

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2009

№4(6)

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



ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**РЕДАКЦИОННАЯ КОЛЛЕГИЯ ЖУРНАЛА
«ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА»**

Агибалов Г. П., д-р техн. наук, проф. (председатель); Девянин П. Н., д-р техн. наук, проф. (зам. председателя); Парватов Н. Г., канд. физ.-мат. наук, доц. (зам. председателя); Черемушкин А. В., д-р физ.-мат. наук, проф. (зам. председателя); Панкратова И. А., канд. физ.-мат. наук, доц. (отв. секретарь); Алексеев В. Б., д-р физ.-мат. наук, проф.; Бандман О. Л., д-р техн. наук, проф.; Евдокимов А. А., канд. физ.-мат. наук, проф.; Евтушенко Н. В., д-р техн. наук, проф.; Закревский А. Д., д-р техн. наук, проф., чл.-корр. НАН Беларуси; Костюк Ю. Л., д-р техн. наук, проф.; Логачев О. А., канд. физ.-мат. наук, доц.; Матросова А. Ю., д-р техн. наук, проф.; Микони С. В., д-р техн. наук, проф.; Салий В. Н., канд. физ.-мат. наук, проф.; Сафонов К. В., д-р физ.-мат. наук, проф.; Фомичев В. М., д-р физ.-мат. наук, проф.; Чеботарев А. Н., д-р техн. наук, проф.; Шоломов Л. А., д-р физ.-мат. наук, проф.

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.

ООО «Издательство научно-технической литературы»

634050, Томск, пл. Ново-Соборная, 1, тел. (3822) 533-335

Редактор *Н. И. Шидловская*

Верстка *Д. А. Стефанцова*

Изд. лиц. ИД. №04000 от 12.02.2001. Подписано к печати 14.12.2009.
Формат 60 × 84 $\frac{1}{8}$. Бумага офсетная. Печать офсетная. Гарнитура «Таймс».
Усл. п. л. 13,29. Уч.-изд. л. 14,88. Тираж 300 экз. Заказ №13.

Отпечатано в типографии «М-Принт», г. Томск, ул. Пролетарская, 38/1

СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

Коломеец Н. А., Павлов А. В. Свойства бент-функций, находящихся на минимальном расстоянии друг от друга	5
Парватов Н. Г. Об инвариантах некоторых классов квазимонотонных функций на полурешётке	21
Семёнов А. А. О преобразованиях Цейтина в логических уравнениях	28
Черемушкин А. В. Рекурсивный способ построения семейств без перекрытий	51

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

Пестунов А. И. Дифференциальный криптоанализ блочного шифра MARS	56
Федюкович В. Е. Протокол аргумента знания слова кода Гоппы и ошибки оценочного веса	64

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

Евсеев А. А., Нечаева О. И. Клеточно-автоматное моделирование диффузионных процессов на триангуляционных сетках	72
Скобелев В. В. «Ленточная» теорема и ее приложения	84

ПРИКЛАДНАЯ ТЕОРИЯ АВТОМАТОВ

Громов М. Л., Евтушенко Н. В. Синтез условных различающих экспериментов для автоматов с недетерминированным поведением	90
--	----

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Колесникова С. И. Модификация метода анализа иерархий для динамических наборов альтернатив	102
--	-----

ИНФОРМАЦИЯ ДЛЯ АВТОРОВ

Стефанцов Д. А. Инструкции и рекомендации по подготовке статей в формате LaTeX для журнала «Прикладная дискретная математика»	110
Тематика журнала	121

СВЕДЕНИЯ ОБ АВТОРАХ	124
---------------------------	-----

АННОТАЦИИ СТАТЕЙ НА АНГЛИЙСКОМ ЯЗЫКЕ	125
--	-----

CONTENTS

THEORETICAL BACKGROUNDS OF APPLIED DISCRETE MATHEMATICS

Kolomeec N. A., Pavlov A. V. Properties of bent functions with minimal distance	5
Parvatov N. G. About invariants for some classes of quasimonotonic functions on a semilattice	21
Semenov A. A. About Tseitin transformation in logical equations	28
Cheremushkin A. V. A recursive algorithm for cover-free family construction	51

MATHEMATICAL METHODS OF CRYPTOGRAPHY

Pestunov A. I. Differential Cryptanalysis of the MARS Block Cipher	56
Fedyukovych V. E. Argument of knowledge protocol for a Goppa codeword and for an error of a bounded weight	64

COMPUTATIONAL METHODS IN DISCRETE MATHEMATICS

Evseev A. A., Nechaeva O. I. Cellular automata simulation on surface triangulation for diffusion processes	72
Skobelev V. V. “Strings” theorem and its applications	84

APPLIED THEORY OF AUTOMATA

Gromov M. L., Yevtushenko N. V. Adaptive tests derivation for nondeterministic automata	90
---	----

MATHEMATICAL BACKGROUNDS OF INTELLIGENT SYSTEMS

Kolesnikova S. I. Modification of hierarchies analysis method for the dynamic set of alternatives	102
---	-----

INFORMATION FOR AUTHORS

Stephantsov D. A. Instructions and recommendations for authors to prepare articles in LaTeX format for “Applied Discrete Mathematics” journal	110
Topics of the journal	122

BRIEF INFORMATION ABOUT THE AUTHORS	124
---	-----

PAPER ABSTRACTS	125
-----------------------	-----

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

DOI 10.17223/20710410/6/1

УДК 519.7

СВОЙСТВА БЕНТ-ФУНКЦИЙ, НАХОДЯЩИХСЯ НА МИНИМАЛЬНОМ РАССТОЯНИИ ДРУГ ОТ ДРУГА¹

Н. А. Коломеец, А. В. Павлов

*Новосибирский государственный университет, г. Новосибирск, Россия***E-mail:** nkolomeec@gmail.com, apavlov.nsk@gmail.com

В работе получено минимальное расстояние Хэмминга в классе бент-функций от n переменных, равное $2^{n/2}$. Доказано, что бент-функции находятся на минимальном расстоянии тогда и только тогда, когда они различаются на линейном многообразии и обе функции на нем аффинны. Описан алгоритм построения всех бент-функций на минимальном расстоянии от заданной бент-функции. Приведены экспериментальные данные для бент-функций от малого числа переменных.

Ключевые слова: бент-функция, CDMA, OFDM.

