индивидуальные и универсальные ключи

Пусть порядок $|m|$ элемента m группы Z_n^* равен t. Это означает, что наименьшая степень, в которой элемент m равен 1 в группе $\mathbb{Z}_n^*$, равна t. Тогда зашифрованное сообщение $c = m^e \bmod n$ может быть расшифровано индивидуальным ключом d_t, определяемым равенством $ed_t = 1 \pmod{t}$. Ключ d_t может оказаться существенно меньше, чем универсальный ключ расшифрования d. Он подходит для расшифрования всех зашифрованных сообщений вида g^e, таких, что $g^t = 1$ в группе $\mathbb{Z}_n^*$ (или, что то же самое, $g^t = 1 \pmod{n}$). Действительно, из равенства $ed_t = 1 \pmod{t}$ следует, что $ed_t = 1 + tl$ для некоторого целого l. Тогда $(g^e)^{d_t} = g^{ed_t} = g^{1+tl} = g \pmod{n}$. В частности, вычисление универсального ключа d, с помощью которого расшифровывается любое зашифрованное сообщение, основано на том, что для любого $g \in \mathbb{Z}_n^*$ по формуле Эйлера $g^{\varphi(n)} = 1 \pmod{n}$.

Пусть $\tau(\mathbb{Z}_n^*)$ обозначает период группы $\mathbb{Z}_n^*$, то есть наименьшее натуральное число t, такое, что $g^t = 1$ для любого элемента g группы $\mathbb{Z}_n^*$. Условие $g^t = 1 \pmod{n}$

равносильно одновременному выполнению условий $g^t = 1 \pmod p$ и $g^t = 1 \pmod q$. Так как мультипликативные группы $\mathbb{Z}_p^*$ и $\mathbb{Z}_p^*$ полей $\mathbb{Z}_p$ и $\mathbb{Z}_q$ циклические порядков $p-1$ и $q-1$ соответственно, для любого элемента g группы $\mathbb{Z}_n^*$ выполнено равенство $g^t = 1$, где $t = [p-1, q-1]$ ($[a,b]$ означает «наименьшее общее кратное чисел a и b»). Заметим, что такое t заведомо меньше, чем $\varphi(n)$, так как при больших простых p и q числа $p-1$ и $q-1$ оба чётные. Число t позволяет определить универсальный ключ расшифрования d_t, не всегда совпадающий с d.

Можно утверждать, что $[p-1, q-1]$ совпадает с периодом $\tau(\mathbb{Z}_n^*)$. Докажем это. Пусть $p-1 = \prod\limits_{i=1}^{k} p_i^{l_i}$ и $q-1 = \prod\limits_{i=1}^{k} p_i^{m_i}$ — разложения в произведения примарных множителей. Тогда $[p-1, q-1] = \prod\limits_{i=1}^{k} p_i^{r_i}$, где $r_i = \max(l_i, m_i)$, $i = 1, \ldots, k$. Достаточно доказать, что в группе $\mathbb{Z}_n^*$ найдётся элемент порядка $t = \prod\limits_{i=1}^{k} p_i^{r_i}$. Пусть $t = t_1 t_2$, где $t_1 = \prod\limits_{i \in \{1,\ldots,k\}: r_i = l_i} p_i^{r_i}$, $t_2 = \prod\limits_{i \in \{1,\ldots,k\}: l_i \neq m_i, r_i = m_i} p_i^{r_i}$. Тогда t_1 делит $p-1$ и t_2 делит $q-1$. Заметим также, что $(t_1, t_2) = 1$.

Пусть f — порождающий элемент группы $\mathbb{Z}_p^*$, h — порождающий элемент группы $\mathbb{Z}_q^*$. Элемент $g_1 = f^{(p-1)/t_1}$ имеет порядок t_1, элемент $g_2 = h^{(q-1)/t_2}$ — порядок t_2. Так как эти порядки взаимно просты, элемент $g = g_1 g_2$ имеет порядок t. Отсюда получаем требуемое утверждение.

Итак, система допускает два способа вычисления ключа расшифрования по соотношениям $ed = 1 \pmod{\varphi(n)}$ и $ed_t = 1 \pmod{\tau(n)}$ соответственно, где $\tau(n) = [p-1, q-1]$. В некоторых случаях эти ключи совпадают, иногда отличаются достаточно сильно. Например, при $p = 19$, $q = 37$, $n = 703$, $\varphi(n) = 648$, $\tau(n) = 36$ для ключа шифрования $e = 5$ получаем $d = 574$, $d_t = 29$.

Приведённые рассуждения наталкивают на мысль построения подсистемы данной системы RSA, позволяющей использовать ключи зашифрования, которые не могут быть таковыми для всей системы. В частности, оказалось возможным использовать в качестве e чётные числа, более того — степени двойки. Более подробно об этом говорится далее.

3. Новая семантически стойкая система шифрования с открытым ключом на базе RSA

Перейдём к описанию основной системы шифрования, предлагаемой в данной работе.

Установка: платформа шифрования

Пусть $n = pq \in \mathbb{Z}^{(2)}$. В качестве платформы для системы шифрования выбираем кольцо вычетов $\mathbb{Z}_n$.

Пусть M — подгруппа мультипликативной группы $G_n = \mathbb{Z}_n^*$, $r = |M|$ — её период (или порядок). Можно также использовать любое число r, делящееся на период подгруппы M. Считаем, что множество всех возможных сообщений m совпадает с M. Таким образом, M выступает в роли пространства сообщений.

Выберем другую подгруппу $H \leqslant G_n$ периода (или порядка) t, взаимно простого с r. В качестве t также можно взять любое число, делящееся на период подгруппы H и взаимно простое с r.

Данные n, M, H открыты. Данные p, q, r, t секретны.

Установка: ключи

Ключ зашифрования e — любое натуральное число, взаимно простое с r. Ключ расшифрования $d = td_1$ вычисляется из равенства

$$(te)d_1 = 1 \pmod r.$$

Это можно сделать, так как te и r — взаимно простые числа. Ключ e открытый, d — секретный.

Алгоритм зашифрования

Для того чтобы передать по незащищённой сети сообщение $m \in M$, Алиса выбирает случайный элемент $h \in H$ и вычисляет элемент $hm \in G_n$. Передача имеет вид

$$A : c = (hm)^e \bmod n \to B.$$

Алгоритм расшифрования

Боб расшифровывает полученное сообщение следующим образом:

$$B : c^d = m \bmod n.$$

Объяснение правильности расшифрования

Так как $ed = 1 \pmod r$, найдётся целое число k, такое, что $ed = 1 + rk$. Тогда

$$c^d = (hm)^{ed} = (h^t)^{ed_1} m^{1+rk} = m \pmod n.$$

Выбор и задание составляющих системы

Кроме обычных рекомендаций, касающихся системы шифрования RSA, нужно определить подгруппы H и M. Предлагается задавать эти подгруппы своими порождающими элементами. Например, они могут быть циклическими подгруппами $H = \text{гр}(h)$ и $M = \text{гр}(g)$, для задания которых достаточно определить элементы h и g взаимно простых порядков t и r соответственно.

Одним из эффективных способов выбора циклических подгрупп заданного порядка является следующая процедура, неоднократно описанная в разных источниках (см., например, [5]). Генерируется простое число p. Для этого выбирается простое число p_1. Случайным образом выбирается чётное число s в интервале $p_1 \leqslant s \leqslant 4p_1 + 2$. Полагаем $p_2 = p_1 s + 1$. Далее случайным образом ищется число a в интервале $2 \leqslant a \leqslant p_2 - 1$, удовлетворяющее условиям

$$a^{p_2 - 1} = 1 \pmod{p_2}, \quad (a^s - 1, p_2) = 1.$$

Если это удалось сделать, то p_2 простое. Мы либо останавливаемся, полагая $p = p_2$, либо продолжаем процесс с p_2 вместо p_1. Известно, что при простом p_2 вероятность выбора подходящего a не меньше $1 - 1/p_1$. Если после достаточного числа проверок число a с нужными свойствами подобрать не удаётся, меняем s. Практика показала, что найти указанным способом простое число удаётся с вероятностью близкой к 1. Таким образом строятся сколь угодно большие простые числа.

Заметим, что в этом построении при достаточно большом p_1 можно выбирать s с любым заранее заданным делителем r. Если при этом получается простое p, то r делит $s = p - 1$. В мультипликативной группе $\mathbb{Z}_p^*$ есть элемент g_1 нужного порядка r. Найти его можно известными способами, в том числе практическими. Для получения элемента g порядка r в группе G_n достаточно составить и решить по китайской теореме об остатках систему сравнений

$$\begin{cases} g = g_1 \pmod p, \\ g = 1 \pmod q, \end{cases}$$

где q — другой простой множитель модуля n.

Заметим, что построение простого числа p описанной процедурой можно осуществлять, получая в итоге не только простое число p, но и разложение на множители порядка $p-1$ мультипликативной группы поля $\mathbb{Z}_p^*$. Это даёт возможность строить подгруппы в $\mathbb{Z}_p^*$ заданных порядков.

Можно предложить и другой выбор простого числа, когда берутся числа вида $2s+1$ нужного размера и проверяются на простоту известными вероятностными алгоритмами. Здесь также можно контролировать разложение $p-1$ на множители. Заметим, что упомянутые два способа являются основными для построения простых параметров в криптографии.

Можно построить несколько элементов $g_1, \ldots, g_l$ соответствующих заданных порядков $r_1, \ldots, r_l$. Тогда порождённая ими подгруппа $\text{гр}(g_1, \ldots, g_l)$ имеет период r, делящий $[r_1, \ldots, r_l]$. Этого достаточно для задания подгруппы M. Аналогичным способом можно задавать и подгруппу H.

Свойства построенной системы шифрования

— Система является семантически стойкой. Это обеспечивается тем, что одно и то же сообщение $m \in M$ передается t различными случайно определяемыми способами. Поэтому, имея два сообщения m_1 и m_2, а также зашифрованное сообщение $(hm)^e \bmod n$, потенциальный взломщик не может определить, какое из двух сообщений зашифровано. Это одно из определений семантической стойкости.
— Ключом зашифрования e может быть любое число, взаимно простое с r. Если, например, r — простое число, то можно допустить выбор ключа e самим пользователем. Действительно, пользователь может выбирать из интервала с правой границей, заведомо меньшей r. Да и в общем случае при большом неизвестном простом r вероятность случайного выбора числа e, делящегося на r, пренебрежимо мала.
— Для компрометации предлагаемой системы шифрования недостаточно знания параметров p и q.

Объясним последнее из перечисленных свойств.

Заметим, что проблема определения порядка произвольного элемента группы $\mathbb{Z}_n^*$ влечёт решение проблемы RSA, то есть определение зашифрованного сообщения m по $c = m^e \bmod n$. Значит, эту проблему можно считать трудноразрешимой. Знание параметров p и q также не позволяет вычислять порядки элементов группы $\mathbb{Z}_n^*$ за полиномиальное время; это возможно только при знании разложений $p-1$ и $q-1$. Действительно, можно сделать так, что $p - 1 = 2n'$, $n' = p'q'$, где p' и q' — большие различные простые числа. Если уметь вычислять порядки случайных элементов мультипликативной группы $\mathbb{Z}_n^*$, то можно получить разложение n'.

При знании p и q можно вычислить $hm \bmod n$ и попытаться определить множители h и m. Эту задачу также можно рассматривать как трудноразрешимую. Например, если взять в качестве M подгруппу квадратичных вычетов в группе $\mathbb{Z}_n^*$, проблема вхождения в которую традиционно рассматривается как трудноразрешимая, то при наличии способа определения множителей h и m можно определить по $hm \bmod n$

принадлежность этого элемента к подгруппе квадратичных вычетов. Это происходит только в случае, когда $h = 1 \pmod{n}$.

Если всё же потенциальному взломщику системы удастся определить параметр t, то для нахождения m ему придётся решать обычную проблему RSA. Заметим, что предлагаемая схема шифрования сводится к RSA при выборе в качестве H единичной подгруппы, а в качестве M — всей мультипликативной группы $\mathbb{Z}_n^*$.

4. Шифрование на подгруппе квадратичных вычетов

Представляется естественной следующая версия RSA, которую назовём *квадратичной подсистемой* RSA.

Установка квадратичной подсистемы RSA

При выборе параметров p и q на них накладывается дополнительное требование: они должны иметь вид $p = 4k + 3$ и $q = 4l + 3$. В качестве платформы шифрования выбирается подгруппа $Q_n \leqslant \mathbb{Z}_n^*$, состоящая из всех квадратичных вычетов группы $\mathbb{Z}_n^*$. Так как каждый элемент $g \in Q_n$ имеет в точности четыре различных корня второй степени, порядок подгруппы Q_n равен нечётному числу $t = \varphi(n)/4 = (2k + 1)(2l + 1)$. Более того, при таком выборе простых чисел возведение в квадрат определяет биективное отображение группы Q_n на себя. Следовательно, извлечение квадратных корней в группе Q_n однозначно. При знании параметров p и q процесс извлечения квадратных корней в группе $\mathbb{Z}_n^*$ и проверки их на принадлежность подгруппе Q_n может быть организован эффективно. Если указанные параметры неизвестны, извлечение корней равносильно разложению модуля на множители. Проверка принадлежности вычета группе Q_n в этом случае также представляется трудноразрешимой задачей.

В качестве ключа шифрования на группе Q_n выбирается любое натуральное число e, взаимно простое с её порядком t. Например, можно взять $e = 2^s$, где s — произвольное натуральное число. Ключ расшифрования d получается из соотношения $ed = 1 \pmod{t}$.

Алгоритмы шифрования и расшифрования

Шифрование осуществляется обычным способом. Зашифрованное сообщение c для $m \in Q_n$ вычисляется по формуле (1). В качестве ключа зашифрования e можно брать любое число, взаимно простое с порядком подгруппы Q_n. Если выбирается нечётный ключ e, то условие на параметры p и q (остаток 3 при делении на 4) можно не накладывать. В особом случае получается

$$c = m^{2^s} \bmod n.$$

Если $e = 2^s$, то расшифрование можно осуществить и другим способом, последовательно извлекая s квадратных корней, начиная с c, оставляя для дальнейших вычислений тот из корней, что принадлежит Q_n. Знание параметров p и q позволяет проводить такой процесс эффективно. В основном мы рассматриваем такую возможность как чисто теоретическую. Указанный способ шифрования можно сравнить с известной системой шифрования Рабина [8]. Основное отличие состоит в том, что платформой шифрования в предлагаемой системе служит не вся группа $\mathbb{Z}_n^*$, а только её подгруппа квадратичных вычетов Q_n. В системе Рабина используется возведение в квадрат, но для любого вычета $m \in \mathbb{Z}_n^*$. Расшифрование даёт четыре варианта для m, из которых только один является правильным. В предлагаемой системе можно использовать возведение в любую степень вида $2^s l$, где l взаимно просто с $\varphi(n)/4$, расшифрование однозначно. Для зашифровывания произвольного $m \in \mathbb{Z}_n^*$ можно использовать следующие соображения.

Замечание 1. Элементы подгруппы Q_n — полные квадраты. Если мы хотим передавать любые сообщения m в пределах от 0 до $n-1$, как это делается в RSA, можно использовать теорему Лагранжа о представлении любого натурального числа m в виде суммы четырёх квадратов $m_1^2+m_2^2+m_3^2+m_4^2$. Известно, что нахождение такого представления для произвольного m эффективно. Остается передать слагаемые, по которым сообщение восстанавливается однозначно. Для упрощения можно даже передать любой квадрат $u^2 \leqslant m$ и открытым способом разницу $m-u^2$.

Использование квадратичной подсистемы RSA имеет ряд отличий, которые можно в определённом смысле считать преимуществами:

— возможность использования чётных ключей шифрования;
— возможность использования ключей шифрования вида 2^s, которые могут выбирать пользователи подсистемы RSA, а не только её владелец;
— ключи вида 2^s могут меняться от раунда к раунду. Это, например, позволяет передавать одинаковые сообщения в разных видах, что даёт возможность обеспечить семантическую секретность;
— расшифрование при использовании ключей вида 2^s можно осуществлять одной и той же программой вычисления квадратных корней в группе Q_n. Отличие только в s — числе шагов работы программы. Процесс расшифрования может быть организован без знания этого параметра;
— шифрование в квадратичной подсистеме можно включить как часть системы, описанной в предыдущем пункте: подгруппу Q_n можно выбрать в качестве фигурирующей там подгруппы M.

Последнее из приведённых свойств представляется наиболее важным для приложений.

5. Использование общего модуля n

В литературе неоднократно объяснялось, что знание ключа расшифрования d позволяет методом Монте-Карло эффективно раскрыть секретные параметры p и q и тем самым дискредитировать данную систему RSA (см., например, [6, с. 94–95]). Однако можно представить себе некоторую сеть, пользователи которой обладают системами RSA с общим модулем n. В сеть передаются шифрованные сообщения, общие для всех пользователей; например, пользователи — отделения одной и той же организации. Раскрывать секретные параметры легитимным пользователям нецелесообразно. В этом случае имеет смысл наделять пользователей разными парами (e,d) ключей шифрования и расшифрования. В противном случае, если будет достаточно много одинаковых ключей шифрования e, нелигитимный пользователь может прочитать зашифрованное сообщение, используя китайскую теорему об остатках. Этот способ также хорошо известен (см., например, [2, с. 51; 5, с. 105]).

При наделении пользователей различными ключами шифрования, в особенности при случайном их выборе, легко возникает ситуация, когда часть этих ключей, скажем $e_1,\ldots,e_l$, окажется в совокупности попарно взаимно простой. Тогда расширенный алгоритм Евклида позволяет эффективно вычислить целые числа $k_1,\ldots,k_l$, такие, что $\sum\limits_{i=1}^{l} e_ik_i = 1$, что позволяет по соответствующему набору зашифрованных вариантов $c_i = m^{e_i} \bmod n$, $i=1,\ldots,l$, одного и того же сообщения m восстановить это сообщение:

$$\prod_{i=1}^{l} c_i^{k_i} = m^{\sum\limits_{i=1}^{l} e_ik_i} = m \pmod{n}.$$

Такое восстановление невозможно при использовании ключей вида 2^s в квадратичной подсистеме RSA.

ЛИТЕРАТУРА

1. *Rivest R., Shamir A., and Adleman L.* A method for obtaining digital signatures and public-key cryptosystems // Comm. ACM. 1978. V. 21(2). P. 120–126.
2. *Hinek M. J.* Cryptanalysis of RSA and its Variants. Boca Raton: Chapman & Hall/CRC, 2010.
3. *Song Y. Y.* Cryptanalytic Attacks on RSA. Springer, 2008.
4. *Stamp M. and Low R. M.* Applied Cryptanalysis. Breaking Ciphers in the Real World. Hoboken: John Wiley & Sons, 2007.
5. *Романьков В. А.* Введение в криптографию. М.: Форум, 2012. 239 с.
6. *Koblitz N.* A Course in Number Theory and Cryptography. N.Y.: Springer, 1994. 235 p.
7. *Maurer U. M.* Fast generation of prime numbers and secure public-key cryptographic parameters // Cryptology. 1995. V. 8. P. 123–155.
8. *Rabin M. O.* Digitalized Signatures and Public Key Functions as Intractable as Factorization. Technical Report. Cambridge: MIT, 1979.

REFERENCES

1. *Rivest R., Shamir A., and Adleman L.* A method for obtaining digital signatures and public-key cryptosystems. Comm. ACM, 1978, vol. 21(2), pp. 120–126.
2. *Hinek M. J.* Cryptanalysis of RSA and its Variants. Boca Raton, Chapman & Hall/CRC Publ., 2010.
3. *Song Y. Y.* Cryptanalytic Attacks on RSA. Springer, 2008.
4. *Stamp M. and Low R. M.* Applied Cryptanalysis. Breaking Ciphers in the Real World. Hoboken, John Wiley & Sons, 2007.
5. *Romankov V. A.* Vvedenie v kriptografiyu [Introduction to Cryptography]. Moscow, Forum Publ., 2012. 239 p. (in Russian)
6. *Koblitz N.* A Course in Number Theory and Cryptography. N.Y., Springer, 1994. 235 p.
7. *Maurer U. M.* Fast generation of prime numbers and secure public-key cryptographic parameters. Cryptology, 1995, vol. 8, pp. 123–155.
8. *Rabin M. O.* Digitalized Signatures and Public Key Functions as Intractable as Factorization. Technical Report. Cambridge, MIT, 1979.

УДК 519.7 DOI 10.17223/20710410/29/4

PROBLEMS, SOLUTIONS AND EXPERIENCE OF THE FIRST INTERNATIONAL STUDENT'S OLYMPIAD IN CRYPTOGRAPHY[1]

S. Agievich*, A. Gorodilova**, N. Kolomeec**,***, S. Nikova****, B. Preneel****, V. Rijmen****, G. Shushuev**,***, N. Tokareva**,***, V. Vitkup**

**Belarusian State University, Minsk, Belarus,*
***Sobolev Institute of Mathematics, Novosibirsk, Russia,*
****Novosibirsk State University, Novosibirsk, Russia,*
*****University of Leuven, KU Leuven, Belgium*

A detailed overview of the problems, solutions and experience of the first international student's Olympiad in cryptography, NSUCRYPTO'2014, is given. We start with the rules of participation and the description of rounds. All 15 mathematical problems of the Olympiad and their solutions are considered in detail. The problems are about differential characteristics of S-boxes, S-box masking, relations between cyclic rotation and additions modulo 2 and 2^n, special linear subspaces in $\mathbb{F}_2^n$, the number of solutions of the equation $F(x)+F(x+a)=b$ over the finite field $\mathbb{F}_{2^n}$ and APN functions. Some unsolved problems in symmetric cryptography are also considered.

Keywords: *cryptography, block ciphers, Boolean functions, AES, Olympiad, NSUCRYPTO.*

Introduction

The First Siberian Student's Olympiad in Cryptography with International participation — NSUCRYPTO'2014 was held on November 2014. There exist several school competitions in cryptography and information security, but this one is the first cryptographic Olympiad for students and professionals. The aim of the Olympiad was to involve students and young researchers in solving the curious and hard scientific problems of the modern cryptography. From the very beginning, the concept was not to stop on the training olympic tasks but to include unsolved research problems at the intersection of mathematics and cryptography.

In this article, we give a detailed overview of the Olympiad. We start with the rules of participation and the description of rounds. Then in two big sections, we discuss 15 problems of the Olympiad and their solutions. Among them, there are both some amusing tasks based on historical ciphers and hard mathematical problems.We consider mathematical problems related to cipher constructing such as studying differential characteristics of S-boxes, S-box masking, determining relations between cyclic rotation and additions modulo 2 and 2^n, constructing special linear subspaces in $\mathbb{F}_2^n$. Problems about the number of solutions of the equation $F(x) + F(x + a) = b$ over the finite field $\mathbb{F}_{2^n}$ and APN functions are discussed. Some unsolved problems are proposed. The problem about the special watermarking ciphers is one of them. All problems were developed by the Program committee of the Olympiad. Solution check was also its duty.

[1]Olympiad was supported by Novosibirsk State University. A. Gorodilova and N. Kolomeec thank RFBR grant No. 15-07-01328 and Grant NSh-1939.2014.1 of President of Russia for Leading Scientific Schools, G. Shushuev, N. Tokareva and V. Vitkup would like to thank RFBR grants No. 12-01-31097 and 15-31-20635 for the financial support.

Organizers of the Olympiad are Novosibirsk State University, Sobolev Institute of Mathematics (Novosibirsk), Tomsk State University, Belarusian State University and University of Leuven (KU Leuven, Belgium). Programm committee was formed by G. Agibalov, S. Agievich, N. Kolomeec, S. Nikova, I. Pankratova, B. Preneel, V. Rijmen, and N. Tokareva. Local organizing committee from Novosibirsk consisted of A. Gorodilova, N. Kolomeec, G. Shushuev, V. Vitkup, D. Pokrasenko and S. Filiyzin. N. Tokareva was the general chair of the Olympiad.

More than 450 participants from 12 countries were registered on the website of the Olympiad, `www.nsucrypto.nsu.ru`. Fifteen participants of the first round and eleven teams of the second round became winners and received prizes. The list of winners can be found in the last section of this paper.

NSUCRYPTO will be a regular annual Olympiad held on November. In 2015, it starts on November, 15. We invite pupils, students and professionals to participate!

1. Organization and rules of the Olympiad

Fig. 1.

Here we briefly formulate key points of the Olympiad. Its emblem is shown in Fig. 1.

Rounds of the Olympiad. There were two independent Internet rounds. The First round (duration 4 hours 30 minutes) was individual and consisted of two sections: school and student's. It was held on November, 16. Theoretical problems in mathematics of cryptography were offered to participants. The second, team, round (duration 1 week; November, 17–24) was devoted to hard research and programming problems of cryptography.

Everybody can participate! To become a participant of the Olympiad, it was necessary and sufficient to register on the website `www.nsucrypto.nsu.ru`. There were no restrictions on status and age of the participants. It means that senior pupils, students and all the others who are interested in cryptography were able to participate. Participants from any countries were welcome. During the registration, every participant had to choose his category: "senior pupil", "student" or "other / professional" and the section of the first round: "school" or "student's". The second round was common for all the participants.

Language of the Olympiad. All problems were given in English. But solutions could be written in English or Russian.

Format of the solutions. We accepted solutions in any electronic format (pdf, jpg, txt, rtf, docx, tex, etc). For example, a participant was able to write his solutions on a paper and send us a picture of it. Solutions should be written with all necessary details.

Prizes. There were several groups of prizes:

- for senior pupils — winners of the school section of the first round;
- for students — winners of the student's section of the first round;
- for participants in category "other / professional" — winners of the student's section of the first round;
- for participants (for every category separately) — winners of the second round;
- special prizes from the Programm committee for unsolved problems.

Interesting moments. Sometimes we were asked: "The Olympiad is via Internet. Are not you afraid that participants will use everything: supercomputers, books, articles, websites on cryptography?" In fact, we only welcome such an active mobilization of all possible resources in purpose of solving the tasks! We hope that, in a future, such a brainstorm will help to solve really hard cryptographic problems.

2. Problem structure of the Olympiad

There were 15 problems on the Olympiad. Some of them were included in both rounds.

Thus the school section of the first round consisted of 6 problems, whereas the student's section contained 8 problems. The first three problems were the same in each section (Tables 1, 2).

Table 1

Problems of the first round (school section)

N	Problem title	Maximal scores
1	A hidden message	4
2	A crypto room	4
3	The musical notation	4
4	Boolean cubes	4
5	A broken cipher machine	4
6	The Snowflake cipher	4

Table 2

Problems of the first round (student's section)

N	Problem title	Maximal scores
1	A hidden message	4
2	A crypto room	4
3	The musical notation	4
4	Linear subspaces	12
5	Number of solutions	8
6	A special parameter	10
7	S-box masking	8
8	Add–Rotate–Xor	10

The second round was composed of 11 problems (Table 3); it was common for all the participants. Three problems presented on the second round are unsolved (with declared special prizes from the Program Committee).

Table 3

Problems of the second round

N	Problem title	Maximal scores
1	Watermarking cipher	Special prize
2	APN permutation	Special prize
3	The Snowflake cipher	4
4	Number of solutions	8
5	Super S-box	Special prize
6	Boolean cubes	4
7	A special parameter	10
8	A pseudo-random generator	6
9	Add–Rotate–Xor	10
10	Linear subspaces	12
11	The musical notation	4

3. Problems

3.1. Problem "A hidden message" (4 scores)

CrYPtogRapHY iS a ScIEnce Of "seCrET wriTinG". FOr aT Least Two THoUsANd yeaRS ThErE haVE bEeN peOPlE WHo WAnTeD to SEnd MESsaGes WHiCh coUlD oNly bEen rEAd bY tHe pEOPLe FoR whOm tHey were iNteNdeD. a loT oF different MEtHODs FoR coNcEalING mEssageS WerE invENtED stARTING WIth AnCIeNt cIPHerS lIKE "SkytaLE" and "ATBAsH" and ending wiTH MOdErn SymmeTRiC ANd PubliC-kEy enCRYptioN ALGOriTHmS SUch aS AeS and Rsa. the dEVELopMENT Of crYPtOgRaPHy cOntiNueS And NEVER sTopS! decrYPt THe mESsaGe tHat iS hIDdEn in thE teXT oF this TASk! tHE aLphabet FoR THE mEssAGE ConsisTs of ALl tWEnTy six enGliSh letTERS from "a" To "z" ANd Six puNCTuaTIoN MARkS " ", ".", ",", "!", "?", "'".

3.2. Problem "A crypto room" (4 scores)

You are in a crypto room with a secret message in hands (Fig. 2). Decrypt it!

Fig. 2.

3.3. Problem "The musical notation" (4 scores)

Alice and Bob invented a new way for encrypting messages based on musical notations of melodies. They are not very good in musical notations, but they know the basic notes "do", "re", "mi", "fa", "sol", "la", "ti" and their places in the staff:

To encrypt a message of length n in English alphabet, Alice chooses a melody consisting of n notes. She writes the message under the musical notation of the melody in such a way that each letter of the message corresponds to exactly one note's position in the musical notation. Then, for each note ("do", "re", ..., "ti"), Alice writes the row of the corresponding letters, takes a random integer number k_i for each $i = 1, \ldots, 7$, and cyclically shifts letters in the i-th row by k_i positions to the right. Finally, Alice forms the ciphertext, writing letters of the shifted sets under the musical notation again.

An example. Suppose that Alice wants to send the message `H E L L O`.

The row for "re" is (`E L`); for "mi" — (`H L O`). Alice takes random numbers 2 and 1 for "re" and "mi" respectively. After cyclical shifting to the right the first row by 2 positions and the second row by 1 position, she gets rows (`E L`) and (`O H L`). Hence, the ciphertext for the message is `O E H L L`.

Decrypt the following ciphertext sent to Bob by Alice:

`R O L E L I S E O E E E H T O M V C P B D E F S O N`

It is known that Alice used the musical notation below.

3.4. Problem "Boolean cubes" (4 scores)

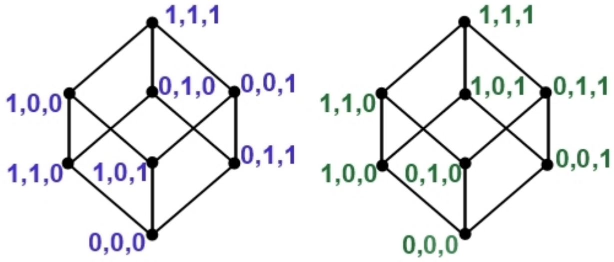
Alice has two cubes E_1 and E_2 of dimension 3 (Fig. 3). Their vertices have labels consisting of three integers; for example, (1,0,1) consists of integers 1, 0, 1. Consider an operation A that can be applied to a cube. The operation A contains three steps:

Step 1. Take an arbitrary edge of the cube;

Step 2. Take a number a which equals 1 or -1;

Step 3. Add a to an arbitrary position of the first vertex of the chosen edge. Add a to an arbitrary position of the second vertex of the edge.

Is it possible to get the cube E_2 from the cube E_1 by applying the operation A as many times as necessary? Give your arguments.



The cube E_1 **The cube E_2**

Fig. 3.

An example of applying the operation. Step 1. Take the edge $((1,0,0);(1,1,0))$. Step 2. Let $a=-1$. Step 3. For the vertex $(1,0,0)$, we choose the position 2 and, for the vertex $(1,1,0)$, we choose the position 1; after adding, the edge $((1,0,0);(1,1,0))$ becomes $((1,-1,0);(0,1,0))$ (Fig. 4).

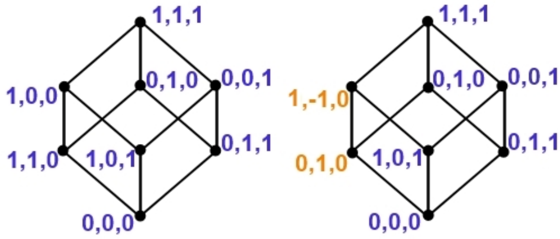


Fig. 4. An illustration of the example

3.5. Problem “A broken cipher machine” (4 scores)

Mary operates a cipher machine that encrypts messages like this:

Step 1. It represents a message as a natural number $n=\overline{abcdef\ldots}$;

Step 2. Then it adds all the digits in the number, $S_n=a+b+c+d+e+f+\ldots$;

Step 3. It inverts the order of digits in the number n and gets the number $n'=\overline{\ldots fedcba}$;

Step 4. As a result of the encryption, the machine prints the number $m=n'+2S_n$.

But now the cipher machine is broken: sometimes it works correctly, sometimes it prints a random number.

After encryption of her secret number n, Mary found out that the result m is the power of two,that is, $m=2^k$ for some integer k.

Determine: was the encryption correct in this case?

3.6. Problem “The Snowflake cipher” (4 scores)

Alice wants to encrypt some text using the Snowflake cipher. The encryption is described by the following algorithm:

Step 1. Choose an arbitrary small triangle in the snowflake (see Fig. 5);

Step 2. Put the first letter of your message into this triangle;

Step 3. Write the next letter of the message (without spaces) into an arbitrary empty neighbouring triangle. Neighbouring means having a common edge. Repeat this step until the end of the message.

Step 4. After inserting all the letters, write down the text from snowflake in horizontal order from top to bottom and from left to right.

Determine what is the maximal possible length of a message that can be encrypted with the Snowflake cipher?

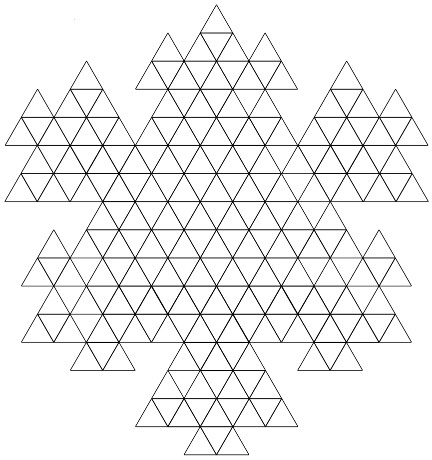


Fig. 5.

An example. We want to encrypt the message: `LOOK HOW IT WORKS`. As a result, we can get the ciphertext: `LHOWOOKITSKROW` (Fig. 6).

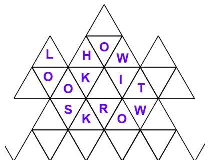


Fig.6.

3.7. Problem "A pseudo-random generator" (6 scores)

Alice and Bob communicate in Russia through the Internet, using some protocol. In the process of communication, Bob sends random numbers to Alice. It is known that Bob's pseudo-random generator works in the following way:

- it generates a binary sequence $u_0, u_1, u_2, \ldots$ such that, for some secret $c_0, \ldots, c_{15} \in \mathbb{F}_2$,
$$u_{i+16} = c_{15}u_{i+15} \oplus c_{14}u_{i+14} \oplus \ldots \oplus c_0u_i \text{ for all integer } i \geqslant 0;$$
- i-th random number r_i, $i \geqslant 1$, is calculated as
$$r_i = u_{16i} + u_{16i+1}2 + u_{16i+2}2^2 + \ldots + u_{16i+15}2^{15};$$
- Bob initializes $u_0, u_1, \ldots, u_{15}$, using some integer number IV (initial value) from the interval $(1, 2^{16} - 1)$ in the same way, that is,
$$IV = u_0 + u_12 + u_22^2 + \ldots + u_{15}2^{15};$$
- it is known that, as the value of IV, Bob uses the number t modulo 2^{16}, where t is the number of seconds from January 1, 1970, 00:00 (in Bob's time zone) to his current time (in his time zone too).

