МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2011 №4(14)

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 надежности, интеллектуальных системах.

Периодичность выхода журнала: 4 номера в год.

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

Подписано к печати 29.11.2011.
Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 12,96. Уч.-изд. л. 14,53. Тираж 300 экз.

Издательство ТГУ. 634029, Томск, ул. Никитина, 4
Отпечатано в типографии ТПУ.

СОДЕРЖАНИЕ

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

DOI 10.17223/20710410/14/1

УДК 519.7

КОНСТРУКЦИЯ МАКСИМАЛЬНОГО КЛОНА ТОЧЕЧНЫХ ФУНКЦИЙ НА ПОЛУРЕШЁТКЕ ИНТЕРВАЛОВ

Н. Г. Парватов

Национальный исследовательский Томский государственный университет, г. Томск, Россия

E-mail: parvatov@mail.tsu.ru

В связи с задачей описания клонов точечных и минимальных точечных функций на верхней полурешётке предлагается конструкция максимальных по включению таких клонов на полурешётке интервалов решётки.

Ключевые слова: *клон, верхняя полурешётка, полурешётка интервалов, решётка интервалов, точечная функция, минимальная точечная функция.*

1. Формулировка результата

Пусть множество L, упорядоченное отношением $\leqslant$, является верхней полурешёткой, но не решёткой [1, 2]. Это означает, что в множестве L для любых двух элементов a и b имеется точная верхняя грань $a + b$, а точной нижней грани $a \cdot b$ может не существовать. Полурешётка называется *точечной*, если всякий её элемент является суммой некоторых минимальных элементов полурешётки.

Основным объектом изучения являются функции $f : L^n \to L$ при $n = 1, 2, \ldots$, множество которых обозначается через P_L. Функция f из P_L, зависящая от n переменных, называется *монотонной*, если она сохраняет упорядочение $\leqslant$, то есть если для любых наборов a и b из L^n выполняется импликация

$$a \leqslant b \Rightarrow f(a) \leqslant f(b),$$

где наборы в L^n сравниваются покомпонентно. Функция f называется *точечной*, если для любого набора a из L^n выполняется равенство

$$f(a) = \sum_x f(x).$$

Здесь суммирование выполняется в полурешётке L по всем (для монотонной функции f достаточно — по некоторым) наборам x из L_0^n, таким, что $x \leqslant a$, где L_0 — множество минимальных элементов полурешётки L. Точечная функция, очевидно, монотонна. Точечная функция, сохраняющая множество L_0, называется *минимальной точечной*. Как видно, точечная функция $f : L^n \to L$ однозначно определяется своим ограничением $f' : L_0^n \to L$. Принято называть f точечным расширением функции f' и обе эти функции обозначать одинаково.

Классы всех точечных и минимальных точечных функций обозначаются через T_L и $\min T_L$ соответственно. Эти классы вместе с некоторыми другими классами полурешёточных функций введены в [3] для описания асинхронных управляющих систем,

обладающих заданным динамическим поведением, то есть отвечающих заданными изменениями выходных состояний на заданные изменения входных. Основные классы функций на полурешётке изучались в [4–7]], в том числе в [6, 7] рассматривались проблемы полноты. В работе [4] сформулирована задача описания клонов (замкнутых классов с селекторами) в множествах точечных и минимальных точечных функций, показано, что всякий клон точечных (минимальных точечных) функций можно расширить до некоторого максимального по включению такого клона и множество последних конечно. В данной работе построены примеры максимальных таких клонов на полурешётке интервалов, введённой впервые в [4] и определяемой ниже.

Пусть множество E является решёткой с упорядочением $\preccurlyeq$ и операциями $\vee$ и $\wedge$ для взятия точных верхних и нижних граней [1, 2]. Интервалом решётки E будем называть пару $[a, b]$ её элементов a и b, таких, что $a \preccurlyeq b$. Интервал $[a, a]$ будем отождествлять с элементом a. Обозначим через $\mathrm{in}(E, \preccurlyeq)$ множество всех интервалов и определим для них упорядочение $\preccurlyeq$ и операции $\vee$ и $\wedge$ покомпонентно:

$$[a, b] \preccurlyeq [c, d] \Leftrightarrow (a \preccurlyeq c \;\&\; b \preccurlyeq d),\ [a, b] \vee [c, d] = [a \vee c, b \vee d],\ [a, b] \wedge [c, d] = [a \wedge c, b \wedge d],$$

где a, b, c, d — элементы из E, такие, что $a \preccurlyeq b$ и $c \preccurlyeq d$. Множество $\mathrm{in}(E, \preccurlyeq)$ становится таким образом решёткой. Оно является также верхней полурешёткой с упорядочением $\leqslant$, операцией $+$ для взятия точной верхней грани и частичной операцией $\cdot$ для взятия точной нижней грани, определёнными так:

$$[a, b] \leqslant [c, d] \Leftrightarrow (a \preccurlyeq c \;\&\; d \preccurlyeq b),\ [a, b] + [c, d] = [a \wedge c, b \vee d],\ [a, b] \cdot [c, d] = [a \vee c, b \wedge d],$$

где a, b, c, d — элементы из E, такие, что $a \preccurlyeq b$ и $c \preccurlyeq d$, а также $a \vee c \preccurlyeq b \wedge d$ в последнем случае. Построенные алгебраические системы $(L, \preccurlyeq)$ и $(L, \leqslant)$, где $L = \mathrm{in}(E, \preccurlyeq)$, называются соответственно *решёткой* и *полурешёткой интервалов решётки* $(E, \preccurlyeq)$.

Пусть Π_L — множество всех предикатов $p : L^n \to \{\text{И}, \text{Л}\}$ при $n = 1, 2, \ldots$ Для любого множества или набора Y предикатов из Π_L будем обозначать через $\mathrm{pol}_L(Y)$ клон всех функций из P_L, сохраняющих все предикаты из Y. Имеет место

Теорема 1. Пусть $(L, \preccurlyeq)$ — решётка и $(L, \leqslant)$ — полурешётка интервалов решётки $(E, \preccurlyeq)$. Клоны $\mathrm{pol}_L(\preccurlyeq, \leqslant)$ и $\mathrm{pol}_L(\preccurlyeq, \leqslant, E)$ являются максимальными по включению среди клонов, являющихся подмножествами классов T_L и $\min T_L$ соответственно.

Доказательству теоремы 1 предпошлём несколько вспомогательных утверждений — лемму 1 и следствия 1 и 2.

2. Вспомогательные утверждения

В соответствии со сказанным множество $L = \mathrm{in}(E, \preccurlyeq)$ интервалов решётки $(E, \preccurlyeq)$ будем рассматривать как полурешётку с упорядочением $\leqslant$ и одновременно как решётку с упорядочением $\preccurlyeq$. Для любого интервала $a = [a_1, a_2]$ из L положим $\mathrm{l}\, a = a_1$ и $\mathrm{r}\, a = a_2$. Положим также $\mathrm{l}\, a = (\mathrm{l}\, a_1, \ldots, \mathrm{l}\, a_n)$ и $\mathrm{r}\, a = (\mathrm{r}\, a_1, \ldots, \mathrm{r}\, a_n)$ для любого набора $a = (a_1, \ldots, a_n)$ из множества L^n, которое, таким образом, можно рассматривать как решётку и полурешётку интервалов решётки E^n. Отметим, что в L^n неравенство $a \leqslant b$ равносильно паре соотношений $\mathrm{l}\, b \preccurlyeq \mathrm{l}\, a$ и $\mathrm{r}\, a \preccurlyeq \mathrm{r}\, b$, а неравенство $a \preccurlyeq b$ — паре $\mathrm{l}\, a \preccurlyeq \mathrm{l}\, b$ и $\mathrm{r}\, a \preccurlyeq \mathrm{r}\, b$. Множество E^n является множеством минимальных элементов полурешётки L^n (упорядоченной отношением $\leqslant$). При этом каждый набор a из L^n представи́м суммой $a = \mathrm{l}\, a + \mathrm{r}\, a$ наборов $\mathrm{l}\, x$ и $\mathrm{l}\, y$ из E^n. В частности, полурешётка L^n точечная.

Лемма 1. Пусть $(L, \preccurlyeq)$ — решётка, $(L, \leqslant)$ — полурешётка интервалов решётки $(E, \preccurlyeq)$ и g — функция из P_L от n переменных.

1. Если функция g принадлежит клону $\mathrm{pol}_L(\preccurlyeq, \leqslant)$, то в множестве P_E найдутся функции g_l и g_r от n переменных, такие, что

1) функции g_l и g_r принадлежат клону $\mathrm{pol}_E(\preccurlyeq)$;
2) $g_\mathrm{l}(x) \preccurlyeq g_\mathrm{r}(x)$ для всех x из E^n;
3) $g(x) = g_\mathrm{l}(\mathrm{l}\,x) + g_\mathrm{r}(\mathrm{r}\,x)$ для всех x из L^n;
4) $g_\mathrm{l}(\mathrm{l}\,x) = \mathrm{l}\,g(x) = \mathrm{l}\,g(\mathrm{l}\,x)$ и $g_\mathrm{r}(\mathrm{r}\,x) = \mathrm{r}\,g(x) = \mathrm{r}\,g(\mathrm{r}\,x)$ для любого набора x из L^n (в частности, функции g_l и g_r однозначно определяются функцией g).

2. Обратно, если для функций g_l и g_r из P_E от n переменных выполняются условия 1–3, то для них выполняется и условие 4 и функция g принадлежит клону $\mathrm{pol}(\preccurlyeq, \leqslant)$.

3. Функция g из клона $\mathrm{pol}(\preccurlyeq, \leqslant)$ принадлежит клону $\mathrm{pol}_L(\preccurlyeq, \leqslant, E)$ тогда и только тогда, когда функции g_l и g_r, определённые условиями 1–3, совпадают.

Доказательство. Докажем первое утверждение леммы. Пусть функция g принадлежит клону $\mathrm{pol}_L(\preccurlyeq, \leqslant)$. Тогда для любого набора x из множества L^n выполняются неравенства $\mathrm{l}\,x \preccurlyeq x \preccurlyeq \mathrm{r}\,x$, а в силу монотонности функции g относительно упорядочения $\preccurlyeq$ выполняются также неравенства $g(\mathrm{l}\,x) \preccurlyeq g(x) \preccurlyeq g(\mathrm{r}\,x)$, из которых следует, что $g(x) \leqslant g(\mathrm{l}\,x) + g(\mathrm{r}\,x)$. Из монотонности функций g и $+$ относительно упорядочения $\leqslant$ следует обратное неравенство (поскольку $\mathrm{l}\,x \leqslant x$ и $\mathrm{r}\,x \leqslant x$) и тогда

$$g(x) = g(\mathrm{l}\,x) + g(\mathrm{r}\,x).$$

Отсюда, учитывая неравенство $g(\mathrm{l}\,x) \preccurlyeq g(\mathrm{r}\,x)$, получаем

$$\mathrm{l}\,g(\mathrm{l}\,x) = \mathrm{l}\,g(x) \preccurlyeq \mathrm{r}\,g(x) = \mathrm{r}\,g(\mathrm{r}\,x).$$

Из доказанного видно, что функции g_l и g_r корректно определены четвёртым условием леммы, принадлежат клону P_E и для них выполняются второе и третье условия. Поскольку первое условие легко следует из монотонности функции g относительно упорядочения $\preccurlyeq$, первое утверждение леммы доказано.

Докажем второе утверждение. Из первых двух условий следует, что для любого набора x из L^n

$$g_\mathrm{l}(\mathrm{l}\,x) \preccurlyeq g_\mathrm{l}(\mathrm{r}\,x) \preccurlyeq g_\mathrm{r}(\mathrm{r}\,x),$$

тогда из третьего условия

$$\mathrm{l}\,g(x) = g_\mathrm{l}(\mathrm{l}\,x) \quad \text{и} \quad \mathrm{r}\,g(x) = g_\mathrm{r}(\mathrm{r}\,x),$$

и, подставляя $\mathrm{l}\,x$ и $\mathrm{r}\,x$ вместо x, получаем

$$\mathrm{l}\,g(\mathrm{l}\,x) = g_\mathrm{l}(\mathrm{l}\,x) \quad \text{и} \quad \mathrm{r}\,g(\mathrm{r}\,x) = g_\mathrm{r}(\mathrm{r}\,x);$$

тем самым установлено четвёртое условие. Монотонность функции g относительно упорядочения $\preccurlyeq$ проверяется непосредственно: для любых наборов x и y из L^n, таких, что $x \preccurlyeq y$, выполняется $\mathrm{l}\,x \preccurlyeq \mathrm{l}\,y$ и $\mathrm{r}\,x \preccurlyeq \mathrm{r}\,y$, откуда с использованием первого и четвёртого условий получаем

$$\mathrm{l}\,g(x) = g_\mathrm{l}(\mathrm{l}\,x) \preccurlyeq g_\mathrm{l}(\mathrm{l}\,y) = \mathrm{l}\,g(y) \quad \text{и} \quad \mathrm{r}\,g(x) = g_\mathrm{r}(\mathrm{r}\,x) \preccurlyeq g_\mathrm{r}(\mathrm{r}\,y) = \mathrm{r}\,g(y).$$

Таким образом, $g(x) \preccurlyeq g(y)$ и функция g монотонна относительно упорядочения $\preccurlyeq$. Монотонность относительно упорядочения $\leqslant$ устанавливается тем же способом с учётом того, что неравенство $x \leqslant y$ равносильно паре соотношений $\mathrm{l}\,y \preccurlyeq \mathrm{l}\,x$ и $\mathrm{r}\,x \preccurlyeq \mathrm{r}\,y$.

Докажем третье утверждение леммы. Для этого заметим, что в силу доказанного всякая функция g, принадлежащая клону $\mathrm{pol}_L(\preccurlyeq, \leqslant)$, точечная, поскольку $g(x) = g(\mathrm{l}\, x) + g(\mathrm{r}\, x)$. Более того, она является точечным расширением функции $g_\mathrm{l} + g_\mathrm{r}$ (где функции g_l и g_r определяются условием 4). Это следует из условия 3, в силу которого $g(x) = g_\mathrm{l}(x) + g_\mathrm{r}(x) = (g_\mathrm{l} + g_\mathrm{r})(x)$ для любого набора x из множества E^n минимальных элементов полурешётки L^n. Функция g является минимальной точечной, т. е. принадлежит клону $\mathrm{pol}_E(\preccurlyeq, \leqslant, E)$, в том и только в том случае, когда функция $g_\mathrm{l} + g_\mathrm{r}$ принадлежит множеству P_E, т. е. принимает значения в множестве E при любых значениях переменных. Это равносильно тому, что функции g_l и g_r совпадают. ■

Следствие 1. Пусть $(L, \preccurlyeq)$ — решётка и $(L, \leqslant)$ — полурешётка интервалов решётки $(E, \preccurlyeq)$. Клон $\mathrm{pol}_L(\preccurlyeq, \leqslant)$ состоит из всевозможных сумм функций из $\mathrm{pol}_L(\preccurlyeq, \leqslant, E)$ с одинаковым числом переменных. В частности, для любого натурального числа n множество n-местных функций клона $\mathrm{pol}_L(\preccurlyeq, \leqslant)$ замкнуто сложением и является точечной полурешёткой с упорядочением $\leqslant$.

Доказательство. В соответствии с леммой 1 функция g из клона $\mathrm{pol}_L(\preccurlyeq, \leqslant)$ есть точечное расширение суммы $g_\mathrm{l} + g_\mathrm{r}$ функций g_l и g_r из P_E и тогда в силу коммутативности сложения является суммой точечных расширений этих функций. Но точечные расширения этих функций являются минимальными точечными функциями из клона $\mathrm{pol}_L(\preccurlyeq, \leqslant, E)$ в силу той же леммы. Первое утверждение следствия доказано. Второе следует из доказанного. ■

Далее через $\mathrm{pol}_{L,E}(\preccurlyeq)$ обозначается множество всех функций $f : E^n \to L$ при $n = 1, 2, \ldots$, сохраняющих упорядочение $\preccurlyeq$.

Следствие 2. Пусть $(L, \preccurlyeq)$ — решётка и $(L, \leqslant)$ — полурешётка интервалов решётки $(E, \preccurlyeq)$. Тогда клон $\mathrm{pol}(\preccurlyeq, \leqslant)$ состоит из всевозможных точечных расширений функций из класса $\mathrm{pol}_{L,E}(\preccurlyeq)$, а клон $\mathrm{pol}_E(\preccurlyeq, \leqslant, E)$ — из всевозможных точечных расширений функций из клона $\mathrm{pol}_E(\preccurlyeq)$.

Доказательство. В силу леммы 1 функция g из клона $\mathrm{pol}_L(\preccurlyeq, \leqslant)$ является точечным расширением суммы $g_\mathrm{l} + g_\mathrm{r}$ функций g_l и g_r из P_E, причём указанная сумма монотонна относительно упорядочения $\preccurlyeq$ вслед за функциями $g_\mathrm{l}, +, g_\mathrm{r}$, то есть принадлежит классу $\mathrm{pol}_{L,E}(\preccurlyeq)$. Для (минимальной точечной) функции g из клона $\mathrm{pol}_L(\preccurlyeq, \leqslant, E)$ функции g_l и g_r совпадают между собой и с их суммой, а потому сама функция g является точечным расширение функции $g_\mathrm{l} + g_\mathrm{r} = g_\mathrm{l} = g_\mathrm{r}$ из клона $\mathrm{pol}_E(\preccurlyeq)$.

Обратно, всякую функцию G из множества $\mathrm{pol}_{L,E}(\preccurlyeq)$ можно представить в виде суммы $G = g_\mathrm{l} + g_\mathrm{r}$ двух функций $g_\mathrm{l} = \mathrm{l}\, G$ и $g_\mathrm{r} = \mathrm{r}\, G$, таких, что $g_\mathrm{l} \preccurlyeq g_\mathrm{r}$, монотонных относительно упорядочения $\preccurlyeq$ вслед за G, что проверяется непосредственно, и совпадающих в случае функции G, принадлежащей клону $\mathrm{pol}_E(\preccurlyeq)$. Тогда функция g, определённая в соответствии с третьим условием из леммы 1, принадлежит клону $\mathrm{pol}_L(\preccurlyeq, \leqslant)$ и является точечным расширением функции G. Взяв функцию G из клона $\mathrm{pol}_E(\preccurlyeq)$, получим совпадающие функции g_l и g_r из $\mathrm{pol}_E(\preccurlyeq)$, точечным расширением которых является функция g. ■

3. Доказательство теоремы 1

Максимальность клона $\mathrm{pol}_L(\preccurlyeq, \leqslant, E)$ следует из того, что клон $\mathrm{pol}_E(\preccurlyeq)$ — предполный в P_E и минимальная точечная функция из $\min T_L$ однозначно определяется своим ограничением из P_E. Приведём, однако, более общее рассуждение, охватывающее оба случая, присутствующих в формулировке теоремы.

Пусть K — клон $\operatorname{pol}_L(\preccurlyeq, \leqslant)$ (или $\operatorname{pol}_L(\preccurlyeq, \leqslant, E)$) и его удаётся расширить до клона $K' \subseteq T_L$ (соответственно до клона $K' \subseteq \min T_L$), содержащего немонотонную относительно упорядочения $\preccurlyeq$ функцию f из T_L, зависящую от n переменных. Для доказательства нужно получить противоречие.

Заметим, что немонотонную относительно упорядочения $\preccurlyeq$ функцию в клоне K' можно выбрать от одной переменной. Действительно, в рассматриваемой ситуации упорядочение $\preccurlyeq$ нарушается функцией f на паре наборов A и B из L^n, для которых выполняется неравенство $A \preccurlyeq B$, в отличие от неравенства $f(A) \preccurlyeq f(B)$. Такие наборы A и B можно выбрать уже в множестве E^n (в противном случае функция f является точечным расширением монотонной относительно упорядочения $\preccurlyeq$ функции из $\operatorname{pol}_{L,E}(\preccurlyeq)$, и тогда она сама сохраняет это упорядочение по следствию 2 вопреки её выбору). Более того, эти наборы можно выбрать отличающимися одной компонентой так, что $A = (a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n)$ и $B = (a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n)$ для некоторых элементов $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n$ и a, b из E. Тогда немонотонной относительно упорядочения $\preccurlyeq$ является одноместная функция $s(x) = f(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_n)$, принадлежащая клону K' (вслед за функцией f и подставленными в неё константами $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n$ из E). Монотонность функции s нарушается на элементах a и b из E, для которых выполняется неравенство $a \preccurlyeq b$ и не выполняется неравенство $s(a) \preccurlyeq s(b)$. Элементы a и b можно выбрать соседними в решётке E так, что в ней не существует элемента c со свойством $a \prec c \prec b$. Для дальнейшего важно, что такой выбор элементов a и b гарантирует отсутствие отличного от них элемента c в множестве E со свойством $c \leqslant a + b$. Невыполнение неравенства $s(a) \preccurlyeq s(b)$ означает, что не выполняется некоторое из неравенств

$$\mathrm{l}\, s(a) \preccurlyeq \mathrm{l}\, s(b) \quad \text{или} \quad \mathrm{r}\, s(a) \preccurlyeq \mathrm{r}\, s(b). \tag{1}$$

Пусть это будет первое. Обозначим через 0 и 1 соответственно наименьший и наибольший элементы решётки $(E, \preccurlyeq)$. Рассмотрим одноместные минимальные точечные функции t_1 и t_2 из $\min T_L$, принимающие значения 0 и 1 на элементах из E в соответствии со следующими условиями: $t_1(x) = 1$, если $b \preccurlyeq x$; $t_2(x) = 1$, если $\mathrm{l}\, s(a) \preccurlyeq x$. По следствию 2 функции t_1 и t_2 принадлежат клону $\operatorname{pol}_L(\preccurlyeq, \leqslant, E)$ и тогда клону K'. Рассмотрим функцию $g(x) = t_1(x) \vee t_2(s(x))$ из K'. Заметим, что $g(a) = g(b) = 1$, так как $t_2(s(a)) = 1$ и $t_1(b) = 1$. Вместе с тем

$$g(a + b) = [0, 1] \neq 1 = g(a) + g(b),$$

так как $t_1(a+b) = [0, 1]$ и $t_2(s(a+b)) \geqslant t_2(s(a)) + t_2(s(b)) = 0+1 = [0, 1]$. Таким образом, функция g не точечная. Получено противоречие.

Случай, когда не выполняется второе неравенство в (1), рассматривается аналогично. Теорема доказана.

4. Замечание

В работе [4] автора допущены следующие неточности: на с. 33 в выделенной формуле должно быть $\bot K = \bot \max(K, \leqslant)$ (то есть max вместо min); в следствии 5 условие $f^{-1}(0) = \bot f^{-1}(1)$ следует пополнить требованием $f^{-1}(1) \neq \varnothing$ и в теореме 10 второе условие — требованием $B^* \subseteq f^{-1}(1) \cup f^{-1}(\top)$, означающим, что функция x^B не принимает значения $\top$, когда функция f принимает значение 0.

ЛИТЕРАТУРА

1. *Биркгоф Г.* Теория решёток. М.: Наука, 1984. 568 с.
2. *Курош А. Г.* Лекции по общей алгебре. М.: Наука, 1973. 400 с.
3. *Агибалов Г. П.* Дискретные автоматы на полурешётках. Томск: Изд-во Том. ун-та, 1993. 227 с.
4. *Парватов Н. Г.* Точечные и сильно точечные функции на полурешётке // Прикладная дискретная математика. 2010. № 3. С. 22–40.
5. *Парватов Н. Г.* Об инвариантах некоторых классов квазимонотонных функций на полурешётке // Прикладная дискретная математика. 2009. № 4. С. 21–28.
6. *Парватов Н. Г.* Функциональная полнота в замкнутых классах квазимонотонных и монотонных трёхзначных функций на полурешётке // Дискрет. анализ и исслед. опер. Сер. 1. 2003. Т. 10. № 1. С. 61–78.
7. *Парватов Н. Г.* Теорема о функциональной полноте в классе квазимонотонных функций на конечной полурешётке // Дискрет. анализ и исслед. опер. Сер. 1. 2006. Т. 13. № 3. С. 62–82.

2011 Теоретические основы прикладной дискретной математики №4(14)
DOI 10.17223/20710410/14/2
УДК 519.7

ЛОКАЛЬНО ОБРАТИМЫЕ БУЛЕВЫ ФУНКЦИИ[1]

С. В. Смышляев

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

E-mail: smyshsv@gmail.com

Изучается свойство локальной обратимости булевых функций. Устанавливается ряд необходимых условий локальной обратимости, позволяющих строить классы функций, соответствующие которым кодирующие устройства не допускают локального обращения. Доказывается критерий, связывающий локальную обратимость произвольной булевой функции с определенными характеристиками булевых функций с барьером.

Ключевые слова: *функции без запрета, совершенно уравновешенные функции, барьеры булевых функций, локальная обратимость, криптография.*

Введение

При исследовании кодирующих устройств, состоящих из регистра сдвига и функции усложнения [1, 2], естественным вопросом является возможность однозначного восстановления части входной последовательности кодирующего устройства по части выходной последовательности. В частности, при использовании таких кодирующих устройств в составе фильтрующих генераторов наличие возможности получения по конечному отрезку выходной последовательности достаточно большой части символов входной последовательности делает возможным нахождение секретного ключа простым решением линейной системы уравнений. Одним из возможных вариантов формализации данного свойства функций усложнения является следующий: существование таких выходных наборов, что в случае их присутствия в выходной последовательности кодирующего устройства оставшиеся символы входной последовательности можно восстанавливать однозначно по последующим символам выходной последовательности.

Данное понятие было формализовано О. А. Логачевым как понятие локальной обратимости булевой функции в работе [3], посвященной изучению соответствующих свойств функций, линейных по последней переменной. Здесь же был получен критерий локальной обратимости порождаемых такими функциями отображений, связывающий данное понятие с так называемым возвратным свойством булевой функции [4].

В настоящей работе проводится исследование свойства локальной обратимости произвольных булевых функций. Устанавливается ряд необходимых свойств локальной обратимости, в частности совершенная уравновешенность [5 – 7] и наличие барьера [8, 9]. Доказывается критерий, связывающий локальную обратимость произвольной булевой функции с описанными в работе [10] характеристиками булевых функций с барьером. Важными следствиями полученных результатов являются методы построения булевых функций с положительными криптографическими свойствами, не допускающих локального обращения. В частности, все описанные в работе [11] методы, как следует из результатов настоящей работы, позволяют строить не допускающие локального обращения совершенно уравновешенные булевы функции.

[1]Работа поддержана РФФИ (номер проекта 09-01-00653-а).

1. Определения и предварительные результаты

Для любого натурального n множество $\{0,1\}^n$ двоичных наборов длины n будем обозначать через V_n; множество булевых функций от n переменных — через $\mathcal{F}_n$.

Пусть $n, l \in \mathbb{N}$, $f \in \mathcal{F}_n$. Рассмотрим систему булевых уравнений

$$f(x_s, x_{s+1}, \ldots, x_{s+n-1}) = y_s,\ s = 1, 2, \ldots, l. \tag{1}$$

Через f_l будем обозначать следующее отображение из V_{l+n-1} в V_l:

$$f_l(x_1, x_2, \ldots, x_{l+n-1}) = \left(f(x_1, \ldots, x_n), f(x_2, \ldots, x_{n+1}), \ldots, f(x_l, \ldots, x_{l+n-1})\right).$$

Легко видеть, что отображение f_l можно понимать как преобразование, производимое l тактами работы кодирующего устройства, полученного с помощью подключения входов булевой функции f к некоторым ячейкам двоичного регистра сдвига [1, 5].

Для $n, l \in \mathbb{N}$, всякого набора $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n-1}) \in V_{n-1}$ через $f_{R,l,n-1}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n-1})}$ будем обозначать отображение из V_l в V_l, определяемое следующим образом:

$$f_{R,l,n-1}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n-1})}(x_1, x_2, \ldots, x_l) = f_l(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n-1}, x_1, x_2, \ldots, x_l),$$

а через $f_{L,l,n-1}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n-1})}$ — отображение из V_l в V_l вида

$$f_{L,l,n-1}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n-1})}(x_1, x_2, \ldots, x_l) = f_l(x_1, x_2, \ldots, x_l, \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n-1}).$$

Отображение $f_{R,l,n-1}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n-1})}$ описывает преобразование, производимое l тактами работы кодирующего устройства с функцией усложнения f на двоичных последовательностях с фиксированным началом из $n-1$ символов.

Определение 1 [5, 12]. Булева функция $f \in \mathcal{F}_n$ называется *функцией без запрета*, если соотношение $|f_l^{-1}(\mathbf{y})| > 0$ выполняется для любого $l \in \mathbb{N}$ и любого $\mathbf{y} \in V_l$.

Определение 2 [5, 12]. Булева функция $f \in \mathcal{F}_n$ называется *совершенно уравновешенной*, если соотношение $|f_l^{-1}(\mathbf{y})| = 2^{n-1}$ выполняется для любого $l \in \mathbb{N}$ и любого $\mathbf{y} \in V_l$. Множество совершенно уравновешенных функций из $\mathcal{F}_n$ обозначим через $\mathcal{PB}_n$.

Определение 3 [5, 12]. Булева функция $f \in \mathcal{F}_n$ называется *функцией без потери информации*, если при любом $l \geqslant n$ система

$$\begin{cases} f_l(x_1, x_2, \ldots, x_{l+n-1}) = f_l(z_1, z_2, \ldots, z_{l+n-1}), \\ (x_1, x_2, \ldots, x_{n-1}) = (z_1, z_2, \ldots, z_{n-1}), \\ (x_{l+1}, x_{l+2}, \ldots, x_{l+n-1}) = (z_{l+1}, z_{l+2}, \ldots, z_{l+n-1}), \\ (x_n, x_{n+1}, \ldots, x_l) \neq (z_n, z_{n+1}, \ldots, z_l) \end{cases}$$

является несовместной.

Теорема 1 [5, 12]. Пусть $f \in \mathcal{F}_n$. Следующие утверждения эквивалентны:

1) f является функцией без запрета;
2) f является совершенно уравновешенной функцией;
3) f является функцией без потери информации.

Введем понятие барьера булевой функции, тесно связанное с понятием совершенной уравновешенности.

Определение 4 [6]. Булева функция $f \in \mathcal{F}_n$ называется функцией с правым барьером длины $b \in \mathbb{N}$, если система уравнений

$$\begin{cases} f_{R,b',n-1}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n-1})}(x_1, x_2, \ldots, x_{b'}) = f_{R,b',n-1}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n-1})}(z_1, z_2, \ldots, z_{b'}), \\ x_1 \neq z_1 \end{cases} \tag{2}$$

имеет решение для любого $b' \in \mathbb{N}$, $b' \leqslant b-1$, а система

$$\begin{cases} f_{R,b,n-1}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n-1})}(x_1, x_2, \ldots, x_b) = f_{R,b,n-1}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n-1})}(z_1, z_2, \ldots, z_b), \\ x_1 \neq z_1 \end{cases} \tag{3}$$

решений не имеет.

Булева функция $f \in \mathcal{F}_n$ называется функцией с левым барьером длины b, если функция $\overleftrightarrow{f}(x_1, x_2, \ldots, x_n) \equiv f(x_n, x_{n-1}, \ldots, x_1)$ является функцией с правым барьером длины b.

Булева функция $f \in \mathcal{F}_n$ имеет барьер, если она имеет правый или левый барьер, или оба сразу. При этом длиной барьера функции называется соответственно длина правого барьера, левого барьера или меньшая из длин барьеров.

Связь совершенной уравновешенности с наличием у функции барьера описывается следующим утверждением.

Теорема 2 [6]. Если функция имеет барьер, то она совершенно уравновешена.

Замечание 1. В работе [6] показано, что наличие барьера не является необходимым условием совершенной уравновешенности; в работе [11] приведен метод построения широких классов совершенно уравновешенных булевых функций без барьера.

Отметим, что для всех утверждений, относящихся к функциям с правым барьером, очевидным образом могут быть построены аналоги для функций с левым барьером. Далее длину правого барьера булевой функции f будем обозначать b_f^R, левого — b_f^L.

Замечание 2. Нетрудно заметить, что наличие правого барьера длины $b_f^R = 1$ (левого барьера длины $b_f^L = 1$) означает линейность функции по последнему (первому) аргументу.

Теорема 3 [10]. Для каждой функции $f \in \mathcal{F}_n$ с правым (левым) барьером можно определить величину $e_f^R \in \{0, 1, 2, \ldots, b_f^R - 1\}$ ($e_f^L \in \{0, 1, 2, \ldots, b_f^L - 1\}$), такую, что для любого $l \geqslant b_f^R - 1$ ($l \geqslant b_f^L - 1$), $\mathbf{x} \in V_{n-1}$ и любого набора $\mathbf{y} \in \mathrm{Im}(f_{R,l,n-1}^{\mathbf{x}})$ ($\mathbf{y} \in \mathrm{Im}(f_{L,l,n-1}^{\mathbf{x}})$) выполняется равенство $|{f_{R,l,n-1}^{\mathbf{x}}}^{-1}(\mathbf{y})| = 2^{e_f^R}$ ($|{f_{L,l,n-1}^{\mathbf{x}}}^{-1}(\mathbf{y})| = 2^{e_f^L}$).

Утверждение 1 [10]. Пусть $f \in \mathcal{F}_n$ имеет правый (левый) барьер. Для любых $l \geqslant b_f^R - 1$ ($l \geqslant b_f^L - 1$) и любого набора $\mathbf{x} \in V_{n-1}$ верно равенство $|\mathrm{Im}(f_{R,l,n-1}^{\mathbf{x}})| = 2^{l-e_f^R}$ ($|\mathrm{Im}(f_{L,l,n-1}^{\mathbf{x}})| = 2^{l-e_f^L}$).

Пусть $f \in \mathcal{F}_n$. Определим для всякого $m \in \mathbb{N}$ и для всякого набора $\mathbf{y} \in V_m$ следующие множества:

$$A_{\mathbf{y}}^f = \left\{\mathbf{x} \in V_{n-1} : {f_{R,m,n-1}^{\mathbf{x}}}^{-1}(\mathbf{y}) \neq \varnothing\right\}; \quad E_{\mathbf{y}}^f = \left\{\mathbf{x} \in V_{n-1} : {f_{L,m,n-1}^{\mathbf{x}}}^{-1}(\mathbf{y}) \neq \varnothing\right\}.$$

Утверждение 2 [10]. Пусть $f \in \mathcal{F}_n$ имеет правый (левый) барьер. Для любого $l \geqslant b_f^R - 1$ ($l \geqslant b_f^L - 1$) и любого набора $\mathbf{y} \in V_l$ верно равенство $|A_{\mathbf{y}}^f| = 2^{n-1-e_f^R}$ ($|E_{\mathbf{y}}^f| = 2^{n-1-e_f^L}$).

Теорема 4 [10]. Пусть $f(x_1, x_2, \ldots, x_n) \in \mathcal{F}_n$ имеет правый (левый) барьер. Тогда

1) $e_f^R = b_f^R - 1$ ($e_f^L = b_f^L - 1$) тогда и только тогда, когда функция f не зависит существенно от переменных $x_{n-b_f^R+2}, x_{n-b_f^R+3}, \ldots, x_n$ и линейна по переменной $x_{n-b_f^R+1}$ (не зависит существенно от переменных $x_1, x_2, \ldots, x_{b_f^L-1}$ и линейна по переменной $x_{b_f^L}$);
2) $e_f^R = 0$ ($e_f^L = 0$) тогда и только тогда, когда $b_f^R = 1$ ($b_f^L = 1$).

Будем обозначать через V_∞ множество всех бесконечных (вправо) двоичных последовательностей, т. е. последовательностей вида $\overline{\mathbf{z}} = (z_1, z_2, \ldots)$, где $z_i \in \{0, 1\}$, $i = 1, 2, \ldots$

Для всякой функции $f \in \mathcal{F}_n$ через f_∞ будем обозначать отображение из V_∞ в V_∞, заданное в соответствии со следующим правилом:

$$f_\infty(x_1, x_2, \ldots) = (f(x_1, x_2, \ldots, x_n), f(x_2, x_3, \ldots, x_{n+1}), \ldots).$$

Кроме того, для всякого набора $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n-1}) \in V_{n-1}$ введем следующее отображение из V_∞ в V_∞:

$$f_{R,\infty,n-1}^{(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_{n-1})}(x_1, x_2, \ldots) \equiv f_\infty(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{n-1}, x_1, x_2, \ldots);$$

для всякой последовательности $\overline{\mathbf{z}} \in V_\infty$ введем следующее множество:

$$A_{\overline{\mathbf{z}}}^f = \left\{\mathbf{x} \in V_{n-1} : {f_{R,\infty,n-1}^{\mathbf{x}}}^{-1}(\overline{\mathbf{z}}) \neq \varnothing\right\}.$$

Замечание 3. Легко видеть, что для любых $f \in \mathcal{F}_n$, $m, l \in \mathbb{N}$, $m < l$ и $\overline{\mathbf{z}} = (z_1, z_2, \ldots) \in V_\infty$ выполняется цепочка вложений $A_{\overline{\mathbf{z}}}^f \subseteq A_{(z_1,z_2,\ldots,z_m,\ldots,z_l)}^f \subseteq A_{(z_1,z_2,\ldots,z_m)}^f$, а также вложение $E_{(z_l,z_{l-1},\ldots,z_m,z_{m-1},\ldots,z_1)}^f \subseteq E_{(z_m,z_{m-1},\ldots,z_1)}^f$.

Определение 5 [3]. Пусть $f \in \mathcal{F}_n$. Обозначим для всякого $m \in \mathbb{N}$ через $T_f^R(m)$ множество наборов $\mathbf{y} \in \mathrm{Im}(f_m)$, для которых выполняется следующее условие: для всякой последовательности $\overline{\mathbf{z}} \in V_\infty$, такой, что $(\mathbf{y}|\overline{\mathbf{z}}) \in \mathrm{Im}(f_\infty)$, существует единственная последовательность $\overline{\mathbf{x}} \in V_\infty$, для которой множество $\{\tilde{\mathbf{x}} \in V_m : f_\infty(\tilde{\mathbf{x}}|\overline{\mathbf{x}}) = (\mathbf{y}|\overline{\mathbf{z}})\}$ непусто.

Функция $f \in \mathcal{F}_n$ называется *слабо локально обратимой вправо* (далее будем для сокращения записей использовать термин «локально обратимая функция»), если множество $T_f^R = \bigcup\limits_{i=1}^{\infty} T_f^R(i)$ непусто.

Для всякого натурального n множество локально обратимых булевых функций от n переменных будем обозначать через $\mathcal{LI}_n$.

Замечание 4. Нетрудно видеть, что для всякой функции $f \in \mathcal{F}_n$, любых $m, r \in \mathbb{N}$ и любых наборов $\mathbf{y} \in V_m$, $\mathbf{y}' \in V_r$ верно, что если $\mathbf{y} \in T_f^R$ и $(\mathbf{y}'|\mathbf{y}) \in \mathrm{Im}(f_{m+r})$, то $(\mathbf{y}'|\mathbf{y}) \in T_f^R$.

Определение 6. Функция $f \in \mathcal{F}_n$ называется *слабо локально обратимой влево*, если функция $\overleftrightarrow{f}$ является слабо локально обратимой вправо.

Определение 7. Функция $f \in \mathcal{F}_n$ называется *сильно локально обратимой вправо*, если для некоторого натурального p верно, что множество T_f^R включает в себя все наборы длины не меньше p, то есть выполняется вложение $\bigcup\limits_{i=p}^{\infty} V_i \subseteq T_f^R$.

Определение 8. Функция $f \in \mathcal{F}_n$ называется *сильно локально обратимой влево*, если функция $\overleftrightarrow{f}$ является сильно локально обратимой вправо.

Для всякого n множество сильно локально обратимых вправо (влево) булевых функций от n переменных будем обозначать через $\mathcal{PLI}_n^R$ (соответственно $\mathcal{PLI}_n^L$).

2. Основные результаты

Докажем сначала несколько вспомогательных утверждений.

Лемма 1. Пусть на множестве двоичных наборов конечной длины определен некоторый предикат $Q\colon \bigcup\limits_{i=1}^{\infty} V_i \mapsto \{\mathrm{T}, \mathrm{F}\}$, для которого выполнены следующие свойства:

1) для всяких $m \in \mathbb{N}$, $\mathbf{z} \in V_m$ и $b \in \{0,1\}$ верно, что если $Q(\mathbf{z}) = \mathrm{F}$, то $Q(\mathbf{z}|b) = \mathrm{F}$;
2) для всякого $m \in \mathbb{N}$ существует набор $\mathbf{z} \in V_m$, такой, что $Q(\mathbf{z}) = \mathrm{T}$.

Тогда существует последовательность $\overline{\mathbf{z}}^* = (z_1^*, z_2^*, \ldots) \in V_\infty$, такая, что $Q(z_1^*, z_2^*, \ldots, z_m^*) = \mathrm{T}$ для любого m.

Доказательство. Для всех $m \in \mathbb{N}$ и $\mathbf{z} \in V_m$ через $W(\mathbf{z})$ обозначим множество $\{\mathbf{z}\} \cup \{\mathbf{w} \in \bigcup\limits_{i=m+1}^{\infty} V_i : \mathbf{w} = (\mathbf{z}|\mathbf{u}), \mathbf{u} \in \bigcup\limits_{i=1}^{\infty} V_i\}$; для каждого $l \in \mathbb{N}$ рассмотрим множество $M_l = \{\mathbf{z} \in V_l : Q(\mathbf{z}) = \mathrm{T}\}$. Для каждого $m \in \mathbb{N}$ введем множество U_m, состоящее из наборов $\mathbf{z} \in V_m$, таких, что для всякого $l \geqslant m$ выполнено $W(\mathbf{z}) \cap M_l \neq \varnothing$.

Пусть $\mathbf{z} \in U_m$. Докажем, что существует $b \in \{0,1\}$, при котором $(\mathbf{z}|b) \in U_{m+1}$. Предположим противное. Если $(\mathbf{z}|0) \notin U_{m+1}$, то при некотором $l' \geqslant m+1$ выполнено $W(\mathbf{z}|0) \cap M_{l'} = \varnothing$, и тогда при любом $l \geqslant l'$, как следует из условия 1, $W(\mathbf{z}|0) \cap M_l = \varnothing$. Аналогично, если $(\mathbf{z}|1) \notin U_{m+1}$, то при некотором $l'' \geqslant m+1$ верно, что для любого $l \geqslant l''$ выполнено $W(\mathbf{z}|1) \cap M_l = \varnothing$. Таким образом, для любого $l \geqslant l''' = \max\{l', l''\}$ верно $\varnothing = (W(\mathbf{z}|0) \cup W(\mathbf{z}|1)) \cap M_l = W(\mathbf{z}) \cap M_l$, что противоречит условию $\mathbf{z} \in U_m$. Из условия 2 и аналогичных рассуждений следует, что множество U_1 непусто.

Пусть $(z_1^*) \in U_1$, $(z_1^*, z_2^*) \in U_2$, $(z_1^*, z_2^*, z_3^*) \in U_3, \ldots$ Нетрудно проверить, что построенная таким образом последовательность $\overline{\mathbf{z}}^* = (z_1^*, z_2^*, \ldots) \in V_\infty$ — искомая. ■

Утверждение 3. Пусть $f \in \mathcal{PB}_n$. Тогда для любых $m \in \mathbb{N}$, $\mathbf{y} \in V_m$, $\overline{\mathbf{z}} \in V_\infty$ выполняются следующие неравенства:

1) $f_\infty^{-1}(\mathbf{y}|\overline{\mathbf{z}}) \neq \varnothing$;
2) $E_{\mathbf{y}}^f \cap A_{\overline{\mathbf{z}}}^f \neq \varnothing$.

Доказательство. Зафиксируем $m \in \mathbb{N}$, $\mathbf{y} \in V_m$, $\overline{\mathbf{z}} \in V_\infty$ и рассмотрим предикат Q, принимающий на наборе $\mathbf{x} \in V_l$ значение T в том и только в том случае, когда либо $l \leqslant n-1$, либо набор $f_{l-n+1}(\mathbf{x})$ равен набору из первых $l-n+1$ символов последовательности $(\mathbf{y}|\overline{\mathbf{z}})$. Нетрудно проверить, что для предиката Q выполняются условия леммы 1 и, следовательно, существует последовательность $\overline{\mathbf{x}} = (x_1, x_2, \ldots) \in {f_\infty}^{-1}(\mathbf{y}|\overline{\mathbf{z}})$. При этом, очевидно, набор $(x_{m+1}, x_{m+2}, \ldots, x_{m+n-1})$ лежит в пересечении множеств $E_{\mathbf{y}}^f$ и $A_{\overline{\mathbf{z}}}^f$. ■

Утверждение 4. Пусть $f \in \mathcal{F}_n$, $\overline{\mathbf{y}} = (y_1, y_2, \ldots) \in V_\infty$. Тогда $A_{\overline{\mathbf{y}}}^f = \bigcap\limits_{i=1}^{\infty} A_{(y_1, y_2, \ldots, y_i)}^f$.

Доказательство. Вложение $A_{\overline{\mathbf{y}}}^f \subseteq \bigcap\limits_{i=1}^{\infty} A_{(y_1, y_2, \ldots, y_i)}^f$ следует из замечания 3.

Чтобы доказать вложение $\bigcap\limits_{i=1}^{\infty} A_{(y_1, y_2, \ldots, y_i)}^f \subseteq A_{\overline{\mathbf{y}}}^f$, рассмотрим произвольный набор $\tilde{\mathbf{x}} \in \bigcap\limits_{i=1}^{\infty} A_{(y_1, y_2, \ldots, y_i)}^f$ и предикат Q, такой, что для произвольного $\mathbf{x} \in V_l$ равенство $Q(\mathbf{x}) = \mathrm{T}$ выполняется тогда и только тогда, когда $f_{R,l,n-1}^{\tilde{\mathbf{x}}}(\mathbf{x}) = (y_1, y_2, \ldots, y_l)$. Применяя лемму 1, приходим к существованию последовательности $\overline{\mathbf{x}} \in {f_{R,\infty,n-1}^{\tilde{\mathbf{x}}}}^{-1}(\overline{\mathbf{y}})$. Таким образом, $\tilde{\mathbf{x}} \in A_{\overline{\mathbf{y}}}^f$, что завершает доказательство утверждения. ■

Лемма 2. Пусть $f \in \mathcal{F}_n$. Функция f имеет правый барьер тогда и только тогда, когда для всякого набора $\mathbf{x} \in V_{n-1}$ и для всякой последовательности $\overline{\mathbf{z}} \in \mathrm{Im}(f^{\mathbf{x}}_{R,\infty,n-1})$ выполнено равенство $|{f^{\mathbf{x}}_{R,\infty,n-1}}^{-1}(\overline{\mathbf{z}})| = 1$.

Доказательство. Необходимость очевидна. Для доказательства достаточности предположим противное: функция f не имеет правого барьера; в таком случае для любого l существуют наборы $(y_1, y_2, \ldots, y_l) \in V_l$, $(x_1, x_2, \ldots, x_{n-1}) \in V_{n-1}$, для которых выполнено

$$\begin{cases} f(x_1, x_2, \ldots, x_{n-1}, 0) = f(x_1, x_2, \ldots, x_{n-1}, 1) = y_1, \\ (x_2, x_2, \ldots, x_{n-1}, 0) \in A^f_{(y_2, y_3, \ldots, y_l)}, \\ (x_2, x_2, \ldots, x_{n-1}, 1) \in A^f_{(y_2, y_3, \ldots, y_l)}. \end{cases} \tag{4}$$

Введем предикат Q, равный T на всех наборах длины меньше n и определенный следующим образом на наборах длины $n{-}1{+}l$, $l = 1, 2\ldots$: $Q(x_1, x_2, \ldots, x_{n-1}, y_1, y_2, \ldots, y_l) =$ $=$ T тогда и только тогда, когда выполнена система (4). Можно проверить, что для определенного таким образом предиката Q выполнены все условия леммы 1 и, значит, существует набор $\mathbf{x} = (x_1, x_2, \ldots, x_{n-1}) \in V_{n-1}$ и последовательность $\overline{\mathbf{y}} = (y_1, y_2, \ldots) \in$ $\in V_\infty$, такие, что для любого l для них выполнена система (4). С учетом утверждения 4 это означает, что $|{f^{\mathbf{x}}_{R,\infty,n-1}}^{-1}(\overline{\mathbf{y}})| \geqslant 2$. Полученное противоречие завершает доказательство достаточности. ■

Следствие 1. Если $f \in \mathcal{F}_n$ не имеет правого барьера, то существует набор $\mathbf{x}^* \in V_{n-1}$ и двоичные последовательности $\overline{\mathbf{x}}', \overline{\mathbf{x}}'', \overline{\mathbf{y}}^* \in V_\infty$, для которых выполняется равенство

$$f_\infty(\mathbf{x}^*|0|\overline{\mathbf{x}}') = f_\infty(\mathbf{x}^*|1|\overline{\mathbf{x}}'') = \overline{\mathbf{y}}^*. \tag{5}$$

Докажем утверждение о необходимом условии обратимости.

Теорема 5. Если $f \in \mathcal{LI}_n$, то f имеет правый барьер.

Доказательство. Предположим противное: пусть некоторая функция f, не имеющая правого барьера, является локально обратимой.

Зафиксируем $m \in \mathbb{N}$ и $\mathbf{y} \in V_m$, такие, что $\mathbf{y} \in T^R_f$; зафиксируем также набор $\mathbf{x}^* \in V_{n-1}$ и двоичные последовательности $\overline{\mathbf{x}}', \overline{\mathbf{x}}'' \in V_\infty$, для которых выполняется условие (5) следствия 1.

Рассмотрим произвольный набор $\mathbf{x} = (x_1, x_2, \ldots, x_{m+n-1})$, лежащий в (непустом по определению) множестве $f^{-1}_m(\mathbf{y})$. Очевидно, что для последовательности $\overline{\mathbf{z}} = f_\infty(x_{m+1},$ $x_{m+2}, \ldots, x_{m+n-1}|\mathbf{x}^*|0|\overline{\mathbf{x}}')$ выполняется вложение $(\mathbf{y}|\overline{\mathbf{z}}) \in \mathrm{Im}(f_\infty)$. При этом также выполняется равенство $f_\infty(\mathbf{x}|\mathbf{x}^*|0|\overline{\mathbf{x}}') = f_\infty(\mathbf{x}|\mathbf{x}^*|1|\overline{\mathbf{x}}'') = (\mathbf{y}|\overline{\mathbf{z}})$, противоречащее, очевидно, принадлежности набора $\mathbf{y}$ множеству T^R_f. Полученное противоречие завершает доказательство теоремы. ■

Следствие 2. Для любого $n \in \mathbb{N}$ выполнено вложение $\mathcal{LI}_n \subseteq \mathcal{PB}_n$.

Учитывая результат теоремы 5, далее при изучении класса локально обратимых функций будем рассматривать только функции с правым барьером. Как следует из утверждения 3, для любой функции f с правым барьером, любого набора $\mathbf{y}$ и любой последовательности $\overline{\mathbf{z}}$ выполняется неравенство $|E^f_{\mathbf{y}} \cap A^f_{\overline{\mathbf{z}}}| \geqslant 1$.

Лемма 3. Пусть $f \in \mathcal{F}_n$ имеет правый барьер. Для всяких $m \in \mathbb{N}$ и $\mathbf{y} \in V_m$ верно, что $\mathbf{y} \in T^R_f$ тогда и только тогда, когда для любой последовательности $\overline{\mathbf{z}} \in V_\infty$ выполняется равенство $|E^f_{\mathbf{y}} \cap A^f_{\overline{\mathbf{z}}}| = 1$.

Доказательство. Необходимость очевидна. Докажем достаточность. Пусть для набора $\mathbf{y} \in V_m$ и любой последовательности $\overline{\mathbf{z}} \in V_\infty$ выполнено $|E^f_{\mathbf{y}} \cap A^f_{\overline{\mathbf{z}}}| = 1$. Зафиксируем $\overline{\mathbf{z}} \in V_\infty$ и тот единственный набор $\mathbf{x}$, который лежит в множестве $E^f_{\mathbf{y}} \cap A^f_{\overline{\mathbf{z}}}$. По лемме 2 выполнено равенство $|{f^{\mathbf{x}}_{R,\infty,n-1}}^{-1}(\overline{\mathbf{z}})| = 1$. Таким образом, $\mathbf{y} \in T^R_f$. ■

Лемма 4. Пусть $f \in \mathcal{F}_n$ имеет правый барьер. Для всякого $m \in \mathbb{N}$ и $\mathbf{y} \in V_m$ верно, что $\mathbf{y} \in T^R_f$ тогда и только тогда, когда существует $M \in \mathbb{N}$, такое, что для любого набора $\mathbf{z} \in V_M$ выполняется равенство $|E^f_{\mathbf{y}} \cap A^f_{\mathbf{z}}| = 1$.

Доказательство. Достаточность очевидным образом следует из леммы 3, так как для всякой последовательности $\overline{\mathbf{z}} = (z_1, z_2, \ldots) \in V_\infty$ выполняется вложение $A^f_{\overline{\mathbf{z}}} \subseteq$ $\subseteq A^f_{(z_1,z_2,\ldots,z_M)}$ и, следовательно, для всякого набора $\mathbf{y}$ выполняется $|E^f_{\mathbf{y}} \cap A^f_{\overline{\mathbf{z}}}| \leqslant$ $\leqslant |E^f_{\mathbf{y}} \cap A^f_{(z_1,z_2,\ldots,z_M)}| = 1$.