Введение

Данная работа посвящена исследованию метрических характеристик класса бент-функций. Бент-функции — это булевы функции от четного числа переменных, максимально удаленные от класса аффинных функций. Впервые бент-функции были введены еще в 60-х годах XX века О. Ротхаузом, и до сих пор интерес к ним не ослабевает. Причиной этого служат как многочисленные теоретические и практические приложения, так и множество открытых вопросов, с ними связанных (см. подробнее в [1]).

Задача исследования метрических свойств бент-функций возникает в теории кодирования и находит свое применение в системах коллективного доступа [2], таких, как стандарты CDMA — Code Division Multiple Access (множественный доступ с кодовым разделением каналов) и OFDM — Orthogonal Frequency Division Multiplexing (ортогональное частотное мультиплексирование). Данные стандарты используют бент-функции для построения кодов постоянной амплитуды (constant-amplitude codes), что позволяет предельно снизить коэффициент отношения пиковой и средней мощностей сигнала (PARP — peak-to-average power ratio). Такие коды состоят из векторов значений бент-функций. Таким образом, являются актуальными задачи построения таких кодов с различными кодовыми расстояниями, что непосредственно связано с исследованием метрической структуры класса бент-функций.

Данная работа имеет следующую структуру. В п. 1 показывается, что минимальное расстояние в классе бент-функций равно $2^{n/2}$, и бент-функции, находящиеся на этом расстоянии друг от друга, должны отличаться на линейном многообразии, причем обе функции должны быть на нем аффинны. Заметим, что этим результатам

¹Работа выполнена при финансовой поддержке гранта Президента РФ для молодых российских ученых (грант МК-1250.2009.1) и РФФИ (проект № 08-01-00671).

предшествовали исследования В. В. Яценко [3] и К. Карле [4] построения различных бент-функций по заданной бент-функции. В п. 2 предлагаются простые способы построения бент-функций на минимальном расстоянии от заданной бент-функции. В п. 3 приводятся примеры бент-функций, для которых существуют бент-функции на минимальном расстоянии. В п. 4 используется аффинная классификация бент-функций для исследования существования бент-функций на минимальном расстоянии для бент-функций от малого количества переменных. В п. 5 приводятся алгоритм построения всех бент-функций на минимальном расстоянии от заданной бент-функции, а также экспериментальные данные для бент-функций от малого числа переменных.

Приведем известные определения и факты, имеющие отношения к бент-функциям.

Под расстоянием между булевыми функциями f и g мы подразумеваем расстояние Хэмминга, обозначим его как $\text{dist}(f, g)$. Через E^n будем обозначать n -мерный куб, через \mathcal{F}_n — множество булевых функций от n переменных.

Определение 1. Множество $L \subseteq E^n$ называется *линейным многообразием* в E^n , если $L = x_0 \oplus U$, где x_0 — элемент из E^n , а U — подпространство в E^n .

Определение 2. Преобразование $W_f : E^n \rightarrow Z$ следующего вида называется *преобразованием Уолша — Адамара* функции f :

$$W_f(w) = \sum_{x \in E^n} (-1)^{f(x) \oplus \langle w, x \rangle}.$$

Число $W_f(w)$ называется *коэффициентом Уолша — Адамара* в точке w (или просто коэффициентом Уолша).

Известно, что две различные функции не могут иметь одинаковые коэффициенты Уолша. Также справедливо равенство Парсеваля: пусть $f \in \mathcal{F}_n$. Тогда имеет место следующее равенство:

$$\sum_{w \in E^n} W_f^2(w) = 2^{2n}.$$

Имеет место формула свертки: пусть $f, g \in \mathcal{F}_n$, тогда

$$W_{f \oplus g}(w) = \frac{1}{2^n} \sum_{x \in E^n} W_f(x) W_g(x \oplus w).$$

Определение 3 (альтернативное определение бент-функций). Булева функция f от четного числа переменных называется *бент-функцией*, если все ее коэффициенты Уолша равны $\pm 2^{n/2}$.

Класс бент-функций от n переменных будем обозначать как \mathfrak{B}_n , а минимальное расстояние между функциями из него — как $d(\mathfrak{B}_n)$.

Определение 4. Пусть f — бент-функция от n переменных. Определим *дуальную функцию* $\tilde{f}(w)$, исходя из равенства

$$(-1)^{\tilde{f}(w)} \cdot 2^{n/2} = W_f(w).$$

Стоит заметить, что дуальная к бент-функции функция также является бент-функцией.

Так как расстояние Хэмминга берется между бент-функциями, то здесь и далее будем считать, что n (так будет обозначаться количество переменных функции) является четным натуральным числом.

1. Критерий расположения бент-функций на минимальном расстоянии

Получим минимальное расстояние между бент-функциями и описание всех бент-функций на минимальном расстоянии от заданной бент-функции.

Обозначим через $D(f, g)$ множество значений аргументов, на которых функции f и g от n переменных отличаются. Будем говорить, что $D(f, g)$ — это *множество разностей* функций f и g . Заметим, что $|D(f, g)| = \text{dist}(f, g)$. Через $I_D(x)$ мы будем обозначать индикатор множества D , то есть булеву функцию, которая принимает значение 1 на всех элементах из D , и только на них. Через $\text{rank } D$ обозначим размерность линейной оболочки векторов из D .

Определение 5. Булева функция f от n переменных *аффинна на множестве* $D \subseteq E^n$, если для некоторых $w_0 \in E^n$, $c \in E$ и для любого $x \in D$ выполняется тождество $f(x) = \langle w_0, x \rangle \oplus c$.

Для удобства введем следующее обозначение:

$$a_{f,g}(w) = \sum_{x \in D(f,g)} (-1)^{f(x) \oplus \langle w, x \rangle}.$$

Утверждение 1. Для любого $w \in E^n$ справедливо $|a_{f,g}(w)| \leq |D(f, g)|$, причем равенство достигается тогда и только тогда, когда функция $f(x) \oplus \langle w, x \rangle$ является константой на $D(f, g)$.