Eve has intercepted the third and the fourth random numbers, namely $r_3 = 9\,731$ and $r_4 = 57\,586$. She lives in Novosibirsk and knows that Bob has initialized the generator on November 17, 2014, at about 12:05 UTC+6 up to several minutes. The number of seconds from January 1, 1970, 00:00 UTC+6 to November 17, 2014, 12:05 UTC+6 is equal to 1 416 225 900. Help Eve to detect Bob's time zone.

3.8. Problem "Number of solutions" (8 scores)

Let $\mathbb{F}_{256}$ be the finite field of characteristic 2 with 256 elements. Consider the function
$$F : \mathbb{F}_{256} \to \mathbb{F}_{256} \text{ such that } F(x) = x^{254}.$$
Since $x^{255} = 1$ for all nonzero $x \in \mathbb{F}_{256}$, we have $F(x) = x^{-1}$ for all nonzero elements of $\mathbb{F}_{256}$. Further, we have $F(0) = 0$.

Alice is going to use the function F as an S-box that maps 8 bits to 8 bits in a block cipher. But before doing this, she wants to find answers to the following questions.

- How many solutions may the equation
$$F(x + a) = F(x) + b \tag{1}$$
have for all different pairs of parameters a and b with nonzero values from $\mathbb{F}_{256}$?
- How many solutions does the equation (1) have for the function $F(x) = x^{2^n-2}$ over the field $\mathbb{F}_{2^n}$ with an arbitrary n?

Please, help Alice!

3.9. Problem "S-box masking" (8 scores)

To provide the security of a block cipher against the side channel attacks, some ideas of masking elements of the cipher are exploited. Here, we discuss masking S-boxes.

Alice takes a bijective function S (S-box) that maps n bits to n bits. Bob claims that, for every such a function S, there exist two bijective S-boxes, say S' and S'', mapping n bits to n bits in such a way that

$$S(x) = S'(x) \oplus S''(x) \text{ for all } x \in \mathbb{F}_2^n.$$

Hence, Alice is able to mask an arbitrary bijective S-box by "dividing it into parts" for realization. But Alice wants to see the proof of this fact. Please help Bob to give the arguments.

3.10. Problem "A special parameter" (10 scores)

In differential cryptanalysis of block ciphers, a special parameter P is used to measure the diffusion strength. In this problem, we study its properties.

Let n, m be positive integers. Let $a = (a_1, \ldots, a_m)$ be a vector with coordinates a_i taken from the finite field $\mathbb{F}_{2^n}$. Denote the number of nonzero coordinates a_i, $i = 1, \ldots, m$, by $\mathrm{wt}(a)$ and call this number the *weight* of a.

The sum of $a = (a_1, \ldots, a_m)$ and $b = (a_1, \ldots, b_m)$ in $\mathbb{F}_{2^n}^m$ is defined as $a + b = (a_1 + b_1, \ldots, a_m + b_m)$.

The *special parameter* P of a function $\varphi : \mathbb{F}_{2^n}^m \to \mathbb{F}_{2^n}^m$ is defined to be

$$P(\varphi) = \min_{a,\, b,\, a \neq b} \{\mathrm{wt}(a + b) + \mathrm{wt}(\varphi(a) + \varphi(b))\}.$$

- Rewrite (simplify) the definition of $P(\varphi)$ when the function φ is linear (recall that a function ℓ is linear if $\ell(x + y) = \ell(x) + \ell(y)$ for any x, y).
- Rewrite the definition of $P(\varphi)$ in terms of linear codes, when the linear transformation φ is given by a $m \times m$ matrix M over $\mathbb{F}_{2^n}$, i. e. $\varphi(x) = M \cdot x$.
- Find the least upper bound for $P(\varphi)$ as a function of m.
- Can you give an example of the function φ with the maximal possible value of P?

3.11. Problem "Add–Rotate–Xor" (10 scores)

Let $\mathbb{F}_2^n$ be the vector space of a dimension n over $\mathbb{F}_2 = \{0, 1\}$ and $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$. The vector x can be interpreted as the integer $x_1 \cdot 2^{n-1} + x_2 \cdot 2^{n-2} + \ldots + x_{n-1} \cdot 2 + x_n$.

Alice can produce hardware implementations for the following functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$:

1) $f_a(x) = x \boxplus a$ — the addition modulo 2^n of vectors x and a as integers for any fixed $a \in \mathbb{F}_2^n$;
2) $g_r(x) = x \lll r$ — the cyclic rotation of a vector x to the left by r positions for any fixed positive integer r, $0 < r < n$;
3) $h_b(x) = x \oplus b$ — the coordinate-wise sum modulo 2 of vectors x and b for any fixed $b \in \mathbb{F}_2^n$.

- Bob asks Alice to construct hardware implementations for the functions S_1 and S_2 from $\mathbb{F}_2^2$ to $\mathbb{F}_2^2$ given by their truth table (Table 4).

Table 4

x	00	01	10	11
$S_1(x)$	01	00	10	11
$S_2(x)$	(01	11	00	01

Can Alice do this? If "yes", show how it can be done; if "no", give an explanation!

- Generalizing the problem above, can we construct any function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$, using only a finite number of compositions of functions f_a, g_r and h_b?
  And what about any permutation over $\mathbb{F}_2^n$?
  Consider at last the cases $n = 2, 3, 4$.
- Is it possible to compute every function h_b, using only functions f_a and g_r?

3.12. Problem "Linear subspaces" (12 scores)

Recall several definitions and notions. Each element $x \in \mathbb{F}_2^k$ is a binary vector of length k, i. e. $x = (x_1, \ldots, x_k)$ and $x_1, \ldots, x_k \in \mathbb{F}_2$. For two vectors x and y in $\mathbb{F}_2^k$, their sum is $x \oplus y = (x_1 \oplus y_1, \ldots, x_k \oplus y_k)$ where $\oplus$ stands for XOR operation. Let $\mathbf{0}$ be the zero element of the vector space, i. e. the vector with all-zero coordinates. A nonempty subset $L \subseteq \mathbb{F}_2^k$ is called a *linear subspace* if, for any $x, y \in L$, $x \oplus y \in L$. It is easy to see that zero vector belongs to every linear subspace. A linear subspace L of $\mathbb{F}_2^k$ has a *dimension* n if it contains exactly 2^n elements.

Problem. For constructing a new secret sharing scheme, Mary has to solve the following task on binary vectors. Let n be an integer number, $n \geqslant 2$. Let $\mathbb{F}_2^{2n}$ be a $2n$-dimensional vector space over $\mathbb{F}_2 = \{0, 1\}$ that is the prime field of characteristic 2.

Do subsets $L_1, \ldots, L_{2^n+1}$ of $\mathbb{F}_2^{2n}$ satisfying the following conditions exist?

- L_i is a linear subspace of dimension n for every $i \in \{1, \ldots, 2^n + 1\}$;
- $L_i \cap L_j = \{\mathbf{0}\}$ for all $i, j \in \{1, \ldots, 2^n + 1\}$, $i \neq j$;
- $L_1 \cup \ldots \cup L_{2^n+1} = \mathbb{F}_2^{2n}$.

If "yes", show how to construct these subspaces for an arbitrary positive integer n.

For example, in the vector space $\mathbb{F}_2^4$, we can choose the following five required subspaces:

$$
\begin{aligned}
L_1 &= \{0000, 0001, 1110, 1111\};\\
L_2 &= \{0000, 0010, 1001, 1011\};\\
L_3 &= \{0000, 0011, 0100, 0111\};\\
L_4 &= \{0000, 0101, 1000, 1101\};\\
L_5 &= \{0000, 0110, 1010, 1100\}.
\end{aligned}
$$

3.13. Problem "Watermarking cipher" (unsolved)

The problem was stated by Gennady Agibalov.

Problem. Let X, Y, and K be the sets of plaintexts, ciphertexts, and keys respectively, $X = Y = \{0, 1\}^n$ and $K = \{0, 1\}^m$ for some integers n and m. Recall that two functions $E : X \times K \to Y$ and $D : Y \times K \to X$ are called *an encryption algorithm* and *a decryption algorithm* respectively if, for any $x \in X$, $k \in K$, the equallity $D(E(x, k), k) = x$ takes place. Together E and D form *a cipher*.

Let us call a cipher *watermarking* if for any key $k \in K$ and any subset $I \subseteq \{1, 2, \ldots, n\}$, there exists a key k_I such that, for any $x \in X$,

$$D(E(x, k), k_I) = x'$$

where x' differs from x in all bits with coordinates from I.

An example of a trivial watermarking cipher. Let $m = n$ and encryption and decryption algorithms be the following: $E(x, k) = x \oplus k$ and $D(y, k_I) = y \oplus k_I$ where k_I is obtained from k by changing all bits with coordinates from I. The main disadvantage of such a cipher is that the every key should be used only once.

How can we use a watermarking cipher? Suppose you own some digital product (for example, video) that you want to sell. Let x represent a binary code of the product.

For each customer of x, you choose an unique set I of bit coordinates in x, encrypt the plaintext x, using a predetermined key k, and send the resulting ciphertext y and the corresponding key k_I to him. Then after receiving y and k_I, the customer decrypts y and gets x'. The difference between the original x and x' is not significant; thus the customer does not know about it. If someone illegally spreads the product x' bought from you, you can easily identify him by the set I.

Summarize the ideas we need to construct a cipher that has to put something like a “watermark” into a video. Let’s try! So the problem is to construct a non-trivial watermarking cipher. Please, think about the easy usage of it by an owner and a customer.

3.14. Problem “APN permutation” (unsolved)

Suppose we have a mapping F from $\mathbb{F}_2^n$ to itself (recall that $\mathbb{F}_2^n$ is the vector space of all binary vectors of length n). This mapping is called a *vector Boolean function in n variables.* Such functions are used, for example, as S-boxes in block ciphers and should have special cryptographic properties. In this problem, we consider the following two properties and the problem of combining them.

- A function F in n variables is a *permutation* if, for all distinct vectors $x, y \in \mathbb{F}_2^n$, it has distinct images, i. e. $F(x) \neq F(y)$.
- A function F in n variables is called *Almost Perfect Nonlinear* (APN) if, for any nonzero vector $a \in \mathbb{F}_2^n$ and any vector $b \in \mathbb{F}_2^n$, the equation $F(x) \oplus F(x \oplus a) = b$ has at most 2 solutions. Here, $\oplus$ is the coordinate-wise sum modulo 2 of vectors.

Try to find an APN permutation in 8 variables or prove that it doesn’t exist.

History of the problem. The question “Does an APN permutation in even number of variables exist?” has been studied for more than 20 years. If the number of variables is odd, APN permutations exist as it was proved by K. Nyberg in 1994 [1]. It is known that, for 2 and 4 variables, the answer is “No”. But for 6 variables, K. Browning, J. F. Dillon, M. McQuistan, and A. J. Wolfe have found such a function in 2009 [2]. You can see it bellow:

$$G = (\begin{array}{cccccccccccccccc} 0 & 54 & 48 & 13 & 15 & 18 & 53 & 35 & 25 & 63 & 45 & 52 & 3 & 20 & 41 & 33 \\ 59 & 36 & 2 & 34 & 10 & 8 & 57 & 37 & 60 & 19 & 42 & 14 & 50 & 26 & 58 & 24 \\ 39 & 27 & 21 & 17 & 16 & 29 & 1 & 62 & 47 & 40 & 51 & 56 & 7 & 43 & 44 & 38 \\ 31 & 11 & 4 & 28 & 61 & 46 & 5 & 49 & 9 & 6 & 23 & 32 & 30 & 12 & 55 & 22 \end{array}).$$

This function is presented as the list of its values, i. e. $G(0) = 0$, $G(4) = 15$, $G(16) = 59$ and so on. For brevity, we use integers instead of binary vectors. A binary vector $x = (x_1, \ldots, x_n)$ corresponds to the integer $k_x = x_1 \cdot 2^{n-1} + x_2 \cdot 2^{n-2} + \ldots + x_{n-1} \cdot 2 + x_n$.

Thus you are welcome to study the next case, $n = 8$.

3.15. Problem “Super S-box” (unsolved)

Another unsolved problem is directly related to AES construction. J. Daemen and V. Rijmen, the designers of AES (Rijndael), have introduced the Super-Sbox representation of two rounds of AES in order to study differential properties [3]. The function G used in the problem can be considered as a simplified Super-Sbox model of two rounds of AES. To study security of AES against the differential cryptanalysis, we welcome you to start with the differential characteristics of the function G.

Problem. Let $\mathbb{F}_{256}$ be the finite field of 256 elements and α be a primitive element (it means that, for any nonzero $x \in \mathbb{F}_{256}$, there exists $i \in \mathbb{N}$ such that $x = \alpha^i$). Let $\mathbb{F}_{256}^4$ be the vector space of dimension 4 over $\mathbb{F}_{256}$. Thus any element $x \in \mathbb{F}_{256}^4$ is $x = (x_1, x_2, x_3, x_4)$ where $x_i \in \mathbb{F}_{256}$. An arbitrary function from $\mathbb{F}_{256}^4$ to $\mathbb{F}_{256}^4$ can be considered as the set of 4 coordinate functions from $\mathbb{F}_{256}^4$ to $\mathbb{F}_{256}$. Introduce the following auxiliary functions $F_4, M : \mathbb{F}_{256}^4 \to \mathbb{F}_{256}^4$:

$$F_4(x_1, x_2, x_3, x_4) = (x_1^{254}, x_2^{254}, x_3^{254}, x_4^{254});$$

$$M(x_1, x_2, x_3, x_4) = (x_1, x_2, x_3, x_4) \times \begin{bmatrix} \alpha+1 & 1 & 1 & \alpha \\ \alpha & \alpha+1 & 1 & 1 \\ 1 & \alpha & \alpha+1 & 1 \\ 1 & 1 & \alpha & \alpha+1 \end{bmatrix}.$$

Define the function $G : \mathbb{F}_{256}^4 \to \mathbb{F}_{256}^4$ by the following combination of F_4 and M:

$$G(x_1, x_2, x_3, x_4) = F_4(M(F_4(x_1, x_2, x_3, x_4))).$$

Find the number of solutions of the equation $G(x + a) = G(x) + b$ with the parameters a and b running all nonzero values from $\mathbb{F}_{256}^4$.

4. Solutions of the problems

Here, we would like to discuss solutions of the problems. Our special attention is payed to (right/wrong and beautiful) solutions given by the participants.

4.1. Problem "A hidden message" (4 scores)

Solution. The famous Bacon's cipher was used here. First of all, everyone can get from the text that the alphabet of the message consists of 32 symbols (26 English letters and 6 punctuation marks) and notice that there are upper and lower case letters in the text. Such observations suggest that a binary code was used and a letter case means either 0 or 1. Since the cardinality of the alphabet is $32 = 2^5$, a string of five 0s and 1s is needed to code each letter of a secrete message. Thus if we delete all spaces and punctuation marks and divide the sequence obtained into words of 5 letters, we get the following text:

CrYPt	ogRap	HYiSa	ScIEn	ceOfs	eCrET	wriTi	nGFOr	aTLea	stTwo
THoUs	ANdye	aRSTh	ErEha	VEbEe	NpeOP	lEWHo	WAnTe	DtoSE	ndMES
saGes	WHiCh	coUlD	oNlyb	EenrE	AdbYt	HepEO	PLeFo	RwhOm	tHeyw
ereiN	teNde	DaloT	oFdif	feren	tMEtH	ODsFo	RcoNc	EalIN	GmEss
ageSW	erEin	vENtE	DstAR	TINGW	IthAn	CIeNt	cIPHe	rSlIK	ESkyt
aLEan	dATBA	sHand	endin	gwiTH	MOdEr	nSymm	eTRiC	ANdPu	bliCk
EyenC	RYpti	oNALG	OriTH	mSSUc	haSAe	SandR	sathe	dEVEL	opMEN
TOfcr	YPtOg	RaPHy	cOnti	NueSA	ndNEV	ERsTo	pSdec	rYPtT	HemES
saGet	HatiS	hIDdE	ninth	EteXT	oFthi	sTASk	tHEaL	phabe	tFoRT
HEmEs	sAGEC	onsis	TsofA	LltWE	nTysi	xenGl	iShle	tTERS	froma
TozAN	dSixp	uNCTu	aTIoN	MARkS					

If the upper case is coded by 0 and the lower case by 1, we get the strange beginning of the text: "`J,FJ,`" and so on. Hence, we should use 0 and 1 for the lower and upper cases respectively. In this way, you get "`We welcome you to the first Siberian student's Olympiad in cryptography with international participation!`". We supposed that the letters are coded in alphabetic order with the numbers from 0 to 25, and the punctuation marks with the numbers from 26 to 31.

This problem was completely solved by 34 participants.

Some ideas and typical mistakes of participants are listed below:

• the most part of those participants who sent their answers for this problem succeeded in its solving; as a rule, they wrote a program in order to decode the message more faster;

• some participants mentioned in their solutions that Bacon‘s cipher was used;

• a few participants used the standard decoder BASE32 for decoding the binary string, but they did not substituted the punctuation marks from the alphabet of this problem;

• the most surprising comment for this problem was “Mendeleyev’s periodic table was somehow used there!”.

4.2. Problem “A crypto room” (4 scores)

The answer to the question “Who is the author?” is Anton Pavlovich Chekhov, a Russian classical writer. The encrypted message hides the name of his famous play “Three sisters”.

Solution. Let us consider the steps of solution. The information contained in the picture allows to find the key needed for decryption. At first, we reflect from right to left the text written on the blackboard. It says “Use all information you can find here to get the key. The first part of it is `AC`”. It becomes clear that the key starts with `AC`. “The next part is `IWP`”, as it is written below in the picture. In the upper right corner, there is the sentence in which each word starts with the last letter of the original correct word. The result of the letters permutation to the initial positions is the sentence “Then use letters on the photos from up to down”. Let us do it. We get the third part of the key `RRVM`. The final part of the key is `JOV` because of “`JOV` completes the key”.

As a result, we obtain the key `ACIWPRRVMJOV`. Observe that its length is equal to 12 letters, that is exactly equal to the length of the encrypted message.

The Vigenere cipher was used for encryption. Each letter has its position number in the alphabet: `A` — 1, `B` — 2, ..., `Z` — 26, or 0. Using the example of ciphering above the portrait of Julius Caesar, we determine the rule of encryption:

$$\mathtt{A} + \mathtt{D} = \mathtt{E};\ \text{that is } 1 + 4 = 5 \pmod{26};$$
$$\mathtt{X} + \mathtt{D} = \mathtt{B};\ \text{that is } 24 + 4 = 2 \pmod{26}.$$

To decrypt the ciphertext, you need to subtract the key from the ciphertext according to the same rule. The decryption of two first letters of the message is given here:

$$\mathtt{U} - \mathtt{A} = \mathtt{T} \text{ since } 21 - 1 = 20 \pmod{26};$$
$$\mathtt{K} - \mathtt{C} = \mathtt{H} \text{ since } 11 - 3 = 8 \pmod{26}.$$

Thus we obtain the secret message `THREESISTERS`. This problem was completely solved by one pupil and 30 students. And only one participant has read the last strange lettering in the picture: it was a word “algorithm” written in Malayalam.

This problem was completely solved by 33 participants. The most detailed solution, illustrated with the step-by-step pictures, was proposed by O. Smirnov (Saratov State University).

Typical mistakes were:

• the inclusion of symbols `5` and `*` into the third part of the key (actually they are not letters);

• using the Caesar cipher with the key being equal to 4, although it was only a helpful (or unhelpful) example.

4.3. Problem “The musical notation” (4 scores)

Solution. This is a classical permutation cipher. At first, according to the ciphertext and the used musical notation, we should form the rows corresponding to all different notes “do”, “re”, ..., “ti” (Table 5).

Table 5

do								E		E							V									N
re		O		E			S													B		E			O	
mi	R		L		L						E			T				C	P		D		F			
fa																										
sol															O											
la									O			E														
ti						I							H			M								S		

Our next task is to find the right cyclical shifts for all the rows to form a readable message. The number of all possible variants is equal to $9 \cdot 6 \cdot 4 \cdot 4 \cdot 2 \cdot 1 = 1728$. Fortunately, we do not need to check all these variants because if we start with two the biggest rows ("mi" and "re"), we find that the beginning and the end of the plaintext could be "`LBEET*O`" and "`DFOREL*S*`" respectively. To find the rest, we cyclically shift other rows. Thus we get the secrete message: `L. Beethoven composed` "`For Elise`".

This problem was presented in both rounds and was solved by 33 participants. It is interesting to note some creative ways for solving. The extraordinary majority of solutions were programming. The participants generated all possible variants of row shifts and then filtered them according to some rules:

- tests for invalid combination of three or four consecutive alphabetic letters. Some participants used the known such combinations, but one participant, D. Zajcev (Saratov State University), generated invalid combinations by himself using the novel "War and peace" by Leo Tolstoy in English. The team of P. Hvoryh and V. Laptev (the winners of the second round of NSUCRYPTO in category "students") used a formula for naturalness of the language;
- one participant (S. Shabelnikov from Saratov State University) wrote a program that puts the next variant into the Internet for a search and then considered the number of results obtained.

There were also solutions made by hand. The list of some ideas is here:

- as well as in the solution presented above, some participants began with determining possible shifts for the biggest rows "mi" and "re". For example, the team of pupils S. Derevyanchenko and E. Klochkova (Specialized Educational Scientific Center of NSU) did this in details;
- we are pleased to tell that many participants have recognized the musical notation and its composer L. V. Beethoven; they tried to find shifts in order to form any of words "Beethoven", "Elise", and "composed". So the team of V. Marchuk, D. Emelyanov, and A. Gusakova (Belarusian State University) mentioned that the mistake of Alice was in taking the famous musical fragment.

4.4. Problem "Boolean cubes" (4 scores)

Solution. In the first cube, let us assign vertices $(0,0,0), (1,0,0), (0,1,0), (0,0,1)$ with "+" and vertices $(1,1,1), (1,1,0), (0,1,1), (1,0,1)$ with "−". If we add all the coordinates of vertices in both subsets "+" and "−" taken with the plus and minus signs respectively, we obtain the following results: for the subset "+", we have $0+1+1+1=3$ (since there is one zero vector and three vectors each with only one nonzero coordinate), for the subset "−", we have $-3-2-2-2=-9$, and the total sum is -6. Consider the operation A. We can see that A adds 1 or -1 in both sums of subsets "+" and "−" simultaneously. This fact means that the total sum is invariant under the multiple applying the operation A and is equal to -6. If we repeat the assignment of vertices in matching layers of the second cube, we will see that the total sum is equal to 0. Thus we can not get the cube E_2 from the cube E_1 by applying the operation A any number of times.

In the first round, two pupils solved this problem with maximal scores, both used the idea of the sum's invariant property. The solution of N. Dobronravov (Lyceum 130, Novosibirsk) was very compact and logical.

This problem was also included in the second round and was properly solved by 18 teams. We want to outline some interesting non-trivial ideas of solutions. The team of G. Beloshapko, A. Taranenko, and E. Fomenko (Novosibirsk State University) introduced the following value for a cube C:

$$\lambda(C) = \left| \sum_{v \in C} (-1)^{|x(v)|} \mathrm{wt}(v) \right|$$

where $\mathrm{wt}(v)$ is the weight of vector v, $x(v)$ is the number of edges between vector v and null vector. They proved that this value does not depend on which vertex is considered to be the zero vertex. After this proof, the team underlined that $\lambda(C)$ is invariant under operation A and since $\lambda(E_1) = 6$ and $\lambda(E_2) = 0$, we can not obtain E_2 from E_1.

The most nontrivial and clear solution belongs to the team of A. Udovenko (Saint Petersburg). The participant divided vertices into 4 subsets with no edges inside each subset and with all edges between neighboring subsets:

1. (0,0,0); 2. (1,1,0), (1,0,1), (0,1,1); 3. (1,0,0), (0,1,0), (0,0,1); 4. (1,1,1).

Any application of the operation A adds or subtracts 1 from sums of two neighboring subsets. We can write this operations as vectors: $(1, 1, 0, 0)$, $(0, 1, 1, 0)$, $(0, 0, 1, 1)$. Then the participant proved that none of the vectors of differences between sums of the subsets for all possible positions of (0,0,0) in E_2 can be obtained as a linear combination of operation's vectors. So it is impossible to get the cube E_2 from E_1, using operation A.

The team of S. Skresanov, A. Miloserdov, and D. Kirin (Novosibirsk State University) found a short and nice solution via colorings of the vertices. O. Smirnov, P. Razumovsky, and A. Ripinen (Saratov State University) transformed the problem to the task about two special segments and showed that one cannot be obtained from the other.

4.5. Problem "A broken cipher machine" (4 scores)

Solution. It is well known that any positive integer n and the sum S_n of its digits are congruent modulo 3, i. e. $n = S_n (\bmod 3)$. It follows from the algorithm that $m = 0 (\bmod 3)$, since $n = n' (\bmod 3)$ and $m \bmod 3 = n' + 2S_n \bmod 3 = n + 2n \bmod 3 = 0$.

Since the result got by Mary was a power of two, we conclude that she obtained an incorrect encrypted message.

This problem was completely solved by 6 pupils. The five of them have sent solutions that are very close to the presented one. For example, such a solution was proposed by the youngest participant A. Dorokhin (Novosibirsk school 159) — the winner of the first round of NSUCRYPTO in category "senior pupil". One participant solved the problem through the implicit proof of the property of congruence between an integer and its sum of digits.

4.6. Problem "The snowflake cipher" (4 scores)

Solution. We may see that the problem is equivalent to finding the longest path through triangles with common edge such that every triangle can be found in this path only once. Let us consider nine *ledges* each consisting of three triangles with only one common edge with the rest of snowflake. Clearly, if some of these ledges is crossed by our path, then it can be only the beginning or the end of the path. So the path may contain not more than 4 triangles from 2 chosen ledges.

Let us paint triangles in the rest part of snowflake in black and white colours such that neighbouring triangles have different colours. Among them, there are 93 black triangles and 114 white, but since colours of triangles in the path should alternate, not more than 93 black triangles and 94 white triangles can be in the longest path. So we can say that length of a path in the snowflake is not more than $93 + 94 + 4 = 191$. We present an example (Fig. 7) of such a path constructed by the team of G. Beloshapko, A. Taranenko, and E. Fomenko (Novosibirsk State University).

Unfortunately, the pupils did not cope with this problem on the highest score, nevertheless, three of them got some scores due to the right but incomplete ideas of estimation.

In the second round, 6 teams have solved this problem with full possible scores.

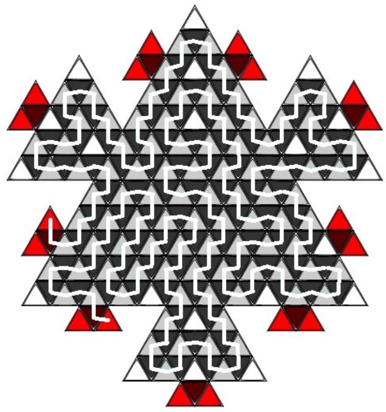


Fig. 7.

Let us consider an interesting solution proposed by A. Udovenko (Saint Petersburg).

• At the first stage, two kinds of ledges were introduced, the first one, P1, is the same as in the solution above, consisting of three triangles, but there are only 6 such ledges in the group, because the rest three of them are supposed to be the part of the second type bigger ledges, P2. We may see that ledges of the type P2 consist of 24 triangles. Define the P3 as the main big triangle, obtained by deleting all P1 and P2 ledges from the original snowflake.

• At the second stage, the author also painted P3 in black and white so that any moving between triangles is possible only if they have different colours. Then he got an estimate that the longest path in P3 will skip at least 11 triangles.

• At third, the author considered all possible pairs "start-end" and, for each pair, counted minimal number of skipped triangles. He obtained that the lower bound on this number is 43 and gave a right example of path with 191 length.

The team of V. Marchuk, D. Emelyanov, and A. Gusakova (Belarusian State University) proposed very nice solution by transforming the problem to the following one: find a Hamilton cycle in a special graph corresponding to the snowflake.

4.7. Problem "A pseudo-random generator" (6 scores)

Solution. It is a simple task. First of all, the binary sequence from r_3 and r_4 is `11000000011001000100111100000111`. Because the length of the generator is 16, linear complexity of the sequence is not more than 16. Next, we have the sequence of 32 consecutive bits and can uniquely restore its recurrent relation using, for example, the Berlekamp — Massey algorithm. This relation is

$$u_{i+16} = u_{i+5} \oplus u_{i+3} \oplus u_{i+1} \oplus u_i.$$

Since we know the recurrent relation, we obtain: IV is 58 390. We also know that the generator was initialized on November 17, 2014, at 12:05 UTC+6. Therefore, IV would be 58 476. In Russia, there are time zones from UTC+2 to UTC+12. So Bob initialized the generator $58\,476 - 58\,390 = 86$ seconds to November 17, 2014, 12:05 UTC+6, i. e. at 12:03:34. Consequently, both Bob and Eve live in Novosibirsk time.

There were 18 solutions from the teams and only one solution was wrong, the most of teams just solved a system of linear equations and have not used the Berlekamp — Massey algorithm.

4.8. Problem "Number of solutions" (8 scores)

Solution. Consider a general case, i. e. $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$, $F(x) = x^{2^n-2}$ (and $F(x) = x$ for $n = 1$). We need to determine how many solutions the equation $F(x + a) = F(x) + b$ may have for all different pairs of nonzero parameters a and b where $a, b \in \mathbb{F}_{2^n}$. Since characteristic of $\mathbb{F}_{2^n}$ is 2, the operations "$-$" and "$+$" coincide.

First of all, note that, for $n = 1$, there exists only one pair $(a, b) = (1, 1)$. In this case, there are 2 solutions of the equation. In what follows let $n > 1$.

Note that the function $F_a(x) = F(x)+F(x+a)$ has some symmetry: $F_a(x) = F_a(x+a)$. This means that 2 divides the number of solutions and, at least for 2^{n-1} distinct $b \in \mathbb{F}_{2^n}$, $F_a(x) \neq b$ for all $x \in \mathbb{F}_{2^n}$.

Therefore, for all $n > 1$, there exist $b \neq 0$ and $a \neq 0$ such that the equation has no solutions. That is why the number of solutions of the equation $F(x + a) = F(x) + b$ can be 0. Consider other possibilities.

Simplify the given equation. For this, note that $y^{2^n-1} = 1$ for all $y \in \mathbb{F}^*_{2^n}$. Then after multiplying the equation $(x + a)^{2^n-2} + x^{2^n-2} = b$ by $x(x + a)$, we get the equation $bx^2 + abx + a = 0$. After multiplying the last by $a^{-2}b^{-1}$, we get $x^2/a^2 + x/a + (ab)^{-1} = 0$. Note that the number of solutions of the equation $x^2/a^2 + x/a + (ab)^{-1} = 0$ only depends on ab, and the values $x = 0$, $x = a$ are not its solutions. Rewrite this equation as

$$z^2 + z + (ab)^{-1} = 0$$

where $z = x/a$. Then we have two cases:

- $x = 0$ and $x = a$ are solutions of $F(x) + F(x + a) = b$. Then $a = b^{-1}$ and $ab = 1$. We already have 2 solutions. Next, solve the equality $z^2 + z + 1 = 0$. Both roots of the polynomial $z^2 + z + 1$ belong to the field $\mathbb{F}_{2^2}$. This field is contained in $\mathbb{F}_{2^n}$ if and only if n is even. So in the case of even n, the equation $F(x) + F(x + a) = b$ has exactly 4 solutions. In the case of odd n, there are exactly 2 solutions of the equation $F(x) + F(x + a) = b$.
- $x = 0$ and $x = a$ are not solutions of $F(x)+F(x+a) = b$. Therefore, $ab \notin \mathbb{F}_2$. Note that the equation $z^2 + z = z(z + 1) = (ab)^{-1}$ has 0 or 2 solutions. Since $(ab)^{-1}$ can be an arbitrary element from $\mathbb{F}_{2^n} \setminus \mathbb{F}_2$, at least for $2^{n-1} - 2$ distinct ab, there are exactly two solutions. So if $n > 2$, then 2 solutions can be in this case. If $n = 2$, then $z^2 + z \in \mathbb{F}_2$, i. e. for $ab \notin \mathbb{F}_2$, there is no solution.

The answer is the following. For $n = 1$, there are always 2 solutions; for $n = 2$, there can be 0 or 4 solutions; for odd n ($n > 1$), there can be 0 or 2 solutions; for even n ($n > 2$), there can be 0, 2 or 4 solutions.

This problem was completely solved during the second round. The teams of P. Hvoryh and V. Laptev (Omsk State Technical University), A. Udovenko (Saint Petersburg), A. Oblaukhov (Novosibirsk State University), S. Belov and G. Sedov (Moscow State University) proposed the right and complete solutions.

4.9. Problem "S-box masking" (8 scores)

Solution. We would like to give the solution proposed by L. Qu et al. in the first variant of the paper [4]. Let us represent an arbitrary bijective function $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ by the sum $S'(x) \oplus S''(x)$ where $S', S'' : \mathbb{F}_2^n \to \mathbb{F}_2^n$ are bijective too. Consider $\mathbb{F}_2^n$ as $\mathbb{F}_{2^n}$. Let $\alpha \in \mathbb{F}_{2^n}$ and $\alpha \neq 0, 1$. If $n > 1$, such an element α does exist. It is clear that $S(x) = \alpha S(x)+(\alpha+1)S(x)$. Note that $\alpha S(x)$ and $(\alpha + 1)S(x)$ are bijective, since each of them is a composition of two bijective mappings. So for $n > 1$, the required representation exists. If $n = 1$, there are only two bijective functions $S(x) = x$ and $S(x) = x \oplus 1$; their sum is a constant and hence there is no the required representation in this case.

There were two complete combinatorial solutions proposed by S. Godzhaev (Moscow State University) and A. Udovenko (Saint Petersburg).

4.10. Problem "A special parameter" (10 scores)

A special parameter that we consider in this problem is called the *differential branch number* of a transformation, see for example book [3]. In differential cryptanalysis of block ciphers, this parameter is used to measure the diffusion strength of a cipher. Some properties of it are discussed in the problem.

Solution. Let $\varphi : \mathbb{F}_{2^n}^m \to \mathbb{F}_{2^n}^m$ be a linear function and $a, b \in \mathbb{F}_{2^n}^m$.

• Since φ is linear, i. e. $\varphi(x+y) = \varphi(x) + \varphi(y)$ for all $x, y \in \mathbb{F}_{2^n}^m$, and the condition $a \neq b$ is equivalent to $a + b \neq 0$, we can rewrite the definition of $P(\varphi)$ in the following way:

$$P(\varphi) = \min_{a+b\neq 0} \{\mathrm{wt}(a+b) + \mathrm{wt}(\varphi(a+b))\}.$$

• Let us consider vectors $(x, \varphi(x)) = (x, M \cdot x)$ of length $2m$ where $x \in \mathbb{F}_{2^n}^m$. Then the set $C = \{(x, M \cdot x) \mid x \in \mathbb{F}_{2^n}^m\}$ is a linear code. Since we have $\mathrm{wt}(a+b) + \mathrm{wt}(\varphi(a+b)) = $ $= \mathrm{wt}((a+b, \varphi(a)+\varphi(b))) = \mathrm{dist}((a, \varphi(a)), (b, \varphi(b)))$, where $\mathrm{dist}(x, y)$ means the Hamming distance between vectors x and y, the parameter $P(\varphi)$ is equal to the minimal distance between distinct codewords of the code C.