Докажем необходимость. Предположим противное: существует набор $\mathbf{y} \in T^R_f$, такой, что для сколь угодно большого M найдется $\mathbf{z} \in V_M$, для которого $|E^f_{\mathbf{y}} \cap A^f_{\mathbf{z}}| \geqslant 2$. Рассмотрим предикат Q, определенный на всех наборах $\mathbf{z}$ конечной длины следующим образом: $Q(\mathbf{z}) = \mathrm{T}$ тогда и только тогда, когда $|E^f_{\mathbf{y}} \cap A^f_{\mathbf{z}}| \geqslant 2$.

Нетрудно убедиться, что для всякого $\mathbf{z}$ выполнено: если $Q(\mathbf{z}) = \mathrm{F}$, то $Q(\mathbf{z}|0) =$ $= Q(\mathbf{z}|1) = \mathrm{F}$. Кроме того, учитывая выбор набора $\mathbf{y}$, для всякого M найдется $\mathbf{z}^* \in V_M$, такой, что $Q(\mathbf{z}^*) = \mathrm{T}$. По лемме 1 из этого следует существование последовательности $\overline{\mathbf{z}} = (z_1, z_2, \ldots)$, такой, что для всякого M выполняется $|E^f_{\mathbf{y}} \cap A^f_{(z_1,z_2,\ldots,z_M)}| \geqslant 2$ и, учитывая замечание 3 и утверждение 4, $|E^f_{\mathbf{y}} \cap A^f_{\overline{\mathbf{z}}}| \geqslant 2$. Как следует из леммы 3, последнее неравенство противоречит $\mathbf{y} \in T^R_f$. ■

Пример 1. Рассмотрим функцию $f(x_1, x_2, x_3, x_4) = x_1x_2x_4 \oplus x_2x_4 \oplus x_3$. Можно проверить, что f имеет правый барьер длины 3. Покажем, что $f \in LI_4$. С учетом леммы 4 достаточно показать, что для $\mathbf{y} = (0,0,0) \in V_3$ и для любого $\mathbf{z} = (z_1) \in V_1$ выполняется $|E^f_{\mathbf{y}} \cap A^f_{\mathbf{z}}| = 1$:

$$E^f_{\mathbf{y}} = \{(0,0,0),(0,0,1)\}; \quad E^f_{\mathbf{y}} \cap A^f_{(0)} = \{(0,0,0)\}; \quad E^f_{\mathbf{y}} \cap A^f_{(1)} = \{(0,0,1)\}.$$

Из утверждения 2 и замечания 3 очевидным образом вытекает следующее утверждение.

Лемма 5. Пусть $f \in \mathcal{F}_n$ имеет правый (левый) барьер. Для любого $l \geqslant b^R_f - 1$ (любого $l \geqslant b^L_f - 1$) и любого набора $(y_1, y_2, \ldots, y_l) \in V_l$ верно равенство $A^f_{(y_1,y_2,\ldots,y_l)} =$ $= A^f_{(y_1,y_2,\ldots,y_{b^R_f-1})}$ (соответственно $E^f_{(y_l,y_{l-1},\ldots,y_1)} = E^f_{(y_{b^L_f-1},y_{b^L_f-2},\ldots,y_1)}$).

С учетом леммы 5 легко получить следующую эквивалентную формулировку леммы 4.

Лемма 6. Пусть $f \in \mathcal{F}_n$ имеет правый барьер. Для всякого $m \in \mathbb{N}$ и $\mathbf{y} \in V_m$ верно, что $\mathbf{y} \in T^R_f$ тогда и только тогда, когда для любого набора $\mathbf{z} \in V_{b^R_f-1}$ выполняется $|E^f_{\mathbf{y}} \cap A^f_{\mathbf{z}}| = 1$.

Лемма 7. Пусть $f \in \mathcal{F}_n$ имеет правый и левый барьер. Тогда $f \in \mathcal{LI}_n$, если и только если множество $T^R_f(b^L_f - 1)$ непусто.

Доказательство. Достаточность очевидна. Для доказательства необходимости заметим, что из лемм 5 и 6 вытекает: для любых $l \in \mathbb{N}$, $\mathbf{y} \in V_{b^L_f-1}$, $\mathbf{y}' \in V_l$ верно, что $(\mathbf{y}'|\mathbf{y}) \in T^R_f$ тогда и только тогда, когда $\mathbf{y} \in T^R_f$. Таким образом, если множество $T^R_f(b^L_f - 1)$ пусто, то и T^R_f пусто. ■

Обозначим $r_f = \min\limits_{m\geqslant 1}\min\limits_{\mathbf{y}\in V_m}|E^f_{\mathbf{y}}|$.

Лемма 8. Пусть $f\in\mathcal{F}_n$ имеет левый барьер. Тогда $r_f = 2^{n-1-e^L_f}$.

Доказательство. Из утверждения 2 следует, что при $m\geqslant b^L_f-1$ для любого $\mathbf{y}\in V_m$ верно равенство $|E^f_{\mathbf{y}}| = 2^{n-1-e^L_f}$, и поэтому $r_f = \min\{2^{n-1-e^L_f}, \min\limits_{m'\leqslant b^L_f-2}\min\limits_{\mathbf{y}\in V_{m'}}|E^f_{\mathbf{y}}|\}$. С другой стороны, из замечания 3 следует неравенство $\min\limits_{m'\leqslant b^L_f-2}\min\limits_{\mathbf{y}\in V_{m'}}|E^f_{\mathbf{y}}|\geqslant\min\limits_{\mathbf{y}\in V_{b^L_f-1}}|E^f_{\mathbf{y}}|$, откуда $r_f = 2^{n-1-e^L_f}$. ■

Пусть $f\in\mathcal{F}_n$, $m\in\mathbb{N}$, $\mathbf{y}\in V_m$, $\tilde{\mathbf{x}}\in V_{n-1}$. Введем следующее обозначение:

$$\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}}) = \{(x_{m+1},x_{m+2},\ldots,x_{m+n-1})\in V_{n-1}:$$
$$\exists(x_n,x_{n+1},\ldots,x_m)\ \big(f^{\tilde{\mathbf{x}}}_{R,m,n-1}(x_n,x_{n+1},\ldots,x_{m+n-1}) = \mathbf{y}\big)\}.$$

Лемма 9. Пусть $f\in\mathcal{F}_n$ имеет правый барьер. Тогда для любых $m\geqslant b^R_f-1$, $\mathbf{y}\in V_m$ и $\tilde{\mathbf{x}}\in A^f_{\mathbf{y}}$ выполняется равенство $|\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}})| = 2^{e^R_f}$.

Доказательство. Зафиксируем произвольным образом $m\geqslant b^R_f-1$, $\mathbf{y}\in V_m$ и $\tilde{\mathbf{x}}\in A^f_{\mathbf{y}}$. По теореме 3 верно равенство $|{f^{\tilde{\mathbf{x}}}_{R,l,n-1}}^{-1}(\mathbf{y})| = 2^{e^R_f}$ и поэтому, очевидно, $|\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}})|\leqslant 2^{e^R_f}$. С другой стороны, как следует из теоремы 1, любой набор множества ${f^{\tilde{\mathbf{x}}}_{R,l,n-1}}^{-1}(\mathbf{y})$ определяется последними $n-1$ символами однозначно, поэтому $|\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}})| = 2^{e^R_f}$. ■

Лемма 10. Пусть $f\in\mathcal{F}_n$ имеет правый барьер. Тогда $r_f\geqslant 2^{e^R_f}$.

Доказательство. Пусть для набора $\mathbf{y}\in V_m$ выполнено $|E^f_{\mathbf{y}}| = r_f$. Учитывая замечание 3, можно считать, что $m\geqslant b^R_f-1$.

Зафиксируем произвольный набор $\tilde{\mathbf{x}}\in A^f_{\mathbf{y}}$. Как следует из леммы 9, множество $\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}})$ имеет мощность $2^{e^R_f}$. Ввиду вложения $\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}})\subseteq E^f_{\mathbf{y}}$ приходим к требуемому неравенству. ■

Из лемм 8 и 10 вытекает следующее утверждение.

Следствие 3. Пусть $f\in\mathcal{F}_n$ имеет правый и левый барьер. Тогда $e^R_f+e^L_f\leqslant n-1$.

Теорема 6. Пусть $f\in\mathcal{F}_n$ имеет правый барьер. Тогда если $r_f = 2^{e^R_f}$, то $f\in\mathcal{LI}_n$, причем любой набор $\mathbf{y}$, такой, что $|E^f_{\mathbf{y}}| = 2^{e^R_f}$, принадлежит множеству T^R_f.

Доказательство. Достаточно показать, что если для некоторого числа $m\in\mathbb{N}$ и набора $\mathbf{y}\in V_m$ выполнено равенство $|E^f_{\mathbf{y}}| = 2^{e^R_f}$, то $\mathbf{y}\in T^R_f$.

Пусть сначала $m\geqslant\max\{b^R_f-1,n\}$. Выберем любой набор $\tilde{\mathbf{x}}\in A^f_{\mathbf{y}}$. Нетрудно видеть, что для такого набора, с одной стороны, выполняется вложение $\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}})\subseteq E^f_{\mathbf{y}}$, а с другой стороны, по лемме 9, $|\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}})| = 2^{e^R_f}$. Следовательно, $\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}}) = E^f_{\mathbf{y}}$.

С учетом леммы 6 достаточно показать: для любого набора $\mathbf{z}\in V_{b^R_f-1}$ выполняется $|\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}})\cap A^f_{\mathbf{z}}| = 1$. Действительно, если предположить, что найдутся $\mathbf{z}\in V_{b^R_f-1}$, $\mathbf{x}',\mathbf{x}''\in\tilde{E}^f_{\mathbf{y}}(\tilde{\mathbf{x}})\cap A^f_{\mathbf{z}}$, $\mathbf{x}'\neq\mathbf{x}''$, то для некоторых $\mathbf{t}',\mathbf{t}''\in V_{m-n+1}$, $\mathbf{s}',\mathbf{s}''\in V_{b^R_f-1}$ разрешима следующая система уравнений:

$$f_{m+b^R_f-1}(\tilde{\mathbf{x}}|\mathbf{t}'|\mathbf{x}'|\mathbf{s}') = f_{m+b^R_f-1}(\tilde{\mathbf{x}}|\mathbf{t}''|\mathbf{x}''|\mathbf{s}'').$$

Разрешимость этой системы противоречит определению правого барьера длины b_f^R.

В случае, если при $m < \max\{b_f^R - 1, n\}$ для некоторого $\mathbf{y} = (y_1, y_2, \ldots, y_m) \in V_m$ выполняется $|E_{\mathbf{y}}^f| = 2^{e_f^R}$ (и тогда $r_f = 2^{e_f^R}$), рассмотрим набор $\mathbf{y}^* = (0, 0, \ldots, 0, y_1, y_2, \ldots, y_m) \in V_{\max\{b_f^R-1,n\}}$. Легко видеть, что тогда $|E_{\mathbf{y}^*}^f| = r_f = |E_{\mathbf{y}}^f|$ и $E_{\mathbf{y}^*}^f = E_{\mathbf{y}}^f$. Как следует из показанного выше, для любого $\mathbf{z} \in V_{b_f^R-1}$ выполнено $|\tilde{E}_{\mathbf{y}^*}^f(\tilde{\mathbf{x}}) \cap A_{\mathbf{z}}^f| = 1$ и, значит, $|E_{\mathbf{y}}^f \cap A_{\mathbf{z}}^f| = 1$. По лемме 6 это означает, что $\mathbf{y} \in T_f^R$. ■

Теорема 7. Пусть $f \in \mathcal{F}_n$ имеет правый и левый барьер. Тогда если $e_f^R + e_f^L = n-1$, то $f \in \mathcal{PLI}_n^R \cap \mathcal{PLI}_n^L$.

Доказательство. Применяя сначала лемму 8 и теорему 6, а затем утверждение 2, получим $\bigcup\limits_{m=b_f^R-1}^{\infty} V_m \subseteq T_f^R$, а это и означает, что $f \in \mathcal{PLI}_n^R$ с $p = b_f^R - 1$.

Применяя аналогичные рассуждения к функции $\overleftrightarrow{f}$, получим, что $f \in \mathcal{PLI}_n^L$ с $p = b_f^L - 1$. ■

Следствие 4. Пусть $f \in \mathcal{F}_n$ имеет правый и левый барьер. Тогда если $e_f^R + e_f^L = n - 1$, то f является локально обратимой (т. е. слабо локально обратимой вправо) и слабо локально обратимой влево.

Теорема 8. Пусть $f \in \mathcal{F}_n$ имеет правый барьер. Тогда если $f \in \mathcal{LI}_n$, то $r_f = 2^{e_f^R}$, причем равенство $|E_{\mathbf{y}}^f| = 2^{e_f^R}$ верно для всех наборов $\mathbf{y} \in T_f^R$.

Доказательство. Пусть для некоторого $m \in \mathbb{N}$ и $\mathbf{y} \in V_m$ верно $\mathbf{y} \in T_f^R$. Из леммы 6 следует, что для любого набора $\mathbf{z} \in V_{b_f^R-1}$ выполняется равенство $|E_{\mathbf{y}}^f \cap A_{\mathbf{z}}^f| = 1$. Пусть $l = m + b_f^R - 1$. Рассмотрим произвольные наборы $\mathbf{x}^* \in E_{\mathbf{y}}^f$ и $\mathbf{z} \in \mathrm{Im}(f_{R,b_f^R-1,n-1}^{\mathbf{x}^*})$. В таком случае $E_{\mathbf{y}}^f \cap A_{\mathbf{z}}^f = \{\mathbf{x}^*\}$ и множество ${f_l}^{-1}(\mathbf{y}|\mathbf{z})$ имеет следующую структуру:

$$
\begin{gathered}
{f_l}^{-1}(\mathbf{y}|\mathbf{z}) = \{(x_1, x_2, \ldots, x_{l+n-1}) \in V_{l+n-1} : (x_{m+1}, x_{m+2}, \ldots, x_{m+n-1}) = \mathbf{x}^*, \\
(x_1, x_2, \ldots, x_m) \in {f_{L,m,n-1}^{\mathbf{x}^*}}^{-1}(\mathbf{y}), (x_{m+n}, x_{m+n+1}, \ldots, x_{l+n-1}) \in {f_{R,b_f^R-1,n-1}^{\mathbf{x}^*}}^{-1}(\mathbf{z})\}.
\end{gathered}
$$

Таким образом, $|{f_l}^{-1}(\mathbf{y}|\mathbf{z})| = 2^{e_f^R} |{f_{L,m,n-1}^{\mathbf{x}^*}}^{-1}(\mathbf{y})|$. С другой стороны, $f \in \mathcal{PB}_n$, поэтому $|{f_l}^{-1}(\mathbf{y}|\mathbf{z})| = 2^{n-1}$ и, следовательно,

$$
|{f_{L,m,n-1}^{\mathbf{x}^*}}^{-1}(\mathbf{y})| = 2^{n-1-e_f^R}. \tag{6}
$$

Так как набор $\mathbf{x}^*$ из множества $E_{\mathbf{y}}^f$ был выбран произвольно, равенство (6) выполняется для любого $\mathbf{x}^* \in E_{\mathbf{y}}^f$. Но в таком случае сумма $\sum\limits_{\mathbf{x}^* \in E_{\mathbf{y}}^f} |{f_{L,m,n-1}^{\mathbf{x}^*}}^{-1}(\mathbf{y})|$, равная, с одной стороны, $|f_m^{-1}(\mathbf{y})| = 2^{n-1}$, равна также $2^{n-1-e_f^R} |E_{\mathbf{y}}^f|$, откуда следует $|E_{\mathbf{y}}^f| = 2^{e_f^R}$ и, с учетом леммы 10, $r_f = 2^{e_f^R}$. ■

Из леммы 8 и теоремы 8 вытекает следующее необходимое условие локальной обратимости функции с правым и левым барьером.

Следствие 5. Пусть $f \in \mathcal{F}_n$ имеет правый и левый барьер. Тогда если $f \in \mathcal{LI}_n$, то $e_f^R + e_f^L = n - 1$.

Следствие 6. Пусть $f \in \mathcal{F}_n$ имеет правый и левый барьер. Тогда если $f \in \mathcal{LI}_n$, то $b_f^R + b_f^L \geqslant n + 1$. Если дополнительно f существенно и нелинейно зависит от первой и последней переменной, то $b_f^R + b_f^L \geqslant n + 3$.

Доказательство. Вытекает из следствия 5, а также теорем 3 и 4. ■

С помощью теорем 5, 6 и 8 легко получить описание множества T_f^R для произвольной функции $f \in \mathcal{F}_n$.

Теорема 9. Для произвольной функции f верно

$$T_f^R = \begin{cases} \varnothing, \text{ если } f \text{ не имеет правого барьера,} \\ \left\{ \mathbf{y} \in \bigcup\limits_{m=1}^{\infty} V_m : \ |E_{\mathbf{y}}^f| = 2^{e_f^R} \right\} \text{ иначе.} \end{cases}$$

Таким образом, получен следующий критерий локальной обратимости.

Следствие 7. Функция $f \in \mathcal{F}_n$ локально обратима тогда и только тогда, когда она имеет правый барьер и $r_f = 2^{e_f^R}$.

Лемма 11. Пусть $f \in \mathcal{F}_n$ не имеет левого (правого) барьера. Тогда $f \notin \mathcal{PLI}_n^R$ (соответственно $f \notin \mathcal{PLI}_n^L$).

Доказательство. Пусть $f \in \mathcal{F}_n$ не имеет левого барьера. Достаточно рассмотреть случай, при котором у f есть правый барьер (иначе $f \notin \mathcal{LI}_n$ и, следовательно, $f \notin \mathcal{PLI}_n^R$).

Так как f не имеет левого барьера, то для любого сколь угодно большого $m \in \mathbb{N}$ существует набор $\mathbf{y} \in V_m$, для которого найдется $\mathbf{z} \in V_{b_f^R-1}$, такой, что в множестве ${f_{m+b_f^R-1}}^{-1}(\mathbf{y}|\mathbf{z})$ есть пара наборов вида $(\mathbf{x}'|0|\tilde{\mathbf{x}}), (\mathbf{x}''|1|\tilde{\mathbf{x}}) \in V_{m+b_f^R+n-2}$, где $\tilde{\mathbf{x}} \in V_{b_f^R-1}$ и $\mathbf{x}', \mathbf{x}'' \in V_{m+n-2}$. Отсюда $|E_{\mathbf{y}}^f \cap A_{\mathbf{z}}^f| \geqslant 2$ и, по лемме 6, $\mathbf{y} \notin T_f^R$.

Таким образом, для всякого $m \in \mathbb{N}$ найдется $\mathbf{y} \notin T_f^R$, поэтому $f \notin \mathcal{PLI}_f^R$.

Аналогично доказывается, что если f не имеет правого барьера, то $f \notin \mathcal{PLI}_f^L$. ■

Теорема 10. Пусть $f \in \mathcal{F}_n$. Тогда следующие утверждения эквивалентны:

1) f обладает правым и левым барьерами и $e_f^R + e_f^L = n - 1$;
2) f обладает левым барьером и является слабо локально обратимой вправо;
3) f обладает правым барьером и является слабо локально обратимой влево;
4) f является слабо обратимой вправо и слабо обратимой влево;
5) $f \in \mathcal{PLI}_n^R$, то есть f является сильно локально обратимой вправо;
6) $f \in \mathcal{PLI}_n^L$, то есть f является сильно локально обратимой влево.

Доказательство. Эквивалентность утверждений 1 и 2, 1 и 3, 1 и 4 вытекает из теоремы 5 и следствий 4 и 5. Кроме того, из теоремы 7 вытекает, что из утверждения 1 следуют утверждения 5 и 6. Чтобы доказать, что из утверждения 5 следует утверждение 1, достаточно сначала применить лемму 11, а затем, учитывая, что $\mathcal{PLI}_n^R \subseteq \mathcal{LI}_n$, теорему 5 и следствие 5. Аналогично доказывается, что из утверждения 6 следует утверждение 1. ■

Таким образом, любая локально обратимая функция является либо функцией с правым барьером и без левого барьера, либо такой функцией с правым и левым барьером, что $e_f^R + e_f^L = n - 1$ (и в таком случае она также является сильно локально обратимой вправо и влево).

ЛИТЕРАТУРА

1. *Preparata F. P.* Convolutional Transformations of Binary Sequences: Boolean Functions and Their Resynchronizing Properties // IEEE Trans. Electron. Comput. 1966. V. 15. No. 6. P. 898–909.

2. *Golic J. Dj.* On the Security of Nonlinear Filter Generators // LNCS. 1996. V. 1039. P. 173–188.
3. *Логачев О. А.* О локальной обратимости одного класса булевых отображений // Материалы IX Междунар. семинара «Дискретная математика и ее приложения», посвященного 75-летию со дня рождения акад. О. Б. Лупанова, Москва, 18–23 июня 2007 года. М.: Изд-во механико-математического факультета МГУ, 2007. С. 440–442.
4. *Рысцов И. К.* Возвратные слова для разрешимых автоматов // Кибернетика и системный анализ. 1994. Т. 6. С. 21–26.
5. *Сумароков С. Н.* Запреты двоичных функций и обратимость для одного класса кодирующих устройств // Обозрение прикладной и промышленной математики. 1994. Т. 1. Вып. 1. С. 33–55.
6. *Логачев О. А., Смышляев С. В., Ященко В. В.* Новые методы изучения совершенно уравновешенных булевых функций // Дискретная математика. 2009. Т. 21. Вып. 2. С. 51–74.
7. *Smyshlyaev S. V.* Perfectly Balanced Boolean Functions and Golic Conjecture // J. Cryptology (accepted, available online). DOI 10.1007/s00145-011-9100-7.
8. *Смышляев С. В.* Барьеры совершенно уравновешенных булевых функций // Дискретная математика. 2010. Т. 22. Вып. 2. С. 66–79.
9. *Смышляев С. В.* О криптографических слабостях некоторых классов преобразований двоичных последовательностей // Прикладная дискретная математика. 2010. № 1(7). С. 5–15.
10. *Смышляев С. В.* Булевы функции без предсказывания // Дискретная математика. 2011. Т. 23. Вып. 1. С. 102–118.
11. *Смышляев С. В.* Построение классов совершенно уравновешенных булевых функций без барьера // Прикладная дискретная математика. 2010. № 3(9). С. 41–50.
12. *Hedlund G. A.* Endomorphisms and automorphisms of the shift dynamical system // Math. Sys. Theory. 1969. No. 3. P. 320–375.

2011 Теоретические основы прикладной дискретной математики №4(14)

DOI 10.17223/20710410/14/3

УДК 512.542

ГОМОМОРФНАЯ УСТОЙЧИВОСТЬ ПАР ГРУПП МАЛОГО ПОРЯДКА[1]

И. А. Шилин*,**, В. В. Китюков*

*Московский авиационный институт, г. Москва, Россия
**Московский государственный гуманитарный университет им. М. А. Шолохова, г. Москва, Россия

E-mail: ilyashilin@li.ru, atum89@gmail.ru

Для каждой пары групп порядка не выше 12 с помощью составленной авторами компьютерной программы изучается алгебраическое строение объединения образов гомоморфизмов одной группы в другую.

Ключевые слова: *гомоморфная устойчивость пары групп, конечная группа.*

Введение

Изучению групп гомоморфизмов отдельных групп, классов групп или алгебраических систем, а также алгебраического строения образов гомоморфизмов в последнее время уделяется много внимания. Например, оказалось, что в некоторых случаях из существования изоморфизма между группами $\mathrm{Hom}(G,G)$ и $\mathrm{Hom}(\tilde{G},\tilde{G})$ следует, что группы G и $\tilde{G}$ тоже изоморфны [1]. Важные результаты о гомоморфизмах абелевых групп получены С. Я. Гриншпоном [2] и Д. Валканом [3], а для случая прямого произведения бесконечных циклических групп Дж. О'Нейллом [4]. Свойства разрешимых групп, вытекающие из строения их конечных гомоморфных образов, рассматриваются в работе [5]. Интересные результаты о конечных образах гомоморфизмов мультипликативных групп алгебр с делением содержатся в [6]. В настоящей работе рассматриваются гомоморфизмы между конечными группами G и H и объединение образов гомоморфизмов $G \longrightarrow H$.

Упорядоченную пару (G,H) групп G и H назовем парой класса 1, если группа H абелева. В противном случае пару (G,H) будем называть парой класса 2. Известно, что для пар класса 1 множество $\mathfrak{H}(G,H)$ гомоморфизмов $G \longrightarrow H$ относительно операции

$$\mathfrak{H}(G,H) \times \mathfrak{H}(G,H) \longrightarrow H^G,\ (\varphi,\psi) \longmapsto \varphi \bullet \psi,\ [\varphi \bullet \psi](a) = \varphi(a)\psi(a) \qquad (1)$$

является группой, а для пар класса 2 это происходит не всегда. Для пар класса 1 обозначение группы $\mathfrak{H}(G,H)$ будем заменять стандартным обозначением $\mathrm{Hom}(G,H)$. Нейтральный элемент группы $\mathrm{Hom}(G,H)$ будем обозначать ε ($\mathrm{Ker}\,\varepsilon = G$).

Пару (G,H) назовем гомоморфно устойчивой, если множество $\bigcup\limits_{\varphi\in\mathfrak{H}(G,H)} \mathrm{Im}\,\varphi$ является подгруппой в H. Будем называть гомоморфную устойчивость сильной, если подгруппа $\bigcup\limits_{\varphi\in\mathfrak{H}(G,H)} \mathrm{Im}\,\varphi$ является нормальным делителем.

В [2] определение гомоморфной устойчивости сформулировано для пар (G,H), в которых обе группы абелевы. Там же доказана гомоморфная устойчивость таких пар в случае периодической группы G. Это утверждение распространяется на все пары

[1]Работа поддержана грантом ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (проект № 586Р-30).

класса 1 с периодической группой G. Следовательно, все пары конечных групп класса 1 гомоморфно устойчивы.

В настоящей работе с помощью программ, составленных на языке Турбо Паскаль, для пар групп порядка не выше 12 получены следующие результаты: 1) для пар класса 1 вычислены группы $\mathrm{Hom}(G,H)$; 2) для пар класса 1 вычислены группы $\bigcup\limits_{\varphi\in\mathrm{Hom}(G,H)} \mathrm{Im}\,\varphi$; 3) для пар класса 2 проверено, является ли множество $\mathfrak{H}(G,H)$ группой относительно операции (1), и в случае положительного ответа эта группа вычислена; 4) для каждой пары класса 2 проверено, является ли пара гомоморфно устойчивой, и в случае положительного ответа вычислена группа $\bigcup\limits_{\varphi\in\mathfrak{H}(G,H)} \mathrm{Im}\,\varphi$. Все группы найдены с точностью до изоморфизма.

Отметим, что нетривиальные группы порядка не выше 12 суть следующие группы [7]: циклические группы $\mathbb{Z}_n$, диэдральные группы $\mathbf{D}_n$ порядка $2n$, прямые произведения $\mathbb{Z}_2^2$, $\mathbb{Z}_2^3$, $\mathbb{Z}_3^2$, $\mathbb{Z}_2\mathbb{Z}_4$ и $\mathbb{Z}_2\mathbb{Z}_6$, кватернионная группа $\mathbb{Q}_8$ порядка 8, знакопеременная группа $\mathbf{A}_4$ и дициклическая группа

$$\mathbf{T}_{12} = \mathbb{Z}_3 \rtimes \mathbb{Z}_4 = \langle a,b,c \mid a^3 = b^2 = c^2 = abc\rangle.$$

1. Группы гомоморфизмов пар класса 1

В частных случаях группы $\mathrm{Hom}(G,H)$ легко вычисляются аналитически (например, $\mathrm{Hom}(\mathbb{Z}_m,\mathbb{Z}_n) = \mathbb{Z}_{\gcd(n,m)}$, где $\gcd(m,n)$ — наибольший общий делитель чисел m и n). В общем случае вид групп $\mathrm{Hom}(G,H)$ остается неизвестным.

Для вычисления группы $\mathrm{Hom}(G,H)$ среди $|H|^{|G|}$ отображений $G \longrightarrow H$ надо выбрать те, которые удовлетворяют определению гомоморфизма. Конструктивное решение задачи, позволяющее существенно сократить вычисления, заключается в предварительном отборе тех отображений, для которых выполняются необходимые условия гомоморфизмов $\varphi(e) = e$ и $\varphi(a^{-1}) = (\varphi(a))^{-1}$. Элементы группы обозначаются числами, причем число 1 всегда обозначает нейтральный элемент. Группы задаются в виде двумерных массивов. Например, используя для группы $\mathbf{T}_{12}$ генетический код

$$\langle s,t \mid s^4 = t^3 = e,\ tst = s\rangle$$

и обозначая ее элементы $e, s, s^2, s^3, t, t^2, st, s^2t, s^3t, st^2, s^2t^2, s^3t^2$ числами 1–12 соответственно, эту группу можно описать массивом

```
 1   2   3   4   5   6   7   8   9  10  11  12
 2   3   4   1   7  10   8   9   5  11  12   6
 3   4   1   2   8  11   9   5   7  12   6  10
 4   1   2   3   9  12   5   7   8   6  10  11
 5  10   8  12   6   1   2  11   4   7   3   9
 6   7  11   9   1   5  10   3  12   2   8   4
 7  11   9   6  10   2   3  12   1   8   4   5
 8  12   5  10  11   3   4   6   2   9   1   7
 9   6   7  11  12   4   1  10   3   5   2   8
10   8  12   5   2   7  11   4   6   3   9   1
11   9   6   7   3   8  12   1  10   4   5   2
12   5  10   8   4   9   6   2  11   1   7   3.
```

Отображения $\varphi \in H^G$ реализуются в виде меняющихся состояний переменного одномерного массива размера $|G|$, в ячейки которого записываются элементы группы G,

причем в первую ячейку, означающую образ нейтрального элемента, всегда записывается число 1, а остальные ячейки заполняются с учетом равенства $\varphi(a^{-1}) = (\varphi(a))^{-1}$.

Вычисляя периоды получившихся гомоморфизмов, можно прийти к выводу о группе $\mathrm{Hom}(G, H)$. Так, в результате работы программы для пары $(\mathbf{T}_{12}, \mathbb{Z}_8)$ получаем следующие данные:

```
1 1 1 1 1 1 1 1 1 1 1 1
1 3 5 7 1 1 3 5 7 3 5 7
1 5 1 5 1 1 5 1 5 5 1 5
1 7 5 3 1 1 7 5 3 7 5 3
```

Здесь 1, 3, 5, 7 обозначают соответственно элементы $0, 2, 4, 6$ группы $\mathbb{Z}_8$. Группа $\mathrm{Hom}(\mathbf{T}_{12}, \mathbb{Z}_8)$, таким образом, состоит из четырех элементов; обозначим эти гомоморфизмы (в той последовательности, в которой они перечислены выше) $\varepsilon, \varphi, \psi$ и σ. С точностью до изоморфизма существует две группы четвертого порядка: циклическая группа $\mathbb{Z}_4$, два элемента в которой имеют период 4, и группа $\mathbb{Z}_2^2$, периоды всех элементов которой, за исключением нейтрального, равны 2. Так как $\operatorname{ord} 2 = \operatorname{ord} 6 = 4$ и $\operatorname{ord} 4 = 2$, то $\operatorname{ord}\varphi = \operatorname{ord}\sigma = 4$, $\operatorname{ord}\psi = 2$, то есть $\mathrm{Hom}(\mathbf{T}_{12}, \mathbb{Z}_8) = \mathbb{Z}_4$.

Полученные результаты содержатся в табл. 1.

Таблица 1

Группы $\mathrm{Hom}(G, H)$

GH	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\mathbb{Z}_5$	$\mathbb{Z}_6$	$\mathbb{Z}_7$	$\mathbb{Z}_8$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\mathbb{Z}_9$	$\mathbb{Z}_3^2$	$\mathbb{Z}_{10}$	$\mathbb{Z}_{11}$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2\mathbb{Z}_6$
$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbb{Z}_3$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3^2$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3$
$\mathbb{Z}_4$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$
$\mathbb{Z}_2^2$	$\mathbb{Z}_4$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\mathbb{Z}_2^6$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$
$\mathbb{Z}_5$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_5$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_5$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$
$\mathbb{Z}_6$	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_6$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\mathbb{Z}_3$	$\mathbb{Z}_3^2$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_6$	$\mathbb{Z}_2\mathbb{Z}_6$
@ $\mathbf{D}_3$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbb{Z}_7$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_7$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$
$\mathbb{Z}_8$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_8$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$
$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\mathbb{Z}_2^6$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^4$
$\mathbb{Z}_2^3$	$\mathbb{Z}_2^3$	$\{\varepsilon\}$	$\mathbb{Z}_2^3$	$\mathbb{Z}_2^6$	$\{\varepsilon\}$	$\mathbb{Z}_2^3$	$\{\varepsilon\}$	$\mathbb{Z}_2^3$	$\mathbb{Z}_2^6$	$\mathbb{Z}_2^9$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2^3$	$\{\varepsilon\}$	$\mathbb{Z}_2^3$	$\mathbb{Z}_2^6$
$\mathbf{D}_4$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\mathbb{Z}_2^6$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$
$\mathbb{Q}_8$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\mathbb{Z}_2^6$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$
$\mathbb{Z}_9$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_9$	$\mathbb{Z}_3^2$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3$
$\mathbb{Z}_3^2$	$\{\varepsilon\}$	$\mathbb{Z}_3^2$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3^2$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3^2$	$\mathbb{Z}_3^4$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3$
$\mathbb{Z}_{10}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_5$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_{10}$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbf{D_5}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbb{Z}_{11}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_{11}$	$\{\varepsilon\}$	$\{\varepsilon\}$
$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_6$	$\{\varepsilon\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\mathbb{Z}_3$	$\mathbb{Z}_3$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2\mathbb{Z}_6$
$\mathbb{Z}_2\mathbb{Z}_6$	$\mathbb{Z}_2^2$	$\mathbb{Z}_3$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\{\varepsilon\}$	$\mathbb{Z}_2\mathbb{Z}_6$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\mathbb{Z}_2^6$	$\mathbb{Z}_3$	$\mathbb{Z}_3^2$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2\mathbb{Z}_6$	$\mathbb{Z}_2^4\mathbb{Z}_3$
$\mathbf{A}_4$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3^2$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3$
$\mathbf{D}_6$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$	$\mathbb{Z}_2^6$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^4$
$\mathbf{T}_{12}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_2$	$\{\varepsilon\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$

2. Объединение образов групп гомоморфизмов пар класса 1

Информация о гомоморфизмах $G \longrightarrow H$ позволяет вручную вычислить группы $\bigcup_{\varphi\in\mathrm{Hom}(G,H)} \mathrm{Im}\,\varphi$. Сведения об этих группах содержатся в табл. 2.

Таблица 2

Группы $\bigcup_{\varphi\in\mathrm{Hom}(G,H)} \mathrm{Im}\,\varphi$

GH	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\mathbb{Z}_5$	$\mathbb{Z}_6$	$\mathbb{Z}_7$	$\mathbb{Z}_8$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\mathbb{Z}_9$	$\mathbb{Z}_3^2$	$\mathbb{Z}_{10}$	$\mathbb{Z}_{11}$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2\mathbb{Z}_6$
$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbb{Z}_3$	$\{e\}$	$\mathbb{Z}_3$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3^2$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3$
$\mathbb{Z}_4$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$
$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbb{Z}_5$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_5$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_5$	$\{e\}$	$\{e\}$	$\{e\}$
$\mathbb{Z}_6$	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_6$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\mathbb{Z}_3$	$\mathbb{Z}_3^2$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_6$	$\mathbb{Z}_2\mathbb{Z}_6$
$\mathbf{D}_3$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbb{Z}_7$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_7$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$
$\mathbb{Z}_8$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_8$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$
$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$
$\mathbb{Z}_2^3$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbf{D}_4$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbb{Q}_8$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbb{Z}_9$	$\{e\}$	$\mathbb{Z}_3$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_9$	$\mathbb{Z}_3^2$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3$
$\mathbb{Z}_3^2$	$\{e\}$	$\mathbb{Z}_3$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3^2$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3$
$\mathbb{Z}_{10}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_5$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_{10}$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbf{D_5}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbb{Z}_{11}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_{11}$	$\{e\}$	$\{e\}$
$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_6$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\mathbb{Z}_3$	$\mathbb{Z}_3^2$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2\mathbb{Z}_6$
$\mathbb{Z}_2\mathbb{Z}_6$	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_6$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\mathbb{Z}_3$	$\mathbb{Z}_3^2$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_6$	$\mathbb{Z}_2\mathbb{Z}_6$
$\mathbf{A}_4$	$\{e\}$	$\mathbb{Z}_3$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\{e\}$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3^2$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3$
$\mathbf{D}_6$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$
$\mathbf{T}_{12}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2\mathbb{Z}_4$	$\mathbb{Z}_2^3$	$\{e\}$	$\{e\}$	$\mathbb{Z}_2$	$\{e\}$	$\mathbb{Z}_4$	$\mathbb{Z}_2^2$

3. Множества гомоморфизмов пар класса 2

Алгоритм вычисления множеств $\mathfrak{H}(G,H)$ ничем не отличается от алгоритма вычисления групп $\mathrm{Hom}(G,H)$. Проверка выполнимости аксиом группы для множеств $\mathfrak{H}(G,H)$ выполняется вручную. Получившиеся результаты содержатся в табл. 3, где для каждой пары (G,H) либо указана группа $\mathfrak{H}(G,H)$, либо отмечено, что множество $\mathfrak{H}(G,H)$ не является группой.

Таблица 3

Множества $\mathfrak{H}(G,H)$ как группы

GH	$\mathbf{D}_3$	$\mathbf{D}_4$	$\mathbb{Q}_8$	$\mathbf{D}_5$	$\mathbf{D}_6$	$\mathbf{A}_4$	$\mathbf{T}_{12}$
$\mathbb{Z}_2$	нет	нет	$\mathbb{Z}_2$	нет	нет	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$
$\mathbb{Z}_3$	$\mathbb{Z}_3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	$\mathbb{Z}_3$	$\mathbb{Z}_3$
$\mathbb{Z}_2^2$	нет	нет	$\mathbb{Z}_2^2$	нет	нет	$\mathbb{Z}_2^4$	$\mathbb{Z}_2^2$
$\mathbb{Z}_6$	$\mathbf{D}_3$	нет	$\mathbb{Z}_2$	нет	$\mathbf{D}_3$	нет	$\mathbb{Z}_6$
$\mathbb{Z}_{10}$	нет	нет	$\mathbb{Z}_2$	нет	нет	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$
$\mathbf{D}_5$	нет	нет	$\mathbb{Z}_2$	нет	нет	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$
$\mathbb{Z}_{12}$	нет	нет	$\mathbb{Q}_8$	нет	нет	нет	нет
$\mathbf{A}_4$	$\mathbb{Z}_3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\mathbb{Z}_3$	нет	$\mathbb{Z}_3$
$\mathbf{T}_{12}$	нет	$\mathbf{D}_4$	$\mathbb{Q}_8$	нет	нет	$\mathbb{Z}_2^2$	нет

4. Гомоморфная устойчивость пар класса 2

Полученные сведения о множествах $\mathfrak{H}(G,H)$ дают информацию о множествах $\bigcup\limits_{\varphi\in\mathfrak{H}(G,H)} \operatorname{Im}\varphi$. Проверка гомоморфной устойчивости выполняется вручную. В итоге получаем результаты, представленные в табл. 4, где для каждой пары (G,H) либо указана группа $\bigcup\limits_{\varphi\in\mathfrak{H}(G,H)} \operatorname{Im}\varphi$, либо отмечено, что это множество не является группой.

Таблица 4

Множества $\bigcup\limits_{\varphi\in\mathfrak{H}(G,H)} \operatorname{Im}\varphi$ как группы

GH	$\mathbf{D}_3$	$\mathbf{D}_4$	$\mathbb{Q}_8$	$\mathbf{D}_5$	$\mathbf{D}_6$	$\mathbf{A}_4$	$\mathbf{T}_{12}$
$\mathbb{Z}_2$	нет	нет	$\mathbb{Z}_2$	нет	нет	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$
$\mathbb{Z}_3$	$\mathbb{Z}_3$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	нет	$\mathbb{Z}_3$
$\mathbb{Z}_2^2$	нет	нет	$\mathbb{Z}_2^2$	нет	нет	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$
$\mathbb{Z}_6$	$\mathbf{D}_3$	нет	$\mathbb{Z}_2$	нет	$\mathbf{D}_6$	$\mathbf{A}_4$	$\mathbb{Z}_6$
$\mathbb{Z}_{10}$	нет	нет	$\mathbb{Z}_2$	$\mathbf{D}_5$	нет	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$
$\mathbf{D}_5$	нет	нет	$\mathbb{Z}_2$	$\mathbf{D}_5$	нет	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$
$\mathbb{Z}_{12}$	$\mathbf{D}_3$	$\mathbf{D}_4$	$\mathbb{Q}_8$	нет	$\mathbf{D}_6$	$\mathbf{A}_4$	$\mathbf{T}_{12}$
$\mathbf{A}_4$	$\mathbb{Z}_3$	$\{e\}$	$\{e\}$	$\{e\}$	$\mathbb{Z}_3$	$\mathbf{A}_4$	$\mathbb{Z}_3$
$\mathbf{T}_{12}$	$\mathbf{D}_3$	$\mathbf{D}_4$	$\mathbb{Q}_8$	нет	нет	$\mathbb{Z}_2^2$	$\mathbf{T}_{12}$

Проверим сильную устойчивость пар из этой таблицы.

Обозначим $|G:\hat{G}|$ индекс подгруппы $\hat{G}$ группы G. Так как $|\mathbf{D}_n:\mathbb{Z}_n|=2$, $|\mathbb{Q}_8:\mathbb{Z}_2^2|=2$ и $|\mathbf{T}_{12}:\mathbb{Z}_6|=2$, то пары $(\mathbb{Z}_3,\mathbf{D}_3)$, $(\mathbf{A}_4,\mathbf{D}_3)$, $(\mathbb{Z}_2^2,\mathbb{Q}_8)$, $(\mathbb{Z}_3,\mathbf{D}_6)$, $(\mathbf{A}_4,\mathbf{D}_6)$ и $(\mathbb{Z}_6,\mathbf{T}_{12})$ сильно гомоморфно устойчивы.

Подгруппа $\mathbb{Z}_2\simeq\mathbf{U}_2\equiv\{1,-1\}$ в группе $\mathbb{Q}_8$ является центром и, следовательно, гомоморфная устойчивость пары $(G,\mathbb{Q}_8)$ при $G\in\{\mathbb{Z}_2,\mathbb{Z}_6,\mathbb{Z}_{10},\mathbf{D}_5\}$ сильная.

Так как группа $\mathbf{A}_4$ является нормализатором подгруппы Клейна $\mathbb{Z}_2^2$ [8, с. 136], то последняя есть нормальный делитель в $\mathbf{A}_4$ [8, с. 131], поэтому устойчивость пар $(G,\mathbf{A}_4)$ в случае $G\in\{\mathbb{Z}_2,\mathbb{Z}_2^2,\mathbb{Z}_{10},\mathbf{D}_5,\mathbf{T}_{12}\}$ сильная.

В некоторых случаях сильную гомоморфную устойчивость целесообразно проверить с помощью компьютерных вычислений. Для этого моделируются внутренние автоморфизмы группы H и проверяется, является ли ее подгруппа $\hat{H}$ неподвижной точкой любого внутреннего автоморфизма. В работе [9] показано, что подгруппы $\mathbb{Z}_2\simeq\{e,s^2\}$ и $\mathbb{Z}_3\simeq\{e,t,t^2\}$ в группе $\mathbf{T}_{12}$ являются нормальными делителями, поэтому в случае $G\in\{\mathbb{Z}_2,\mathbb{Z}_3,\mathbb{Z}_2^2,\mathbb{Z}_{10},\mathbf{D}_5,\mathbf{A}_4\}$ пара $(G,\mathbf{T}_{12})$ сильно гомоморфно устойчива.

С учетом того, что несобственные подгруппы являются нормальными делителями, приходим к выводу, что для всех пар из табл. 4 гомоморфная устойчивость сильная.

ЛИТЕРАТУРА

1. *Себельдин А. М., Сморкалова А. Е.* Определяемость абелевых групп группами гомоморфизмов и полугруппами автоморфизмов // Матем. заметки. 2008. Т. 84. № 4. С. 595–601.
2. *Гриншпон С. Я.* Гомоморфная устойчивость абелевых групп // Фундам. и прикл. мат. 2008. Т. 14. № 5. С. 67–76.
3. *Valkan D.* On some homomorphisms of direct sums of modules // Proc. Razmodze Math. Inst. 2000. V. 121. P. 151–162.
4. *O'Neill J. D.* On homomorphisms between direct products of infinite cyclic groups // Commun. Algebra. 2000. V. 28. No. 11. P. 5047–5052.

5. *Тушев А. В.* О разрешимых группах, чьи конечные гомоморфные образы имеют ограниченный ранг // Матем. заметки. 1994. Т. 56. № 5. С. 136–139.
6. *Segev Y.* On finite homomorphic images of the multiplicative group of a division algebra // Ann. Math. 1999. V. 149. No. 1. P. 219–251.
7. *Conway J. H., Curtis R. T., Norton S. P., et al.* Atlas of Finite Groups. Oxford: University Press, 1985. 252 p.
8. *Шилин И. А.* Введение в алгебру. Часть первая. М.: МГСГИ, 2010. 160 с.
9. *Александров А. А., Нижников А. И., Шилин И. А.* Компьютерное вычисление подгрупп и нормальных делителей неабелевых групп порядка не выше 20 // Преподаватель XXI века. 2011. № 1. Ч. 2. С. 214–220.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

DOI 10.17223/20710410/14/4

УДК 519.7

ПОЧТИ СОВЕРШЕННЫЕ ШИФРЫ И КОДЫ АУТЕНТИФИКАЦИИ

А. Ю. Зубов

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

E-mail: Zubovanatoly@yandex.ru

Предлагаются конструкции почти совершенных шифров с экономным расходом ключа, совмещающие функции шифрования и аутентификации при равновероятном выборе ключей и открытых текстов.

Ключевые слова: *почти совершенный шифр, код аутентификации.*

Пусть $\Sigma = (S, K, M, E, D)$ — шифр[1], для которого S, M — множества открытых и шифрованных текстов; K — множество ключей; E — множество правил зашифрования, состоящие из инъективных отображений $E_k : S \to M$ для каждого $k \in K$; D — множество правил расшифрования, состоящее из отображений $D_k : M \to M \cup \{\varnothing\}$, таких, что

$$D_k(m) = \begin{cases} s, & \text{если } E_k(s) = m, \\ \varnothing, & \text{если } m \notin E_k(S). \end{cases}$$

Пусть на множествах S, K определены распределения вероятностей $\mathsf{P}(S)$, $\mathsf{P}(K)$, состоящие соответственно из вероятностей $p_S(s)$ и $p_K(k)$, строго больших нуля для каждого $s \in S$ и $k \in K$. Будем полагать, что при шифровании ключи и открытые тексты выбираются независимо друг от друга. Тогда $\mathsf{P}(S)$, $\mathsf{P}(K)$ естественным образом индуцируют распределение $\mathsf{P}(M)$ на множестве M, состоящее из вероятностей

$$p_M(m) = \sum_{k \in K(m)} p_K(k) \cdot p_S(D_k(m)), \quad m \in M, \tag{1}$$

где $K(m) = \{k \in K : D_k(m) \neq \varnothing\}$.

Так же естественно вводятся условные распределения $\mathsf{P}(S \,|\, M)$ и $\mathsf{P}(M \,|\, S)$. Для каждой пары (s, m) вероятности $p_{S\,|\,M}(s \,|\, m)$ и $p_{M\,|\,S}(m \,|\, s)$ определяются соответственно формулами

$$p_{M\,|\,S}(m \,|\, s) = \sum_{k \in K(s,m)} p_K(k); \tag{2}$$

$$p_{S\,|\,M}(s \,|\, m) = \frac{p_S(s) \cdot p_{M\,|\,S}(m \,|\, s)}{p_M(m)}, \tag{3}$$

где $K(s, m) = \{k \in K : E_k(s) = m\}$.

К. Шеннон назвал шифр Σ совершенным, если для любых $s \in S$, $m \in M$ имеет место равенство

$$p_{S\,|\,M}(s \,|\, m) = p_S(s). \tag{4}$$

[1]Пользуемся определением шифра, приведённым в [1, 2].

Интерес к таким шифрам связан с тем, что шифртекст не даёт вероятностной информации об открытом тексте. В этом смысле не менее интересны шифры Σ, для которых величина

$$\Delta(\Sigma) = \max_{(s,m)} \left| p_{S\,|\,M}(s\,|\,m) - p_S(s) \right|$$

не превосходит достаточно малого действительного числа $\varepsilon \geqslant 0$. Назовём такие шифры почти совершенными, или ε-совершенными. Совершенный шифр является ε-совершенным при $\varepsilon = 0$. Ясно, что такие шифры относятся (как и совершенные шифры) к классу теоретически стойких шифров, поскольку, независимо от используемых потенциальным противником вычислительных ресурсов, они не позволяют определить по шифртексту открытый текст с вероятностью, превосходящей ε. Как известно, недостатком совершенного шифра является большой расход ключа, поскольку для совершенного шифра должны выполняться неравенства $|S| \leqslant |M| \leqslant |K|$ (см. [1]). В связи с этим возникает вопрос о возможности сокращения расхода ключа при переходе от жёсткого условия (4) к менее жёсткому условию $\Delta(\Sigma) \leqslant \varepsilon$.

В работе предлагаются некоторые классы ε-совершенных шифров с равномерными распределениями $\mathsf{P}(S)$, $\mathsf{P}(K)$, допускающие эффективную реализацию, для которых $|K| \ll |S|$. Помимо стойкого шифрования, рассматриваемые шифры гарантируют также стойкую аутентификацию.

Заметим, что для равномерных распределений $\mathsf{P}(S)$, $\mathsf{P}(K)$ вероятность $p_{S\,|\,M}(s\,|\,m)$ представляется в виде

$$p_{S\,|\,M}(s\,|\,m) = \frac{|K(s,m)|}{|K(m)|}.$$

Это следует из формул (1)–(3). Таким образом, в рассматриваемых условиях величина

$$\Delta(s,m) = \left| p_{S\,|\,M}(s\,|\,m) - p_S(s) \right|$$

выражается формулой

$$\Delta(s,m) = \left| \frac{|K(s,m)|}{|K(m)|} - \frac{1}{|S|} \right|. \tag{5}$$

Приведём примеры почти совершенных шифров, построенных на основе кодов из [3].

Пример 1. Пусть F_q — поле характеристики 2, состоящее из q элементов, и r — произвольное натуральное число. Рассмотрим шифр Σ_1, для которого $S = (F_q)^r$, $K = (F_q)^2$ и $M = (F_q)^{r+1}$. Правило зашифрования строки $s = (s_1, \ldots, s_r)$ на ключе $k = (a, b)$ определим формулой

$$E_k(s) = \left(u_1, \ldots, u_r, a + u_1 \cdot b^1 + \ldots + u_r \cdot b^r\right),$$

где

$$u_i = s_i + c_i \cdot a + d_i \cdot b, \;\; i = 1, \ldots, r,$$

а $c_1, \ldots, c_r, d_1, \ldots, d_r$ — произвольные ненулевые константы из F_q, такие, что

$$c_i \cdot d_j \neq c_j \cdot d_i \tag{6}$$

при $i \neq j$.

Утверждение 1. Шифр Σ_1 с равномерными распределениями $\mathsf{P}(S)$, $\mathsf{P}(K)$ является ε-совершенным шифром для $\varepsilon < q^{-1}$.

Доказательство. Заметим, что для любого $m = (u_1, \ldots, u_r, w) \in M$ выполняется равенство $|K(m)| = q$. В самом деле, пусть $E_{(a,b)}(s_1, \ldots, s_r) = (u_1, \ldots, u_r, w)$. Тогда $s_i = u_i + c_i \cdot a + d_i \cdot b$, $i = 1, \ldots, r$, и $w = a + u_1 \cdot b + \ldots + u_r \cdot b^r$. Нужное утверждение следует из того, что для любого $b \in F_q$ однозначно определяется параметр a из последнего равенства.

Покажем теперь, что для любых $s \in S$ и $m \in M$ справедливо неравенство $|K(s, m)| \leqslant 1$. В самом деле, пусть для двух различных пар (a, b) и (a', b') выполняются равенства $u_i = s_i + c_i \cdot a + d_i \cdot b = s_i + c_i \cdot a' + d_i \cdot b'$, $i = 1, \ldots, r$. Из этих равенств получаем, в частности, следствия для любых различных $i, j = 1, \ldots, r$:

$$c_i \cdot (a + a') = d_i \cdot (b + b'),$$
$$c_j \cdot (a + a') = d_j \cdot (b + b').$$

Константы c_i, d_i ненулевые, поэтому $a + a' \neq 0$, так как в противном случае $b + b' = 0$, и тогда $(a, a') = (b, b')$, что противоречит условию. В таком случае из последних соотношений получаем равенства $c_i \cdot {d_i}^{-1} = (b + b')(a + a')^{-1} = c_j \cdot {d_j}^{-1}$, или $c_i \cdot d_j = c_j \cdot d_i$, что противоречит условию (6). Полученное противоречие доказывает требуемое свойство.