Утверждение 2. Пусть $f, g \in \mathcal{F}_n$. Тогда для любого w из E^n выполняется

$$W_f(w) - W_g(w) = 2a_{f,g}(w).$$

Доказательство. Запишем коэффициенты Уолша функций f и g следующим образом:

$$\begin{aligned} W_f(w) &= \sum_{x \in D(f,g)} (-1)^{f(x) \oplus \langle w, x \rangle} + \sum_{x \in E^n \setminus D(f,g)} (-1)^{f(x) \oplus \langle w, x \rangle}, \\ W_g(w) &= \sum_{x \in D(f,g)} (-1)^{g(x) \oplus \langle w, x \rangle} + \sum_{x \in E^n \setminus D(f,g)} (-1)^{g(x) \oplus \langle w, x \rangle}. \end{aligned}$$

Отсюда разность этих коэффициентов равна

$$W_f(w) - W_g(w) = \sum_{x \in D(f,g)} (-1)^{f(x) \oplus \langle w, x \rangle} - \sum_{x \in D(f,g)} (-1)^{g(x) \oplus \langle w, x \rangle} = 2a_{f,g}(w). \blacksquare$$

Имеет место следующая нижняя оценка минимального расстояния.

Лемма 1 (о расстоянии). Для всех четных n выполняется неравенство

$$d(\mathfrak{B}_n) \geq 2^{n/2}.$$

Доказательство. Предположим, что существуют различные $f, g \in \mathfrak{B}_n$, такие, что $\text{dist}(f, g) < 2^{n/2}$. Из предыдущего утверждения имеем: $W_f(w) - W_g(w) = 2a_{f,g}(w)$ для любого w из E^n . Так как f и g — бент-функции, то все их коэффициенты Уолша — Адамара по модулю равны $2^{n/2}$, следовательно,

$$a_{f,g}(w) \in \{0, 2^{n/2}, -2^{n/2}\}.$$

По утверждению 1 $|a_{f,g}(w)| \leq |D(f, g)| < 2^{n/2}$ для всех $w \in E^n$. Получается, что $a_{f,g}(w) = 0$, следовательно, $W_f(w) = W_g(w)$ для любого w .

Но коэффициенты Уолша однозначно определяют функцию, поэтому $f = g$. Получаем противоречие с тем, что f и g различны. \blacksquare

Следующие леммы потребуются для доказательства основной теоремы, следствием которой будет описание всех бент-функций на минимальном расстоянии от заданной бент-функции.

Лемма 2. Пусть $f, g \in \mathfrak{B}_n$. Тогда $|D(f, g)| = |D(\tilde{f}, \tilde{g})|$.

Доказательство. Воспользуемся формулой свертки:

$$W_{f \oplus g}(w) = \frac{1}{2^n} \sum_{x \in E^n} W_f(x) W_g(x \oplus w),$$

отсюда

$$W_{f \oplus g}(0) = \frac{1}{2^n} \sum_{x \in E^n} W_f(x) W_g(x) = \sum_{x \in E^n} (-1)^{\tilde{f}(x) \oplus \tilde{g}(x)} = W_{\tilde{f} \oplus \tilde{g}}(0).$$

Осталось заметить, что

$$W_{f \oplus g}(0) = 2^n - 2|D(f, g)|,$$

откуда и следует утверждение леммы. ■

Следствие 1. Пусть $f, g \in \mathfrak{B}_n$. Тогда $\text{dist}(f, g) = d$, если и только если существует ровно d векторов $w \in E^n$, таких, что $W_f(w) \neq W_g(w)$.

Лемма 3. Пусть $D \subseteq E^n$, $|D| = 2^k$, $\text{rank } D = k + 1$ и для любых $x, y \in D$ выполняется $x \oplus y \notin D$. Тогда D — линейное многообразие.

Доказательство. Пусть $\langle D \rangle$ — линейная оболочка D , $U = \langle D \rangle \setminus D$, $x_0 \in D$. Покажем, что $U = x_0 \oplus D$. Очевидно включение $x_0 \oplus D \subseteq U$. Также верно

$$|U| = |\langle D \rangle| - |D| = 2^{k+1} - 2^k = 2^k = |D| = |x_0 \oplus D|.$$

Следовательно, $U = x_0 \oplus D$.

Теперь покажем, что U — подпространство E^n . Множество U непусто, следовательно, достаточно показать только его замкнутость относительно \oplus .

Пусть $x, y \in U$, тогда $x = x_0 \oplus x'$, $y = x_0 \oplus y'$, $x', y' \in D$. Имеем

$$x \oplus y = (x_0 \oplus x') \oplus (x_0 \oplus y') = x' \oplus y' \in U,$$

так как по условию леммы для любых $x', y' \in D$ справедливо $x' \oplus y' \in U$.

Таким образом, U — подпространство E^n . И следовательно, D — линейное многообразие. ■

Лемма 4. Любая аффинная функция, заданная на линейном многообразии и не являющаяся константой, уравновешена (принимает значения 1 и 0 одинаковое количество раз).

Доказательство. Пусть f — аффинная функция, заданная на линейном многообразии U , и f — не константа. По определению линейного многообразия $U = x_0 \oplus L$ для некоторого подпространства L и вектора x_0 . Рассмотрим функцию $f'(x) = f(x \oplus x_0) \oplus f(x_0)$, определенную на подпространстве L . Очевидно, что f' линейна на L . Также легко понять, что f уравновешена на U тогда и только тогда, когда f' уравновешена на L (сдвиг множества является взаимнооднозначным отображением, а прибавление к функции константы сохраняет свойство уравновешенности), поэтому нам достаточно доказать утверждение леммы для функции f' .

Рассмотрим множества $L_0 = \{x \in L \mid f'(x) = 0\}$ и $L_1 = \{x \in L \mid f'(x) = 1\}$. Покажем, что множества L_0 и L_1 равноможны.

Функция f' — не константа на L , так как f — не константа на U . Поэтому выберем элемент y_0 из L_1 . Тогда для любого $x \in L_0$

$$f'(x \oplus y_0) = f'(x) \oplus f'(y_0) = 0 \oplus 1 = 1.$$

Значит, $y_0 \oplus L_0 \subseteq L_1$. Отсюда $|L_0| \leq |L_1|$ в силу взаимнооднозначности сдвига множества на y_0 . Но для любого $y \in L_1$

$$f'(y \oplus y_0) = f'(y) \oplus f'(y_0) = 1 \oplus 1 = 0.$$

Следовательно, $y_0 \oplus L_1 \subseteq L_0$ и $|L_1| \leq |L_0|$. Таким образом, $|L_0| = |L_1|$. А это и означает уравновешенность функции f' . ■

Следующая теорема дает необходимые и достаточные условия принадлежности функции на расстоянии $2^{n/2}$ от заданной бент-функции к классу бент-функций.