• Since $a \neq b$, the minimal value of $\mathrm{wt}(a+b)$ is equal to 1. The maximal possible value of $\mathrm{wt}(\varphi(a)+\varphi(b))$ is m by the definition. Thus the maximal possible value of $P(\varphi)$ is not more than $m+1$.

• One can construct an example of such a transformation using a Maximal Distance Separable code with parameters $[2m, m, m+1]$ (for example, Reed — Solomon code).

This problem was completely solved by two teams: P. Hvoryh and V. Laptev (Omsk State Technical University); K. Kogos and S. Kyazhin (Moscow Engineering Physics Institute).

4.11. Problem "Add–Rotate–Xor" (10 scores)

The complicated algebraic solution of this problem was given by T. Zieschang in 1997 [5]. Here, we introduce a simple solution proposed by the participants of the Olympiad.

Solution.

- $S_1(x) = g_1(f_1(g_1(f_2(x))))$. It is obvious that the functions f_a, g_r, and h_b are all bijective. Therefore, any its composition is bijective too. The function S_2 is not bijective, so it can not be represented as a composition of them.
- Only permutations on $\mathbb{F}_{2^n}$ can be constructed in this way. It is well known that the compositions of the function f_1 (a cycle of length 2^n) and the transpositions of adjacent elements in the cycle give us all the permutations on $\mathbb{F}_{2^n}$. The following construction gives a certain transposition τ: $\tau(x) = g_1(f_{2^n-1}(g_{n-1}(f_2(x))))$. Indeed, $g_{n-1}(y) = 2^{n-1}y_n + \lfloor y/2 \rfloor$ and if $x < 2^n - 2$, then $f_2(x) = x + 2$, so

$$g_{n-1}(f_2(x)) = g_{n-1}(x+2) = 2^{n-1}(x+2)_n + \lfloor (x+2)/2 \rfloor =$$
$$= 2^{n-1}x_n + \lfloor x/2 \rfloor + 1 = (x_n, x_1, \ldots, x_{n-1}) + 1.$$

  Next, f_{2^n-1} eliminates "+1" and g_1 cyclically rotates $(x_n, x_1, \ldots, x_{n-1})$ to the left by one position. That is, $\tau(2^n - 1) = 2^n - 2$. Also, $\tau(2^n - 2) = 2^n - 1$ because $f_2(2^n - 2) = 0$, $g_{n-1}(0) = 0$, $f_{2^n-1}(0) = 2^n - 1$, $g_1(2^n - 1) = 2^n - 1$. Therefore, τ transposes $2^n - 1$ with $2^n - 2$ and does not change all other elements.

- Yes. The third item is obvious, since we can construct any permutation on $\mathbb{F}_{2^n}$ using the mentioned functions.

In the second round, there were 7 right solutions. The most of teams have found a full cycle and a transposition. The solution given above is based on the solution that was presented by the team of G. Beloshapko, A. Taranenko, and E. Fomenko (Novosibirsk State University). Very clear and simple solution was proposed by the team of R. Zhang and A. Luykx (KU Leuven, Belgium); they constructed all transpositions in an explicit way.

4.12. Problem "Linear subspaces" (12 scores)

Here, the solution given by the program committee is presented.

Solution. Consider $\mathbb{F}_2^{2n}$ as 2-dimensional vector space V over Galois field $\mathbb{F}_{2^n}$ consisting of 2^n elements. Define the following family of sets:

$$L_\alpha = \{(x, \alpha x) : x \in \mathbb{F}_{2^n}\},\ \alpha \in \mathbb{F}_{2^n};\ \ L_{2^n+1} = \{(0, y) : y \in \mathbb{F}_{2^n}\}.$$

It is obvious that every such a set is a linear subspace in V and contains exactly 2^n elements. Let us show that an arbitrary element $(x, y) \in V$ is covered by the union of these subspaces. If $x = 0$, it is covered by L_{2^n+1}. Otherwise, $(x, y) = (x, (y/x)x)$ belongs to the subspace $L_{y/x}$. Note that every two subspaces have only one common element, $\mathbf{0}$, since the cardinality of V is exactly $2^{2n} = (2^n + 1)(2^n - 1) + 1$. Thus, the answer for the problem is "yes" and the system is constructed.

Another approach (but still using Galois fields) was proposed by the team of P. Hvoryh and V. Laptev (Omsk State Technical University) during the second round. They constructed linear subspaces in the following way:

$$L_0 = \{0, \alpha^{0\cdot(2^n+1)}, \alpha^{1\cdot(2^n+1)}, \ldots, \alpha^{(2^n-2)(2^n+1)}\} \ \text{ and } \ L_i = \{\alpha^i x : x \in L_0\},\ i = 1, \ldots, 2^n,$$

where α is a generating element of $\mathbb{F}^*_{2^{2n}}$.

There was also a nice right solution obtained by the team of S. Skresanov, A. Miloserdov, and D. Kirin (Novosibirsk State University). This is more close to the solution above.

Some teams proposed algorithms for constructing subspaces, but they made mistakes in their constructions.

4.13. Problem "Watermarking cipher" (unsolved)

The most deep analysis of this problem was proposed by R. Zhang and A. Luykx (winners of the second round of NSUCRYPTO in category "professional"), but nobody has introduced a concrete solution. May be you can do it?

Some details on this problem and some non-trivial watermarking ciphers based on symmetric stream ciphers with functional keys are considered in the talk of G. P. Agibalov on the conference Sibecrypt'15 and published in [6].

4.14. Problem "APN permutation" (unsolved)

This is another unsolved problem of the Olympiad; it is the well known long standing problem of cryptography. Some ideas on it were proposed by the team of G. Beloshapko, A. Taranenko, and E. Fomenko (winners of the second round of NSUCRYPTO in category "students"). They proved some basic properties of APN functions; namely, if a permutation is an APN function, then its inverse function is APN too.

One participant, D. Svitov from NSU, has proposed an online service for distributed search of APN permutations [7]. But till now this problem is unsolved.

4.15. Problem "Super S-box" (unsolved)

The problem is still unsolved. Only one team of G. Beloshapko, A. Taranenko, and E. Fomenko from Novosibirsk State University has sent a solution with an analysis of the problem for smaller field. They considered $\mathbb{F}_{16}$ and found the exact number of pairs a, b for each number of solutions. It seems that any even number between 0 and 44 can be the number of solutions. They proposed a hypothesis that the same result is true for $\mathbb{F}_{256}$: it may be any even number of solutions bounded by some number.

5. Awarding

Awarding of the winners was held on December 2014 in Novosibirsk State University (Fig. 8).

Fig. 8.

6. Winners of the Olympiad-2014

In this section, we publish the names of winners of NSUCRYPTO-2014 and some information about them. There are 15 winners in the first round and 11 teams in the second one (Tables 6–11).

Table 6

Winners of the first round in school section ("senior pupils")

Place	Name	Country	City	School	Class	Scores
1	Alexander Dorokhin	Russia	Novosibirsk	MOU 159	8	12
2	Nikita Dobronravov	Russia	Novosibirsk	Lyceum 130	9	10
3	Artem Uskov	Russia	Novosibirsk	Gymnasium 3	11	8
3	Egor Dobronravov	Russia	Novosibirsk	Lyceum 130	9	8

Table 7

Winners of the first round in student's section (in category "students")

Place	Name	City	University	Department	Course	Scores
1	George Beloshapko	Novosibirsk, Russia	Novosibirsk State University	Mechanics and Mathematics	6	23
2	Roman Lebedev	Novosibirsk, Russia	Novosibirsk State University	Physics	2	14
3	Dmitry Zajcev	Saratov, Russia	Saratov State University	Computer Science and Information Technology	5	12
3	Gleb Shalyganov	Saratov, Russia	Saratov State University	Computer Science and Information Technology	4	12
3	Samir Godzhaev	Moscow, Russia	Moscow State University	Mechanics and Mathematics	1	12
3	Alexander Shein	Saratov, Russia	Saratov State University	Computer Sciences and Information Technologies	5	12
3	Pavel Grachev	Saratov, Russia	Saratov State University	Computer Sciences and Information Technologies	5	12
3	Sergey Shabelnikov	Saratov, Russia	Saratov State University	Computer Sciences and Information Technologies	5	12
3	Angelina Sadohina	Saratov, Russia	Saratov State University	Computer Sciences and Information Technologies	4	12
3	Alexander Tkachev	Novosibirsk, Russia	Novosibirsk State University	Information Technologies	2	12

Table 8

Winners of the first round in student's section (in category "professionals")

Place	Name	Country	City	Organization	Scores
1	Alexey Udovenko	Russia	Saint-Petersburg	—	12

Table 9

Winners of the second round (in category "senior pupils")

Place	Name	Country	City	School	Class	Scores
1	Stepan Derevyanchenko, Elizaveta Klochkova	Russia	Novosibirsk	Specialized Educational Scientific Center of NSU	11	6

Table 10

Winners of the second round (in category "students")

Place	Names	City	University	Department	Course	Scores
1	George Beloshapko, Anna Taranenko, Evarist Fomenko	Novosibirsk, Russia	Novosibirsk State University	Mechanics and Mathematics	5-6	55
1	Pavel Hvoryh, Vladimir Laptev	Omsk, Russia	Omsk State Technical University	Information Technologies and Computer Systems, RTF	3	55
2	Saveliy Skresanov, Alexey Miloserdov, Denis Kirin	Novosibirsk, Russia	Novosibirsk State University	Mechanics and Mathematics	2	36
3	Alexey Oblaukhov	Novosibirsk, Russia	Novosibirsk State University	Mechanics and Mathematics	3	33
3	Vadim Marchuk, Dmitry Emelyanov, Anna Gusakova	Minsk, Belarus	Belarusian State University	Applied Mathematics and Computer Science	6	33
3	Oleg Smirnov, Peter Razumovsky, Alexey Ripinen	Saratov, Russia	Saratov State University	Computer Science and Information Technology	4	29
3	Sergey Belov, Grigory Sedov	Obninsk, Moscow, Russia	Moscow State University	Computational Mathematics and Cybernetics	5	28

Table 11

Winners of the second round (in category "professional")

Place	Names	Country	City	Organization	Scores
1	Ren Zhang, Atul Luykx	Belgium	Leuven	KU Leuven, COSIC	65
2	Konstantin Kogos, Sergey Kyazhin	Russia	Moscow	Moscow Engineering Physics Institute	41
3	Alexey Udovenko	Russia	Saint-Petersburg	—	37

Acknowledgements

We thank Novosibirsk State University for the financial support of the Olympiad and invite you to take part in the next NSUCRYPTO that starts on November 15, 2015. Your ideas on the mentioned unsolved problems are also very welcome and can be sent to `olymp@nsucrypto.ru`.

We are very grateful to Gennadiy Agibalov and Irina Pankratova for their valuable contribution to this paper.

REFERENCES

1. *Nyberg K.* Differentially uniform mappings for cryptography. Eurocrypt'93, LNCS, 1994, vol. 765, no. 2, pp. 55–64.
2. *Browning K. A., Dillon J. F., McQuistan M. T., and Wolfe A. J.* An APN Permutation in Dimension Six. Post-proceedings of the 9-th Intern. Conf. on Finite Fields and Their Applications Fq'09, Contemporary Math., AMS, 2010, vol. 518, pp. 33–42.
3. *Daemen J. and Rijmen V.* The Design of Rijndael: AES — The Advanced Encryption Standard. Springer, 2002. 238 p.

4. *Qu L., Fu S., Dai Q., and Li C.* When a Boolean Function can be Expressed as the Sum of two Bent Functions. Cryptology ePrint Archive, 2014/048.

5. *Zieschang T.* Combinatorial Properties of Basic Encryption Operations. Eurocrypt'97, LNCS, 1997, vol. 1233, pp. 14–26.

6. *Agibalov G. P.* Shifry s vodyanymi znakami [Watermarking Ciphers]. Prikladnaya diskretnaya matematika. Prilozhenie, 2015, no. 8, pp. 54–59. (in Russian)

7. `http://writeupsd.blogspot.ru/2014/11/apn-permutation-finder.html`.

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

УДК 519.1

КОЛИЧЕСТВО НЕДОСТИЖИМЫХ СОСТОЯНИЙ В КОНЕЧНЫХ ДИНАМИЧЕСКИХ СИСТЕМАХ ДВОИЧНЫХ ВЕКТОРОВ, АССОЦИИРОВАННЫХ С ОРИЕНТАЦИЯМИ ПАЛЬМ

А. В. Жаркова

Саратовский государственный университет имени Н. Г. Чернышевского, г. Саратов, Россия

Рассматриваются конечные динамические системы двоичных векторов, ассоциированных с ориентациями пальм. Данной системе изоморфна конечная динамическая система (B^{s+c}, γ), $s > 0$, $c > 1$, состояниями которой являются все возможные двоичные векторы размерности $s + c$. Приведены формулы для подсчёта количества недостижимых и, как следствие, количества достижимых состояний в рассматриваемых динамических системах, представлены соответствующие статистические таблицы для систем с различными параметрами s и c.

Ключевые слова: *конечная динамическая система, недостижимое состояние, пальма, сверхстройное (звездообразное) дерево.*

DOI 10.17223/20710410/29/5

NUMBER OF INACCESSIBLE STATES IN FINITE DYNAMIC SYSTEMS OF BINARY VECTORS ASSOCIATED WITH PALMS ORIENTATIONS

A. V. Zharkova

Saratov State University, Saratov, Russia

E-mail: VAnastasiyaV@gmail.com

Finite dynamic systems of binary vectors associated with palms orientations are considered. A palm is a tree which is a union of paths having a common end vertex and all these paths, except perhaps one, have the length 1. States of a dynamic system (P_{s+c}, γ), $s > 0$, $c > 1$, are all possible orientations of a palm with trunk length s and leafs number c, and evolutionary function γ transforms the given palm orientations by reversing all arcs that enter into sinks. The following formula for the number of inaccessible states in the considered dynamic systems is proved: $2^{s+c}-2^s-2^{s-3}+\Omega(-1)-2\Omega(1)+\Omega(3)$, where $\Omega(x) = \sum\limits_{i=1}^{[(s-x)/4]} (-1)^{i+1}\cdot 2^{s-x-4i}\cdot C^i_{s-x-3i}$. As a corollary, the number of accessible states equals $2^s+2^{s-3}-\Omega(-1)+2\Omega(1)-\Omega(3)$. The corresponding statistical tables for systems with different parameters s and c are given.

Keywords: *finite dynamic system, inaccessible state, palm, starlike tree.*

Введение

Графовые модели, в которых отказы процессоров интерпретируются как удаление соответствующих вершин, а отказы сетевых каналов — как удаление дуг, занимают важное место в задачах, связанных с отказоустойчивостью компьютерных сетей. Здесь можно выделить следующие три основные конструкции, получившие и самостоятельное значение в теории графов: минимальное расширение графа [1, 2], Т-неприводимое расширение графа [3], бесконтурный граф с заданной структурой источников и стоков [4]. В модели [4] в качестве механизма восстановления работоспособности сети предлагается так называемая SER-динамика бесконтурных связных ориентированных графов. Это позволяет использовать при изучении модельных графов идеи и методы теории конечных динамических систем и, в частности, динамических систем двоичных векторов (см., например, [5, 6]) — когда имеется естественная двоичная кодировка графов рассматриваемого класса. В указанных работах по отказоустойчивости графовых систем основные результаты получены для систем, в основе которых лежат цепи, циклы и частные типы деревьев. К числу деревьев, для которых найдено описание как минимальных, так и Т-неприводимых расширений, относятся пальмы [2, 3]. Дерево называется *пальмой*, если оно является объединением цепей, имеющих общую концевую вершину, причём все эти цепи, за исключением, быть может, одной, имеют длину 1. Пальма является частным случаем *сверхстройного* (*звездообразного*) *дерева* (дерево, в котором в точности одна вершина имеет степень больше 2). В настоящей работе пальмы изучаются с точки зрения динамического подхода к отказоустойчивости графовых систем.

Под *конечной динамической системой* понимается пара (S, δ), где S — конечное непустое множество, элементы которого называются *состояниями системы*; $\delta : S \to S$ — отображение множества состояний в себя, называемое *эволюционной функцией системы*. Таким образом, каждой конечной динамической системе сопоставляется карта, представляющая собой орграф с множеством вершин S и дугами, проведёнными из каждой вершины $s \in S$ в вершину $\delta(s)$. Компоненты связности графа, задающего динамическую систему, называются её *бассейнами*. Получается, что каждый бассейн представляет собой контур с входящими в него деревьями. Контуры, в свою очередь, называются предельными циклами или *аттракторами*.

Основными проблемами теории конечных динамических систем являются задачи отыскания эволюционных параметров без проведения динамики. К их числу относится *ветвление* (количество непосредственных предшественников данного состояния) и, в частности, свойство *недостижимости* состояния (то есть когда состояние имеет нулевое ветвление). Автором составлены программы для ЭВМ, позволяющие вычислять различные параметры динамических систем двоичных векторов, ассоциированных с некоторыми типами графов, в частности [7]; описаны недостижимые состояния конечных динамических систем двоичных векторов, ассоциированных с графами [8]; подсчитаны количества недостижимых состояний в системах, связанных с ориентациями цепей и циклов [9].

В данной работе предлагается формула для подсчёта количества недостижимых состояний в конечных динамических системах двоичных векторов, ассоциированных с ориентациями пальм.

1. Описание динамической системы

Пусть пальма p образована объединением цепей p_0, p_1, ..., p_c, имеющих общую концевую вершину. Будем считать, что p_0 имеет среди этих цепей максимальную длину $s \geqslant 1$. Назовём p_0 *стволом пальмы* p, цепи p_1, p_2, ..., p_c, имеющие длину 1, — её *листьями*, а их совокупность — *кроной*. Будем говорить, что p является пальмой типа (s, c). Пальма с точностью до изоморфизма определяется своим типом. При $c = 1$ пальма вырождается в цепь (см., например, [6, 9]), поэтому далее не будем рассматривать этот случай, считая $c > 1$.

Пусть имеется пальма p типа (s, c), $s + c = n$. Перенумеруем рёбра пальмы p, как показано на рис. 1.

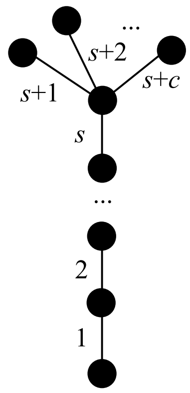


Рис. 1. Нумерация рёбер пальмы

Придадим каждому ребру пальмы произвольную ориентацию и сопоставим полученному ориентированному графу p n-мерный двоичный вектор $v(p)$, полагая его i-ю компоненту равной 1, если i-е ребро пальмы p ориентировано от корня (начальной вершины ствола), и 0 — в противном случае. Теперь можно последовательно выписать получившуюся последовательность из нулей и единиц: $v = v_1 \dots v_s.v_{s+1} \dots v_{s+c}$, где v_i, $0 < i \leqslant s + c$, принимает значение 0 или 1 в зависимости от ориентации i-го ребра пальмы (для упрощения будем работать с двоичным вектором именно в такой записи). Таким образом, каждой ориентации пальмы типа (s, c) сопоставляется n-мерный двоичный вектор, где $n = s + c$. В свою очередь, каждый такой вектор $v = v_1 \dots v_s.v_{s+1} \dots v_{s+c}$ однозначно определяет некоторую ориентацию пальмы $p(v)$ типа (s, c). Таким образом, между множеством P_{s+c}, $s > 0$, $c > 1$, всевозможных ориентированных пальм типа (s, c) указанного вида и множеством B^{s+c}, $s > 0$, $c > 1$, всех двоичных векторов размерности $n = s + c$ указанного вида устанавливается взаимно однозначное соответствие. В дальнейшем ориентации пальмы для простоты также будем называть пальмами, часть $v_1 \dots v_s$ вектора v будем называть *стволом вектора* v, а $v_{s+1} \dots v_{s+c}$ — его *кроной*.

Опишем конечную динамическую систему ориентаций (s, c)-пальмы p на языке двоичных векторов. Пусть состоянием динамической системы в данный момент времени является вектор $v = v_1 \dots v_s.v_{s+1} \dots v_{s+c} \in B^{s+c}$. Тогда в следующий момент времени она окажется в состоянии $\gamma(v) = v'$, полученном путём одновременного применения следующих правил:

1) если $v_1 = 0$, то $v'_1 = 1$;
2) если $v_i = 1$ и $v_{i+1} = 0$ для некоторого $0 < i < s$, то $v'_i = 0$ и $v'_{i+1} = 1$;
3) если $v_i = 1$ для некоторого $s < i \leqslant s + c$, то $v'_i = 0$;
4) если $v_s = 1$ и $v_i = 0$ для всех $s < i \leqslant s+c$, то $v'_s = 0$ и $v'_i = 1$ для всех $s < i \leqslant s+c$;
5) других отличий между v и $\gamma(v)$ нет.

Например, на рис. 2 показана эволюция вектора 011.11 в динамической системе (B^{3+2}, γ).

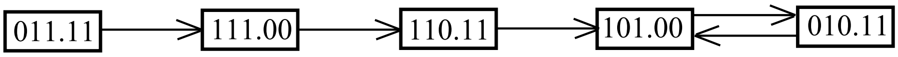
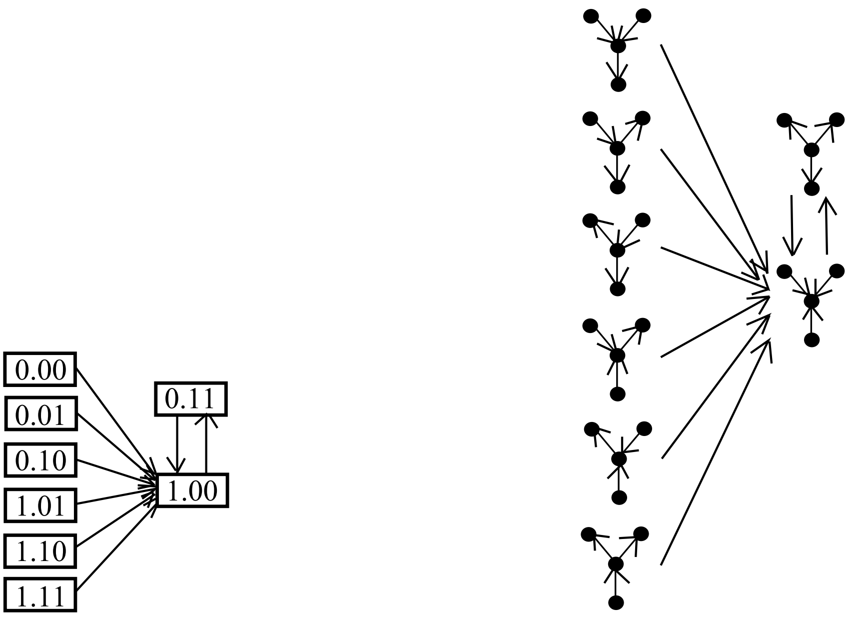


Рис. 2. Эволюция состояния 011.11 в динамической системе (B^{3+2}, γ)

Пусть теперь имеется n-рёберная (s, c)-пальма. На языке ориентаций пальм эволюция динамической системы вводится следующим образом: если дана некоторая ориентированная пальма $p \in P_{s+c}$, то её динамическим образом $\gamma(p)$ является пальма, получаемая из p одновременным превращением всех стоков в источники. Напомним, что *стоком* в ориентированном графе называется вершина с нулевой степенью исхода, а *источником* — вершина с нулевой степенью захода. Это частный случай динамики бесконтурных связных ориентированных графов, введённой в [4]. Преобразования ориентаций пальм в динамической системе (P_{s+c}, γ), $s > 0$, $c > 1$, соответствуют эволюционным преобразованиям соотносимых им двоичных векторов в динамической системе (B^{s+c}, γ), $s > 0$, $c > 1$, и обратно, а именно $v(\gamma(p)) = \gamma(v(p))$ [10]. Таким образом, динамические системы (B^{s+c}, γ) и (P_{s+c}, γ), $s > 0$, $c > 1$, изоморфны. Например, на рис. 3 изображены карты изоморфных динамических систем (B^{1+2}, γ) и (P_{1+2}, γ).



а *б*

Рис. 3. Карты динамических систем: *а* — (B^{1+2}, γ); *б* — (P_{1+2}, γ)

2. Недостижимые состояния динамической системы (B^{s+c}, γ)

Состояния, обладающие свойством недостижимости, будем назвать *недостижимыми*, в ином случае — *достижимыми*.

В работе [8] получены следующие свойства недостижимости состояний динамической системы (B^{s+c}, γ), $s > 0$, $c > 1$.

Теорема 1 [8]. Состояние $v = v_1 \dots v_s . v_{s+1} \dots v_{s+c}$ динамической системы (B^{s+c}, γ), $s > 0$, $c > 1$, недостижимо тогда и только тогда, когда выполняется хотя бы одно из следующих условий:

1) $v_1 v_2 = 00$;
2) $v_i v_{i+1} v_{i+2} v_{i+3} = 1100$ для некоторого $0 < i < s - 1$;
3) среди последних c компонент имеются различные;
4) $v_s = v_{s+1} = \dots = v_{s+c} = 1$.

С помощью программы [7] получены данные по количеству недостижимых состояний (КНС) в динамической системе (B^{s+c}, γ), представленные для $0 < s < 8$, $1 < c < 8$ в табл. 1.

Таблица 1

Количество недостижимых состояний в системах (B^{s+c}, γ), $0 < s < 8$, $1 < c < 8$

s	c					
	2	**3**	**4**	**5**	**6**	**7**
1	6	14	30	62	126	254
2	12	28	60	124	252	508
3	24	56	120	248	504	1 016
4	50	114	242	498	1 010	2 034
5	102	230	486	998	2 022	4 070
6	208	464	976	2 000	4 048	8 144
7	424	936	1 960	4 008	8 104	16 296

Выведем формулу для вычисления количества недостижимых состояний в динамической системе (B^{s+c}, γ), $s > 0$, $c > 1$.

Напомним формулу включений и исключений из комбинаторики, которая понадобится. Пусть даны непустые множества $A_1, A_2, \dots, A_m$. Обозначим через $k(A)$ количество элементов, принадлежащих множеству A. Тогда количество различных элементов в объединении множеств $A_1, A_2, \dots, A_m$ подсчитывается по формуле

$$\begin{gathered} k(A_1 \cup A_2 \cup \dots \cup A_m) = k(A_1) + k(A_2) + \dots + k(A_m) - k(A_1 \cap A_2) - k(A_1 \cap A_3) - \\ - \dots - k(A_1 \cap A_m) - k(A_2 \cap A_3) - \dots - k(A_2 \cap A_m) - \dots - k(A_{m-1} \cap A_m) + \\ + k(A_1 \cap A_2 \cap A_3) + \dots ; \end{gathered} \tag{1}$$

если число пересекающихся множеств нечётно, то слагаемое входит со знаком плюс, если чётно — со знаком минус.

Теорема 2. Количество недостижимых состояний в динамической системе (B^{s+c}, γ), $s > 0$, $c > 1$, равно

$$\text{КНС}_{(s+c,\gamma)} = 2^{s+c} - 2^s - 2^{s-3} + \Omega(-1) - 2\Omega(1) + \Omega(3),$$

где

$$\Omega(x) = \sum_{i=1}^{[(s-x)/4]} (-1)^{i+1} \cdot 2^{s-x-4i} \cdot C_{s-x-3i}^{i}, \tag{2}$$

причём если коэффициенты или степени принимают отрицательные значения, то соответствующие выражения принимают значение 0.

Доказательство. В соответствии с видом недостижимых состояний обозначим множество недостижимых состояний, у которых $v_1v_2 = 00$, через A; у которых $v_iv_{i+1}v_{i+2}v_{i+3} = 1100$ для некоторого i, $0 < i < s-1$, — через B; у которых среди последних c компонент имеются различные — через C; у которых $v_s = v_{s+1} = \ldots = v_{s+c} = 1$ — через D. Подсчитаем общее количество недостижимых состояний в системе, для чего применим формулу включений и исключений (1). Получим, что

$$\begin{gathered}\text{КНС}_{(s+c,\gamma)} = k(A\cup B\cup C\cup D) = k(A) + k(B) + k(C) + k(D) - k(A\cap B)-\\ -k(A\cap C) - k(A\cap D) - k(B\cap C) - k(B\cap D) - k(C\cap D) + k(A\cap B\cap C)+\\ +k(A\cap B\cap D) + k(A\cap C\cap D) + k(B\cap C\cap D) - k(A\cap B\cap C\cap D) =\\ = k(A) + k(B) + k(C) + k(D) - k(A\cap B) - k(A\cap C) - k(A\cap D) - k(B\cap C)-\\ -k(B\cap D) + k(A\cap B\cap C) + k(A\cap B\cap D).\end{gathered} \tag{3}$$

1. Подсчитаем $k(A)$. Для состояний, у которых $v_1v_2 = 00$, получается, что две начальные компоненты зарезервированы, а остальные компоненты занимают $s+c-2$ позиции и принимают значения 0 или 1. Таким образом, $k(A) = 2^{s+c-2}$.

2. Подсчитаем $k(B)$. Рассмотрим подробно первые $s+1$ компонент состояния, при этом заметим, что остальные $c-1$ компонент состояния могут принимать значения 0 или 1, то есть $k(B) = 2^{c-1}\cdot k_1(B)$. В данном случае может присутствовать от одной до $[(s+1)/4]$ тетраграмм 1100 включительно. Обозначим количество этих тетраграмм через l. Компоненты, занимаемые тетраграммами, зарезервированы, а остальные занимают $s+1-4l$ позиций и принимают значения 0 или 1. При этом эти l тетраграмм могут занимать различные позиции в состоянии и их количество определяется при помощи формулы числа сочетаний: $C^l_{s+1-4l+l} = C^l_{s+1-3l}$. Но при рассмотрении состояний, содержащих $l+1$ тетраграмм, некоторые тетраграммы уже были учтены, поэтому, применив формулу включений и исключений (1), получим

$$k_1(B) = k(B_{1100(1)}) - k(B_{1100(2)}) + k(B_{1100(3)}) - \ldots$$

и так далее до $[(s{+}1)/4]$-го слагаемого включительно, где $B_{1100(x)}$ — множество недостижимых состояний, имеющих в составе рассматриваемых компонент x тетраграмм 1100. Тогда

$$k_1(B) = 2^{s+1-4}\cdot C^1_{s+1-3} - 2^{s+1-8}\cdot C^2_{s+1-6} + 2^{s+1-12}\cdot C^3_{s+1-9} - \ldots$$

и так далее до $[(s+1)/4]$-го слагаемого включительно. В итоге получается, что

$$k(B) = 2^{c-1}\sum_{i=1}^{[(s+1)/4]}(-1)^{i+1}\cdot 2^{s+1-4i}\cdot C^i_{s+1-3i}.$$

3. Подсчитаем $k(C)$. Для этого нужно исключить состояния, в кроне которых имеются только одинаковые компоненты, то есть имеем $k(C) = 2^s(2^c-2)$.

4. Подсчитаем $k(D)$. Для состояний, у которых $v_s = v_{s+1} = \ldots = v_{s+c} = 1$, получается, что $c+1$ компонент зарезервированы, а остальные компоненты занимают $s-1$ позиций и принимают значения 0 или 1. Таким образом, $k(D) = 2^{s-1}$.

5. Подсчитаем $k(A\cap B)$. Для состояний, которые имеют в своём составе одновременно начальную биграмму 00 и у которых $v_iv_{i+1}v_{i+2}v_{i+3} = 1100$ для некоторого i, $0 < i < s-1$, получаем ситуацию, аналогичную рассмотренной в п. 2, только тут ещё постоянно зарезервированы первые две компоненты. Таким образом,

$$k(A\cap B) = 2^{c-1}\sum_{i=1}^{[(s-1)/4]}(-1)^{i+1}\cdot 2^{s-1-4i}\cdot C^i_{s-1-3i}.$$

6. Подсчитаем $k(A \cap C)$. Для состояний, у которых $v_1v_2 = 00$ и среди последних c компонент имеются различные, рассмотрим два случая:

а) $s = 1$. Получаем, что две первые компоненты состояния постоянно зарезервированы, а остальные компоненты принимают значения 0 или 1, однако нужно исключить ещё состояния, в стволе которых имеются только одинаковые компоненты. Так как $v_2 = 0$, нужно исключить единственное состояние, у которого $v_i = 0$ для $0 < i \leqslant s + c$. Таким образом, $k(A \cap C) = 2^{s+c-2} - 1$.

б) $s > 1$. Получаем, что две первые компоненты ствола состояния постоянно зарезервированы, а остальные компоненты принимают значения 0 или 1, однако нужно исключить ещё состояния, в кроне которых имеются только одинаковые компоненты. Таким образом, $k(A \cap C) = 2^{s-2}(2^c - 2) = 2^{s+c-2} - 2^{s-1}$.

В общем случае имеем $k(A \cap C) = 2^{s+c-2} - 2^{s-1}$.

7. Подсчитаем $k(A \cap D)$. Для состояний, у которых $v_1v_2 = 00$ и $v_s = v_{s+1} = \ldots = v_{s+c} = 1$, обязательно будет $s > 2$, при этом три компоненты ствола и все компоненты кроны зарезервированы, а остальные принимают значения 0 или 1. Тогда $k(A \cap D) = 2^{s-3}$.

8. Подсчитаем $k(B \cap C)$. Для состояний, у которых $v_iv_{i+1}v_{i+2}v_{i+3} = 1100$ для некоторого i, $0 < i < s - 1$, и у которых среди последних c компонент имеются различные, получаем ситуацию, аналогичную рассмотренной в п. 2, при этом компоненты кроны принимают значения 0 или 1, однако нужно учесть случай, когда в кроне все компоненты одинаковы. Таким образом,

$$k(B \cap C) = (2^{c-1} - 1) \sum_{i=1}^{[(s+1)/4]} (-1)^{i+1} \cdot 2^{s+1-4i} \cdot C^i_{s+1-3i}.$$

9. Подсчитаем $k(B \cap D)$. Для состояний, у которых $v_iv_{i+1}v_{i+2}v_{i+3} = 1100$ для некоторого i, $0 < i < s - 1$, и $v_s = v_{s+1} = \ldots = v_{s+c} = 1$, получаем ситуацию, аналогичную рассмотренной в п. 2, при этом последние $c + 1$ компоненты зарезервированы. Таким образом,

$$k(B \cap D) = \sum_{i=1}^{[(s-1)/4]} (-1)^{i+1} \cdot 2^{s-1-4i} \cdot C^i_{s-1-3i}.$$

10. Подсчитаем $k(A \cap B \cap C)$. Для состояний, у которых $v_1v_2 = 00$, $v_iv_{i+1}v_{i+2}v_{i+3} = 1100$ для некоторого i, $0 < i < s - 1$, и у которых среди последних c компонент имеются различные, получаем ситуацию, аналогичную рассмотренной в п. 8, при этом первые две компоненты зарезервированы. Таким образом,

$$k(A \cap B \cap C) = (2^{c-1} - 1) \sum_{i=1}^{[(s-1)/4]} (-1)^{i+1} \cdot 2^{s-1-4i} \cdot C^i_{s-1-3i}.$$

11. Подсчитаем $k(A \cap B \cap D)$. Для состояний, у которых $v_1v_2 = 00$, $v_iv_{i+1}v_{i+2}v_{i+3} = 1100$ для некоторого i, $0 < i < s-1$, и $v_s = v_{s+1} = \ldots = v_{s+c} = 1$, получаем ситуацию, аналогичную рассмотренной в п. 9, при этом первые две компоненты зарезервированы. Таким образом,

$$k(A \cap B \cap D) = \sum_{i=1}^{[(s-3)/4]} (-1)^{i+1} \cdot 2^{s-3-4i} \cdot C^i_{s-3-3i}.$$

Подставляя полученные выражения в формулу (3) и используя введённую функцию $\Omega(x)$ (2), имеем

$$\text{КНС}_{(s+c,\gamma)} = 2^{s+c-2} + 2^{c-1}\Omega(-1) + 2^s(2^c - 2) + 2^{s-1} - 2^{c-1}\Omega(1) - 2^{s+c-2} + \\ +2^{s-1} - 2^{s-3} - (2^{c-1} - 1)\Omega(-1) - \Omega(1) + (2^{c-1} - 1)\Omega(1) + \Omega(3).$$

После приведения подобных слагаемых в итоге имеем, что количество недостижимых состояний в динамической системе (B^{s+c}, γ), ассоциированной с (s, c)-пальмой, $s > 0$, $c > 1$, равно

$$\text{КНС}_{(s+c,\gamma)} = 2^{s+c} - 2^s - 2^{s-3} + \Omega(-1) - 2\Omega(1) + \Omega(3),$$

причём если коэффициенты или степени принимают отрицательные значения, то это значит, что при таких размерностях s и c просто не возникает подобных ситуаций, и эти выражения принимают значение 0. ■

В табл. 2 приведены данные по количеству недостижимых состояний в динамических системах (B^{s+c}, γ) для различных s и c, полученные с помощью вычислительных экспериментов.