Теперь, используя (5), получаем неравенства $\Delta(s, m) \leqslant q^{-1} - q^{-r} < q^{-1}$, откуда $\Delta(\Sigma_1) = \max\limits_{(s,m)} \Delta(s, m) < q^{-1}$, что и требуется. ■

Рассмотрим вопрос об использовании шифра Σ_1 для обеспечения аутентификации данных. Формально речь идёт об оценке имитостойкости шифра Σ_1, то есть его стойкости к атакам типа имитации и подмены в случае, когда каждый ключ используется для передачи не более одного сообщения. Этот вопрос подробно рассмотрен в [2], где мерой стойкости служит вероятность успеха атаки. Тот же вопрос можно рассматривать и с других позиций. Шифр Σ_1 представляет собой код аутентификации с секретностью [4]. В случае, когда распределения $\mathsf{P}(S)$, $\mathsf{P}(K)$ равномерны, стойкость кода аутентификации к активным атакам определяется вероятностями p_0 и p_1, где (см., например, [5, 6])

$$p_0 = \max_{m \in M} \frac{|K(m)|}{|K|}; \tag{7}$$

$$p_1 = \max_{m \in M} \max_{n \in M \setminus \{m\}} \frac{|K(m, n)|}{|K(m)|}, \tag{8}$$

$$|K(m, n)| = |K(m)| \cap |K(n)|.$$

Как код аутентификации Σ_1 работает следующим образом. Получатель сообщения $m = (u_1, \ldots, u_r, w)$, владеющий секретным ключом $k = (a, b)$, восстанавливает передаваемое состояние источника $s = (s_1, \ldots, s_r)$, пользуясь соотношениями

$$s_i = u_i + c_i \cdot a + d_i \cdot b, \;\; i = 1, \ldots, r.$$

Критерием аутентичности сообщения служит равенство $w = a + u_1 \cdot b + \ldots + u_r \cdot b^r$.

Утверждение 2. Для кода аутентификации Σ_1 с равномерными распределениями $\mathsf{P}(S)$, $\mathsf{P}(K)$ выполняются соотношения

$$p_0 = q^{-1}, \;\; p_1 < r \cdot q^{-1}. \tag{9}$$

Доказательство. Как отмечалось в доказательстве утверждения 1, для любого $m \in M$ имеет место равенство $|K(m)| = q$. Пусть $(a, b) \in K(m, n)$, где $m = (u_1, \ldots, u_{r+1})$, $n = (v_1, \ldots, v_{r+1})$. Тогда a и b находятся из системы уравнений

$$\begin{cases} a + u_1 \cdot b + \ldots + u_r \cdot b^r = u_{r+1}, \\ a + v_1 \cdot b + \ldots + v_r \cdot b^r = v_{r+1}. \end{cases} \tag{10}$$

Поскольку уравнение $(u_1+v_1)\cdot b+\ldots+(u_r+v_r)\cdot b^r = u_{r+1}+v_{r+1}$ имеет в поле F_q не более r решений, системе (10) может удовлетворять не более r значений b. Для каждого из них из первого уравнения системы однозначно находится значение a. Это доказывает неравенство $|K(m)| \leqslant r$ для любых $m, n \in M$. Соотношения (9) следуют теперь из формул (7), (8). ■

Проиллюстрируем уровень стойкости шифра Σ_1 числовым примером. Пусть $q = 2^{64}$, $r = 2^{16}$. Тогда Σ_1 при использовании 128-битного ключа обеспечивает (2^{-64})-совершенное шифрование сообщений длины, не превосходящей $2^{16} \cdot 64 = 4194304$ бит, и их аутентификацию с уровнем стойкости, определяемым параметрами $p_0 = 2^{-64}$, $p_1 < 2^{-48}$.

Пример 2. Пусть $q = 2^r$, $Q = 2^{r+t}$, r, t — натуральные числа, такие, что $2t \geqslant r$. Рассмотрим шифр Σ_2, для которого $S = (F_Q)^{2^t+1}$, $K = F_Q \times F_Q \times F_q$, $M = (F_Q)^{2^t+2}$. Правило зашифрования строки $s = (s_{2^t}, \ldots s_1, s_0)$ на ключе $k = (a, b, c)$ определим формулой

$$E_k(s) = \left(u_{2^t}, \ldots, u_0, c + \left[b \cdot \left(u_{2^t} \cdot a^{2^t} + \ldots + u_1 \cdot a + u_0 \right) \right]_q \right), \tag{11}$$

где $u_i = s_i + h_i \cdot a + g_i \cdot b$, $i = 0, 1, \ldots, 2^t$, а $h_0, \ldots, h_{2^t}, g_0, \ldots, g_{2^t}$ — произвольные константы из F_Q, такие, что $h_i \cdot g_j \neq h_j \cdot g_i$ при $i \neq j$. Запись $[\alpha]_q$ означает приведение элемента α по модулю q, то есть вычёркивание t старших координат двоичного представления α.

Утверждение 3. Шифр Σ_2 с равномерными распределениями $\mathsf{P}(S)$, $\mathsf{P}(K)$ является ε-совершенным шифром для

$$\varepsilon = \left| \frac{1}{2^{r+2t}} - \frac{1}{2^{(r+1)\cdot(2^t+1)}} \right|.$$

Доказательство. Для любого $m = (u_{2^t}, \ldots, u_0, w) \in M$ выполняется равенство

$$|K(m)| = 2^{r+2t}. \tag{12}$$

В самом деле, пусть $E_{(a,b,c)}(s_{2^t}, \ldots, s_0) = (u_{2^t}, \ldots, u_0, w)$. Тогда $s_i = u_i + h_i \cdot a + g_i \cdot b$, $i = 0, 1, \ldots, 2^t$, и $w = c + \left[b \cdot \left(u_{2^t} \cdot a^{2^t} + \ldots + u_1 \cdot a + u_0 \right) \right]_q$. Для любых $a, b \in F_Q$ из последнего равенства однозначно определяется параметр $c \in F_q$. Равенство (12) следует теперь из того, что для любого $\gamma \in F_q$ число пар $(\alpha, \beta) \in F_Q \times F_Q$, таких, что $[\alpha, \beta]_q = \gamma$, равно Q^2/q.

Точно так же, как и в утверждении 1, доказывается, что для любых $s \in S$ и $m \in M$

$$|K(s, m)| \leqslant 1. \tag{13}$$

Теперь из (5), (12) и (13) получаем равенства

$$\Delta\left(\Sigma_2\right) = \max_{(s,m)} \Delta(s,m) = \max\left\{\frac{1}{2^{(r+1)\cdot(2^t+1)}}, \left|\frac{1}{2^{r+2t}} - \frac{1}{2^{(r+1)\cdot(2^t+1)}}\right|\right\} = \left|\frac{1}{2^{r+2t}} - \frac{1}{2^{(r+1)\cdot(2^t+1)}}\right|,$$

откуда следует (11). ■

Как и Σ_1, шифр Σ_2 можно рассматривать как код аутентификации с секретностью.

Утверждение 4. Для кода аутентификации Σ_2 с равномерными распределениями $\mathsf{P}(S)$, $\mathsf{P}(K)$ выполняются равенства

$$p_0 = 2^{-r}, \;\; p_1 = 2^{-(r-1)} - 2^{-2r}. \tag{14}$$

Доказательство. Вероятности p_0, p_1 можно вычислить непосредственно, как и в утверждении 2. Вместо этого воспользуемся известным утверждением. В [3] изучается код аутентификации Σ с теми же, что и у Σ_2, множествами S, K, M. Отличие состоит лишь в том, что правило кодирования кода Σ задаётся формулой

$$\overline{E}_{(a,b,c)}(s) = \left(s, c + \left[s(a)\cdot b\right]_q\right), \tag{15}$$

где $s = (s_{2^t}, \ldots, s_0)$ определяет коэффициенты многочлена $s(x) = s_{2^t}\cdot x^{2^t} + \ldots + s_0$. В [3] доказано, что для кода Σ вероятности успеха имитации и подмены определяются формулами (14). Покажем, что вероятности p_0, p_1 одинаковы для Σ и Σ_2. Для этого достаточно заметить, что одинаковы матрицы инцидентности кодов.

Напомним, что матрица инцидентности кода аутентификации — это матрица $(i(k,m))$ размера $|K|\times|M|$, состоящая из элементов

$$i(k,m) = \begin{cases} 1, & \text{если } E_k(s) = m \text{ для некоторого } s\in S,\\ 0 & \text{в противном случае.}\end{cases}$$

Заметим, что правила кодирования (11) и (15) связаны соотношением

$$E_k(s) = \overline{E}_k\left(\varphi_k(s)\right), \tag{16}$$

где $\varphi_k : S\to S$ — биекция для каждого $k\in K$. Очевидно, что матрицы инцидентности кодов, связанных соотношением (16), одинаковы. Это позволяет сделать вывод о том, что для любых сообщений $m, n \in M$ величины $|K(m)|$ и $|K(m,n)|$, определяющие вероятности p_0, p_1 по формулам (7), (8), одинаковы для Σ и Σ_2 при условии, что распределения $\mathsf{P}(S)$, $\mathsf{P}(K)$ равномерны. ■

Проиллюстрируем уровень стойкости шифра Σ_2 примером. Пусть $t = 32, r = 64$. Тогда Σ_2 при использовании 256-битного ключа обеспечивает (2^{-128})-совершенное шифрование сообщений длины, не превосходящей $(64+32)\cdot(2^{32}+1) = 412316860512$ бит, и их аутентификацию с уровнем стойкости, определяемым параметрами $p_0 = 2^{-64}$ и $p_1 < 2^{-63}$.

Шифры Σ_1 и Σ_2 допускают эффективную реализацию, поскольку операции сложения и умножения в полях характеристики 2 реализуются несложно. Выбор констант c_i, d_i (так же как и констант g_i, h_i), удовлетворяющих соотношениям (6), не представляет труда. Например, в качестве c_i можно выбрать элемент, двоичная запись которого представляет число 2^i, а в качестве d_i — элемент, двоичная запись которого представляет число $2^i + 1$.

Интересен вопрос об оценке стойкости рассмотренных шифров в случае, когда распределение $\mathsf{P}(S)$ неравномерно.

ЛИТЕРАТУРА

1. *Алфёров А. П., Зубов А. Ю., Кузьмин А. С., Черёмушкин А. В.* Основы криптографии. М.: Гелиос АРВ, 2005. 480 с.
2. *Зубов А. Ю.* Криптографические методы защиты информации. Совершенные шифры. М.: Гелиос АРВ, 2005. 192 с.
3. *Kabatianskii G. A., Johansson T., and Smeets B.* On the cardinality of systematic A-codes via error correcting codes // IEEE Trans. Inform. Theory. 1996. V. IT-42. No. 2. P. 566–578.
4. *Зубов А. Ю.* Математика кодов аутентификации. М.: Гелиос АРВ, 2007. 480 с.
5. *Зубов А. Ю.* Оценка стойкости кода аутентификации с двумя состояниями источника при случайном и равновероятном выборе ключей // Безопасность информационных технологий. 2008. № 2. С. 92–96.
6. *Зубов А. Ю.* О выборе оптимальной стратегии защиты для кода аутентификации с двумя состояниями источника // Дискретная математика. 2009. Т. 21. № 4. С. 136–147.

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

DOI 10.17223/20710410/14/5

УДК 519.17

О МИНИМАЛЬНЫХ ВЕРШИННЫХ 1-РАСШИРЕНИЯХ СОЕДИНЕНИЙ ГРАФОВ СПЕЦИАЛЬНОГО ВИДА

М. Б. Абросимов

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

E-mail: mic@rambler.ru

В 2001 г. была высказана гипотеза, что минимальное вершинное 1-расширение графа вида $G + G^*$, где G — произвольный граф, а G^* — некоторое его минимальное вершинное 1-расширение, единственно с точностью до изоморфизма и имеет вид $G^* + G^*$. В работе строится два контрпримера к этой гипотезе, которые показывают, что в общем случае она неверна. Доказывается также, что для многих графов утверждение гипотезы справедливо.

Ключевые слова: *граф, минимальное вершинное 1-расширение, точное вершинное 1-расширение, предполный граф, оптимальная отказоустойчивая реализация.*

Введение

Графом (неориентированным) называется пара $G = (V, \alpha)$, где α — симметричное и антирефлексивное отношение на множестве вершин V. Отношение α называется *отношением смежности*. Степенью вершины v в графе G будем называть количество вершин в G, смежных с данной, и обозначать через $\mathrm{d}(v)$. Вектор, составленный из степеней вершин графа G в порядке убывания, будем называть *вектором степеней*. Здесь и далее определения даются по [1].

Подграфом графа $G = (V, \alpha)$ называется пара $G' = (V', \alpha')$, где $V' \subseteq V$ и $\alpha' = (V' \times V') \cap \alpha$. Подграф графа G называется *максимальным*, если он получается из G удалением одной вершины и всех связанных с нею ребер.

Вложением графа $G_1 = (V_1, \alpha_1)$ в граф $G_2 = (V_2, \alpha_2)$ называется такое однозначное отображение $\varphi : V_1 \to V_2$, что для любых вершин $u, v \in V_1$ выполняется следующее условие: $(u, v) \in \alpha_1 \Rightarrow (\varphi(u), \varphi(v)) \in \alpha_2$.

Назовем граф $G^* = (V^*, \alpha^*)$ *вершинным k-расширением* графа $G = (V, \alpha)$, если граф G вкладывается в каждый подграф графа G^*, получающийся удалением любых его k вершин и всех связанных с ними ребер.

Соединением двух графов $G_1 = (V_1,\ \alpha_1)$ и $G_2 = (V_2,\ \alpha_2)$, не имеющих общих вершин, называется граф $G_1 + G_2 = (V_1 \cup V_2,\ \alpha_1 \cup \alpha_2 \cup V_1 \times V_2 \cup V_2 \times V_1)$.

Из определения видно, что соединение графов обладает свойствами коммутативности и ассоциативности.

Граф $G_t = (V_t, \alpha_t)$ называется *тривиальным k-расширением* графа $G = (V, \alpha)$, если граф G_t получается из графа G добавлением k вершин, соединением их со всеми вершинами графа G и друг с другом, то есть граф G_t есть соединение графа G и

полного графа $K_k = (V_k, V_k \times V_k \setminus \Delta)$:

$$G_t = (V_t, \alpha_t) = (V \cup V_k, \alpha \cup (V_k \times V_k \setminus \Delta) \cup V \times V_k \cup V_k \times V).$$

Очевидно, что тривиальное k-расширение графа является и его вершинным k-расширением, причем $|V_t| = |V| + k$. В самом деле, любую вершину тривиального k-расширения можно заменить на одну из добавленных вершин.

Граф $G^* = (V^*, \alpha^*)$ называется *минимальным вершинным k-расширением* (k натуральное) n-вершинного графа $G = (V, \alpha)$, если выполняются следующие условия:

1) G^* является вершинным k-расширением G, то есть граф G вкладывается в каждый подграф графа G^*, получающийся удалением любых его k вершин;

2) G^* содержит $n + k$ вершин, то есть $|V^*| = |V| + k$;

3) α^* имеет минимальную мощность при выполнении условий 1 и 2.

Понятие минимального вершинного k-расширения введено на основе понятия оптимальной k-отказоустойчивой реализации, которое предложено Дж. П. Хейзом в работе [2] при построении модели отказоустойчивости, основанной на графах. С вычислительной точки зрения задача является сложной: в работе [3] доказывается, что соответствующая задача распознавания является NP-полной. В общем случае граф может иметь много минимальных вершинных k-расширений. Введем частичную операцию получения минимального вершинного 1-расширения графа, считая ее неопределенной для графов, которые имеют более одного минимального вершинного 1-расширения.

Пусть G — некоторый граф, а G^* — его минимальное вершинное 1-расширение. Обозначим через $(G)^*$ результат операции получения минимального вершинного 1-расширения графа G. Тогда если граф G имеет единственное минимальное вершинное 1-расширение, то $(G)^* = G^*$. В противном случае результат операции получения минимального вершинного 1-расширения считается неопределенным для графа G.

Вершина называется *полной*, если она смежна со всеми остальными вершинами графа. Граф называется *предполным*, если у него есть хотя бы одна полная вершина. Граф, все вершины которого полные, называется *полным* и обозначается K_n. Граф с пустым отношением смежности называется *вполне несвязным* и обозначается O_n. Предполный граф можно записать в виде $K_1 + G$ или $O_1 + G$. Класс предполных графов является достаточно большим: n-вершинных предполных графов столько же, сколько всего неизоморфных $(n-1)$-вершинных графов. Интересны эти графы тем, что в работе [4] удалось найти полное аналитическое решение задачи описания всех минимальных вершинных k-расширений предполных графов. Приведем один из результатов этой работы для случая $k = 1$.

Теорема 1. Относительно предполных графов справедливы следующие утверждения.

1. При четном n любой n-вершинный предполный граф $K_1 + G$ имеет единственное с точностью до изоморфизма минимальное вершинное 1-расширение — тривиальное, то есть $(K_1 + G)^* = K_1 + (K_1 + G) = K_2 + G$.

2. При нечетном n:

а) При $n \leqslant 3$ существует один и только один n-вершинный предполный граф G_{np1}, для которого тривиальное расширение не является минимальным. Граф G_{np1} может быть представлен в виде $O_1 + O_2 + ... + O_2$. При этом граф G_{np1} имеет единственное с точностью до изоморфизма минимальное вершинное 1-расширение, и оно содержит $n - 1$ дополнительных ребер: $(O_1 + O_2 + ... + O_2)^* = O_2 + O_2 + ... + O_2$.

б) При $n \leqslant 5$ существует один и только один n-вершинный предполный граф G_{np2}, имеющий два неизоморфных минимальных вершинных 1-расширения: вида $O_2 + O_2 +$

$+...+O_2$ и тривиальное O_1+G_{np2}. Граф G_{np2} получается из графа G_{np1} удалением любого ребра, соединяющего две вершины степени $n-2$.

в) Любой n-вершинный предполный граф G, не изоморфный G_{np1} и G_{np2}, имеет единственное с точностью до изоморфизма минимальное вершинное 1-расширение — тривиальное: $(K_1+G)^*=K_1+(K_1+G)=K_2+G$.

Минимальный по числу вершин граф вида G_{np1} при $n=3$ представляет собой 3-вершинную цепь O_1+O_2, а его минимальное вершинное 1-расширение — 4-вершинный цикл O_2+O_2 (рис. 1). Минимальный по числу вершин граф вида G_{np2} при $n=5$ и два его минимальных вершинных 1-расширения изображены на рис. 2.

Рис. 1. Цепь P_3 и ее единственное минимальное вершинное 1-расширение — цикл C_4

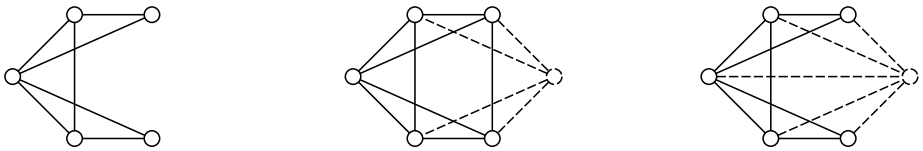


Рис. 2. Граф G_{5p2} и два его минимальных вершинных 1-расширения

Этот результат можно переформулировать следующим образом: почти все графы вида K_r+G, где $r\in N$, а G — произвольный граф, имеют единственное с точностью до изоморфизма минимальное вершинное 1-расширение, которое имеет вид $K_{r+1}+G$. Заметим, что K_{r+1} является минимальным вершинным 1-расширением графа K_r. Таким образом, для графа, представляющего собой соединение полного графа с произвольным графом, можно указать все минимальные вершинные 1-расширения. Возможна ли подобная ситуация для соединений еще каких-либо графов? Сформулируем этот вопрос в виде задачи.

Задача. Даны два графа G_1 и G_2 и их минимальные вершинные k-расширения. Найти все минимальные вершинные k-расширения графа G_1+G_2.

В общем случае решение задачи получить не удается, и результат для предполных графов пока является наилучшим. Рассмотрим еще несколько частных случаев, когда поставленная задача имеет решение.

Далее нас будут интересовать минимальные вершинные 1-расширения графов вида $G+G^*$ и $G+G^*+...+G^*=G+(G^*)^m$, где G^* — минимальное вершинное 1-расширение графа G.

1. Соединение графа с его точным вершинным 1-расширением

Дополнением графа $G=(V,\alpha)$ называется граф $G'=(V,\alpha')$, где $\alpha'=(V\times V)/(\alpha\cup\cup\Delta)$. Будем говорить, что граф G обладает свойством *дополнительности 1-расши-*

рения, если дополнение хотя бы одного его минимального вершинного 1-расширения является минимальным вершинным 1-расширением дополнения графа G.

Граф H называется *точным вершинным k-расширением* графа G, если любой граф, получающийся удалением произвольных k вершин графа H, изоморфен графу G. Понятие точного вершинного k-расширения было введено Ф. Харари и Дж. П. Хейзом в работе [5]. В работе [6] доказывается, что минимальное вершинное 1-расширение графа тогда и только тогда является его точным вершинным 1-расширением, когда граф обладает свойством дополнительности 1-расширения. Более того, если число вершин графа больше 1, то точное вершинное 1-расширение является единственным минимальным вершинным 1-расширением. Там же доказывается, что среди однородных графов свойством дополнительности 1-расширения обладают только полные и вполне несвязные графы. Для остальных графов результат формулируется следующим образом.

Теорема 2. Для того чтобы граф G обладал свойством дополнительности 1-расширения, необходимо и достаточно, чтобы граф G имел степенное множество вида $\{b, b-1\}$, причем число вершин степени $b-1$ в точности равнялось b, а его минимальное вершинное 1-расширение было однородным графом порядка b. При этом и граф G, и его дополнение имеют единственные с точностью до изоморфизма минимальные вершинные 1-расширения.

Теорема 3. Единственное минимальное вершинное 1-расширение графа $K_n + {+}(K_n)^*$ имеет вид $(K_n)^* + (K_n)^*$, причем

$$(K_n + (K_n)^*)^* = (K_n + K_{n+1})^* = (K_{2n+1})^* = K_{2n+2} = K_{n+1} + K_{n+1} = (K_n)^* + (K_n)^*.$$

Доказательство. Граф K_{n+1} является по теореме 1 единственным минимальным вершинным 1-расширением графа K_n и $K_n + K_m = K_{n+m}$, таким образом, все преобразования в формулировке теоремы корректны. ■

Теорема 4. Единственное минимальное вершинное 1-расширение графа $O_n + {+}(O_n)^*$ имеет вид $O_n^* + (O_n)^*$, причем

$$(O_n + (O_n)^*)^* = (O_n + O_{n+1})^* = (O_{2n+1})^* = O_{2n+2} = O_{n+1} + O_{n+1} = O_n^* + O_n^*.$$

Доказательство. Заметим, что граф O_{n+1} является точным вершинным 1-расширением графа O_n, а следовательно, граф O_{n+1} является единственным минимальным вершинным 1-расширением графа O_n. Таким образом, все преобразования в условии теоремы корректны. ■

Цепью P_n называется граф $G = (V, \alpha)$, где $V = \{v_1, v_2, ..., v_n\}$ и $\alpha = \{(v_i, v_j) : |i-j| = 1\}$, а *циклом* C_n — граф $G = (V, \alpha)$, где $V = \{v_1, v_2, ..., v_n\}$ и $\alpha = \{(v_i, v_j) : |i-j| = 1\} \cup \{(v_1, v_n), (v_n, v_1)\}$. Легко убедиться (см. [2]), что цикл C_{n+1} является минимальным вершинным 1-расширением цепи P_n.

Теорема 5. Единственное минимальное вершинное 1-расширение графа $P_n + P_n^*$ имеет вид $P_n^* + P_n^*$, причем $(P_n + P_n^*)^* = (P_n + C_{n+1})^* = C_{n+1} + C_{n+1} = P_n^* + P_n^*$.

Доказательство. Аналогично предыдущей теореме, цикл C_{n+1} является точным вершинным 1-расширением цепи P_n, а следовательно, граф C_{n+1} является единственным минимальным вершинным 1-расширением графа P_n, и все преобразования в условии теоремы корректны. ■

Теоремы 3–5 подсказывают более общее утверждение.

Теорема 6. Пусть G — граф, обладающий свойством дополнительности 1-расширения, а G^* — его точное вершинное 1-расширение. Тогда граф $G_+ = G+G^*+...+G^* = G + (G^*)^m$ также обладает свойством дополнительности 1-расширения, причем его точное вершинное 1-расширение имеет вид $G_+^* = G^* + G^* + ... + G^* = (G^*)^{m+1}$:

$$(G + G^* + ... + G^*)^* = G^* + G^* + ... + G^*.$$

Доказательство. Пусть n-вершинный граф G обладает свойством дополнительности 1-расширения, G^* — его точное вершинное 1-расширение, а граф G_+ получается соединением графа G и m графов G^*.

Заметим, что максимальный подграф графа $G_+^* = G^* + G^* + ... + G^*$ есть граф $G_+ = G+G^*+...+G^*$, поэтому граф G_+^* является точным вершинным 1-расширением графа G_+, а следовательно, и единственным его минимальным вершинным 1-расширением, что доказывает корректность формулы в условии теоремы. ■

Следующий результат, являющийся усилением предыдущей теоремы, связывает операции соединения и дополнения с конструкцией минимального вершинного 1-расширения.

Теорема 7. Граф $G_+ = G+G^*+...+G^* = G+(G^*)^m$, где G — произвольный граф, а G^* — его минимальное вершинное 1-расширение, тогда и только тогда обладает свойством дополнительности 1-расширения, когда граф G^* является точным вершинным 1-расширением графа G. При этом граф $G_+^* = G^* + G^* + ... + G^* = (G^*)^{m+1}$ является точным вершинным 1-расширением графа G_+.

Доказательство. Необходимость. Пусть G — n-вершинный граф, G^* — его минимальное вершинное 1-расширение, а граф $G_+ = G + G^* + ... + G^* = G + (G^*)^m$ обладает свойством дополнительности 1-расширения.

Введем обозначения: $a = \max\limits_{v\in V(G)} \mathrm{d}(v)$, $b = \min\limits_{v\in V(G)} \mathrm{d}(v)$;
аналогично для графа G^*: $a^* = \max\limits_{v\in V(G^*)} \mathrm{d}(v)$, $b^* = \min\limits_{v\in V(G^*)} \mathrm{d}(v)$.

Очевидно, что $a \leqslant a^*$ и $b \leqslant b^*$.

Определим наибольшие и наименьшие степени вершин в графе G_+.

Вершины, имеющие в графе G степени a и b, в графе G_+ будут иметь степени $a + m(n+1) = a + mn + m$ и $b + m(n+1) = b + mn + m$ соответственно.

Вершины, имеющие в графе G^* степени a^* и b^*, в графе G_+ будут иметь степени $a^* + (m-1)(n+1) + n = a^* + mn + m - 1$ и $b^* + (m-1)(n+1) + n = b^* + mn + m - 1$.

По условию граф G_+ обладает свойством дополнительности 1-расширения, тогда, как было указано выше, либо граф G_+ является однородным и в этом случае он является вполне несвязным или полным и

$$\min\{a + mn + m, a^* + mn + m - 1\} = \max\{b + mn + m, b^* + mn + m - 1\},$$

то есть $\min\{a, a^* - 1\} = \max\{b, b^* - 1\}$, либо он попадает под действие теоремы 2:

$$\min\{a + mn + m, a^* + mn + m - 1\} = \max\{b + mn + m, b^* + mn + m - 1\} - 1,$$

или

$$\min\{a, a^* - 1\} = \max\{b, b^* - 1\} - 1. \qquad (1)$$

Рассмотрим четыре случая.

1) $a = a^*,\ b = b^*$. Тогда из соотношения (1) получаем $a = a^* = b = b^*$, что возможно лишь тогда, когда G и G^* являются вполне несвязными графами, то есть G^* является точным вершинным 1-расширением графа G.

2) $a = a^*,\ b < b^*$. Из соотношения (1) получаем $a-1 = b^*-2 > b-2$, откуда $a > b-1$, что возможно лишь при $a = b$. Поскольку $a = a^*$, то граф G содержит изолированные вершины, значит, $a = b = 0$ и граф G является вполне несвязным. Однако минимальное вершинное 1-расширение вполне несвязного графа является также вполне несвязным графом, то есть $a^* = b^* = 0$, что противоречит условию $a = a^*$, $b < b^*$. Следовательно, случая 2 быть не может.

3) $a < a^*,\ b = b^*$. Из (1) получаем, что $a = b - 1$. Далее $a^* > a = b - 1 = b^* - 1$, но поскольку $a^* \leqslant b^*$, то $a^* = b^*$. Таким образом, $a + 1 = b = a^* = b^*$.

Обозначим через n_1, n_2 число вершин степеней a, b графа G соответственно. Очевидно, что граф G_+ имеет степенное множество $\{a + mn + m, b + mn + m\}$, причем число вершин степени $a + mn + m$ равно $n_1 + (n + 1)m$, а число вершин степени $b+mn+m$ равно n_2. По предположению граф G_+ обладает свойством дополнительности 1-расширения, причем его степенное множество имеет вид $\{a+mn+m, b+mn+m\}$, где $a + mn + m = b + mn + m - 1$. По теореме 2 число вершин степени $a + mn + m$ должно быть в точности равно $b+mn+m$, то есть $n_1+(n+1)m = b+mn+m$, откуда $n_1 = b$.

Таким образом, граф G имеет степенное множество вида $\{b - 1, b\}$, число вершин степени $b - 1$ в точности равно b, и его минимальное вершинное 1-расширение — граф G^* — является однородным графом порядка b. По теореме 2 граф G обладает свойством дополнительности 1-расширения, а граф G^* является его точным вершинным 1-расширением.

4) $a < a^*,\ b < b^*$. Этот случай возможен лишь тогда, когда G и G^* являются полными графами.

Таким образом, в каждом из возможных случаев 1, 3 и 4 граф G^* является точным вершинным 1-расширением графа G, что доказывает необходимость. Достаточность следует из теоремы 6. ■

2. Соединение графа с его минимальным вершинным 1-расширением

На основании предыдущих рассуждений представляется разумным высказать предположение (см. [7]).

Гипотеза 1. Пусть G — произвольный граф, а G^* — некоторое его минимальное вершинное 1-расширение. Тогда граф вида $G + G^*$ всегда имеет единственное минимальное вершинное 1-расширение, и оно может быть представлено в виде $G^* + G^*$, то есть

$$(G + G^*)^* = G^* + G^*.$$

Однако оказалось, что ситуация является более сложной, и удалось найти два контрпримера к этой гипотезе.

Обозначим однородный n-вершинный граф порядка p через $R_{n,p}$. В работе [4] доказывается, что при четном n существует единственный с точностью до изоморфизма однородный граф $R_{n,n-2}$, причем он имеет вид $O_2 + ... + O_2$, а его единственным минимальным вершинным 1-расширением является тривиальное 1-расширение.

Теорема 8. Пусть G — граф вида $R_{n,n-2}$, а G^* — его минимальное вершинное 1-расширение. Тогда минимальное вершинное 1-расширение графа $G+G^*$ единственно с точностью до изоморфизма и имеет вид $R_{2n+2,2n}$.

Доказательство. Как было отмечено, граф $R_{n,n-2}$ имеет вид $O_2 + ... + O_2$, а его минимальное вершинное 1-расширение G^* можно записать как $G^* = K_1 + O_2 + ... + O_2$.

Таким образом, граф $G + G^*$ имеет вид

$$G + G^* = (O_2 + ... + O_2) + (K_1 + O_2 + ... + O_2) = K_1 + O_2 + ... + O_2$$

и является графом вида G_{np1}. По теореме 1 получаем требуемое утверждение. ■

Теорема 9. Пусть G — предполный граф вида G_{np2}, а G^* — его минимальное вершинное 1-расширение вида $O_2 + ... + O_2$. Тогда граф $G + G^*$ имеет два неизоморфных минимальных вершинных 1-расширения, одно из которых имеет вид $G^* + G^*$, а второе — $O_2 + ... + O_2$.

Доказательство. По теореме 1 граф G_{np2} имеет два неизоморфных минимальных вершинных 1-расширения: тривиальное 1-расширение и расширение вида $O_2 + ... + O_2$. Если рассматривать первое минимальное вершинное 1-расширение, то утверждение гипотезы, как несложно заметить, справедливо. Рассмотрим соединение графа G_{np2} со вторым его минимальным вершинным 1-расширением $O_2 + ... + O_2$.

Легко увидеть, что это соединение снова является предполным графом вида $G_{(2n+2)p2}$ и, следовательно, по теореме 1 имеет два неизоморфных минимальных вершинных 1-расширения. ■

Таким образом, в общем виде гипотеза 1 является ошибочной. Однако для большого числа графов ее утверждение является справедливым, что в дополнении к предыдущим теоремам показывает и следующая

Теорема 10. Пусть G — граф, неизоморфный графу вида $R_{n,n-2}$, а его тривиальное 1-расширение G^* является и его минимальным вершинным 1-расширением. Тогда минимальное вершинное 1-расширение графа $G + G^*$ единственно с точностью до изоморфизма и имеет вид $G^* + G^*$, то есть $(G + G^*)^* = G^* + G^*$.

Доказательство. Так как G^* является тривиальным 1-расширением графа G, то $G^* = K_1 + G$ и в G^* есть хотя бы одна полная вершина. Граф $G + G^* = G + (K_1 + G) = K_1 + (G + G)$ также является предполным. По теореме 1 если граф $G + G^*$ не является графом вида G_{np1} или G_{np2}, то он имеет единственное минимальное вершинное 1-расширение, которым является его тривиальное 1-расширение, то есть получаем утверждение теоремы. В самом деле, тривиальное 1-расширение графа $G + G^*$ можно записать так: $K_1 + (G + G^*) = (K_1 + G) + G^* = G^* + G^*$.

Вспомним, что граф вида G_{np1} — это граф вида $K_1 + O_2 + ... + O_2$, и поэтому нужно исключить графы G вида $R_{n,n-2}$.

Граф вида G_{np2} — это граф вида $K_1 + (O_2 + ... + O_2 - e)$, где e — некоторое ребро. В нашем случае граф $G + G^*$ не может быть изоморфен графу G_{np2}, так как, исключив из рассмотрения полную вершину, нужно подобрать граф G таким образом, чтобы выполнялось $G + G = O_2 + ... + O_2 - e$, что невозможно. ■

Много ли графов попадает под действие этой теоремы? Как показывает следующее утверждение, это почти все предполные графы, но и кроме них многие графы имеют минимальное вершинное 1-расширение, которым является их тривиальное 1-расширение. Так, например, по материалам работы [8] из 1043 7-вершинных графов 405 имеют минимальным вершинным 1-расширением тривиальное 1-расширение. Для 6-вершинных — 65 из 155; для 5-вершинных — 10 из 33.

Следствие 1. Пусть G — произвольный предполный граф не вида G_{np2}, а G^* — его минимальное вершинное 1-расширение. Тогда граф $G + G^*$ имеет единственное

минимальное вершинное 1-расширение, и оно может быть представлено в виде $G^* + G^*$, то есть $(G + G^*)^* = G^* + G^*$.

Доказательство. По теореме 1 все предполные графы, кроме графов вида G_{np1} и G_{np2}, имеют минимальным вершинным 1-расширением тривиальное 1-расширение и, таким образом, попадают под условие теоремы 10. Остается рассмотреть предполные графы вида G_{np1}. По теореме 1 минимальное вершинное 1-расширение графа вида G_{np1} является точным вершинным 1-расширением, и по теореме 6 получаем требуемое утверждение. ■

Заключение

Удалось показать, что гипотеза о том, что граф вида $G+G^*$, где G — произвольный граф, а G^* — некоторое его минимальное вершинное 1-расширение, имеет единственное с точностью до изоморфизма минимальное вершинное 1-расширение вида $G^* + G^*$, в целом является ложной: такой граф может иметь и более одного минимального вершинного 1-расширения или одно, но иного вида. Тем не менее доказывается, что для многих графов утверждение гипотезы является справедливым.

ЛИТЕРАТУРА

1. *Богомолов А. М., Салий В. Н.* Алгебраические основы теории дискретных систем. М.: Наука, 1997.
2. *Hayes J. P.* A graph model for fault-tolerant computing system // IEEE Trans. Comput. 1976. V. C.25. No. 9. P. 875–884.
3. *Абросимов М. Б.* О сложности некоторых задач, связанных с расширениями графов // Матем. заметки. 2010. № 5(88). С. 643–650.
4. *Абросимов М. Б.* Минимальные k-расширения предполных графов // Изв. вузов. Математика. 2003. № 6(493). С. 3–11.
5. *Harary F. and Hayes J. P.* Node fault tolerance in graphs // Networks. 1996. V. 27. P. 19–23.
6. *Абросимов М. Б.* Минимальные расширения дополнений графов // Теоретические задачи информатики и ее приложений. Саратов: Изд-во Сарат. ун-та, 2001. Вып. 4. С. 11–19.
7. *Абросимов М. Б.* Минимальные расширения графов: автореф. дис. ... канд. физ.-мат. наук. Саратов: СГУ, 2001. 16 с.
8. *Абросимов М. Б.* Минимальные вершинные расширения 4-, 5-, 6- и 7-вершинных графов. Саратов: СГУ, 2000. 26 с. Деп. в ВИНИТИ 06.09.2000, № 2352 В00.

2011 Прикладная теория графов №4(14)

DOI 10.17223/20710410/14/6

УДК 519.177+519.174.2+519.171.4

КРАТНОСТИ СУММ В ЯВНЫХ ФОРМУЛАХ ДЛЯ ПОДСЧЁТА ЦИКЛОВ ФИКСИРОВАННОЙ ДЛИНЫ В НЕОРИЕНТИРОВАННЫХ ГРАФАХ

А. Н. Воропаев

Петрозаводский государственный университет, г. Петрозаводск, Россия

E-mail: voropaev@psu.karelia.ru

Явные формулы для подсчёта циклов длиной k представляют собой комбинации сумм, соответствующих формам замкнутых маршрутов длиной k. Ранее было показано, что наибольшая кратность суммы в формуле равна $[k/2]$, начиная с $k = 8$. В данной работе исследуется вопрос, каких значений могут достигать кратности сумм для частных семейств графов: двудольных, без треугольников, планарных, с ограниченными степенями вершин, а также их пересечений. Оказалось, что при больших значениях k только для двудольных графов и для графов со степенями вершин не более трёх наибольшая кратность суммы уменьшается на 1, если $k \equiv 2, 3 \pmod 4$. В случае $k \leqslant 20$ возникает ряд исключений, когда для некоторых семейств графов наибольшая кратность уменьшается на 1 или 2.

Ключевые слова: *подсчёт циклов в графах, формы замкнутых маршрутов, призматические графы.*

Введение

Количество циклов заданной длины — важная числовая характеристика структуры графа. Например, одно из отличительных свойств реальных сетей, называемое кластеризацией, состоит в том, что вершины, имеющие общую смежную вершину, часто сами оказываются смежны. Мерой этого свойства служит коэффициент кластеризации $C = 3\,c_3/p_3 \in [0; 1]$, где c_3 и p_3 — числа циклов и цепей с тремя вершинами [1, 2]. Двудольные графы, для которых всегда $C = 0$, характеризуют двудольным коэффициентом кластеризации, вычисляемым по формуле $C_b = 4\,c_4/p_4$.

Более полную информацию о сети даёт статистика циклов по длинам. Однако всю статистику находить крайне затруднительно, поэтому обычно ограничиваются подсчётом коротких циклов. Например, авторы [1] вывели формулы для степени баланса знакового графа, основанные на подсчёте циклов длиной 3 и 4. В [3] числа циклов длиной 3, 4 и 5 используются при исследовании схем городских улиц. С помощью статистики коротких циклов можно оценивать качество кодов с низкой плотностью проверок на чётность [4].

Вычислительная сложность подсчёта циклов заданной длины малоизучена. Известно, что задача подсчёта гамильтоновых циклов является $\#P$-полной [5], причём остаётся таковой даже в случае планарных регулярных графов степени три [6]. В рамках теории параметрической сложности авторы [7] показали, что задача подсчёта циклов длиной k в произвольном графе $\#W[1]$-полна, когда параметром является длина цикла k. Другими словами, не существует $f(k) \cdot n^{\mathrm{const}}$-алгоритма решения этой задачи, где n — порядок графа. При этом, очевидно, перебор всех циклов длиной k в графах со степенями вершин не более фиксированного значения d можно осуществлять со сложностью $\mathrm{O}((d-1)^k \cdot n)$.

Результат [7] означает, что порядок сложности подсчёта циклов длиной k в произвольных графах неизбежно растёт вместе с k, но не даёт информации о характере этого роста. В литературе не встречаются и оценки сложности данной задачи в случае частных семейств графов. Многие известные универсальные алгоритмы — для произвольных графов и произвольных значений длины цикла — характеризуются сложностью $\mathrm{O}(n^k)$ или $\mathrm{O}(n^{k-1})$ (при фиксированном k), что сопоставимо с ростом самого количества циклов длиной k. Ряд таких методов перечислен в [8].

Среди универсальных алгоритмов подсчёта циклов длиной k наиболее эффективны явные формулы, известные для небольших значений длины [8 – 11]. При $k \leqslant 7$ вычисления по этим формулам выполняются с тем же порядком сложности, что умножение $n \times n$-матриц (n — порядок графа) [12]. Начиная с $k = 8$, вычислительная сложность явных выражений оценивается величиной $\mathrm{O}(n^{[k/2]})$ [8, 11], которая связана с появлением $[k/2]$-кратных сумм. Сами формулы выведены до значения $k = 13$ [8].

Способ вывода явных выражений [8] интересен также и тем, что позволяет легко учитывать некоторые структурные свойства графов при выводе формул. В результате получаются более компактные и, возможно, более эффективные выражения. Например, учёт двудольности в случаях $k = 8, 10, 14$ уменьшает порядок сложности на 1, а при подсчёте циклов длиной не более удвоенного обхвата в двудольных графах наиболее трудоёмкой операцией является умножение матриц [11, 13].

В данной работе исследуется зависимость наибольшей кратности суммы в явных формулах от длины цикла k для нескольких частных семейств графов. Рассматриваются множества двудольных графов, графов без треугольников, планарных графов, графов со степенями вершин не более 3 и их различные пересечения (12 семейств).

В п. 2 представлена структура явных выражений на примере случая $k = 7$, в п. 3 приведены правила упрощения сумм (уменьшения их кратностей). Именно в рамках этих правил в п. 4 выполняются оценки наибольших кратностей сумм. В большинстве случаев верхняя оценка равна $[k/2]$, как для произвольных графов. Только для двудольных графов и для графов со степенями вершин не более 3 оценка снижается на 1 при $k \equiv 2, 3 \pmod 4$.

В п. 5 приводятся параметрические семейства форм для значений $k \geqslant 16$, на которых достигаются оценки, выведенные в п. 4. Для меньших значений k в п. 6 указаны отдельные примеры «тяжёлых» форм — таких, которые приводят к суммам наибольшей кратности. При $10 \leqslant k \leqslant 20$ возникают 15 исключений, когда наибольшие кратности сумм отклоняются от общих закономерностей на 1 или 2 в меньшую сторону. Все эти случаи отмечены в Приложении А, которое содержит таблицы со значениями наибольшей кратности и количеством сумм наибольшей кратности. В Приложении Б приводится список всех 18 «тяжёлых» форм для $k = 8, 9, 10, 11$.

1. Основные определения

В работе используется в основном терминология [14] и рассматриваются неориентированные конечные графы без петель и кратных рёбер.

Маршрутом длиной k называется упорядоченный набор $(v_1, v_2, \ldots, v_{k+1})$ вершин графа, такой, что вершины v_i и v_{i+1} смежны. Цепь — это маршрут, в котором все вершины различны, а цикл есть маршрут длиной не менее 3, в котором все вершины, исключая последнюю, различны, а последняя вершина совпадает с первой. При подсчёте циклы, отличающиеся только выбором начальной вершины или направления обхода вершин, рассматриваются как один.

Гомоморфизм графа $G_1 = (V_1; E_1)$ на граф $G_2 = (V_2; E_2)$ — это сюръективное отображение $f : V_1 \to V_2$, такое, что $E_2 = \{\{f(u), f(v)\} : \{u, v\} \in E_1\}$. Другими словами, граф G_2 получается из G_1 путём ряда отождествлений пар несмежных вершин. Сам граф G_2 называют гомоморфным образом G_1.

В п. 2–4 и 7 символ n закреплён за количеством вершин (порядком) графа, в котором подсчитываются циклы. Граф, состоящий из одного простого цикла с k вершинами, обозначается символом C_k, а из одной простой цепи с k вершинами — P_k.

Для записи элементов матрицы смежности графа будем использовать обозначение a_{ij}, а для элементов её l-х степеней — обозначение $a_{ij}^{(l)}$. Диагональ квадрата матрицы смежности состоит из степеней вершин графа, поэтому вместо $a_{ii}^{(2)}$ обычно применяется символ d_i.

2. Структура формул

Сопоставим с маршрутом подграф, состоящий из вершин и рёбер, по которым проходит маршрут. Например, циклу длиной k соответствует подграф, изоморфный C_k, а замкнутому маршруту чётной длины k, проходящему по единственному ребру, — подграф, изоморфный P_2. При этом один и тот же подграф может соответствовать нескольким маршрутам. Так, для каждого подграфа, изоморфного C_k, существует $2\,k$ циклов длиной k, а для всякого подграфа, изоморфного P_2, — только два замкнутых маршрута чётной длины k. Отвлекаясь от обозначений вершин, будем говорить о графе, соответствующем маршруту, как о форме этого маршрута. На рис. 1 изображены всевозможные формы замкнутых маршрутов длиной 7. Номера вершинам графов H_1, H_5 и H_6 приписаны в целях составления сумм, рассматриваемых ниже.

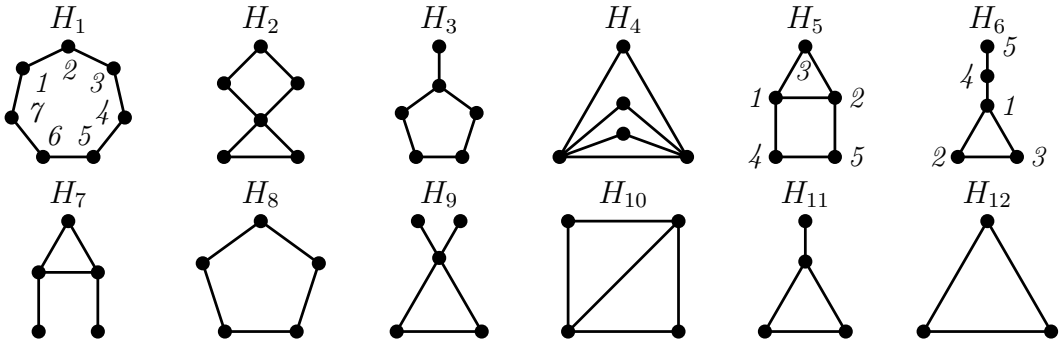


Рис. 1. Всевозможные формы замкнутых маршрутов длиной 7

Каждая форма замкнутых маршрутов длиной k является гомоморфным образом цикла C_k и наоборот. Гомоморфные образы самих форм для заданного значения k также являются формами замкнутых маршрутов длиной k.

Для подсчёта подграфов, изоморфных форме $G = (\{1, \ldots, l\}; E)$, удобно воспользоваться суммой

$$\alpha(G) = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \ldots \sum_{i_l=1}^{n} \prod_{\{u,v\} \in E} a_{i_u i_v}. \tag{1}$$

Количество индексов, по которым выполняется суммирование, будем называть кратностью суммы. В (1) каждый индекс соответствует вершине формы, а каждый множитель — элемент матрицы смежности — ребру формы. Например, формам H_1 и H_5

(рис. 1) соответствуют суммы

$$\alpha(H_1) = \sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\dots\sum_{i_7=1}^{n} a_{i_1i_2}a_{i_2i_3}\dots a_{i_7i_1} = \sum_{i_1=1}^{n} a^{(7)}_{i_1i_1},$$

$$\alpha(H_5) = \sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\dots\sum_{i_5=1}^{n} a_{i_1i_2}a_{i_1i_3}a_{i_3i_2}a_{i_1i_4}a_{i_4i_5}a_{i_5i_2} =$$

$$= \sum_{i_1=1}^{n}\sum_{i_2=1}^{n} a_{i_1i_2}\left(\sum_{i_3=1}^{n} a_{i_1i_3}a_{i_3i_2}\right)\left(\sum_{i_4=1}^{n}\sum_{i_5=1}^{n} a_{i_1i_4}a_{i_4i_5}a_{i_5i_2}\right) = \sum_{i_1=1}^{n}\sum_{i_2=1}^{n} a_{i_1i_2}a^{(2)}_{i_1i_2}a^{(3)}_{i_1i_2}.$$

Сама по себе величина $\alpha(G)$ не равна количеству подграфов, изоморфных G, так как за счёт совпадения значений индексов в сумму вносят вклад и формы меньшего порядка. Например, при вычислении суммы $\alpha(H_8)$ учитываются подграфы, изоморфные как форме H_8, так и формам H_{11} и H_{12}. Кроме того, один и тот же подграф может встречаться несколько раз. Так, в сумме $\alpha(C_k)$ каждый цикл учитывается $2\,k$ раз, по числу способов выбора начальной вершины и направления обхода вершин.

Обозначим символом $\beta(G)$ количество подграфов, изоморфных G. Располагая совокупностью сумм $\alpha(G)$, можно составить треугольную систему линейных уравнений с неотрицательными целыми коэффициентами для величин $\beta(G)$:

$$\alpha(G) = \sum_{H} \gamma_G(H)\beta(H).$$

Здесь G пробегает множество всех гомоморфных образов цикла C_k; H пробегает множество всех гомоморфных образов графа G; $\gamma_G(H)$ равно числу гомоморфизмов графа G на граф H. Решение этой системы даёт выражения величин $\beta(G)$ в виде комбинаций сумм $\alpha(H)$ с рациональными коэффициентами. Например, для количества циклов длиной 7, равного $\beta(C_7)$, получается следующая формула:

$$\begin{aligned} c_7 = {} & \frac{1}{14}\sum_{i=1}^{n} a^{(7)}_{ii} + \frac{1}{2}\sum_{i=1}^{n}\Big(-a^{(4)}_{ii}a^{(3)}_{ii} - a^{(5)}_{ii}d_i + a^{(5)}_{ii} + 2\,a^{(3)}_{ii}d_i^2 - 11\,a^{(3)}_{ii}d_i + 8\,a^{(3)}_{ii}\Big) + \\ & + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\Big(\Big(a^{(2)}_{ij}\Big)^3 a_{ij} + 3\,a^{(3)}_{ij}a^{(2)}_{ij}a_{ij} + a^{(2)}_{ij}a^{(3)}_{ii} + a^{(2)}_{ij}a_{ij}d_id_j - 4\,\Big(a^{(2)}_{ij}\Big)^2 a_{ij}\Big). \end{aligned} \tag{2}$$

Соотношения, подобные (2), в которых количество циклов напрямую представлено через элементы матрицы смежности и вспомогательных матриц (например, степеней матрицы смежности), будем называть явными формулами. Структура явных формул и их вывод подробно обсуждаются в [8].

3. Уменьшение кратностей сумм

В исходном виде сумма (1) имеет кратность, равную порядку формы l. Однако за счёт использования вспомогательных матриц (например, степеней матрицы смежности, как в (2)) и векторов кратность суммы может быть уменьшена. Если какой-либо индекс h встречается в парах ровно с одним другим индексом i, то суммирование по h можно выполнить отдельно и сохранить результат в векторе (v_i) для дальнейших расчётов. Аналогично исключается индекс h, встречающийся в парах с двумя другими индексами i и j. Тогда возникает матрица (b_{ij}). Например, для формы H_6 (рис. 1) можно выполнить следующие преобразования:

$$\alpha(H_6) = \sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\dots\sum_{i_5=1}^{n} a_{i_1i_2}a_{i_2i_3}a_{i_3i_1}a_{i_1i_4}a_{i_4i_5} =$$

$$= \sum_{i_1=1}^{n}\left(\sum_{i_2=1}^{n}\sum_{i_3=1}^{n} a_{i_1i_2}a_{i_2i_3}a_{i_3i_1}\right)\left(\sum_{i_4=1}^{n}\sum_{i_5=1}^{n} a_{i_1i_4}a_{i_4i_5}\right) = \sum_{i_1=1}^{n} a^{(3)}_{i_1i_1}v_{i_1}, \quad \text{где} \quad v_i = \sum_{j=1}^{n} a^{(2)}_{ij}.$$

Если достигнута кратность три, то каждый индекс заведомо встречается в парах не более чем с двумя другими индексами, и возможно дальнейшее уменьшение кратности. При условии $k \leqslant 7$ кратности всех сумм, входящих в формулу для подсчёта циклов длиной k, понижаются до значений менее трёх, как, например, в (2). Наиболее трудоёмкой операцией при этом оказывается умножение матриц.

Начиная со значения $k = 8$, появляются суммы кратностью более трёх, в которых каждый индекс встречается в парах более чем с двумя другими индексами. К таким суммам описанные выше правила понижения кратности неприменимы. В результате сложность вычисления всего выражения для количества циклов определяется наибольшей кратностью суммы, входящей в него. Например, в формуле для подсчёта циклов длиной 8 участвует четырёхкратная сумма, соответствующая форме K_4 (это единственная «тяжёлая» сумма во всём выражении).

4. Верхние оценки кратностей сумм

Правила понижения кратности суммы, представленные выше, применимы, пока остаётся хотя бы один индекс, встречающийся в парах не более чем с двумя другими индексами. Очевидно, что в результате каждого применения данных правил число «парных» индексов для всякого индекса не увеличивается. Следовательно, кратность суммы после упрощения не превосходит количества индексов исходной суммы (1), каждый из которых встречается в парах не менее чем с тремя другими индексами. Иными словами, кратность суммы $\alpha(G)$ после упрощения оценивается числом вершин формы G со степенями не менее трёх. Оценим число таких вершин.

Утверждение 1. Пусть G — форма замкнутых маршрутов длиной k. Тогда количество вершин формы G со степенями не менее трёх не превосходит $[k/2]$.