Теорема 1 (критерий расположения бент-функций на минимальном расстоянии). Пусть $f, g \in \mathcal{F}_n$, $f \in \mathfrak{B}_n$, $\text{dist}(f, g) = 2^{n/2}$. Тогда $g \in \mathfrak{B}_n$, если и только если $D(f, g)$ — линейное многообразие и функция f аффинна на $D(f, g)$.

Доказательство. Необходимость. Пусть $f, g \in \mathfrak{B}_n$, $|D(f, g)| = 2^{n/2}$. Покажем, что $D(f, g)$ — линейное многообразие и f аффинна на $D(f, g)$. Введем обозначения для следующих множеств:

$$W_{=0} = \{w \in E^n \mid a_{f,g}(w) = 0\},$$

$$W_{\neq 0} = \{w \in E^n \mid a_{f,g}(w) \neq 0\}.$$

По утверждению 2

$$W_f(w) - W_g(w) = 2a_{f,g}(w).$$

Так как $f, g \in \mathfrak{B}_n$, то $a_{f,g}(w) \in \{0, 2^{n/2}, -2^{n/2}\}$. Согласно следствию 1, $|W_{\neq 0}| = |D(f, g)|$.

По утверждению 1 $|a_{f,g}(w_0)| = |D(f, g)| = 2^{n/2} \iff \forall x \in D(f, g) f(x) = \langle w_0, x \rangle \oplus c$ для подходящей константы c . То есть аффинность мы доказали.

Таким образом, все $w \in W_{\neq 0}$ являются решениями одной из следующих систем:

$$\langle b, w \rangle = \langle b, w_0 \rangle, \quad b \in D(f, g),$$

$$\langle b, w \rangle = \langle b, w_0 \rangle \oplus 1, \quad b \in D(f, g).$$

Обозначив $w \oplus w_0$ за x , получаем равносильные системы уравнений:

$$\langle b, x \rangle = 0, \quad b \in D(f, g), \tag{1}$$

$$\langle b, x \rangle = 1, \quad b \in D(f, g). \tag{2}$$

Видно, что решения систем не пересекаются.

Теперь оценим $|W_{\neq 0}|$, исходя из того, что обе системы разрешимы:

$$2^{n/2} = |W_{\neq 0}| \leq 2 \cdot 2^{n - \text{rank } D(f, g)} = 2^{n - \text{rank } D(f, g) + 1}$$

(первая система всегда имеет $2^{n - \text{rank } D(f, g)}$ решений, а вторая либо имеет столько же решений, либо вообще не имеет). Отсюда

$$\text{rank } D(f, g) \leq n/2 + 1.$$

Но, исходя из мощности множества $D(f, g)$, получаем

$$\text{rank } D(f, g) \in \{n/2, n/2 + 1\}.$$

Рассмотрим два случая.

Случай 1: $\text{rank } D(f, g) = n/2$. Очевидно, в этом случае $D(f, g)$ является подпространством.

Случай 2: $\text{rank } D(f, g) = n/2 + 1$. В этом случае обе системы должны иметь решение, поэтому у второй системы должно существовать частное решение, т. е. существует $t_0 \in E^n$, такое, что $\langle t_0, b \rangle = 1$ для всех $b \in D(f, g)$.

Если существуют $x, y \in D(f, g)$, такие, что $x \oplus y \in D(f, g)$, то вторая система будет противоречива. Действительно, с одной стороны,

$$\langle t_0, x \rangle = 1, \langle t_0, y \rangle = 1 \implies \langle t_0, x \oplus y \rangle = 0,$$

с другой стороны, $\langle t_0, x \oplus y \rangle = 1$. Следовательно, выполняется условие леммы 3 для множества $D(f, g)$. Таким образом, $D(f, g)$ — линейное многообразие. Необходимость доказана.

Достаточность. Пусть $D(f, g)$ — линейное многообразие и f аффинна на $D(f, g)$. Для начала покажем, что

$$a_{f,g}(w) \in \{0, 2^{n/2}, -2^{n/2}\}.$$

Так как f аффинна на $D(f, g)$, то функция $h_w(x) = f(x) \oplus \langle w, x \rangle$ будет также аффинной на $D(f, g)$. Если $h_w(x)$ не является константой на $D(f, g)$, то по лемме 4 функция h_w уравновешена на $D(f, g)$. Поэтому

$$a_{f,g}(w) = \sum_{x \in D(f,g)} (-1)^{h_w(x)} = 0.$$

Если же $h_w(x)$ — константа на $D(f, g)$, то $|a_{f,g}(w)| = |D(f, g)| = 2^{n/2}$ по утверждению 1. Следовательно, $a_{f,g}(w) \in \{0, 2^{n/2}, -2^{n/2}\}$.

Осталось доказать, что $g \in \mathfrak{B}_n$. Имеем

$$W_g(w) = W_f(w) - 2a_{f,g}(w).$$

Следовательно,

$$W_g(w) \in \{2^{n/2}, -2^{n/2}, 3 \cdot 2^{n/2}, -3 \cdot 2^{n/2}\}.$$

Отсюда

$$\min_{w \in E^n} |W_g(w)| \geq 2^{n/2},$$

поэтому из равенства Парсеваля следует, что $|W_g(w)| = 2^{n/2}$ для любого w . Получаем, что $g \in \mathfrak{B}_n$. Достаточность доказана. ■

Следствие 2 (минимальное расстояние в классе \mathfrak{B}_n). Справедливо $d(\mathfrak{B}_n) = 2^{n/2}$.

Доказательство. Как известно, $f(x) = x_1 x_2 \oplus \dots \oplus x_{n-1} x_n$ является бент-функцией. Пусть

$$D = \{(y_1, 0, \dots, y_{n/2}, 0) \mid y_i \in E\};$$

очевидно, что

$$\begin{aligned} D &\subseteq E^n, \\ |D| &= 2^{n/2}, \\ \forall x \in D \quad f(x) &= 0. \end{aligned}$$

Пусть $g(x) = f(x) \oplus I_D(x)$. Понятно, что $D(f, g) = D$. Тогда по теореме 1 $g \in \mathfrak{B}_n$, т. е. $f, g \in \mathfrak{B}_n$, $\text{dist}(f, g) = 2^{n/2}$. А из леммы 1 мы знаем, что $d(\mathfrak{B}_n) \geq 2^{n/2}$. Следовательно, $d(\mathfrak{B}_n) = 2^{n/2}$. ■

Пусть $L_{\text{all}}(f)$ — всевозможные линейные многообразия размерности $n/2$, на которых f аффинна. Теперь можно более компактно описать бент-функции на минимальном расстоянии от заданной бент-функции.