Таблица 2

Количество недостижимых состояний в системе (B^{s+c}, γ)

s	c			
	2	**3**	**18**	**35**
8	862	1886	67108702	8796093022046
9	1750	3798	134217430	17592186044118
10	3548	7644	268434908	35184372088284
11	7184	15376	536869904	70368744176656
12	14530	30914	1073739970	140737488353474
13	29358	62126	2147480238	281474976707246
14	59264	124800	4294961024	562949953415040
15	119536	250608	8589923056	1125899906831088
16	240926	503070	17179847966	2251799813664030
17	485262	1009550	34359699342	4503599627331470
18	976796	2025372	68719404956	9007199254669212
19	1965128	4062280	137438821448	18014398509349960
20	3951474	8145778	274877664114	36028797018721138
27	519578784	1056449696	35184354796704	4611686018410095776
35	135174079768	272613033240	9007196989867288	1180591620715146429720

Следствие 1. Количество достижимых состояний в динамической системе (B^{s+c}, γ), $s > 0$, $c > 1$, равно

$$\text{КДС}_{(s+c,\gamma)} = 2^s + 2^{s-3} - \Omega(-1) + 2\Omega(1) - \Omega(3),$$

где $\Omega(x)$ задаётся формулой (2), причём если коэффициенты или степени принимают отрицательные значения, то соответствующие выражения принимают значение 0.

Согласно следствию 1, количество достижимых состояний в динамических системах (B^{s+c}, γ), $s > 0$, $c > 1$, не зависит от c; таким образом, в данных системах при равенстве s совпадает и количество достижимых состояний.

В табл. 3 приведены данные по количеству достижимых состояний (КДС) в динамических системах (B^{s+c}, γ) для $0 < s < 41$ и $c > 1$, полученные с помощью вычислительных экспериментов.

Таблица 3

Количество достижимых состояний в системах (B^{s+c}, γ), $0 < s < 41$, $c > 1$

s	**КДС**$_{(s+c,\gamma)}$	s	**КДС**$_{(s+c,\gamma)}$	s	**КДС**$_{(s+c,\gamma)}$	s	**КДС**$_{(s+c,\gamma)}$
1	2	**11**	1008	**21**	446634	**31**	197900192
2	4	**12**	1854	**22**	821488	**32**	363995202
3	8	**13**	3410	**23**	1510952	**33**	669491554
4	14	**14**	6272	**24**	2779074	**34**	1231386948
5	26	**15**	11536	**25**	5111514	**35**	2264873704
6	48	**16**	21218	**26**	9401540	**36**	4165752206
7	88	**17**	39026	**27**	17292128	**37**	7662012858
8	162	**18**	71780	**28**	31805182	**38**	14092638768
9	298	**19**	132024	**29**	58498850	**39**	25920403832
10	548	**20**	242830	**30**	107596160	**40**	47675055458

3. Дополнительные замечания

Рассмотрев последовательности для количества достижимых состояний в динамических системах (B^{s+c}, γ), $0 < s < 41$, $c > 1$, введём функцию

$$f(k) = \sum_{i=1}^{3} f(k-i),\ k > 3;\quad f(1) = 2,\ f(2) = 4,\ f(3) = 8. \tag{4}$$

Первые 40 элементов рассматриваемой последовательности вычисляются с помощью рекуррентной формулы (4). Если эта закономерность распространяется на всю последовательность, то количество недостижимых состояний динамической системы (B^{s+c}, γ), $s > 0$, $c > 1$, можно подсчитать с помощью рекуррентного соотношения (4) по формуле

$$\text{КНС}_{(s+c,\gamma)} = 2^{s+c} - f(s).$$

Заметим, что соответствующие последовательности для количества достижимых состояний динамических систем (B^{s+c}, γ), $0 < s < 41$, $c > 1$, и для количества достижимых состояний динамических систем (B^n, δ) [9], $1 < n < 42$, совпадают.

В онлайн-энциклопедии целочисленных последовательностей присутствует последовательность A135491 [11]: «2, 4, 8, 14, ..., 2264873704», элементы которой получаются по формуле $a(k) = a(k-1) + a(k-2) + a(k-3)$ для $3 < k < 36$ при $a(1) = 2$, $a(2) = 4$, $a(3) = 8$. Данная последовательность связана с задачей о бросании монеты [12] и подсчитывает количество способов подбросить монету k раз так, чтобы в результате в последовательности исходов не было четырёх подряд стоящих одинаковых исходов. Последовательность A135491 совпадает с последовательностями достижимых состояний в системах (B^{s+c}, γ) с 1 по 35 элемент (именно столько элементов приведено в онлайн-энциклопедии).

Проанализировав последовательность недостижимых состояний для $0 < s < 41$ и $1 < c < 41$, можно также заметить, что

$$\text{КНС}_{(s+c,\gamma)} = 2^{s+c-3} + \sum_{i=1}^{3} \text{КНС}_{(s+c-i,\gamma)}$$

при соответствующих начальных значениях.

Заключение

В работе получены формулы для подсчёта количества недостижимых и, как следствие, количества достижимых состояний в конечных динамических системах двоичных векторов, ассоциированных с ориентациями пальм, приведены различные статистические данные.

ЛИТЕРАТУРА

1. *Hayes J. P.* A graph model for fault-tolerant computing system // IEEE Trans. Comput. 1976. V. C25. No. 9. P. 875–884.
2. *Абросимов М. Б.* Графовые модели отказоустойчивости. Саратов: Изд-во Сарат. ун-та, 2012. 192 с.
3. *Курносова С. Г.* Т-неприводимые расширения для некоторых классов графов // Теоретические проблемы информатики и ее приложений. 2004. Вып. 6. С. 113–125.
4. *Barbosa V. C.* An Atlas of Edge-Reversal Dynamics. Boca Raton: Chapman&Hall/CRC, 2001. 385 p.
5. *Colon-Reyes O., Laubenbacher R., and Pareigis B.* Boolean monomial dynamical systems // Ann. Combinatorics. 2004. V. 8. P. 425–439.
6. *Салий В. Н.* Об одном классе конечных динамических систем // Вестник Томского государственного университета. Приложение. 2005. № 14. С. 23–26.
7. *Власова А. В.* Исследование эволюционных параметров в динамических системах двоичных векторов // Свидетельство о государственной регистрации программы для ЭВМ № 2009614409, выданное Роспатентом. Зарегистрировано в Реестре программ для ЭВМ 20 августа 2009 г.
8. *Жаркова А. В.* Аттракторы в конечных динамических системах двоичных векторов, ассоциированных с ориентациями пальм // Прикладная дискретная математика. 2014. № 3(25). С. 58–67.
9. *Жаркова А. В.* Недостижимые состояния в динамических системах, ассоциированных с цепями и циклами // Изв. Сарат. ун-та. Нов. сер. 2011. Т. 11. Сер. Математика. Механика. Информатика. Вып. 4. С. 116–123.
10. *Власова А. В.* Динамические системы, определяемые пальмами // Компьютерные науки и информационные технологии: Материалы Междунар. науч. конф. Саратов: Изд-во Сарат. ун-та, 2009. С. 57–60.
11. `https://oeis.org/A135491` — Sequence A135491. Онлайн-энциклопедия целочисленных последовательностей. Дата обращения: 04.08.2015.
12. `http://mathworld.wolfram.com/CoinTossing.html` — Coin tossing. Wolfram MathWorld: the web's most extensive mathematical resource. Дата обращения: 04.08.2015.

REFERENCES

1. *Hayes J. P.* A graph model for fault-tolerant computing system. IEEE Trans. Comput., 1976, vol. C25, no. 9, pp. 875–884.
2. *Abrosimov M. B.* Grafovye modeli otkazoustoychivosti [Graph Models for Fault-Tolerance]. Saratov, SSU Publ., 2012. 192 p. (in Russian)
3. *Kurnosova S. G.* T-neprivodimye rasshireniya dlya nekotorykh klassov grafov [T-irreducible extensions of some classes of graphs]. Teoreticheskie problemy informatiki i ee prilozheniy, 2004, iss. 6, pp. 113–125. (in Russian)
4. *Barbosa V. C.* An Atlas of Edge-Reversal Dynamics. Boca Raton: Chapman&Hall/CRC, 2001. 385 p.

5. *Colon-Reyes O., Laubenbacher R., and Pareigis B.* Boolean monomial dynamical systems. Ann. Combinatorics, 2004, vol. 8, pp. 425–439.
6. *Salii V. N.* Ob odnom klasse konechnykh dinamicheskikh sistem [On a class of finite dynamic systems]. Vestnik Tomskogo gosudarstvennogo universiteta. Prilozhenie, 2005, no. 14, pp. 23–26. (in Russian)
7. *Vlasova A. V.* Issledovanie evolyutsionnykh parametrov v dinamicheskikh sistemakh dvoichnykh vektorov [The study of evolutionary parameters in dynamic systems of binary vectors]. Certificate of state registration of the computer program No. 2009614409, 20 august 2009. (in Russian)
8. *Zharkova A. V.* Attraktory v konechnykh dinamicheskikh sistemakh dvoichnykh vektorov, assotsiirovannykh s orientatsiyami pal'm [Attractors in finite dynamic systems of binary vectors associated with palms orientations]. Prikladnaya diskretnaya matematika, 2014, no. 3(25), pp. 58–67. (in Russian)
9. *Zharkova A. V.* Nedostizhimye sostoyaniya v dinamicheskikh sistemakh, assotsiirovannykh s tsepyami i tsiklami [Inaccessible states in dynamic systems associated with paths and cycles]. Izv. Sarat. un-ta. Nov. ser., 2011, vol. 11. Ser. Matematika. Mekhanika. Informatika, vyp. 4, pp. 116–123. (in Russian)
10. *Vlasova A. V.* Dinamicheskie sistemy, opredelyaemye pal'mami [Dynamic systems defined by palm trees]. Komp'yuternye nauki i informatsionnye tekhnologii: Materialy Mezhdunar. nauch. konf. Saratov, SSU Publ., 2009. pp. 57–60. (in Russian)
11. `https://oeis.org/A135491` — Sequence A135491. The online encyclopedia of integer sequences. Date use: 04.08.2015.
12. `http://mathworld.wolfram.com/CoinTossing.html` — Coin tossing. Wolfram MathWorld: the web's most extensive mathematical resource. Date use: 04.08.2015.

УДК 519.172

ЭВОЛЮЦИОННО-ФРАГМЕНТАРНЫЙ АЛГОРИТМ НАХОЖДЕНИЯ МАКСИМАЛЬНОГО ПЛАНАРНОГО СУГРАФА

И. В. Козин, С. В. Курапов, С. И. Полюга

Запорожский национальный университет, г. Запорожье, Украина

Рассматривается задача нахождения максимального планарного суграфа в несепарабельном неориентированном графе. Показано, что эта задача может быть представлена как задача оптимизации на фрагментарной структуре. Предложен эволюционно-фрагментарный алгоритм поиска приближённых решений задачи.

Ключевые слова: *граф, максимальный планарный суграф, изометрические циклы, фрагментарная структура, эволюционно-фрагментарный алгоритм.*

DOI 10.17223/20710410/29/6

EVOLUTIONARILY-FRAGMENTED ALGORITHM FOR FINDING A MAXIMAL FLAT PART OF A GRAPH

I. V. Kozin, S. V. Kurapov, S. I. Poljuga

Zaporizhzhya National University, Zaporizhzhya, Ukraine

E-mail: ainc00@gmail.com, lilili5050@rambler.ru, veta99@mail.ru.

The problem of finding a maximal flat part of a separable undirected graph is considered. It is shown that this problem can be represented as an optimization problem on a fragmented structure. An evolutionary-fragmented algorithm for finding approximate solutions of the problem is proposed.

Keywords: *graph, maximally flat part of graph, isometric cycles, fragmented structure, evolutionarily-fragmented algorithm.*

Введение

При построении рисунка графа [1, 2] в автоматизированных системах проектирования плоских конструктивов, таких, как печатные платы, интегральные микросхемы, СБИС и т. д., возникает задача выделения плоской части графа [3, 4]. Будем рассматривать задачу построения максимальной плоской части графа $G(X, U)$. Известно, что эта задача является трудно решаемой [5, 6]. Поэтому оправдан поиск методов приближённого решения задачи.

Определение 1 [7]. Неориентированный граф без петель и кратных рёбер, без мостов и точек сочленения, без вершин с локальной степенью 1 и 2 будем называть несепарабельным неориентированным графом.

Определение 2. Планарный суграф с максимальным количеством рёбер графа называется максимальным планарным суграфом этого графа.

Множество суграфов в несепарабельном неориентированном графе можно рассматривать как линейное пространство над полем $\mathbb{Z}_2$. Операция суммы $G_1 \oplus G_2$ суграфов G_1

и G_2 определена как суграф, множество рёбер которого является симметрической разностью множеств рёбер суграфов G_1 и G_2 [8].

В качестве базисной системы векторов пространства суграфов выбираются суграфы с единственным ребром. Размерность пространства суграфов графа G, состоящего из n вершин и m рёбер, равна m. В пространстве суграфов можно выделить два подпространства, которые называются подпространством разрезов и подпространством циклов графа. Размерность подпространства разрезов равна $n-1$, а элементы этого подпространства называются квалиразрезами. Размерность подпространства циклов равна $m-n+1$, а элементы этого подпространства называются квазициклами [9].

В любом несепарабельном неориентированном графе можно выделить подмножество простых циклов — множество C_τ изометрических циклов графа [10].

Определение 3 [11, 12]. Изометрическим циклом в графе называется простой цикл, для которого кратчайший путь между любыми двумя его вершинами состоит из рёбер этого цикла.

Будем искать приближённое решение задачи построения максимального планарного суграфа в виде суммы некоторого набора изометрических циклов.

1. Фрагментарная структура и фрагментарный алгоритм

Определение 4. Фрагментарной структурой (Y, E) на конечном множестве Y называется семейство его подмножеств $E = \{E_1, E_2, \dots, E_n\}$, такое, что

$$\forall E_i \in E, E_i \neq \varnothing\ \exists e \in E_i\ (E_i \setminus \{e\} \in E).$$

Элементы из множества E будем называть допустимыми фрагментами.

Таким образом, для любого допустимого фрагмента E_i существует нумерация его элементов $E_i = \{e_{i1}, e_{i2}, \dots, e_{is_i}\}$, такая, что $\{e_{i1}, e_{i2}, \dots e_{ik}\} \in E$ для всех $k = 1, 2, \dots, s_i$.

Определение 5. Одноэлементные множества, которые являются допустимыми фрагментами, будем называть элементарными фрагментами.

Определение 6. Фрагмент называется максимальным, если он не является подмножеством никакого другого фрагмента.

Очевидны следующие свойства фрагментов:

С в о й с т в о 1. Пустое множество является допустимым фрагментом: $\varnothing \in E$.

С в о й с т в о 2. Пусть $M = \max\limits_{i \in \{1,\dots,n\}} |E_i|$. Тогда для любого целого числа m в интервале $0 \leqslant m \leqslant M$ найдётся элемент в множестве E, мощность которого равна m.

Теорема 1 [13]. Если (Y, E) — фрагментарная структура на множестве Y, то для любого непустого множества $A \in E$ существует нумерация его элементов $A = \{x_1, x_2, \dots, x_m\}$, такая, что для всех $k = 1, 2, \dots, m$ множество $\{x_1, x_2, \dots, x_k\} \in E$.

Из теоремы вытекает, что всякий допустимый фрагмент можно построить из пустого множества, последовательно добавляя к нему элементы так, чтобы на каждом шаге такой процедуры полученное подмножество было допустимым фрагментом.

Максимальный фрагмент может быть построен с помощью следующего «жадного» алгоритма:

- элементы множества Y линейно упорядочиваются;
- на начальном шаге выбирается пустое множество $Y_0 = \varnothing$;
- на шаге с номером $k+1$ выбирается первый по порядку элемент $y \in Y \setminus Y_k$, такой, что $Y_k \cup \{y\} \in E$. Строится множество $Y_{k+1} = Y_k \cup \{y\}$;

— алгоритм заканчивает работу, если на очередном шаге не удалось найти элемент $y \in Y \setminus Y_k$ с требуемым свойством.

Приведённый алгоритм будем называть *фрагментарным алгоритмом*. Результат применения фрагментарного алгоритма определяется заданным линейным порядком на множестве Y. Таким образом, любой максимальный фрагмент может быть задан некоторой перестановкой элементов множества Y. Пусть $A \in E$. Условие для элемента $y \in Y$, при котором $A \cup \{y\} \in E$, будем называть условием присоединения элемента y.

2. Фрагментарная структура на множестве циклов

С множеством изометрических циклов в графе $G(X, U)$ связан ряд инвариантов графа [10]. Одним из таких инвариантов является мощность множества изометрических циклов $\operatorname{card} C_\tau$. Другим инвариантом может служить вектор количества изометрических циклов, упорядоченный по возрастанию их длин. Следующим инвариантом, по аналогии с вектором локальных степеней, является вектор количества изометрических циклов, проходящих по рёбрам графа, — будем называть его вектором циклов по рёбрам:

$$V_u = (a_{u_1}, a_{u_2}, a_{u_3}, \ldots, a_{u_m}),$$

где a_{u_i} — количество изометрических циклов суграфа, проходящих по ребру u_i.

Инвариантом является также вектор количества изометрических циклов, проходящих по вершинам графа. Будем называть его вектором циклов по вершинам:

$$V_x = (a_{x_1}, a_{x_2}, a_{x_3}, \ldots, a_{x_n}),$$

где a_{x_j} — количество циклов, проходящих через вершину x_j.

Для любого подмножества изометрических циклов C_τ мощностью k можно определить значение функционала Маклейна:

$$F(C_\tau)_k = \sum_{i=1}^{m} a_{u_i}^2 - 3 \sum_{i=1}^{m} a_{u_i} + 2m, \tag{1}$$

где a_{u_i} — количество циклов, проходящих по ребру с номером i, а m — количество рёбер графа (суграфа).

Следует заметить, что функционал Маклейна принимает нулевое значение тогда, когда выбранное подмножество изометрических циклов определяет планарный суграф в графе G:

$$F(C_\tau)_k = 0. \tag{2}$$

Между количеством выбранных изометрических циклов k, количеством рёбер m и количеством вершин n для данного подмножества изометрических циклов существует связь (формула Эйлера):

$$k - m + n - 1 = 0. \tag{3}$$

Определение 7. Сумму элементов подмножества изометрических циклов мощностью k будем называть ободом:

$$c_{0k} = c_1 \oplus c_2 \oplus \ldots \oplus c_k.$$

Будем рассматривать множество циклов графа как фрагментарную структуру, причём изометрические циклы представляют элементарные фрагменты этой структуры. Любой планарный суграф графа G можно представить в виде суммы изометрических циклов (элементарных фрагментов).

Построение максимального фрагмента для задачи построения максимальной плоской части графа опишем с помощью приведённого выше фрагментарного алгоритма с соблюдением следующих свойств:

а) элементы множества изометрических циклов C_τ представляем в виде кортежа (упорядочиваем и нумеруем);
б) на начальном шаге выбираем пустое множество изометрических циклов $C_p = \varnothing$;
в) на шаге с номером k выбираем следующий по порядку элемент кортежа, имеющий общее ребро с внешним простым циклом (ободом), построенным как сумма циклов предыдущих шагов $c_{0k} = c_{0(k-1)} \oplus c_k$, при условии, что обод не является пустым множеством;
г) если выполнены условия (2) и (3), то выбранный изометрический цикл помещается в множество C_p;
д) алгоритм заканчивает работу, если на очередном шаге не удалось найти элемент $c_i \in C_\tau$ с требуемым свойством.

Рассмотрим работу данного алгоритма на примере.

Пример 1. Выделим плоскую часть графа G, представленного на рис. 1.

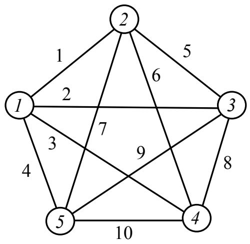


Рис. 1. Граф G

Множество изометрических циклов графа G представляем в виде кортежа:

$$C_\tau = \langle c_1, c_2, c_3, c_4, c_5 c_6, c_7, c_8, c_9, c_{10} \rangle :$$
$$c_1 = \{u_1, u_2, u_5\} \to (x_1, x_2, x_3)\,;\;\; c_2 = \{u_1, u_3, u_6\} \to (x_1, x_2, x_4)\,;$$
$$c_3 = \{u_1, u_4, u_7\} \to (x_1, x_2, x_5)\,;\;\; c_4 = \{u_2, u_3, u_8\} \to (x_1, x_3, x_4)\,;$$
$$c_5 = \{u_2, u_4, u_9\} \to (x_1, x_3, x_5)\,;\;\; c_6 = \{u_3, u_{10}, u_7\} \to (x_1, x_4, x_5)\,;$$
$$c_7 = \{u_3, u_6, u_8\} \to (x_2, x_3, x_4)\,;\;\; c_8 = \{u_5, u_7, u_9\} \to (x_2, x_3, x_5)\,;$$
$$c_9 = \{u_6, u_7, u_{10}\} \to (x_2, x_4, x_5)\,;\;\; c_{10} = \{u_8, u_9, u_{10}\} \to (x_3, x_4, x_5)\,.$$

Для простоты записи иногда рёбра будем обозначать числами.

Выбираем в множестве изометрических циклов первый по порядку цикл и помещаем его в список циклов, характеризующий плоскую часть графа:

$$c_1 = \{u_1, u_2, u_5\} \to (x_1, x_2, x_3)\,,\;\; C_p = \{c_1\}\,.$$

Строим векторы циклов по рёбрам $V_u = (1,1,0,0,1,0,0,0,0,0)$ и по вершинам $V_x = (1,1,1,0,0)$. Вычисляем значение функционала Маклейна (1) для выбранной системы циклов: $F(C_p) = 0$; ограничение (3) также выполнено: $1 - 3 + 3 - 1 = 0$.

Так как цикл единственный, его внешний простой цикл (обод) определяется самим циклом:

$$c_0 = \varnothing \oplus c_1 = \{u_1, u_2, u_5\}\,.$$

Выбираем в множестве изометрических циклов следующий по порядку пересекающийся (в множественном смысле) цикл $c_2 = \{u_1, u_3, u_6\} \to (x_1, x_2, x_4)$, $|c_0 \cap c_2| = 1$. Строим векторы $V_u = (2, 1, 1, 0, 1, 1, 0, 0, 0, 0)$ и $V_x = (2, 2, 1, 1, 0)$. Отсюда определяем значение функционала Маклейна (1) для выбранной системы циклов: $F(C_p) = 0$; ограничение (3) также выполнено: $2 - 5 + 4 - 1 = 0$. Обод определяется суммированием циклов c_1 и c_2:

$$c_0 = c_1 \oplus c_2 = \{1, 2, 5\} \oplus \{1, 3, 6\} = \{2, 3, 5, 6\}.$$

Обод $c_0 \neq \varnothing$. Все условия выполнены. Помещаем цикл c_2 в множество, характеризующее плоскую часть графа: $C_p = \{c_1, c_2\}$.

Выбираем в множестве изометрических циклов следующий пересекающийся цикл:

$$c_3 = \{u_1, u_4, u_7\} \to (x_1, x_2, x_5),\ |c_0 \cap c_3| = 1.$$

Но мы не можем поместить его в список циклов, характеризующий плоскую часть графа, так как значение функционала Маклейна для выбранной совокупности циклов больше нуля, согласно $V_u = (3, 1, 1, 1, 1, 1, 1, 0, 0, 0)$.

Выбираем в множестве изометрических циклов следующий пересекающийся цикл:

$$c_4 = \{u_2, u_3, u_8\} \to (x_1, x_3, x_4),\ |c_0 \cap c_4| = 1.$$

Строим векторы $V_u = (2, 2, 2, 0, 1, 1, 0, 1, 0, 0)$ и $V_x = (3, 2, 2, 2, 0)$. Отсюда определяем значение функционала Маклейна (1) для выбранной системы циклов: $F(C_p) = 0$; ограничение (3) также выполняется: $3 - 6 + 4 - 1 = 0$. Обод определяется как сумма предыдущего обода c_0 и c_4:

$$c_0 = \{2, 3, 5, 6\} \oplus \{2, 3, 8\} = \{5, 6, 8\}.$$

Все условия выполнены, помещаем цикл c_4 в множество, характеризующее плоскую часть графа: $C_p = \{c_1, c_2, c_4\}$.

Циклы c_5 и c_6 не имеют общих рёбер с ободом и исключаются из рассмотрения. Следующим циклом, имеющим с предыдущим ободом общее ребро, является цикл $c_7 = \{u_5, u_6, u_8\} \to (x_2, x_3, x_4)$. Но его присоединение нарушает ограничение (3): $4 - 6 + 4 - 1 \neq 0$. Кроме того, внешний простой цикл (обод) в этом случае есть пустое множество: $c_0 = \{5, 6, 8\} \oplus \{5, 6, 8\} = \varnothing$. Следовательно, цикл не может быть добавлен в множество C_p.

Просматривая далее список, последовательно присоединим цикл c_8, а затем c_9:

$$c_8 = \{u_5, u_7, u_9\} \to (x_2, x_3, x_5)\,;\ c_9 = \{u_6, u_7, u_{10}\} \to (x_2, x_4, x_5)\,.$$

Ограничения (2) и (3) выполняются и циклы c_8 и c_9 могут быть помещены в множество C_p.

В конечном итоге для множества $C_p = \{c_1, c_2, c_4, c_8, c_9\}$ получаем $V_u = (2, 2, 2, 0, 2, 2, 2, 1, 2, 2)$ и $V_x = (3, 4, 3, 3, 2)$. Значение функционала Мак-Лейна (1) для выбранной системы циклов равно нулю: $F(C_p) = 0$; ограничение (3) также выполнено: $5 - 9 + 5 - 1 = 0$. Обод определяется как сумма предыдущего обода и c_8 и c_9:

$$c_0 = \{5, 6, 8\} \oplus \{5, 7, 9\} \oplus \{6, 7, 10\} = \{8, 9, 10\}.$$

На последующих шагах не удается найти цикл, удовлетворяющий условиям (2) и (3); конец работы алгоритма выделения фрагментов (см. рис. 2).

Таким образом, мы выделили множество изометрических циклов, характеризующее плоскую часть графа, описываемую перестановкой $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$. Количество рёбер в данной плоской части $K_u = 9$, а количество вершин $K_x = 5$.

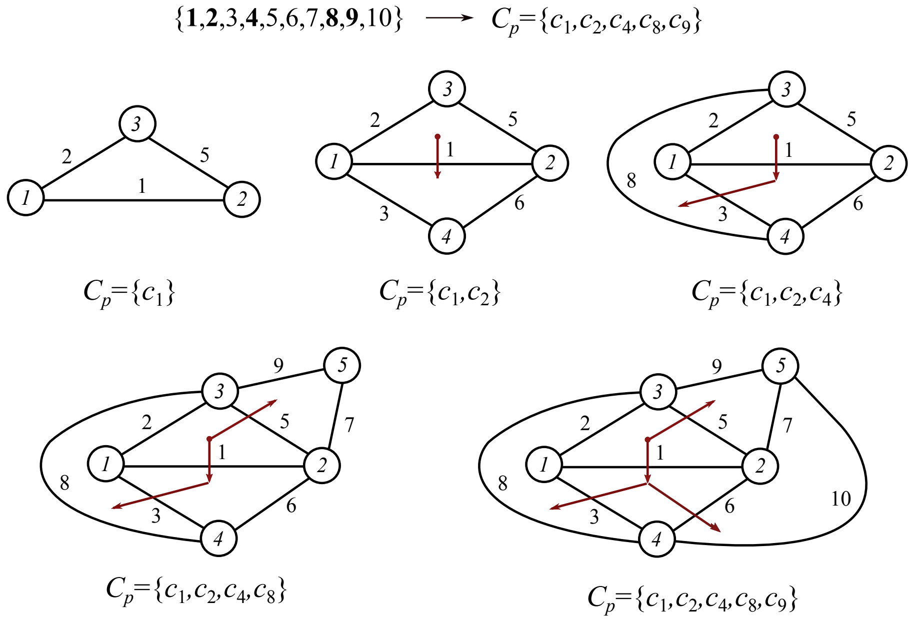


Рис. 2. Плоская часть графа

Отметим в заключение следующее свойство построенной фрагментарной модели задачи: любой максимальный планарный суграф в несепарабельном неориентированном графе G может быть построен предлагаемым методом при надлежащем выборе перестановки элементарных фрагментов. Это свойство фрагментарной модели называется свойством достижимости.

3. Приближённые алгоритмы поиска оптимальных решений на фрагментарных структурах

Из результатов п. 2 вытекает, что задача поиска планарного суграфа с максимальным числом рёбер сводится к поиску максимального фрагмента во фрагментарной структуре, т. е. к комбинаторной задаче $F(s) \xrightarrow[s\in S_N]{} \max$ отыскания некоторой экстремальной перестановки на множестве перестановок изометрических циклов (N — мощность множества изометрических циклов). При этом любая перестановка является допустимой.

Поставим в соответствие каждой перестановке значение критерия на соответствующем ей максимальном фрагменте, а именно число рёбер планарного суграфа. На множестве перестановок можно предложить несколько простых алгоритмов приближённого поиска оптимальной перестановки. Опишем некоторые из них.

а) Метод локального поиска. Множество перестановок может рассматриваться как метрическое пространство с некоторой метрикой $p : S_N \times S_N \to R^1_+$ [14]. Выбирается начальная перестановка и ищется оптимальная перестановка в её ϵ-окрестности. Эта найденная перестановка является базовой для следующего шага. Условие остановки алгоритма: в ϵ-окрестности не существует перестановки с лучшим критерием. В качестве ϵ можно брать любое натуральное число $1, 2, \ldots$

б) Хорошие результаты даёт эволюционная модель на фрагментарной структуре [13, 15]. В этой модели в качестве пространства поиска выбирается множество $S_N = \{i_1, i_2, \ldots, i_N\}$ всех перестановок чисел $1, 2, \ldots, N$.

Оператор построения начальной популяции выделяет произвольное подмножество заданной мощности Q из множества Y. Правило вычисления критерия селекции устроено следующим образом: по заданной перестановке фрагментов с помощью фрагментарного алгоритма строится максимальный допустимый фрагмент. Вычисляется значение целевой функции для этого фрагмента.

Опишем оператор кроссовера Kr. Пусть $U = (u_1, u_2, \ldots, u_N)$ и $V = (v_1, v_2, \ldots, v_N)$ — две произвольные перестановки. Перестановка-потомок строится следующим образом: последовательности U и V просматриваются в порядке следования элементов. На k-м шаге выбирается минимальный из первых элементов последовательностей и добавляется в новую перестановку-потомок. Затем этот элемент удаляется из двух последовательностей-родителей. Например, $\mathrm{Kr}((4,7,3,2,8,1,6,5),(1,4,5,2,6,3,8,7)) = (1,4,5,2,6,3,7,8)$.

Оператор мутации M_α выполняет случайную транспозицию в перестановке. Оператор селекции выбирает случайным образом набор P пар из текущей популяции. Оператор отбора упорядочивает элементы промежуточной популяции по убыванию значения критерия селекции. В качестве новой текущей популяции выбираются первые Q элементов последовательности.

Обычное правило остановки — количество поколений достигло предельного значения L. Лучшая по значению критерия селекции перестановка из последней построенной популяции определяет приближённое решение задачи.

Оценим вычислительную сложность эволюционно-фрагментарного алгоритма. В соответствии с [12] трудоёмкость построения множества изометричных циклов определяется величиной $\mathrm{O}(m^2 n/\log_2 n + mn^2)$. Здесь n — число вершин, m — число рёбер графа. Максимально возможная мощность множества изометрических циклов равна $N = \mathrm{O}(n^3)$. Трудоёмкость фрагментарного алгоритма [15] в рассматриваемом случае ограничена величиной $\mathrm{O}(N^2)$. Таким образом, получаем оценку трудоёмкости эволюционно-фрагментарного алгоритма в виде $\mathrm{O}(QN\log_2 N + LPN^3)$. Следовательно, трудоёмкость приведённого в настоящей работе алгоритма выделения максимального плоского суграфа в графе может быть оценена величиной $\mathrm{O}(3Qn^3\log_2 n + LPn^9)$.

С целью апробации результатов исследования в рамках программной системы «Эволюционно-фрагментарное моделирование» был поставлен численный эксперимент по отысканию максимального планарного суграфа. Для поиска решений использовалась авторская программа, реализующая эволюционную модель на перестановках. Программа написана на VBA под управлением WINDOWS XP/7. Вычисления проводились на компьютере ACPIx64-based PC с процессором 4x 3400 MHz. Время расчёта для каждой задачи было ограничено 3 мин. Количество поколений в разных задачах составляло от 500 до 1000, использовался пропорциональный принцип отбора родителей. Количество эволюций (отдельных запусков алгоритма) менялось от 1 до 10 в зависимости от задачи. Условием остановки являлось достижение границы времени расчёта.

Для оценки эффективности алгоритма было выполнено сравнение с алгоритмом локального поиска со случайным выбором начальной точки и с жадным алгоритмом со случайным выбором порядка элементов. Рассмотрено 100 тестовых задач. Граф в каждой задаче генерировался случайным образом в соответствии с моделью Эрдоша — Реньи [16]. Количество вершин графа менялось в диапазоне 100–500.

Параметры алгоритмов были следующие:

а) локальный поиск со случайным выбором начальной точки: для каждой задачи проводилось 250 запусков алгоритма, выбирался лучший результат;

б) жадный алгоритм со случайным выбором последовательности рёбер графа: количество запусков составляло 2500;

в) эволюционный алгоритм: размер популяции 500, количество поколений 100, количество пар скрещивания в каждом поколении 20, пропорциональный метод отбора пар.

В результате эксперимента установлено, что в 73 % тестов эволюционно-фрагментарный алгоритм даёт лучшие и в 97 % тестов не худшие результаты, чем локальный алгоритм и алгоритм случайного поиска.