Доказательство. Обозначим I множество всех вершин формы G, а $I' \subset I$ — множество всех вершин со степенями не менее трёх. Поскольку форма G является образом цикла C_k при некотором гомоморфизме f, то каждая вершина $i \in I'$ имеет по крайней мере два прообраза в C_k при f. Следовательно,

$$k = \sum_{i \in I} |f^{-1}(i)| \geqslant \sum_{i \in I'} |f^{-1}(i)| \geqslant 2\,|I'|, \tag{3}$$

откуда $|I'| \leqslant [k/2]$. ■

Замечание 1. На основе соотношения (3) можно сделать вывод о структуре формы, содержащей $[k/2]$ вершин степени не менее трёх. При чётном k такая форма имеет $k/2$ вершин, и их степени равны 3 или 4. Если же k нечётно, то возможны два варианта:

1) $[k/2]$ вершин степени 3 или 4 и одна вершина степени 1 или 2;
2) $([k/2] - 1)$ вершин степени 3 или 4 и одна вершина степени 3, 4, 5 или 6.

Оценка, сформулированная в утверждении 1, встречается в [8, 11]. В [8] также приводится класс форм, которым соответствуют $[k/2]$-кратные суммы, начиная с $k = 8$. Данная работа посвящена изучению наибольших кратностей сумм, соответствующих формам с определёнными свойствами: двудольным, без треугольников, планарным, со степенями вершин не более трёх. Оказывается, что для перечисленных множеств форм, а также их пересечений достигаются почти те же значения кратности, что и в общем случае. Только для двудольных графов и для графов со степенями вершин не более трёх оценка снижается на 1 при $k \equiv 2, 3 \pmod 4$.

Утверждение 2. Пусть G — двудольная форма замкнутых маршрутов длины k. Тогда количество вершин формы G со степенями не менее трёх не превосходит $2[k/4]$.

Доказательство. При $k \equiv 0 \,(\mathrm{mod}\, 4)$ значение $2[k/4]$ совпадает с общей оценкой $[k/2]$. Покажем, что в случае $k \equiv 2 \,(\mathrm{mod}\, 4)$ всякая форма H с $k/2$ вершинами степени не менее трёх недвудольна.

Согласно замечанию 1 и соотношению (3), каждая вершина формы H является образом ровно двух вершин графа C_k при гомоморфизме f. Следовательно, множество вершин C_k можно разбить на пары $\{i, j\}$, такие, что $f(i) = f(j)$. Пусть вершины цикла C_k пронумерованы числами от 1 по k в некотором направлении обхода.

В силу $k \equiv 2 \,(\mathrm{mod}\, 4)$ множество $\{1, \ldots, k\}$ содержит нечётное количество чётных чисел и нечётное количество нечётных чисел. Это означает, что в любом разбиении этого множества на пары обязательно найдётся пара чисел разной чётности. Но тогда в форме H существует замкнутый маршрут нечётной длины и, следовательно, нечётный цикл. По теореме Кёнига заключаем, что форма H недвудольна.

Таким образом, при $k \equiv 2 \,(\mathrm{mod}\, 4)$ форма G содержит не более $k/2 - 1$ вершин степени не менее трёх. ■

Утверждение 3. Пусть G — форма замкнутых маршрутов длиной k, вершины которой имеют степени два или три. Тогда количество вершин формы G степени три не превосходит $2[k/4]$.

Доказательство. Когда $k \equiv 0, 1 \,(\mathrm{mod}\, 4)$, значение $2[k/4]$ совпадает с общей оценкой $[k/2]$. Рассмотрим случай $k \equiv 2 \,(\mathrm{mod}\, 4)$. Предположим, что форма G содержит $k/2$ вершин степени три. В силу замечания 1 форма G является кубическим графом. При этом число вершин формы нечётно, что противоречит лемме о рукопожатиях. Следовательно, форма G содержит не более $k/2 - 1$ вершин степени три.

В случае $k \equiv 3 \,(\mathrm{mod}\, 4)$ форма G также не может содержать $[k/2]$ вершин степени три. Иначе, согласно условию данного утверждения и замечанию 1, сумма степеней всех вершин снова оказалась бы нечётной. ■

Замечание 2. В условии утверждения 3 исключены формы с висячими вершинами. Среди них можно отыскать формы, имеющие $[k/2]$ вершин степени три при $k \equiv 2, 3 \,(\mathrm{mod}\, 4)$. Например, одной из форм замкнутых маршрутов длиной 11 является граф, изображённый на рис. 2.

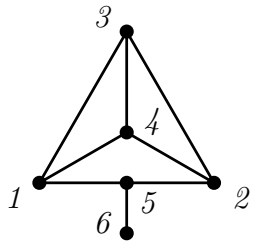


Рис. 2. Форма замкнутых маршрутов длиной 11, содержащая 5 вершин степени 3

Однако для всех таких форм кратности соответствующих сумм заведомо уменьшаются до значения менее $[k/2]$. Индексы, соответствующие висячей вершине и смежной с ней вершине (в примере — *6* и *5*), в совокупности встречаются в парах только с двумя другими индексами. Следовательно, оба этих индекса можно исключить, используя вспомогательную матрицу с элементами $b_{ij} = \sum a_{ik} d_k a_{kj}$. Так, для формы, указанной на рис. 2, получаем сумму

$$\sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \ldots \sum_{i_6=1}^{n} \left(a_{i_1 i_5} a_{i_5 i_6} a_{i_5 i_2} \right) a_{i_2 i_3} a_{i_3 i_1} a_{i_1 i_4} a_{i_2 i_4} a_{i_3 i_4} =$$

$$= \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \sum_{i_3=1}^{n} \sum_{i_4=1}^{n} b_{i_1 i_2} a_{i_2 i_3} a_{i_3 i_1} a_{i_1 i_4} a_{i_2 i_4} a_{i_3 i_4}.$$

Далее приводятся параметрические классы форм и отдельные формы, на которых достигаются установленные выше верхние оценки (за исключением нескольких случаев). При этом, в отличие от примера из замечания 2, в соответствующих суммах не исключается ни один индекс, «смежный», по крайней мере, с тремя другими индексами. Таким образом, представленные ниже примеры форм обеспечивают наибольшие кратности сумм в выражениях для подсчёта циклов; такие формы будем называть «тяжёлыми».

Частные «тяжёлые» формы для значений $k \leqslant 20$ (см. Приложение А, табл. 1 и 2) были найдены в ходе анализа списков всех форм, сгенерированных в системе компьютерной алгебры [15]. Параметрические семейства построены путём «экстраполяции» ряда частных форм. Далее обоснуем, что предъявленные графы являются формами замкнутых маршрутов определённой длины.

5. Примеры «тяжёлых» форм для значений $k \geqslant 16$

Начиная со значения длины 16, удобно определить параметрические классы «тяжёлых» форм, одновременно обладающие многими свойствами. Под свойствами подразумеваются двудольность, отсутствие треугольников, планарность, ограниченность степеней вершин значением три. Само число 16 — это наименьшее значение длины, для которого существует форма, обеспечивающая наибольшую возможную кратность суммы $[k/2]$ и обладающая всеми перечисленными свойствами (см. табл. 1). Кроме того, при $k > 16$ возникают всего два исключения, когда наибольшие кратности оказываются меньше верхних оценок.

Основу примеров составляют призматические графы (рис. 3).

Определение 1. Призматическим графом называется граф $C_n \times P_2$, $n \geqslant 3$.

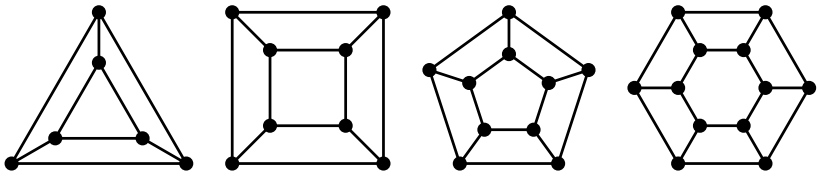


Рис. 3. Примеры призматических графов

Призматический граф является планарным регулярным графом степени 3, а при чётном значении n также двудольным. При $n > 3$ граф $C_n \times P_2$ не содержит треугольников. Именно наличие данных свойств в совокупности и простота структуры призматических графов послужили причиной их выбора в качестве примеров «тяжёлых» форм.

Утверждение 4. Призматический граф $C_n \times P_2$ является формой замкнутых маршрутов длиной $4n$.

Доказательство. Укажем пример замкнутого маршрута длиной $4n$, проходящего по всем рёбрам призматического графа. Из какой-либо вершины «внешнего» цикла можно попасть во «внутренний» цикл и, обойдя его, вернуться в исходную вершину (рис. 4). Затем остаётся пройти по «внешнему» циклу, включая рёбра, которые соединяют его с «внутренним» циклом. В результате на $2n$ рёбер обоих циклов затрачивается по одному шагу, а на остальные n рёбер — по два. Следовательно, длина маршрута в точности равна $4n$. ■

Призматические графы образуют бесконечную последовательность «тяжёлых» форм, но лишь для значений длины маршрута $k \equiv 0 \pmod 4$, без соблюдения двудоль-

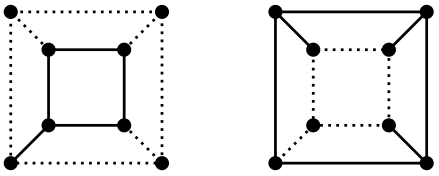


Рис. 4. Построение замкнутого маршрута в призматическом графе $C_4 \times P_2$

ности, и $k \equiv 0 \pmod 8$, с соблюдением двудольности. Путём незначительной модификации призматических графов можно получить примеры «тяжёлых» форм для остальных случаев. Утверждения 5 и 6, а также их доказательства сформулированы в терминах плоского изображения призматического графа, как на рис. 3.

В следующем утверждении указан класс «тяжёлых» форм замкнутых маршрутов длиной $k \equiv 2 \pmod 4$, $k \geqslant 22$. Формы из этого класса недвудольны и имеют по одной вершине степени четыре, но они планарны и не содержат треугольников.

Утверждение 5. Пусть $G = C_n \times P_2$, $n \geqslant 5$, а (v_1, v_2, v_3, v_4) — цепочка во «внутреннем» цикле графа G (рис. 5). Тогда граф, получаемый из G удалением ребра $\{v_1, v_2\}$ и добавлением вершины w и рёбер $\{v_1, w\}$, $\{w, v_2\}$, $\{w, v_4\}$, является формой замкнутых маршрутов длиной $4n + 2$.

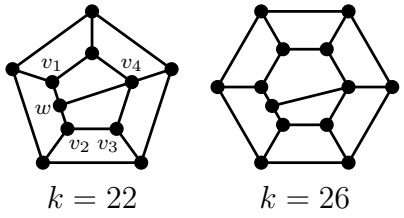


Рис. 5. Примеры «тяжёлых» форм без треугольников для случая $k \equiv 2 \pmod 4$

Доказательство. На рис. 6 показана структура одного из замкнутых маршрутов длиной $4n + 2$ в построенном графе. Начальные участки маршрутов длиной $n + 3$ соответствуют обходу «внутреннего» цикла (рис. 4) и здесь не приводятся. ■

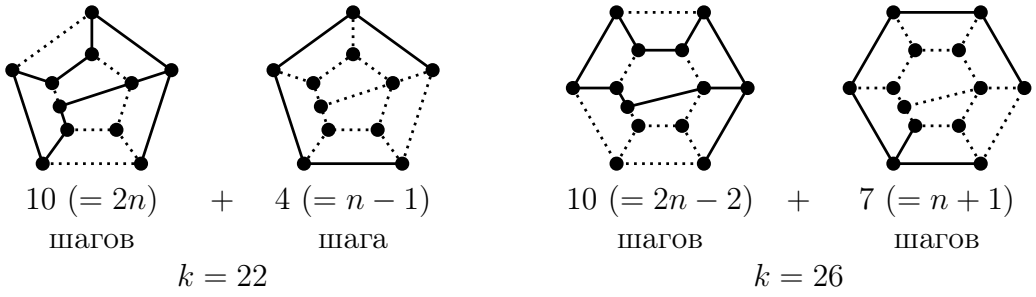


Рис. 6. Построение замкнутых маршрутов в «тяжёлых» формах без треугольников (случай $k \equiv 2 \pmod 4$)

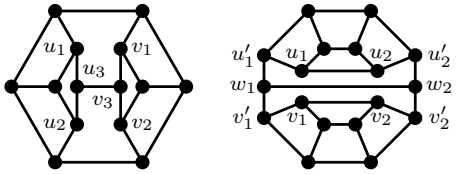
В утверждении 6 определён класс «тяжёлых» двудольных форм замкнутых маршрутов длиной $k \equiv 4 \pmod 8$, $k \geqslant 28$. Формы из этого класса планарны и являются регу-

лярными графами степени три. Благодаря последнему свойству, построенные графы также служат примерами «тяжёлых» форм со степенями вершин не более трёх.

Утверждение 6. Пусть $G = C_n \times P_2$, $n \geqslant 6$, $\{u_1, v_1\}$ и $\{u_2, v_2\}$ — два «параллельных» ребра «внутреннего» цикла графа G, а u'_1, v'_1, u'_2, v'_2 — вершины «внешнего» цикла, смежные с вершинами u_1, v_1, u_2, v_2 соответственно (как на рис. 7). Тогда граф, получаемый из G

1) удалением рёбер $\{u_1, v_1\}$, $\{u_2, v_2\}$ и добавлением вершин u_3, v_3 и рёбер $\{u_3, v_3\}$, $\{u_1, u_3\}$, $\{u_3, u_2\}$, $\{v_1, v_3\}$, $\{v_3, v_2\}$ в случае $n \equiv 2 \pmod 4$;
2) удалением рёбер $\{u_1, v_1\}$, $\{u_2, v_2\}$, $\{u'_1, v'_1\}$, $\{u'_2, v'_2\}$ и добавлением вершин w_1, w_2 и рёбер $\{w_1, w_2\}$, $\{u_1, u_2\}$, $\{v_1, v_2\}$, $\{u'_1, w_1\}$, $\{w_1, v'_1\}$, $\{u'_2, w_2\}$, $\{w_2, v'_2\}$ в случае $n \equiv 0 \pmod 4$,

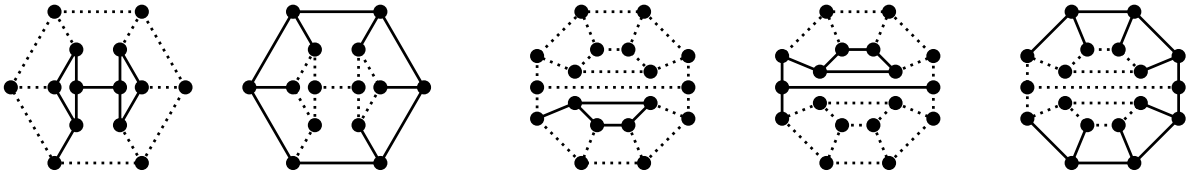
является формой замкнутых маршрутов длиной $4n + 4$.



$k = 28$ $\qquad$ $k = 36$

Рис. 7. Примеры «тяжёлых» двудольных форм для случая $k \equiv 4 \pmod 8$

Доказательство. В первом случае маршрут можно составить как для призматического графа (рис. 4). При этом на обход «внутренней» части затрачивается на четыре шага больше (рис. 8). В итоге длина всего маршрута равна $4n + 4$. Пример маршрута для второго случая указан на рис. 8. ■



12 $(= n + 6)$ шагов + 16 $(= 3n - 2)$ шагов

$k = 28$

6 $(= n/2 + 2)$ шагов + 10 $(= n/2 + 6)$ шагов + 20 $(= 3n - 4)$ шагов

$k = 36$

Рис. 8. Построение замкнутых маршрутов в «тяжёлых» двудольных формах (случай $k \equiv 4 \pmod 8$)

Обе формы, изображённые на рис. 7, выходят за рамки сгенерированных наборов форм [15] и были найдены с помощью программы `geng` из пакета `nauty` [16].

При увеличении $k \equiv 0, 2 \pmod 4$ на 1 значение оценки $[k/2]$ не изменяется, как и значение оценки $2[k/4]$ при увеличении $k \equiv 0, 4 \pmod 8$ на 2. Данные соотношения и утверждение 7 позволяют получить примеры «тяжёлых» форм для случаев $k \equiv 1, 3 \pmod 4$, без соблюдения двудольности, и $k \equiv 2, 6 \pmod 8$, с соблюдением двудольности, из перечисленных выше графов (призматических и их модификаций).

Утверждение 7.

1) Форма замкнутых маршрутов длиной k является также формой замкнутых маршрутов длиной $k + 2$.

2) Пусть граф G является формой замкнутых маршрутов длиной k, $\{u, v\}$ — ребро этого графа. Тогда граф, получаемый из G удалением ребра $\{u, v\}$ и добавлением вершины w и рёбер $\{u, w\}$, $\{w, v\}$ (другими словами, ребро $\{u, v\}$ «разбивается» на два ребра), есть форма замкнутых маршрутов длиной $k + 1$.

Доказательство. В первом случае маршрут можно «удлинить» на два шага, пройдя, например, по какому-либо ребру, инцидентному начальной вершине маршрута. Во втором случае новый маршрут получается заменой участка (u, v) или (v, u) в старом маршруте на участок (u, w, v) или (v, w, u) соответственно. ■

Утверждения 4–7 в совокупности обеспечивают примеры «тяжёлых» планарных форм без треугольников, начиная с $k = 16$, кроме 18 и 19; примеры «тяжёлых» планарных двудольных форм со степенями вершин не более трёх, начиная с $k = 16$, кроме 20 и 22; примеры «тяжёлых» планарных форм без треугольников со степенями вершин не более трёх, начиная с $k = 16$. Данные формы также являются «тяжёлыми» в более широких семействах форм. В первом случае — вплоть до множества всех форм, во втором случае — вплоть до множества всех двудольных форм, а в третьем случае — вплоть до множества всех форм со степенями вершин не более трёх.

Путём вычислительных экспериментов, выполненных в рамках данной работы, установлено, что при $k = 18$ нет «тяжёлых» форм, которые и планарны, и не содержат треугольников. Примеры планарной формы и формы без треугольников изображены на рис. 9. В случае $k = 19$ существует единственная «тяжёлая» форма, которая планарна и не содержит треугольников (рис. 9). Похожая ситуация сложилась со значениями $k = 20, 22$ для «тяжёлых» двудольных форм. При $k = 20$ существуют единственная планарная форма, но она содержит вершину степени четыре, и формы со степенями вершин не более трёх, но они непланарны (рис. 9). Если же $k = 22$, то среди «тяжёлых» двудольных форм можно отыскать планарные формы со степенями вершин не более трёх. Пример одной из них представлен на рис. 9.

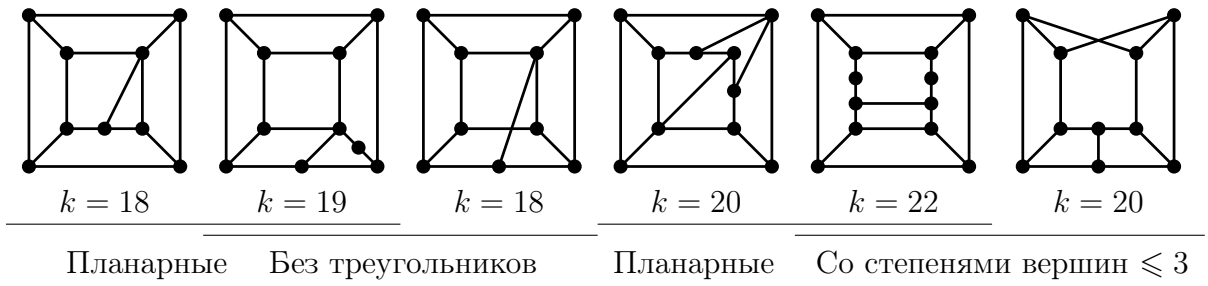


Рис. 9. Примеры «тяжёлых» форм ($k = 18, 19$) и «тяжёлых» двудольных форм ($k = 20, 22$) с разными свойствами

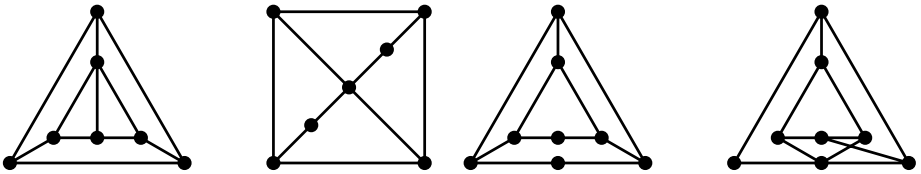
6. Примеры «тяжёлых» форм для значений $k < 16$

При небольших значениях k возникает ряд исключений, когда наибольшие кратности сумм оказываются меньше по сравнению с общими закономерностями. Все такие исключения отмечены в табл. 1 (Приложение А). Два из них уже рассмотрены в предыдущем пункте. Приложение Б содержит списки всех «тяжёлых» форм для значений $k = 8, 9, 10, 11$.

Рассмотрим оставшиеся случаи $k = 12, 13, 14, 15$. С учётом утверждения 7, для заданного набора свойств (строки табл. 1 и 2) и значения кратности достаточно указать форму маршрута наименьшей длины, на которой достигается эта кратность. Так, од-

ной из «тяжёлых» форм со степенями вершин не более трёх при $k = 12$ является призматический граф $C_3 \times P_2$ (см. рис. 3). Поскольку для следующих трёх значений длины (13, 14 и 15) достигается та же наибольшая кратность суммы, утверждение 7 позволяет «экстраполировать» данный пример на эти случаи.

Тот же граф $C_3 \times P_2$ является «тяжёлой» планарной формой для $k = 12$. В случае $k = 14$ достигается кратность на 1 больше. Соответствующий пример указан на рис. 10. «Тяжёлые» планарные формы автоматически являются «тяжёлыми» среди всех форм, так как приводят к суммам той же наибольшей кратности $[k/2]$.



$k = 14$ Планарная | $k = 12$ Планарные двудольные | $k = 14$ | $k = 15$ Без треугольников

Рис. 10. Примеры «тяжёлых» форм для разных наборов свойств

Среди форм без условия планарности кратность 6 обеспечивает, например, граф $K_{3,3}$. В некоторых случаях это единственная «тяжёлая» форма (табл. 2).

К пятикратной и шестикратной суммам ($k = 12, 14$, с условием планарности) приводят двудольные формы, изображённые на рис. 10. В случае $k = 12$ указанная форма является единственной «тяжёлой» как среди планарных двудольных форм, так и среди планарных форм без треугольников.

Наконец, при $k = 15$ семикратную сумму даёт единственная форма без треугольников (рис. 10).

7. Сложность вычисления значений сумм

Наибольшая кратность суммы является важным, но не абсолютным «показателем» вычислительной сложности формулы. Так, буквальное вычисление суммы

$$\alpha(K_4) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} a_{ij} a_{ik} a_{il} a_{jk} a_{jl} a_{kl} \tag{4}$$

действительно требует времени $\mathrm{O}(n^4)$. Принимая во внимание природу множителей, составляющих общий член в (4), данную сумму можно упростить:

$$\alpha(K_4) = \sum_{i=1}^{n} \sum_{j \in I_i} \sum_{k \in I_i} \sum_{l \in I_i} a_{jk} a_{jl} a_{kl}. \tag{5}$$

Символом I_i в выражении (5) обозначено множество номеров всех вершин, смежных с вершиной, имеющей номер i. Вычислительные затраты на создание списков смежности, причём однократное, пренебрежимо малы на фоне матричных операций.

В виде (5) сумма $\alpha(K_4)$ по-прежнему характеризуется сложностью $\mathrm{O}(n^4)$ применительно к случаю произвольных графов. Если же известно, что степени всех вершин ограничены значением d, то оценка становится $\mathrm{O}(d^3 \cdot n)$. При достаточно малом значении d может оказаться целесообразным аналогичное преобразование всех сумм (1), входящих в формулу для подсчёта циклов. Эффективность такого подхода затруднительно обсуждать без проведения вычислительных экспериментов.

Авторы [12] заметили, что внутренняя двойная сумма в (5) представляет собой элемент куба подматрицы A_i матрицы смежности, образованной пересечением строк и столбцов с номерами из множества I_i:

$$\alpha(K_4) = \sum_{i=1}^{n} \sum_{j=1}^{|I_i|} (A_i^3)_{jj}.$$

Вычисление суммы в таком виде имеет сложность $\mathrm{O}(n \cdot r(n))$, где $r(n)$ — сложность умножения $n \times n$-матриц. Однако данная оценка не сильно отличается от $\mathrm{O}(n^4)$ и, тем более, не ясно, насколько практически эффективен указанный приём (экспериментальные результаты в литературе не встречаются).

В случае планарных графов расчёт суммы $\alpha(K_4)$ можно заменить перечислением всех подграфов, изоморфных графу K_4, которое выполняется со сложностью $\mathrm{O}(n)$ [17]. Значение самой суммы в 24 раза больше найденного количества подграфов.

Два последних приёма (использование подматриц и учёт планарности) известны только для суммы $\alpha(K_4)$, самой простой из «тяжёлых». Представляет интерес их обобщение и экспериментальное исследование.

ПРИЛОЖЕНИЕ А

Наибольшие кратности сумм

В табл. 1 и 2 пустые клетки соответствуют трём случаям: наиболее трудоёмкой операцией является умножение матриц; графы двудольны, а длина цикла нечётна; множество форм не сгенерировано (по причине больших вычислительных затрат). Курсивом выделены исключения, когда наибольшие кратности сумм меньше значений, получаемых по формулам, указанным в последнем столбце.

Таблица 1

Наибольшая кратность суммы в формулах для подсчёта циклов длиной k

Типы графов	k													
	8	9	10	11	12	13	14	15	16	17	18	19	20	$\geqslant 10$
Все или планарные	4	4	5	5	6	6	7	7	8	8				$[k/2]$
Степ. $\leqslant 3$ (все или план.)	4	4	4	4	6	6	6	6	8	8				$2\,[k/4]$
Без треугольников			*4*	*4*	6	6	*6*	7	8	8	9	9		$[k/2]$
Без треуг., план.			*4*	*4*	*5*	*5*	*6*	*6*	8	8	*8*	9		$[k/2]$
Без треуг., степ. $\leqslant 3$			4	4	6	6	6	6	8	8	8	8		$2\,[k/4]$
Без треуг., план., степ. $\leqslant 3$			4	4	*4*	*4*	6	6	8	8	8	8		$2\,[k/4]$
Двуд. (все или ст. $\leqslant 3$)			4		6		6		8		8		10	$2\,[k/4]$
Двуд., план.			4		*5*		6		8		8		10	$2\,[k/4]$
Двуд., план., степ. $\leqslant 3$			4		*4*		6		8		8		*8*	$2\,[k/4]$

Таблица 2

Количество сумм наибольшей кратности в формулах для подсчёта циклов длиной k

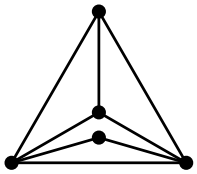
Типы графов	k												
	8	9	10	11	12	13	14	15	16	17	18	19	20
Все	1	3	2	6	6	27	14	119	63	712			
Планарные	1	3	1	5	4	19	7	66	28	316			
Со степенями $\leqslant 3$	1	2	5	9	2	4	16	40	5	18			
План., степ. $\leqslant 3$	1	2	5	9	1	3	11	30	3	11			
Без треугольников			*1*	*2*	1	1	*13*	1	5	17	6	71	
Без треуг., план.			*1*	*2*	*1*	*4*	*3*	*17*	1	1	*29*	1	
Без треуг., степ. $\leqslant 3$			1	2	1	1	7	18	2	5	32	126	
Без треуг., план., степ. $\leqslant 3$			1	2	*9*	*16*	2	8	1	1	12	51	
Двудольные			1		1		8		4		60		16
Двуд., план.			1		*1*		2		1		14		1
Двуд., степ. $\leqslant 3$			1		1		3		1		6		2
Двуд., план., степ. $\leqslant 3$			1		*5*		1		1		4		*37*

ПРИЛОЖЕНИЕ Б

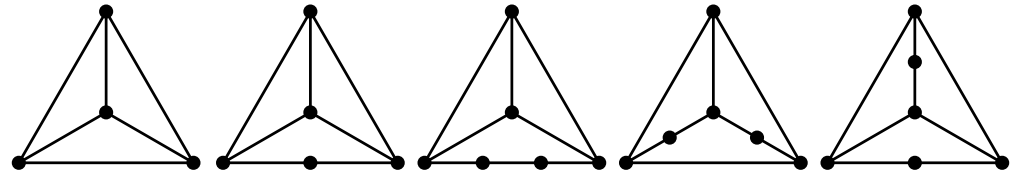
Списки «тяжёлых» форм для $k = 8, 9, 10, 11$

Суффикс «d3» указывает на множество форм со степенями вершин не более трёх.

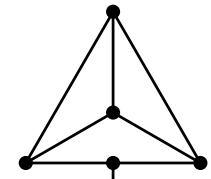
9 (все 3 формы)



10d3 (все 5 форм)

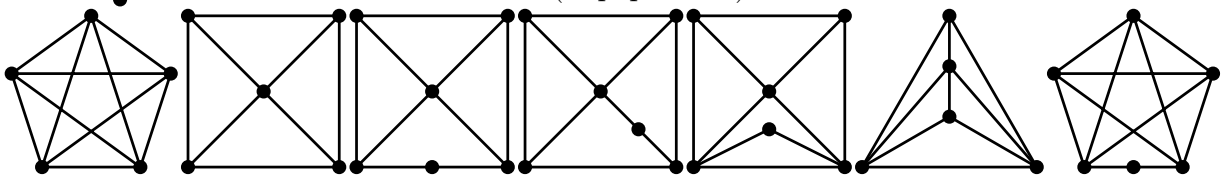


8 (ед. форма)



11d3 (4 формы из 9)

11d3 (5 форм из 9)



10 (все 2 формы)

11 (все 6 форм)

ЛИТЕРАТУРА

1. *Harary F. and Kommel H. J.* Matrix measures for transitivity and balance // J. Math. Sociol. 1979. V. 6. No. 2. P. 199–210.
2. *Newman M. E. J.* The structure and function of complex networks // SIAM Rev. 2003. V. 45. No. 2. P. 167–256.
3. *Cardillo A., Scellato S., Latora V., and Porta S.* Structural properties of planar graphs of urban street patterns // Phys. Rev. E. 2006. V. 73. No. 6. P. 066107(8).

4. *Halford T. R. and Chugg K. M.* An algorithm for counting short cycles in bipartite graphs // IEEE Trans. Inform. Theory. 2006. V. 52. No. 1. P. 287–292.
5. *Valiant L. G.* The complexity of enumeration and reliability problems // SIAM J. Comput. 1979. V. 8. No. 3. P. 410–421.
6. *Liśkiewicz M., Ogihara M., and Toda S.* The complexity of counting self-avoiding walks in subgraphs of two-dimensional grids and hypercubes // Theor. Comput. Sci. 2003. V. 304. No. 1–3. P. 129–156.
7. *Flum J. and Grohe M.* The parameterized complexity of counting problems // SIAM J. Comput. 2004. V. 33. No. 4. P. 892–922.
8. *Воропаев А. Н.* Вывод явных формул для подсчёта циклов фиксированной длины в неориентированных графах // Информационные процессы. 2011. Т. 11. № 1. С. 90–113.
9. *Harary F. and Manvel B.* On the number of cycles in a graph // Matematický časopis. 1971. V. 21. No. 1. P. 55–63.
10. *Chang Y. C. and Fu H. L.* The number of 6-cycles in a graph // The Bulletin of the Institute of Combinatorics and Its Applications. 2003. V. 39. P. 27–30.
11. *Perepechko S. N. and Voropaev A. N.* The number of fixed length cycles in an undirected graph. Explicit formulae in case of small lengths // Int. Conf. "Mathematical Modeling and Computational Physics" (MMCP'2009). Dubna: JINR, 2009. P. 148–149.
12. *Alon N., Yuster R., and Zwick U.* Finding and counting given length cycles // Algorithmica. 1997. V. 17. No. 3. P. 209–223.
13. *Воропаев А. Н.* Учёт обхвата при подсчёте коротких циклов в двудольных графах // Информационные процессы. 2011. Т. 11. № 2. С. 225–252.
14. *Харари Ф.* Теория графов. М.: Мир, 1973. 301 с.
15. `flowproblem.ru/cycles/explicit-formulae/ck-graphs` — Ck-graphs: FlowProblem. 2011.
16. `cs.anu.edu.au/~bdm/nauty` — The nauty page. 2011.
17. *Papadimitriou C. H. and Yannakakis M.* The clique problem for planar graphs // Inform. Processing Lett. 1981. V. 13. No. 4, 5. P. 131–133.

DOI 10.17223/20710410/14/7

УДК 519.17

КОЛИЧЕСТВЕННЫЕ ОЦЕНКИ НЕКОТОРЫХ СВЯЗНОСТНЫХ ХАРАКТЕРИСТИК ПРЕДФРАКТАЛЬНЫХ ГРАФОВ[1]

А. А. Кочкаров*, Л. И. Сенникова**

*Институт прикладной математики им. М. В. Келдыша РАН, г. Москва, Россия,
**Ставропольский институт управления, г. Ставрополь, Россия

E-mail: akochkar@gmail.com, s-ludhen@yandex.ru

Работа посвящена исследованию связностных характеристик предфрактальных графов. Получены достижимые оценки для числа точек сочленения и числа мостов предфрактального графа. Свойство самоподобия определяет получение прогнозируемых диапазонов количественных оценок для перечисленных связностных характеристик.

Ключевые слова: *самоподобные графы, фрактальные (предфрактальные) графы, сетевые системы, точки сочленения, мосты.*

Введение

Развитие глобальных сетей (информационных, социальных, технических) и накопление за последние десятилетия эмпирического материала спровоцировали новый виток изучения сложных многоэлементных сетевых систем [1, 2] и предопределили появление так называемой «сетевой науки» (Network Science) [1].

Если формализовать структуру сетевой системы в виде графа [3], то изменения, происходящие в ее структуре, могут быть описаны простейшими теоретико-графовыми операциями: стягивание ребра, удаление (добавление) ребра, удаление (добавление) вершины. Изменения структуры системы могут быть разовыми, а могут быть постоянными. Для второго случая принято использовать понятие *структурной динамики* [4]. Несомненно, для описании структурной динамики лучше всего подходит аппарат теории графов.

Одним из наиболее распространенных сценариев структурной динамики является *рост структуры*. Рост структуры — это регулярное появление новых элементов и связей в структуре системы. Рост структуры может происходить по строго сформулированным правилам, не исключая наличия в них фактора случайности.

Исследование структурной динамики как модели изменчивости связей многоэлементных сетевых систем представляется актуальной задачей.

В работе рассматривается одно из возможных правил, задающих структурную динамику сложных многоэлементных сетевых систем. Формальным представлением изменения структур сетевых систем по этому правилу являются масштабно-инвариантные, или самоподобные [5], графы большой размерности, называемые *фрактальными* (*предфрактальными*) [6]. Правила порождения предфрактального графа позволяют прогнозировать его качественные и количественные характеристики, а также оценивать изменение этих характеристик в процессе роста структуры сетевой системы. Доказанные в работе теоремы устанавливают зависимость характеристик всего

[1]Работа поддержана грантом РФФИ № 10-01-00786-а.

предфрактального графа от характеристик его самой меньшей несамоподобной части — затравки, что позволяет оценить диапазон изменения важных характеристик, относящихся к структурной стойкости сетевых систем [7].

1. Фрактальные и предфрактальные графы

Фрактальные графы [8] используются для моделирования структур, растущих по одним и тем же правилам независимо от точки роста. Не исключается множественный одновременный рост во всей структуре системы. Формальным отражением этих правил является операция *замены вершины затравкой* (ЗВЗ) [8], она же лежит в основе определения фрактальных графов.

Термином «*затравка*» условимся называть какой-либо связный граф $H = (W, Q)$. Суть операции ЗВЗ заключается в следующем. В данном графе $G = (V, E)$ у намеченной для замещения вершины $\tilde{v} \in V$ выделяется множество $\widetilde{V} = \{\tilde{v}_j : j = 1, 2, \ldots, |\widetilde{V}|\}$ смежных ей вершин. Далее из графа G удаляется вершина $\tilde{v}$ и все инцидентные ей ребра. Затем каждая вершина $\tilde{v}_j \in \widetilde{V}$, $j = 1, 2, \ldots, |\widetilde{V}|$, соединяется ребром с одной из вершин затравки $H = (W, Q)$. Вершины соединяются произвольно (случайным образом) или по определенному правилу при необходимости.

Предфрактальный граф будем обозначать через $G_L = (V_L, E_L)$, где V_L — множество вершин графа; E_L — множество его ребер. Определим его рекуррентно, поэтапно, заменяя каждый раз в построенном на предыдущем этапе $l = 1, 2, \ldots, L-1$ графе $G_l = (V_l, E_l)$ каждую его вершину затравкой $H = (W, Q)$. На этапе $l = 1$ предфрактальному графу соответствует затравка $G_1 = H$. Об описанном процессе говорят, что *предфрактальный граф G_L порожден затравкой H*. Процесс порождения предфрактального графа G_L по существу есть процесс построения последовательности предфрактальных графов $G_1, G_2, \ldots, G_l, \ldots, G_L$, называемой *траекторией*. Фрактальный граф $G = (V, E)$, порожденный затравкой H, определяется бесконечной траекторией. Ранг L фактически определяет «возраст» (число этапов порождения) и размер (число вершин) предфрактального графа.

Для предфрактального графа G_L ребра, появившиеся на l-м, $l \in \{1, 2, \ldots, L\}$, этапе порождения, будем называть *ребрами ранга l*. *Новыми ребрами* предфрактального графа G_L назовем ребра ранга L, а все остальные ребра назовем *старыми*.

Если из предфрактального графа G_L, порожденного n-вершинной затравкой H, последовательно удалить все старые ребра (ребра ранга l, $l = 1, 2, \ldots, L-1$), то исходный граф распадется на множество связных компонент $\{B_L^{(1)}\}$, каждая из которых изоморфна затравке H. Компоненты $B_L^{(1)}$ будем называть *блоками первого ранга*. Аналогично при удалении из предфрактального графа G_L всех старых ребер рангов $l = 1, 2, \ldots, L-2$ получим множество блоков $\{B_L^{(2)}\}$ *второго ранга*. Обобщая, скажем, что при удалении из предфрактального графа G_L всех ребер рангов $l = 1, 2, \ldots, L-r$ получим множество $\{B_{L,i}^{(r)}\}$, $r \in \{1, 2, \ldots, L-1\}$, блоков *$r$-го ранга*, где $i = 1, 2, \ldots, n^{L-r}$ — порядковый номер блока. Блоки $B_L^{(1)}$ первого ранга будем называть также *подграф-затравками H* предфрактального графа G_L.Очевидно, что всякий блок $B_L^{(r)} = \left(U_L^{(r)}, M_L^{(r)}\right)$, $r \in \{1, 2, \ldots, L-1\}$, является предфрактальным графом $B_r = (U_r, M_r)$, порожденным затравкой H.

Обобщением описанного процесса порождения предфрактального графа G_L является случай, когда вместо единственной затравки H используется множество затравок $\mathscr{H} = \{H_1, H_2, \ldots, H_T\}$, $T \geqslant 2$. Суть этого обобщения состоит в том, что при переходе от графа G_{l-1} к графу G_l каждая вершина замещается некоторой затравкой $H_t \in \mathscr{H}$,

которая выбирается случайно или согласно определенному правилу, отражающему специфику моделируемого процесса или структуры.

Термином *подграф-затравка* $z_s^{(l)}$ будем называть блок $B_{l,s}^{(1)}$, $s = 1, \ldots, n^{l-1}$, первого ранга предфрактального графа G_l, $l = 1, \ldots, L$, из траектории. Подграф-затравки $z_s^{(l)}$ графов $G_1, G_2, \ldots, G_L$ из траектории предфрактального графа G_L объединим в множество $Z(G_L) = \{z_s^{(l)} : l = 1, \ldots, L, s = 1, \ldots, n^{l-1}\}$. В траектории переход от графа G_{l-1} к G_l осуществляется $|V_{l-1}| = n^{l-1}$ операциями ЗВЗ, поэтому общее число использованных затравок в порождении предфрактального графа G_L равно $1 + n + n^2 + ... + n^{L-1} = (n^L - 1)/(n - 1)$. Тогда мощность множества всех подграф-затравок из траектории графа G_L также равна $|Z(G_L)| = (n^L - 1)/(n - 1)$.

На рис. 1–3 показана траектория G_1, G_2, G_3 предфрактального графа $G_3 = (V_3, E_3)$, порожденного затравкой $H = (W, Q)$ — полным 4-вершинным графом (рис. 1). Самыми «жирными» линиями на представленных рисунках изображены ребра подграф-затравки $z_1^{(1)}$. Линиями средней «жирности» (рис. 2) нарисованы ребра подграф-затравок $z_1^{(2)}$, $z_2^{(2)}$, $z_3^{(2)}$ и $z_4^{(2)}$. И наконец, тонкими линиями (рис. 3) нарисованы новые ребра предфрактального графа G_3, которые образуют подграф-затравки $z_s^{(3)}$, $s = 1, \ldots, 16$.

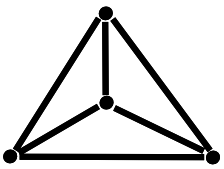


Рис. 1. $H = (W, Q)$

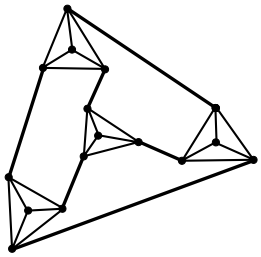


Рис. 2. $G_2 = (V_2, E_2)$

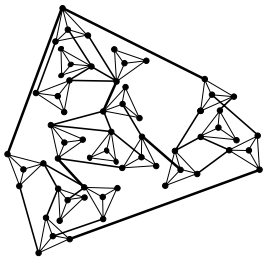


Рис. 3. $G_3 = (V_3, E_3)$

2. Оценка числа точек сочленения предфрактального графа

Число точек сочленения графа $H = (W, Q)$ обозначим через $m(H)$.

Теорема 1. Для всякого предфрактального графа G_L, порожденного n-вершинной затравкой H с сохранением смежности старых ребер одного ранга, справедливы верхняя и нижняя оценки числа точек сочленения

$$m(H)n^{L-1} \leqslant m(G_L) \leqslant m(H)n^{L-1} + |Z(G_L)|,$$

где $Z(G_L)$ — множество всех подграф-затравок предфрактального графа G_L.

Доказательство. Рассмотрим траекторию предфрактального графа G_L, порожденного затравкой H, имеющей $m(H)$ точек сочленения. Точкой пересечения старых ребер одного ранга будем называть вершину, в которой сохраняется их смежность. На втором этапе порождения все точки пересечения старых ребер (первого ранга) могут совпасть с точками сочленения затравок и в этом случае $m(G_2) = m(H)n$, где n — число затравок графа G_2. При выполнении условий теоремы меньше чем $m(H)n$ точек сочленения быть не может. В случае же несовпадения всех точек пересечения старых ребер с точками сочленения затравок число точек сочленения графа G_2 определяется равенством $m(G_2) = m(H)n + n$, поскольку каждая точка пересечения старых ребер является точкой сочленения графа G_2. При произвольном размещении смежных старых ребер число точек сочленения графа G_2 ограничивается неравенствами $m(H)n \leqslant m(G_2) \leqslant m(H)n + n$.

Продолжая рассуждения аналогичным образом, на l-м этапе, $l = 3, \ldots, L$, порождения в траектории графа G_L получим, что число точек сочленения графа G_l равно $m(G_l) = m(H)n^{l-1}$ при совпадении точек пересечения старых ребер с точками сочленения затравок. В противном случае, если никакая точка пересечения старых ребер одного ранга не совпадает ни с одной из точек сочленения затравок, а значит, каждая из точек пересечения старых ребер дает по одной точке сочленения для графа G_l, то достигается верхняя оценка, которая равна $m(G_l) = m(H)n^{l-1} + (n^l - n)/(n-1)$.

При произвольной инцидентности старых ребер с точками сочленений затравок или другими точками сочленения, полученными в результате смежности старых ребер одного ранга, число точек сочленения графа G_l оценивается двойным неравенством $m(H)n^{l-1} \leqslant m(G_l) \leqslant m(H)n^{l-1} + |Z(G_l)|$, $l = 1, \ldots, L$. ■

3. Оценка числа мостов предфрактального графа

Число мостов графа $H = (W, Q)$ обозначим через $k(H)$.

Теорема 2. Для всякого предфрактального графа G_L, порожденного затравкой H, справедливы верхняя и нижняя оценки числа мостов

$$k(H) \leqslant k(G_L) \leqslant k(H)|Z(G_L)|,$$

где $Z(G_L)$ — множество всех подграф-заставок предфрактального графа G_L.

Доказательство. Рассмотрим траекторию предфрактального графа G_L, порожденного затравкой $H = (W, Q)$. На затравке $H = (W, Q)$ выделим мост $e = \{v_1, v_2\} \in Q$, удаление которого приводит к разделению затравки на две компоненты. На втором этапе порождения предфрактального графа G_L, после замещения всех вершин затравки, выделенное ребро (уже старое ребро 1-го ранга) $e = \{v_1', v_2'\} \in E_2$ случайным образом соединит вершины двух подграф-затравок предфрактального графа $G_2 = (V_2, E_2)$. Но удаление ребра e приведет к разделению графа G_2 на компоненты, поскольку мост $e \in E_2$ — единственная цепь, соединяющая концы ребра e, независимо от того, какие вершины ($v_1 \in W$ и $v_2 \in W$ или же $v_1' \in V_2$ и $v_2' \in V_2$) она соединяет. Из этих рассуждений вытекает, что все мосты затравки H остаются мостами на всей траектории графа G_L, поэтому число мостов предфрактального графа G_L не меньше числа мостов затравки H: $k(G_L) \geqslant k(H)$.

Рассмотрим произвольный предфрактальный граф $G_l^* = (V_l^*, E_l^*)$, $l = 2, \ldots, L$, выделим на нем подграф-затравку $H = (W, Q)$, которая, в отдельном от графа виде, имеет $k(H)$ мостов. Возникает вопрос, останутся ли мостами для графа G_l^* ребра, являющиеся мостами в отдельно взятой затравке H? Пусть при удалении моста $e^* = \{v_1^*, v_2^*\} \in Q$ затравка H распадается на две компоненты $H' = (W', Q')$ и $H'' = (W'', Q'')$. Значит, для того чтобы ребро $e^* \in Q$ перестало быть мостом в графе G_l^*, достаточно, чтобы подграфам H'и H'' были инцидентны хотя бы по одному старому ребру $(l-1)$-го ранга, тем самым нейтрализуя мост e^* и сохраняя связность графа G_l^* при удалении ребра e^*.

Если старые ребра $(l-1)$-го ранга сохраняют смежность, то ребро e^* будет мостом и для всего графа G_l^*, поскольку при его удалении связность предфрактального графа G_l^* нарушается.

Теперь ясно, что число мостов $k(G_L)$ предфрактального графа G_L, порожденного затравкой H, зависит от того, как часто сохраняется смежность старых ребер графа G_L. Если старые ребра $(l-1)$-го ранга предфрактального графа G_l, $l = 2, \ldots, L$, из траектории предфрактального графа G_L подходят к подграф-затравкам таким образом, что каждый из $k(H)$ мостов образует с какими-либо двумя из старых ребер одного

ранга простую цепь, то мосты отдельно взятой затравки не будут мостами графа G_l, так как при их удалении из подграф-затравки H компоненты, на которые должна была бы распасться отдельная затравка, будут инцидентными старым ребрам, сохраняя этим связность самого графа G_l. В случае выполнения этого правила для всех графов G_l, $l = 2, \ldots, L$, из траектории предфрактального графа G_L получим, что все мосты затравок, появляющихся в процессе порождения графа G_L, нейтрализуются старыми ребрами, кроме тех $k(H)$ мостов, которые существовали изначально в графе $G_1 = H$, поскольку для этих мостов не существует старых ребер меньшего ранга. Отсюда следует, что нижняя граница числа мостов графа G_L определяется неравенством $k(G_L) \geqslant k(H)$.

Для нахождения верхней границы достаточно рассмотреть случай, когда в предфрактальном графе G_L сохраняется смежность старых ребер любого ранга. Действительно, при сохранении смежности старых ребер $(l-1)$-го ранга предфрактального графа G_l, $l = 2, \ldots, L$, все старые ребра, подходящие к затравке H, будут смежны одной из компонент, полученных при удалении одного из мостов. Следовательно, все $k(H)$ мостов затравки H останутся мостами и в графе G_l. Учитывая, что в G_l число затравок равно n^{l-1} (n — число вершин затравки H), замечаем, что число его мостов по сравнению с графом G_{l-1} увеличится на $n^{l-1}k(H)$:

$$k(G_1) = k(H), \quad k(G_2) = k(H) + nk(H), \quad \ldots, \quad k(G_l) = k(H)(n^l - 1)/(n-1).$$

При произвольном построении фрактального графа G_L число его мостов ограничивается неравенствами $k(H) \leqslant k(G_L) \leqslant k(H)|Z(G_L)|$. ■

Заключение

Для проведения анализа работоспособности всякой сетевой системы, имеющей сложную изменяющуюся структуру, необходимо моделировать динамику самих изменений в структуре. Это позволяет анализировать изменения важных для сетевой системы связностных характеристик, к которым относятся число точек сочленения и число мостов. В качестве инструментария моделирования структурной динамики можно рассмотреть многие подходы. Процесс порождения предфрактальных графов, несомненно, является сильно ограниченным с точки зрения описания всех возможных сценариев роста структуры сетевой системы, но обладает возможностью прогнозирования изменений характеристик. Основная цель настоящей работы — продемонстрировать возможность получения прогнозируемых диапазонов количественных оценок для различных характеристик предфрактальных графов.

ЛИТЕРАТУРА

1. *Евин И. А.* Введение в теорию сложных сетей // Компьютерные исследования и моделирование. 2010. Т. 2. № 2. С. 121–141.
2. *Newman M. E. J.* Networks: an introduction. New York: Oxford University Press, 2010.
3. *Емеличев В. А., Мельников О. И., Сарванов В. И., Тышкевич Р. И.* Лекции по теории графов. М.: Наука, 1990.
4. *Охтилев М. Ю., Соколов Б. В., Юсупов Р. М.* Интеллектуальные технологии мониторинга и управления структурной динамикой сложных технических объектов. М.: Наука, 2006.
5. *Ахромеева Т. С., Курдюмов С. П., Малинецкий Г. Г., Самарский А. А.* Нестационарные структуры и диффузионный хаос. М.: Наука, 1992.
6. *Мелроуз Дж.* Иерархические фрактальные графы и блуждания на них // Фракталы в физике. М.: Мир, 1988. С. 519–523.

7. *Кочкаров А. А., Малинецкий Г. Г.* Моделирование распространения внешних воздействий по структуре сложной системы // Матем. моделирование. 2006. Т. 18. № 2. С. 51–60.
8. *Кочкаров А. А., Кочкаров Р. А.* Параллельный алгоритм поиска кратчайшего пути на предфрактальном графе // Журн. вычисл. матем. и матем. физики. 2004. Т. 44. № 6. С. 1157–1162.

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ДИСКРЕТНЫХ АВТОМАТОВ

DOI 10.17223/20710410/14/8

УДК 519.7

КОДИРОВАНИЕ СОСТОЯНИЙ ДИСКРЕТНОГО АВТОМАТА, ОРИЕНТИРОВАННОЕ НА УМЕНЬШЕНИЕ ЭНЕРГОПОТРЕБЛЕНИЯ РЕАЛИЗУЮЩЕЙ СХЕМЫ

Ю. В. Поттосин

Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь

E-mail: pott@newman.bas-net.by

Рассматривается задача кодирования состояний дискретного автомата с целью уменьшения интенсивности переключений элементов памяти в реализующей схеме. Предлагается метод решения этой задачи, сочетающий идеи методов «желательных соседств» и «сборки» булева гиперкуба.

Ключевые слова: *дискретный автомат, кодирование состояний, энергосбережение.*

Введение

В последнее время при проектировании дискретных устройств управления на основе сверхбольших интегральных схем большое внимание уделяется проблеме снижения энергопотребления проектируемой схемы. Это обусловлено стремлением, с одной стороны, увеличить время действия источника энергии в портативных приборах и, с другой стороны, снизить остроту проблемы отвода тепла при проектировании сверхбольших интегральных схем. Поэтому одним из основных критериев оптимизации при проектировании дискретных устройств является величина потребляемой схемой энергии. Как отмечено в работах [1, 2], потребляемая мощность схемы, построенной на основе КМОП-технологии, пропорциональна частоте изменения сигналов. Это дает возможность частично решать данную проблему на уровне логического проектирования. В частности, снижения энергопотребления можно добиваться при кодировании состояний и декомпозиции автомата [3 – 5]. Кодировать состояния при этом надо таким образом, чтобы при переходе автомата из одного состояния в другое меняли свое состояние как можно меньше элементов памяти.

При решении задачи кодирования состояний автомата в работе [3] использован информационный подход, при котором учитываются вероятности состояний. В работе [5] эта задача сводится к укладке графа поведения автомата в полный булев граф, или булев гиперкуб. Наибольшая эффективность метода из работы [5] достигается, когда граф, полученный из графа поведения автомата удалением ориентации и кратности ребер, оказывается подграфом полного булева графа, т. е. он полностью укладывается в гиперкуб. Тогда при каждом переходе автомата только один элемент памяти изменяет свое состояние. В противном случае минимизируется число ребер, не укладываемых в гиперкуб. Алгоритмизация этого метода осуществлена в работе [6], где описан еще один алгоритм кодирования состояний, который основан на использовании матрицы

смежности графа и карты Карно и заключается в построении последовательности конфигураций из ребер и квадратов, образующих подграфы гиперкуба. Под квадратом понимается цикл длины 4 в графе.