Следствие 3 (общий вид функций на минимальном расстоянии). Пусть $f \in \mathfrak{B}_n$. Тогда существует $g \in \mathfrak{B}_n$ на минимальном расстоянии от f , если и только если множество $L_{\text{all}}(f)$ непусто, причем $g(x) = f(x) \oplus I_L(x)$, где $L \in L_{\text{all}}(f)$.

Следствие 4. Бент-функция от n переменных не может быть аффинна на линейных многообразиях размерности больше чем $n/2$.

Доказательство. Предположим, что для бент-функции f и линейного многообразия L утверждение неверно. Введем $g(x) = f(x) \oplus I_L(x)$ и $h_w(x) = f(x) \oplus \langle w, x \rangle$. Далее приводим рассуждения, аналогичные доказательству достаточности теоремы 1: так как f аффинна на L , то и h_w аффинна на L . Если размерность $D(f, g)$ больше чем $n/2$, то для всех элементов w , таких, что h_w константа (т. е. $a_{f,g}(w) \neq 0$) $|a_{f,g}(w)| > 2^{n/2}$, а для w , при которых h_w не константа, $a_{f,g}(w) = 0$ в силу уравновешенности функции h_w на L (лемма 4). Тогда из $W_g(w) = W_f(w) - 2a_{f,g}(w)$ следует

$$\min_{w \in E^n} |W_g(w)| \geq 2^{n/2},$$

но у различных функций все коэффициенты Уолша совпадать не могут. Таким образом, существует w_0 , для которого $a_{f,g}(w_0) \neq 0$ и, следовательно, $|W_g(w_0)| > 2^{n/2}$, что противоречит равенству Парсеваля. ■

Следствие 5. Пусть $D(f, g) = x_0 \oplus L$, L — подпространство и $f(x) = \langle w_0, x \rangle \oplus c$ для всех $x \in D(f, g)$ и некоторых w_0 и c . Тогда коэффициенты Уолша функций f и g отличаются только на элементах множества $w_0 \oplus L^\perp$.

Доказательство. В доказательстве необходимости теоремы 1 показано, что $W_{\neq 0} = w_0 \oplus U$, где U — множество решений следующих систем уравнений:

$$\langle b, x \rangle = 0, \quad b \in D(f, g),$$

$$\langle b, x \rangle = 1, \quad b \in D(f, g).$$

Так как $D(f, g)$ — линейное многообразие, то эти системы эквивалентны следующим системам:

$$\begin{cases} \langle b, x_0 \rangle = 0, \\ \langle b, x \rangle = 0, \quad b \in L \end{cases} \quad \text{и} \quad \begin{cases} \langle b, x_0 \rangle = 1, \\ \langle b, x \rangle = 0, \quad b \in L. \end{cases}$$

А эти системы, в свою очередь, эквивалентны одной системе

$$\langle b, x \rangle = 0, \quad b \in L.$$

Следовательно, $U = L^\perp$ и $W_{\neq 0} = w_0 \oplus L^\perp$. ■

Замечание. Результатам, полученным в теореме 1 и следствиях, предшествовали следующие исследования: В. В. Яценко [3] и К. Карле [4] рассматривали задачу построения различных бент-функций по имеющейся бент-функции путем прибавления индикатора линейного многообразия. Как показывает следствие 3, эта задача тесно

связана с задачей построения бент-функций на минимальном расстоянии от заданной бент-функции. В работе В. В. Яценко предлагалось к бент-функции прибавлять индикатор линейного многообразия, на котором она аффинна. К. Карле рассматривал индикаторы линейных многообразий произвольной размерности. В своей работе он предложил необходимые и достаточные условия на бент-функцию, при которых прибавление индикатора линейного многообразия к ней дает в результате бент-функцию.

Таким образом, достаточность в теореме 1 следует из результатов В. В. Яценко и К. Карле.

2. Индикаторы линейных многообразий и координатные подпространства

В связи с полученным описанием бент-функций на минимальном расстоянии от заданной бент-функции интересно рассмотреть, что из себя представляют индикаторы линейных многообразий размерности $n/2$.

Рассмотрим $I_U(x)$, где U — подпространство.

Пусть $a_1, a_2, \dots, a_{n/2}$ — базис U^\perp . Исходя из $U^{\perp\perp} = U$, имеем

$$x \in U \iff \forall i \in \{1, \dots, n/2\} \langle a_i, x \rangle = 0.$$

Отсюда получаем формулу для индикатора:

$$I_U(x) = (\langle a_1, x \rangle \oplus 1) \cdot (\langle a_2, x \rangle \oplus 1) \cdot \dots \cdot (\langle a_{n/2}, x \rangle \oplus 1).$$

Таким образом, индикатор подпространства размерности $n/2$ является конъюнкцией $n/2$ сомножителей, каждый из которых — отрицание скалярного произведения вектора переменных и базисного вектора U^\perp .

Рассмотрим $I_L(x)$, где L — линейное многообразие.

Пусть $x_0 \in L$, $U = x_0 \oplus L$. Очевидно, что U — подпространство. Тогда

$$I_L(x) = I_U(x_0 \oplus x).$$

Утверждение 3. Пусть U — линейное многообразие размерности $n/2$. Тогда $\deg(I_U) = n/2$.

Следствие 6. Если бент-функции находятся на минимальном расстоянии друг от друга, то хотя бы одна из них имеет алгебраическую степень $n/2$.

В общем случае задача нахождения подпространств, на которых f аффинна, достаточно сложна и не решается методом «пристального взгляда». Но если полином Жегалкина функции f содержит мало слагаемых, то часто существуют координатные подпространства (подпространства, базисом которых являются векторы веса 1 из E^n), на которых f аффинна. Рассмотрим алгоритм построения бент-функции, находящейся на минимальном расстоянии от данной, с использованием координатных подпространств.

Алгоритм 1

Вход: $f(x)$ — исходная бент-функция.

Выход: $g(x)$ — бент-функция на минимальном расстоянии от функции $f(x)$.