Заключение

В работе предложен подход к поиску приближённого решения задачи построения максимальной плоской части графа на основе фрагментарной структуры на множестве изометрических циклов графа. Как следствие, задача поиска оптимального решения сводится к поиску оптимальной перестановки на множестве перестановок. Для таких задач эффективным является применение универсального эволюционно-фрагментарного алгоритма. Построена фрагментарная модель для решения рассматриваемой задачи. Показано, что оптимальное решение задачи выделения максимального планарного суграфа достижимо в рамках фрагментарной модели.

ЛИТЕРАТУРА

1. *Касьянов В. Н., Евстигнеев В. А.* Графы в программировании: обработка, визуализация и применение. СПб.: БХВ-Петербург, 2003. 1104 с.
2. *Tamassia R.* Handbook of Graph Drawing and Visualization. Boca Raton: Chapman and Hall/CRC, 2013. 844 p.
3. *Курапов С. В., Толок А. В.* Методы построения топологического рисунка графа // Автоматика и телемеханика. 2013. № 9. С. 78–97.
4. *Курапов С. В., Давидовский М. В.* Два подхода к проведению соединений в плоских конструктивах // Компоненты и технологии. 2015. № 7. С. 142–147.
5. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
6. *Пападимитриу Х., Стайглиц К.* Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир, 1985. 512 с.
7. *Харари Ф.* Теория графов. М.: Мир, 1973. 300 с.
8. *Свами М., Тхуласираман К.* Графы, сети и алгоритмы. М.: Мир, 1984. 455 с.
9. *Зыков Е. И.* Основы теории графов. М.: Наука, 1987. 384 с.
10. *Курапов С. В., Похальчук Т. А.* Единичные циклы и разрезы для определения изоморфизма графов // Вісник Запорізького національного університету: Збірник наукових праць. Фіз.-мат. науки. 2011. № 2. С. 61–68.
11. *Курапов С. В., Савин В. В.* Векторная алгебра и рисунок графа. Запоріжжя: ЗДУ, 2003. 200 с.
12. *Kavitha T., Liebchen C., Mehlhorn K., Michail D., et al.* Cycle bases in graphs — characterization, algorithms, complexity, and applications // Comput. Sci. Rev. 2009. No. 3. P. 199–243.
13. *Козин И. В., Полюга С. И.* Фрагментарные модели для некоторых экстремальных задач на графах // Математичні машини і системи. 2014. № 1. С. 143–150.
14. *Деза Е. И., Деза М. М.* Энциклопедический словарь расстояний. М.: Наука, 2008. 432 с.
15. *Козин И. В., Полюга С. И.* О свойствах фрагментарных структур // Вісник Запорізького національного університету: Збірник наукових праць. Фіз.-мат. науки. 2012. № 1. С. 99–106.

16. *ErdSos P. and Rɾ'enyi A.* On random graphs I // Publ. Math. Debrecen. 1959. V. 6. P. 290–297.

REFERENCES

1. *Kas'yanov V. N., Evstigneev V. A.* Grafy v programmirovanii: obrabotka, vizualizatsiya i primenenie [Graphs in Programming: Processing, Visualization and Application]. St. Petersburg, BKhV-Peterburg Publ., 2003. 1104 p. (in Russian)
2. *Tamassia R.* Handbook of Graph Drawing and Visualization. Boca Raton: Chapman and Hall/CRC, 2013. 844 p.
3. *Kurapov S. V., Tolok A. V.* The topological drawing of a graph: Construction methods. Automation and Remote Control, 2013, vol. 74, iss. 9, pp. 1494–1509.
4. *Kurapov S. V., Davidovskiy M. V.* Dva podkhoda k provedeniyu soedineniy v ploskikh konstruktivakh [Two approaches to connections conducting in flat form factor]. Komponenty i tekhnologii, 2015, no. 7, pp. 142—147. (in Russian)
5. *Garey M. R. and Johnson D. S.* Computers and Intractability. San Francisco, W. H. Freeman and Co., 1979. 338 p.
6. *Papadimitriou C. H. and Steiglitz K.* Combinatorial Optimization. Algorithms and Complexity. NJ, USA, Prentice-Hall, 1982.
7. *Harary F.* Graph Theory. Addison–Wesley, 1969.
8. *Swamy M. N. S. and Thulasiraman K.* Graphs, Networks and Algorithms. Wiley, 1980. 612 p.
9. *Zykov E. I.* Osnovy teorii grafov [Basics of Graph Theory]. Moscow, Nauka Publ., 1987. 384 p. (in Russian)
10. *Kurapov S. V., Pokhal'chuk T. A.* Edinichnye tsikly i razrezy dlya opredeleniya izomorfizma grafov [Single cycles and cuts to determine the graph isomorphism]. Vistnyk Zaporizkoho Natsionalnoho Universytetu: Zbirnyk naukovykh prats. Fiz.-mat. nauky, 2011, no. 2, pp. 61–68. (in Russian)
11. *Kurapov S. V., Savin V. V.* Vektornaya algebra i risunok grafa [Vector Algebra and Graph Drawing]. Zaporizhzhya, ZDU Publ., 2003. 200 p. (in Russian)
12. *Kavitha T., Liebchen C., Mehlhorn K., Michail D., et al.* Cycle bases in graphs — characterization, algorithms, complexity, and applications. Comput. Sci. Rev., 2009, no. 3, pp. 199–243.
13. *Kozin I. V., Polyuga S. I.* Fragmentarnye modeli dlya nekotorykh ekstremal'nykh zadach na grafakh [Fragmentary model for some extreme problems on graphs]. Matematychni mashyny i sistemy, 2014, no. 1, pp. 143–150. (in Russian)
14. *Deza E. I., Deza M. M.* Entsiklopedicheskiy slovar' rasstoyaniy [Encyclopedic Dictionary of Distances]. Moscow, Nauka Publ., 2008. 432 p. (in Russian)
15. *Kozin I. V., Polyuga S. I.* O svoystvakh fragmentarnykh struktur [Properties of fragmented structures]. Vistnyk Zaporizkoho Natsionalnoho Universytetu: Zbirnyk naukovykh prats. Fiz.-mat. nauky, 2012, no. 1, pp. 99–106. (in Russian)
16. *ErdSos P. and Rɾ'enyi A.* On random graphs I. Publ. Math. Debrecen, 1959, vol. 6, pp. 290–297.

УДК 519.171 + 519.175.1 + 519.17:5

ОБ АЛЬТЕРНАТИВНОМ СПОСОБЕ ЗАДАНИЯ КОНЕЧНЫХ ГРАФОВ

М. Н. Назаров

Национальный исследовательский университет «МИЭТ», г. Москва, Россия

Рассматривается линейная нотация — полный инвариант графов, который позиционируется как альтернатива для описания конечных графов. Данный инвариант строится с помощью алгоритма, близкого к алгоритму поиска канонических форм графов. Хранение линейной нотации в памяти вместо обычного графа позволяет проще решать две основные задачи: построение иллюстраций для графов и сравнение графов на изоморфизм. Для полученного описания дополнительно демонстрируется переносимость таких понятий теории графов, как раскраски и пути, непосредственно на линейные нотации.

Ключевые слова: *изоморфизм графов, классы автоморфизма вершин и рёбер, инварианты графов.*

DOI 10.17223/20710410/29/7

AN ALTERNATIVE WAY OF DEFINING FINITE GRAPHS

M. N. Nazarov

National Research University of Electronic Technology, Moscow, Russia

E-mail: Nazarov-Maximilian@yandex.ru

In this paper, we consider the graph linear notation — a complete graph invariant, which is positioned as an alternative to description of finite graphs. This invariant is constructed using an algorithm which is close to the search algorithm for canonical forms of graphs. The storage in a memory of a graph linear notation instead of the graph itself simplifies the procedures for constructing graph illustrations and testing two graphs for isomorphism. We demonstrate how the main graph theory concepts including colouring and graph paths can be defined in terms of graph linear notations.

Keywords: *graph isomorphism, automorphism classes of vertices and edges, graph invariants.*

Введение

В работе [1] введена линейная нотация абстрактного графа $I[G]$. Для нотации $I[G]$ продемонстрировано, что она является полным инвариантом и по своим свойствам во многом аналогична обычным графам G. В частности, для $I[G]$ введены понятия абстрактных вершин и абстрактных рёбер, а также подграфы, раскраски, пути и некоторые элементарные операции на графах. Отдельно следует отметить, что алгоритм перехода от $I[G]$ к обычным графам G является полиномиальным. Поскольку $I[G]$ является полным инвариантом, данный факт служит основанием для использования нотации $I[G]$ в качестве альтернативного способа хранения графов.

Реализована линейная нотация $I[G]$ с помощью номеров $I(\overline{v})$ и $I(\overline{u,v})$ классов автоморфизма вершин и рёбер графов. Для данных номеров доказано, что они являются однозначными идентификаторами для классов автоморфизма вершин и рёбер и позволяют ввести на множестве этих классов линейные порядки: $\overline{v_1} \leqslant \overline{v_2} \Leftrightarrow I(\overline{v_1}) \leqslant I(\overline{v_2})$ и $(\overline{v_1,u_1}) \leqslant (\overline{v_2,u_2}) \Leftrightarrow I(\overline{v_1,u_1}) \leqslant I(\overline{v_2,u_2})$. Это означает, что можно пронумеровать классы вершин числами $1,\ldots,m$, а классы рёбер — числами $1,\ldots,k$, если у графа m классов автоморфных вершин и k классов автоморфных рёбер. Нужно отметить, что в работе [1] была допущена неточность: во всех иллюстрациях и примерах вершины и рёбра помечены именно такими порядковыми индексами, а не $I(\overline{v})$ и $I(\overline{u,v})$, как утверждается в подписях к ним. В данной работе мы в первую очередь проведём разграничение между индексами и номерами классов автоморфизма вершин и рёбер и переформулируем все основные результаты из [1] в приложении к индексам классов автоморфизма.

Для демонстрации преимуществ линейной нотации $I[G]$ по отношению к классическому описанию графов дополнительно рассмотрим примеры задания стандартных графов, а также приведём алгоритмы для построения путей и реализации раскрасок на $I[G]$.

1. Построение линейных нотаций для классов изоморфных графов

Мы исключаем из рассмотрения ориентированные графы, графы с петлями, кратными рёбрами и бесконечные графы [2]. Отметим, что построить обобщение нашей методики будет относительно просто только для классов ориентированных графов и графов с петлями.

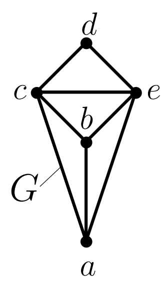
Определение 1. *Графом* будем называть пару $G = (V, E)$, где множество вершин V — это любое конечное множество, а множество рёбер $E \subseteq V \times V$ — это бинарное отношение на V, для которого выполняются следующие условия:

1) $\forall a, b \in V \quad ((a,b) \in E \Rightarrow (b,a) \in E)$ — отношение симметрично;
2) $\forall a \in V \quad ((a,a) \notin E)$ — отношение антирефлексивно.

Определение 2. Пусть дан произвольный граф $G = (V, E)$, у которого $|V| = n$. Тогда, если выбрать некоторый порядок $\alpha = (v_1, \ldots, v_n)$ на множестве вершин V, то можно поставить в соответствие графу G *матрицу смежности* A по следующему правилу:

$$A(i,j) = 1 \Leftrightarrow (v_i, v_j) \in E \quad \& \quad A(i,j) = 0 \Leftrightarrow (v_i, v_j) \notin E.$$

На рис. 1 представлен пример двух матриц смежности, которые построены для разных порядков α_1 и α_2 на множестве вершин V графа G.



$\alpha_1 = (a, b, c, d, e)$

A_1	a	b	c	d	e
a	0	**1**	**1**	0	**1**
b	**1**	0	**1**	0	**1**
c	**1**	**1**	0	**1**	**1**
d	0	0	**1**	0	**1**
e	**1**	**1**	**1**	**1**	0

$\alpha_2 = (c, e, b, d, a)$

A_2	c	e	b	d	a
c	0	**1**	**1**	**1**	**1**
e	**1**	0	**1**	**1**	**1**
b	**1**	**1**	0	0	**1**
d	**1**	**1**	0	0	0
a	**1**	**1**	**1**	0	0

Рис. 1. Пример двух разных матриц смежности A_1 и A_2 для одного графа

Определение 3. *Кодом матрицы смежности* A конечного графа G будем называть число $\mu(A) = A(n,n) + A(n,n-1)\cdot 2 + \ldots + A(n,1)\cdot 2^{n-1} + A(n-1,n)\cdot 2^n + \ldots + A(1,1)\cdot 2^{n^2-1}$, где $A(i,j)$ — элементы матрицы смежности A.

Например, для двух матриц, изображённых на рис. 1, получим следующие коды: $\mu(A_1) = 1101101011101100101111110$ и $\mu(A_2) = 1111101111100111000111 00$.

Определение 4. Наибольший из всех возможных кодов $\mu(A)$ матриц смежности A графа G назовём *макси-кодом* графа $\mu_{\max}(G)$. Если для некоторого порядка вершин α код матрицы смежности $\mu(A) = \mu_{\max}(G)$, то говорим, что α *соответствует* макси-коду.

В [1] доказано следующее утверждение о макси-коде $\mu_{\max}(G)$.

Утверждение 1 [1]. Если два порядка вершин $\alpha_1 = (v_1^1, \ldots, v_n^1)$ и $\alpha_2 = (v_1^2, \ldots, v_n^2)$ соответствуют макси-коду $\mu_{\max}(G)$ графа G, то вершины в этих порядках попарно автоморфны: $v_i^1 \sim v_i^2$ для всех $i = 1, \ldots, n$.

Элементы последовательностей вершин $\alpha = (v_1, \ldots, v_n)$ будем обозначать $v_i = \alpha(i)$.

Определение 5. Пусть макси-коду графа G соответствует некоторый порядок следования вершин α. Назовём *номером класса автоморфизма вершин* $\overline{v}$ натуральное число $N(\overline{v})$, которое равно первому вхождению в порядок α вершины из класса $\overline{v}$: $N(\overline{v}) = \min\limits_{i:\alpha(i)\in\overline{v}} i$.

Определение 6. Пусть макси-коду графа G соответствует некоторый порядок следования вершин α. Назовём *номером класса автоморфизма рёбер* $(\overline{x,y})$ натуральное число $N(\overline{x,y})$, равное номеру первого вхождения ребра из класса $(\overline{x,y})$ в матрицу смежности A графа G, построенную в соответствии с α (при нумерации элементов матрицы по строкам): $N(\overline{x,y}) = \min\limits_{\substack{A(i,j)=1\\(\alpha(i),\alpha(j))\sim(x,y)}} j + (i-1)\cdot n$.

Утверждение 2. Номера $N(\overline{v})$ естественным образом порождают линейный порядок на классах автоморфных вершин: $\overline{v_1} \leqslant \overline{v_2} \Leftrightarrow N(\overline{v_1}) \leqslant N(\overline{v_2})$.

Доказательство. По определению номера $N(\overline{v})$ являются натуральными числами; согласно утверждению 1, номера совпадают только у автоморфных вершин. Получаем, что множество номеров $N(\overline{v_1}), \ldots, N(\overline{v_k})$ линейно упорядочено и, следовательно, порядок $\overline{v_1} \leqslant \overline{v_2} \Leftrightarrow N(\overline{v_1}) \leqslant N(\overline{v_2})$ линейный. ■

Утверждение 3. Номера $N(\overline{v,u})$ естественным образом порождают линейный порядок на классах автоморфных ребер: $(\overline{v_1,u_1}) \leqslant (\overline{v_2,u_2}) \Leftrightarrow N(\overline{v_1,u_1}) \leqslant N(\overline{v_2,u_2})$.

Доказательство. Доказательство аналогично доказательству утверждения 2. Если два порядка вершин α_1 и α_2 соответствуют макси-коду графа $\mu_{\max}(G)$, то матрицы смежности для этих порядков совпадают. В результате рёбра, которые соответствуют одинаковым позициям в этих матрицах, автоморфны. ■

Определение 7. Введём для графа G *индексы классов автоморфизма вершин* $I(\overline{v})$ через номера классов $N(\overline{v})$ с помощью следующих индуктивных правил:

1) $N(\overline{v}) = \min\limits_{\overline{u}} N(\overline{u}) \;\Rightarrow\; I(\overline{v}) = 1$ — базис индукции для определения первого индекса;

2) $$\left.\begin{array}{c}(N(\overline{v}) > N(\overline{u}))\,\&\\ \&\ \forall u^* \neq u\ (N(\overline{v}) > N(\overline{u^*}) \Rightarrow N(\overline{u}) > N(\overline{u^*}))\end{array}\right\} \Rightarrow I(\overline{v}) = I(\overline{u}) + 1 \text{ — переход.}$$

Определение 8. Аналогично определению для вершин введём понятие *индекса классов автоморфизма рёбер* $I(\overline{x,y})$, заменив классы автоморфизма вершин на классы автоморфизма рёбер.

Теорема 1. Если $G \cong H$, то для $u \in V(G)$, $v \in V(H)$ равенство $I(\overline{u}) = I(\overline{v})$ выполняется тогда и только тогда, когда существует изоморфизм $\phi : V(G) \to V(H)$, такой, что $\phi(u) = v$.

Доказательство. Докажем сначала, что из $\phi(u) = v$, где ϕ — изоморфизм, следует совпадение индексов классов автоморфизма $I(\overline{u}) = I(\overline{v})$.

Если два графа изоморфны, то у них один и тот же макси-код $\mu_{\max}(G) = \mu_{\max}(H)$. В этом случае изоморфизм можно представить в виде $\phi = \begin{pmatrix} u_1 \dots u_n \\ v_1 \dots v_n \end{pmatrix}$, где $\alpha_1 = (u_1, \dots, u_n)$ — некоторый порядок, который соответствует макси-коду в G, а $\alpha_2 = (v_1, \dots, v_n)$ — порядок, соответствующий макси-коду в H. Поскольку номер $N(\overline{u})$ — натуральное число, то $\alpha_1(N(\overline{u}))$ задаёт некоторую вершину $u^\star = \alpha_1(N(\overline{u}))$. Эта вершина $u^\star$ обладает двумя свойствами: $u \sim u^\star$ и среди всех симметричных по отношению к u вершинам $u^\star$ имеет наименьший индекс в α_1. Если допустим, что $N(\overline{u}) \neq N(\overline{v})$, то получим $\phi(u^\star) \nsim \phi(u)$. В результате получаем противоречие: $\phi(u^\star) \nsim \phi(u)$ и $u \sim u^\star$. Таким образом, у всех изоморфных вершин ($\phi(u) = v$) номера классов автоморфизма обязательно совпадают ($N(\overline{u}) = N(\overline{v})$). Пользуясь утверждением 2 и учитывая определение для индексов $I(\overline{v})$, получаем, что $I(\overline{u}) = I(\overline{v})$.

Пусть $G \cong H$ и совпали индексы $I(\overline{u}) = I(\overline{v})$. Поскольку $G \cong H$, то существует некоторый изоморфизм $\phi_0 : V(G) \to V(H)$. Для образа $v^* = \phi_0(u)$ по первой части данного утверждения получим $I(\overline{v^*}) = I(\overline{u}) = I(\overline{v})$. Из этого можно заключить, что вершины v, v^* автоморфны. Обозначим автоморфизм, переводящий эти вершины одна в другую, через ψ: $\psi(v^*) = v$. Тогда искомый изоморфизм ϕ можно определить в виде композиции двух изоморфизмов $\phi = \phi_0 \circ \psi$. ■

Теорема 2. Если $G \cong H$, то для двух рёбер $(u_1, u_2) \in E(G)$ и $(v_1, v_2) \in E(H)$ равенство $I(\overline{u_1, u_2}) = I(\overline{v_1, v_2})$ выполняется тогда и только тогда, когда существует изоморфизм $\phi : V(G) \to V(H)$, такой, что $\phi(u_1) = v_1$ и $\phi(u_2) = v_2$, или $\phi(u_1) = v_2$ и $\phi(u_2) = v_1$.

Доказательство. Доказательство достаточности аналогично доказательству для вершин. Если допустить, что существует изоморфизм $\phi : V(G) \to V(H)$, такой, что $\phi(u_1) = v_1$ и $\phi(u_2) = v_2$, то совпадут номера $N(\overline{u_1, u_2}) = N(\overline{v_1, v_2})$. Из совпадения номеров для всех классов рёбер следует $I(\overline{u_1, u_2}) = I(\overline{v_1, v_2})$.

Необходимость также можно доказать по аналогии с теоремой 1. Пусть $G \cong H$ и $I(\overline{u_1, u_2}) = I(\overline{v_1, v_2})$. Для образов изоморфизма $v_1^* = \phi_0(u_1)$ и $v_2^* = \phi_0(u_2)$ по первой части данного утверждения получим $I(\overline{u_1, u_2}) = I(\overline{v_1, v_2}) = I(\overline{v_1^*, v_2^*})$. Из этого можно заключить, что рёбра автоморфны: $(v_1, v_2) \sim (v_1^*, v_2^*)$. Обозначим через ψ следующий автоморфизм: $\forall i\ (\psi(v_i^*) = v_i)$. Искомый изоморфизм — композиция $\phi = \phi_0 \circ \psi$. ■

Непосредственным следствием из теорем 1 и 2 является тот факт, что индексы $I(\overline{v})$ и $I(\overline{v_1, v_2})$ являются уникальными идентификаторами для классов автоморфизма. Таким образом, эти индексы можно использовать вместо самих вершин и рёбер при определении инварианта для класса изоморфных графов $[G]$. В частности, можно сначала представить граф G в виде строки символов с помощью алгоритма, похожего на алгоритм построения линейной нотации для молекулярных графов SMILES [3], а затем в данной строке заменить названия вершин v на их индексы $I(\overline{v})$ и получить инвариант для графа G.

Определение 9. *Симметрическая линейная нотация* $\mathfrak{L}(G)$ для связного графа G — это строка символов, которая определяется на основе следующего алгоритма.

Ш а г 0. Введём множества исключённых вершин V_E и временно заменённых вершин V_R. На начало работы алгоритма оба эти множества полагаются пустыми: $V_R = V_E = \varnothing$. Каждой новой вершине v, помещённой в множество замененных V_R, будем присваивать специальный код $\#m$, где m — число элементов $|V_R|$ после добавления вершины v. Признаком окончания работы алгоритма является помещение всех вершин $V(G)$ графа G в множество исключённых вершин V_E.

Ш а г 1. Выбираем любую вершину v_1 первого индекса ($I(\overline{v}_1) = 1$) и записываем её в нотацию $\mathfrak{L}(G)$, добавляя к ней открывающую скобку первого номера: $\mathfrak{L}(G) = v_1 \Big[_1 \ldots$ При этом помещаем v_1 в множество V_R и присваиваем ей первый специальный код $\#1$.

Ш а г 2 (рекурсивный). Пусть линейная нотация доведена до некоторой вершины v_k. При этом часть из рассмотренных вершин $v_1, \ldots, v_k$ помещена в множество исключённых V_E, а другая часть — в множество замененных V_R. Вместо заменённых вершин можем использовать коды $\#1, \ldots, \#m$ (код $\#m$ соответствует v_k). Среди всех вершин из окружения v_k, которые не были удалены или заменены на предыдущих этапах, выбираем любую v_{k+1}, которая удовлетворяет следующим условиям:

1) наибольший приоритет у вершины с минимальным индексом $I(\overline{v_{k+1}})$;
2) если у двух вершин индексы совпали ($I(\overline{v_{k+1}}) = I(\overline{v^*_{k+1}})$), то предпочтение отдаётся вершине v_{k+1}, у которой меньше индекс ребра: $I(\overline{v_k, v_{k+1}}) < I(\overline{v_k, v^*_{k+1}})$;
3) при совпадении индексов вершин и рёбер у v_{k+1} и v^*_{k+1} предпочтение отдаётся вершине, которая в графе G ближе к вершине кода $\#1$ (то есть может быть соединена с ней путём из меньшего числа вершин). Если эти расстояния совпадают для вершин v_{k+1} и v^*_{k+1}, то вершине, которая ближе к вершине кода $\#2$, и так далее до $\#(m-1)$;
4) если вершины v_{k+1} и v^*_{k+1} из окружения v_k не удалось различить с помощью условий 1–3 по индексам вершин, рёбер и расстояниям до вершин кодов $\#1, \ldots, \#m$, то выбирается любая из них.

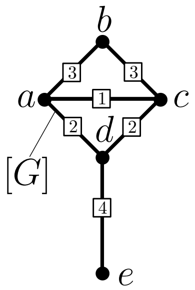
Если с помощью условий 1–4 удалось найти вершину v_{k+1}, то добавляем её в линейную нотацию $\mathfrak{L}(G) = \ldots \xrightarrow{i_k} v_k \Big[_m \ldots ; \xrightarrow{i_{k+1}} v_{k+1} \Big[_{m+1} \ldots$ и в множество V_R. После этого переходим к рекурсивному шагу 3 для v_{k+1}. Если вершину v_{k+1} не удалось найти, то удаляем v_k из множества V_R и переводим её в V_E, полностью исключая из рассмотрения. Затем закрываем скобку уровня m в линейной нотации $\mathfrak{L}(G)$ и повторяем шаг 2 для той вершины v_j, которая на текущий момент соответствует коду $\#(m-1)$.

Ш а г 3 (рекурсивный). Пусть линейная нотация доведена до некоторой вершины v_k. В первую очередь проверяем, нет ли в графе ребра, которое соединяет вершину v_k и вершину кода $\#1$. Если такое ребро имеется и оно до сих пор *не было учтено в нотации*, то добавляем запись: $\mathfrak{L}(G) = \ldots \xrightarrow{i_k} v_k \Big[_m \xrightarrow{i} \#1; \ldots$, где $i = I(\overline{v, v_k})$. Повторяем процесс для кода $\#2$ и так далее до последнего кода $\#(m-1)$. По завершении переходим к рекурсивному шагу 2 для вершины v_k.

Определение 10. *Линейной нотацией класса изоморфных графов* $[G]$ назовём строку $I[G]$, которая получается из любой произвольной симметрической линейной нотации $\mathfrak{L}(G)$ путём замены всех вершин v на индексы классов вершин $I(\overline{v})$.

Пример 1. Рассмотрим работу алгоритма для построения $\mathfrak{L}(G)$ и перехода к линейной нотации $I[G]$ с пошаговым выводом множеств V_R и V_E (рис. 2).

Согласно определению 9, одному графу G может соответствовать несколько различных симметрических нотаций $\mathfrak{L}(G)$. Для этого достаточно, чтобы у графа G хотя



$I(a) = 1 \;\; I(c) = 1$

$I(d) = 2 \;\; I(b) = 3$

$I(e) = 4$

Step	Symmetric Linear Notation	V_R	V_E
1	$a[_1$	(a)	$\varnothing$
2	$a[_1\!\to c[_2$	(a, c)	$\varnothing$
3	$a[_1\!\to c[_2\!\to d[_3$	(a, c, d)	$\varnothing$
4	$a[_1\!\to c[_2\!\to d[_3\!\to\#1$	(a, c, d)	$\varnothing$
5	$a[_1\!\to c[_2\!\to d[_3\!\to\#1;\!\to e[$	(a, c, d, e)	$\varnothing$
6	$a[_1\!\to c[_2\!\to d[_3\!\to\#1;\!\to e[\,]$	(a, c, d)	e
7	$a[_1\!\to c[_2\!\to d[_3\!\to\#1;\!\to e[\,]]$	(a, c)	e, d
8	$a[_1\!\to c[_2\!\to d[_3\!\to\#1;\!\to e[\,]];\!\to b[_3$	(a, c, b)	e, d
9	$a[_1\!\to c[_2\!\to d[_3\!\to\#1;\!\to e[\,]];\!\to b[_3\!\to\#1]$	(a, c)	e, d, b
10	$a[_1\!\to c[_2\!\to d[_3\!\to\#1;\!\to e[\,]];\!\to b[_3\!\to\#1]]]$	$\varnothing$	e, d, b, c, a

$$I[G] = 1[_1 \xrightarrow{1} 1[_2 \xrightarrow{2} 2[_3 \xrightarrow{2} \#1;\; \xrightarrow{4} 4[\,]];\; \xrightarrow{3} 3[_3 \xrightarrow{3} \#1]]]$$

Рис. 2. Пример построения линейной нотации $I[G]$

бы две вершины были автоморфными. Например, если в графе G есть две вершины первого индекса, то нотацию $\mathfrak{L}(G)$ можно начать с любой из них. Таким образом, необходимо доказать, что линейная нотация $I[G]$ определена корректно и не зависит от выбора $\mathfrak{L}(G)$.

Утверждение 4. Линейная нотация $I[G]$ определяется однозначным образом для любого конечного связного графа G.

Доказательство. Проведём доказательство однозначности $I[G]$ по индукции.

Базис индукции. Рассмотрим шаг 1 алгоритма определения $\mathfrak{L}(G)$. После замены вершины v_1 на её индекс $I(\overline{v_1}) = 1$ имеем одну и ту же начальную строчку $I[G] = 1\Big[_1 \ldots$ для любых $\mathfrak{L}(G)$.

Индуктивный переход. Допустим, что неоднозначность в определении $I[G]$ не возникла до начала отдельной итерации рекурсивного шага 2 алгоритма определения $\mathfrak{L}(G)$. После замены вершин v_k и v_{k+1} на их индексы $I(\overline{v_k})$ и $I(\overline{v_{k+1}})$ получим одну и ту же строчку $I[G] = \ldots \xrightarrow{i_k} I(\overline{v_k})\Big[_m \ldots ; \xrightarrow{i_{k+1}} I(\overline{v_{k+1}})\Big[_{m+1} \ldots$ для любых $\mathfrak{L}(G)$.

Допустим, что неоднозначность в определении $I[G]$ не возникла до начала отдельной итерации рекурсивного шага 3 алгоритма определения $\mathfrak{L}(G)$. Если предположим, что коды $\#j$, с которыми связана v_k, различаются для двух разных $\mathfrak{L}(G)$, то получим противоречие с условием 3 выбора новых вершин на рекурсивном шаге 2. Таким образом, переход к $I[G] = \ldots \xrightarrow{i_k} I(\overline{v_k})\Big[_m \xrightarrow{i} \#1; \ldots$ на шаге 3 также однозначный. ■

Замечание 1. Отметим, что для линейной нотации $I[G]$ можно ввести понятия абстрактных вершин и абстрактных рёбер, связав их с соответствующими индексами автоморфизма. При этом у $I[G]$ может быть несколько экземпляров абстрактных вершин, которые соответствуют одному и тому же индексу j, и то же самое для абстрактных рёбер.

Определение 11. *Раскраской для линейной нотации* $I[G]$ отдельной абстрактной *вершины* j *в цвет* α будем называть такую строку $I^*_{\langle j,\alpha\rangle}[G]$, которая получается из линейной нотации $I[G]$ путём замены первого вхождения j на пару $\langle j, \alpha\rangle$. Раскраску для большего количества вершин и большего количества цветов определим индуктивно как последовательную раскраску графа по одной отдельной абстрактной вершине.

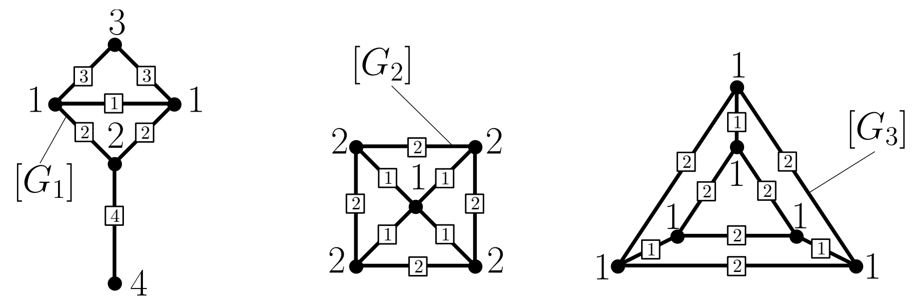
При этом достаточно потребовать, чтобы на каждом шаге раскраски выбирались только те вершины, которые не были раскрашены на предыдущих.

Теорема 3. Линейная нотация класса изоморфных графов $I[G]$ является полным инвариантом для любого конечного связного графа G.

Доказательство. Докажем необходимость, то есть что из изоморфизма графов $G_1 \cong G_2$ следует, что $I[G_1] = I[G_2]$. Из теорем 1 и 2 получаем, что у графов G_1 и G_2 один и тот же набор индексов для вершин и рёбер. Как следствие, с учётом утверждения 4, нотации $\mathfrak{L}(G_1)$ и $\mathfrak{L}(G_2)$ отличаются только названиями вершин. Таким образом, $I[G_1] = I[G_2]$.

Докажем достаточность, то есть что из совпадения линейных нотаций $I[G_1] = I[G_2]$ следует изоморфизм графов $G_1 \cong G_2$. Для этого с помощью раскраски названиями V_1 вершин $G_1 = (V_1, E_1)$ перейдём от $I[G_1]$ к $\mathfrak{L}(G_1)$. При этом остановимся на том варианте раскраски $\mathfrak{L}(G_1)$, который однозначно соответствует графу G_1 по связям E_1 между вершинами, потенциально перебрав все $|V_1|!$ вариантов. Затем повторим аналогичную процедуру для G_2 и получим $\mathfrak{L}(G_2)$. Построим изоморфизм в явном виде, отобразив все подряд идущие вершины из $\mathfrak{L}(G_1)$ в соответствующие им по порядку вершины из $\mathfrak{L}(G_2)$. Если допустить для данного отображения, что условие изоморфизма не выполняется, то немедленно получим противоречие. Действительно, нарушение условия изоморфизма означает, что коды $\#1, \ldots, \#m$ стоят на разных позициях в $\mathfrak{L}(G_1)$ и $\mathfrak{L}(G_2)$, что невозможно ввиду тождества $I[G_1] = I[G_2]$ и однозначного определения $I[G]$ на основе $\mathfrak{L}(G)$. ■

Для демонстрации изложенных результатов на рис. 3 приводятся примеры $I[G]$ для трёх графов G_1, G_2 и G_3, чьи вершины помечены индексами $I(\overline{v})$, а рёбра — $I(\overline{u,v})$.



$$I[G_1] = 1[_1 \xrightarrow{1} 1[_2 \xrightarrow{2} 2[_3 \xrightarrow{2} \#1; \xrightarrow{4} 4[\,]]; \xrightarrow{3} 3[_3 \xrightarrow{3} \#1]]]$$

$$I[G_2] = 1[_1 \xrightarrow{1} 2[_2 \xrightarrow{2} 2[_3 \xrightarrow{1} \#1; \xrightarrow{2} 2[_4 \xrightarrow{1} \#1; \xrightarrow{2} 2[_5 \xrightarrow{1} \#1; \xrightarrow{2} \#2]]]]$$

$$I[G_3] = 1[_1 \xrightarrow{1} 1[_2 \xrightarrow{2} 1[_3 \xrightarrow{1} 1[_4 \xrightarrow{2} \#1; \xrightarrow{2} 1[_5 \xrightarrow{2} \#1; \xrightarrow{1} 1[_6 \xrightarrow{2} \#2; \xrightarrow{2} \#3]]]]]$$

Рис. 3. Примеры линейных нотаций $I[G]$ для классов изоморфных графов $[G_1]$, $[G_2]$ и $[G_3]$

2. Практическое приложение линейных нотаций

Классический способ применения инвариантов графов сводится к вычислению и сравнению инвариантов каждый раз при проверке двух графов на изоморфизм.