Для решения задачи энергосберегающего кодирования состояний автомата можно использовать метод «желательных соседств» [7, 8], направленный на получение наиболее простой схемы. При этом минимизируется величина $W = \Sigma w_{ij} d_{ij}$, где суммирование ведется по всем парам состояний заданного автомата, а d_{ij} — расстояние по Хэммингу между кодами состояний q_i и q_j. В случае энергосберегающего кодирования в качестве w_{ij} надо выбрать некоторый показатель, связанный с переходами между состояниями q_i и q_j. Как отмечено еще в работе [7], абсолютный минимум величины W получить довольно сложно, да и не всегда он соответствует оптимальному решению конкретной задачи. Поэтому чаще используются некоторые эвристики при размещении состояний автомата в булевом пространстве. В частности, для решения данной задачи можно использовать предлагаемый в работе [9] алгоритм размещения вершин графа, которым может быть гиперкуб, в простой цепи с минимизацией той же суммы W, только в ней w_{ij} — некоторая функция на ребрах графа, а d_{ij} — расстояния в заданной простой цепи.

В настоящей работе предлагается метод кодирования состояний автомата, приводящий к снижению переключательной активности элементов памяти в реализующей схеме. Он отличается от известных методов способом размещения состояний автомата в булевом пространстве внутренних переменных. Используется подход, изложенный в работах [10, 11] и основанный на методе из [7, 8]; размещение состояний в булевом пространстве представляется как процесс построения k-мерного булева гиперкуба, напоминающий сборку некоторой простой механической конструкции. Здесь k — число элементов памяти в проектируемой схеме.

1. Построение булева гиперкуба

Формально задачу можно поставить следующим образом. Пусть задан автомат M, состояния которого, образующие множество $Q = \{q_1, q_2, \ldots, q_\gamma\}$, должны быть закодированы булевыми векторами так, чтобы минимизировать сумму $W = \Sigma w_{ij} d_{ij}$, где суммирование ведется по всем парам состояний заданного автомата; d_{ij} — расстояние по Хэммингу между кодами состояний q_i и q_j; $w_{ij} = w(q_i, q_j)$; w — функция, принимающая вещественные значения на множестве пар состояний из Q. Смысл функции w и ее определение зависят от конкретной задачи. В работах [10, 11] она выбрана из соображений получения наиболее простой схемы автомата. В случае энергосберегающего кодирования функция w должна отражать интенсивность переключений элементов памяти. Так же, как в [6], в качестве w возьмем целочисленную функцию, значения которой для пар q_i, q_j пропорциональны вероятностям перехода между состояниями q_i и q_j, неважно, в каком направлении. Эту вероятность можно вычислить методом Чэпмена — Колмогорова [12], который подробно описан также в [6].

Критерием качества размещения состояний в вершинах булева гиперкуба можно считать введенную в [6] величину, названную дефектом отображения (ребер графа переходов на ребра гиперкуба), которая в терминах излагаемого метода вычисляется по формуле $D = \Sigma w_{ij}(d_{ij} - 1)$, где суммирование берется по всем парам состояний, соответствующим парам вершин в гиперкубе. Очевидно, чем меньше значение D, тем лучше результат размещения, и дефект отображения равен нулю, если всем парам состояний, связанным переходами, соответствуют ребра гиперкуба. Тогда при любом переходе из состояния в состояние переключается ровно один элемент памяти.

Предлагаемый метод является эвристическим, т. е. не гарантирует получения абсолютного минимума суммы W. Он основан на том соображении, что для того, чтобы величина W не сильно отличалась от минимума, надо, чтобы состояния q_i и q_j, для которых значение w_{ij} велико, кодировались по возможности близкими кодами.

Исходными данными для построения n-мерного гиперкуба являются значения функции w и число состояний γ заданного автомата M. Полагается, что это число минимальное или по какой-то причине его не надо минимизировать. Если γ не является целой степенью двойки, его надо путем введения фиктивных состояний увеличить до 2^n, где $n = \lceil \log_2 \gamma \rceil$. Считаем, что $w_{kl} = 0$, если хотя бы одно из состояний q_k или q_l является фиктивным. В начале процесса вершины, которые представляют состояния автомата M и должны быть вершинами гиперкуба, образуют пустой граф (без ребер).

Построение n-мерного гиперкуба представляется как последовательность n шагов. На s-м шаге рассматривается множество $(s-1)$-мерных гиперкубов, они объединяются в пары, и из каждой пары получается один s-мерный гиперкуб путем соответствующего добавления ребер. При этом, по возможности, для соединения ребрами выбираются те вершины q_i и q_j, которым соответствуют наибольшие из оставшихся значения w_{ij}. В результате выполнения n шагов получим искомый n-мерный гиперкуб. Вершинам приписываем n-компонентные булевы векторы с соблюдением отношения соседства, представленного ребрами гиперкуба.

На первом шаге из γ изолированных вершин, или 0-мерных гиперкубов, строятся 1-мерные гиперкубы в виде $\gamma/2$ попарно несмежных ребер. На последнем n-м шаге из двух $(n-1)$-мерных гиперкубов собирается один n-мерный гиперкуб путем добавления 2^{n-1} ребер.

Рассмотрим более подробно образование s-мерных гиперкубов на s-м шаге. Для компьютерной реализации метода важным моментом является способ представления гиперкубов. Любой k-мерный гиперкуб, являющийся подграфом n-мерного гиперкуба, можно представить последовательностью S из 2^k номеров вершин, взятых из множества $\{1, 2, ..., 2^n\}$. Ребра заданы неявно: считается, что две вершины связаны ребром тогда и только тогда, когда занимаемые ими места в последовательности S соответствуют местам соседних кодов в равной по длине последовательности кодов Грея.

Перед выполнением любого шага число гиперкубов всегда четно, точнее, для s-го шага оно равно 2^{n-s+1}. Текущая ситуация характерна наличием некоторого множества s-мерных гиперкубов C_s (перед выполнением шага оно пусто) и некоторого остатка в виде множества $(s-1)$-мерных гиперкубов C_{s-1}. Перебираются все пары гиперкубов из множества C_{s-1} и выбирается одна из них в соответствии с критерием, указанным ниже. К этой паре добавляются ребра, чтобы образовать s-мерный гиперкуб, который вводится в множество C_s. Выбранная пара исключается из C_{s-1}. Выполнение шага заканчивается, когда C_{s-1} оказывается пустым.

2. Объединение гиперкубов в пары

Для двух $(s-1)$-мерных гиперкубов, представленных последовательностями S' и S'', подсчитывается сумма Σw_{ij}, где суммирование ведется по всем парам i и j номеров вершин, занимающих одноименные места в S' и S''. Величина, представляемая указанной суммой, меняется с перестановкой вершин в одной из последовательностей, например в S''. Конечно, только те последовательности могут быть приняты к рассмотрению, которые сохраняют отношение соседства между вершинами.

Выбирается та пара гиперкубов, для которой величина Σw_{ij} максимальна, и из них строится s-мерный гиперкуб соединением ребрами вершин, номера которых за-

нимают одноименные места в последовательностях S' и S'' (после соответствующей перестановки). Последовательность, представляющая построенный гиперкуб, получается соединением последовательностей S' и S'', одна из которых меняет свой порядок на обратный.

Пусть, например, для восьми состояний получены два 2-мерных гиперкуба, которые представляются следующими последовательностями, связанными с кодом Грея: q_1, q_2, q_5, q_3 и q_4, q_6, q_8, q_7. Первый гиперкуб имеет ребра q_1q_2, q_2q_5, q_3q_5 и q_1q_3; второй — q_4q_6, q_6q_8, q_7q_8 и q_4q_7. При таком расположении вершин построенный с помощью добавления ребер q_1q_4, q_2q_6, q_5q_8 и q_3q_7 3-мерный гиперкуб представится последовательностью q_1, q_2, q_5, q_3, q_7, q_8, q_6, q_4. Если взять перестановку q_8, q_6, q_4, q_7 состояний второй последовательности, то получим q_1, q_2, q_5, q_3, q_7, q_4, q_6, q_8.

3. Сложность перебора вариантов объединения гиперкубов в пары

Рассмотрим перестановку, задаваемую оператором E_k, в результате действия которого все соседние по k-му измерению вершины меняются местами. Очевидные свойства операторов такого типа, $E_iE_k = E_kE_i$ и $E_iE_i = 1$, позволяют в каждом варианте пары гиперкубов перебирать еще 2^{s-1} различных вариантов, порождаемых перестановками вершин в одном из гиперкубов рассматриваемой пары, получаемыми путем применения всевозможных сочетаний операторов E_1, E_2, ... , E_{s-1}.

Количество перебираемых вариантов пар гиперкубов на s-м шаге выражается следующей формулой:

$$L'_s = \sum_{i=1}^{2^{n-s}} (2^{n-s} - i + 1)(2^{n-s+1} - 2i + 1) = 2^{n-s-1}((2^{2(n-s+1)} - 1)/3 + 2^{n-s}).$$

Принимая во внимание перестановки, определяемые всевозможными сочетаниями операторов E_k $(k = 1, 2, ..., s-1)$ для каждой пары $(s-1)$-мерных гиперкубов, получим количество перебираемых вариантов на s-м шаге:

$$L_s = 2^{s-1}L'_s = 2^{n-2}((2^{2(n-s+1)} - 1)/3 + 2^{n-s}).$$

Полный перебор всех возможных перестановок вершин $(s-1)$-мерного гиперкуба с сохранением отношения соседства на s-м шаге, предусмотренный в работе [10] и допустимый при небольшом числе состояний автомата, резко увеличивает трудоемкость получения решения при увеличении числа состояний (число вариантов при этом составит $L'_s(s-1)! \cdot 2^{s-1}$). Поэтому данный случай рассматривать здесь не будем.

Окончательно получим число вариантов, перебираемых указанным способом в течение всего процесса построения n-мерного булева гиперкуба:

$$L = \sum_{s=1}^{n} L_s = 2^{n-2}((2^{2(n+1)} - 3n - 13)/9 + 2^n).$$

Поскольку 2^n в данном случае является числом состояний автомата, количество вариантов перебора при сборке n-мерного гиперкуба оценивается как полином степени 3 от числа состояний. Процесс выбора пары $(s-1)$-мерных гиперкубов на s-м шаге может быть сокращен, если величина Σw_{ij} достигла верхней границы, которую легко подсчитать, взяв сумму наибольших значений w_{ij} для пар вершин, еще не связанных ребрами.

4. Примеры

Процесс сборки гиперкуба продемонстрируем на примере из [6]. Воспользуемся результатами вычислений функции w_{ij}, приведенными в данной работе, и представим значения w_{ij} в табл. 1 (значение w_{ij} пропорционально вероятности перехода между состояниями q_i и q_j), где строки и столбцы соответствуют состояниям автомата. К автомату с шестью состояниями добавляем два фиктивных состояния q_7 и q_8, чтобы придать числу состояний значение целой степени двойки. Нулевые значения w_{i7} и w_{i8} в табл. 1 не представлены.

Таблица 1

Значения w_{ij}

q_2	q_3	q_4	q_5	q_6	
18	11	0	7	1	q_1
	0	2	0	7	q_2
		7	15	2	q_3
			8	8	q_4
				0	q_5

Максимальное значение имеет $w_{12} = 18$. Поэтому в первую очередь соединяем ребром вершины q_1 и q_2. Затем ребрами соединяются вершины q_3 и q_5, q_4 и q_6, q_7 и q_8. Таким образом, на первом шаге получаем четыре одномерных гиперкуба, изображенных на рис. 1.

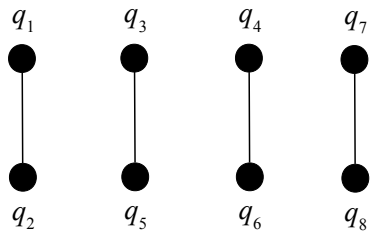


Рис. 1. Одномерные гиперкубы

Теперь требуется сформировать двумерный гиперкуб, добавив два ребра q_iq_j и q_kq_l так, чтобы величина $w_{ij} + w_{kl}$ была максимальной среди всех возможно добавляемых ребер. Все варианты присоединения ребер с соответствующей величиной $w_{ij} + w_{kl}$ приведены в табл. 2. Ребра, инцидентные вершинам q_7 и q_8, не рассматриваются.

Таблица 2

Варианты присоединения ребер на втором шаге

Ребра	q_1q_3, q_2q_5	q_1q_5, q_2q_3	q_1q_4, q_2q_6	q_1q_6, q_2q_4	q_3q_4, q_5q_6	q_3q_6, q_4q_5
$w_{ij} + w_{kl}$	11	7	7	3	7	10

Выбрав первый вариант, получим, как показано на рис. 2, один двумерный гиперкуб на вершинах q_1, q_2, q_3 и q_5 и второй гиперкуб путем добавления оставшихся ребер.

Для окончания работы требуется добавить еще четыре ребра, чтобы получить трехмерный гиперкуб. Варианты добавления ребер представлены в табл. 3. Среди них только два ребра могут иметь ненулевое значение w_{ij}. Поэтому рассматриваются только пары ребер.

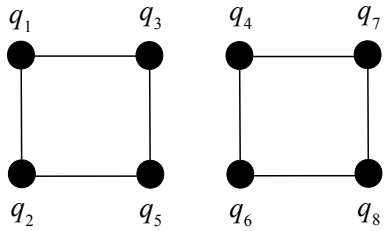


Рис. 2. Результат второго шага сборки гиперкуба

Таблица 3

Варианты присоединения ребер на третьем шаге

Ребра	q_1q_6,q_2q_4	q_1q_4,q_2q_6	q_2q_6,q_4q_5	q_2q_4,q_5q_6	q_3q_4,q_5q_6	q_3q_6,q_4q_5	q_1q_6,q_2q_4	q_1q_4,q_3q_6
$w_{ij}+w_{kl}$	3	7	15	2	7	10	8	2

Выбрав третий вариант, получим результат, показанный на рис. 3. Согласно отношению соседства, представленному графом на рис. 3, шесть состояний заданного автомата могут быть закодированы следующим образом: $q_1 - 000$, $q_2 - 001$, $q_3 - 010$, $q_4 - 111$, $q_5 - 011$, $q_6 - 110$.

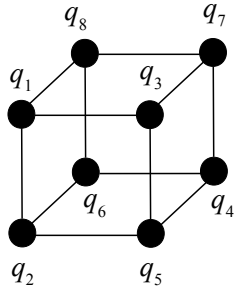


Рис. 3. Окончательный результат сборки

Получим дефект отображения $D = \Sigma w_{ij}(d_{ij} - 1)$. В рассмотренном примере к парам состояний, связанным переходами и соответствующим парам несвязных вершин в гиперкубе, относятся (q_1, q_5), (q_1, q_6), (q_2, q_4), (q_3, q_6) и (q_3, q_4). Соответствующими весами являются 7, 1, 2, 2 и 7, а расстояниями — 2, 2, 2, 3 и 2. В итоге получаем $D = 21$, что значительно меньше значения, полученного для того же примера в работе [6].

Для сравнения предлагаемого метода с методом квадратов, описанным в [6], рассмотрим другой из приведенных там примеров — автомат с 11 состояниями. В данном случае $w_{ij} = 1$, если состояния q_i и q_j связаны переходом, и $w_{ij} = 0$ в противном случае, поскольку вероятности переходов здесь не учитываются. Значения w_{ij} на парах состояний автомата удобно задать в виде табл. 4.

Для доведения числа состояний до целой степени двойки вводим пять фиктивных состояний и определим $w_{ij} = 0$, если хотя бы одно из q_i и q_j является фиктивным. Одномерные гиперкубы, получаемые на первом шаге, изображены на рис. 4, где ребра, у которых оба конца соответствуют фиктивным состояниям, не присутствуют.

Используя тот же способ, что и в предыдущем примере, вводим ребра q_1q_6, q_3q_7, q_2q_9, q_4q_8, q_5q_{11} и $q_{10}q_{12}$; получим три двумерных гиперкуба (четвертый гиперкуб с нулевыми весами ребер не представляет интереса), изображенных на рис. 5. При переборе

Таблица 4

Значения w_{ij}

q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	
0	1	0	0	1	0	1	0	0	0	q_1
	0	1	0	1	0	0	1	0	1	q_2
		0	0	1	1	0	1	0	0	q_3
			1	0	0	1	0	0	0	q_4
				0	0	0	0	1	1	q_5
					1	0	0	1	0	q_6
						0	1	0	0	q_7
							1	0	0	q_8
								0	0	q_9
									1	q_{10}

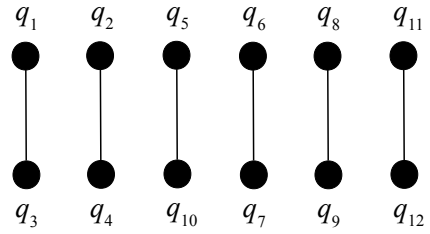


Рис. 4. Одномерные гиперкубы

вариантов введения ребер следует учитывать верхнюю границу суммы их весов, на данном шаге равную 2.

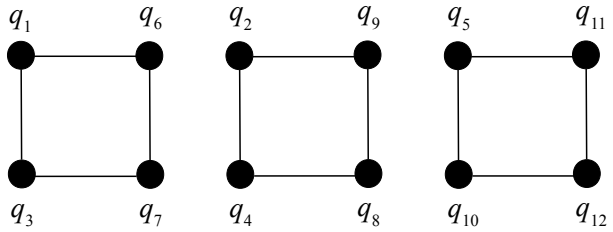


Рис. 5. Двумерные гиперкубы

Следующим шагом является построение двух трехмерных гиперкубов путем введения двух четверок ребер. При этом в данном примере достаточно рассматривать только такую четверку, в которой каждое ребро имеет не более одного конца, соответствующего фиктивному состоянию.

Все рассматриваемые варианты присоединения ребер с соответствующей величиной суммы их весов приведены в табл. 5, которая имеет три секции, соответствующие трем парам гиперкубов. Первая строка соответствует исходному положению гиперкубов, последующие строки — перестановкам, определяемым операторами E_1, E_2 и их сочетанию E_1E_2. Выбирается четвертый вариант с максимальной суммой весов, равной 2. Результатом выполнения этого шага является трехмерный гиперкуб, изображенный на рис. 6. Во втором гиперкубе, также изображенном на рис. 6, одна из гиперграней полностью состоит из вершин, соответствующих фиктивным состояниям, и для сборки этого гиперкуба не надо перебирать ребра.

Таблица 5

Варианты присоединения ребер на третьем шаге

Ребра	Σw_{ij}	Ребра	Σw_{ij}	Ребра	Σw_{ij}
q_1q_2, q_6q_9, q_3q_4, q_7q_8	0	q_1q_5, q_6q_{11}, q_7q_{12}, q_3q_{10}	0	q_2q_5, q_9q_{11}, q_8q_{12}, q_4q_{10}	0
q_1q_9, q_2q_6, q_3q_8, q_4q_7	1	q_1q_{11}, q_5q_6, q_7q_{10}, q_3q_{12}	0	q_2q_{11}, q_5q_9, q_8q_{10}, q_4q_{12}	1
q_1q_4, q_6q_8, q_7q_9, q_2q_3	1	q_1q_{10}, q_6q_{12}, q_7q_{11}, q_3q_5	0	q_2q_{10}, q_9q_{12}, q_8q_{11}, q_4q_5	1
q_1q_8, q_4q_6, q_2q_7, q_3q_9	2	q_1q_{12}, q_6q_{10}, q_5q_7, q_3q_{11}	1	q_2q_{12}, q_9q_{10}, q_5q_8, q_4q_{11}	0

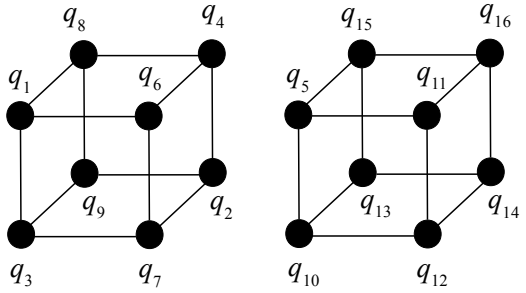


Рис. 6. Результат формирования трехмерных гиперкубов

Перебор вариантов соединения полученных трехмерных гиперкубов для получения окончательного решения в виде четырехмерного гиперкуба представлен в табл. 6. На последнем шаге качество сборки удобно оценивать по дефекту отображения ребер D, значения которого показаны в последней строке табл. 6. Наименьшее значение $D = 6$ получено в результате применения оператора E_2.

Таблица 6

Варианты последнего шага сборки гиперкуба

Код Грея	Исходное положение		Применение операторов E_i						
			E_1	E_2	E_3	E_1E_2	E_1E_3	E_2E_3	$E_1E_2E_3$
000	q_1	q_5	q_{15}	q_{11}	q_{10}	q_{16}	q_{13}	q_{12}	q_{14}
001	q_3	q_{10}	q_{13}	q_{12}	q_5	q_{14}	q_{15}	q_{11}	q_{16}
011	q_7	q_{12}	q_{14}	q_{10}	q_{11}	q_{13}	q_{16}	q_5	q_{15}
010	q_6	q_{11}	q_{16}	q_5	q_{12}	q_{15}	q_{14}	q_{10}	q_{13}
110	q_4	q_{16}	q_{11}	q_{15}	q_{14}	q_5	q_{12}	q_{13}	q_{10}
111	q_2	q_{14}	q_{12}	q_{13}	q_{16}	q_{10}	q_{11}	q_{15}	q_5
101	q_9	q_{13}	q_{10}	q_{14}	q_{15}	q_{12}	q_5	q_{16}	q_{11}
100	q_8	q_{15}	q_5	q_{16}	q_{13}	q_{11}	q_{10}	q_{14}	q_{12}
D		10	9	6	8	8	8	8	8

Окончательное решение в виде четырехмерного гиперкуба получаем, соединив ребрами q_1q_{11}, q_2q_{13}, q_3q_{12}, q_4q_{15}, q_5q_6, q_7q_{10}, q_8q_{16} и q_9q_{14} трехмерные гиперкубы, показанные на рис. 6. Это решение приведено на рис. 7. Коды состояний автомата можно получить по табл. 6. Для этого к кодам Грея, соответствующим состояниям из второго столбца, надо приписать слева 0, а к кодам Грея, соответствующим состояниям из столбца E_2, приписать слева 1. Тогда получим следующее кодирование для 11 состояний: $q_1 - 0000$, $q_2 - 0111$, $q_3 - 0001$, $q_4 - 0110$, $q_5 - 1001$, $q_6 - 0010$, $q_7 - 0011$, $q_8 - 0100$, $q_9 - 0101$, $q_{10} - 1011$, $q_{11} - 1000$.

Заметим, что дефект отображения D, характеризующий качество решения, здесь оказался равным дефекту отображения, полученному для того же примера в [6]. Если

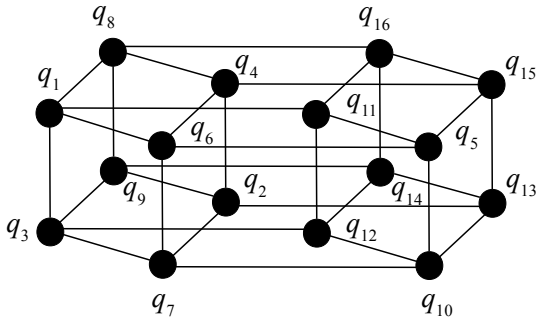


Рис. 7. Результат сборки четырехмерного гиперкуба

взять способ построения гиперкуба из работы [6], при котором выбирается вариант соединения четырьмя ребрами грани $q_5q_{11}q_{12}q_{10}$ с одной из граней куба с вершинами q_1, q_2, q_3, q_4, q_6, q_7, q_8 и q_9, то удается получить $D = 4$, т. е. на 2 меньше, чем в [6], но этот способ намного более трудоемкий.

Заключение

Предлагаемый метод кодирования состояний автомата рассчитан на использование его в автоматизированной системе логического проектирования. Минимизация интенсивности переключений не противоречит минимизации количества элементов в схеме. Предлагаемый метод допускает частичное совместное решение этих двух задач на этапе кодирования состояний. Решение примеров показывает, что предлагаемый подход не хуже известных и может быть использован для энергосберегающего кодирования состояний автомата.

Автор выражает благодарность рецензенту за ценные замечания, позволившие устранить ошибки и повысить качество изложения.

ЛИТЕРАТУРА

1. *Мурога С.* Системное проектирование сверхбольших интегральных схем. В 2 кн. Кн. 1. М.: Мир, 1985. 288 с.
2. *Pedram M.* Power minimization in IC design: Principles and applications // ACM Trans. Design Automat. Electron. Syst. 1996. V. 1. P. 3–56.
3. *Kashirova L., Keevallik A., and Meshkov M.* State assignment of finite state machine for decrease of power dissipation // Second Inter. Conf. Computer-Aided Design Discrete Devices — CAD DD'97. Minsk, Republic of Belarus, November 12–14, 1997. V. 1. Minsk: National Academy of Sciences of Belarus, Institute of Engineering Cybernetics, 1997. P. 60–67.
4. *Sudnitson A.* Partition search for FSM low power synthesis // Fourth Inter. Conf. Computer-Aided Design of Discrete Devices — CAD DD'2001. Minsk, Republic of Belarus, November 14–16, 2001. V. 1. Minsk: National Academy of Sciences of Belarus, Institute of Engineering Cybernetics, 2001. P. 44–49.
5. *Закревский А. Д.* Об оптимальном размещении графа в булевом пространстве // Вестник Томского госуниверситета. Приложение. 2005. № 14. С. 13–17.
6. *Закревский А. Д.* Алгоритмы энергосберегающего кодирования состояний автомата // Информатика. 2011. № 1(29). С. 68–78.
7. *Armstrong D. B.* A programmed algorithm for assigning internal codes for sequential machines // IRE Trans., EC-11. 1962. No. 4. P. 466–472.

8. *Armstrong D. B.* On the efficient assignment of internal codes to sequential machines // IRE Trans, EC-11. 1962. No. 5. P. 611–622.
9. *Оранов А. М.* Размещение множества вершин взвешенного графа в простой цепи // Логическое проектирование дискретных устройств. Минск: Ин-т техн. кибернетики АН БССР, 1984. С. 54–61.
10. *Закревский А. Д., Поттосин Ю. В., Черемисинова Л. Д.* Логические основы проектирования дискретных устройств. М.: Физматлит, 2007. 592 с.
11. *Pottosin Yu. V.* «Assembling» a Boolean hypercube: an approach to state assignment of finite state machines // Second Inter. Conf. Computer-Aided Design of Discrete Devices — CAD DD'97. Minsk, Republic of Belarus, November 12–14, 1997. V. 1. Minsk: National Academy of Sciences of Belarus, Institute of Engineering Cybernetics, 1997. P. 54–59.
12. *Macii E., Pedram M., and Somenzi F.* High-level power modeling, estimation and optimization // IEEE Trans. Computer-Aided Design Integrated Circuits and Systems. 1998. V. 17. No. 11. P. 1061–1079.

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

DOI 10.17223/20710410/14/9

УДК 519.7

РЕГУЛЯРНЫЕ ОЦЕНКИ СЛОЖНОСТИ УМНОЖЕНИЯ МНОГОЧЛЕНОВ И УСЕЧЕННОГО ДПФ[1]

И. С. Сергеев

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

E-mail: isserg@gmail.com

Строятся схемы для умножения многочленов и усеченного ДПФ (дискретного преобразования Фурье), эффективные с точки зрения сложности и глубины или сложности и объема памяти. Как следствие, умножение многочленов суммарной степени $n-1$, где $n = 2^{n_1} + \ldots + 2^{n_s}$, $n_1 > \ldots > n_s$, над кольцом, в котором обратима двойка, можно выполнить со сложностью $M(n_1) + \ldots + M(n_s) + \mathrm{O}(n)$ арифметических операций в этом кольце и глубиной $\max\limits_i\{D(n_i)\} + \mathrm{O}(\log n)$, где $M(k)$ и $D(k)$ — соответственно сложность и глубина схемы умножения по модулю $x^{2^k}+1$. Усеченное ДПФ порядка n (т. е. ДПФ порядка $2^{\lceil \log_2 n \rceil}$, приведенное к векторам длины n) можно реализовать схемой сложности $1{,}5n\log_2 n + \mathrm{O}(n)$ и объема памяти $n+1$.

Ключевые слова: *арифметические схемы, сложность, глубина, объем памяти, умножение, дискретное преобразование Фурье (ДПФ).*

Введение

В настоящей работе рассматривается вопрос о построении схем для умножения многочленов со сложностью, регулярно (равномерно, гладко) зависящей от размерности входа (т. е. степеней многочленов). Типичной является ситуация, когда известны эффективные (базовые) схемы умножения по модулям $x^{2^k}+1$, но требуется построить схему умножения многочленов произвольной суммарной степени $n-1$ и чтобы при этом она имела сложность, асимптотически так же зависящую от n, как и в случае $n = 2^k$ (асимптотика, естественно, предполагается гладкой функцией). В теоретических работах эта цель обычно достигается видоизменением базового алгоритма умножения, например расширением его на модули вида $x^{s2^k}+1$. Но подобные приемы обычно ухудшают уже второй член в оценке сложности. (Термин «второй член» здесь и далее употребляется неформально.) Интересно, что лучшего результата можно добиться, совсем не вмешиваясь в базовый алгоритм (используя его в качестве «черного ящика»). Для этого достаточно уметь эффективно приводить многочлен по набору модулей $x^{2^i}+1$ и восстанавливать его по известным остаткам от деления на эти же модули. Об этом и пойдет речь далее.

[1]Работа выполнена при финансовой поддержке РФФИ, проекты 11–01–00508 и 11–01–00792–а, и программы фундаментальных исследований Отделения математических наук РАН «Алгебраические и комбинаторные методы математической кибернетики и информационные системы нового поколения» (проект «Задачи оптимального синтеза управляющих систем»).

Вышеуказанной задаче созвучна задача эффективной реализации усеченного ДПФ (в западной литературе TFT — *Truncated Fourier Transform*). Наиболее эффективно вычисляется ДПФ порядка степени двойки, поэтому в работе [1] предложено вместо ДПФ произвольного порядка n использовать n компонент ДПФ порядка $2^{\lceil \log_2 n \rceil}$; соответствующее преобразование в [1] предложено называть усеченным ДПФ.

В работе строятся схемы для решения перечисленных задач, эффективные с точки зрения сложности (потенциально может сохраняться более одного члена из асимптотики сложности базовой схемы) и одновременно глубины или объема требуемой памяти. Постановка задачи о минимизации объема памяти позаимствована из [2].

Дополнительно рассмотрена задача эффективной реализации умножения многочленов при помощи схем умножения по модулям $x^{2\cdot 3^k}+x^{3^k}+1$. Такие схемы предложил использовать для умножения А. Шёнхаге [3]. Метод Шёнхаге обобщается на умножение по модулям $x^{(p-1)\cdot p^k} + x^{(p-2)\cdot p^k} + \ldots + x^{p^k} + 1$, где p — простое число [4]. Однако при $p \neq 3$ это обобщение, по всей видимости, практического значения пока не имеет (см. также [5, 6]). Эти схемы обычно применяются тогда, когда двойка необратима в кольце коэффициентов, зато обратима тройка (популярный пример — конечные поля характеристики 2). В этом случае, как показано ниже, для умножения многочленов суммарной степени $n-1$ можно построить схемы с регулярной оценкой сложности для $n \in \bigcup\limits_{k\in\mathbb{N}} [2\cdot 3^k,\ 3^{k+1})$ и с ухудшением асимптотики сложности в не более чем $4/3$ раза для прочих n при условии, что базовые схемы не модифицируются.

Изложим существо вопроса более подробно, попутно вводя необходимые понятия.

Пусть $\mathbf{K}$ — коммутативное (и ассоциативное) кольцо с единицей. Элемент $\zeta \in \mathbf{K}$ называется *примитивным корнем* степени N из единицы, если $\zeta^N = 1$ и при любом простом $p|N$ элемент $\zeta^{N/p} - 1$ не является делителем нуля в $\mathbf{K}$.

Дискретным преобразованием Фурье порядка N называется $(\mathbf{K}^N \to \mathbf{K}^N)$-преобразование

$$\text{ДПФ}_{N,\zeta}(\gamma_0, \ldots, \gamma_{N-1}) = (\gamma_0^*, \ldots, \gamma_{N-1}^*), \qquad \gamma_j^* = \sum_{i=0}^{N-1} \gamma_i \zeta^{ij},$$

где ζ — примитивный корень степени N. Если элемент $N = 1 + \ldots + 1 \in \mathbf{K}$ обратим, то существует обратное к ДПФ преобразование (называемое *обратным ДПФ*), удовлетворяющее соотношению $\text{ДПФ}_{N,\zeta}^{-1} = (1/N)\text{ДПФ}_{N,\zeta^{-1}}$.

Удобно иметь в виду полиномиальную интерпретацию ДПФ как изоморфизма колец, отождествляя аргументы преобразования с вектором коэффициентов многочлена:

$$\text{ДПФ}_{N,\zeta} : \mathbf{K}[x]/(x^N - 1) \to \mathbf{K}^N, \quad \text{ДПФ}_{N,\zeta}(\Gamma(x)) = \left(\Gamma(\zeta^0), \ldots, \Gamma(\zeta^{N-1})\right).$$

Подробнее о ДПФ см., например, в [6].

Для анализа эффективности обсуждаемых далее алгоритмов удобно использовать стандартную вычислительную модель *схемы из функциональных элементов* [7, 8] (далее просто *схемы*). А именно, рассматриваются схемы над арифметическим базисом $\{x \pm y,\ xy\} \cup \{ax : a \in \mathbf{K}\}$ (*арифметические схемы*). Для программистов ближе понятие *неветвящейся программы*: по существу, неветвящаяся программа (далее просто *программа*) — это схема, для которой фиксирована последовательность выполнения операций (срабатывания элементов схемы). Таким образом, одной схеме могут соответствовать несколько программ.

Стандартным образом определяются несколько мер сложности схем (программ). Собственно *сложность* — число функциональных элементов в схеме (программе). *Глубина* схемы — максимальное число элементов в ориентированной цепи, соединяющей

вход и выход схемы. *Объем памяти* программы — максимальное по всем итерациям число промежуточных данных (включая все уже вычисленные выходы), используемых в последующих итерациях. Несколько искусственно объем памяти схемы можно определить как минимальный объем памяти по всем программам, соответствующим данной схеме.

Содержательно сложность отвечает времени выполнения программы на однопроцессорной машине или площади микросхемы, реализующей схему; глубина отвечает времени срабатывания микросхемы или числу параллельных шагов, выполняемых многопроцессорной машиной; объем памяти в определении соответствует объему памяти, отводимому для хранения переменных при реализации программы на ЭВМ.

При анализе объема памяти процесс вычислений удобно представлять следующим образом. Промежуточные данные хранятся в ячейках памяти (емкость ячейки равна одному элементу кольца $\mathbf{K}$). В начале работы ячейки содержат входные данные. Элементарные (базисные) операции выполняются последовательно. Каждая операция использует данные, находящиеся в памяти в момент ее выполнения. Результат выполнения операции сохраняется в некоторой ячейке.

Наиболее эффективно реализуется ДПФ порядка степени двойки. Как следствие, быстрые алгоритмы эффективно умножают многочлены суммарной степени $n-1$, если $n = 2^k$, в предположении, что кольцо коэффициентов или подходящее его расширение допускает ДПФ порядка степени двойки.

Как известно, методом Кули — Тьюки [9] ДПФ порядка 2^k может быть реализовано схемой из $k2^k$ элементов сложения-вычитания, $(k-2)2^{k-1}+1$ элементов умножения на степени примитивного корня (степени 2^k), глубины $2k-1$ (если учитывать только аддитивные операции, то глубина равна k). Эта схема может быть перестроена так, чтобы с той же сложностью иметь объем памяти 2^k+1. Добавив $k2^{k-1}$ элементов умножения на 2, объем памяти можно сократить до 2^k; здесь имеется в виду, что основное преобразование схемы, ДПФ порядка 2 $(x, y) \to (x+y, x-y)$, можно вычислить со сложностью 2 и объемом памяти 3 в цепочке $(x, y) \to (x+y, x, y) \to (x+y, x-y)$ или со сложностью 3 и объемом памяти 2 в цепочке $(x, y) \to (x + y, y) \to (x + y, 2y) \to (x + y, x - y)$.

Обратное ДПФ можно реализовать так же, как и прямое, заменив ζ на ζ^{-1} и выполнив в конце 2^k умножений на константу $2^{-k} \in \mathbf{K}$. Впрочем, часть этих умножений можно совместить с внутренними умножениями на степени примитивного корня.

Если $n \neq 2^k$, то «по умолчанию» можно использовать алгоритм умножения многочленов суммарной степени $2^{\lceil \log_2 n \rceil} - 1$. Но при этом сложность алгоритма умножения с ростом n растет нерегулярно, возникают примерно двукратные скачки при переходе через степени двойки. Известны приемы, позволяющие сгладить функцию сложности так, чтобы при любом n она асимптотически так же выражалась через n, как и в случае $n = 2^k$. Например, ДПФ порядка степеней двойки можно заменить на ДПФ порядка $s2^l$, где $s \ll 2^l$ [10], или чуть более общими многократными ДПФ [11]. Указанные приемы, однако, как правило, ухудшают второй по порядку роста член функции сложности и переносят скачки в него.

Другой подход предложил Ван дер Хувен в [1, 12], где реализуется усеченное ДПФ, вычисляющее значение ДПФ на некотором подмножестве точек из множества, соответствующего ДПФ порядка ближайшей сверху степени двойки. *Усеченное ДПФ* (*УДПФ*) *порядка* n определяется как набор из n некоторых компонент вектора $\text{ДПФ}_{2^{\Lambda(n)},\zeta}(\gamma_0, \ldots, \gamma_{n-1}, 0, \ldots, 0)$, где $\Lambda(n) = \lceil \log_2 n \rceil$.

Ван дер Хувен [1] предложил в УДПФ включать компоненты — значения в точках $\zeta^{\rho(i)}$, $i = 0, \ldots, n-1$, где $\rho(i)$ — число, двоичная запись которого получается из двоичной записи числа i дополнением нулями до $\Lambda(n)$ разрядов и последующим обращением.

Схема прямого УДПФ в методе [1] получается из схемы ДПФ порядка $2^{\Lambda(n)}$ удалением «ненужных» элементов. Схема обратного УДПФ устроена несколько сложнее. Для обеих схем получена оценка сложности $n\Lambda(n) + 2^{\Lambda(n)}$ аддитивных операций и $\lceil (n\Lambda(n) + 2^{\Lambda(n)})/2 \rceil$ умножений на степени примитивного корня (умножения на степени двойки в расчете игнорируются, точнее, допускаются операции вида $x \pm 2^s y$). Глубина схемы прямого УДПФ равна $\Lambda(n)$; глубина схемы обратного УДПФ несколько больше, но также $\mathrm{O}(\log n)$. Объем памяти этой схемы (точнее, схемы, перестроенной аналогично упомянутой выше схеме обычного ДПФ), как отмечено в [2], равен $2^{\Lambda(n)}$. В [2] приведена конструкция схемы с объемом памяти $n + \mathrm{O}(1)$, однако большей асимптотической сложности, хотя по порядку и той же самой, $\mathrm{O}(n \log n)$.

С позиции задачи умножения многочленов близкие к использованию УДПФ идеи ранее высказывались в [13], где предлагалось выполнять умножение по двум модулям, допускающим быстрый алгоритм умножения, и окончательно восстанавливать произведение, опираясь на китайскую теорему об остатках, и позже в [5], где предлагалось использовать несколько модулей. На языке ДПФ это означает выбор УДПФ порядка l, где число l имеет малый (двоичный) вес.

В работе [14] предложено в качестве точек УДПФ порядка n выбирать корни многочленов $x^{2^i} + 1$ суммарной степени n.

Для дальнейшего анализа удобно ввести понятие *нечетного ДПФ* (*НДПФ*) *порядка* N:

$$\text{НДПФ}_{N,\zeta} : \mathbf{K}[x]/(x^N + 1) \to \mathbf{K}^N, \quad \text{НДПФ}_{N,\zeta}(\Gamma(x)) = \left(\Gamma(\zeta^1), \Gamma(\zeta^3), \ldots, \Gamma(\zeta^{2N-1})\right),$$

где ζ — примитивный корень степени $2N$. Другими словами, компонентами НДПФ порядка N являются компоненты ДПФ порядка $2N$, отличные от компонент ДПФ порядка N (если иметь в виду полиномиальную интерпретацию ДПФ). Простой способ реализации $\text{НДПФ}_{N,\zeta}$ состоит в композиции замены переменной $x \to \zeta x$ и $\text{ДПФ}_{N,\zeta^2}$. Соответственно обратное НДПФ можно реализовать как композицию $\text{ДПФ}^{-1}_{N,\zeta^2}$ и замены переменной $x \to x/\zeta$.

Пусть $n = 2^{n_1} + \ldots + 2^{n_s}$, где $n_1 > \ldots > n_s$. Схема УДПФ [14] строится из схем НДПФ порядков $2^{n_1}, \ldots, 2^{n_s}$ плюс дополнительно используется $2^{\Lambda(n)} + n$ аддитивных операций и n умножений. В схеме обратного УДПФ, помимо схем обратных НДПФ порядков $2^{n_1}, \ldots, 2^{n_s}$, используется $3 \cdot 2^{\Lambda(n)} + n$ аддитивных операций и n умножений. Указанные оценки сложности рассчитаны в [14] для чуть более общей задачи и могут быть уменьшены. Глубина и объем памяти схем [14] оцениваются приблизительно так же, как и для [1].

Далее получены несколько более точные оценки сложности данного УДПФ вместе с оценками глубины и показано, что УДПФ порядка n можно реализовать схемой с объемом памяти n (или $n + 1$) и дополнительной сложностью $\mathrm{O}(n)$.

Вопрос об оптимизации объема памяти естественно рассмотреть и в задаче умножения многочленов. В работе [2] на основе эффективной схемы для УДПФ построена схема умножения многочленов степени $n/2 - 1$ со сложностью $\mathrm{O}(n \log n)$ и объемом памяти $2n + \mathrm{O}(1)$, причем n «ячеек» памяти, отводимых под хранение коэффициентов исходных многочленов, не модифицируются.

Из полученных далее оценок вытекает, в частности, что для умножения многочленов суммарной степени $n - 1$ можно построить схему сложности $\mathrm{O}(n \log n)$ и объема

памяти $2n$. Запрет на переписывание коэффициентов исходных многочленов при этом не накладывается. Если иметь в виду, что фактически строится схема для умножения по модулю некоторого многочлена степени n, то объем памяти такой схемы не может быть далее понижен.

Результаты для задач реализации УДПФ и умножения многочленов формулируются в п. 1.1 и доказываются в п. 1.2. В п. 2 схемы для умножения многочленов строятся из схем умножения по модулям $x^{2\cdot 3^k} + x^{3^k} + 1$.

1. УДПФ и двоичный метод умножения

1.1. Основные результаты

Пусть в нашем распоряжении есть схемы для НДПФ порядка 2^k сложности $\Phi(k) = \Phi_A(k) + \Phi_2(k) + \Phi_C(k)$, глубины $d_\Phi(k)$ и объема памяти $v_\Phi(k)$. Здесь $\Phi_A(k)$ — число аддитивных элементов; $\Phi_2(k)$ — число умножений на степени двойки; $\Phi_C(k)$ — число прочих скалярных умножений. Аналогичные обозначения со штрихами введем для параметров схем обратных НДПФ.

Зафиксируем обозначение $n = 2^{n_1} + \ldots + 2^{n_s}$, где $n_1 > \ldots > n_s$.

Теорема 1. Усеченное ДПФ порядка n можно реализовать схемой

а) из $2n + \sum\limits_i \Phi_A(n_i)$ аддитивных операций, $\sum\limits_i \Phi_2(n_i)$ и $\sum\limits_i \Phi_C(n_i)$ умножений на степени двойки и прочие константы соответственно, и глубины $n_1 + \max\limits_i\{d_\Phi(n_i) - n_i\} + 1$;

б) из $4n - 2^{n_1+1} + \sum\limits_i \Phi_A(n_i)$ аддитивных операций, $4n - 3\cdot 2^{n_1} + \sum\limits_i \Phi_2(n_i)$ и $\sum\limits_i \Phi_C(n_i)$ умножений на степени двойки и прочие константы соответственно, и объема памяти $n + \max\limits_i\{v_\Phi(n_i) - 2^{n_i}\}$.

Обратное усеченное ДПФ порядка n можно реализовать схемой

а) из $4n - 3\cdot 2^{n_1} + \sum\limits_i \Phi'_A(n_i)$ аддитивных операций, $2n - 2^{n_1+1} + \sum\limits_i \Phi'_2(n_i)$ и $\sum\limits_i \Phi'_C(n_i)$ умножений на степени двойки и прочие константы соответственно, и глубины $n_1 - n_s + {}$ $+ 2s - 1 + \max\limits_i\{d'_\Phi(n_i)\}$;

б) из $4n - 2^{n_1+1} + \sum\limits_i \Phi'_A(n_i)$ аддитивных операций, $2n - 2^{n_1+1} + \sum\limits_i \Phi'_2(n_i)$ и $\sum\limits_i \Phi'_C(n_i)$ умножений на степени двойки и прочие константы соответственно, и объема памяти $n + \max\limits_i\{v'_\Phi(n_i) - 2^{n_i}\}$.

Теорема вытекает непосредственно из доказываемых ниже лемм 2 и 3.

Конкретные оценки можно получить при подстановке параметров схем НДПФ, указанных во введении. Например, $\Phi_A(k) = \Phi'_A(k) = k2^k$, $\Phi_C(k) = \Phi'_C(k) = k2^{k-1}$, $\Phi_2(k) = 0$, $\Phi'_2(k) = 2^k$ и либо $d_\Phi(k) = d'_\Phi(k) = 2k$, либо $v_\Phi(k) = v'_\Phi(k) = 2^k + 1$.

Теперь пусть имеются схемы для умножения многочленов по модулям $x^{2^k}+1$ с коэффициентами над кольцом, в котором обратим элемент 2. Сложность, глубину и объем памяти таких схем будем обозначать через $M(k) = M_A(k) + M_2(k) + M_C(k) + M_N(k)$, $d_M(k)$ и $v_M(k)$, где $M_N(k)$ обозначает число нескалярных умножений в схеме (остальные обозначения аналогичны введенным выше для схем НДПФ). Из лемм 2 и 3 также следует

Теорема 2. Для умножения многочленов суммарной степени $n-1$ можно построить схему

а) из $6n-3\cdot 2^{n_1}+\sum\limits_i M_A(n_i)$ аддитивных операций, $2n-2^{n_1+1}+\sum\limits_i M_2(n_i)$, $\sum\limits_i M_C(n_i)$ и $\sum\limits_i M_N(n_i)$ умножений на степени двойки, прочие константы и нескалярных умножений соответственно, и глубины $\max\limits_i\{d_M(n_i)-n_i\}+2n_1-n_s+2s$;

б) из $12n-6\cdot 2^{n_1}+\sum\limits_i M_A(n_i)$ аддитивных операций, $10n-8\cdot 2^{n_1}+\sum\limits_i M_2(n_i)$, $\sum\limits_i M_C(n_i)$ и $\sum\limits_i M_N(n_i)$ умножений на степени двойки, прочие константы и нескалярных умножений соответственно, и объема памяти $2n+\max\limits_i\{v_M(n_i)-2^{n_i+1}\}$.

Учитывая, что $s\leqslant n_1+1$, получаем

Следствие 1. Пусть $M(k)=f(2^k)$ и для любых $x,y\geqslant 1$ справедливо $f(x+y)\geqslant \geqslant f(x)+f(y)$. Пусть также $d_M(k)-k\leqslant d_M(l)-l$ и $v_M(k)-2^{k+1}\leqslant v_M(l)-2^{l+1}$ при любых $k\leqslant l$. Тогда для умножения многочленов суммарной степени $n-1$ можно построить схему

а) сложности $f(n)+8n-5\cdot 2^{n_1}$ и глубины $d_M(n_1)+3n_1+2$;

б) сложности $f(n)+22n-14\cdot 2^{n_1}$ и объема памяти $2n+v_M(n_1)-2^{n_1+1}$.

Выбирая ближайшее сверху к n число n', кратное $2^{n_1-\alpha(n)}$, где $\alpha(n)$ — медленно растущая натуральная функция, и переходя к схеме умножения многочленов суммарной степени не выше $n'-1$, получаем еще одно следствие из теоремы 2, п. *а* (используя неравенства $n_1-n_s\leqslant\alpha(n)$, $s\leqslant\alpha(n)+1$):

Следствие 2. В условиях следствия 1 дополнительно предположим, что $f(x)/x\to\infty$ при $x\to\infty$ и $f(x)=x^{\mathrm{O}(1)}$. Тогда для умножения многочленов суммарной степени $n-1$ можно построить схему сложности не более $(1+\mathrm{o}(1))f(n)$ и глубины не более $d_M(\lfloor\log_2(n+o(n))\rfloor)+\mathrm{o}(\log n)$.

Заметим, что предположения в формулировках следствий являются естественными. Во-первых, исходим из того, что схемы умножения по модулям $x^{2^k}+1$ устроены единообразно. Во-вторых, предполагаем, что частично определенная функция сложности схем умножения $M(k)$, $k\in\{2^i : i\in\mathbb{N}\}$, может быть доопределена до функции $f(x)$, удовлетворяющей условию суперлинейности $f(x+y)\geqslant f(x)+f(y)$ (следствие 1) и имеющей нелинейный рост (следствие 2). Первое из условий на $f(x)$ фактически следует из второго, а второе опирается на широко распространенное предположение о нелинейности функции сложности умножения. Если какое-то из условий все же не может быть удовлетворено, то вместо $f(x)$ можно взять подходящую функцию $f_>(x)\geqslant f(x)$, удовлетворяющую обоим условиям.

Конкретные оценки можно получить при подстановке известных параметров схем умножения по модулю $x^{2^k}+1$ — они приводятся, например, в [5, 6, 14, 15].

Стандартным образом схему умножения по модулю $x^{2^k}+1$ можно построить из двух схем НДПФ, схемы обратного НДПФ порядка 2^k (если ДПФ порядка 2^{k+1} существует в рассматриваемом кольце) и 2^k нескалярных умножений, выполняемых на одном уровне. Подставляя известные параметры этих схем, в частности для схемы из п. *б* теоремы 2 получаем оценку сложности $\mathrm{O}(n\log n)$ при объеме памяти $2n$.

1.2. Вспомогательные утверждения

Пусть $a(x)=\sum\limits_{l=0}^{n-1}a_lx^l$. Положим формально $a_l=0$ при $l\geqslant n$. Введем обозначения $a_{k,l}$ и $b_{k,l}$:

$$a(x)\bmod(x^{2^k}-1)=\sum_{l=0}^{2^k-1}a_{k,l}x^l,\quad a(x)\bmod(x^{2^k}+1)=\sum_{l=0}^{2^k-1}b_{k,l}x^l.$$

Очевидно, что коэффициенты $a_{k,l}$ и $b_{k,l}$ являются соответственно знакопостоянными и знакопеременными суммами коэффициентов a_i с шагом 2^k (по индексам):

$$a_{k,l} = a_{k+1,l} + a_{k+1,2^k+l} = \sum_{j2^k<n-l} a_{j2^k+l}, \quad b_{k,l} = a_{k+1,l} - a_{k+1,2^k+l} = \sum_{j2^k<n-l} (-1)^j a_{j2^k+l}.$$

Следующая лемма используется в первую очередь при построении схем, эффективных с точки зрения памяти.

Лемма 1. Пусть $n_{i+1} < k \leqslant n_i$. Справедливы следующие формулы:

$$a_{k,l} = \sum_{j_i=0}^{2^{n_i-k}-1} \left(b_{n_i,\, j_i 2^k+l} + 2 \sum_{j_{i-1}=0}^{2^{n_{i-1}-n_i-1}-1} \left(b_{n_{i-1},\, (2j_{i-1}+1)2^{n_i}+j_i 2^k+l} + \right.\right.$$
$$+2 \sum_{j_{i-2}=0}^{2^{n_{i-2}-n_{i-1}-1}-1} \left(b_{n_{i-2},\, (2j_{i-2}+1)2^{n_{i-1}}+(2j_{i-1}+1)2^{n_i}+j_i 2^k+l} + \dots \right.$$
$$\left.\left.\left. \dots + 2 \sum_{j_1=0}^{2^{n_1-n_2-1}-1} b_{n_1,\, (2j_1+1)2^{n_2}+\dots+(2j_{i-1}+1)2^{n_i}+j_i 2^k+l} \dots \right) \right) \right) + 2^i a_{2^{n_1}+\dots+2^{n_i}+l};$$

$$b_{n_i,l} = \sum_{j=0}^{1} (-1)^j \left(\sum_{j_{i-1}=0}^{2^{n_{i-1}-n_i-1}-1} \left(b_{n_{i-1},\, (2j_{i-1}+j)2^{n_i}+l} + \right.\right.$$
$$+2 \sum_{j_{i-2}=0}^{2^{n_{i-2}-n_{i-1}-1}-1} \left(b_{n_{i-2},\, (2j_{i-2}+1)2^{n_{i-1}}+(2j_{i-1}+j)2^{n_i}+l} + \dots \right.$$
$$\left.\left.\left. \dots + 2 \sum_{j_1=0}^{2^{n_1-n_2-1}-1} b_{n_1,\, (2j_1+1)2^{n_2}+\dots+(2j_{i-2}+1)2^{n_{i-1}}+(2j_{i-1}+j)2^{n_i}+l} \dots \right) \right) \right) +$$
$$+2^{i-1} \left(a_{2^{n_1}+\dots+2^{n_{i-1}}+l} - a_{2^{n_1}+\dots+2^{n_i}+l} \right).$$

Доказательство. Первая из формул получается рекурсивным применением простых соотношений

$$a_{k,l} = \begin{cases} a_{k+1,l} + a_{k+1,2^k+l}, & k \notin \{n_i : i = 1, \dots, s\}, \\ b_{k,l} + 2a_{k+1,2^k+l}, & k \in \{n_i : i = 1, \dots, s\}, \end{cases}$$

отталкиваясь от $a_{n_1+1,l} = a_l$ и учитывая, что $a_l = 0$ при $l \geqslant n$. Вторая формула получается из первой ввиду $b_{n_i,l} = a_{n_i+1,l} - a_{n_i+1,2^{n_i}+l}$. ■

Лемма 2. Пусть $m \leqslant n$. Тогда приведение многочлена степени $m-1$ по модулям $x^{2^{n_i}}+1$, $i = 1, \dots, s$, может быть выполнено схемой

а) из $2(m-1)$ аддитивных элементов и глубины $n_1 - n_s + 1$, причем коэффициенты остатка от деления на $x^{2^{n_i}}+1$ вычисляются на глубине $n_1 - n_i + 1$;

б) из $2(2n - 2^{n_1})$ аддитивных элементов, $4n - 3 \cdot 2^{n_1}$ умножений на степени двойки и объема памяти n.