- 1) Фиксируем значения любых $n/2$ переменных так, чтобы функция от оставшихся $n/2$ переменных стала аффинной.
- 2) Если первый шаг завершился успешно, то получаем индикатор многообразия в следующем виде:

$$I_L(x) = (x_{i_1} \oplus x_{i_1}^0 \oplus 1) \cdot (x_{i_2} \oplus x_{i_2}^0 \oplus 1) \cdot \dots \cdot (x_{i_{n/2}} \oplus x_{i_{n/2}}^0 \oplus 1),$$

где x_{i_j} — переменные, которые мы зафиксировали на первом шаге алгоритма; $x_{i_j}^0$ — значения зафиксированных переменных.

Таким образом, если мы фиксировали переменную x_i значением 0, то в индикатор многообразия войдет отрицание этой переменной, если же мы фиксировали значением 1, то в индикатор войдет сама переменная.

3) Получаем

$$g(x) = f(x) \oplus I_L(x).$$

Пример 1. Пусть

$$f(x) = x_1x_2 \oplus x_3x_4 \oplus \dots \oplus x_{n-1}x_n.$$

1) Фиксируем все переменные с чётными номерами значением 1. Получаем следующую функцию:

$$f(x_1, 1, x_3, 1, \dots, x_{n-1}, 1) = x_1 \oplus x_3 \oplus \dots \oplus x_{n-1}.$$

Видно, что получилась аффинная функция.

2) Индикатор будет выглядеть следующим образом:

$$I_L(x) = x_2x_4 \dots x_n.$$

3) Получаем следующую выходную функцию:

$$g(x) = x_1x_2 \oplus x_3x_4 \oplus \dots \oplus x_{n-1}x_n \oplus x_2x_4 \dots x_n.$$

3. Известные подклассы бент-функций с точки зрения существования функций на минимальном расстоянии

Рассматривается существование бент-функций на минимальном расстоянии для бент-функций из известных подклассов класса \mathfrak{B}_n .

3.1. Класс Мэйорана — Мак-Фарланда

Класс Мэйорана — Мак-Фарланда обозначается как \mathcal{M} . В этом классе содержатся бент-функции следующего вида:

$$f(x, y) = \langle x, \pi(y) \rangle \oplus \psi(y), \quad (3)$$

где $x, y \in E^{n/2}$, $\psi \in \mathcal{F}_{n/2}$, π — подстановка на $E^{n/2}$.

Более подробно об этой конструкции можно узнать в [5].

Покажем, что для любой функции из класса \mathcal{M} существует бент-функция на минимальном расстоянии: достаточно воспользоваться алгоритмом 1.

1) Фиксируем значение y : пусть $y = y_0$. Тогда

$$f(x, y_0) = \langle x, \pi(y_0) \rangle \oplus \psi(y_0).$$

Понятно, что это аффинная функция.

2) Получаем индикатор, в который входят только переменные из y .

3) Получаем бент-функцию в классе \mathcal{M} на минимальном расстоянии от исходной.

Класс \mathcal{M} является простым и достаточно богатым.

$$|\mathcal{M}_n| = 2^{n/2!} \cdot 2^{2^{n/2}}. \quad (4)$$

Известно (см. описание бент-функций в виде линейного разветвления в [6]), что если бент-функция имеет вид $f(x, y) = \langle x, \pi(y) \rangle \oplus \psi(y)$, где $x, y \in E^{n/2}$, $\psi \in \mathcal{F}_{n/2}$, то отображение π взаимнооднозначно и, следовательно, функция $f(x)$ принадлежит классу \mathcal{M}_n .

Мощность класса \mathcal{M}_n часто используется как нижняя оценка мощности класса \mathfrak{B}_n . Если рассматривать функции, аффинно эквивалентные функциям из \mathcal{M}_n , то идею Мэйорана — Мак-Фарланда можно трактовать следующим образом:

- 1) Берем подпространство U размерности $n/2$ в E^n .
- 2) Разбиваем E^n на смежные классы относительно U .
- 3) На каждом смежном классе U строим аффинную функцию.

Также следует отметить, что бент-функции, аффинно эквивалентные функциям из класса Мэйорана — Мак-Фарланда, не являются всеми бент-функциями, имеющими непустое L_{all} . Пример такой бент-функции будет приведен в п. 4.

3.2. Partial Spreads

Класс бент-функций Partial Spreads обозначается как \mathcal{PS} . Он состоит из двух подклассов \mathcal{PS}^- , \mathcal{PS}^+ , определяемых следующим образом.

\mathcal{PS}^- . Пусть

- 1) L_i — подпространства в E^n , $i \in \{1, 2, \dots, 2^{n/2-1}\}$;
- 2) размерность L_i равна $n/2$;
- 3) $L_i \cap L_j = \{0\} \forall i \neq j$.

Тогда функции вида

$$f(x) = \bigoplus_{i=1}^{2^{n/2-1}} I_{L_i}(x) \quad (5)$$

принадлежат \mathcal{PS}^- .

\mathcal{PS}^+ . Пусть

- 1) L_i — подпространства в E^n , $i \in \{1, 2, \dots, 2^{n/2-1} + 1\}$;
- 2) размерность L_i равна $n/2$;
- 3) $L_i \cap L_j = \{0\} \forall i \neq j$.

Тогда функции вида

$$f(x) = \bigoplus_{i=1}^{2^{n/2-1}+1} I_{L_i}(x) \quad (6)$$

принадлежат \mathcal{PS}^+ .

Класс \mathcal{PS} впервые был описан Дж. Диллоном в [7].

С точки зрения существования бент-функций на минимальном расстоянии больший интерес представляет класс \mathcal{PS}^+ (для функций из \mathcal{PS}^- это сложный вопрос). Нетрудно убедиться, что все функции из класса \mathcal{PS}^+ имеют непустые множества L_{all} : для всех подпространств L_i из определения функций в этом классе выполняется

$$\forall x \in L_i \setminus \{0\} f(x) = 1,$$

потому что пространства пересекаются только по нулевому элементу. Кроме того,

$$f(0) = 1,$$

так как 0 лежит во всех пространствах, а количество пространств нечетно. Следовательно,

$$\forall x \in L_i f(x) = 1.$$

Таким образом, $L_i \in L_{\text{all}}(f)$ для всех i , причем функции вида

$$g(x) = f(x) \oplus I_{L_i}(x)$$

будут принадлежать \mathcal{PS}^- , и порождающими пространствами для них будут все подпространства для f , кроме L_i .