Достаточно очевидно, что подобная схема совершенно неприменима для полных инвариантов, так как задача их построения относится к классу NP. В частности, для нотаций $I[G]$ имеем следующие оценки (параметр $n = |V(G)|$):

1) переход от G к $I[G]$ — задача эквивалентна задаче об изоморфизме графов;
2) проверка на совпадение для $I[G]$ — задача сложности $\mathrm{O}(n^2)$;
3) переход от $I[G]$ к G на основе раскраски — задача сложности $\mathrm{O}(n^2)$.

Общая идея заключается в том, чтобы *хранить в памяти компьютера* $I[G]$ вместо обычного графа G. При таком подходе упрощается проверка на изоморфизм, поскольку не требуется предварительно вычислять инвариант $I[G]$, а процедура сводится к полиномиальной задаче проверки на совпадение $I[G_1] = I[G_2]$. Для графов с небольшим числом вершин можно использовать базовый алгоритм перехода от G к $I[G]$, несмотря на его принадлежность к классу NP, поскольку потребуется применить его один раз перед сохранением линейной нотации $I[G]$. Однако для графов с большим числом вершин такой подход непрактичен. Фактически, построение линейных нотаций $I[G]$ для больших графов требует определения стандартных классов графов непосредственно на основе $I[G]$, а также введения некоторого набора операций полиномиальной сложности на $I[G]$ для получения более сложных классов изоморфных графов.

Рассмотрим, как можно с помощью линейной нотации $I[G]$ описать в общем виде следующие стандартные классы графов: пути P_n, циклы C_n, полные графы K_n и полные двудольные графы $K_{n,m}$ [2, 4].

Пример 2. Для путей P_n рассмотрим два случая: когда n чётное ($n = 2k$) и нечётное ($n = 2k-1$):

$$I[P_{2k}] = 1\Big[_1 \xrightarrow{1} 1\Big[_2 \xrightarrow{2} 2\Big[_3 \dots \xrightarrow{n} n[\,]\Big] \dots\Big]; \xrightarrow{2} 2\Big[_2 \xrightarrow{3} 3\Big[_3 \dots \xrightarrow{n} n[\,]\Big] \dots\Big]\Big],$$
$$I[P_{2k-1}] = 1\Big[_1 \xrightarrow{1} 2\Big[_2 \xrightarrow{2} 3\Big[_3 \dots \xrightarrow{n-1} n[\,]\Big] \dots\Big]; \xrightarrow{1} 2\Big[_2 \xrightarrow{2} 3\Big[_3 \dots \xrightarrow{n-1} n[\,]\Big] \dots\Big]\Big].$$

Пример 3. Линейная нотация для циклов C_n задана однозначно:

$$I[C_n] = 1\Big[_1 \xrightarrow{1} 1\Big[_2 \dots \xrightarrow{1} 1\Big[_n \xrightarrow{1} \#1\Big] \dots\Big]\Big].$$

Пример 4. Линейная нотация для полного графа K_n также задана однозначно:

$$I[K_n] = 1\Big[_1 \xrightarrow{1} 1\Big[_2 \xrightarrow{1} 1\Big[_3 \xrightarrow{1} \#1; \xrightarrow{1} 1\Big[_4 \dots \xrightarrow{1} 1\Big[_n \xrightarrow{1} \#1; \xrightarrow{1} \#2; \dots; \xrightarrow{1} \#n-2\Big] \dots\Big]\Big]\Big]\Big].$$

Пример 5. Для двудольных графов $K_{n,m}$ рассмотрим два случая: $n < m$ и $n = m$:

$$I[K_{n,m}] = 1\Big[_1 \dots \xrightarrow{1} 1\Big[_{2m-1} \xrightarrow{1} \#2; \dots; \xrightarrow{1} \#2m-4; \xrightarrow{1} 2\Big[_{2m} \xrightarrow{1} \#1; \xrightarrow{1} \#3; \dots; \xrightarrow{1} \#2m-3\Big];$$
$$\dots; \xrightarrow{1} 2\Big[_{2m} \xrightarrow{1} \#1; \xrightarrow{1} \#3; \dots; \xrightarrow{1} \#2m-3\Big]\Big] \dots\Big],$$
$$I[K_{n,n}] = 1\Big[_1 \dots \xrightarrow{1} 1\Big[_{2n-1} \xrightarrow{1} \#2; \dots; \xrightarrow{1} \#2n-4; \xrightarrow{1} 1\Big[_{2n} \xrightarrow{1} \#1; \xrightarrow{1} \#3; \dots; \xrightarrow{1} \#2n-3\Big]\Big] \dots\Big].$$

Замечание 2. Как видно из примеров для $I[P_{2k-1}]$ и $I[K_{n,m}]$, линейную нотацию $I[G]$ легко можно адаптировать для сжатия данных, если использовать специальную нотацию $S \times n$ для подстрок, которые повторяются n раз подряд. В случае с $I[P_{2k-1}]$ получим $I^*[P_{2k-1}] = 1\Big[_1 \xrightarrow{1} 2\Big[_2 \xrightarrow{2} 3\Big[_3 \dots \xrightarrow{n-1} n[\,]\Big] \dots\Big]; \times 2\Big]$.

Перейдём к рассмотрению элементарных операций на графах. При этом произвольную операцию на классах изоморфных графов будем описывать в форме алгоритма строковой обработки для линейных нотаций $I[G]$.

Пример 6. Введём операцию удлинения цепочек, определив результирующий граф $\tilde{G}$ как полученный из исходного графа G путём присоединения ко всем вершинам первой степени ровно по одной новой вершине (реализацию операции см. на рис. 4).

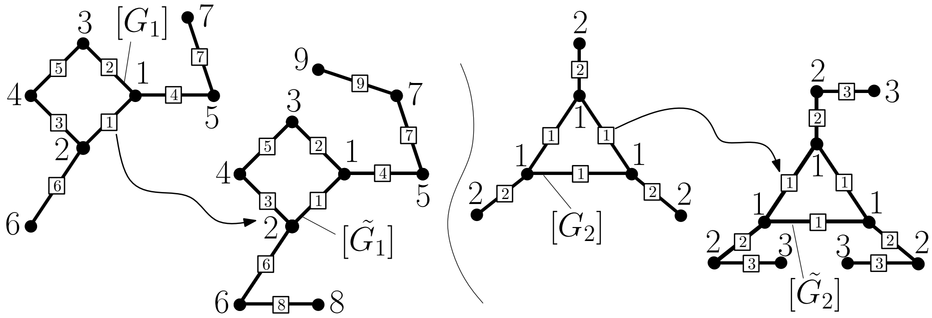


Рис. 4. Пример использования операции удлинения цепочек

Алгоритм построения линейной нотации $I[\tilde{G}]$ для результирующего графа на основе исходной нотации $I[G]$ можно свести к следующим трём шагам строковой обработки.

Ш а г 1. Проводим поиск по строке $I[G]$ вершин, которые соответствуют шаблону $(\ldots \xrightarrow{j} k[\,]\ldots)$ для вершины первой степени. Среди всех этих вершин вычисляем минимальный индекс автоморфизма k и максимальный $k+m$, которые соответствуют данному шаблону.

Ш а г 2. Вводим новые индексы классов $k+m+1, k+m+2, \ldots, k+2m$, которые будут соответствовать новым вершинам графа $\tilde{G}$.

Ш а г 3. Получаем $I[\tilde{G}]$ из исходной линейной нотации $I[G]$ путём замены подстрок по шаблону: строку $\Big(\ldots \xrightarrow{j+i} k{+}i[\,]\ldots\Big)$ меняем на $\Big(\ldots \xrightarrow{j+i} k{+}i\Big[\xrightarrow{j+i+m} k{+}m{+}i[\,]\Big]\ldots\Big)$.

Пример 7. Рассмотрим операцию присоединения вершин первой степени к вершинам первого класса автоморфизма (реализацию операции для конкретных графов см. на рис. 5). Формально при отработке данной операции мы к каждой вершине v с индексом автоморфизма $I(\overline{v}) = 1$ присоединяем ровно одну вершину v^* степени 1.

Алгоритм построения линейной нотации $I[G^*]$ для результирующего графа на основе исходной нотации $I[G]$ можно свести к следующим трём шагам.

Ш а г 1. Проводим поиск по строке $I[G]$ вершин, которые соответствуют шаблону $(\ldots \xrightarrow{j} k[\,]\ldots)$ и при этом связаны ребром с вершиной первого класса. Если таких вершин нет, то переходим к шагу 2. В противном случае добавляем в окружение каждой вершины первого класса одну дополнительную вершину класса k, осуществив строковую замену по шаблону: строки $\Big(1\Big[\ldots \xrightarrow{j} k[\,]\Big]\ldots\Big)$ меняем на $\Big(1\Big[\ldots \xrightarrow{j} k[\,]; \xrightarrow{j} k[\,]\Big]\ldots\Big)$; выход.

Ш а г 2. Ищем вершины первой степени $(\ldots \xrightarrow{j} k[\,]\ldots)$. Среди всех этих вершин вычисляем минимальный индекс автоморфизма k и максимальный $k+m$.

Ш а г 3. Производим строковую замену всех индексов вершин от k до $k+m$ по следующему шаблону: строки $\Big(\ldots \xrightarrow{j+i} k{+}i[\,]\ldots\Big)$ меняем на $\Big(\ldots \xrightarrow{j+i+1} k{+}i{+}1[\,]\ldots\Big)$. После этого добавляем в окружение каждой вершины класса 1 по одной новой вершине класса k в соответствии с шаблоном: $\Big(1\Big[\ldots \xrightarrow{j} k[\,]\Big]\ldots\Big)$.

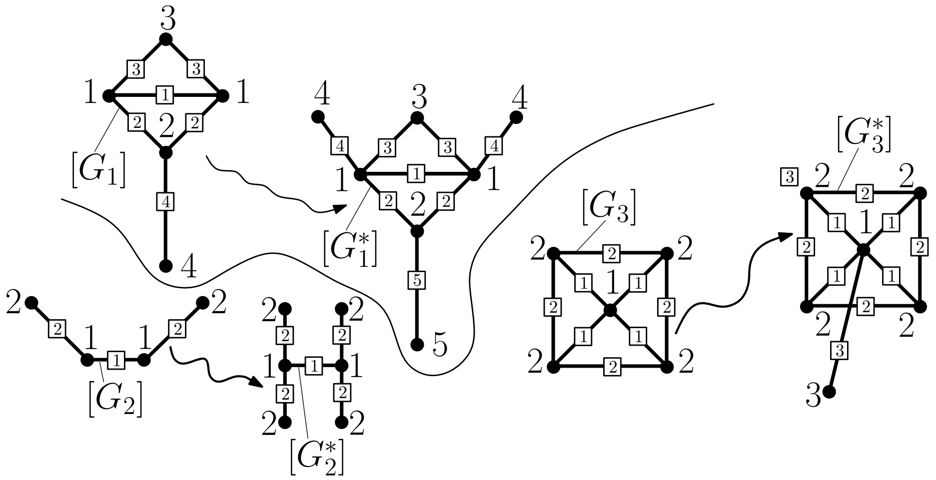
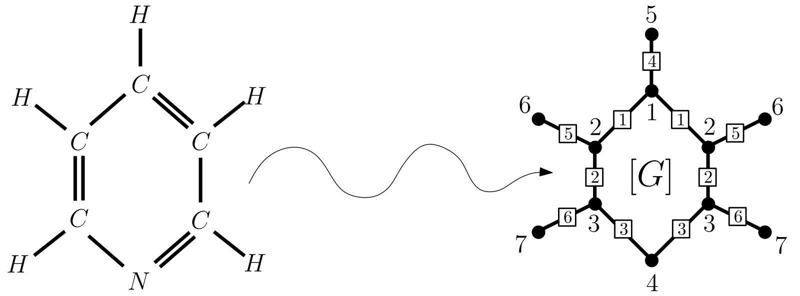


Рис. 5. Пример использования операции присоединения новой вершины к вершинам первого класса

Замечание 3. Для всех рассмотренных операций, а также для описания стандартных классов графов опустим доказательство корректности. При желании эти доказательства можно легко получить по аналогии с уже рассмотренными.

В определении 11 введено понятие раскраски для отдельных абстрактных вершин линейной нотации $I[G]$. По аналогии с этим определением можно ввести раскраску и для абстрактных рёбер на $I[G]$. Если у исходного графа G было n вершин, то, раскрасив итерационно все абстрактные вершины $I[G]$ в n разных цветов, получим $\mathfrak{L}(G^*)$ и восстановим исходный граф с точностью до изоморфизма $G^* \cong G$. Важно отметить, что подобная операция перехода от $I[G]$ к обычному графу $G^* \cong G$ в предельном случае потребует восстановить связи между каждыми двумя вершинами графа и, как следствие, имеет максимальную сложность $\mathrm{O}(n^2)$. Помимо восстановления обычного графа на основе $I[G]$, алгоритм раскраски можно использовать для построения молекулярных графов [5], если в качестве красок для вершин использовать названия химических элементов H, O, C, N, а в качестве красок для рёбер — типы связей.

Пример 8. На рис. 6 представлен молекулярный граф пиридина, а также маркированная линейная нотация, которая получена для этого графа с помощью алгоритма итерационной раскраски для вершин и рёбер (двойные связи показаны двойной стрелкой $\Rightarrow$).



$$I^*[G] = \langle 1,C\rangle \big[_1 \xrightarrow{1} \langle 2,C\rangle \big[_2 \overset{2}{\Rightarrow} \langle 3,C\rangle \big[_3 \xrightarrow{3} \langle 4,N\rangle \big[_4 \overset{3}{\Rightarrow} \langle 3,C\rangle \big[_5 \xrightarrow{2} \langle 2,C\rangle \big[_6 \overset{1}{\Rightarrow} \#1; \xrightarrow{5} \langle 6,H\rangle \big]; \xrightarrow{6} \langle 7,H\rangle \big]\big]; \xrightarrow{6} \langle 7,H\rangle \big]; \xrightarrow{5} \langle 6,H\rangle \big]; \xrightarrow{4} \langle 5,H\rangle \big]$$

Рис. 6. Пример задания молекулярного графа в виде маркированной линейной нотации

При описании молекулярных графов на основе маркированных линейных нотаций $I^*[G]$ получаем три очевидных преимущества:

— проверка на изоморфизм двух графов имеет сложность $\mathrm{O}(n^2)$;

— все возможные симметрии молекулы представлены в удобной форме через индексы классов автоморфизма вершин и рёбер;

— задание реакций между молекулами в виде операций на нотациях $I[G]$ сводится к процедурам строковой обработки, что упрощает их реализацию.

Алгоритм раскраски для $I[G]$, помимо уже рассмотренных приложений, может быть использован для построения путей на графе. В этом случае раскраска проводится натуральными числами $1, 2, \ldots, n$, а две соседние вершины, получившие порядковые номера i, $i+1$, должны быть соединены ребром в $I[G]$. Отметим, что отдельный интерес представляет поиск на линейной нотации $I[G]$ гамильтонова пути [6]. На основании перебора графов с числом вершин $|V| \leqslant 7$ была выдвинута следующая гипотеза относительно существования гамильтонова пути.

Гипотеза 1. Для того чтобы на $I[G]$ существовал гамильтонов путь, необходимо и достаточно, чтобы $I[G]$ удовлетворял одному из двух шаблонов: либо $I[G] = 1\Big[_1 \xrightarrow{j} k\Big[_2 \ldots\Big]\Big]$, либо $I[G] = 1\Big[_1 \xrightarrow{j} k\Big[_2 \ldots\Big]; \xrightarrow{l} m\Big[_2 \ldots\Big]\Big]$, где внутри скобки $\Big[_2 \ldots\Big]$ число вложенных скобок в точности равно числу абстрактных вершин.

Заключение

Таким образом, можно утверждать, что линейная нотация $I[G]$ способна стать удобной альтернативой для классических графов G в приложении к целому ряду практических задач. Представляется актуальным продолжение исследований в данном направлении и построение новых алгоритмов на $I[G]$, в том числе и алгоритмов для реализации операций на графах. Более того, если удастся построить базис (набор операций, с помощью которых можно получить любой конечный граф, применяя их суперпозицию к некоторым простейшим графам) операций на $I[G]$ полиномиальной сложности, то нотацию $I[G]$ можно будет использовать на практике для хранения больших графов.

Сложность хранения в памяти $I[G]$ асимптотически равна $\mathrm{O}(n^2)$, где n — число вершин графа. Можно повысить коэффициент сжатия данных для $I[G]$, если воспользоваться приёмом замечания 1, а также если отбросить индексы m для всех скобок $\Big[_m \ldots\Big]$ в записи $I[G]$ (данные индексы введены в определении $\mathfrak{L}(G)$ только для более наглядного представления нотации).

Помимо уже рассмотренных приложений для линейных нотаций $I[G]$, можно попробовать построить оптимизированный алгоритм для полного перебора графов из n вершин с отбрасыванием всех изоморфных копий. Фактически, если перебирать $I[G]$, а не матрицы смежности A графов G, то можно добиться многократного прироста производительности такого алгоритма, так как число классов изоморфных графов из n вершин существенно меньше числа таблиц из нулей и единиц размера $n \times n$ [7].

Как отмечено в работе [1], линейную нотацию $I[G]$ можно достаточно легко обобщить на случай упорядоченных графов, а также на гиперграфы [8]. С другой стороны, для классов автоморфизма вершин $\overline{v}$ и рёбер $(\overline{u,v})$ можно ввести альтернативные индексы, если выбрать вместо макси-кода другой полный численный инвариант матрицы смежности. В частности, можно задействовать мини-код вместо макси-кода. С практической точки зрения такие альтернативные конструкции будут целесообразны, если удастся показать, что с их помощью упрощается определение некоторых алгоритмов на классах изоморфных графов.

ЛИТЕРАТУРА

1. *Назаров М. Н.* Альтернативные подходы к описанию классов изоморфных графов // Прикладная дискретная математика. 2014. №3. С. 86–97.
2. *Зыков А. А.* Основы теории графов. М.: Вузовская книга, 2004. 664 с.
3. *Weininger D., Weininger A., and Weininger J.* SMILES. 2. Algorithm for generation of unique SMILES notation // J. Chem. Inf. Comput. Sci. 1989. V. 29. No. 2. P. 97–101.
4. *Белов В. В., Воробьев Е. М., Шаталов В. Е.* Теория графов. М.: Высшая школа, 1976. 391 с.
5. *Brecher J.* Graphical representation of stereochemical configuration // Pure Appl. Chem. 2006. V. 78. No. 2. P. 1897–1970.
6. *Харари Ф.* Теория графов. М.: КомКнига, 2006. 296 с.
7. *Harary F. and Palmer E. M.* Graphical Enumeration. N.Y.: Academic Press, 1973. 240 p.
8. *Zykov A. A.* Hypergraphs // Russian Mathematical Surveys. 1974. V. 29. P. 89–154.

REFERENCES

1. *Nazarov M. N.* Al'ternativnye podkhody k opisaniyu klassov izomorfnykh grafov [Alternative approaches to the description of classes of isomorphic graphs]. Prikladnaya diskretnaya matematika, 2014, no. 3, pp. 86–97. (in Russian)
2. *Zykov A. A.* Osnovy teorii grafov [Basics of Graph Theory]. Moscow, Vuzovskaya Kniga Publ., 2004. 664 p. (in Russian)
3. *Weininger D., Weininger A., and Weininger J.* SMILES. 2. Algorithm for generation of unique SMILES notation. J. Chem. Inf. Comput. Sci., 1989, vol. 29, no. 2, pp. 97–101.
4. *Belov V. V., Vorob'ev E. M., Shatalov V. E.* Teoriya grafov [Graph Theory]. Moscow, Vysshaya Shkola Publ., 1976. 391 p. (in Russian)
5. *Brecher J.* Graphical representation of stereochemical configuration. Pure Appl. Chem., 2006, vol. 78, no. 2, pp. 1897–1970.
6. *Kharari F.* Teoriya grafov [Graph Theory]. Moscow, KomKniga Publ., 2006. 296 p. (in Russian)
7. *Harary F. and Palmer E. M.* Graphical Enumeration. N.Y., Academic Press, 1973. 240 p.
8. *Zykov A. A.* Hypergraphs. Russian Mathematical Surveys, 1974, vol. 29, pp. 89–154.

УДК 519.17

О МАКСИМАЛЬНЫХ ВНЕШНЕПЛОСКИХ ГРАФАХ С ДВУМЯ СИМПЛИЦИАЛЬНЫМИ ВЕРШИНАМИ

Ю. Л. Носов

г. Липецк, Россия

Рассматриваются максимальные внешнеплоские графы (МВП-графы) с двумя симплициальными вершинами. Для графов этого класса получены: рекурсивная характеризация, формула для расчёта количества непомеченных графов и полный инвариант, отличающийся от известного полного инварианта произвольных МВП-графов. Описан полиномиальный алгоритм вычисления полного инварианта.

Ключевые слова: *максимальные внешнеплоские графы, 2-цепь, рекурсивная характеризация, непомеченные графы, полный инвариант.*

DOI 10.17223/20710410/29/8

MAXIMAL OUTERPLANE GRAPHS WITH TWO SIMPLICIAL VERTICES

Y. L. Nosov

Lipetsk, Russia

E-mail: yl.nosov@yandex.ru

The class of maximal outerplane graphs with two simplicial vertices is considered. The following results are obtained for graphs of this class: recursive characterization, a formula for computation of unlabeled graphs, a complete invariant that differs from the known complete invariant of arbitrary maximal outerplane graphs, and a polynomial algorithm for computation of the complete invariant.

Keywords: *maximal outerplanar graphs, 2-path, enumeration, unlabeled graphs, complete invariant.*

Введение

Одним из объектов изучения теории графов являются максимальные внешнеплоские графы (МВП-графы), которые впервые были исследованы Г. Чартрэндом и Ф. Харари в 1967 г. [1].

Внешнеплоским графом называется плоский граф, все вершины которого принадлежат внешней грани. Максимальным внешнеплоским графом называется такой внешнеплоский граф, который при добавлении хотя бы одного ребра перестаёт быть внешнеплоским [2]. Графы этого вида являются триангуляциями выпуклого многоугольника и относятся к классу хордальных графов [3].

Работы по исследованию МВП-графов достаточно хорошо представлены в математической литературе (см., например, [1 – 11]). Основные свойства МВП-графов описаны в [2]. Характеризация в терминах запрещённых подграфов дана в [1], а рекурсивная характеризация — в [4, 5]. Некоторые «конструктивные» свойства МВП-графов рассмотрены в [5 – 7], а условия возможности укладки МВП-графов на двух параллельных линиях — в [8]. В работе [9] получены производящие функции для помеченных и непомеченных МВП-графов и формула для числа непомеченных МВП-графов, использующая числа Каталана. Полный инвариант МВП-графов и полиномиальный алгоритм определения изоморфизма представлены в [10]. В работе [11] разработан полиномиальный алгоритм расчёта индекса Винера МВП-графов. Такой широкий интерес к изучению МВП-графов вызван тем, что МВП-графы входят в класс хордальных графов, для которых многие NP-трудные задачи теории графов полиномиально разрешимы.

Структура МВП-графов определяется количеством вершин степени 2, являющихся симплициальными (т. е. такими, что их окрестность порождает клику). Все МВП-графы, имеющие более двух симплициальных вершин, суть простые 2-деревья [6], а МВП-графы с двумя симплициальными вершинами — это 2-цепи [12]. Далее эти графы будем называть МВП-графами класса 2-цепь.

МВП-графы класса 2-цепь могут быть использованы в качестве графовой модели потоковых сетей с источником и стоком, сетевых графиков, электрических двухполюсников, электрических сетей с двумя источниками и с поперечными связями [13, 14]. Однако несмотря на многочисленные исследования по МВП-графам, пока не получено полного описания МВП-графов класса 2-цепь. Это определяет актуальность исследования графов данного класса.

В работе полностью описаны МВП-графы класса 2-цепь. В п. 1 приводятся типы МВП-графов класса 2-цепь, их основные свойства и условия укладки графов на двух параллельных линиях. Рекурсивная характеризация, полный инвариант и формула для числа непомеченных МВП-графов класса 2-цепь представлены в п. 2, 3 и 4. В п. 3 описан полиномиальный алгоритм вычисления их полных инвариантов.

1. Типы МВП-графов класса 2-цепь и их свойства

Введём необходимые обозначения и определения. Везде далее через $G(V, E)$ обозначается конечный неориентированный связный граф G без петель и кратных рёбер с множеством вершин V и множеством рёбер E. Число вершин и рёбер графа обозначается через $n(G) = |V|$ и $m(G) = |E|$ соответственно. Все основные термины и обозначения приводятся в соответствии с [3].

Полный граф из k вершин есть *k-дерево*; k-дерево с $n+1$ вершинами получается из k-дерева с n вершинами путём добавления в него вершины v и k рёбер, соединяющих вершину v со всеми вершинами некоторой клики размера k (k-клики) [6].

Простым k-деревом называется k-дерево, в котором любая его k-клика является подграфом не более двух $(k+1)$-клик. Простое k-дерево с двумя симплициальными вершинами есть *2-цепь* [12].

Внутренним графом $\mathrm{Int}(G)$ МВП-графа G называется связный граф, образующийся после удаления всех внешних рёбер графа G [7]. *Граф типа «гусеница»* (или просто *гусеница*) — это дерево, в котором удаление всех вершин степени 1 приводит к образованию простой цепи [15]. Простая цепь, остающаяся после удаления из гусеницы всех вершин степени 1, называется *спином*. Вершины степени 1 называются *концевыми вершинами* или *листьями*.

Двудольный граф $G(A \cup B, E)$ называется *BA-планарным*, если его можно уложить на плоскости так, что все вершины размещены на двух параллельных линиях (вершины из множества A на одной линии, а вершины из множества B — на другой) без пересечений рёбер и каждое ребро является отрезком прямой. Уже уложенный таким образом граф называется *плоской BA-укладкой* графа $G(A \cup B, E)$ или *BA-плоским графом* [8].

Граф G называется LL*-планарным*, если его можно уложить на плоскости так, что все вершины размещены на двух параллельных линиях без пересечений рёбер и каждое ребро является отрезком прямой. Уже уложенный таким образом граф называется *плоской* LL*-укладкой* графа $G(V, E)$ или LL*-плоским графом* [8].

Гамильтоновой степенной последовательностью МВП-графа называется список $D = (d_1, d_2, \ldots, d_n)$ степеней вершин графа, упорядоченный в соответствии с порядком следования вершин в гамильтоновом цикле $H = (v_1, v_2, \ldots, v_n, v_1)$. В работе [10] показано, что гамильтонова степенная последовательность МВП-графа является его полным инвариантом.

Основные свойства МВП-графов описаны в [2, 4–6, 8–11]. Рассмотрим основные свойства МВП-графов класса 2-цепь.

Пусть $G(V, E)$ — МВП-граф класса 2-цепь с n вершинами. Тогда граф G имеет две несмежные симплициальные вершины s_1, s_2, которые разделяют гамильтонов цикл C_n графа на две цепи $P_1 = (s_1, x_1, x_2, \ldots, x_a, s_2)$ и $P_2 = (s_1, y_1, y_2, \ldots, y_b, s_2)$, для которых

$$\begin{gathered} P_1 \cup P_2 = C_n, \quad P_1 \cap P_2 = \{s_1, s_2\}, \\ X = \{x_1, x_2, \ldots, x_a\}, \quad Y = \{y_1, y_2, \ldots, y_b\}. \end{gathered} \tag{1}$$

Таким образом, множество вершин V состоит из подмножества V' ($|V'| = n - 2$), разделённого на две доли X и Y, и симплициальных вершин s_1, s_2:

$$V = V' \cup \{s_1, s_2\}, \quad V' = X \cup Y, \quad X \cap Y = \varnothing. \tag{2}$$

Множество рёбер E состоит из подмножеств внешних рёбер E_O ($|E_O| = n$) и внутренних рёбер E_I ($|E_I| = n - 3$), причём любое внутреннее ребро $\{u, w\}$ соединяет пару вершин u и w из множеств X и Y:

$$E = E_O \cup E_I, \quad E_O = \{e_i : e_i \in C_n\}, \quad E_I = \{e_i : e_i \notin C_n\}. \tag{3}$$

Этим условиям удовлетворяют три типа МВП-графов класса 2-цепь. Графы первого типа будем называть графами типа «веер», второго типа — графами типа «лестница», третьего типа — графами типа «цепь». Образцы графов этих трёх типов приведены на рис. 1 (внутренние рёбра изображены утолщёнными линиями, а внешние рёбра — тонкими).

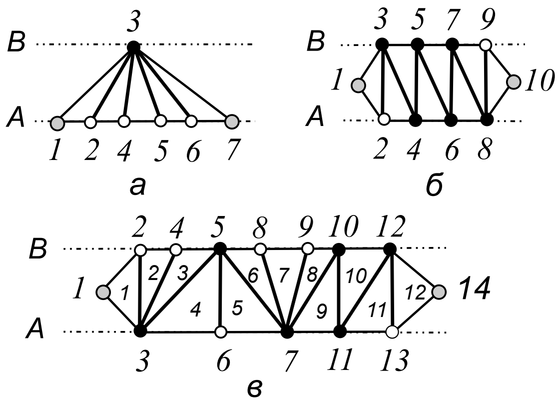


Рис. 1. МВП-графы класса 2-цепь: *а* — типа «веер»; *б* — типа «лестница»; *в* — типа «цепь»

Нетрудно заметить, что при удалении внешних ребер МВП-графа получается внутренний граф и несколько изолированных вершин, каждая из которых в исходном МВП-графе была симплициальной вершиной.

Таким образом, внутренний граф МВП-графа представляет собой граф, полученный удалением внешних рёбер с последующим удалением изолированных вершин. Используя принятые в выражениях (1)–(3) обозначения множеств вершин и рёбер, для внутреннего графа H некоторого МВП-графа G класса 2-цепь можно написать

$$\mathrm{Int}(G) = H(V', E_I) = H(V \setminus \{s_1, s_2\}, E \setminus E_O).$$

Утверждение 1 [7]. Если внутренний граф МВП-графа — дерево, то этот граф есть гусеница.

Следствие 1. Внутренний граф МВП-графа класса 2-цепь есть гусеница.

Доказательство следует из того, что для любого МВП-графа $G(V, E)$ класса 2-цепь его внутренний граф $H(V', E_I)$ является деревом.

Определение 1. В МВП-графе G класса 2-цепь вершины степени 3 будем называть терминальными вершинами, а вершины степени 4 и более — спинальными.

На рис. 1 терминальные вершины изображены простыми кружками, а спинальные — зачернёнными.

Замечание 1. Нетрудно заметить, что после удаления внешних рёбер МВП-графа G класса 2-цепь спинальные и терминальные вершины графа будут соответственно вершинами спина и концевыми вершинами внутреннего графа $\mathrm{Int}(G)$.

С учётом введённой терминологии имеем:

а) внутренний граф n-вершинного МВП-графа типа «веер» есть звезда S_{n-2} с $(n-2)$ вершинами (вырожденная гусеница с одной вершиной спина);
б) внутренний граф n-вершинного МВП-графа типа «лестница» есть простая цепь P_{n-2} с $(n-2)$ вершинами (вырожденная гусеница с двумя вершинами степени 1);
в) внутренний граф n-вершинного МВП-графа типа «цепь» есть невырожденная гусеница.

Определим условия возможности плоской LL-укладки МВП-графов класса 2-цепь. Для этого используем результаты, полученные в [6, 8, 16].

Утверждение 2 [16]. Двудольный граф G имеет плоскую BA-укладку тогда и только тогда, когда G есть гусеница.

Утверждение 3 [8]. Двусвязный граф G имет плоскую LL-укладку тогда и только тогда, когда:

1) граф G есть внешнепланарный;
2) слабый двойственный граф $D^W(\widetilde{G})$ внешнеплоской укладки $\widetilde{G}$ графа G есть простая цепь.

Утверждение 4 [6]. Любой МВП-граф класса 2-цепь имеет слабый двойственный граф, изоморфный простой цепи.

Из утверждений 3 и 4 вытекает

Следствие 2. Любой МВП-граф класса 2-цепь имеет плоскую LL-укладку.

Пример плоской LL-укладки МВП графов класса 2-цепь представлен на рис. 1, *а*. Графы, изображённые на рис. 1, *б*,*в*, также имеют плоскую LL-укладку, для этого достаточно поместить их симплициальные вершины на параллельных линиях A и B.

Следствие 3. МВП-граф G класса 2-цепь является LL-плоским графом тогда и только тогда, когда его внутренний граф $\mathrm{Int}(G)$ является BA-плоским графом.

Анализируя всё вышеизложенное, можно сделать следующие выводы:

а) структура и вид (укладка) МВП-графа класса 2-цепь полностью определяются его внутренним графом;

б) любые два МВП-графа класса 2-цепь с одинаковым количеством вершин отличаются друг от друга своими внутренними графами.

2. Характеризация МВП-графов класса 2-цепь

МВП-графы имеют следующие, имеющие конструктивный характер, свойства.

Утверждение 5 [5]. Из любого n-вершинного МВП-графа можно получить другой МВП-граф с $(n+1)$ вершинами посредством добавления одной вершины и двух рёбер, соединяющих эту вершину с двумя вершинами ребра, принадлежащего внешней грани исходного МВП-графа.

Утверждение 6. Из любого n-вершинного МВП-графа можно получить другой МВП-граф с $(n-1)$ вершинами посредством удаления симплициальной вершины и двух инцидентных ей рёбер.

Доказательства этих утверждений очевидны и непосредственно вытекают из определения МВП-графов. Описанная выше операция добавления вершины и двух рёбер, иногда называемая *элементарным расширением* [5], стала основой для рекурсивной характеризации МВП-графов [10].

Любой МВП-граф, имеющий $n \geqslant 3$ вершин, может быть определён рекурсивно следующим образом:

1) K_3 есть МВП-граф;
2) если G есть МВП-граф, уложеный на плоскости так, что все его вершины принадлежат внешней грани, а граф G' получен добавлением новой вершины и двух рёбер, соединяющих её с вершинами ребра, принадлежащего внешней грани, то G' есть МВП-граф;
3) H есть МВП-граф тогда и только тогда, когда он может быть получен из K_3 выполнением конечной последовательности операций 2.

Дадим характеризацию МВП-графов класса 2-цепь. Для этого понадобится детализация ранее введённого определения элементарного расширения.

Определение 2. *Элементарным расширением s-типа* будем называть элементарное расширение, при котором новая вершина соединяется рёбрами с вершинами ребра, инцидентного симплициальной вершине.

Утверждение 7. Пусть G — МВП-граф класса 2-цепь, имеющий n вершин. Тогда после выполнения операции элементарного расширения s-типа будет получен МВП-граф G' класса 2-цепь с количеством вершин $n+1$.

Доказательство. Пусть G — МВП-граф класса 2-цепь с симплициальными вершинами s_1 и s_2 (см. рис. 2). Для выполнения операции элементарного расширения s-типа выберем симплициальную вершину, например s_2. Эта вершина имеет две смежные с ней вершины x_a и y_b. Вершины x_a, y_b, s_2 и соединяющие их рёбра $\{x_a, y_b\}, \{x_a, s_2\}$ и $\{y_b, s_2\}$ образуют концевую грань.

Выполним операцию элементарного расширения s-типа, то есть добавим новую вершину s_2' и соединим её рёбрами с вершинами одного из двух рёбер, инцидентных симплициальной вершине s_2, например с вершинами ребра $\{s_2, x_a\}$ (или с вершинами ребра $\{s_2, y_b\}$).