Доказательство. Пункт *а* доказывает простая конструкция [14].

Несложно видеть, что все суммы $a_{k,l}$, где $k = n_s+1, \dots, n_1$, можно вычислить одной схемой сложности не выше $m-1$, в которой на глубине не более l вычисляются $a_{k,l}$ при $k = n_1 + 1 - l$.

Искомые коэффициенты $b_{n_i,l}$ получаются присоединением к построенной схеме элементов вычитания, расположенных на глубине $n_1 - n_i + 1$ в количестве

$$\begin{cases} 0, & m \leqslant 2^{n_i}, \\ m - 2^{n_i}, & 2^{n_i} \leqslant m \leqslant 2^{n_i+1}, \\ 2^{n_i}, & m \geqslant 2^{n_i+1}. \end{cases}$$

Поскольку $2^{n_i} > 2^{n_{i+1}} + \ldots + 2^{n_s}$, для числа вычитаний (при подходящем i) получаем оценку

$$m - 2^{n_i} + 2^{n_{i+1}} + \ldots + 2^{n_s} \leqslant m - 1,$$

откуда следует оценка $2(m-1)$ для сложности всей схемы.

Доказательство п. *б* достаточно провести для случая $m = n$. Перестроим схему из предыдущего пункта так, чтобы в ней явно вычислялись только те суммы $a_{k,l}$, для которых «достаточно памяти».

Обозначим $L_i = 2^{n_i} + \ldots + 2^{n_s}$. При любом $k = n_1, \ldots, n_s + 1$ явно вычисляем суммы $a_{k,0}, \ldots, a_{k,L_i-1}$, где $n_{i-1} \geqslant k > n_i$, учитывая, что $n - L_i$ прочих «ячеек памяти» отводится под хранение коэффициентов $b_{n_j,l}$, где $j < i$.

Все вычисление удобно разбить на этапы, нумеруя их числами от n_1 до n_s в порядке убывания. На этапе k вычисляются суммы $a_{k,0}, \ldots, a_{k,L_i-1}$ (где $n_{i-1} \geqslant k > n_i$) и, если $k \in \{n_i : i = 1, \ldots, s\}$, то вычисляются также коэффициенты $b_{k,0}, \ldots, b_{k,2^k-1}$.

Если сумма $a_{k,l}$ не вычисляется явно, то вместо нее используется правая часть формулы (обозначим ее $\varphi_{k,l}$) из леммы 1, выражающая ее через уже вычисленные коэффициенты $b_{n_j,t}$, где $n_j \geqslant k$. Заметим, что поскольку $l \geqslant L_{i+1}$ при $n_i \geqslant k > n_{i+1}$, то последнее слагаемое $a_{2^{n_1}+\ldots+2^{n_i}+l}$ в $\varphi_{k,l}$ равно нулю.

Обозначим через ρ_k число переменных в формуле $\varphi_{k,l}$ (это число не зависит от l, что следует из вида формулы). Тогда прибавление (вычитание) $\varphi_{k,l}$ выполняется за ρ_k аддитивных операций и i умножений на степени двойки без дополнительной памяти, где $n_i \geqslant k > n_{i+1}$.

Соответствующий способ проиллюстрируем на примере. Пусть требуется выполнить преобразование $a \to a + (b + 2c + 4d)$. Вычисления проведем в следующем порядке: a, $a + b$, $2^{-1}(a + b)$, $c + 2^{-1}(a + b)$, $2^{-1}(c + 2^{-1}(a + b))$, $d + 2^{-1}(c + 2^{-1}(a + b))$, $2^2(d + 2^{-1}(c + 2^{-1}(a + b))) = a + b + 2c + 4d$.

Оценим сложность схемы. Рассмотрим этап с номером k, где $n_{i-1} > k > n_i$. Доступны коэффициенты $a_{k+1,0}, \ldots, a_{k+1,L_i-1}$, остальные $a_{k+1,l}$ выражаются формулами $\varphi_{k+1,l}$. Заметим, что $L_i < 2^{n_i+1} \leqslant 2^k$. Тогда каждый из коэффициентов $a_{k,l}$, где $l < L_i$, вычисляется как $a_{k+1,l} + \varphi_{k+1,2^k+l}$ со сложностью ρ_{k+1} аддитивных операций и $i-1$ умножений на степени двойки. Сложность этапа оценивается как $L_i\rho_{k+1}$ аддитивных операций и $(i-1)L_i$ умножений на степени двойки.

Рассмотрим этап с номером $k = n_i$. Доступны коэффициенты $a_{k+1,0}, \ldots, a_{k+1,L_i-1}$. При этом $L_i = 2^k + L_{i+1}$. Сначала вычислим все $b_{k,l}$ по формулам $a_{k+1,l} - a_{k+1,2^k+l}$ при $l < L_{i+1}$ и $a_{k+1,l} - \varphi_{k+1,2^k+l}$ при остальных l. При этом коэффициентами $b_{k,l}$ «перезаписываются» $a_{k+1,l}$, $l \geqslant L_{i+1}$. Затем вычислим $a_{k,l}$ для $l < L_{i+1}$ по формулам $2a_{k+1,l} - b_{k,l}$. Сложность этапа оценивается как $2L_{i+1} + (2^k - L_{i+1})\rho_{k+1}$ аддитивных операций и $(i-1)2^k - (i-2)L_{i+1} = (i-1)L_i - (2i-3)L_{i+1}$ умножений на степени двойки.

Найдем ρ_k. Из вида формулы леммы 1 непосредственно следует, что

$$\rho_{n_i} = 2^{n_1-n_i-(i-1)} + 2^{n_2-n_i-(i-2)} + \ldots + 2^{n_{i-1}-n_i-1} + 1, \tag{1}$$

$\rho_k = 2^{n_i-k}\rho_{n_i}$ при $n_{i+1} < k < n_i$.

Оценим суммарную аддитивную сложность вычислений. Сумма сложностей этапов $n_i - 1, \ldots, n_{i+1}$ не превосходит

$$C_i = L_{i+1}\rho_{n_i}(1 + 2 + \ldots + 2^{n_i-n_{i+1}-2}) + 2L_{i+2} + (2^{n_{i+1}} - L_{i+2})2^{n_i-n_{i+1}-1}\rho_{n_i} \leqslant$$
$$\leqslant \rho_{n_i}2^{n_i-n_{i+1}-1}(L_{i+1} - L_{i+2} + 2^{n_{i+1}}) - \rho_{n_i}L_{i+1} + 2L_{i+2} = \rho_{n_i}(2^{n_i} - L_{i+1}) + 2L_{i+2}.$$

При $i \geqslant 1$ последнее выражение не превосходит $2^{n_i}\rho_{n_i}$, если учесть, что $L_{i+1} > 2L_{i+2}$ и $\rho_{n_i} \geqslant 1$.

Для сложности всех этапов, кроме этапа n_1, используя (1), получаем оценку

$$C_1 + \ldots + C_{s-1} = \sum_{i=1}^{s-1} 2^{n_i}\rho_{n_i} = \sum_{i=1}^{s-1}(2^{n_1-(i-1)} + 2^{n_2-(i-2)} + \ldots + 2^{n_i}) <$$
$$< (2^{n_1} + 2^{n_1-1} + \ldots) + (2^{n_2} + 2^{n_2-1} + \ldots) + \ldots + (2^{n_s} + 2^{n_s-1} + \ldots) < 2n.$$

Окончательно, оценивая сложность этапа n_1 как $2L_2 = 2(n - 2^{n_1})$, получаем утверждение п. *б* в части аддитивной сложности.

Число умножений на степени двойки на этапах $n_i - 1, \ldots, n_{i+1}$ оценим грубо как

$$D_i = (n_i - n_{i+1} - 1)iL_{i+1} + iL_{i+1} - (2i-1)L_{i+2} \leqslant$$
$$\leqslant i(n_i - n_{i+1})L_{i+1} \leqslant i2^{n_i-n_{i+1}-1}2^{n_{i+1}+1} = i2^{n_i}.$$

Тогда для суммарного числа умножений на всех этапах, кроме этапа n_1, имеем оценку

$$D_1 + \ldots + D_{s-1} = 2^{n_1} + 2 \cdot 2^{n_2} + 3 \cdot 2^{n_3} + \ldots + (s-1) \cdot 2^{n_{s-1}} =$$
$$= n + (n - 2^{n_1}) + (n - 2^{n_1} - 2^{n_2}) + \ldots <$$
$$< n + (n - 2^{n_1}) + (1/2)\,(n - 2^{n_1}) + (1/2)^2\,(n - 2^{n_1}) + \ldots = n + 2(n - 2^{n_1}).$$

Добавляя $n - 2^{n_1}$ умножений на 2 на этапе n_1, получаем итоговую оценку. ■

Лемма 3. Восстановление многочлена степени $n-1$ по заданным остаткам по модулям $x^{2^{n_i}} + 1$, $i = 1, \ldots, s$, может быть выполнено схемой

а) из $4n - 3 \cdot 2^{n_1}$ аддитивных элементов, $2(n - 2^{n_1})$ умножений на степени двойки и глубины не более $n_1 - n_s + 2s - 1$;

б) из $2(2n - 2^{n_1})$ аддитивных элементов, $2(n - 2^{n_1})$ делений на 2 и объема памяти n.

Доказательство. Воспользуемся леммой 1, чтобы выразить разности

$$h_{i,l} = a_{2^{n_1}+\ldots+2^{n_{i-1}}+l} - a_{2^{n_1}+\ldots+2^{n_i}+l}, \tag{2}$$

где $i = 1, \ldots, s-1$ и $l = 0, \ldots, L_i - 1$, через коэффициенты $b_{n_j,t}$. Заметим, что вычитаемый коэффициент равен нулю при $l \geqslant L_{i+1}$.

Для построения схемы из п. *а* вычисляем вспомогательные величины $c_{i,l}$ и $d_{i,l}$, определяемые равенствами

$$d_{1,l} = b_{n_1,l}, \quad c_{i,l} = \sum_{j=0}^{2^{n_{i-1}-n_i-1}-1} d_{i-1,\,j2^{n_i+1}+l}, \quad d_{i,l} = b_{n_i,l} + 2c_{i,2^{n_i}+l},$$

где $i \geqslant 2$. Затем находим искомые коэффициенты $h_{i,l}$ по формулам $h_{i,l} = 2^{1-i}(b_{n_i,l} - c_{i,l} + c_{i,2^{n_i}+l})$, а в случае $i = 1$ просто $h_{1,l} = b_{n_1,l}$.

Сложность вычислений, заключенная в формулах для $c_{i,l}$, $i = 2, \ldots, s$ и $l = 0, \ldots, 2^{n_i+1} - 1$, оценивается как

$$\sum_{i=2}^{s} 2^{n_i+1}(2^{n_{i-1}-n_i-1} - 1) = \sum_{i=2}^{s}(2^{n_{i-1}} - 2^{n_i+1}) < n - 2(n - 2^{n_1}).$$

Сложность, заключенная в формулах для $d_{i,l}$, $i = 2, \ldots, s-1$ и $l = 0, \ldots, 2^{n_i} - 1$, оценивается как $2^{n_2}+\ldots+2^{n_{s-1}} < n-2^{n_1}$ аддитивных операций и столько же умножений на 2. Для завершения вычисления $h_{i,l}$ нужно выполнить еще $2(2^{n_2}+\ldots+2^{n_s}) = 2(n-2^{n_1})$ аддитивных операций и $n - 2^{n_1}$ умножений на степени двойки.

Без труда проверяется, что $c_{i,l}$ при этом вычисляется на глубине $n_1 - n_i + i - 3$ и, следовательно, $h_{i,l}$ — на глубине $n_1 - n_i + i$.

По набору $h_{i,l}$ несложно восстанавливаются коэффициенты многочлена $a(x)$. Коэффициенты $a_{2^{n_1}+\ldots+2^{n_{i-1}}+l}$ просто совпадают с $h_{i,l}$ при $l \geqslant L_{i+1}$. В частности, в случае $i = s$ известны все коэффициенты. Это позволяет последовательно в порядке убывания i определить недостающие коэффициенты $a_{2^{n_1}+\ldots+2^{n_{i-1}}+l}$, $l = 0, \ldots, L_{i+1} - 1$, прямо по формулам (2) с общей сложностью

$$L_s + L_{s-1} + \ldots + L_2 = 2^{n_2} + 2 \cdot 2^{n_3} + 3 \cdot 2^{n_4} + \ldots < 2(n - 2^{n_1}).$$

Глубина этих вычислений, очевидно, не превосходит $s - 1$.

Складывая все оценки, получаем утверждение п. *а*.

Для доказательства п. *б* построим схему, последовательно в порядке убывания i преобразующую каждый коэффициент $b_{n_i,l}$ по формуле леммы 1 в соответствующую разность $h_{i,l}$.

Несложно видеть, что преобразование каждого коэффициента $b_{n_i,l}$ выполняется за σ_i аддитивных операций и $i - 1$ делений на 2, где σ_i — число коэффициентов $b_{n_j,t}$ в правой части второй формулы леммы 1. Непосредственно проверяется, что

$$\sigma_i = 2^{n_1-n_i-(i-2)} + 2^{n_2-n_i-(i-3)} + \ldots + 2^{n_{i-1}-n_i}.$$

Аддитивную сложность вычисления разностей $h_{i,l}$ теперь можно оценить как

$$\sum_{i=2}^{s} 2^{n_i}\sigma_i = \sum_{i=2}^{s}(2^{n_1-(i-2)} + 2^{n_2-(i-3)} + \ldots + 2^{n_{i-1}}) <$$
$$< (2^{n_1} + 2^{n_1-1} + \ldots) + (2^{n_2} + 2^{n_2-1} + \ldots) + \ldots + (2^{n_s} + 2^{n_s-1} + \ldots) < 2n.$$

Число делений на 2 оценивается как

$$\sum_{i=2}^{s}(i-1)2^{n_i} = 2^{n_2} + 2 \cdot 2^{n_3} + 3 \cdot 2^{n_4} + \ldots < 2(n - 2^{n_1}).$$

Заключительная часть схемы такая же, как в п. *а*. ■

2. Троичный метод умножения

2.1. Основные результаты

Здесь, если явно не оговаривается иное, будем полагать $n = 2\,(3^{n_1} + 3^{n_2} + \ldots + 3^{n_s})$, где $n_1 > n_2 > \ldots > n_s$. Заметим, что $n \in [2 \cdot 3^{n_1},\, 3^{n_1+1})$.

Рассмотрим способ использования схем для умножения многочленов по модулям $x^{2\cdot 3^k}+x^{3^k}+1$ с коэффициентами над кольцом, в котором обратим элемент 3. Сложность, глубину и объем памяти таких схем будем обозначать через $M(k)=$ $=M_A(k)+M_3(k)+M_C(k)+M_N(k)$, $d_M(k)$ и $v_M(k)$, где $M_A(k)$ обозначает число аддитивных операций; $M_3(k)$ — число умножений на степени тройки; $M_C(k)$ — число прочих скалярных умножений; $M_N(k)$ — число нескалярных умножений.

Из лемм 5 и 6, приведенных ниже, вытекает

Теорема 3. Для умножения многочленов суммарной степени не выше $n-1$ можно построить схему

а) из $5{,}5n-5\cdot 3^{n_1}+\sum\limits_i M_A(n_i)$ аддитивных операций, $1{,}5n-3^{n_1+1}+\sum\limits_i M_3(n_i)$, $\sum\limits_i M_C(n_i)$ и $\sum\limits_i M_N(n_i)$ умножений на степени тройки, прочие константы и нескалярных умножений соответственно, и глубины $4n_1+\max\limits_i\{d_M(n_i)-2n_i\}-2n_s+s+1$. В случае кольца характеристики 2 справедлива на $0{,}5n$ меньшая оценка аддитивной сложности;

б) из $15{,}5n-19\cdot 3^{n_1}+\sum\limits_i M_A(n_i)$ аддитивных операций, $4{,}75n-8{,}5\cdot 3^{n_1}+\sum\limits_i M_3(n_i)$, $\sum\limits_i M_C(n_i)$ и $\sum\limits_i M_N(n_i)$ умножений на степени тройки, прочие константы и нескалярных умножений соответственно, и объема памяти $2n+\max\limits_i\{v_M(n_i)-4\cdot 3^{n_i}\}$.

Следствие 3. Пусть $M(k)=f(2\cdot 3^k)$, где для любых $x,y\geqslant 1$ справедливо $f(x+y)\geqslant f(x)+f(y)$. Пусть также $d_M(k)-2k\leqslant d_M(l)-2l$ и $v_M(k)-4\cdot 3^k\leqslant v_M(l)-4\cdot 3^l$ при любых $k\leqslant l$. Тогда для умножения многочленов суммарной степени не выше $n-1$ можно построить схему

а) сложности $f(n)+7n-8\cdot 3^{n_1}$ и глубины $d_M(n_1)+3n_1+2$, а в случае кольца характеристики 2 для сложности схемы справедлива оценка $f(n)+5(n-3^{n_1})$;

б) сложности $f(n)+20{,}25n-27{,}5\cdot 3^{n_1}$ и объема памяти $2n+v_M(n_1-4\cdot 3^{n_1})$, а в случае кольца характеристики 2 сложность схемы не превосходит $f(n)+15{,}5n-19\cdot 3^{n_1}$.

В общем случае, а именно для $n\in\bigcup\limits_i[3^i,\,2\cdot 3^i)$, оценку сложности вида $(1+\mathrm{o}(1))f(n)$, не модифицируя базовый алгоритм умножения, получить пока не удается. Но можно доказать оценку $(4/3+\mathrm{o}(1))f(n)$: справедливо

Следствие 4. Пусть в условиях следствия 3 дополнительно выполняется $f(x)/x\to\infty$ при $x\to\infty$ и $f(x)=x^{\mathrm{O}(1)}$. Тогда для умножения многочленов суммарной степени не выше $n-1$ можно построить схему сложности не более

$$\begin{cases}(1+\mathrm{o}(1))f(n), & 2\cdot 3^k\leqslant n<3^{k+1},\\ \bigl(2-3^k/n+\mathrm{o}(1)\bigr)\,f(n), & 3^k\leqslant n<3^{k+1}/2,\\ \bigl(2\cdot 3^k/n+\mathrm{o}(1)\bigr)\,f(n), & 3^{k+1}/2<n<2\cdot 3^k\end{cases}$$

и глубины не более $d_M(\lfloor\log_3(2n+\mathrm{o}(n))\rfloor-1)+\mathrm{o}(\log n)$.

Доказательство. В первом случае конструкция та же, что и в следствии 2. В третьем случае используется схема умножения многочленов суммарной степени не более $2\cdot 3^k-1$.

В случае $n\in[3^k,\,3^{k+1}/2)$ рассмотрим ближайшее сверху число n', кратное $2\cdot 3^{k-\alpha(n)}$, где $\alpha(n)$ — медленно растущая натуральная функция. Если $n'>3^{k+1}/2$, то действуем, как в третьем случае.

Иначе, если $n'<3^{k+1}/2$, вычислим произведение по модулям $x^{2\cdot 3^i}+x^{3^i}+1$, где $i=k-1,\ldots,k-\alpha(n)$, схемой сложности $f(3^k-3^{k-\alpha(n)})+\mathrm{O}(n)$ и глубины $\mathrm{O}(\alpha(n))$ методом теоремы 3.

Фактически это дает нам $3^k - 3^{k-\alpha(n)}$ линейных уравнений на коэффициенты искомого произведения. Для получения оставшихся $n'' = n' - 3^k + 3^{k-\alpha(n)}$ уравнений заметим, что $n'' = L + U$, где

$$2L,\, 2U \in \{0\} \cup \left(\bigcup_{j=1}^{\alpha(n)} [2 \cdot 3^{k-j},\, 3^{k-j+1}) \right) \cap \{2 \cdot 3^{k-\alpha(n)}\mathbb{N}\}.$$

Действительно, любое натуральное число можно представить в виде суммы двух чисел, запись которых в системе счисления с основанием 3 состоит только из нулей и единиц. Все такие числа содержатся в множестве $\{0\} \cup \bigcup\limits_{j=1}^{\infty} [3^j,\, 3^{j+1}/2)$.

Перемножая отдельно младшие части многочленов суммарной степени $2L - 2$ и старшие части суммарной степени $2U - 2$, определяем L младших и U старших коэффициентов произведения. Это выполняется со сложностью $f(2L) + f(2U) + \mathrm{O}(L + U)$ и глубиной $\mathrm{O}(\alpha(n))$, согласно теореме 3.

Далее при помощи леммы 5 находим остатки от деления известной части искомого произведения на многочлены $x^{2\cdot 3^i} + x^{3^i} + 1$, $i = k-1, \ldots, k-\alpha(n)$, откуда находим остатки от деления неизвестной части $a(x)x^L$, где $\deg a < n' - L - U$, на те же многочлены. Все это выполняется со сложностью $\mathrm{O}(n)$ и глубиной $\mathrm{O}(\alpha(n))$.

Многочлен $f(x)$ восстанавливается слегка модифицированным методом леммы 6 (см. ниже; эту модификацию несложно построить) также со сложностью $\mathrm{O}(n)$ и глубиной $\mathrm{O}(\alpha(n))$. Следствие доказано. ■

Конструкции и оценки сложностных характеристик схем умножения по модулю $x^{2\cdot 3^k} + x^{3^k} + 1$ приводятся, например, в [5, 6, 14–16].

2.2. Вспомогательные утверждения

Пусть $a(x) = \sum\limits_{l=0}^{n-1} a_l x^l$. Положим формально $a_l = 0$ при $l \geqslant n$. Введем обозначения

$$a(x) \bmod (x^{3^k} - 1) = \sum_{l=0}^{3^k-1} a_{k,l} x^l, \quad a(x) \bmod (x^{2\cdot 3^k} + x^{3^k} + 1) = \sum_{r=0}^{1} \sum_{l=0}^{3^k-1} b_{k,r,l} x^l.$$

Коэффициенты $a_{k,l}$ и $b_{k,r,l}$ связывают простые соотношения

$$a_{k,l} = a_{k+1,l} + a_{k+1,3^k+l} + a_{k+1,2\cdot 3^k+l}, \quad b_{k,r,l} = a_{k+1,r3^k+l} - a_{k+1,2\cdot 3^k+l}.$$

На них основана

Лемма 4. Пусть $n_{i+1} < k \leqslant n_i$. Справедливы следующие формулы:

$$a_{k,l} = \sum_{j_i=0}^{3^{n_i-k}-1} \Big(b_{n_i,\,0,\,j_i 3^k+l} + b_{n_i,\,1,\,j_i 3^k+l} +$$
$$+3 \sum_{j_{i-1}=0}^{3^{n_{i-1}-n_i-1}-1} \Big(b_{n_{i-1},\,0,\,(3j_{i-1}+2)3^{n_i}+j_i 3^k+l} + b_{n_{i-1},\,1,\,(3j_{i-1}+2)3^{n_i}+j_i 3^k+l} +$$
$$+3 \sum_{j_{i-2}=0}^{3^{n_{i-2}-n_{i-1}-1}-1} \Big(b_{n_{i-2},\,0,\,(3j_{i-2}+2)3^{n_{i-1}}+(3j_{i-1}+2)3^{n_i}+j_i 3^k+l} +$$
$$+ b_{n_{i-2},\,1,\,(3j_{i-2}+2)3^{n_{i-1}}+(3j_{i-1}+2)3^{n_i}+j_i 3^k+l} + \ldots$$

$$\ldots+3\sum_{j_1=0}^{3^{n_1-n_2-1}-1}\Big(b_{n_1,\,0,\,(3j_1+2)3^{n_2}+\ldots+(3j_{i-1}+2)3^{n_i}+j_i3^k+l}+$$

$$+\,b_{n_1,\,1,\,(3j_1+2)3^{n_2}+\ldots+(3j_{i-1}+2)3^{n_i}+j_i3^k+l}\Big)\ldots\Big)\Big)\Big)+3^i a_{2(3^{n_1}+\ldots+3^{n_i})+l};$$

$$b_{n_i,r,l}=\sum_{j=0}^{1}(-1)^j\Bigg(\sum_{j_{i-1}=0}^{3^{n_{i-1}-n_i-1}-1}\Big(b_{n_{i-1},\,0,\,(3j_{i-1}+r+j(2-r))3^{n_i}+l}+$$

$$+b_{n_{i-1},\,1,\,(3j_{i-1}+r+j(2-r))3^{n_i}+l}+$$

$$+3\sum_{j_{i-2}=0}^{3^{n_{i-2}-n_{i-1}-1}-1}\Big(b_{n_{i-2},\,0,\,(3j_{i-2}+2)3^{n_{i-1}}+(3j_{i-1}+r+j(2-r))3^{n_i}+l}+$$

$$+b_{n_{i-2},\,1,\,(3j_{i-2}+2)3^{n_{i-1}}+(3j_{i-1}+r+j(2-r))3^{n_i}+l}+\ldots$$

$$\ldots+3\sum_{j_1=0}^{3^{n_1-n_2-1}-1}\Big(b_{n_1,\,0,\,(3j_1+2)3^{n_2}+\ldots+(3j_{i-2}+2)3^{n_{i-1}}+(3j_{i-1}+r+j(2-r))3^{n_i}+l}+$$

$$+b_{n_1,\,1,\,(3j_1+2)3^{n_2}+\ldots+(3j_{i-2}+2)3^{n_{i-1}}+(3j_{i-1}+r+j(2-r))3^{n_i}+l}\Big)\ldots\Big)\Big)\Big)+$$

$$+3^{i-1}\left(a_{2(3^{n_1}+\ldots+3^{n_i-1})+r3^{n_i}+l}-a_{2(3^{n_1}+\ldots+3^{n_i})+l}\right).$$

Доказательство. Доказательство полностью аналогично доказательству леммы 1, только используем соотношения

$$a_{k,l}=\begin{cases}a_{k+1,l}+a_{k+1,3^k+l}+a_{k+1,\,2\cdot3^k+l}, & k\notin\{n_i:i=1,\ldots,s\},\\ b_{k,0,l}+b_{k,1,l}+3a_{k+1,\,2\cdot3^k+l}, & k\in\{n_i:i=1,\ldots,s\},\end{cases}$$

учитывая, что $a_{n_1+1,l}=a_l$ и, кроме того, $a_l=0$ при $l\geqslant n$. Вторая формула получается из первой как $b_{n_i,r,l}=a_{n_i+1,\,r3^{n_i}+l}-a_{n_i+1,\,2\cdot3^{n_i}+l}$. ■

Лемма 5. Пусть $m\leqslant 2n$, а n и n_i — такие, как в теореме 3. Тогда приведение многочлена степени не выше $m-1$ по модулям $x^{2\cdot3^{n_i}}+x^{3^{n_i}}+1$, $i=1,\ldots,s$, может быть выполнено схемой

а) из $2(m-1)$ аддитивных элементов и глубины $2(n_1-n_s)+1$, причем коэффициенты остатка от деления на $x^{2\cdot3^{n_i}}+x^{3^{n_i}}+1$ вычисляются на глубине $2(n_1-n_i+1)$. Кроме того, в случае кольца характеристики 2 для сложности схемы справедлива оценка $1{,}5(m-1)$;

б) из $6n-8\cdot3^{n_1}$ аддитивных элементов, $13n/8-11\cdot3^{n_1}/4$ умножений на степени тройки и объема памяти n.

Доказательство. Все суммы $a_{k,l}$, где $k=n_s+1,\ldots,n_1$, можно вычислить одной схемой сложности не выше $m-3^{n_s+1}$, в которой глубина вычисления $a_{k,l}$ не превосходит $2(n_1+1-k)$.

Коэффициенты $b_{n_i,r,l}$ остатка от деления на $x^{2\cdot3^{n_i}}+x^{3^{n_i}}+1$ получаются присоединением к построенной схеме элементов вычитания, расположенных на глубине 1 относительно $a_{n_i+1,l}$, в количестве

$$\begin{cases}0, & m\leqslant 2\cdot3^{n_i},\\ 2(m-2\cdot3^{n_i}), & 2\cdot3^{n_i}\leqslant m\leqslant 3^{n_i+1},\\ 2\cdot3^{n_i}, & m\geqslant 3^{n_i+1}.\end{cases}$$

Число вычитаний в двух последних случаях можно также оценить сверху как $m - 3^{n_i}$. Поскольку $3^{n_i} > 2(3^{n_{i+1}} + \ldots + 3^{n_s})$, для числа вычитаний (при подходящем i) получаем оценку

$$m - 3^{n_i} + 2\left(3^{n_{i+1}} + \ldots + 3^{n_s}\right) \leqslant m - 1,$$

откуда следует оценка $2(m-1)$ для сложности всей схемы.

Если характеристика равна 2, то один из двух коэффициентов в каждой паре $b_{n_i,0,l}$, $b_{n_i,1,l}$ можно использовать при вычислении $a_{n_i,l}$ (кроме случая $i = s$). Для сложности схемы в этом случае имеем оценку $m - 3^{n_s+1} + 0{,}5(m-1) + 3^{n_s} < 1{,}5(m-1)$. П. *а* доказан.

Доказательство п. *б* достаточно провести для случая $m = n$. Перестроим схему из предыдущего пункта так, чтобы в ней явно вычислялись только те суммы $a_{k,l}$, для которых «достаточно памяти».

Обозначим $L_i = 2(3^{n_i} + \ldots + 3^{n_s})$. При любом $k = n_1, \ldots, n_s + 1$ мы явно вычисляем суммы $a_{k,0}, \ldots, a_{k,L_i-1}$, где $n_{i-1} \geqslant k > n_i$, учитывая, что $n - L_i$ прочих «ячеек памяти» отводится под хранение коэффициентов $b_{n_j,r,l}$, где $j < i$.

Все вычисление удобно разбить на этапы, нумеруя их числами от n_1 до n_s в порядке убывания. На этапе k вычисляются суммы $a_{k,0}, \ldots, a_{k,L_i-1}$, и если $k \in \{n_i : i = 1, \ldots, s\}$, то вычисляются также коэффициенты $b_{k,r,l}$, $r \in \{0, 1\}$, $l = 0, \ldots, 3^k - 1$.

Если сумма $a_{k,l}$ не вычисляется явно, то вместо нее используется правая часть формулы (обозначим ее $\psi_{k,l}$) из леммы 4, выражающая ее через известные коэффициенты $b_{n_j,q,t}$, где $n_j \geqslant k$. Заметим, что поскольку $l \geqslant L_{i+1}$ при $n_i \geqslant k > n_{i+1}$, последнее слагаемое $a_{2(3^{n_1}+\ldots+3^{n_i})+l}$ в $\psi_{k,l}$ равно нулю.

Обозначим через ρ_k число переменных в формуле $\psi_{k,l}$ (это число не зависит от l, что следует из вида формулы). Тогда прибавление (вычитание) $\psi_{k,l}$ выполняется за ρ_k аддитивных операций и i умножений на степени тройки без дополнительной памяти, где $n_i \geqslant k > n_{i+1}$.

Оценим сложность схемы. Рассмотрим этап с номером k, где $n_{i-1} > k > n_i$. Доступны коэффициенты $a_{k+1,0}, \ldots, a_{k+1,L_i-1}$, остальные $a_{k+1,l}$ выражаются формулами $\psi_{k+1,l}$. Заметим, что $L_i < 3^{n_i+1} \leqslant 3^k$. Тогда каждый из коэффициентов $a_{k,l}$, где $l < L_i$, вычисляется как $a_{k+1,l} + \psi_{k+1,3^k+l} + \psi_{k+1,2\cdot3^k+l}$ со сложностью $2\rho_{k+1}$ аддитивных операций и $i - 1$ умножений на степени тройки (умножения на степени тройки при прибавлении двух формул типа ψ можно совместить). Сложность этапа оценивается как $2L_i\rho_{k+1}$ аддитивных операций и $(i-1)L_i$ умножений на степени тройки.

Рассмотрим случай $k = n_i$. Доступны коэффициенты $a_{k+1,0}, \ldots, a_{k+1,L_i-1}$. При этом $L_i = 2 \cdot 3^k + L_{i+1}$. Сначала вычислим все $b_{k,r,l}$ по формулам $a_{k+1,r3^k+l} - a_{k+1,2\cdot3^k+l}$ при $l < L_{i+1}$ и $a_{k+1,r3^k+l} - \psi_{k+1,2\cdot3^k+l}$ при прочих l. При этом коэффициентами $b_{k,r,l}$ перезаписываются $a_{k+1,l}$, $l < 2 \cdot 3^k$. Затем вычислим $a_{k,l}$ для $l < L_{i+1}$ по формулам $3a_{k+1,2\cdot3^k+l} + b_{k,0,l} + b_{k,1,l}$. Сложность этапа оценивается как $4L_{i+1} + 2(3^k - L_{i+1})\rho_{k+1}$ аддитивных операций и $2(i-1)3^k - (2i-3)L_{i+1}$ умножений на степени тройки. Эта оценка справедлива и в случае $k = n_1$, так как можно положить $\rho_{n_1+1} = 0$.

Найдем ρ_k. Из вида формулы леммы 4 непосредственно следует, что

$$\rho_{n_i} = 2\left(3^{n_1-n_i-(i-1)} + 3^{n_2-n_i-(i-2)} + \ldots + 3^{n_{i-1}-n_i-1} + 1\right), \tag{3}$$

а $\rho_k = 3^{n_i-k}\rho_{n_i}$ при $n_{i+1} < k < n_i$.

Оценим суммарную аддитивную сложность вычислений. Сумма сложностей этапов $n_i - 1, \ldots, n_{i+1}$ не превосходит

$$C_i = 2L_{i+1}\rho_{n_i}(1 + 3 + \ldots + 3^{n_i - n_{i+1} - 2}) + 4L_{i+2} + 2(3^{n_{i+1}} - L_{i+2})3^{n_i - n_{i+1} - 1}\rho_{n_i} \leqslant$$
$$\leqslant 2\rho_{n_i}3^{n_i - n_{i+1} - 1}\left(\frac{1}{2}L_{i+1} - L_{i+2} + 3^{n_{i+1}}\right) - \rho_{n_i}L_{i+1} + 4L_{i+2} =$$
$$= 2\rho_{n_i}3^{n_i - n_{i+1} - 1}\left(2 \cdot 3^{n_{i+1}} - \frac{1}{2}L_{i+2}\right) - \rho_{n_i}L_{i+1} + 4L_{i+2} =$$
$$= \rho_{n_i}(4 \cdot 3^{n_i - 1} - L_{i+2}3^{n_i - n_{i+1} - 1} - L_{i+1}) + 4L_{i+2}.$$

При $i \geqslant 1$ последнее выражение не превосходит $4 \cdot 3^{n_i - 1}\rho_{n_i}$, если учесть, что $L_{i+1} > 3L_{i+2}$ и $\rho_{n_i} \geqslant 1$.

Для сложности всех этапов, кроме этапа n_1, используя (3), получаем оценку

$$C_1 + \ldots + C_{s-1} = 4\sum_{i=1}^{s-1} 3^{n_i - 1}\rho_{n_i} = 8\sum_{i=1}^{s-1}(3^{n_1 - i} + 3^{n_2 - (i-1)} + \ldots + 3^{n_i - 1}) <$$
$$< 8\left((3^{n_1 - 1} + 3^{n_1 - 2} + \ldots) + (3^{n_2 - 1} + 3^{n_2 - 2} + \ldots) + \ldots + (3^{n_s - 1} + 3^{n_s - 2} + \ldots)\right) < 2n.$$

Окончательно, оценивая сложность этапа n_1 как $4L_2 = 4(n - 2 \cdot 3^{n_1})$, получаем утверждение п. *б* в части аддитивной сложности.

Число умножений на степени тройки на этапах $n_i, \ldots, n_{i+1} + 1$ оценивается как

$$D_i = (n_i - n_{i+1} - 1)iL_{i+1} + 2(i-1)3^{n_i} - (2i-3)L_{i+1} =$$
$$= i(n_i - n_{i+1})L_{i+1} + (i-1)(L_i - 4L_{i+1}).$$

Сумму первых слагаемых можно оценить как

$$\sum_{i=1}^{s} i(n_i - n_{i+1})L_{i+1} \leqslant \sum_{i=1}^{s} i3^{n_i - n_{i+1} - 1}(3^{n_{i+1}+1}/2) = \frac{1}{2}\sum_{i=1}^{s} i3^{n_i} <$$
$$< \frac{1}{2}\left((3^{n_1} + 3^{n_2} + \ldots) + (3^{n_2} + 3^{n_3} + \ldots) + \ldots\right) <$$
$$< \frac{1}{2}\left(\frac{n}{2} + (n/2 - 3^{n_1}) + \frac{1}{3}(n/2 - 3^{n_1}) + \ldots\right) = \frac{n}{4} + \frac{3}{4}(n/2 - 3^{n_1}).$$

Сумму вторых слагаемых оценим грубо как

$$(L_2 - 4L_3) + 2(L_3 - 4L_4) + 3(L_4 - 4L_5) + \ldots \leqslant L_2 = n - 2 \cdot 3^{n_1}.$$

Складывая последние две оценки, завершаем доказательство п. *б*. ■

Лемма 6. Восстановление многочлена степени $n - 1$ по заданным остаткам от деления на многочлены $x^{2 \cdot 3^{n_i}} + x^{3^{n_i}} + 1$, $i = 1, \ldots, s$, может быть выполнено схемой

а) из $3{,}5n - 5 \cdot 3^{n_1}$ аддитивных элементов, $1{,}5(n - 2 \cdot 3^{n_1})$ умножений на степени тройки и глубины $2(n_1 - n_s) + s + 1$;

б) из $3{,}5n - 3^{n_1+1}$ аддитивных элементов, $1{,}5(n - 2 \cdot 3^{n_1})$ делений на 3 и объема памяти n.

Доказательство. Из леммы 4 выразим разности

$$h_{i,r,l} = a_{2(3^{n_1} + \ldots + 3^{n_i - 1}) + r3^{n_i} + l} - a_{2(3^{n_1} + \ldots + 3^{n_i}) + l}, \tag{4}$$

где $i = 1, \ldots, s-1$, $r \in \{0, 1\}$ и $l = 0, \ldots, L_i - 1$, через коэффициенты $b_{n_j,q,t}$. Заметим, что вычитаемый коэффициент равен нулю при $l \geqslant L_{i+1}$.

Для построения схемы из п. *а* вычисляем вспомогательные величины $c_{i,l}$ и $d_{i,l}$, определяемые равенствами

$$d_{1,l} = b_{n_1,0,l} + b_{n_1,1,l}, \quad c_{i,l} = \sum_{j=0}^{3^{n_{i-1}-n_i-1}-1} d_{i-1,\,j3^{n_i+1}+l}, \quad d_{i,l} = b_{n_i,0,l} + b_{n_i,1,l} + 3c_{i,2\cdot 3^{n_i}+l},$$

где $i \geqslant 2$. Искомые коэффициенты $h_{i,r,l}$ при $i \geqslant 2$ выражаются формулами $h_{i,r,l} = 3^{1-i}(b_{n_i,r,l} - c_{i,r3^{n_i}+l} + c_{i,2\cdot 3^{n_i}+l})$, а при $i = 1$ просто $h_{1,r,l} = b_{n_1,r,l}$.

Сложность вычислений, заключенная в формулах для $c_{i,l}$, $i = 2, \ldots, s$ и $l = 0, \ldots, 3^{n_i+1} - 1$, оценивается как

$$\sum_{i=2}^{s} 3^{n_i+1}(3^{n_{i-1}-n_i-1} - 1) = \sum_{i=2}^{s}(3^{n_{i-1}} - 3^{n_i+1}) < n/2 - 3(n/2 - 3^{n_1}) = 3^{n_1+1} - n.$$

Сложность, заключенная в формулах для $d_{i,l}$, $i = 1, \ldots, s-1$ и $l = 0, \ldots, 3^{n_i} - 1$, оценивается как $3^{n_1} + 2(3^{n_2} + \ldots + 3^{n_{s-1}}) < n - 3^{n_1}$ аддитивных операций и $3^{n_2} + \ldots + 3^{n_{s-1}} < n/2 - 3^{n_1}$ умножений на 3. Для завершения вычисления $h_{i,r,l}$ нужно выполнить еще $4(3^{n_2} + \ldots + 3^{n_s}) = 2(n - 2 \cdot 3^{n_1})$ аддитивных операций и $n - 2 \cdot 3^{n_1}$ умножений на степени тройки.

Без труда проверяется, что $c_{i,l}$ при этом вычисляется на глубине $2(n_1 - n_i) - 1$ и, следовательно, $h_{i,r,l}$ — на глубине $2(n_1 - n_i) + 2$.

По набору $h_{i,r,l}$ несложно восстанавливаются коэффициенты многочлена $a(x)$. Коэффициенты $a_{2(3^{n_1}+\ldots+3^{n_{i-1}})+r3^{n_i}+l}$ совпадают с $h_{i,r,l}$ при $l \geqslant L_{i+1}$. В частности, в случае $i = s$ известны все коэффициенты. Это позволяет последовательно в порядке убывания i определить недостающие коэффициенты $a_{2(3^{n_1}+\ldots+3^{n_{i-1}})+r3^{n_i}+l}$, $l = 0, \ldots, L_{i+1} - 1$, из формул (4) со сложностью

$$L_s + L_{s-1} + \ldots + L_2 = 2(3^{n_2} + 2 \cdot 3^{n_3} + 3 \cdot 3^{n_4} + \ldots) < 1{,}5(n - 2 \cdot 3^{n_1})$$

аддитивных операций. Глубина этих вычислений не превосходит $s - 1$.

Складывая все оценки, получаем утверждение п. *а*.

Для доказательства п. *б* построим схему, последовательно в порядке убывания i преобразующую каждый коэффициент $b_{n_i,r,l}$ по формуле леммы 4 в соответствующую разность $h_{i,r,l}$.

Несложно видеть, что преобразование каждого коэффициента $b_{n_i,r,l}$ выполняется за τ_i аддитивных операций и $i - 1$ делений на 3, где τ_i — число коэффициентов $b_{n_j,q,t}$ в правой части второй формулы леммы 4. Непосредственно проверяется, что

$$\tau_i = 4\left(3^{n_1-n_i-(i-1)} + 3^{n_2-n_i-(i-2)} + \ldots + 3^{n_{i-1}-n_i-1}\right).$$

Аддитивную сложность вычисления разностей $h_{i,r,l}$ теперь можно оценить как

$$\sum_{i=2}^{s} 2 \cdot 3^{n_i}\tau_i = \sum_{i=2}^{s} 8\left(3^{n_1-(i-1)} + 3^{n_2-(i-2)} + \ldots + 3^{n_{i-1}-1}\right) <$$
$$< 8\left((3^{n_1-1} + 3^{n_1-2} + \ldots) + (3^{n_2-1} + 3^{n_2-2} + \ldots) + \ldots + (3^{n_s-1} + 3^{n_s-2} + \ldots)\right) < 2n.$$

Число делений на 3 оценивается как

$$\sum_{i=2}^{s} 2(i-1)3^{n_i} = 2\left(3^{n_2} + 2 \cdot 3^{n_3} + 3 \cdot 3^{n_4} + \ldots\right) < 1{,}5(n - 2 \cdot 3^{n_1}).$$

Заключительная часть схемы такая же, как в п. *а*. ■

ЛИТЕРАТУРА

1. *Van der Hoeven J.* The truncated Fourier transform and applications // Proc. ISSAC 2004 (Santander, Spain). NY: ACM Press, 2004. P. 290–296.
2. *Harvey D. and Roche D. S.* An in-place truncated Fourier transform and application to polynomial multiplication // Proc. ISSAC 2010 (Munich, Germany). NY: ACM Press, 2010. P. 325–329.
3. *Schönhage A.* Schnelle multiplikation von polynomen über körpern der charakteristik 2 // Acta Inf. 1977. V. 7. P. 395–398.
4. *Cantor D. and Kaltofen E.* On fast multiplication of polynomials over arbitrary algebras // Acta Inf. 1991. V. 28. No. 7. P. 693–701.
5. *Bernstein D. J.* Fast multiplication and its applications // Algorithmic Number Theory, MSRI Publ. 2008. V. 44. P. 325–384.
6. *Von zur Gathen J. and Gerhard J.* Modern computer algebra. Cambridge: Cambridge University Press, 1999. 768 p.
7. *Лупанов О. Б.* Асимптотические оценки сложности управляющих систем. М.: Изд-во Моск. ун-та, 1984. 138 с.
8. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986. 384 с.
9. *Cooley J. and Tukew J.* An algorithm for the machine calculation of complex Fourier series // Math. Comp. 1965. V. 19. P. 297–301.
10. *Schönhage A.* Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients // Proc. EuroCAM-82 (Marseille, France). LNCS. V. 144. Berlin; Heidelberg; NY: Springer, 1982. P. 3–15.
11. *Сергеев И. С.* Регуляризация некоторых оценок сложности умножения многочленов // Материалы VII молодежной научной школы по дискретной математике и ее приложениям (Москва, 2009 г.). Ч. II. М.: Изд-во Института прикладной математики РАН, 2009. С. 26–32.
12. *Van der Hoeven J.* Notes on the truncated Fourier transform // Tech. Report. Univ. Paris-Sud, Orsay, France, 2005.
13. *Crandall R. and Fagin B.* Discrete weighted transforms and large-integer arithmetic // Math. Comput. 1994. V. 62. P. 305–324.
14. *Mateer T.* Fast Fourier algorithms with applications // Ph. D. Thesis. Clemson University, 2008.
15. *Гашков С. Б., Сергеев И. С.* Алгоритмы быстрого преобразования Фурье // Дискретная математика и ее приложения. Ч. V. М.: Изд-во Института прикладной математики РАН, 2009. С. 3–23.
16. *Гашков С. Б., Сергеев И. С.* О сложности и глубине булевых схем для умножения и инвертирования в некоторых полях $GF(2^n)$ // Вестник МГУ. Сер. 1. Математика. Механика. 2009. № 4. С. 3–7.

ДИСКРЕТНЫЕ МОДЕЛИ РЕАЛЬНЫХ ПРОЦЕССОВ

DOI 10.17223/20710410/14/10

УДК 681.3; 519.711

ПОПУЛЯЦИИ ВЗАИМОДЕЙСТВУЮЩИХ АВТОМАТОВ

Ю. В. Березовская, В. А. Воробьев

Северный (Арктический) федеральный университет им. М. В. Ломоносова, г. Архангельск, Россия

E-mail: vva100@atnet.ru

Описана модель коллективного поведения автоматов — популяция автоматов. Для моделирования динамики популяции применяется каузальная сеть Петри. Позициям сети соответствуют состояния автоматов. Маркировка сети задаёт число автоматов, находящихся в соответствующих состояниях. Переходы отображают события, возникающие в результате совместных действий элементов популяции. На переходах сети заданы их вероятности. Это позволяет составить систему дифференциальных уравнений, которые описывают динамику среднего числа автоматов в позициях при выполнении логических условий, заданных сетью Петри. Система решается численно, методом потактного компьютерного моделирования.

Ключевые слова: *популяция автоматов, каузальная сеть, сеть Петри, динамика средних, моделирование.*

Введение

Популяция автоматов — это система из $N \geqslant 2$ взаимодействующих автоматов (не обязательно одинаковых), в которых смена состояний отдельного автомата обусловлена состояниями некоторых других автоматов. А именно, состояния «воздействующих» автоматов влияют на «изменяемые» автоматы и переводят их в новые состояния, причём способ передачи воздействий и связи между автоматами не рассматриваются. Предполагается, что: 1) все N_i автоматов в состоянии i равномерно распределены по системе, и вероятность найти в любом месте автомат в состоянии i равна N_i/N (сильное перемешивание); 2) все потоки событий в системе ординарные.

Популяции автоматов пригодны для исследования разнообразных массовых объектов: биологических, экономических и технических систем, параллельных программ [1]. С этой целью автоматы должны иметь стохастические характеристики — вероятности переходов в каждом такте. Поскольку число состояний популяции чрезвычайно велико, вычисления проводятся не для всех состояний популяции, а для среднего числа автоматов в различных состояниях. Таким образом, полученный случайный процесс представляет динамику популяции «в среднем».

Трудность состоит в том, что в известном методе динамики средних [2] все компоненты независимы друг от друга. Между тем основное свойство, которое влияет на поведение популяции, — взаимодействие между автоматами. Следует как-то учесть это в методе динамики средних. Отсутствие метрики в популяции позволяет исследовать такие случайные системы, используя достижения теории параллельных процес-

сов [1, 3]. В настоящей работе развиты идеи из [1, 3] с использованием теории сетей Петри и марковских процессов.

1. Каузальная сеть

1.1. Определение каузальной сети

Каузальная сеть (К-сеть) — это маркированная сеть Петри, в которой для каждого перехода задана интенсивность события перехода как функция от маркировки входных позиций перехода. Вид этих функций зависит от предметной области и задаётся отдельно в каждом конкретном случае. Потоки событий переходов простейшие, т. е. стационарные (интенсивности меняются медленно), ординарные и без последействия.

Определение 1. *Каузальная сеть* — это двудольный граф $G = \langle Q, D, In, Out, M, R\rangle$, где

- $Q = \{q_i : i = 0, 1, \ldots, n\}$ — множество позиций, соответствующее объединению множеств состояний всех автоматов;
- $D = \{d_j : j = 1, 2, \ldots, m\}$ — множество переходов автоматов из состояния в состояние;
- In — функция предшествования, которая каждой паре (q_i, d_j) ставит в соответствие неотрицательное число k_{ij} — вес дуги из позиции q_i в переход d_j; если соответствующей дуги нет, то $k_{ij} = 0$;
- Out — функция следования, которая каждой паре (d_j, q_i) ставит в соответствие неотрицательное число k_{ji} — вес дуги из перехода d_j в позицию q_i; если соответствующей дуги нет, то $k_{ji} = 0$;
- $M_t = \{N_{it} : i = 1, 2, \ldots, n\}$ — вектор маркировки, компоненты которого задают число автоматов, находящихся в момент времени t в каждом из состояний множества Q;
- $R = \{p_j(M_t(^*d_j)) : j = 1, \ldots, m\}$ — вектор-функция интенсивностей переходов, определяющая среднее число срабатываний перехода d_j в течение одного такта или число таких срабатываний в единицу времени, зависящее от маркировки множества $^*d_j$ — входных позиций перехода.

Функция $p_j(M_t(^*d_j))$ для перехода d_j в простейшем случае является линейной. В этом случае если $^*d_j$ содержит несколько позиций, то находится позиция $q_i \in {}^*d_j$ с минимальной маркировкой $N_{i\min}$, и тогда интенсивность перехода d_j равна $p_j N_{i\min}$, где p_j — вероятность срабатывания перехода d_j для одного элемента системы. В более сложных случаях интенсивность перехода может задаваться нелинейной функцией. Например, если для взаимодействия двух атомов в растворе необходимо их столкновение, то вероятность такого события пропорциональна произведению плотностей этих атомов.

Позиция $q_0 \in Q$ называется внешней, имеет сколь угодно большое или единичное (если надо) значение маркера N_0, не меняет его при переходах и может не изображаться на рисунке графа. Состояния автоматов и позиции множества $\{q_i : i = 1, \ldots, n\}$ назовём собственными. Граф G изображает причинно-следственные связи между состояниями автоматов и интенсивности этих связей.

В отличие от канонической сети Петри, множество весовых коэффициентов дуг К-сети — это положительные действительные числа, приписанные входным и выходным дугам j-го перехода — k_{ij} или k_{ji} соответственно. Точно так же будем допускать действительные числа в качестве маркеров N_i для позиций. Это позволит маркиро-

вать сеть вероятностями состояний автоматов и вообще избавиться от целых чисел. В таких случаях будем считать популяцию счётным множеством.

1.2. Описание K-сети

Описание K-сети состоит в описании двух ее частей:

1) статическая часть — маркировка M_0 в начальный момент времени $t = 0$;
2) динамическая часть — описание переходов.

Каждый переход d_j описывается следующим образом:

1) перечислением элементов множества ${}^*d_j$ с коэффициентами k_{ij};
2) перечислением элементов множества d_j^* с коэффициентами k_{ji};
3) интенсивностью $p_j(M_t({}^*d_j))$;
4) типом перехода.

В общем случае описание перехода — это выражение вида

$$ {}^*d_j > d_j^* \ : \ p_j(M_t({}^*d_j)) \ : \ \text{тип}. $$

Внешнее состояние в описании не присутствует, так что допустимы переходы с неполной левой или правой частью.