4. Аффинная эквивалентность бент-функций и минимальное расстояние

Рассмотрим известные факты об аффинной классификации бент-функций и взаимосвязь этих фактов с расстоянием между бент-функциями.

Определение 6. Булевы функции f и g от n переменных называются *аффинно эквивалентными*, если существует невырожденная матрица A размера $n \times n$, вектор b длины n и аффинная функция l от n переменных, такие, что $g(x) = f(A \cdot x \oplus b) \oplus l(x)$ для любого x .

Рассмотрим известные факты об аффинной классификации бент-функций.

Утверждение 4 [8]. Все бент-функции степени 2 аффинно эквивалентны функции $f(x) = x_1x_2 \oplus x_3x_4 \oplus \dots \oplus x_{n-1}x_n$.

Утверждение 5 [9] и др. Каждая бент-функция от 6 переменных аффинно эквивалентна одной из следующих функций:

- 1) $f_1^6 = x_1x_2 \oplus x_3x_4 \oplus x_5x_6$;
- 2) $f_2^6 = x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6$;
- 3) $f_3^6 = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_2 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_5 \oplus x_4x_5$;
- 4) $f_4^6 = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_4 \oplus x_3x_5 \oplus x_3x_6 \oplus x_4x_5 \oplus x_4x_6$.

Утверждение 6 [10]. Каждая бент-функция от 8 переменных степени не больше 3 аффинно эквивалентна одной из следующих функций:

- 1) $f_1^8 = x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \oplus x_7x_8$;
- 2) $f_2^8 = x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6 \oplus x_7x_8$;
- 3) $f_3^8 = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4 \oplus x_2x_6 \oplus x_1x_7 \oplus x_5x_8$;
- 4) $f_4^8 = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_3 \oplus x_1x_5 \oplus x_2x_6 \oplus x_3x_4 \oplus x_7x_8$;
- 5) $f_5^8 = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_2x_6 \oplus x_2x_5 \oplus x_1x_7 \oplus x_4x_8$;
- 6) $f_6^8 = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_1x_3 \oplus x_1x_4 \oplus x_2x_7 \oplus x_6x_8$;
- 7) $f_7^8 = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_2x_6 \oplus x_2x_5 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1x_4 \oplus x_7x_8$;
- 8) $f_8^8 = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_1x_6 \oplus x_2x_7 \oplus x_4x_8$;
- 9) $f_9^8 = x_1x_2x_7 \oplus x_3x_4x_7 \oplus x_5x_6x_7 \oplus x_1x_4 \oplus x_3x_6 \oplus x_2x_5 \oplus x_4x_5 \oplus x_7x_8$;
- 10) $f_{10}^8 = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_1x_4x_7 \oplus x_3x_5 \oplus x_2x_7 \oplus x_1x_5 \oplus x_1x_6 \oplus x_4x_8$.

Рассмотрим связь расстояний между бент-функциями и аффинной эквивалентностью.

Так как класс \mathfrak{B}_n замкнут относительно отрицания функции, то из формулы $d(\mathfrak{B}_n) = 2^{n/2}$ следует

Утверждение 7. Наибольшее расстояние в классе \mathfrak{B}_n , не равное 2^n , равно $2^n - 2^{n/2}$.

Определение 7 (спектр расстояний для функции). Пусть $f \in \mathfrak{B}_n$. Тогда вектор r из \mathbb{Z}^{2^n} называется *спектром расстояний* для бент-функции f , если i -я компонента вектора r равна количеству бент-функций на расстоянии i от функции f .

Утверждение 8. Спектры расстояний для аффинно эквивалентных бент-функций одинаковы.

Доказательство. Имеют место следующие равенства:

$$\text{dist}(f(x), g(x)) = \text{dist}(f(A \cdot x \oplus b), g(A \cdot x \oplus b)), \det(A) \neq 0;$$

$$\forall h \in \mathcal{F}_n \quad \text{dist}(f, g) = \text{dist}(f \oplus h, g \oplus h).$$

Так как класс бент-функций замкнут относительно аффинного преобразования переменных и относительно прибавления любой аффинной функции, то из этих равенств очевидным образом следует утверждение. ■

Следовательно, чтобы получить всевозможные спектры расстояний для бент-функций, достаточно найти спектры расстояний для одной бент-функции из каждого класса аффинной эквивалентности.

Утверждение 9. Любая бент-функция степени 2 имеет непустое $L_{\text{all}}(f)$.

Доказательство. Из утверждения 4 известно, что все бент-функции степени 2 аффинно эквивалентны функции $f(x) = x_1 \cdot x_2 \oplus x_3 \cdot x_4 \oplus \dots \oplus x_{n-1} \cdot x_n$. А для этой функции условие теоремы выполнено. ■

Утверждение 10. Все функции из класса \mathfrak{B}_6 имеют непустое L_{all} .

Доказательство. Применяем алгоритм 1 к аффинно неэквивалентным бент-функциям от 6 переменных:

- 1) $f_1^6(x_1, 0, x_3, 0, x_5, 0) = 0;$
- 2) $f_2^6(0, 0, 0, x_4, x_5, x_6) = 0;$
- 3) $f_3^6(0, 0, x_3, x_4, 0, x_6) = 0;$
- 4) $f_4^6(x_1, 0, 0, 0, x_5, x_6) = 0,$

то есть первый шаг алгоритма для каждой функции завершился успешно. Отсюда следует утверждение теоремы. ■

Утверждение 11. Все бент-функции от 8 переменных степени не больше 3 имеют непустое L_{all} .

Доказательство. Применяем алгоритм 1 к аффинно неэквивалентным бент-функциям от 8 переменных степени не больше 3:

- 1) $f_1^8(0, x_2, 0, x_4, 0, x_6, 0, x_8) = 0;$
- 2) $f_2^8(0, 0, 0, x_4, x_5, x_6, 0, x_8) = 0;$
- 3) $f_3^8(0, 0, x_3, 0, 0, x_6, x_7, x_8) = 0;$
- 4) $f_4^8(0, 0, 0, x_4, x_5, x_6, 0, x_8) = 0;$
- 5) $f_5^8(0, 0, 0, 0, x_5, x_6, x_7, x_8) = 0;$
- 6) $f_6^8(0, 0, 0, x_4, x_5, 0, x_7, x_8) = 0;$
- 7) $f_7^8(0, 0, 0, x_4, x_5, x_6, 0, x_8) = 0;$
- 8) $f_8^8(0, 0, 0, 0, x_5, x_6, x_7, x_8) = 0;$
- 9) $f_9^8(x_1, 0, 0, 0, x_5, x_6, 0, x_8) = 0;$
- 10) $f_{10}^8(0, 0, 0, 0, x_5, x_6, x_7, x_8) = 0,$

то есть первый шаг алгоритма для каждой функции завершился успешно. Отсюда следует утверждение теоремы. ■

Утверждение 12. Пусть $f \in \mathfrak{B}_n$. Тогда f аффинно эквивалентна функции из класса Мэйорана — Мак-Фарланда тогда и только тогда, когда существует $L \in L_{\text{all}}(f)$, такое, что $x_0 \oplus L \in L_{\text{all}}(f)$ для всех $x_0 \in E^n$.