В графе G' появляется новая симплициальная вершина s_2', при этом общее количество симплициальных вершин не изменится, поскольку в графе G' вершина s_2 уже не является симплициальной. Следовательно, в результате получается новый МВП-граф G' класса 2-цепь с количеством вершин $n+1$ (рис. 2). ■

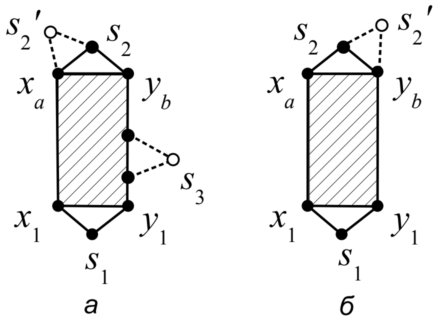


Рис. 2. Операция элементарного расширения s-типа; добавление новой вершины слева (a) и справа ($б$) от концевой грани $\{x_a, y_b, s_2\}$

Замечание 2. Пусть G — МВП-граф, имеющий n вершин и $K_S(G)$ симплициальных вершин. Тогда после выполнения операции элементарного расширения, не являющейся операцией s-типа, будет получен МВП-граф G' с количеством вершин $n+1$ и с количеством симплициальных вершин $K_S(G') = K_S(G)+1$.

На рис. 2, *а* в качестве примера представлен МВП-граф класса 2-цепь с симплициальной вершиной s_3, добавленной при выполнении операции элементарного расширения, не являющейся операцией s-типа.

По аналогии с термином «базовая клика k-дерева», введённым в [7], дадим определение базового подграфа МВП-графа класса 2-цепь.

Определение 3. Подграф G_0 МВП-графа G класса 2-цепь, имеющий $n_0 \geqslant 4$ вершин, называется *базовым подграфом* (*базой*) графа G, если он является МВП-графом класса 2-цепь, индуцирован n_0 вершинами G и одной из вершин G_0 является симплициальная вершина G.

Для графа, изображённого на рис. 1, *в*, базовыми подграфами являются подграфы, индуцированные вершинами 1, 2, 3, 4, или вершинами 1, 2, 3, 4, 5, и т. д.

Утверждение 8. Пусть G_0 — МВП-граф класса 2-цепь порядка $n_0 \geqslant 4$, $k \geqslant 1$ — целое число. Тогда справедливы следующие утверждения:

а) любой n-вершинный ($n = n_0 + k$) МВП-граф класса 2-цепь, имеющий базовый подграф G_0, можно получить из графа G_0 посредством последовательного применения k операций элементарного расширения s-типа;

б) количество вариантов такого построения равно 2^{n-n_0}.

Доказательство. Рассмотрим процесс выполнения последовательности k операций элементарного расширения s-типа. Для сохранения базового подграфа G_0 добавление новых вершин в этом процессе должно производиться только со стороны одной симплициальной вершины G_0 так, чтобы другая его симплициальная вершина всегда оставалась симплициальной. Пусть этот процесс имеет некоторое количество N_R реализаций (вариантов) его выполнения.

Обозначим через $G_{r,1}, G_{r,2}, \ldots, G_{r,k}$ МВП-графы, получаемые из G_0 после исполнения первой, второй, ..., k-й операции элементарного расширения s-типа в r-й реализации. Из утверждения 7 следует, что для любой r-й ($r = 1, 2, \ldots, N_R$) реализации все эти графы являются МВП-графами класса 2-цепь и число $n(G_{r,i})$ вершин графа $G_{r,i}$, $i = 1, 2, \ldots, k$, определяется выражением $n(G_{r,i}) = n(G_{r,i-1}) + 1 = n(G_0) + i$.

Следовательно, для любой r-й реализации после выполнения k операций элементарного расширения s-типа будет получен n-вершинный МВП-граф $G_{r,k}$ класса 2-цепь с базовым подграфом G_0. Количество возможных вариантов такого построения равно $N_R = 2^k = 2^{n-n_0}$. ■

Нетрудно заметить, что величина N_R является верхней оценкой числа непомеченных n-вершинных МВП-графов класса 2-цепь, имеющих базовый подграф G_0.

Утверждения 7 и 8 отражают конструктивные свойства МВП-графов класса 2-цепь, аналогичные свойствам МВП-графов, описанным в утверждениях 5 и 6. Но при этом исходным элементом для построения МВП-графов класса 2-цепь является 4-вершинный МВП-граф, полученный удалением одного ребра из полного графа K_4, который обозначают через D_4 и называют графом диамонд.

Утверждение 9. Любой МВП-граф класса 2-цепь с $n \geqslant 5$ вершинами имеет граф D_4 в качестве базового подграфа.

Доказательство. Из утверждения 4 следует, что слабый двойственный граф $D^W(G)$ графа G имеет две концевые вершины со степенью 1, все остальные вершины имеют степень 2. Следовательно, в графе G есть две грани (будем называть их концевыми), каждая из которых смежна с одной гранью, причём в каждую концевую грань входит одна симплициальная вершина. Все грани графа G треугольные. Таким образом, концевая грань вместе со смежной ей гранью образует подграф D_4, который является базовым подграфом графа G. ■

Утверждение 10. Граф G, имеющий $n \geqslant 4$ вершин, является МВП-графом класса 2-цепь тогда и только тогда, когда он является графом диамонд D_4 или получен из D_4 с помощью конечной последовательности элементарных расширений s-типа.

Доказательство. Необходимость. Пусть G — МВП-граф класса 2-цепь. Тогда из рекурсивного определения МВП-графов следует, что G можно получить из графа K_3 посредством k операций элементарного расширения. Нетрудно заметить, что операция элементарного расширения, выполняемая на графе K_3, является операцией s-типа, в результате которой получается МВП-граф D_4. Таким образом, граф G получается из МВП-графа D_4 посредством выполнения $(k-1)$ операций элементарного расширения.

Из замечания 2 следует, что все $(k-1)$ операции элементарного расширения тоже должны быть операциями s-типа, поскольку в противном случае граф G не был бы МВП-графом класса 2-цепь. Количество операций элементарного расширения s-типа, выполняемых при построении графа G из D_4, равно $(n-4)$.

Достаточность. Пусть D_4 — граф диамонд, $k \geqslant 1$ — целое число. Очевидно, что D_4 есть МВП-граф класса 2-цепь с четырьмя вершинами. По утверждению 8 из графа D_4 посредством k-кратного выполнения операции элементарного расширения s-типа будет получено множество M_G всех n-вершинных ($n = 4+k$) МВП-графов класса 2-цепь, имеющих граф D_4 в качестве базового подграфа. Докажем, что в множество M_G входят все n-вершинные МВП-графы класса 2-цепь.

От противного. Предположим, что существует n-вершинный МВП-граф класса 2-цепь, не принадлежащий множеству M_G, т. е. не имеющий D_4 в качестве базового подграфа. Но по утверждению 9 такое невозможно. ■

Утверждение 11. Пусть G — МВП-граф класса 2-цепь, имеющий $n \geqslant 5$ вершин. Тогда справедливы следующие утверждения:

а) из двух вершин, смежных с симплициальной вершиной, одна является спинальной, а другая — терминальной;

б) после удаления из G симплициальной вершины будет получен МВП-граф класса 2-цепь с количеством вершин $n-1$.

Доказательство. Пусть G — МВП-граф класса 2-цепь, s_2 — одна из его симплициальных вершин, x_a, y_b — вершины, смежные с s_2.

а) Рассмотрим распределение степеней вершин, смежных с вершиной s_2. При этом возможны три случая распределения степеней вершин x_a, y_b (рис. 3).

В первом случае (рис. 3, *а*) вершины x_a, y_b имеют степени $\deg(x_a) \geqslant 4$ и $\deg(y_b) \geqslant 4$. Во втором случае (рис. 3, *б*) вершины x_a, y_b имеют степени равные 3, то есть $\deg(x_a) = 3$ и $\deg(y_b) = 3$. И в третьем случае (рис. 3, *в*) $\deg(x_a) \geqslant 4$ и $\deg(y_b) = 3$ или $\deg(x_a) = 3$ и $\deg(y_b) \geqslant 4$.

Первый случай невозможен, так как в этом случае граф G не плоский. Второй случай невозможен, так как в этом случае граф G не МВП-граф, поскольку одна из его граней не треугольная. Следовательно, возможен только третий случай и, в соответствии с определением 1, одна из вершин x_a, y_b является терминальной, а другая — спинальной.

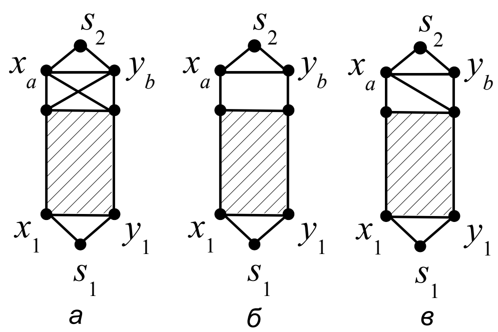


Рис. 3. Распределение степеней вершин x_a, y_b, смежных с симплициальной вершиной s_2 МВП-графа класса 2-цепь: первый случай (*а*), второй случай (*б*), третий случай (*в*)

б) По свойству п. а только одна из вершин, смежных с s_2, имеет степень 3. Именно эта вершина после удаления вершины s_2 станет симплициальной. Другая вершина (с большей степенью) не будет симплициальной. ■

3. Полный инвариант МВП-графов класса 2-цепь

Так как укладка МВП-графа полностью определяется его внутренним графом, естественно определять полный инвариант МВП-графа 2-цепь через полный инвариант его внутреннего графа. Поэтому рассмотрим свойства графов типа гусеница, поскольку они являются внутренними графами МВП-графов. Введём некоторую числовую характеристику гусеницы.

Определение 4 [17]. Пусть T — гусеница со спином $(w_1, w_2, \ldots, w_q)$. Тогда s-кодом гусеницы T называется последовательность целых неотрицательных чисел $(t_1, t_2, \ldots, t_q)$, где t_i есть количество вершин степени 1, смежных с w_i в T.

Для каждого $i = 1, 2, \ldots, q$ величина t_i и степень $\deg_T(w_i)$ вершины w_i спина связаны следующими соотношениями:

$$t_1 = \deg_T(w_1) - 1, \quad t_q = \deg_T(w_q) - 1, \quad t_i = \deg_T(w_i) - 2, \quad i = 2, 3, \ldots, q-1.$$

Сумма всех величин $t_1, t_2, \ldots, t_q$ определяет количество n_f концевых вершин гусеницы. Для n-вершинной гусеницы с количеством вершин спина q верно $n_f = n - q$. В дальнейшем s-код гусеницы будем обозначать через $S(T)$.

Очевидно, что любая невырожденная гусеница T имеет два s-кода:

$$S_1(T) = (t_1, t_2, \ldots, t_q), \quad S_2(T) = (t_q, t_{q-1}, \ldots, t_1).$$

В качестве примера рассмотрим внутренние графы МВП-графов, изображённых на рис. 1. Граф звезда S_5, который является внутренним графом МВП-графа типа «веер», имеет s-код $S(S_5) = (4)$, граф P_8 — внутренний граф МВП-графа типа «лестница» — имеет два одинаковых s-кода:

$$S_1(P_8) = (1, 0, 0, 0, 0, 1), \quad S_2(P_8) = (1, 0, 0, 0, 0, 1).$$

Внутренний граф T' графа, изображённого на рис. 1, *в*, имеет два разных s-кода: $S_1(T') = (2, 1, 2, 0, 0, 1)$ и $S_2(T') = (1, 0, 0, 2, 1, 2)$.

В [17] предложено использовать уникальный c-код гусеницы T как лексикографически максимальный из его кодов $S_1(T)$ и $S_2(T)$. Так, c-код внутреннего графа T' графа, изображённого на рис. 1, *в*, равен $S_1(T')$, то есть $c(T') = (2, 1, 2, 0, 0, 1)$, поскольку из его двух s-кодов $S_1(T')$ и $S_2(T')$ код $S_1(T')$ лексикографически максимальный.

Нетрудно заметить, что по гусенице T однозначно определяется её c-код $c(T)$, а по c-коду определяется гусеница.

Лемма 1. Два графа типа «гусеница» T_1 и T_2 изоморфны тогда и только тогда, когда $c(T_1) = c(T_2)$.

Доказательство. Необходимость. Докажем, что по гусенице однозначно получается её c-код. Действительно, по определению c-кода, порядок следования и величины элементов c-кода определяются порядком следования и величинами степеней вершин спина и не зависят ни от нумерации вершин, ни от способа укладки гусеницы. Если две гусеницы T_1 и T_2 изоморфны, то между множествами их вершин есть взаимно однозначное соответствие, сохраняющее смежность и степени вершин. Следовательно, их c-коды равны друг другу, то есть $c(T_1) = c(T_2)$.

Достаточность. Докажем, что из c-кода однозначно получается гусеница. Пусть имеем c-код $(c_1, c_2, \ldots, c_q)$. По нему строится простая цепь с вершинами $w_1, w_2, \ldots, w_q$. Затем к каждой вершине w_i добавляется c_i вершин, каждая из которых соединяется ребром с вершиной w_i. Полученный граф является гусеницей, в которой каждому элементу c_i c-кода соответствует одна вершина w_i спина и подмножество из c_i концевых вершин, смежных с вершиной w_i (рис. 4, *б*).

Следовательно, из двух равных друг другу c-кодов получаются две изоморфные гусеницы, то есть из равенства $c(T_1) = c(T_2)$ следует $T_1 \cong T_2$. ■

Легко показать, что алгоритм построения гусеницы по её c-коду имеет временную сложность $\mathrm{O}(n)$.

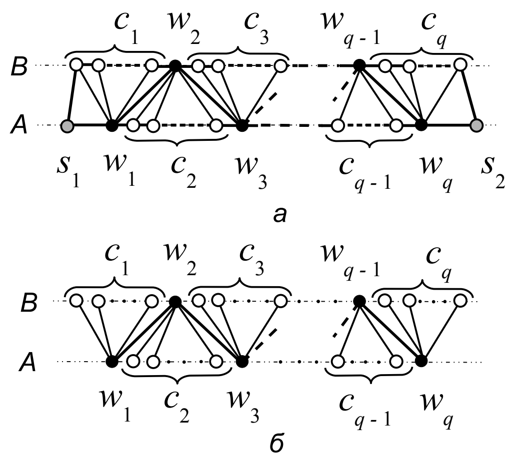


Рис. 4. МВП-граф класса 2-цепь (a) и его внутренний граф — гусеница ($б$)

Из леммы 1 следует, что c-код гусеницы является её полным инвариантом и любая гусеница может быть однозначно восстановлена по её c-коду.

Докажем, что код гусеницы является и полным инвариантом МВП-графа класса 2-цепь, для которого гусеница является его внутренним графом. Для этого сначала докажем следующую лемму.

Лемма 2. Из c-кода гусеницы T_C однозначно получается МВП-граф G класса 2-цепь, для которого гусеница T_C является внутренним графом.

Доказательство. Пусть имеем гусеницу T_C с общим количеством вершин p, количеством q вершин спина и c-кодом $(c_1, c_2, \ldots, c_q)$.

Плоская BA-укладка гусеницы T_C является внутренним графом некоторого LL-плоского МВП-графа G класса 2-цепь порядка $n = p + 2$. Определим структуру графа G. Она полностью определяется плоской BA-укладкой его внутреннего графа T_C (см. рис. 1 и 4). Отсюда, используя утверждение 11, а и двудольность гусениц, нетрудно показать, что граф G имеет следующие свойства:

1) первая спинальная вершина и одна из смежных с ней терминальных вершин используются при создании списка рёбер концевой грани $\{1, 2, 3\}$;
2) терминальные вершины, смежные с i-й спинальной вершиной, и $(i + 1)$-я спинальная вершина находятся на одной линии, отличной от линии, на которой находится i-я спинальная вершина;
3) если пронумеровать спинальные вершины (в порядке их следования в спине) числами $1, 2, \ldots, q$, то все вершины с нечётными номерами будут располагаться на одной линии, а с чётными номерами — на другой.

Таким образом, предлагается следующий алгоритм построения МВП-графа класса 2-цепь из c-кода $c = (c_1, c_2, \ldots, c_q)$:

1) Построить две параллельные линии A и B; построить граф K_3 с вершинами 1, 2, 3; при построении графа вершину 2 разместить на линии B, вершину 3 — на линии A так, чтобы ребро $\{2, 3\}$ располагалось вертикально, а вершину 1 — слева от ребра $\{2, 3\}$.
2) Для $i = 1, 2, \ldots, q$ выполнить следующие действия:

а) определить количество k_i добавляемых вершин по следующим выражениям: если $i = 1$ и $q > 1$, то $k_i = c_1$; если $i = 1$ и $q = 1$, то $k_i = c_1 - 1$; если $1 < i < q$, то $k_i = (c_i + 1)$; если $i = q$, то $k_i = c_q$;

б) выполнить k_i раз операцию элементарного расширения s-типа; каждая добавляемая вершина соединяется двумя рёбрами с последними вершинами, находящимися на линиях A, B, и размещается на одной из этих линий в зависимости от чётности номера i.

3) Добавляется новая вершина и два ребра, соединяющие её с двумя последними вершинами, находящимися на линиях A и B.

Легко видеть, что предложенный алгоритм позволяет построить МВП-граф класса 2-цепь по c-коду его внутреннего графа, являющегося гусеницей. Однозначность такого построения следует из леммы 1. ■

Определим временную сложность алгоритма. Алгоритм имеет один внешний и один вложенный цикл. Внешний цикл выполняется q раз. Количество k_i повторений вложенного цикла (для i-й итерации внешнего цикла) равно $k_i = c_i + 1$ при $i \neq q$ и $k_i = c_i$ при $i = 1$ и $i = q$. Общее количество выполняемых итераций равно

$$\sum_{i=1}^{q}(c_i + 1) - 2 = \sum_{i=1}^{q} c_i + (q - 2) = n_f + (q - 2) = n - 4.$$

Следовательно, временная сложность алгоритма получения МВП-графа класса 2-цепь из c-кода равна $\mathrm{O}(n)$.

Из леммы 2 вытекает следующее утверждение об изоморфизме МВП-графов класса 2-цепь.

Лемма 3. Два n-вершинных МВП-графа класса 2-цепь изоморфны тогда и только тогда, когда изоморфны их внутренние графы.

Доказательство. Необходимость. Пусть имеем два изоморфных n-вершинных МВП-графа G_1 и G_2 класса 2-цепь, а T_1 и T_2 — их внутренние графы, являющиеся гусеницами. Так как внутренние графы T_1 и T_2 получены из графов G_1 и G_2 удалением n внешних рёбер, то изоморфным МВП-графам однозначно соответствуют изоморфные внутренние графы, то есть если $G_1 \cong G_2$, то $T_1 \cong T_2$.

Достаточность. Пусть T_1 и T_2 — изоморфные внутренние графы графов G_1 и G_2. По лемме 1 графы T_1 и T_2, являющиеся гусеницами, имеют одинаковые c-коды, то есть $c(T_1) = c(T_2)$. Из леммы 2 следует, что из c-кода графа типа «гусеница» T_c с p вершинами можно с точностью до изоморфизма построить n-вершинный ($n = p + 2$) МВП-граф G класса 2-цепь, для которого T_c будет внутренним графом. Следовательно, из двух одинаковых c-кодов $c(T_1)$ и $c(T_2)$ получаются два изоморфных МВП-графа G_1 и G_2 класса 2-цепь, то есть если $T_1 \cong T_2$, то $G_1 \cong G_2$. ■

Из лемм 1 и 3 следует справедливость следующего утверждения.

Теорема 1. Два n-вершинных МВП-графа G_1 и G_2 класса 2-цепь с внутренними графами T_1 и T_2 изоморфны тогда и только тогда, когда равны c-коды их внутренних графов T_1 и T_2, то есть $G_1 \cong G_2 \Leftrightarrow c(T_1) = c(T_2)$.

Таким образом, c-код является полным инвариантом для гусениц (лемма 1) и для МВП-графов класса 2-цепь (теорема 1).

Полученный полный инвариант МВП-графа класса 2-цепь отличается от полного инварианта, описанного в [10], как по количеству его элементов, так и по их значениям.

Нетрудно заметить, что c-код является линейным списком в отличие от гамильтоновой степенной последовательности, которая является циклическим списком. Поэтому количество элементов c-кода может быть легко увеличено посредством добавления новых симплициальных вершин.

Приведём алгоритм расчёта полного инварианта МВП-графов класса 2-цепь. Будем исходить из того, что для любого МВП-графа класса 2-цепь его c-код определяется из последовательности спинальных вершин $w_1, w_2, \ldots, w_q$ по их степеням $\deg_G(w_1), \deg_G(w_2), \ldots, \deg_G(w_q)$. Действительно, в МВП-графе G класса 2-цепь при $q > 1$ для каждого $i = 1, 2, \ldots, q$ величина t_i и степень $\deg_G(w_i)$ спинальной вершины w_i связаны следующими соотношениями:

$$t_1 = \deg_G(w_1) - 3, \quad t_q = \deg_G(w_q) - 3, \quad t_i = \deg_G(w_i) - 4, \quad i = 2, 3, \ldots, q-1. \qquad (4)$$

В случае, когда в МВП-графе есть только одна спинальная вершина, то есть при $q = 1$, для t_1 выполняется равенство

$$t_1 = \deg_G(w_1) - 2. \qquad (5)$$

С учётом этого алгоритм получения c-кода МВП-графа класса 2-цепь состоит из следующих шагов:

1) создать список D степеней вершин, список $p = \{s_1, s_2\}$ симплициальных вершин и определить количество q спинальных вершин;
2) определить две вершины a и b, смежные с симплициальной вершиной $s_1 = p(1)$;
3) из двух вершин a и b выбрать одну спинальную вершину w_1;
4) создать упорядоченный список W спинальных вершин;
5) определить s-коды S_1 и S_2;
6) определить c-код как лексикографически максимальный из s-кодов.

Исходный МВП-граф G удобно задавать списком A_L смежностей вершин.

На первом шаге создаётся список степеней вершин. Определение симплициальных вершин s_1, s_2 и всех спинальных вершин производится по их степеням.

На втором шаге определение вершин a и b, смежных с симплициальной вершиной s_1, производится по списку смежностей.

На третьем шаге из вершин a и b выбирается спинальная вершина w_1.

На четвертом шаге создаётся список меток вершин, заполненный нулями. Симплициальные вершины s_1, s_2 и первая спинальная вершина w_1 помечаются как уже просмотреннные. Затем создаётся пустой список W спинальных вершин и в него добавляется вершина w_1. Далее для каждой спинальной вершины w_c определяется список N_c смежных с ней вершин, вершины из этого списка просматриваются и каждая непомеченная вершина $v \in N_c$ со степенью $\deg_G(v) \geqslant 4$ добавляется в список W, затем все просмотренные вершины помечаются.

На пятом шаге проверяется условие $q > 1$. При $q = 1$ выполняется расчёт s-кода S_1 по выражению (5), при $q > 1$ — по выражениям (4). Далее с помощью реверсирования S_1 получается s-код S_2.

На шестом шаге (в цикле) элементы кодов S_1 и S_2 сравниваются друг с другом и в качестве c-кода принимается лексикографически максимальный из s-кодов.

Определим временную сложность алгоритма. Количество операций, выполняемых на первом шаге, равно сумме степеней всех вершин и, следовательно, равно $2m$. Так как для МВП-графов $m = 2n - 3$, количество операций, выполняемых на первом шаге, равно $4n - 6$. Количество операций, выполняемых на четвёртом шаге, равно сумме степеней спинальных вершин и, следовательно, равно количеству терминальнальных вершин, то есть равно $(n - q - 2)$. Количество операций, выполняемых на пятом шаге, определяется количеством спинальных вершин и равно $2q$. Максимальное количество операций шестого шага равно $\lfloor q/2 \rfloor$.

Таким образом, временная сложность алгоритма получения c-кода МВП-графа класса 2-цепь равна $\mathrm{O}(n)$.

Замечание 3. Полный инвариант МВП-графов класса 2-цепь можно определить методом последовательного удаления симплициальных вершин. В основе метода лежит свойство МВП-графов класса 2-цепь, доказанное в утверждении 11, а. Однако алгоритм, реализующий этот метод, довольно сложен и здесь не рассматривается.

4. Подсчёт количества непомеченных МВП-графов класса 2-цепь

Найдём точное количество непомеченных МВП-графов класса 2-цепь с использованием основных свойств гусениц, которые являются внутренними графами МВП-графов класса 2-цепь. Будем использовать работу [15], в которой определено количество непомеченных гусениц.

Утверждение 12 [15]. Количество n-вершинных ($n \geqslant 3$) непомеченных гусениц равно $C_n = 2^{n-4} + 2^{\lfloor (n-4)/2 \rfloor}$

Для $n = 1, 2, 3, 4, \ldots$ количество n-вершинных гусениц определяет последовательность $1, 1, 1, 2, 3, 5, 10, 20, 36, 72, \ldots$ (последовательность A005418 в [18]).

Теорема 2. Количество n-вершинных ($n \geqslant 5$) непомеченных МВП-графов класса 2-цепь равно $2^{n-6} + 2^{\lfloor (n-6)/2 \rfloor}$.

Заключение

В работе дано полное описание МВП-графов класса 2-цепь. Получена их рекурсивная характеризация, в основу которой положено понятие элементарного расширения s-типа. Основными результатами являются: условие изоморфизма (лемма 3), полный инвариант (теорема 1) и формула для числа непомеченных МВП-графов класса 2-цепь (теорема 2). Получены полиномиальные алгоритмы построения гусениц и МВП-графов класса 2-цепь по c-коду, а также полиномиальный алгоритм расчёта полного инварианта МВП-графов класса 2-цепь.

Перспективным направлением дальнейших исследований является использование c-кода МВП-графов класса 2-цепь для кодирования информации.

ЛИТЕРАТУРА

1. *Chartrand G. and Harary F.* Planar permutation graphs // Ann. Inst. Henri Poincare, Sec. B. 1967. V. 3. No. 4. P. 433–438.
2. *Харари Ф.* Теория графов. М.: Мир, 1973. 300 с.
3. *Евстигнеев В. А., Касьянов В. Н.* Словарь по графам в информатике. Новосибирск: ООО «Сибирское Научное Издательство», 2009. 300 с.
4. *Mitchel S.* Linear algorithms to recognize outerplanar and maximal outerplanar graphs // Inform. Process. Lett. 1979. V. 5. No. 1. P. 229–232.
5. *Asratian A. S. and Oksimets N.* Graphs with hamiltonian balls // Australasian J. Combinatorics. 1998. V. 17. No 4. P. 185–198.
6. *Markov M.* On the vertex separation of maximal outerplanar graphs // Serdica J. Computing. 2008. V. 154. No 5. P. 207–238.
7. *Hedetniemi S. M., Proskurowski A. J., and Syslo M. M.* Interior graphs of maximal outerplane graphs // J. Combin. Theory. Ser. B. 1985. V. 38. No. 2. P. 156–167.

8. *Cornelsen S., Schank T., and Wagner D.* Drawing graphs on two and three lines // J. Graphs Algorithms Appl. 2004. V. 8. No. 2. P. 161–177.
9. *Guanzhang Hu.* Catalan number and enumeration of maximal outerplanar graphs // Tsinghua Sci. Technol. 2000. V. 5. No. 1. P. 109–114.
10. *Beyer T., Jones W., and Mitchell S.* Linear algorithms for isomorphism of maximal outerplanar graphs // J. ACM. 1979. V. 26. No. 4. P. 603–610.
11. *Носов Ю. Л.* Индекс Винера максимальных внешнеплоских графов // Прикладная дискретная математика. 2014. № 4(26). С. 112–122.
12. *Markenzon L., Justel C. M., and Paciornik N.* Subclasses of k-trees: Characterization and recognition // Discr. Appl. Math. 2006. V. 154. No. 5. P. 818–825.
13. *Майника Э.* Алгоритмы оптимизации на сетях и графах. М.: Мир, 1981. 324 с.
14. *Свами М., Тхуласираман К.* Графы, сети и алгоритмы. М.: Мир, 1984. 455 с.
15. *Harary F. and Schwenk A. J.* The number of caterpillars // Discr. Math. 1973. V. 6. No. 4. P. 359–365.
16. *Harary F. and Schwenk A. J.* A new crossing number for bipartite graphs // Utilitas Math. 1972. V. 1. P. 203–209.
17. *Gionfrido M., Harary F., and Tuza Z.* The color cost of caterpillar // Discr. Math. 1997. V. 174. No. 1–3. P. 125–130.
18. `http://www.oeis.org/A005418` — Sloane N. J. A. The On-Line Encyclopedia of Integer Sequences.

REFERENCES

1. *Chartrand G. and Harary F.* Planar permutation graphs. Ann. Inst. Henri Poincare, Sec. B, 1967, vol. 3, no. 4, pp. 433–438.
2. *Harary F.* Graph Theory. Perseus Books, 1994.
3. *Evstigneev V. A., Kas'yanov V. N.* Slovar' po grafam v informatike [Dictionary of Graphs in Computer Science]. Novosibirsk, OOO «Sibirskoe Nauchnoe Izdatel'stvo», 2009. 300 p. (in Russian)
4. *Mitchel S.* Linear algorithms to recognize outerplanar and maximal outerplanar graphs. Inform. Process. Lett., 1979, vol. 5, no. 1, pp. 229–232.
5. *Asratian A. S. and Oksimets N.* Graphs with hamiltonian balls. Australasian J. Combinatorics, 1998, vol. 17, no. 4, pp. 185–198.
6. *Markov M.* On the vertex separation of maximal outerplanar graphs. Serdica J. Computing, 2008, vol. 154, no. 5, pp. 207–238.
7. *Hedetniemi S. M., Proskurowski A. J., and Syslo M. M.* Interior graphs of maximal outerplane graphs. J. Combin. Theory. Ser. B, 1985, vol. 38, no. 2, pp. 156–167.
8. *Cornelsen S., Schank T., and Wagner D.* Drawing graphs on two and three lines. // J. Graphs Algorithms Appl., 2004, vol. 8, no. 2, pp. 161–177.
9. *Guanzhang Hu.* Catalan number and enumeration of maximal outerplanar graphs. Tsinghua Sci. Technol., 2000, vol. 5, no. 1, pp. 109–114.
10. *Beyer T., Jones W., and Mitchell S.* Linear algorithms for isomorphism of maximal outerplanar graphs. J. ACM, 1979, vol. 26, no. 4, pp. 603–610.
11. *Nosov Yu. L.* Indeks Vinera maksimal'nykh vneshneploskikh grafov [The Wiener index of maximal outerplane graphs.] Prikladnaya diskretnaya matematika, 2014, no. 4(26), pp. 112–122. (in Russian)
12. *Markenzon L., Justel C. M., and Paciornik N.* Subclasses of k-trees: Characterization and recognition. Discr. Appl. Math., 2006, vol. 154, no. 5, pp. 818–825.

13. *Minieka E.* Optimization Algorithms for Networks and Graphs. N.Y., Basel, Marcel Dekker, Inc., 1978. 323 p.
14. *Swamy M. N. S. and Thulasiraman K.* Graphs, Networks and Algorithms. N.Y., Wiley, 1981. 592 p.
15. *Harary F. and Schwenk A. J.* The number of caterpillars. Discr. Math., 1973, vol. 6, no. 4, pp. 359–365.
16. *Harary F. and Schwenk A. J.* A new crossing number for bipartite graphs. Utilitas Math., 1972, vol. 1, pp. 203–209.
17. *Gionfrido M., Harary F., and Tuza Z.* The color cost of caterpillar. Discr. Math., 1997, vol. 174, no. 1–3, pp. 125–130.
18. `http://www.oeis.org/A005418` — Sloane N. J. A. The On-Line Encyclopedia of Integer Sequences.

ДИСКРЕТНЫЕ МОДЕЛИ РЕАЛЬНЫХ ПРОЦЕССОВ

УДК 519.6:576.3

ПОСТРОЕНИЕ НЕОДНОРОДНОГО МАССИВА ЯЧЕЕК ДЛЯ ЗАДАЧ КЛЕТОЧНО-АВТОМАТНОГО МОДЕЛИРОВАНИЯ РОСТА И ДЕЛЕНИЯ КЛЕТОК БАКТЕРИЙ[1]

А. А. Витвицкий

Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск, Россия

Представлен метод организации массива ячеек клеточного автомата, который позволяет моделировать динамику поверхности клеток бактерий различных форм. Основная идея метода заключается в том, что поверхность разбивается на слои ячеек, что даёт возможность изменять отдельные части структуры моделируемой поверхности, не перестраивая всю поверхность целиком. Кроме того, такая организация ячеек позволила реализовать быстрый алгоритм определения соседства ячеек, также представленный в этой работе. На основе предложенных алгоритмов реализован программный комплекс и проведены вычислительные эксперименты, которые показали эффективность предложенного подхода в задачах, где необходимо изучение двусторонней связи между процессами, приводящими к динамике поверхности, и непосредственно самой динамикой.

Ключевые слова: *клеточные автоматы, самоорганизация, динамика поверхности, компьютерное моделирование, рост и деление клеток, E.coli.*

DOI 10.17223/20710410/29/9

CONSTRUCTION OF INHOMOGENEOUS 3D MESH FOR SIMULATION OF BACTERIAL CELL GROWTH AND DIVISION BY CELLULAR AUTOMATA

A. A. Vitvitsky

Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia

E-mail: vitvit@ssd.sscc.ru

A method for constructing a 3D mesh for computer simulating the dynamics of a bacterial cells surface by cellular automata is presented. The idea of the proposed method is based on the dividing the cell surface into layers having the form of rings of nodes. This enables to change separate parts of the mesh structure without rebuilding the entire surface. Moreover, a fast algorithm for determining the neighbourhood of nodes on the spheroidal parts of the cell surface has been developed. The proposed algorithms have been implemented as a software package. A series of computational

[1]Работа поддержана грантом РФФИ № 14-01-31425 mol_a.

experiments showed the effectiveness of the proposed method for simulation of the interactions between the complex processes inside bacterial cells, leading to dynamical change of their surface.

Keywords: *cellular automata, self-organization, surface dynamics, computer simulation, cell growth and division, E.coli.*

Введение

Механизмы, управляющие ростом и делением бактериальных клеток, до сих пор мало изучены. Доподлинно известно, что деление клетки бактерии начинается с образования Z-кольца в геометрической середине клетки (рис. 1). Z-кольцо представляет собой сложный белковый комплекс, который стягивает клетку, образуя разделительную септу, и, в конечном итоге, делит бактерию на две дочерних [1, 2]. Выбор места сборки Z-кольца (а значит, и места деления) является критически важным для жизни бактерии, так как деление в неправильном месте приводит к большой вероятности смерти дочерних клеток. Как установлено в [3], за координацию позиции сборки Z-кольца отвечают некоторые самоорганизационные механизмы, белковый состав которых известен, но точный принцип действия не до конца изучен. Последнее обстоятельство обусловлено трудностями, которые возникают при попытке отследить динамику отдельных белков, участвующих в этих механизмах, в лабораторных условиях. По этой причине исследователи интенсивно применяют компьютерное моделирование, которое помогает проверить различные теоретические модели [4 – 6].