2. Каузальные модели популяций

Как и сеть Петри, K-сеть может использоваться для моделирования сложных систем, состоящих из множества взаимодействующих элементов — популяций. Абстрагируясь от природы популяции, будем называть её элементы автоматами. Динамическую модель, построенную на основе K-сети, будем называть *каузальной моделью* (*K-моделью*). Этот термин подчёркивает динамический характер и модельную функцию K-сети.

2.1. K-модель мобилизации

Проще всего пояснить основные идеи популяционного моделирования на примере. Рассмотрим замкнутую «популяцию», которую представим как популяцию автоматов-особей с двумя состояниями L (*live*, живая) и D (*dead*, мёртвая). Вероятность гибели особи (перехода из L в D) в течение одного такта равна p. Живая особь может поглотить одну мёртвую и разделиться на двух живых или, иначе говоря, оживить мёртвую особь. Вероятность этого события в течение одного такта равна q. Временем на поглощение и деление можно пренебречь по сравнению со временем поиска добычи. В популяции определены два перехода: гибель, независимая от состояния популяции, и восстановление, возможное только при наличии одной живой и одной мёртвой особи.

При описании K-моделей общего вида мы использовали обозначение N_i для числа автоматов, находящихся в состоянии i. Чтобы не загромождать запись уравнений буквой N, её можно опускать, и тогда имя состояния будет обозначать и число автоматов в этом состоянии, как это показано на рис. 1 и принято в элементарной алгебре. С той же целью экономии обозначений будем записывать функцию времени $X(t)$ в виде X_t или вообще без индекса.

Предложенная K-модель имеет множество различных интерпретаций. Это и модель заселения мест (М) экологической ниши живыми (Ж) организмами, и модель замены отказавших узлов системы (М) исправными узлами (Ж), это и модель спасения раненых на поле боя живыми бойцами, это, наконец, модель мобилизации призывников (М) теми, кто уже призван в армию (Ж), причём призывники активно избегают призыва — «умирают». Последняя интерпретация позволяет называть описанную популяцию

моделью мобилизации, как это принято в экономической литературе. Автоматы этой модели будем называть особями.

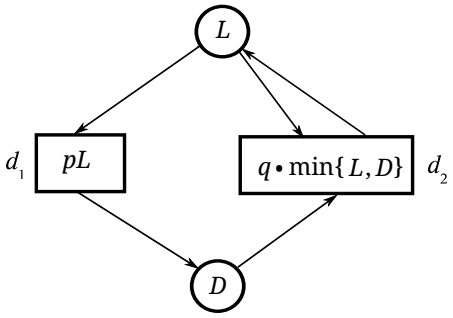


Рис. 1. Каузальная сеть линейной модели мобилизации

Зададим модель мобилизации с линейными интенсивностями переходов. К-сеть для неё показана на рис. 1. Описание этой популяции:

$L_0 = L_{\text{beg}}, D_0 = D_{\text{beg}}$;

$L > D : pL$: линейный;

$L : D > 2L : q \cdot \min\{L, D\}$: линейный.

Интерпретация предложенного описания популяции зависит от единицы времени. Если единица времени достаточно мала, то $p \ll 1$ и $q \ll 1$ — вероятности срабатывания переходов в течение такта, а pL и $q \cdot \min\{L, D\}$ — среднее число автоматов, изменяющих состояние за такт. Если единица времени велика, то p и q — интенсивности переходов одного автомата, а величины pL или $q \cdot \min\{L, D\}$ — это интенсивности переходов на всём множестве автоматов, готовых к переходу.

2.2. Динамика К-модели

Граф, который мы назвали К-сетью, — это статическая модель популяции автоматов. Она задаёт только причинно-следственные связи между элементами системы автоматов. Динамическая модель популяции — К-модель — определяется функционированием К-сети. Функционирование К-сети подобно несущей сети Петри с учётом интенсивностей переходов, а именно: переход d_j срабатывает только тогда, когда маркировка его входа такова, что $M(^*d_j) \geqslant In(d_j)$. Один переход К-сети описывает множество допустимых изменений состояний автоматов, заданное интенсивностью. В нашем примере $L \geqslant 1$ и $\min\{L, D\} \geqslant 1$. При срабатывании перехода маркировка на его входе уменьшается, а на выходе увеличивается согласно интенсивностям. В нашем примере переход $L > D : pL$ уменьшает маркировку L и увеличивает маркировку D на pL, а переход $L : D > 2L : q \cdot \min\{L, D\}$: линейный уменьшает D на $q \cdot \min\{L, D\}$ и увеличивает L на $q \cdot \min\{L, D\}$.

Пусть $N_t = \sum_{i=1}^{n} N_{it}$ — численность популяции в момент t, $P_{it} = N_{it}/N_t$ — доля автоматов, находящихся в момент t в состоянии N_i. При $N_t \to \infty$ величина P_{it} — это вероятность пребывания автомата в i-м состоянии. Вектор-функция $P_t = \{P_{it} : i = 1, \ldots, n\}$ задаёт динамику популяции в среднем, или её *поведение*.

2.3. Уравнения динамики средних для замкнутой популяции

Пусть $\mathbf{R} = d_1, d_2, d_3, d_4, \ldots, d_s$ — допустимая последовательность переходов К-сети. Рассмотрим только тот случай, когда сеть такова, что каждый из её переходов неоднократно найдётся в последовательности $\mathbf{R}$ при достаточно длительном наблюдении.

Вектор R длины m, в котором j-я компонента есть число вхождений перехода d_j в последовательность $\mathbf{R}$, называется *характеристикой* последовательности $\mathbf{R}$ или *характеристикой динамики* К-сети. Взяв период наблюдения К-сети за единицу времени, отождествим характеристику R и вектор-функцию $R = \{p_j(M_t(^*d_j)) : j = 1, \ldots, m\}$ интенсивностей переходов. Пусть теперь R — вектор-столбец длины m, характеризующий динамику (число переходов) К-сети за единичное время. Тогда вектор-столбец $R\Delta t$ — характеристика динамики за время Δt.

Функции In и Out для К-сети задаются $(n \times m)$-матрицами инциденций, где строкам соответствуют позиции сети (кроме внешней), а столбцам — переходы. Элементы этих матриц — значения k_{ij} и k_{ji} соответственно. Наглядно это демонстрирует табл. 1.

Таблица 1

Матричное описание К-модели мобилизации

In	d_1	d_2
L	1	1
D	0	1

Out	d_1	d_2
L	0	2
D	1	0

D	d_1	d_2
L	-1	1
D	1	-1

$R = \{p_j(M_t(^*d_j)) : j = 1, 2\}$
pL
$q \cdot \min\{L, D\}$

Динамика К-сети (К-модель) задаётся матричным уравнением К-сети, аналогичным фундаментальному уравнению сети Петри:

$$\Delta M = Out \cdot R\Delta t - In \cdot R\Delta t = (Out - In) \cdot R\Delta t = D \cdot R\Delta t,$$

где $Out - In = D$ — оператор изменения маркировки сети, или D-оператор (от *Derivative* — производная); ΔM — вектор-столбец длины n — изменение маркировки сети при срабатывании любой допустимой последовательности переходов с характеристикой $R\Delta t$; $\cdot$ — матричное умножение.

Устремим Δt к нулю и заменим его на дифференциал dt. Тогда и ΔM станет величиной более высокого порядка малости по отношению к M: $\Delta M = \mathrm{o}(M)$, и её тоже можно заменить на dM. Теперь можно перейти к дифференциальному виду и записать дифференциальное уравнение динамики средних К-сети в векторном виде:

$$\frac{dM}{dt} = D \cdot R.$$

Приравняв производную нулю, получим уравнение для стационарного режима:

$$D \cdot R = 0.$$

Кроме этих уравнений, для замкнутых популяций справедлива нормировка

$$\sum_{i=1}^{n} P_{it} = 1 \text{ или, что то же, } \sum_{i=1}^{n} N_{it} = N.$$

Особенностью полученных линейных систем является тот факт, что уравнения в них могут быть расщепляемыми.

В нашем примере с популяцией особей имеем систему

$$\begin{cases} \dfrac{dL}{dt} = q \cdot \min\{L, D\} - pL, \\ \dfrac{dD}{dt} = pL - q \cdot \min\{L, D\}, \end{cases}$$

где каждое из уравнений расщепляется на два различных уравнения, действующих в зависимости от соотношения значений L и D. Поскольку популяция замкнута и $N = L + D = \text{const}$, эти два дифференциальных уравнения сводятся к одному

$$\frac{dL}{dt} = q \cdot \min\{L, N - L\} - pL,$$

которое расщепляется на два уравнения и интегрируется.

Если в K-сети найдётся m переходов, меняющих интенсивности как функция $\min\{x, y\}$, то система из n уравнений даст $O(2^m)$ вариантов систем в соответствии с различными наборами значений функций $\min\{x, y\}$. При этом популяция может иметь несколько различных стационарных состояний.

2.4. Классификация K-моделей

K-модели классифицируются по нескольким критериям.

Во-первых, следует выделить линейные и нелинейные модели взаимодействий. Эти модели отличаются видом функций интенсивностей переходов $p_j(M_t(^*d_j))$.

Во-вторых, о сложности модели можно судить по числу состояний и автоматов, участвующих в одном акте взаимодействия и в переходе.

В-третьих, K-модели различаются по составу моделируемой популяции. Все эти различия решающим образом влияют на моделирование популяций и, следовательно, требуют более содержательного описания.

Линейные и нелинейные взаимодействия

Линейные взаимодействия таковы, что интенсивности $p_j(M_t(^*d_j))$ пропорциональны минимальным маркерам N_{it} в $M_t(^*d_j)$. Линейные модели адекватны в тех случаях, когда интенсивность перехода является имманентным свойством каждого отдельного автомата. Так, в нашей гипотетической популяции способность восстанавливать мертвую особь присуща только части особей, находящихся в состоянии L, но если уж эта способность есть, то она реализуется, и эта реализация единственна в данном такте.

Иной способ понимания свойства линейности K-модели — положить, что взаимодействия между автоматами являются дальнодействующими, т. е. взаимосвязи (отношение соседства) между элементами системы можно представить полным графом. Рассмотрим, например, боевое столкновение двух воюющих сторон 1 и 2 в чистом поле, когда каждый «живой» боец видит каждого «живого» противника и может поразить его из дальнобойной винтовки, т. е. перевести его в состояние «мёртвый». Пусть L_1 и L_2 — численности «живых» с обеих воюющих сторон в данный момент времени, p и q — вероятности попадания в противника у стрелков соответствующих сторон, D — численность «мёртвых». Общая численность мест, занятых всеми живыми и мёртвыми солдатами, равна $L_1 + L_2 + D$.

Линейная K-сеть такого столкновения показана на рис. 2. В ней два перехода d_1, d_2 и три позиции L_1, L_2, D.

Маркировка M_t этой K-сети задаётся тройкой $\{L_1, L_2, D\}$, а интенсивности уничтожения противников (число врагов, убитых каждой стороной в каждом такте времени) вычисляются по следующим формулам:

$$p_1(M_t(^*d_1)) = p_1(L_1, L_2) = p \cdot \min\{L_1, L_2\}, \quad p_2(M_t(^*d_2)) = p_2(L_1, L_2) = q \cdot \min\{L_1, L_2\},$$

где $p_1(L_1, L_2)$ — интенсивность перехода d_1 из L_2 в D; $p_2(L_1, L_2)$ — интенсивность перехода d_2 из L_1 в D. Член $\min\{L_1, L_2\}$ отражает тот очевидный факт, что число одновременных выстрелов не может превышать ни числа видимых противников, ни числа

стреляющих. Этот член может появиться в сети автоматически. Таким образом, *чтобы задать линейные переходы, достаточно указать только их вероятности* (в данном случае p и q).

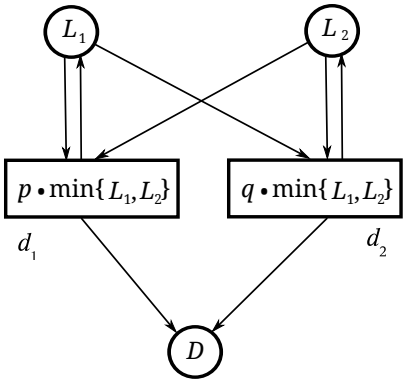


Рис. 2. Линейная K-сеть боевого столкновения

Нелинейные популяции таковы, что $p_j(M_t(^*d_j))$ зависит от степеней и произведений маркировок N_{it} из $M_t(^*d_j)$. Нелинейная модель адекватна, если интенсивность перехода пропорциональна вероятности возможных благоприятных сочетаний, причём все эти сочетания и соответствующие переходы могут быть реализованы в одном такте с заданной вероятностью. Так, в модели мобилизации следовало бы учесть вероятность «встречи» одной живой и одной мёртвой особи, точнее, попадания мёртвой особи в поле зрения живой. Тогда интенсивность восстановления пропорциональна величине $L \cdot M/N$.

По аналогии с линейной популяцией можно представить нелинейный случай, как *близкодействующее* взаимодействие автоматов. При этом взаимодействие происходит на некотором конечном расстоянии. Это расстояние можно задать коэффициентом, имеющим смысл эффективного сечения или зоны, в которой только и возможно взаимодействие. Будем называть эту зону *областью действия* для действующего автомата и *областью восприимчивости* для автомата, подвергающегося воздействию. В случае взаимодействия двух автоматов область действия и область восприимчивости совпадают и могут называться просто *областью взаимодействия*.

Так, в модели боевого столкновения следует полагать, что оружие (например, меч или копьё) не обладает бесконечной дальностью поражения. Для обеих воюющих сторон дальность поражения, т. е. область действия, может быть различной и задаётся коэффициентами k_1 и k_2. Теперь для того, чтобы задать интенсивность взаимодействия, мало задать вероятности p и q. Необходимо домножить их на коэффициенты k_1 и k_2 и на отношение $(N_1 \cdot N_2)/N$, которое задаёт интенсивность встречи двух близкодействующих бойцов. Значение $(N_1 \cdot N_2)/N$ можно получить автоматически, а величины pk_1 и qk_2 следует задать «руками». Ясно, что эти задаваемые величины теперь уже не обязаны быть вероятностями и могут превышать единицу. Более того, и итоговые интенсивности взаимодействий $(pk_1N_1N_2)/N$ и $(qk_2N_1N_2)/N$ превышают единицу, т. е. являются не вероятностями, а интенсивностями — средним числом переходов в такте. Нелинейная K-сеть боевого столкновения показана на рис. 3.

Простые популяции

Сложность популяции задаётся числом взаимодействующих автоматов в каждом переходе. Полезно выделить наиболее простые случаи.

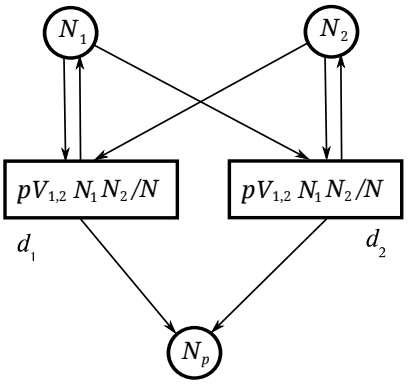


Рис. 3. Нелинейная K-сеть боевого столкновения

Простые популяции таковы, что взаимодействуют только пары автоматов. Каждый переход инициируется одним или двумя состояниями, а простой D-оператор состоит из нулей и единиц, может быть с разными знаками. В столбцах матрицы D находится не более двух единиц, как в примере с «особями».

Простейшая популяция — это простая линейная популяция.

Автоматная популяция — это простейшая популяция, в которой все элементы являются автономными вероятностными автоматами. Этот случай относится к известному методу динамики средних [2].

Простые растворы и смеси

Автоматизируем подсчёт интенсивностей переходов в простых K-моделях популяций. Будем полагать, что моделируемые популяции *сильно перемешаны*, т. е. каждый автомат в любом состоянии i может оказаться в любом месте с одинаковой вероятностью N_i/N независимо от расстояния до него.

Что касается *линейных* взаимодействий, то формулы для интенсивностей переходов, представленные в п. 2.1, в комментариях не нуждаются и их вычисление всегда можно провести автоматически. Однако *нелинейные* взаимодействия допускают различные интерпретации и способы вычисления интенсивностей. Рассмотрим только простые нелинейные популяции, поскольку результаты легко обобщаются на нелинейные популяции общего вида. Выделим два типа простых нелинейных популяций: *раствор* и *смесь*.

Популяция типа *раствор* представляет собой равномерное размещение взаимодействующих автоматов во множестве мощности N мест, большинство из которых — места, не содержащие взаимодействующих автоматов. Примерами таких популяций в биологии являются популяции белых медведей и птиц, которые не собираются в одном месте для воспроизводства. Проблемой таких популяций является низкая вероятность встречи, если ареал расселения велик по сравнению с численностью популяции. Вероятность встречи самца и самки может оказаться меньше, чем вероятность смерти отдельной особи, и популяция вымирает из-за малой численности в большом ареале расселения.

Популяция типа *смесь* собирается для взаимодействия в ограниченной области, как, например, птичьи базары, пассажиры в трамвае или покупатели в магазине. Так что вероятность взаимодействия данной пары не зависит от общего числа мест, а зависит только от количества автоматов, собравшихся вместе для взаимодействий.

Ясно, что в сложных системах различные взаимодействия могут происходить поразному: и линейно, и как в растворе, и как в смеси. Поэтому тип популяции мо-

жет быть (и чаще всего бывает) *смешанным*, а характеристики — линейный, раствор и смесь — относятся далее не ко всей популяции, а отдельно к каждому взаимодействию. Рассмотрим интенсивности переходов для различных типов взаимодействий. Для экономии слов будем употреблять термины «раствор» и «смесь» достаточно вольно. Например, будем говорить о взаимодействиях или переходах в растворе или в смеси.

Пусть тройка (i, j, k) обозначает переход автомата из состояния j в состояние k под воздействием автомата, находящегося в состоянии i, а $p(i, j, k)$ — вероятность такого перехода. При этом число автоматов в состоянии i не изменяется, если $i \neq k$. Будем полагать, что каждый автомат занимает ровно одно место в пространстве, где «живёт» популяция. Общее число таких мест равно N. При этом часть мест может быть пустой, т. е. популяция как бы «растворена» во множестве мощности N. Пусть теперь:

— N_i — число автоматов в состоянии i;

— N_{ijk} — общая интенсивность перехода (i, j, k), т. е. число автоматов, совершающих этот переход в каждом такте моделирования;

— V_{ij} — область взаимодействия для перехода (i, j, k) — число мест в окрестности состояния i (или j), в которые должны попасть оба автомата i и j, чтобы взаимодействие состоялось. Ясно, что области взаимодействия состояний i и j одинаковы, т. е. $V_{ij} = V_{ji}$; следует также иметь в виду, что области V_{ij} должны быть порядка одного места — $V_{ij} \approx 1$, чтобы можно было пренебречь их пересечениями;

— $K(i, j, k) = p(i, j, k)V_{ij}$ — интенсивность перехода (i, j, k) для одной пары автоматов, попавших в область взаимодействия размера V_{ij} в состояниях i и j.

Задача состоит в вычислении общей интенсивности перехода N_{ijk} для различных типов взаимодействия. В растворе плотность автоматов в состоянии j равна $p_j = N_j/N$, общий объём области взаимодействия равен $V_{ij} \cdot \min\{N_i, N_j\}$. Пусть $N_i = \min\{N_i, N_j\}$. Тогда число взаимодействующих автоматов равно $V_{ij} \cdot \min\{N_i, N_j\} \cdot p_j = (V_{ij}N_iN_j)/N$. Отсюда следует, что для взаимодействия в растворе

$$N_{ijk} = p(i, j, k)\frac{V_{ij}N_iN_j}{N} = K(i, j, k)\frac{N_iN_j}{N}.$$

Точно те же рассуждения для взаимодействия в смеси приводят к выводу, что значение N следует заменить на сумму $(N_i + N_j)$, поскольку в этом случае оба взаимодействующих множества автоматов «смешаны друг с другом», а прочие автоматы роли не играют. Так, для заражения гриппом больные и здоровые люди должны встречаться в каком-нибудь тесном месте, например в офисе, чтобы обменяться вирусом, а прочие обстоятельства — размер города или всей страны — роли не играют. Итак, для взаимодействия в смеси

$$N_{ijk} = p(i, j, k)\frac{V_{ij}N_iN_j}{(Ni + Nj)} = K(i, j, k)\frac{N_iN_j}{(N_i + N_j)}.$$

Другой способ получения этих же формул состоит в следующем. Коль скоро $p_j = N_j/N$ — плотность состояния j в объёме N, то вероятность встречи двух состояний i и j в области взаимодействия V_{ij}, близкой к единице объёма, равна $(N_iN_j)/N^2$. Тогда во всём объёме N состоится $(N_iN_j)/N$ встреч, вероятность события-перехода в момент встречи равна $p(i, j, k)$, а интенсивность перехода $K(i, j, k) = p(i, j, k)V_{ij}$, где V_{ij} — коэффициент, учитывающий, что взаимодействие происходит на некотором расстоянии между автоматами. Интенсивность перехода $K(i, j, k)$ может превышать 1, и обычно она является эмпирическим коэффициентом. Очевидно, что величина $K(i, j, k)$ для одной пары автоматов — это параметр взаимодействия, который следует задать

«руками», а остальная часть формулы для интенсивности перехода может быть вычислена автоматически, если указать тип перехода — линейный, раствор или смесь.

2.5. Инвариантность К-моделей

Поведение популяции мы определили как вектор-функцию $P_t = \{P_{it} : i = 1, \ldots, n\}$ вероятностей состояний, зависящую от времени. Очевидно, что поведение любой популяции не зависит ни от единиц измерения времени, ни от её точной численности. Поэтому будем говорить, что две вектор-функции P_t и P'_t инвариантны и задают одно и то же поведение, если одна из них переходит в другую при смене масштаба времени и/или численности. Такие преобразования масштабов назовём масштабированием. Соответственно модели, порождающие инвариантное поведение, эквивалентны. Рассмотрим условия, при которых такая инвариантность возможна.

Обратим внимание, что матрица D имеет как положительные, так и отрицательные элементы. Положительные элементы задают прирост ($+\Delta N_j$) числа автоматов в состоянии j, отрицательные — убыль числа автоматов ($-\Delta N_j$) в этом состоянии. Мы предполагали ранее, что эти приращения меньше, чем общее число автоматов: $|\Delta N_j| \ll N_j$ для всех $j = 1, \ldots, m$, чем обеспечивается достаточная точность вычислений. При этом графики функций в переходном режиме достаточно гладкие, т. е. не имеют точек излома и смены направления роста N_{jt} для всех $j = 1, \ldots, m$. Невнятный термин «достаточно» станет понятен при анализе конкретных моделей популяций. Пока предположим, что для данной К-модели существует максимальный вектор $R_{\max} = \{p_{j\max}(M_t(^*d_j)) : j = 1, \ldots, m\}$ интенсивностей переходов, такой, что выполняется $|\Delta N_j| \ll N_j$ и обеспечивается достаточная точность вычислений и гладкость графиков N_{jt} для всех $j = 1, \ldots, m$.

К-модели имеют следующие свойства.

Масштабируемость

К-модели масштабируемы. Это означает следующее:

1) вектор-функцию $R_{\max}$ интенсивностей переходов можно умножать на число $0 < r < 1$ без изменения стационарного поведения популяции;
2) компоненты вектора M_0 можно одновременно умножать на произвольное действительное число $s > 0$ без изменения стационарного поведения популяции.

Нормировка

Если в замкнутой К-модели сумма всех (кроме N_0) компонент вектора M_t равна 1, то $M_t = P_t$ — вектор вероятностей её собственных состояний.

Эти свойства справедливы в силу исходного предположения, что все потоки событий в системе простейшие, т. е. плотность вероятности события и интенсивность потока событий связывает соотношение $dp = \lambda dt$. Другим, уже чисто математическим, основанием для утверждений об инвариантности и вероятностях состояний является тот факт, что изменение масштаба времени или численности популяции эквивалентно умножению матриц In, Out, D, R и M на одно и то же число, что не меняет характера решения соответствующих уравнений (кроме, разумеется, масштабов).

Для вектор-функции интенсивностей переходов выполняется условие $R \ll M_t$, только если длительность такта Δt и соответственно компоненты вектора R достаточны малы. В этом случае модель популяции допускает компьютерное моделирование *методом* Δt, где за величину приращения времени Δt берётся один такт, и каждое новое значение N_{jt} вычисляется потактно. Это не что иное, как численное интегриро-

вание дифференциальных уравнений динамики средних для популяции. Такая модель называется *синхронной*.

Масштабируемость имеет место только для стационарного состояния K-модели. Дело в том, что если потактные изменения ΔN_{jt} будут слишком велики, то возникнет эффект малой точности вычислений в переходном режиме. В результате процесс моделирования может «проскакивать» малые изменения N_{jt}. Модель станет слишком грубой, и это исказит график функции N_{jt}, нарушит его гладкость. В таких случаях компоненты вектора R могут и превышать значения компонент вектора M_t, как в уравнениях Колмогорова — Чепмена или в уравнениях динамики средних с интенсивностями потоков событий. Соответственно такие модели называются *асинхронными* и могут превращаться в синхронные умножением интенсивностей на подходящее число $0 < r < 1$ согласно утверждению об инвариантности. Асинхронная модель не может быть непосредственно реализована методом Δt, однако она имеет свои методологические преимущества.

Конвертируемость

Умножением R на подходящее число $r > 0$ можно превращать синхронную модель популяции в асинхронную и наоборот.

Могут ли синхронные модели, реализуемые методом Δt, иметь компоненты вектора R, превышающие 1? Такие модели могут возникать, например, для растущих популяций.

Соразмерность

Соразмерность — ещё одно свойство вектор-функции $P_t = \{P_{it} : i = 1, \ldots, n\}$, которое следует из утверждений инвариантности. Напомним, что при определении интенсивности перехода (i, j, k) используются вероятности переходов $p(i, j, k)$, которые являются исходными характеристиками системы и задаются «руками». Верно следующее утверждение.

Вероятности $p(i, j, k)$ входят в выражения для P_{it} и N_{it} соразмерно, т. е. так, что если умножить все эти вероятности на одно и то же действительное число $r > 0$, то формулы для P_{it} и N_{it} (согласно свойству масштабируемости) не изменятся, т. е. число r сократится.

Следствие из свойства соразмерности

Если поведение P_t популяции зависит только от двух вероятностей $p_1(i, j, k)$ и $p_2(i, j, k)$, то оно зависит только от их отношения $p_1(i, j, k)/p_2(i, j, k)$.

3. Метод компьютерного моделирования популяций

Опишем метод моделирования с помощью компьютерной программы «Популяция» простой популяции автоматов, функционирующей в дискретном времени $T = = 1, 2, 3, \ldots, t, \ldots$ Вообще говоря, такие системы можно моделировать линейными или нелинейными системами дифференциальных уравнений, для решения которых можно использовать известные численные методы, писать программы или применять готовые системы прикладных программ. Проблема состоит в высокой трудоёмкости этого пути. Между тем построение требуемых уравнений и их численное решение — настолько стандартная процедура, что можно ограничиться только заданием K-сети. Кроме того, в дидактических целях программа должна быть простой, легко управляемой и давать наглядные результаты в виде графиков и массивов результатов моделирования.

3.1. Моделирование простых и автоматных популяций

Пусть N — количество автоматов, n — число состояний, в которых может находиться каждый из автоматов. При этом каждый конкретный автомат не обязательно имеет все n состояний. Популяция может состоять из автоматов различных классов, отличающихся набором состояний и поведением. В каждом i-м состоянии пребывает N_i автоматов, так что $N = N_1 + N_2 + \cdots + N_n$ (n — натуральное, N_i — неотрицательное действительное при $i = 1, \ldots, n$). Использование действительных чисел вместо целых позволяет не задавать очень большие числа N_i и избежать погрешности при округлении. В конечном итоге нас все равно интересует только динамика, или поведение популяции. При этом можно использовать вероятности и другие статистические характеристики.

В каждом такте, то есть через заданный промежуток времени, количество автоматов в j-м состоянии изменяется или остается тем же. Это происходит следующим образом. Некоторое состояние i влияет на состояние j и переводит его в состояние k с интенсивностью $K(i, j, k)$. Множество всех таких интенсивностей — трехмерный массив $P = \{K(i, j, k) : i = 1, \ldots, n, j = 1, \ldots, n, k = 1, \ldots, n\}$.

Это означает, что N_i автоматов, находящихся в состоянии i, влияют на N_j автоматов, находящихся в состоянии j, и переводят некоторое их количество M_{ijk} в состояние k, что можно кратко записать как $i : j \to k$. Значение M_{ijk} вычисляется следующим образом.

При линейном (дальнодействующем) взаимодействии $M_{ijk} = K(i, j, k) \cdot \min\{N_i, N_j\}$; при нелинейном взаимодействии в растворе (при $i \neq j$) $M_{ijk} = K(i, j, k) \cdot \dfrac{N_i N_j}{N}$; при нелинейном взаимодействии в растворе (при $i = j$) $M_{ijk} = K(i, j, k) \cdot \dfrac{N_i N_j}{2N}$; при нелинейном взаимодействии в смеси $M_{ijk} = K(i, j, k) \cdot \dfrac{N_i N_j}{N_i + N_j}$, где $j = 1, \ldots, n$, т. е. одно i-е состояние может воздействовать на множество j-х. Соотношения $(N_i N_j)/N$, $(N_i N_j)/(2N)$ и $(N_i N_j)/(N_i + N_j)$ задают интенсивности встреч автоматов в i-м и j-м состояниях в области действия всех i-х автоматов.

Обратим внимание, что если никаких других состояний, кроме i и j, в системе нет, то $N_i + N_j = N$ и интенсивности переходов в растворе и смеси совпадают. Кроме того, при условии $i = j$ область взаимодействия в растворе сократится вдвое, что и отражено в формулах. В смеси это обстоятельство учитывается автоматически, поскольку $N_i + N_j = 2N_j = 2N$.

При линейном взаимодействии условие $i = j$ означает, что автомат, находящийся в состоянии i, самостоятельно переходит в состояние k независимо ни от каких других автоматов. Если все переходы в популяции линейные и для всех переходов $i = j$, то имеет место простейшая, т. е. линейная автоматная популяция.

При нелинейном переходе условие $i = j$ означает, что для взаимодействия необходима встреча двух автоматов в i-м состоянии, один из которых перейдет в k-е состояние с интенсивностью $K(i, i, k)$.

Если при переходах автоматов из одних состояний в другие общее число автоматов N неизменно, то популяция является замкнутой. В замкнутой популяции можно получить статистические вероятности состояний автоматов $P_i = N_i/N$.

Интерес представляют открытые популяции, где возможно удаление автоматов и появление новых. Для записи этих действий используется *внешнее* или 0-*состояние* q_0, обозначаемое символом «*» или «−». Будем считать, что автоматов, находящихся

в 0-состоянии, всегда достаточно для реализации переходов, а их численность не меняется.

Появление новых автоматов происходит под воздействием уже существующих. Автоматы, находящиеся в состоянии i, добавляют в состояние j новые автоматы с интенсивностью $K(i,*,j)$. Количество появившихся автоматов равно $K(i,*,j)N_i$ при всех типах взаимодействий. Отметим, что при порождении новых автоматов интенсивность — это любое число, соответствующее числу «потомков». Например, пусть рыба мечет 5 тыс. икринок, состояние с — самка, готовая метать икру, и — икринка; тогда $K(\text{с},*,\text{и}) = 5000$.

Удаление автоматов, находящихся в состоянии j, происходит под воздействием уже существующих автоматов, находящихся в некотором i-м состоянии. Удаление происходит с интенсивностью $K(i,j,*)$, т. е. автоматы в i-м состоянии «убивают» автоматы в j-м состоянии с вероятностью $p(i,j,*)$ в окрестности взаимодействия.

В общем случае может оказаться так, что число удаляемых автоматов в состоянии i больше N_i. В этом случае удаляются только N_i автоматов из числа находящихся в состоянии i, т. е. получается $N_i = 0$. Эта возможность учтена в формулах для числа удаляемых автоматов. Кроме того, в программе моделирования популяций предусмотрен сторож, не допускающий значения $N_i < 0$. Обратим внимание, что свойства инвариантности на случай получения $N_i < 0$ не распространяются.

Итак, количество автоматов в такте t изменяется по правилу $N_{i(t+1)} = N_{it} + \Delta N_i$, где $\Delta N_i = V_i - I_i + R_i$; $V_i = \sum\limits_{j=1}^{n}\sum\limits_{k=1}^{n} M_{jki}$ — число автоматов, перешедших в i-е состояние; $I_i = \sum\limits_{j=1}^{n}\sum\limits_{k=0}^{n} \min\{N_i, M_{jik}\}$ — число автоматов, покинувших i-е состояние; $R_i = \sum\limits_{j=1}^{n} N_j \cdot K(j,^*,i)$ — число автоматов, «родившихся» в i-м состоянии. Здесь M_{ijk} вычисляются по формулам, приведенным выше. Чтобы получить среднее количество автоматов в каждом состоянии на $(t+1)$-м шаге, необходимо воспользоваться этими формулами t раз.

3.2. Дополнительные возможности К-моделирования

Выше описано только использование простых переходов, изображённых на рис. 4. Однако некоторые очевидные изменения в программе «Популяция» значительно расширяют возможности моделирования. Добавленные виды переходов изображены на рис. 5–8. Это сохраняющий, удаляющий, остаточный и ингибиторный переходы.

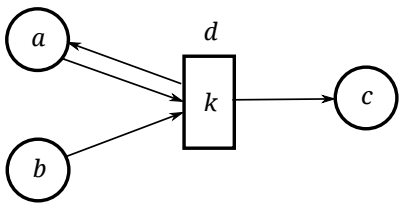


Рис. 4. К-сеть простого перехода $a : b > c : k$: тип 0, 1, 2. Число состояний a — сохраняется, b — убывает, c — увеличивается

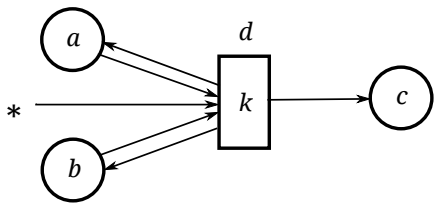


Рис. 5. К-сеть сохраняющего перехода $a, b :> c : k$: тип 3, 4, 5. Число состояний a и b — сохраняется, c — увеличивается. Внешнее состояние «*» не записывается в переход

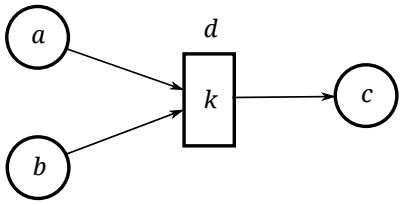


Рис. 6. К-сеть удаляющего перехода $d : a, b > c : k$: тип 6, 7, 8. Число состояний a и b — убывает, c — увеличивается

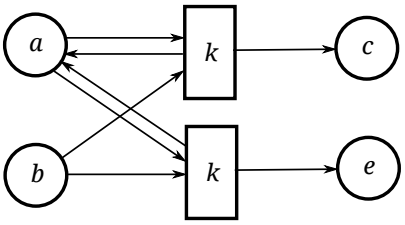


Рис. 7. К-сеть остаточного перехода $a : b > e : k$: тип 9, 10, 11 в комбинации с простым переходом $a : b > c : k$: тип 0, 1, 2. Число состояний a — сохраняется, все автоматы из b переходят в c и e

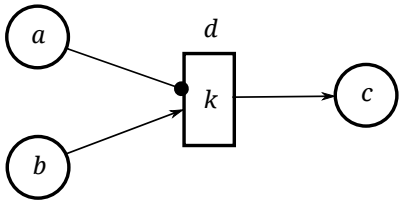


Рис. 8. К-сеть перехода с ингибитором $a : b > c : k$: тип 0, 1, 2. Число состояний a — сохраняется, b — убывает, c — увеличивается. Ингибитор изображён стрелкой с кружком на конце

Итак, имеем следующие переходы:

1) *Простой* переход, описанный правилом «Состояние a переводит состояние b в состояние c с интенсивностью k по типу 0, 1, 2».
2) *Сохраняющий* переход, не изменяющий числа состояний в обеих входных позициях, что позволяет моделировать, например, гиперболический рост с обострением. Его правило «Состояния a и b не изменяются, но порождают новые состояния c с интенсивностью k по типу 3, 4, 5».
3) *Удаляющий* переход, извлекающий маркеры из обеих входных позиций, как это делается в стандартной сети Петри. Его правило «Состояния a и b исчезают и порождают новые состояния c с интенсивностью k по типу 6, 7, 8».
4) *Остаточный* переход — модификация простого перехода, такая, что изменяют свои состояния все те автоматы популяции, которые не изменили его в аналогичном простом переходе. Его правило «Состояние a переводит в состояние c те автоматы, которые находятся в состоянии b, но не перешли бы в состояние c при простом переходе. Интенсивность k; типы 9, 10, 11».
5) *Ингибиторный* переход, дополнительный к простому: «Если в области действия нет состояния a, то состояние b переходит в состояние c с интенсивностью k по типу 0, 1, 2». Здесь состояние a играет роль не инициатора, а ингибитора, запрещающего переход.

Все эти переходы могут работать линейно, в растворе и в смеси. Таким образом, имеется три типа и пять видов переходов, отличающихся способом подсчета интенсивностей. В результате получается следующая таблица возможных переходов в К-моделях.

Таблица 2

Виды и типы переходов программы «Популяция»

Вид перехода	**Тип перехода**		
	Линейный	Раствор	Смесь
Простой	0	1	2
Сохраняющий	3	4	5
Остаточный	6	7	8
Удаляющий	9	10	11
Ингибиторный	12	13	14

Кроме того, допустимы и временные запаздывания действующих состояний К-сети. Например, число повзрослевших особей в популяции определяется не наличным числом рождённых, а рождёнными некоторое время назад, определяемое возрастом взрослой особи.

Заключение

Итак, впервые получен метод составления и численного решения линейных и нелинейных дифференциальных уравнений динамики средних (К-модели) на базе маркированного графа параллельных процессов в популяции взаимодействующих автоматов (К-сети).

Несмотря на некоторую ограниченность данного подхода, класс моделируемых популяций включает множество интересных систем, допускающих множество интерпретаций в различных предметных областях: вычислительной технике, биологии, социологии, истории, экономике и т. д. — везде, где поведение системы можно представить как параллельное функционирование множества автоматов, взаимодействующих между собой.

Имея программу «Популяция», достаточно описать поведение отдельных элементов исследуемой системы в их связи с другими элементами, и получается модель, которая легко модифицируется и быстро даёт наглядные результаты. При этом будет представлено и поведение популяции в переходном режиме. А это уже немало в тех предметных областях, где господствуют качественные рассуждения. Это касается гуманитарных наук и, особенно, истории. Проблема математизации исторических исследований давно стоит на повестке дня [4 – 7].

ЛИТЕРАТУРА

1. *Ачасова С. М., Бандман О. Л.* Корректность параллельных вычислительных процессов. Новосибирск: Наука, 1990. 314 с.
2. *Вентцель Е С., Овчаров Л. А.* Теория случайных процессов и её инженерные приложения. Учебн. пособие для втузов. 2-е изд., стер. М.: Высшая школа, 2000. 383 с.
3. *Воробьев В. А., Кочнев А. И.* Популяционное моделирование коллективного поведения автоматов // Вестник Томского госуниверситета. Приложение. 2007. № 23. С. 270–275.
4. *Капица С. П., Курдюмов С. П., Малинецкий Г. Г.* Синергетика и прогнозы будущего. 2-е изд. М.: Эдиториал УРСС, 2001. 288 с.
5. *Воробьев В. А., Воробьева Т. В.* Экологическая пауза — системный кризис человечества // Труды АНИГ «Прогноз». Вып. 1. Исследования в области глобального катастрофизма. Новосибирск: НГУ, 2006. С. 69–109.

6. *Коротаев А. В., Комарова Н. Л., Халтурина Д. А.* Законы истории: вековые циклы и тысячелетние тренды. Демография, экономика, войны. 2-е изд., испр. и доп. / под ред. Н. Н. Крадина. М.: КомКнига, 2007. 256 с.
7. *Коротаев А. В., Малков А. С., Халтурина Д. А.* Законы истории: математическое моделирование развития Мир-Системы. Демография, экономика, культура. 2-е изд., испр. и доп. / под ред. Н. Н. Крадина. М.: КомКнига, 2007. 224 с.

АНАЛИТИЧЕСКИЕ ОБЗОРЫ

DOI 10.17223/20710410/14/11

УДК 519.7

SIBECRYPT'11. ОБЗОР ЛЕКЦИЙ И ДОКЛАДОВ

Г. П. Агибалов

Национальный исследовательский Томский государственный университет, г. Томск, Россия

E-mail: agibalov@isc.tsu.ru

Приводится аналитический обзор лекций и докладов, представленных на Sibecrypt'11 — X Всероссийской конференции «Сибирская научная школа-семинар с международным участием „Компьютерная безопасность и криптография“», состоявшейся 5 – 9 сентября 2011 г. в Национальном исследовательском Томском государственном университете (г. Томск).

Ключевые слова: *прикладная дискретная математика, криптография, компьютерная безопасность, защита информации.*

Введение

Sibecrypt — это Всероссийская конференция под названием «Сибирская научная школа-семинар с международным участием „Компьютерная безопасность и криптография“». Её ежегодно, начиная с 2002 г., организует и в первой трети сентября проводит кафедра защиты информации и криптографии Национального исследовательского Томского государственного университета (ТГУ, г. Томск) в сотрудничестве с кафедрой программирования и компьютерной безопасности Института криптографии, связи и информатики (ИКСИ, г. Москва) на базе того или иного вуза или научного учреждения Сибири. Кроме докладов, на конференции Sibecrypt для её участников, а также для сотрудников и студентов принимающей организации (вуза, НИИ) ведущими специалистами в данной области (из числа участников конференции) читаются лекции по современным проблемам компьютерной безопасности, защиты информации и криптографии. Материалы конференции публикуются в приложении к журналу «Прикладная дискретная математика».

В 2011 г. Sibecrypt состоялась в 10-й раз — с аббревиатурой Sibecrypt'11, на этот раз — 5 – 9 сентября в Томске на базе ТГУ. Тезисы докладов, представленных в её программу, опубликованы в [1]. Аналитический обзор их содержания, а также содержания лекций, прочитанных на Sibecrypt'11, является целью данной статьи.

1. Лекции по криптографии и компьютерной безопасности

В лекции М. М. Глухова «К анализу некоторых систем открытого распределения ключей, основанных на неабелевых группах» проанализированы некоторые известные системы открытого распределения ключей в некоммутативных группах, основанные на проблеме сопряжённости в последних (A. Yamamura (1998, 1999); I. Anshel, M. Anshel, D. Goldfeld (1999); K. H. Koo, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang, C. Park (2000)) и на её композиции (неявной или явной) с проблемой логарифмирования в таких

группах — так называемые MOR-системы (S. H. Paeng, K. C. Ha, J. H. Kim, S. Chee, C. Park (2001); S. H. Paeng, D. Kwon, K. C. Ha, J. H. Kim (2001); C. Tobias (2004); E. Sakalauskas, P. Tvarijonas, A. Raulynaitis (2007)), а также система отечественных авторов из компании Молдовянов, сходная с последней из перечисленных MOR-систем (Moldovyan N. A., Moldovyan P. A. (2009); Moldovyan D. N., Moldovyan N. A. (2009); Молдовян Д. Н., Куприянов А. И., Костина А. А., Захаров Д. В. (2009); Молдовян Н. А. (2010)). Показано, в частности, что система E. Sakalauskas, P. Tvarijonas, A. Raulynaitis является частным видом систем, предложенных в работах В. М. Сидельникова, М. А. Черепнева, В. В. Ященко (1993) и В. М. Сидельникова (1994), а криптографическая стойкость системы открытого распределения ключей Молдовянов по порядку не превосходит сложности проблемы дискретного логарифмирования в циклической подгруппе порядка q мультипликативной группы простого поля $\mathbb{Z}_p$ или его расширении 2-й степени, где q является делителем числа $p-1$ или $p+1$ соответственно. Такой же результат получается и для всех других криптосистем подобного типа, в которых используются группы, определенные на алгебрах размерности 4 над полем $\mathbb{Z}_p$, и которые предлагаются в ряде работ Н. А. Молдовяна и его коллег.

Методы алгебраической геометрии в криптографии как предмет для фундаментальных научных исследований и учебной математической дисциплины представлены в одноимённой лекции И. А. Круглова. Интерес специалистов к ним вызван, прежде всего, их широким применением в реальных криптографических системах, в том числе в обретших статус государственного стандарта, и потребностями в их дальнейшем развитии в интересах науки и практики защиты информации. Наибольшее внимание в лекции уделено исследованию эллиптических кривых над конечным полем — их групп точек и реализаций на проективной плоскости, рациональных функций и дивизоров на проективных эллиптических кривых, эллиптических конфигураций и алгоритмов логарифмирования в группе точек эллиптической кривой и факторизации целых чисел с помощью эллиптических кривых.

Важнейшей составляющей компьютерной безопасности является безопасность программного обеспечения. Анализ программных реализаций и защита программ от анализа, программные закладки, пути их внедрения и методы противодействия им, уязвимости в программах и методы их использования, языки безопасного программирования и создание безопасных программ, интеграция политик безопасности и программных продуктов — вот далеко не полный перечень тех проблем, решением которых занимается наука под условным названием «Безопасное программирование». В лекции В. Г. Проскурина рассказано о методах решения части этих проблем, связанных с защитой программ от анализа и закладок. Примечательность этих методов и их особенная ценность заключается в том, что они не продукт досужего ума и теоретических изысков, но выстраданы, рождены и прошли серьёзнейшие испытания в многолетней практической работе Вадима Геннадьевича по защите реальных программных систем в условиях реальных угроз и атак. Заинтересованный читатель может найти их в подробном изложении в учебном пособии: *В. Г. Проскурин.* Защита программ и данных. М.: Издательский дом «Академия», 2011. 208 с.

История криптографии с древнейших времён до сегодняшнего дня стала предметом лекции А. В. Черёмушкина. В ней посредством ярких и запоминающихся иллюстраций прослежен путь развития и применения мировой криптографии от шифра простой замены до современных криптосистем с открытым ключом и отмечены некоторые из наиболее выдающихся достижений отечественных и зарубежных криптографов.

Криптография, базирующаяся на бесконечных разрешимых группах, представлена в одноимённой лекции, прочитанной В. А. Романьковым. Важнейшим тезисом лекции является демонстрация возможности сведения базовых математических задач криптографии к решению систем диофантовых уравнений.

Некоторые малоизвестные факты, относящиеся к первым десяти годам в 50-летней истории криптографии в Томском государственном университете, приведены в лекции Г. П. Агибалова. Они касаются, главным образом, истории создания и опубликования конечно-автоматного шифра, предложенного А. Д. Закревским в 1959 г., и алгоритмов криптоанализа и оценок теоретической стойкости, полученных Г. П. Агибаловым в 1964–1966 гг. для генераторов ключевого потока трёх классов: 1) линейных автономных автоматов над конечным полем; 2) нелинейных автономных автоматов с функцией выхода в качестве ключа и 3) нелинейных генераторов, порождающих многозначные нормальные (максимального периода) рекуррентные последовательности.

2. Теоретические основы прикладной дискретной математики

Как всегда, это направление на конференции Sibecrypt широко представлено результатами исследований дискретных функций и подстановок. Традиционными на ней становятся также методы алгебраической геометрии и комбинаторного анализа.

Два доклада Н. Г. Парватова посвящены проблемам полноты и выразимости в пространствах дискретных функций. Их решение имеет фундаментальное значение для выяснения важнейших закономерностей, существующих в мире дискретных математических объектов, и их приложений к математической кибернетике, информатике, защите информации и криптографии. Установлены необходимые и достаточные условия существования конечных нижних окрестностей у произвольных или заданных конечно порождаемых классов произвольного пространства и пространства с замыканием Галуа. Введены в рассмотрение сильно предупорядоченные пространства и установлено существование в них конечных нижних окрестностей у конечно порождаемых классов и конечных запрещающих множеств у классов с конечными верхними окрестностями. Установлена сильная предупорядоченность относительно подстановки переменных ряда функциональных пространств, в том числе пространства переключательных функций с замыканием. Построена теория Галуа для пространств переключательных функций с замыканием, описывающая его как замыкание Галуа. Тем самым найдено единое обобщение ряда различных теорий Галуа, содержащее их в качестве частных случаев и имеющее собственные приложения в теории переключательных схем. В силу этих результатов проблема выразимости для конечно порождаемого класса переключательных функций имеет решение в виде конечной нижней окрестности, классы которой имеют конечные запрещающие множества и одноэлементные описания. В связи с проблемой конечной порождаемости выделено новое семейство конечно порождаемых клонов — содержащих конечно порождаемый d- или произвольный (c, d)-подклон при натуральном c. Клоны с (c, d)-подклонами охарактеризованы свойствами инвариантных предикатов. Установлена возможность (c, r)-разложений клона над (c, d)-подклоном, известная ранее лишь в случае $c = 0$. Найдены предикатные и-описания клонов квазимонотонных и слабо существенных квазимонотонных функций, монотонных частей этих клонов. В частности, установлено, что монотонная часть является 2-подклоном в клоне квазимонотонных и $(1, 2)$-подклоном в клоне слабо существенных квазимонотонных функций. В связи с задачей выделения замкнутых классов в множествах точечных и минимальных точечных функций доказано, что в каждом из указанных двух множеств всякий замкнутый класс расширяется до некоторого макси-

мального из конечного множества. Построены примеры максимальных таких клонов. Явно описаны классы троичных функций, вычисляемых дизъюнктивными формами и произвольными формулами в каноническом базисе. Конструктивно доказано, что класс минимальных точечных функций на дистрибутивной точечной полурешётке порождается двухместными функциями. В качестве доказательства этого предложен метод формульного представления минимальных точечных функций на дистрибутивной точечной полурешётке в бинарных базисах, содержащих все одноместные минимальные точечные функции и некоторый набор специальных двухместных функций. Установлены необходимые и достаточные условия максимальности подклона, заданного расширенным и-описанием. На основе этого построена безызбыточная критериальная система в клоне квазимонотонных функций на полурешётке при суперпозиции со слабо существенными функциями. Найдена асимптотика её мощности в случае полурешётки всех непустых подмножеств множества k-элементного множества. Для функций на трёхэлементной полурешётке найдены безызбыточная нижняя окрестность множества минимальных точечных функций в клоне монотонных функций и безызбыточные критериальные системы в клонах монотонных и квазимонотонных функций. Эти результаты представляют особый интерес, так как функциями на трёхэлементной полурешётке описывается динамическое поведение дискретных асинхронных управляющих систем с двоичными статическими состояниями, и полученные решения проблем полноты и выразимости для них имеют прямое применение в проектировании таких систем.

Н. А. Коломеец в своём докладе описал все бент-функции, находящиеся на минимальном расстоянии от произвольной квадратичной бент-функции, и подсчитал их количество. Для функции от $2k$ переменных это число равно $2^k(2^1+1)(2^2+1)\cdot\ldots\cdot(2^k+1)$.

В докладе Е. П. Корсаковой введено графовое представление квадратичной булевой функции, в котором вершины графа суть аргументы функции, а рёбра соединяют те пары вершин, которые образуют слагаемые в АНФ функции. Типом графа назван упорядоченный по убыванию набор степеней его вершин. Функции названы графово эквивалентными, если их графы изоморфны. Для всех квадратичных бент-функций от 6 переменных определены типы их графов и построены классы их графовой эквивалентности. Оказалось, число первых равно 37, вторых — 50.

Для множества $\mathcal{B}_n$ бент-функций и множества $\mathcal{BI}_n$ итеративных бент-функций от n переменных Н. Н. Токарева доказала, в частности, что $|\mathcal{BI}_{n+2}| \geqslant |\mathcal{B}_n|^4/|X_n|$, где X_n есть множество всех булевых функций от n переменных, представимых суммой двух бент-функций, и сформулировала три гипотезы о числах $|\mathcal{B}_n|$ и $|\mathcal{BI}_n|$: 1) последняя оценка асимптотически точна; 2) $|\mathcal{B}_n|$ асимптотически равно 2^s, где $s = 2^{n-c} + d\binom{n}{n/2}$, c и d — константы, $1 \leqslant c \leqslant 2$; 3) $|\mathcal{B}_n|$ и $|\mathcal{BI}_n|$ асимптотически совпадают. Она высказала также гипотезу о том, что каждая булева функция от n переменных степени не больше $n/2$ представима суммой двух бент-функций от n переменных, из которой гипотеза 2 следует немедленно.

Говорят, что булева функция f *статистически не зависит* от подмножества Z своих переменных, если для любой её подфункции f', полученной фиксированием значений переменных в Z, имеет место $\Pr[f' = 1] = \Pr[f = 1]$. В докладе О. Л. Колчевой и И. А. Панкратовой, наряду с несколькими простейшими свойствами этого понятия, установлено, что если x, y, z суть непустые наборы различных булевых переменных и $f(x, y)$ статистически не зависит от переменных в x, то для любой булевой функции g от $|z| + 1$ переменной суперпозиция $g(f(x, y), z)$ также статистически не зависит от x, и это утверждение не допускает обобщения на случай, когда под знаком g вместо одной f стоят не менее двух функций от x, y, статистически не зависящих от x.

Дано описание подстановок степени n, представимых произведениями двух подстановок с фиксированным числом q мобильных точек (А. Б. Пичкур). Так, если $n \geqslant 8$, $4 \leqslant q \leqslant n/2$, то всякая подстановка G степени n с числом m мобильных точек не больше $2q-2$ представима произведением двух подстановок степени n с q мобильными точками. В других рассмотренных случаях, а именно когда $n \geqslant 4$, $2 \leqslant q \leqslant n/2$ и $m = 2q$ или $m = 2q-1$, подстановка G, имеющая r неединичных циклов с длинами $m_1, \ldots, m_r$, допускает требуемое представление тогда и только тогда, когда соответственно существует такое $I \subseteq \{1, \ldots, r\}$, что $\sum\limits_{i \in I} m_i = q$, или существуют такие $j \in \{1, \ldots, r\}$ и $I \cup \{k\} \subseteq \{1, \ldots, r\} \setminus \{j\}$, что $m_j > 2$ и $q - m_k + \sum\limits_{i \in I} m_i \in \{2, \ldots, m_j - 1\}$.

Во многих блочных шифрах блоки замены являются подстановками. Вместо приближений их аффинными функциями можно строить для них приближения другими подстановками, в некотором смысле более простыми. В докладе Б. А. Погорелова и М. А. Пудовкиной в роли последних рассматриваются подстановки, сохраняющие некоторую нетривиальную систему W областей импримитивности и образующие, таким образом, некоторую импримитивную группу IG_W. Множество всех таких систем W с r областями импримитивности мощности w обозначается $W_{w,r}$. Расстояние между подстановками определяется по Хэммингу, и для произвольной подстановки g определяются её порядки *W-примитивности* и *(w,r)-примитивности* как наименьшие расстояния от неё до подстановки соответственно в группе IG_W и в объединении $IG_{w,r}$ групп IG_W для всех $W \in W_{w,r}$. Авторам доклада удалось описать некоторые классы подстановок максимального порядка W-примитивности — так называемых бент-подстановок относительно заданной системы импримитивности W — и построить оценки числа таких подстановок. Порядок (w,r)-примитивности подстановки определяется однозначно её цикловой структурой. Перечислены цикловые структуры для всех подстановок в $IG_{w,r}$ и получены порядки их (w,r)-примитивности при чётной степени подстановки и для $w = r = 2$. Вычислены также порядки (w,r)-примитивности для блоков замены AES, ARIA, Whirlpool, MISTY1, Camellia, FOX.

Множество вейерштрассовых точек алгебраического функционального поля, ассоциированного с алгебраической кривой, является её инвариантом и может быть использовано для многих целей, в том числе для изучения группы автоморфизмов кривой, в частности её порядка. В докладе Е. С. Алексеенко предложен алгоритм для вычисления вейерштрассовых точек такого поля произвольной характеристики. Алгоритм сформулирован в предположении о наличии сепарирующего элемента в поле и известных процедур вычисления точек, дивизоров и пространств, ассоциированных с заданным дивизором.

Для натуральных ε и δ отображение f множества вершин графа G в множество вершин графа H называется $\langle \varepsilon, \delta \rangle$-вложением G в H, если для любой вершины v первого f обладает свойством $\langle \varepsilon, \delta \rangle$-ограниченного искажения: $f(S_\delta(v)) \subseteq S_\varepsilon(f(v))$ и сохраняет $\langle \varepsilon, \delta \rangle$-отделимость: $\operatorname{Im} f \cap S_\varepsilon(f(v)) \subseteq f(S_\delta(v))$, где $S_k(v)$ есть шар радиуса k с центром в v. В докладе А. А. Евдокимова конструктивно показано существование $\langle 4, 3 \rangle$-вложения целочисленной решётки размера $m \times m$ в n-мерный булев куб с асимптотически минимальной избыточностью и с мощностью решётки, удовлетворяющей соотношению $m^2 > 2^s$, где $s = n - 2\log_2 n(1+\varepsilon_n)$ и $\varepsilon_n \to 0$ при $n \to \infty$.

Функция $f : \mathbb{Z}_q^n \to \mathbb{Z}_q$ *корреляционно-иммунная порядка* $n-m$, если мощность пересечения грани размерности m в гиперкубе $\mathbb{Z}_q^n$ с множеством $f^{-1}(a)$ зависит только от $a \in \mathbb{Z}_q^n$; в этом случае наибольшее из таких $n-m$ обозначается $\operatorname{cor}(f)$. Плотность булевозначной функции $f : \mathbb{Z}_q^n \to \mathbb{Z}_2$ есть число $\rho(f) = |S_f|/q^n$,

где $S_f = \{a \in \mathbb{Z}_q^n : f(a) = 1\}$. Отображение $\mathrm{col} : \mathbb{Z}_q^n \to \mathbb{Z}_2$ называется совершенной 2-раскраской гиперкуба $\mathbb{Z}_q^n$, если существуют целые неотрицательные числа m_{ij} для i, j в $\{0, 1\}$, называемые параметрами раскраски, такие, что для каждой вершины $a \in \mathbb{Z}_q^n$ цвета $i = \mathrm{col}(a)$ число соседей цвета j равно m_{ij}. Среднее число вершин в $S \subseteq \mathbb{Z}_q^n$, находящихся на расстоянии 1 от вершины из дополнения $\mathbb{Z}_q^n \setminus S$, обозначается $A(S)$, т. е. $A(S) = \sum\limits_{x \notin S} |\{y \in S : d(x, y) = 1\}|/(q^n - |S|)$, где $d(x, y)$ — расстояние Хэмминга между наборами x и y. Доказано, что для любой булевозначной функции $f : \mathbb{Z}_q^n \to \mathbb{Z}_2$ справедливо неравенство $q\rho(f)(\mathrm{cor}(f)+1) \leqslant A(S_f)$, и f является совершенной 2-раскраской тогда и только тогда, когда в этом соотношении выполняется равенство (В. Н. Потапов).

Аналог теоремы Клини для языков конечных автоматов представлен в докладе Е. А. Пряничниковой для языков, представимых в отмеченных графах (с отмеченными только вершинами или с отмеченными только дугами). В нём регулярные выражения строятся с помощью операций теоретико-множественного объединения и обобщённых конкатенации и итерации языков. Операция обобщённой конкатенации $\overset{n}{\circ}$ имеет неотрицательный целочисленный параметр n, является бинарной, частичной и над произвольными словами u, w определяется как $u \overset{n}{\circ} w = xyz$, если $u = xy$, $w = yz$, $|y| = n$, и результат операции не определён в противном случае. Над языками L и R она определяется так: $L \overset{n}{\circ} R = \{u \overset{n}{\circ} w : u \in L, w \in R\}$. Унарная операция обобщённой итерации также параметрическая, и результатом её применения к языку L является объединение языков L_i, $i = 1, 2, \ldots$, где $L_1 = L$, $L_{i+1} = L_i \overset{n}{\circ} L$ для всех $i \geqslant 1$.

3. Математические методы криптографии

Важной характеристикой стойкости поточных шифров является расстояние единственности используемого ключевого потока. Основой построения многих ключевых потоков являются линейные рекуррентные последовательности (ЛРП) над конечными полем или кольцом. Часто за ключевой поток принимается последовательность значений старшего разряда в двоичном представлении элементов некоторой ЛРП. В этом случае характеристический многочлен последней называют характеристическим многочленом и данного ключевого потока. Расстояние единственности такого ключевого потока определяется как длина кратчайшего префикса, которым он (поток) отличается от всех других ключевых потоков с тем же характеристическим многочленом. Наибольшее из расстояний единственности ключевых потоков с одним и тем же характеристическим многочленом рассматривают как расстояние единственности самого многочлена. Это есть наименьшее натуральное число l, такое, что по префиксу длины l любого ключевого потока с данным характеристическим многочленом однозначно восстанавливается весь поток. Доклад А. В. Аборнева и Д. Н. Былкова посвящён поиску многочленов над примарными кольцами вычетов с расстоянием единственности, равным двум степеням многочлена. Эти многочлены интересны тем, что префиксы длины $2m$ всех ключевых потоков с одним и тем же характеристическим многочленом степени m задают на кольце подстановку степени m, которая в роли раундовой функции итеративного блочного шифра с аддитивным раундовым ключом часто обладает свойствами, противостоящими дифференциальному и линейному криптоанализам этого шифра. Примерами многочленов степени m с расстоянием единственности $2m$ являются тривиальные многочлены вида $x^m + 1 \pmod 2$. Существование других, нетривиальных, таких многочленов при любом m пока не установлено. Показано, однако, что многочлены вида $f_0(x)+2f_1(x)$, где $f_0(x) \equiv (x+1)^m \pmod 2$, $(f_1(x)+x^s, x+1) = 1$, $m = 2^k + s$, k натуральное и s нечётно, имеют расстояние единственности $2m$.

Отображения декартовой степени A^n конечного множества A в A^n со свойством идентифицируемости на подмножестве их координат рассмотрены в докладе Л. Н. Андреевой. Предложен способ расширения их до отображений большей степени, сохраняющих это свойство. В случае, когда такие отображения являются инволюциями, применяемыми в схемах разделения секрета, и в множество участников схемы вводятся новые участники, данный результат позволяет неавторизованные множества прежней схемы включить в неавторизованные множества новой схемы. Для произвольной инволюции q на A^n сформулирован и доказан тест идентифицируемости её на подмножестве координат I, заключающийся в проверке условий $q(x)[I] \neq q(y)[I]$ для всех x, y в A^n, где $x \neq y$.

Предложена доказуемо безопасная схема групповой подписи, построенная на основе известной схемы BBS, с возможностью отзыва права подписи у любого члена группы и добавления в группу новых членов (А. В. Артамонов, П. Н. Васильев, Е. Б. Маховенко). В ней ключом члена группы является тройка $(A, x, y) \in G_1 \times \mathbb{Z}_p^2$, где $A^{x+\gamma} = g_1 h^y$; $\gamma \in \mathbb{Z}_p$ — секретный ключ выпускающего менеджера группы; $\langle g_1 \rangle = G_1$ — циклическая группа простого порядка p; $h \in G_1$ — элемент открытого ключа группы. Для обеспечения полной анонимности в предложенной схеме вместо CPA-стойкой линейной схемы шифрования BBS используется модифицированная CCA-стойкая линейная схема Крамера — Шоупа. Для решения технических проблем, связанных с сохранением корректности ранее сгенерированных подписей после отзыва права подписи, с синхронизацией остальных субъектов и их баз данных, а также с принудительным обновлением локальных копий ключей всеми субъектами и всей базы данных членов группы выпускающим менеджером, в схему введён доверенный субъект. В результате процесс формирования подписи стал интерактивным с участием удостоверяющего центра, а проверяющий имеет возможность использовать для проверки актуальный на момент создания подписи открытый ключ группы.

Алгебраическая атака на один раунд упрощенного шифра AES, известного как S-AES, исследована в докладе Р. И. Воронина. При известных блоке открытого текста (ОТ) и соответствующем блоке шифртекста (ШТ) атака состоит в решении нелинейной системы из 32 булевых уравнений, связывающих 16 неизвестных бит раундового ключа с известными битами блоков ОТ и ШТ. В своей атаке автор исходит из двух пар известных блоков ОТ/ШТ при одном и том же раундовом ключе, заменяя (аппроксимируя) две соответствующие им нелинейные подсистемы уравнений одной — линейной системой L с 32 уравнениями и 16 неизвестными. Последняя, естественно, может быть несовместной. В докладе показано, что в случае случайного и фиксированного ключа справедливы следующие предложения: 1) при случайном равновероятном выборе блоков ОТ система L совместна с вероятностью 0,6074; 2) в отсутствие дефекта в сумме (побитовой по mod 2) одного блока ОТ с ключом и при случайном равновероятном выборе другого блока ОТ система L совместна с вероятностью 0,7725. Здесь под дефектом в булевом векторе длины 16 понимается наличие блока из одних нулей в его разбиении на 4 блока по 4 бита в каждом. О том, насколько эффективна данная атака, можно судить из следующего экспериментального факта: для некоторых ключа и одного блока ОТ находятся более 68 % вторых блоков ОТ, при которых система L имеет единственное решение — истинный ключ.

С. Ю. Ерофеев доказал диофантовость дискретного логарифма, построив систему уравнений E в натуральных переменных и известных натуральных i, p, n с простым p, такую, что если E имеет решение в натуральных числах и k° — значение некоторой переменной k в этом решении, то $n^{k^\circ} \equiv i \pmod{p}$, и наоборот, если $n^{k'} \equiv i \pmod{p}$

для некоторого натурального k', то система E имеет решение в натуральных числах и $k^\circ \equiv k' \pmod p$ для значения k° переменной k в этом решении.

В докладе С. Ю. Ерофеева и В. А. Романькова предложены схема построения односторонней функции (точнее, кандидата в односторонние функции) в свободной группе G с разрешимой проблемой равенства и неразрешимой проблемой эндоморфной сводимости и протокол идентификации в группе, в которой, кроме того, неразрешима проблема двукратной эндоморфной сводимости. Если $X = \{x_1, \ldots, x_n\}$ — базис группы G, $g(x_1, \ldots, x_n)$ есть выражение элемента $g \in G$ через базисные элементы и отображение $\varphi : X \to G$ продолжается до эндоморфизма на G, то значение односторонней функции определяется на $g(x_1, \ldots, x_n)$ как элемент $f \in G$, такой, что $f = g(\varphi(x_1), \ldots, \varphi(x_n))$. Протокол идентификации построен по трёхшаговой схеме запрос-ответной идентификации — аналогично схеме Фиата — Шамира, но на другом математическом аппарате. В нём открытый ключ доказывающей стороны A есть пара различных элементов f, g в свободной метабелевой группе M_n достаточно большого ранга n ($n \geqslant 13$), а её закрытый ключ — эндоморфизм $\varphi \in \mathrm{End}(M_n)$, такой, что $\varphi(g) = f$. 1) Сторона A выбирает обязательство $\psi \in_{\mathrm{R}} End(M_n)$ и посылает проверяющей стороне B свидетельство $v = \psi(f)$. 2) Сторона B направляет стороне A запрос $c \in_{\mathrm{R}} \{0, 1\}$. 3) Если $c = 0$, то A посылает B ответ ψ; в противном случае — ответ $\psi\varphi$. 4) Сторона B проверяет равенство $v = \psi(f)$ или $v = \psi\varphi(g)$ соответственно. Протокол повторяется t раз, и если проверяемое равенство выполняется всякий раз, то идентификация принимается, в противном случае — отвергается. Авторы доклада полагают, что безопасность данного протокола основана на неразрешимости проблем эндоморфной и двукратной эндоморфной сводимости в M_n. Первая доказана В. А. Романьковым ранее (1979 г.), вторая утверждается в докладе.

Информация об аппаратной реализации шифра из японской криптосистемы FAPKC (*Finite Automata Public Key Cryptosystem*) представлена в докладе Д. С. Ковалёва и В. Н. Тренькаева. Реализация осуществлена с помощью САПР Xilinx WebPack ISE на базе ПЛИС Spartan-3 XC3S1500. Показано, в частности, что коэффициент эффективности (отношение производительности к количеству используемых ресурсов) этой реализации на порядок выше коэффициента эффективности известной аналогичной реализации RSA. Оказалось также, что с увеличением величины задержки расшифрования в ней число требуемых ресурсов ПЛИС значительно возрастает, а их рабочая частота убывает незначительно.

Стойкость режима шифрования к некоторой атаке принято оценивать разностью величин, i-я из которых есть вероятность того, что в данный шифртекст в данном режиме посредством данной атаки преобразуется i-й из двух случайно выбранных открытых текстов, $i = 1, 2$. В докладе И. А. Кукало проанализирована стойкость режимов шифрования в ГОСТ 28147-89 к атаке, основанной на парадоксе дня рождения. Приведены оценки такой стойкости для режимов простой замены, гаммирования и гаммирования с обратной связью, из которых следует, что для безопасного шифрования количество блоков открытого текста не должно превышать в первом режиме числа 1, во втором — числа 2^{31}, в третьем — числа $\sqrt{2^{64}/3}$.

В известной многошаговой корреляционной атаке на шифр Keeloq для выбора так называемых слайдовых пар на каждом шаге приходится перебирать пары булевых векторов длины 32 и для каждой из них проводить корреляционную атаку. В докладе О. Н. Лебедевой предложено сократить этот перебор, отсеивая те пары, для которых не выполняется некоторое вероятностное соотношение между битами в паре и известными в этот момент битами ключа.

Разности с нулевой вероятностью, называемые невозможными, успешно используются в криптоанализе ряда блочных шифров. В докладе М. А. Пудовкиной доказано существование трёхраундовых невозможных разностей и сформулированы условия существования четырёхраундовых невозможных разностей в XSL-алгоритмах блочного шифрования.

4. Математические основы компьютерной безопасности

Главные достижения отечественной науки в этом направлении, имеющие мировое значение, связаны с разработкой и исследованием математических моделей безопасности компьютерных систем (КС), и им мы обязаны Петру Николаевичу Девянину, его ученикам и последователям. На Sibecrypt эти достижения традиционно наиболее представительные, вызывают наибольший интерес, особенно в среде молодых и начинающих учёных.

В докладе П. Н. Девянина рассказано о результатах разработки ролевой ДП-модели управления доступом и информационными потоками в ОС семейства Linux, о её основных отличиях от базовой модели этого класса, в частности о наличии в ней механизма ограничений в условиях и результатах применения правил преобразования состояний, потребовавшего использования наряду с монотонными и немонотонные правила при передаче прав доступа ролей или возникновения информационных потоков, что, в свою очередь, потребовало введения ограничений, инвариантных относительно немонотонных правил преобразования состояний КС в том смысле, что на любой траектории системы применение или неприменение немонотонного правила не влияет на выполнение ограничений в следующих за ним правилах преобразований. Доказано, что при наличии в КС только инвариантных ограничений в анализе условий передачи прав доступа ролей и реализации информационных потоков достаточно использовать только монотонные правила преобразования состояний.

Наиболее распространённым программным средством безопасного управления доступом и информационными потоками в ОС семейства GNU/Linux является КС SELinux. Естественно, возникает задача анализа безопасности самого этого средства. В докладе М. А. Качанова сообщается о разработке с этой целью ролевой ДП-модели безопасности управления доступом и информационными потоками в КС SELinux и о методе её применения для проверки возможности получения в КС права доступа и реализации информационного потока. В модели данная КС представляется системой, где каждое состояние задаётся набором объектов, а каждый переход из состояния в состояние осуществляется по одному из правил преобразования состояний. Применение модели на практике распадается на два этапа. На первом этапе строится начальное состояние модели по набору конфигурационных файлов моделируемой КС, на втором — его memory- и time-замыкания. Возможность получения заданного права доступа или реализации заданного информационного потока в КС обнаруживается проверкой истинности некоторых предикатов на построенных замыканиях.

Проблеме построения ДП-моделей для КС сетевого управления доступом посвящён доклад Д. Н. Колегова. Проблема осложняется рядом особенностей, присущих таким КС, не поддающихся адекватному отражению в ДП-моделях КС других типов. Важнейшие из них связаны со свойствами распределённости компонентов управления, его динамичности, множественности правил управления доступом для одних и тех же субъектов, принадлежности сущностей одновременно нескольким иерархиям и др. Для адекватного описания этих особенностей предложено язык ДП-моделей расширить, включив в него средства для задания множества функций иерархии сущностей,

множеств сущностей, параметрически ассоциированных с субъектом, прав доступа, свойственных сетевой КС, таких, как право доступа к её сушности, право конфигурирования её сущностей и т. п., множеств учётных записей и векторов доступа последних с узлов сети к сущностям, а также правила преобразования состояний с целью создания сессии удалённого доступа с правами доступа учётной записи и назначения субъекту в рамках этой сессии права доступа учётной записи.

Аналогичная проблема, но относящаяся к разработке дискреционной ДП-модели защищенных ОС, решается в докладе В. Г. Проскурина. Для адекватного моделирования таких КС в ДП-модель вводится ряд новых элементов, в том числе: множество учётных записей пользователей, делегирующих субъектам права доступа от имени свих учётных записей; право доступа *grant* для предоставления субъектом-владельцем ограниченного доступа к сущности субъектам, выполняющимся от имени других учётных записей; множество видов совместного действия и функции, задающие текущие доступы и разрешённые совместные доступы к сущностям-не субъектам; множество сущностей-параметров, не являющихся субъектами, для каждого права доступа к сущности; средства описания системы мандатного контроля целостности; правила преобразования состояний с целью порождения первого субъекта в сеансе работы пользователя и прекращения доступа субъекта к сущности.

Один из часто встречаемых видов атак на КС называется *фишинг*. Его название происходит от английского *phishing*, или *password fishing*, переводимого дословно как *выуживание паролей* и означающего по существу получение доступа к конфиденциальной информации обманным путём, использующим слабости человеческого фактора. Наиболее распространённым способом обмана в фишинге является создание вебсайта (фишингового ресурса), внешне почти неотличимого от другого сайта — жертвы фишинговой атаки. Пользователь последнего может не заметить отличий от него в первом и выдать тому свой пароль со всеми вытекающими для себя тяжёлыми последствиями. Для успешной борьбы с фишинговыми атаками такого рода требуется научиться обнаруживать фишинговые ресурсы, а для этого надо изучить характерные признаки таких ресурсов и на их основе разработать эффективные методы оценивания степени опасности информационного ресурса и определения потенциально опасных ресурсов. Этому посвящён доклад молодых исследователей А. В. Милошенко, Т. М. Соловьёва, Р. И. Черняка и М. В. Шумской. В нём описаны многие характерные признаки фишингового ресурса, в том числе сходство графического и (или) текстового контента (его и атакуемого сайта), наличие ресурса в фишинговых базах, применение редко используемых параметров формата URL, подозрительные регистрационные данные ресурса, наличие ресурса на IP-адресе ранее выявленного фишингового ресурса, наличие графического изображения текста, использование слишком большого числа скриптов. Предложены методы оценивания степени опасности ресурса по его признакам, использующие аппарат булевых функций, нейронных сетей, линейной регрессии, и механизмы выбора потенциально опасных ресурсов по отношению к заданному ресурсу, основанные на сравнении доменных имён ресурсов. Авторы полагают, что система защиты каждого информационного ресурса от фишинговых атак должна включать генерацию списка потенциально опасных доменных имён, получение списка зарегистрированных и доступных потенциально опасных информационных ресурсов, определение степени опасности каждого потенциально опасного ресурса, пополнение фишинговых баз вновь обнаруженными опасными ресурсами.

Для борьбы со спамом, распространяемым бот-сетями, нередко применяют графические капчи — картинки с проверочным текстом из символов с искажённым изобра-

жением, таким, что текст легко читается человеком, но не распознаётся компьютером. С их помощью можно предотвращать автоматические регистрацию почтовых ящиков, отправку сообщений, скачивание файлов, массовую рассылку и другие операции, необходимые для осуществления автоматического распространения спама. KCAPTCHA — один из программных продуктов с открытым кодом, предназначенный для генерации графических капч. В нём применяются около 20 шрифтов и волновые алгоритмы искажения символов. В докладе М. Б. Абросимова и А. А. Маторина сообщается о разработанной ими компьютерной программе для автоматического распознавания текста графических капч. В ней реализован алгоритм распознавания отдельных символов на основе анализа контрольных точек скелета изображения и связей между ними. С помощью этой программы были проанализированы цифры и латинские буквы, генерируемые в искажённом виде системой KCAPTCHA, и получен следующий результат: точность распознавания лежит в пределах от 87 до 99 %; время распознавания одного символа на компьютере с тактовой частотой 2,4 ГГц составляет около 30 мс.

О средствах безопасности веб-сервисов, реализованных в интернет-системе поддержки муниципальных заказов администрации г. Красноярска, сообщается в докладе Д. Д. Кононова и С. В. Исаева. Аутентификация пользователя для входа в систему осуществляется по его имени и паролю. Дальнейшие действия по редактированию (добавлению, модификации, удалению) данных в системе подтверждаются цифровой подписью, обеспечивающей юридическую силу и подлинность документов. Сертификат ЭЦП выполнен в стандарте X.509. На стороне клиента хранится только идентификатор сессии, действительный до её завершения. Всю криптографическую службу в системе несёт криптопровайдер КриптоПРО CSP.

5. Математические основы информатики и программирования

В последнее время в теории вычислительной сложности интенсивное развитие получило новое направление, связанное с рассмотрением так называемых параметризированных задач и алгоритмов. *Параметризированная*, или *П-*, задача состоит в том, что для заданных языка $L \subseteq A^* \times \mathbb{N}$ и пары $(I, k) \in A^* \times \mathbb{N}$ требуется определить, является ли (I, k) элементом L. Для алгоритма, решающего эту задачу, I есть вход и k — параметр задачи. Этот алгоритм называется параметризированным, или *П-алгоритмом*, если его вычислительная сложность есть некоторая функция $t(n, k)$ от длины входа $n = |I|$ и параметра k. П-задача считается разрешимой с фиксированным параметром, или FPT-разрешимой (от *Fixed-Parameter Tractable*), если она может быть решена некоторым П-алгоритмом за время $t(n, k) = \mathrm{O}(n^{\mathrm{O}(1)} \cdot f(k))$ для функции f, зависящей только от k. П-алгоритм, решающий такую задачу, называется *FPT-алгоритмом*. В докладе В. В. Быковой вводится мера сложности алгоритмов, называемая частной эластичностью, по которой можно сравнивать между собой и классифицировать все П-алгоритмы, в частности FPT-алгоритмы. Для произвольной функции $z = z(x, y)$ её *частной эластичностью* $E_x(z)$ по аргументу x называется эластичность переменной z как функции только от x при любом фиксированном значении y, где под эластичностью функции одного аргумента понимается предел отношения относительного приращения этой функции к относительному приращению её аргумента, т. е. $E_x(z) = z'_x \cdot x/z$. Аналогично определяется $E_y(z) = z'_y \cdot y/z$. В случае $z = z(x, y) = q(x)f(y)$, что характерно для функции сложности П-алгоритма, $E_x(z) = E_x(q(x))$, $E_y(z) = E_y(f(y))$ — обычные эластичности. По величинам $E_x(q(x))$, $E_y(f(y))$ пара функций $q(x)$, $f(y)$ может быть отнесена к одной из пар сложностных классов SUBPOLY, POLY, SUBEXP, EXP, HYPEREXP. Для П-алгоритма с вычислительной сложностью $z(x, y) = q(x)f(y)$

эта пара классов характеризует сложность данного алгоритма и по длине входа x, и по значению параметра y, и П-алгоритм является FPT-алгоритмом, если и только если она есть (SUBPOLY, POLY).

В аналитической теории формальных языков непосредственно составляющих (нс-языков) найдено новое достаточное условие, при котором система символьных уравнений, сопоставляемая нс-грамматике, имеет решение в виде формальных степенных рядов (К. В. Сафонов, Д. А. Калугин-Балашов). Для их нахождения система линейными заменами переменных приводится к виду, в котором она решается методом последовательных приближений, и ряды в решении исходной системы получаются как линейные комбинации рядов в решении приведённой системы.

О денотационном описании семантики языка аспектно-ориентированного программирования AspectTalk сообщено в докладе Д. А. Стефанцова и А. Е. Крюковой. Оно состоит из множеств L, S, M синтаксических областей, доменов и семантических отображений соответственно. Каждая синтаксическая область есть язык, грамматика которого получается из грамматики AspectTalk заменой аксиомы некоторым нетерминалом. Доменами выражаются сущности языка. Среди них есть, например, домен процедур — функций из домена состояний в домен состояний, и домен программ — функций из домена входных последовательностей в домен выходных последовательностей. Семантические отображения (они из L в S) задают интерпретацию языка.

Разработке и реализации библиотеки ORM (Object-Relational Mapping) на языке C++ посвящён доклад Д. А. Стефанцова, Н. О. Ткаченко, Д. В. Чернова и Р. В. Шамовой. В нём авторы проанализировали недостатки существующих подобных разработок — библиотек ODB (наличие дополнительного транслятора, невозможность автоматической проверки корректности программ до трансляции) и Wt::Dbo (необходимость поддержания данных от своих пользователей, использование частей строк запроса на языке SQL, возможность SQL-инъекций) и сообщили о своей разработке — о библиотеке COT (C++ ORM on Templates), лишённой этих недостатков. В ней использовано метапрограммирование на шаблонах, отмеченное в названии библиотеки. Предварительные испытания показали, что к тому же COT в среднем на 9 % превосходит ODB по производительности.

6. Вычислительные методы в дискретной математике

В связи с вычислительной сложностью комбинаторных алгоритмов решения криптаналитических задач внимание ряда специалистов приковано к проблеме распараллеливания таких алгоритмов. На Sibecrypt на этот раз представлены два доклада по этой проблеме. В докладе О. С. Заикина сообщено о попытках распараллеливания SAT-решателей в грид-системах, а в докладе В. М. Фомичёва — о параллельной реализации метода встречи посередине на кластерных и распределенных вычислительных системах.

Знакомство с этими и другими результатами по распараллеливанию комбинаторных алгоритмов показывает, что, к сожалению, усилия их авторов ограничиваются пока методами распараллеливания алгоритмов по входным данным, которые (методы) на практике принципиально не способны давать ощутимого положительного эффекта, когда речь идёт о задачах криптанализа. Дело в том, что ускорение вычислений от подобных методов в лучшем случае пропорционально количеству используемых процессоров в вычислительной системе, и если размер входных данных алгоритма достигает, скажем, 1000 бит, что по меркам криптанализа на самом деле очень малое число, а количество процессоров в системе равно, скажем, 2^{50}, что на самом деле пока

недостижимо, то лучшее, что может дать метод распараллеливания по входным данным, — это сократить объём перебора возможных вариантов решения задачи с 2^{1000} до 2^{950}, т. е. до величины, которая практически столь же неохватная, как и для последовательного алгоритма.

В докладе А. А. Семёнова, И. В. Отпущенникова и С. Е. Кочемазова традиционно (для А. А. Семёнова и К°) пропагандируется подход к решению всех комбинаторных алгоритмических задач посредством SAT-решателей, т. е. путём сведения задачи к SAT-задаче и решения последней с помощью известных программных комплексов. На этот раз среди решаемых так задач упоминаются анализ недетерминированного автомата, поиск неподвижных точек и циклов автоматных отображений в генных сетях, задачи о назначениях и целочисленного линейного программирования.

Библиотеку программ *Boolean Functions* на языке C++ разработали Н. А. Коломеец и А. В. Павлов. Она работает с булевыми функциями, задаваемыми в АНФ, таблично и в форме следа. На данный момент в ней есть программы перевода функций из одной формы в другую, проверки двух функций на их аффинную эквивалентность, порождения функций, аффинно эквивалентных данной, проверки функции на свойство бент, порождения бент-функций от 4, 5, 6 переменных, построения всех бент-функций на минимальном расстоянии от заданной, построения кодов из векторов значений бент-функций со свойством: прибавлением любой бент-функции к коду образуется линейный код, и др.

Шевченко М. Ю. (в устном сообщении) сформулировал некоторые достаточные условия, при которых задача коммивояжёра сводится с полиномиальной сложностью к задаче о назначениях и тем самым может быть решена за полиномиальное время.

7. Прикладная теория автоматов

Предложена (Ю. В. Березовская, В. А. Воробьёв) каузально-сетевая модель поведения популяции автоматов — совокупности взаимодействующих вероятностных конечных автоматов. Это есть *каузальная*, или *K-*, *сеть*, представляющая собой маркированную сеть Петри, в которой дополнительно для каждого перехода задана интенсивность его срабатывания как функция от маркировки входных позиций перехода. В ней позиции — это возможные состояния автоматов популяции, а вектор маркировки позиций своими компонентами задаёт количества автоматов, находящихся в соответствующих позициям состояниях. В отличие от сети Петри, в качестве маркеров позиций и весов дуг в K-сети допускаются положительные действительные числа, что позволяет маркировать позиции и помечать дуги вероятностями состояний и переходов между ними в автоматах. В качестве примера приводится K-сеть известной популяции «хищник — жертва» в ограниченной экологической нише.

Конечный автомат A называется скелетным, если отношение взаимной достижимости на множестве его состояний тождественно. Нумерация состояний автомата A числами из начального отрезка натурального ряда правильная, если состояния, достижимые из данного состояния, имеют меньшие, чем у него, номера. В докладе В. Н. Салия доказано, что 1) в автомате A существует правильная нумерация состояний, если и только если автомат A скелетный; 2) для автоматов A и B с $\mathrm{Sub}A \cong \mathrm{Sub}B$ число t состояний в B не меньше числа d классов взаимной достижимости в A и для любого автомата A существует скелетный автомат B, такой, что $\mathrm{Sub}A \cong \mathrm{Sub}B$ и $t = d$. Показано также, как путём удаления из диаграммы переходов автомата A минимального числа дуг можно получить скелетный автомат.

Автоматный шифр Закревского допускает аппаратную реализацию в виде перестраиваемого автомата T, в котором ключ шифра задаётся парой (k, s_0), где k — настройка и s_0 — начальное состояние. Автомат T устроен так, что в нём в каждый момент времени функция переходов выбирается из двух вариантов в зависимости от настройки и текущих входного символа и состояния. В докладе В. Н. Тренькаева, автора этой реализации, предложена атака аппаратного сбоя на неё с выбором открытого текста. Целью атаки является построить инициальный сильносвязный автомат Z (обычный, не перестраиваемый), который преобразует входные слова в выходные так же, как и автомат T с неизвестными, но фиксированными k, s_0. Суть атаки следующая. В автомате T вызывается неисправность, при которой функция переходов в нём выбирается в некотором одном известном варианте. С помощью установочного эксперимента определяется текущее состояние s этого неисправного автомата, после чего неисправность ликвидируется и с автоматом T проводится условный идентификационный эксперимент по определению неизвестной таблицы переходов эквивалентного ему автомата Z.

8. Прикладная теория графов

Графовые модели вычислительных систем (ВС) остаются по-прежнему наиболее эффективным средством в анализе и синтезе отказоустойчивых ВС. В роли адекватных моделей ВС, устойчивых к отказам компонент в системе и связей между ними, часто выступают соответственно вершинные и рёберные расширения требуемой кратности графа системы. Их исследования занимают важное место в тематике конференции Sibecrypt.

В докладе М. Б. Абросимова и П. П. Бондаренко рассмотрены минимальные вершинные 1-расширения циклов, в которых (циклах) одна вершина одного типа, а остальные — другого типа. Показано, что для цикла с n вершинами такие расширения имеют $(3n + 4)/2$ рёбер при чётном n и $(3n + 5)/2$ рёбер при нечётном n. Приведены примеры их при всех $n \in \{4k, 4k+1, 4k+2, 4k+3\}$, а также количества неизоморфных из них при $n = 2, 3, \ldots, 8$, подсчитанные путём порождения их всех на компьютере.

М. Б. Абросимов и А. А. Долгов в своём докладе показали, что диграфы из семейств Стокмейера не являются точными вершинными 1-расширениями никаких орграфов, из чего следует, что если есть орграф с тремя или более вершинами и с двумя или более неизоморфными точными вершинными 1-расширениями, то число вершин в нём не меньше 13, и эти его расширения нереконструируемые и не входят ни в одно известное семейство нереконструируемых орграфов.

В докладе М. Б. Абросимова и Д. Д. Комарова минимальное рёберное 1-расширение графа из двух звёзд с соединёнными центрами строится путём соединения рёбрами каждой из двух выбранных вершин степени 1, расстояние между которыми равно 3, со всеми вершинами степени 1, расстояния до которых от неё равно 3, и показано, как такое же расширение можно построить для стройного дерева, являющегося объединением некоторого числа цепей длины 2 и не менее двух цепей длины 1.

Наконец, в докладе М. Б. Абросимова и О. В. Моденовой рассмотрены свойства орграфа G, сохраняемые в его точном — G_t и минимальном — G_m вершинных k-расширениях. Показано, в частности, что 1) отношения смежности в G и G_t (G_m) являются одновременно рефлексивными либо антирефлексивными; 2) симметризация G_t является точным вершинным k-расширением симметризации G, симметризация G_m является вершинным k-расширением симметризации G; 3) если G — диграф с числом вершин больше 1, то G_t, если оно существует, есть также диграф; 4) дополнение G_t является

точным вершинным k-расширением дополнения G; 5) обращение G_t (G_m) является точным (соответственно минимальным) вершинным k-расширением обращения G.

В рамках проблемы синтеза компактных структур ВС рассмотрена задача синтеза компактных графов — регулярных графов минимального диаметра (В. А. Мелентьев). Сформулирован алгоритм синтеза таких графов, использующий представление графа проекциями — слоями вершин S_1, S_2, ..., достижимых из некоторой начальной вершины простыми путями длины 1, 2, ... соответственно. Получены нижняя и верхняя оценки для числа вершин в компактных графах заданной степени и заданного диаметра d, имеющих в себе цикл длины k для $3 \leqslant k \leqslant 2d-1$.

Важнейшей числовой характеристикой любого графа является его древовидная ширина, являющаяся мерой древовидности графа — того, насколько граф близок к дереву. Графы с ограниченной древовидной шириной образуют класс так называемых частичных k-деревьев. Многие NP-трудные задачи теории графов полиномиально разрешимы на частичных k-деревьях. *Древовидная ширина* $\mathrm{tw}(G)$ графа G определяется через понятие дерева декомпозиции этого графа, которое есть дерево T со следующими свойствами: 1) его вершины являются подмножествами множества вершин в G, образующими его покрытие; 2) для всякого ребра графа G имеется хотя бы одна вершина в T, содержащая обе вершины этого ребра; 3) для каждой вершины v графа G подмножество вершин дерева T, содержащих вершину v, порождает поддерево в T. Ширина дерева T есть $\mathrm{w}(T) = \max(|x|-1)$, где max берётся по всем вершинам x в T, и $\mathrm{tw}(G) = \min \mathrm{w}(T)$, где min берётся по всем деревьям декомпозиции T графа G. В докладе В. В. Быковой дан краткий обзор основных свойств древовидной ширины графа, методов её вычисления (точных и приближённых) для произвольных и некоторых специальных графов — хордальных, последовательно-параллельных и др., её нижних и верхних оценок, вычисляемых через другие параметры графа — наименьшую степень вершины, число вершинной связности, плотность, хроматическое число.

Проблеме факторизации графов посвящён доклад Е. А. Кармановой со следующими результатами: 1) связный граф тогда и только тогда является фактор-графом m-рёберной цепи, когда в нём есть обход длины m; 2) связный граф с m рёбрами является фактор-графом цепи P_{2m-1}; 3) для связного графа G с m рёбрами минимальное число $p(G)$ рёбер цепи, фактор-графом которой является данный граф, лежит в границах $m \leqslant p(G) \leqslant 2m-2$; 4) для дерева T с диаметром d и m рёбрами $p(T) = 2m-d$; 5) для звезды S_m верно $p(S_m) = 2m-2$.

В докладе А. А. Кочкарова, Л. И. Сенниковой и Н. Н. Болурова рассмотрены свойства предфрактальных графов. Указаны нижняя и верхняя оценки для числа точек сочленения и для числа мостов в предфрактальном графе, выраженные через длину траектории его получения, количество рёбер и число соответственно точек сочленения и мостов в затравке.

Улучшенные верхние оценки экспонентов (показателей) ε примитивных графов приведены в докладе В. М. Фомичёва. Для n-вершинных орграфов, где $n > 2$, с двумя контурами без общих вершин и с взаимно простыми длинами l и λ эти оценки линейные, а именно: $\varepsilon \leqslant l\lambda - 2l - 3\lambda + 3n$, если контуры не пересекаются, и $\varepsilon \leqslant l\lambda - l - 3\lambda + h + 2n$, если контуры имеют h общих вершин. Для неориентированнного графа с $n > 1$ вершинами и с наибольшей длиной l простого цикла нечётной длины $\varepsilon \leqslant 2n - l - 1$, а если простые циклы нечётных длин содержат все вершины графа, то $\varepsilon \leqslant n-1$. Показано также, что абсолютная оценка $\varepsilon \leqslant n^2 - 2n + 2$ для экспонента ε любого примитивного n-вершинного орграфа, установленная Виландтом, достижима на графах Виландта, и только на них. Здесь под графом Виландта под-

разумевается гамильтонов контур с дополнительной дугой между некоторыми двумя вершинами, находящимися в контуре на расстоянии 2. Множество всех таких графов с n вершинами состоит из $n!$ изоморфных графов. Для всех остальных n-вершинных примитивных орграфов с нечётным $n > 3$ верна оценка $\varepsilon \leqslant n^2 - 3n + 4$. Для неориентированных примитивных графов с $n > 1$ вершинами абсолютная оценка для экспонента есть $\varepsilon \leqslant 2n - 2$. Она достигается на графах, состоящих из гамильтоновой цепи и петли на одном из её концов, и только на них. Их множество состоит из $n!$ изоморфных графов.

Пару (S, δ) с отображением δ на конечном множестве S называют динамической системой, элементы в S — её состояниями. В её графе вершинами являются состояния, а дуги идут из вершин s в вершины $\delta(s)$. Он распадается на компоненты связности, каждая из которых представляет собой контур с входящими в него деревьями. Контуры всех компонент связности называются *аттракторами* системы. В докладе А. В. Власовой сформулирован критерий принадлежности состояния аттрактору в динамической системе (B^n, θ), где B^n есть множество всех булевых векторов длины n, и если каждый вектор $v \in B^n$ рассматривать как цикл, в котором первая компонента следует за последней, то $\theta(v)$ получается заменой в v каждой биграммы 10 биграммой 01. В таком векторе v максимальные из k-грамм $00\ldots0$ и $11\ldots1$ для $k \geqslant 2$ называются его соответственно 0- и 1-блоками длины $k-1$. Суммы длин всех 0-блоков и 1-блоков в v обозначаются $p_0(v)$ и $p_1(v)$ соответственно. Доказано, что состояние v динамической системы (B^n, θ) принадлежит аттрактору, если и только если $p_0(v) = 0$ или $p_1(v) = 0$; в этом случае в аттракторе, содержащем v, следующее (по стрелке) состояние получается из предыдущего циклическим сдвигом на одну компоненту соответственно влево или вправо.

В докладе В. С. Грунского и С. В. Сапунова рассмотрена задача определения своего местонахождения мобильным агентом (МА), блуждающим по графу G с помеченными вершинами. Предложено её решение с использованием понятий топологического идентификатора (ТИ) и диагностического тестового графа (ДТГ). Если через S_g обозначен подграф в G, порождённый всеми вершинами в G, достижимыми из вершины g, то ТИ вершины g есть помеченный граф D_g, такой, что для любой вершины h в G изоморфизм $D_g \cap S_g \cong D_g \cap S_h$ существует тогда и только тогда, когда $g = h$, где $D_g \cap S_h$ есть наибольший связный подграф в G, содержащий вершину g и изоморфно вложимый в S_h с отображением g в h. ДТГ получается отождествлением в ТИ D_g для всех g в G их одинаково помеченных инициальных вершин и преемников каждой вершины, имеющих одинаковые метки с заменой получающихся кратных дуг одной. Рассматриваемая задача решается в два этапа. Сначала по G строится ДТГ. Затем МА, стартуя из неизвестной ему вершины h графа G, на каждом шаге проверяет наличие в G путей, совпадающих по разметке с путями в ДТГ из его инициальных вершин, последовательно сокращая по результатам проверок множество возможных стартовых вершин до единственной. Утверждается, что на графах, в которых в замкнутой 1-окрестности каждой вершины все вершины помечены разными символами, данный алгоритм имеет полиномиальную сложность.

ЛИТЕРАТУРА

1. Тез. докл. X Сибирской научной школы-семинара с международным участием «Компьютерная безопасность и криптография» — Sibecrypt'11 (Томск, ТГУ, 5–9 сентября 2011 г.) // Прикладная дискретная математика. Приложение. 2011. № 4. 111 с.

СВЕДЕНИЯ ОБ АВТОРАХ

АБРОСИМОВ Михаил Борисович — доцент, кандидат физико-математических наук, доцент Саратовского государственного университета им. Н. Г. Чернышевского, г. Саратов. E-mail: **mic@rambler.ru**

АГИБАЛОВ Геннадий Петрович — профессор, доктор технических наук, заведующий кафедрой защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск. E-mail: **agibalov@isc.tsu.ru**

БЕРЕЗОВСКАЯ Юлия Владимировна — ассистент кафедры информационных технологий Северного (Арктического) федерального университета им. М. В. Ломоносова, г. Архангельск. E-mail: **myumla.myu@gmail.com**

ВОРОБЬЕВ Владимир Анатольевич — профессор, доктор технических наук, профессор кафедры прикладной математики Северного (Арктического) федерального университета им. М. В. Ломоносова, г. Архангельск. E-mail: **vva100@atnet.ru**

ВОРОПАЕВ Антон Николаевич — аспирант Петрозаводского государственного университета, г. Петрозаводск. E-mail: **voropaev@psu.karelia.ru**

ЗУБОВ Анатолий Юрьевич — кандидат физико-математических наук, доцент, старший научный сотрудник Московского государственного университета им. М. В. Ломоносова, г. Москва. E-mail: **Zubovanatoly@yandex.ru**

КИТЮКОВ Вячеслав Вячеславович — студент Московского авиационного института, г. Москва. E-mail: **atum89@gmail.ru**

КОЧКАРОВ Азрет Ахматович — кандидат физико-математических наук, заместитель директора Научно-образовательного центра Института прикладной математики им. М. В. Келдыша РАН, г. Москва. E-mail: **akochkar@gmail.com**

ПАРВАТОВ Николай Георгиевич — кандидат физико-математических наук, доцент Национального исследовательского Томского государственного университета, г. Томск. E-mail: **parvatov@mail.tsu.ru**

ПОТТОСИН Юрий Васильевич — доцент, кандидат физико-математических наук, ведущий научный сотрудник Объединенного института проблем информатики НАН Беларуси, г. Минск. E-mail: **pott@newman.bas-net.by**

СЕННИКОВА Людмила Игоревна — старший преподаватель Ставропольского института управления, г. Ставрополь. E-mail: **s-ludhen@yandex.ru**

СЕРГЕЕВ Игорь Сергеевич — кандидат физико-математических наук, старший научный сотрудник Московского государственного университета им. М. В. Ломоносова, г. Москва. E-mail: **isserg@gmail.com**

СМЫШЛЯЕВ Станислав Витальевич — аспирант Московского государственного университета им. М. В. Ломоносова, г. Москва. E-mail: **smyshsv@gmail.com**

ШИЛИН Илья Анатольевич — доцент, кандидат физико-математических наук, доцент кафедры математического моделирования Московского авиационного института и кафедры высшей математики Московского государственного гуманитарного университета им. М. А. Шолохова, г. Москва. E-mail: **ilyashilin@li.ru**

АННОТАЦИИ СТАТЕЙ НА АНГЛИЙСКОМ ЯЗЫКЕ

Parvatov N. G. **CONSTRUCTION OF MAXIMAL CLONES IN THE SET OF POINT FUNCTIONS ON INTERVAL SEMILATTICE.** The description problem for clones in the sets of all point and all minimal point functions on a semilattice is considered. Examples of maximal such clones on the interval semilattice of a lattice are given.
Keywords: *clone, upper semilattice, interval semilattice, interval lattice, point function, minimal point function.*

Smyshlyaev S. V. **LOCALLY INVERTIBLE BOOLEAN FUNCTIONS.** The property of local invertibility of Boolean functions is considered. A number of necessary conditions of local invertibility is proven; they can be used to construct functions that are not locally invertible. We prove a new criterion of local invertibility connecting this property with some properties of Boolean functions with barrier.
Keywords: *perfectly balanced functions, barriers of Boolean functions, local invertibility, cryptography.*

Shilin I. A., Kityukov V. V. **HOMOMORPHIC STABILITY OF PAIRS OF SMALL ORDER GROUPS.** Let the orders of groups G and H aren't greater than 12. We apply a computer program created by the authors for researching algebraic structure of the set of $G \longrightarrow H$ homomorphism's images.
Keywords: *finite groups homomorphic stability.*

Zubov A. U. **ALMOST PERFECT CIPHERS AND AUTHENTICATION CODES.** Constructions of almost-perfect ciphers combining secrecy and authentication functions with economical key expenditure are proposed under equiprobable choice of key and plaintext.
Keywords: *almost-perfect cipher, authentication code.*

Abrosimov M. B. **ON MINIMAL VERTEX 1-EXTENSIONS OF SPECIAL TYPE GRAPH UNION.** In 2001, it was conjectured that the minimal vertex 1-extension of a graph $G + G^*$, where G^* is a minimal vertex 1-extension of graph G, is unique up to isomorphism and has the form $G^* + G^*$. We construct two counterexamples to this conjecture showing that, in general, it is wrong. Also, we show that the statement is true for many graphs.
Keywords: *graph, minimal vertex extension, exact vertex extension, fault tolerance.*

Voropaev A. N. **MULTIPLICITIES OF SUMS IN THE EXPLICIT FORMULAE FOR COUNTING FIXED LENGTH CYCLES IN UNDIRECTED GRAPHS.** An explicit formula for counting k-cycles in graphs is the combination of sums corresponding to the shapes of closed k-walks. It was shown that the maximum multiplicity of a sum in the formula is $[k/2]$ for, starting with $k = 8$. In this work, we study the maximum sum multiplicity for some families of graphs: bipartite, triangle-free, planar, maximum vertex degree three, and their intersections. When k is large, the biparticity and degree boundednesses are the only properties which decrease the maximum sum multiplicity by 1, providing $k \equiv 2, 3 \pmod 4$. Some combinations of properties in the case of $k \leqslant 20$ yield the decrease by 1 or 2.
Keywords: *counting cycles in graphs, shapes of closed walks, prism graphs.*

Kochkarov A. A., Sennikova L. I. **SOME PREFRACTAL GRAPH'S CONNECTIVITY CHARACTERISTICS ESTIMATIONS.** The paper is devoted to the research of prefractal graph's connectivity characteristics. Some estimations being achievable on their boundaries are received for the number of prefractal graph's cutpoints and bridges. The estimations are obliged to the self-similarity of prefractal graphs.
Keywords: *self-similar graphs, fractal and prefractal graphs, network systems, cutpoints and bridges.*

Pottosin Yu. V. **STATE ASSIGNMENT IN A DISCRETE AUTOMATON TARGETING AN IMPLEMENTING LOW POWER CIRCUIT.** The problem of the state assignment in a discrete automaton aimed to decrease the switching activity of memory elements in an implementing circuit is considered. A method for solving this problem based on an approach connected with the "desirable neighborhood" method is proposed.
Keywords: *discrete automaton, state assignment.*

Sergeev I. S. **REGULAR ESTIMATES FOR THE COMPLEXITY OF POLYNOMIAL MULTIPLICATION AND TRUNCATED FOURIER TRANSFORM.** In the present paper, some polynomial multiplication circuits being efficient either in complexity and depth or in complexity and memory size are proposed. Consequently, for instance, the multiplication of polynomials of the sum degree $n-1$, where $n = 2^{n_1} + \ldots + 2^{n_s}$, $n_1 > \ldots > n_s$, over a ring with invertible 2 can be implemented via $M(n_1) + \ldots + M(n_s) + {}+\mathrm{O}(n)$ arithmetic operations over the ring with the depth $\max\limits_i\{D(n_i)\} + \mathrm{O}(\log n)$, where $M(k)$ and $D(k)$ are respectively the complexity and the depth of the modulo $x^{2^k}+1$ multiplication circuit. As another example, the truncated DFT of order n (i. e. the DFT of order $2^{\lceil \log_2 n \rceil}$ reduced to the vectors of dimension n) can be implemented by a circuit of complexity $1{,}5n \log_2 n + \mathrm{O}(n)$ and memory size $n+1$.
Keywords: *arithmetic circuits, complexity, depth, memory size, multiplication, Discrete Fourier Transform.*

Berezovsky Yu. V., Vorob'ev V. A. **POPULATIONS OF INTERACTING AUTOMATA.** The population of automata is a model of collective behavior of automata. Modeling of population dynamics is implemented by a Causal Petri Net. The places in it represent the states of automata. The net marking specifies a number of automata that are in corresponding states. The transitions in the net represent events that result from the joint actions of the elements in the population. For each transition, a value is specified defining the probability (rate) of the transition response so that a system of differential equations can be built. These equations describe the dynamics of the average number of automata in places under logical conditions specified by Petri net. The numerical solution of the system is obtained by using a computer simulation.
Keywords: *population of automata, causal net, Petri net, mean value dynamics, modeling.*

Agibalov G. P. **SIBECRYPT'11 REVIEW.** This is a survey of lectures and papers presented at 10th Siberian Workshop SIBECRYPT devoted to mathematical problems in computer security and cryptography and held in Tomsk, Russia, in September 5–9, 2011.
Keywords: *applied discrete mathematics, computer security, cryptography.*

Журнал «Прикладная дискретная математика» включен в перечень ВАК рецензируемых российских журналов, в которых должны быть опубликованы основные результаты диссертаций, представляемых на соискание учёной степени кандидата и доктора наук, а также в перечень журналов, рекомендованных УМО в области информационной безопасности РФ в качестве учебной литературы по специальности «Компьютерная безопасность».

Журнал «Прикладная дискретная математика» распространяется по подписке; его подписной индекс 38696 в объединённом каталоге «Пресса России». Полнотекстовые электронные версии вышедших номеров журнала доступны на его сайте vestnik.tsu.ru/pdm и на Общероссийском математическом портале www.mathnet.ru. На сайте журнала можно найти также и правила подготовки рукописей статей в журнал.

Тематика публикаций журнала:

- *Теоретические основы прикладной дискретной математики*
- *Математические методы криптографии*
- *Математические методы стеганографии*
- *Математические основы компьютерной безопасности*
- *Математические основы надежности вычислительных и управляющих систем*
- *Прикладная теория кодирования*
- *Прикладная теория автоматов*
- *Прикладная теория графов*
- *Логическое проектирование дискретных автоматов*
- *Математические основы информатики и программирования*
- *Вычислительные методы в дискретной математике*
- *Дискретные модели реальных процессов*
- *Математические основы интеллектуальных систем*
- *Исторические очерки по дискретной математике и ее приложениям*