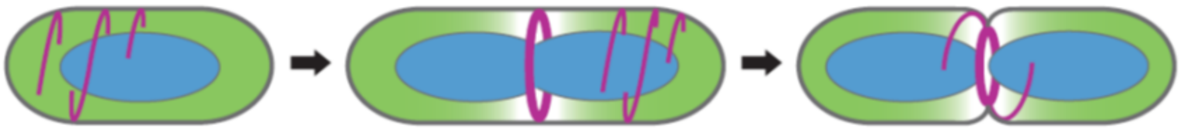


Рис. 1. Z-кольцо собирается в середине клетки, где концентрация негативных регуляторов сборки достаточно мала. Рисунок адаптирован из [1]

Однако большинство существующих моделей, которые применяются для изучения вышеупомянутых самоорганизационных механизмов, отображают процессы, происходящие в статичных структурах, имитирующих форму поверхности клетки бактерии, не изменяемую во времени. Последнее исключает прямое изучение обратной связи между самоорганизационными механизмами, приводящими к росту и делению, и самими процессами роста и деления. Данная сложность объясняется тем, что классические инструменты моделирования, используемые в таких моделях, обладают слабой эффективностью в задачах с динамикой структуры моделируемой поверхности. Например, в [7] успешно смоделирована самоорганизация системы белков MinCDE, которая контролирует выбор места сборки Z-кольца в бактериях *E.coli.* В качестве инструмента моделирования использована система дифференциальных уравнений, решаемая при помощи метода конечных элементов. Однако моделирование производилось на статичной трёхмерной сетке, которая имитировала поверхность клетки *E.coli.* В [7] смоделирован также рост клетки во время самоорганизации белков MinDE, но только в одномерном случае, что объясняется сложностью моделирования динамики поверхности в больших размерностях (при помощи выбранного инструмента моделирования).

Клеточные автоматы (КА) показали свою эффективность при моделировании многих самоорганизационных механизмов [8 – 10] , что делает их весьма привлекательными для задач моделирования процессов, происходящих в клетках бактерий. КА является дискретной математической моделью, состоящей из:

1) массива ячеек, где каждая ячейка имеет координату и обладает состоянием;
2) правил переходов, при помощи которых ячейки изменяют свои состояния.

Кроме того, КА обладают такими неоспоримыми преимуществами, как простота составления модельных правил и естественный мелкозернистый параллелизм [11, 12]. Однако в классическом КА массив ячеек представляется в виде однородной регулярной структуры. Такая организация затрудняет моделирование структур, изменяемых во времени (например, рост и деление). Таким образом, в этой работе поставлена задача разработать метод организации неоднородного массива ячеек КА, позволяющий:

1) имитировать различные 3D-поверхности бактериальных клеток;
2) изменять структуру поверхности во времени (т. е. моделировать процессы роста и деления бактериальных клеток);
3) КА, использующий такой массив ячеек, должен иметь возможность моделировать какие-либо дополнительные процессы, происходящие во время роста и деления бактериальной клетки (например, самоорганизацию белков MinCDE).

1. Организация массива ячеек с неоднородной структурой

Организацию массива ячеек, предлагаемую в этой работе, можно представить в виде иерархии структур (рис. 2). На верхнем уровне располагается моделируемая поверхность (в данном случае поверхность клетки бактерии). Далее поверхность разбивается на слои (кольца), которые, в свою очередь, состоят из ячеек. Такая организация в виде колец удобна в задачах с динамикой моделируемой поверхности, так как позволяет изменять структуру поверхности, не изменяя всю поверхность целиком. Таким образом, осуществляется редукция измерений: из трёхмерного пространства поверхности (XYZ, где ось Z направлена вдоль поверхности) к двумерной плоскости кольца (XY) и далее к одномерному вектору ячеек (X).

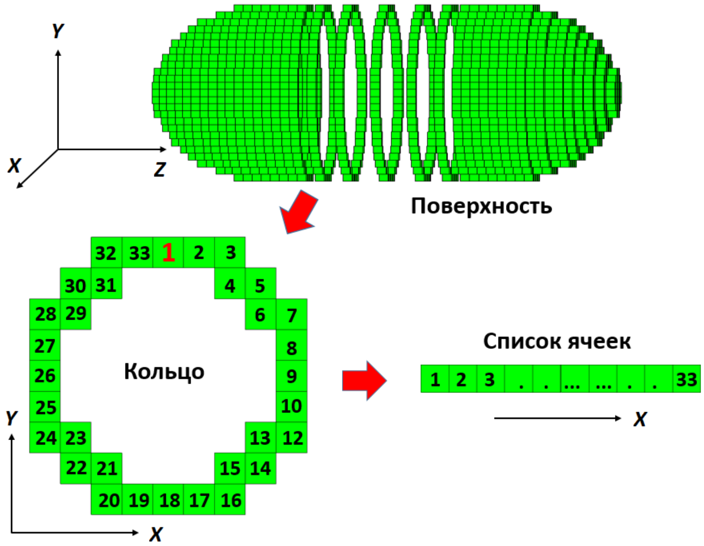


Рис. 2. Иерархия организации массива ячеек клеточного автомата

Основной структурной единицей поверхности является кольцо. Каждое кольцо характеризуется тремя параметрами: радиусом, 3D-координатой относительно поверхности и списком ячеек. Каждая ячейка из списка, в свою очередь, обладает состоянием и

2D-координатой относительно центра кольца. Относительная 2D-координата выбрана для того, чтобы при смене положения кольца в пространстве не нужно было переписывать координату каждой отдельной ячейки этого кольца. Следует отметить, что обход ячеек в кольце начинается с ячейки, имеющей максимальную по оси Y координату (позиция $\pi/2$ кольца). Этот факт важен при определении соседства ячеек и будет объяснён в п. 2.

1.1. Алгоритм построения колец

Алгоритм построения кольца ячеек на плоскости XY (рис. 3):

1) Плоскость разбивается на ячейки с шагом h (размер ячейки) так, что центр одной из ячеек лежит в начале координат (т. е. имеет координаты $x_i = 0$ и $y_i = 0$).
2) Ячейки (x_i, y_i), для которых выполняется условие $r - h \leqslant d_i < r + h$, где r — заданный радиус кольца, $d_i = \sqrt{x_i^2 + y_i^2}$, образуют список ячеек данного кольца.

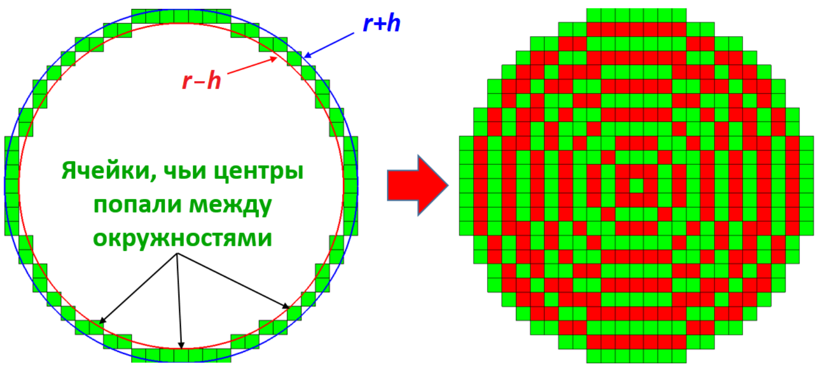


Рис. 3. Графическое представление алгоритма генерации колец ячеек и результат его работы (соседние кольца представлены чередующимися цветами)

Таким способом можно построить кольца различного радиуса, которые располагаются плотно друг к другу, не оставляя промежутков и не пересекаясь между собой (рис. 3).

Утверждение 1. Плоскость может быть плотно и без пересечений заполнена кольцами ячеек, если выполняются условия:

1) $r_i = a_i h$, где r_i — радиус i-го кольца, $a_i \in \mathbb{N}$, $h > 0$ — размер ячейки;
2) $\forall a_i \, \exists a_j \, (|a_i - a_j| = 1)$, т. е. для каждого кольца существует по крайней мере одно соседнее кольцо с радиусом, отличным на величину h, что гарантирует плотность заполнения;
3) $\forall i \neq j \, (a_i \neq a_j)$, т. е. не существует колец с одинаковыми радиусами, что гарантирует отсутствие пересечений.

Утверждение 2. Пространство может быть плотно и без пересечений заполнено кольцами ячеек, если для колец с одинаковой координатой z_i выполняется утверждение 1 и условия этого утверждения распространяются на координату z_i так, что $z_i = b_i h$, где $b_i \in \mathbb{R}$.

1.2. Алгоритм построения поверхности клетки бактерии

Клетку бактерии можно условно разделить на тело клетки и её полюсы. Тело удобно представить в виде цилиндра, а полюсы — в виде двух половин одного сфероида (рис. 4). Таким образом, задача организации структуры, имитирующей поверхность клетки бактерии, сводится к построению из ячеек цилиндра и двух сфероидов. Цилиндр легко составляется из колец ячеек (рис. 5, слева) и может быть определён высотой, которая задаёт число колец в цилиндре, и радиусом основания, задающим радиус всех колец цилиндра. Сфероид, в свою очередь, определяется полуосью a, которая параллельна оси координат Z, и двумя равными полуосями b и c, параллельными осям X и Y соответственно (рис. 5, справа). Построение сфероида из ячеек является более сложной задачей, чем построение цилиндра, так как дополнительно требуется определить радиус для каждого кольца.

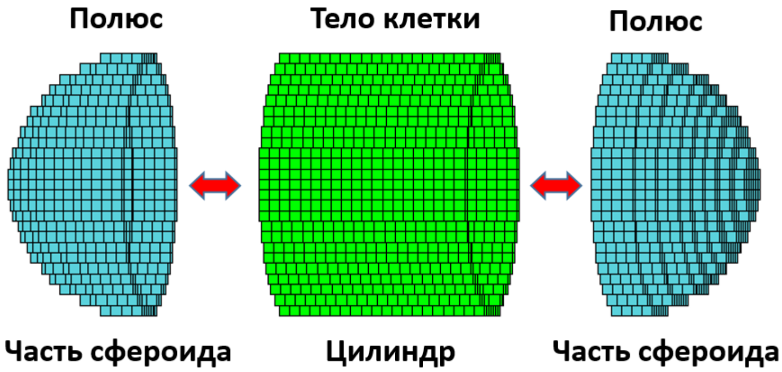


Рис. 4. Представление поверхности клетки в виде цилиндра и двух половин сфероида

Рассмотрим *алгоритм построения сфероида* (рис. 6):

1) В плоскости YOZ строятся два вспомогательных эллипса (вернее, только их первая четверть) с центрами в начале координат и осями $(a - h, b - h)$ и $(a + h, b + h)$, где h — размер ячейки.
2) На плоскость YOZ накладывается сетка с шагом h так, что один из узлов лежит в начале координат, а координаты остальных узлов больше либо равны 0.
3) Узлы, попавшие между вспомогательными эллипсами, определяют радиусы r_i и координаты z_i каждого из колец сфероида.

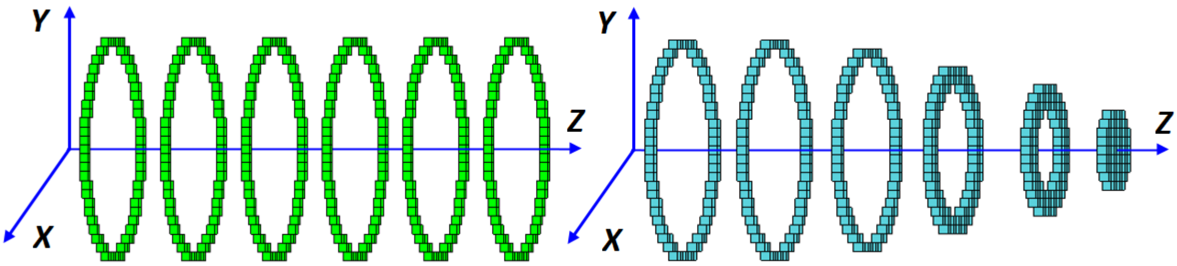


Рис. 5. Построение цилиндра (слева) и половины сфероида (справа)

Стоит отметить, что алгоритм построения сфероида идейно схож с алгоритмом построения колец, однако имеет решающее отличие: в алгоритме построения колец мы работаем в плоскости XY и по заданным параметрам определяем ячейки, составляющие кольцо, в то время как в алгоритме построения сфероида мы работаем в плоскости YZ и только определяем радиусы и координаты колец; далее по найденным параметрам уже строятся сами кольца. На рис. 6 также можно увидеть, что алгоритм построения сфероида допускает вложенные кольца (т. е. кольца с одной координатой z, но разными радиусами r_i).

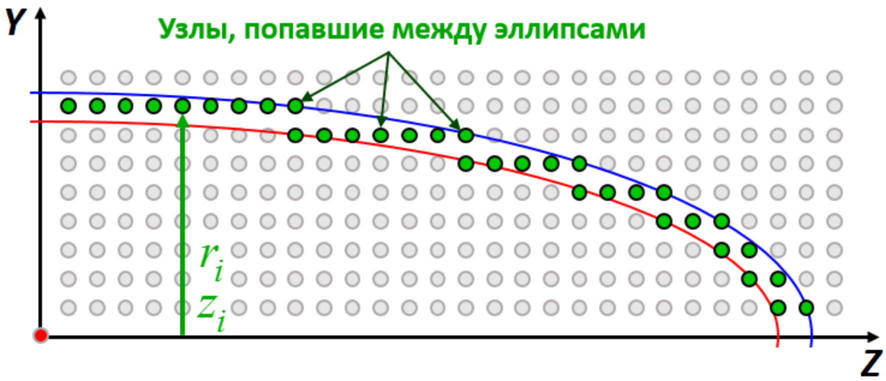


Рис. 6. Графическое представление алгоритма определения параметров колец сфероида

1.3. Примеры различных вариантов поверхностей

На рис. 7 представлены три поверхности, которые имитируют различные формы бактериальной клетки. Эти поверхности получены в результате варьирования параметров при построении цилиндра и сфероида. В правом нижнем углу изображена увеличенная поверхность, на которой можно видеть, что она состоит из отдельных ячеек. Клеточный автомат, работающий на этих массивах ячеек, моделирует процесс самоорганизации белков MinDE.

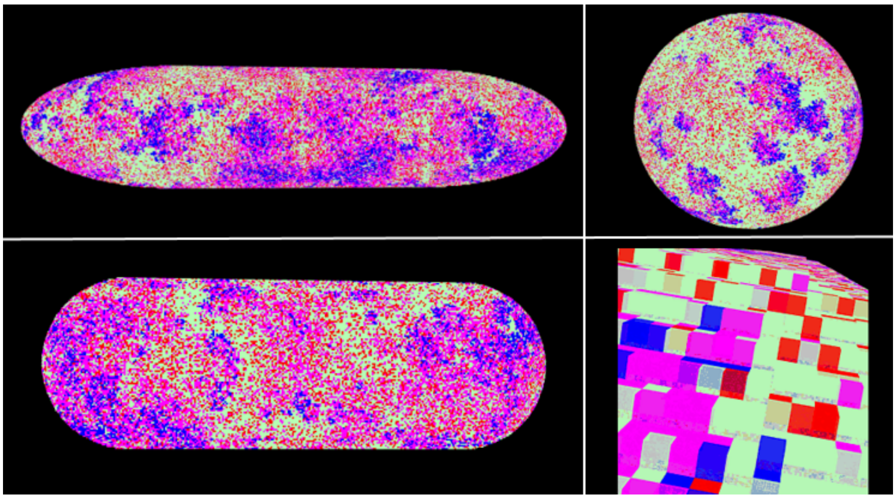


Рис. 7. Различные варианты поверхностей, моделирующих клетку

2. Алгоритм определения соседства ячеек

Задача определения соседства ячеек играет одну из ключевых ролей при составлении правил КА-модели. Вычисление соседства на регулярной решётке требует лишь расчёта смещений по индексам (особое внимание уделяется только граничным случаям). Однако ситуация кардинально меняется на неоднородной решётке, моделирующей какую-либо поверхность. Функция соседства в этом случае весьма нетривиальна, а вычисление соседства с достаточно большим радиусом может быть очень ресурсоёмкой операцией, выполнение которой, в большинстве случаев, может потребоваться на каждой итерации для каждой ячейки. Одним из способов решения этой проблемы является однократный расчёт соседства для каждой ячейки и занесение этого результата в таблицу. Однако с ростом числа ячеек и величины радиуса соседства может потребоваться громадное количество памяти, требуемое для хранения таких таблиц. К примеру, кубическая решётка со стороной в 1000 клеток и радиусом соседства 10 клеток требует $1000^3 \cdot 10^3 \cdot 8$ байт памяти (в случае 64-разрядных систем).

Кроме того, в задачах с динамикой структуры может потребоваться производить перерасчёт этих таблиц после каждого изменения структуры пространства. Поэтому одним из основных критериев, который принят при выборе метода организации неоднородного массива ячеек в этой работе, является возможность реализовать на таком массиве простой и эффективный алгоритм определения соседства.

Самый очевидный способ вычислить на трёхмерной поверхности соседство для конкретной ячейки заключается в поиске пересечения этой поверхности со сферой заданного радиуса соседства и с центром в заданной ячейке. Все ячейки поверхности, которые попали в сферу, и составляют соседство для этой ячейки. Ввиду того, что поверхность состоит из колец ячеек, задача поиска пересечения поверхности со сферой сводится к нахождению пересечения сферы с кольцами, которые являются соседними для кольца заданной ячейки. Кроме того, кольцо является симметричной фигурой, а центры всех колец, составляющих поверхность, лежат на оси Z. Поэтому все сферы, чьи центры расположены в ячейках исходного кольца, в результате пересечения с одним и тем же кольцом-соседом образуют сегменты равной длины. Таким образом, достаточно вычислить пересечения лишь одной такой сферы с кольцом-соседом, найти число ячеек, попадающих в соседство, и при помощи смещений вычислить соседство для каждой ячейки исходного кольца.

Каждое кольцо может быть рассмотрено как множество центров ячеек, лежащих на окружности (с координатой z_r^c центра своего кольца). Поэтому точки пересечения кольца-соседа и сферы могут быть найдены при помощи системы уравнений

$$\begin{cases}(x-x_s)^2+(y-y_s)^2+(z-z_s)^2=r_s^2,\\ (x-x_r)^2+(y-y_r)^2+z_r^c=r_r^2,\end{cases} \tag{1}$$

где первое уравнение — уравнение сферы в прямоугольной системе координат; второе — уравнение окружности (кольца) в пространстве; (x_s, y_s, z_s) и r_s — координаты сферы и её радиус; (x_r, y_r, z_r^c) и r_r — координаты кольца-соседа (z_r^c — константа) и его радиус; (x, y, z) — искомые точки пересечения (или единственная точка, в случае касания).

Чтобы упростить эту систему уравнений, начало системы координат сдвигается в центр кольца-соседа, что приводит к обнулению координат $x_r = y_r = z_r^c = 0$. Координаты z точек пересечения сферы и кольца-соседа (если они существуют) имеют ту же координату z, что и кольцо-сосед. Отсюда третье слагаемое первого уравнения принимает вид $(z-z_s)^2 = z_s^2$. Если в качестве ячейки, для которой вычисляется её соседство, взять ячейку, находящуюся на позиции $\pi/2$ кольца, то координата $x_s = 0$, так как центр сферы лежит в центре этой ячейки. Таким образом, уравнение (1) принимает вид

$$\begin{cases}y=(y_s^2+z_s^2+r_r^2-r_s^2)/2y_s,\\ x=\sqrt{r_r^2-y^2}.\end{cases} \tag{2}$$

Решение системы уравнений (2) может привести к следующим результатам:

1) *Точки пересечения отсутствуют*: либо сфера и кольцо-сосед не пересекаются (т. е. в кольце отсутствуют соседи для заданной ячейки), либо кольцо вписано в сферу (тогда все ячейки входят в соседство).

2) *Одна точка пересечения*: соседом заданной ячейки является ячейка, находящаяся в кольце-соседе ближе других к точке пересечения.
3) *Две точки пересечения*: все ячейки, лежащие выше точек пересечения (по координате y), являются соседями заданной ячейки.

Обобщая всё вышесказанное, сформулируем алгоритм определения соседства для ячеек кольца i в кольце j:

1) решить уравнение (2) для ячейки, лежащей на позиции $\pi/2$ исходного кольца i; найти длину сегмента $l = 2\arccos(y/r_r)r_r$, полученного в результате пересечения;
2) при помощи полученной длины l вычислить количество n ячеек, попадающих в соседство: $n = l/L_j \cdot N_j$, где L_j — длина кольца j; N_j — число ячеек в кольце j;
3) для ячейки с индексом k исходного кольца i соседями являются n последовательно расположенных ячеек кольца j, начиная с индекса $index = (k-1)N_j/N_i - n/2$, где N_i — число ячеек в кольце i.

Последний пункт алгоритма объясняется тем, что ячейки в кольце располагаются упорядоченным образом, начиная с индекса 1 (на позиции $\pi/2$ кольца) и далее по часовой стрелке. Следует отметить, что при вычислении индексов может быть получен $index < 1$ либо $index+n > N_j$. В этом случае необходимо сделать нормализацию индексов с учётом того, что индексы замкнуты в круг, т. е. после максимального индекса N идут индексы $1, 2, \ldots$ и наоборот — перед 1 идут $N, N-1, \ldots$ и т. д.

Предложенный алгоритм обладает большой скоростью выполнения и не требует значительных затрат памяти, но допускает некоторую аппроксимацию при вычислении соседств маленького радиуса ($r < 3$). Однако с увеличением радиуса эта погрешность становится незначительной для большинства задач КА-моделирования (например, при вычислении концентрации вещества в некоторой точке могут быть использованы состояния сотни смежных ячеек, где ошибка из-за попадания в соседство лишних ячеек, расположенных на краю соседства, минимальна).

3. Моделирование роста и деления

Моделирование роста бактериальной клетки осуществляется вставкой новых колец в цилиндр. При этом состояния ячеек нового кольца устанавливаются в зависимости от решаемой задачи. Например, состояния могут быть установлены случайным образом с вероятностью, которая зависит от осреднённой концентрации некоторого вещества в соседних кольцах; или после вставки нового кольца может быть произведена дополнительная операция диффузии, которая сгладит концентрацию вещества в этой области. На рис. 8 (слева) изображено три последовательных кадра, которые отображают процесс роста моделируемой бактерии *E.coli* во время самоорганизации белков MinDE. В этом примере после каждой вставки нового кольца в цилиндр сразу выполнялось несколько операций диффузии.

При делении происходит сужение центральной части клетки. Сужение имитируется последовательным уменьшением радиуса колец, находящихся в центральной части цилиндра. Как и в случае со вставкой нового кольца, необходимо решить, каким способом установить состояния ячеек кольца с новым радиусом, так как в большинстве случаев после операции сужения количество ячеек в кольце уменьшается. Какие именно кольца сужать и в какой последовательности, определяется некоторой заранее заданной функцией с условием, что после всех последовательных операций сужения центральная часть поверхности моделируемой клетки примет отражённый вид её по-

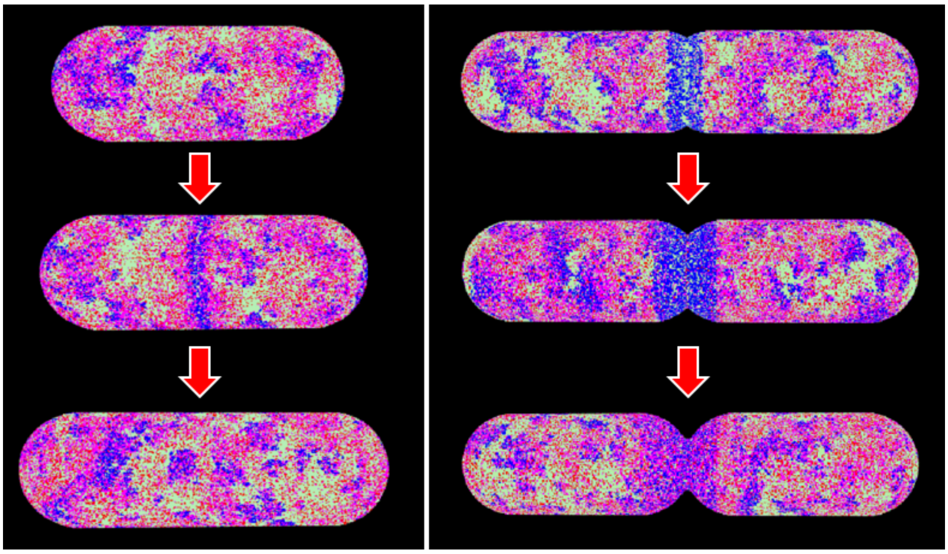


Рис. 8. Моделирование клеточного роста (слева) и деления (справа)

люсов (т. е. форму половин сфероида, обращённых друг к другу). На рис. 8 (справа) приведён пример процесса деления моделируемой клетки *E.coli*.

Заключение

Предложен метод организации массива ячеек клеточного автомата в виде иерархии (поверхность → кольца → списки ячеек), который позволяет генерировать поверхности бактериальных клеток различных форм. Основная идея метода заключается в том, что вся поверхность разбивается на кольца, состоящие из ячеек. Массив, организованный таким образом, позволяет изменять структуру моделируемой поверхности во времени (т. е. моделировать рост и деление клеток), не перестраивая при этом всю поверхность целиком. Кроме того, данный способ организации позволяет реализовать эффективный алгоритм определения соседства ячеек, не требующий значительных затрат вычислительных ресурсов, что является критически важным в задачах компьютерного моделирования.

На основе предложенного метода разработан программный комплекс на языке C++ с использованием библиотек Qt и OpenGL. При помощи этого комплекса проведены вычислительные эксперименты, которые показали, что массив ячеек, организованный таким образом, позволяет эффективно исследовать двустороннюю связь между процессами, приводящими к росту и делению бактериальных клеток, и непосредственно самими процессами роста и деления.

Следующим шагом развития метода является адаптация предложенных алгоритмов для задач моделирования поверхностей других объектов живых систем [13], например, моделирование роста и деления эукариотических клеток, моделирование динамики апикальной меристемы побега, морфогенез цианобактерий и т. п.

ЛИТЕРАТУРА

1. *Lutkenhaus J.* Assembly dynamics of the bacterial MinCDE system and spatial regulation of the *Z* ring // Annu. Rev. Biochem. 2007. No. 76. P. 539–562.
2. *Ivanov V. and Mizuuchi K.* Multiple modes of interconverting dynamic pattern formation by bacterial cell division proteins // Proc. Natl. Acad. Sci. USA. 2010. V. 107. P. 8071–8078.

3. *Hu Z. and Lutkenhaus J.* Topological regulation of cell division in Escherichia coli involves rapid pole to pole oscillation of the division inhibitor MinC under the control of MinD and MinE // Mol. Microbiol. 1999. No. 34. P. 82–90.
4. *Витвицкий А. А.* Компьютерное моделирование процесса самоорганизации бактериальной системы белков MinCDE // Мат. биол. и биоинф. 2014. Т. 9. № 2. С. 453–463.
5. *Loose M.* Spatial regulators for bacterial cell division self-organize into surface waves in vitro // Science. 2008. V. 320. P. 789–792.
6. *Vitvitsky A.* Cellular automata simulation of self-organization in the bacterial MinCDE system // Bull. Nov. Comp. Center, Comp. Sci. 2014. V. 36. P. 103–113.
7. *Bonny M., Fischer-Friedrich E., Loose M., et al.* Membrane binding of MinE allows for a comprehensive description of min-protein pattern formation // PLOS Comput. Biol. 2013. V. 9. No. 12. P. 1–12.
8. *Бандман О. Л., Киреева А. Е.* Стохастическое клеточно-автоматное моделирование колебаний и автоволн в реакционно-диффузионных системах // СибЖВМ. 2015. Т. 18. № 3. С. 251–269.
9. *Kireeva A.* Parallel implementation of totalistic Cellular Automata model of stable patterns formation // LNCS. 2013. V. 7979. P. 347–360.
10. Simulating Complex Systems by Cellular Automata. Understanding Complex Systems / eds. A. G. Hoekstra, J. Kroc., P. M. A. Sloot. Berlin: Springer, 2010. 385 p.
11. *Бандман О. Л.* Мелкозернистый параллелизм в вычислительной математике // Программирование. 2001. № 4. С. 1–18.
12. *Bandman O.* Cellular automata diffusion models for multicomputer implementation // Bull. Nov. Comp. Center, Computer Sci. 2014. V. 36. P. 21–31.
13. *Витвицкий А. А.* Клеточные автоматы с динамической структурой для моделирования роста биологических тканей // СибЖВМ. 2014. Т. 17. № 4. С. 315–328.

REFERENCES

1. *Lutkenhaus J.* Assembly dynamics of the bacterial MinCDE system and spatial regulation of the *Z* ring. Annu. Rev. Biochem., 2007, no. 76, pp. 539–562.
2. *Ivanov V. and Mizuuchi K.* Multiple modes of interconverting dynamic pattern formation by bacterial cell division proteins. Proc. Natl. Acad. Sci. USA, 2010, vol. 107, pp. 8071–8078.
3. *Hu Z. and Lutkenhaus J.* Topological regulation of cell division in Escherichia coli involves rapid pole to pole oscillation of the division inhibitor MinC under the control of MinD and MinE. Mol. Microbiol., 1999, no. 34, pp. 82–90.
4. *Vitvitsky A. A.* Komp'yuternoe modelirovanie protsessa samoorganizatsii bakterial'noy sistemy belkov MinCDE [Computer simulation of self-organization in the bacterial MinCDE system]. Math. biol. i bioinf., 2014, vol. 9, no. 2, pp. 453–463. (in Russian)
5. *Loose M.* Spatial regulators for bacterial cell division self-organize into surface waves in vitro. Science, 2008, vol. 320, pp. 789–792.
6. *Vitvitsky A.* Cellular automata simulation of self-organization in the bacterial MinCDE system. Bull. Nov. Comp. Center, Comp. Sci., 2014, vol. 36, pp. 103–113.
7. *Bonny M., Fischer-Friedrich E., Loose M., et al.* Membrane binding of MinE allows for a comprehensive description of min-protein pattern formation. PLOS Comput. Biol., 2013, vol. 9, no. 12, pp. 1–12.
8. *Bandman O. L., Kireeva A. E.* Stokhasticheskoe kletochno-avtomatnoe modelirovanie kolebaniy i avtovoln v reaktsionno-diffuzionnykh sistemakh [Stochastic cellular automata modeling of vibrations and autowaves in reaction-diffusion systems]. Sibirskii Zhurnal Vychislitel'noi Matematiki, 2015, vol. 18, no. 3, pp. 251–269. (in Russian)

9. *Kireeva A.* Parallel implementation of totalistic Cellular Automata model of stable patterns formation. LNCS, 2013, vol. 7979, pp. 347–360.
10. Simulating Complex Systems by Cellular Automata. Understanding Complex Systems / eds. A. G. Hoekstra, J. Kroc., P. M. A. Sloot. Berlin, Springer, 2010. 385 p.
11. *Bandman O. L.* Melkozernistyy parallelizm v vychislitel'noy matematike [Fine-grained parallelism in computational mathematics]. Programmirovanie, 2001, no. 4, pp. 1–18.
12. *Bandman O.* Cellular automata diffusion models for multicomputer implementation. Bull. Nov. Comp. Center, Computer Sci., 2014, vol. 36, pp. 21–31.
13. *Vitvitsky A. A.* Kletochnye avtomaty s dinamicheskoy strukturoy dlya modelirovaniya rosta biologicheskikh tkaney [Cellular automata with a dynamic structure for simulating the biological tissues growth]. Sibirskii Zhurnal Vychislitel'noi Matematiki, 2014, vol. 17, no. 4, pp. 315–328. (in Russian)

СВЕДЕНИЯ ОБ АВТОРАХ

NIKOVA, Svetla — Ph.D., research expert in the research group COSIC in the Electrical Engineering Department of KU Leuven. E-mail: **svetla.nikova@esat.kuleuven.be**

PRENEEL, Bart — Ph.D., full professor in the research group COSIC of the Electrical Engineering Department of KU Leuven. E-mail: **bart.preneel@esat.kuleuven.be**

RIJMEN, Vincent — Ph.D., associate professor in the research group COSIC in the Electrical Engineering Department of KU Leuven. E-mail: **vincent.rijmen@kuleuven.be**

АГИЕВИЧ Сергей Валерьевич — кандидат физико-математических наук, заведующий лабораторией НИИ прикладных проблем математики и информатики Белорусского государственного университета, г. Минск. E-mail: **agievich@bsu.by**

АНДРЕЕВ Александр Андреевич — аспирант Московского государственного университета им. М. В. Ломоносова, г. Москва. E-mail: **sanchez_14@mail.ru**

БИЛЯК Иван Богданович — г. Москва. E-mail: **bil-ib@mail.ru**

ВИТВИЦКИЙ Антон Александрович — аспирант Института вычислительной математики и математической геофизики СО РАН, г. Новосибирск.
E-mail: **vitvit@ssd.sscc.ru**

ВИТКУП Валерия Александровна — аспирантка Института математики им. С. Л. Соболева СО РАН, г. Новосибирск. E-mail: **vvitkup@yandex.ru**

ГОРОДИЛОВА Анастасия Александровна — аспирантка Института математики им. С. Л. Соболева СО РАН, г. Новосибирск. E-mail: **gorodilova@math.nsc.ru**

ЖАРКОВА Анастасия Владимировна — кандидат физико-математических наук, доцент кафедры теоретических основ компьютерной безопасности и криптографии Саратовского государственного университета имени Н. Г. Чернышевского, г. Саратов.
E-mail: **VAnastasiyaV@gmail.com**

КАМЛОВСКИЙ Олег Витальевич — г. Москва. E-mail: **ov-kam@yandex.ru**

КОЗИН Игорь Викторович — доктор физико-математических наук, профессор, профессор кафедры экономической кибернетики Запорожского национального университета, г. Запорожье. E-mail: **ainc00@gmail.com**

КОЛОМЕЕЦ Николай Александрович — кандидат физико-математических наук, научный сотрудник Института математики им. С. Л. Соболева СО РАН, ассистент Новосибирского государственного университета, г. Новосибирск.
E-mail: **nkolomeec@gmail.com**

КУРАПОВ Сергей Всеволодович — кандидат физико-математических наук, доцент, доцент кафедры математического моделирования Запорожского национального университета, г. Запорожье. E-mail: **lilili5050@rambler.ru**

НАЗАРОВ Максим Николаевич — старший преподаватель кафедры ВМ-1 Национального исследовательского университета «МИЭТ», г. Москва.
E-mail: **Nazarov-Maximilian@yandex.ru**

НОСОВ Юрий Леонидович — г. Липецк. E-mail: **yl.nosov@yandex.ru**

ПОЛЮГА Светлана Игоревна — ассистент кафедры математического анализа Запорожского национального университета, г. Запорожье. E-mail: **veta99@mail.ru**

РОМАНЬКОВ Виталий Анатольевич — доктор физико-математических наук, профессор Омского государственного университета им. Ф. М. Достоевского, г. Омск. E-mail: **romankov48@mail.ru**

ТОКАРЕВА Наталья Николаевна — кандидат физико-математических наук, старший научный сотрудник Института математики им. С. Л. Соболева СО РАН, доцент Новосибирского государственного университета, г. Новосибирск.
E-mail: **tokareva@math.nsc.ru**

ШУШУЕВ Георгий Иннокентьевич — аспирант Института математики им. С. Л. Соболева СО РАН, ассистент Новосибирского государственного университета, г. Новосибирск. E-mail: **shyshyev@gmail.com**