Доказательство. Необходимость. Пусть f представлена в виде

$$f(x, y) = \langle x, h(y) \rangle \oplus g(y), \quad x, y \in E^{n/2}.$$

Тогда зададим L следующим образом:

$$L = \{(x, 0) \mid x \in E^{n/2}\}.$$

Очевидно, что данное L удовлетворяет условию утверждения.

Достаточность. Пусть существует линейное многообразие L , удовлетворяющее заданному условию. Понятно, что невырожденным аффинным преобразованием мы можем привести функцию f к виду

$$f'(x, y) = \langle x, h(y) \rangle \oplus g(y), \quad x, y \in E^{n/2}.$$

Из критерия Ященко о представлении бент-функции в виде линейного разветвления (см. [6]) следует, что $h(y)$ взаимнооднозначно и f' принадлежит классу \mathcal{M}_n . ■

Утверждение 13. В классе \mathfrak{B}_8 существуют бент-функции с непустым L_{all} , аффинно неэквивалентные функциям из класса Мэйорана — Мак-Фарланда.

Пример 2. Бент-функция от 8 переменных с непустым L_{all} , аффинно неэквивалентная функциям из класса Мэйорана — Мак-Фарланда:

$$\begin{aligned} f(x) = & x_1x_2x_5x_8 \oplus x_1x_2x_6x_8 \oplus x_1x_4x_5x_8 \oplus x_1x_4x_6x_8 \oplus x_1x_5x_6x_8 \oplus x_2x_4x_5x_8 \oplus x_2x_4x_6x_8 \oplus \\ & x_3x_4x_5x_8 \oplus x_4x_5x_6x_8 \oplus x_1x_2x_3 \oplus x_1x_2x_5 \oplus x_1x_2x_6 \oplus x_1x_2x_8 \oplus x_1x_4x_5 \oplus x_1x_4x_6 \oplus x_1x_4x_8 \oplus \\ & x_1x_5x_6 \oplus x_1x_5x_8 \oplus x_1x_6x_8 \oplus x_2x_4x_5 \oplus x_2x_4x_6 \oplus x_2x_4x_8 \oplus x_2x_5x_8 \oplus x_2x_6x_8 \oplus x_3x_4x_5 \oplus x_3x_4x_8 \oplus \\ & x_3x_5x_8 \oplus x_4x_5x_6 \oplus x_4x_6x_8 \oplus x_5x_6x_8 \oplus x_1x_2 \oplus x_1x_5 \oplus x_1x_6 \oplus x_1x_8 \oplus x_2x_4 \oplus x_2x_6 \oplus x_2x_8 \oplus \\ & x_3x_4 \oplus x_3x_5 \oplus x_3x_6 \oplus x_3x_8 \oplus x_4x_6 \oplus x_5x_6 \oplus x_6x_8 \oplus x_7x_8 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_6. \end{aligned}$$

С помощью программы, реализующей алгоритм 2 из п. 5, было показано, что у данной функции имеются линейные многообразия размерности 4, на которых она аффинна, но при этом ни для одного из них не выполняется условие утверждения 12.

5. Алгоритм перебора бент-функций, находящихся на минимальном расстоянии от заданной бент-функции

Из теоремы 1 следует, что для того чтобы перебрать все бент-функции, находящиеся на минимальном расстоянии от заданной бент-функции, достаточно перебрать все многообразия размерности $n/2$, на которых заданная бент-функция аффинна.

Базис подпространства размерности $n/2$ задаётся $n/2$ строками матрицы ступенчатого вида, состоящей из нулей и единиц, такой, что

- 1) каждая из последующих строк заканчивается меньшим количеством подряд идущих нулей, чем предыдущая строка;
- 2) под каждой ведущей единицей (первая единица в строке) в столбце стоят нули;
- 3) остальные элементы произвольны.

Приведём пример такой матрицы и бент-функций, находящихся на минимальном расстоянии друг от друга.

Пример 3. Пусть дана бент-функция от шести переменных:

$$f_1(x) = x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6.$$

Для построения бент-функции, находящейся на минимальном расстоянии от данной, нужно найти такое многообразие $L = x_0 \oplus U$, что f_1 будет аффинна на этом многообразии. На компьютере с помощью алгоритма, который приведен ниже, были построены

все базисы подпространств U и все x_0 , такие, что f_1 аффинна на $L = x_0 \oplus U$. Приведем пример таких x_0 и U . Пусть

$$A_U = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \text{ — базисная матрица подпространства } U;$$

$x_0 = (0 \ 1 \ 0 \ 0 \ 0 \ 0)$. Тогда базисная матрица ортогонального подпространства

$$A_{U^\perp} = (a_{ij}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Теперь построим индикатор, являющийся конъюнкцией $n/2$ сомножителей, где i -й сомножитель получен из i -й строки матрицы A_{U^\perp} таким образом:

$$(a_{i1}x_1 \oplus a_{i2}x_2 \oplus \dots \oplus a_{in}x_n \oplus 1).$$

Отсюда следует $I_L(x) = I_U(x \oplus x_0) = I_U(x_1, x_2 \oplus 1, x_3, x_4, x_5, x_6) = (x_6 \oplus 1)(x_2 \oplus 1 \oplus 1)(x_1 \oplus x_3 \oplus x_4 \oplus 1) = x_1x_2x_6 \oplus x_2x_3x_6 \oplus x_2x_4x_6 \oplus x_2x_6 \oplus x_1x_2 \oplus x_2x_3 \oplus x_2x_4 \oplus x_2$. Отсюда получаем вторую бент-функцию $f_2(x) = f_1(x) \oplus I_L(x) = x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6 \oplus x_1x_2x_6 \oplus x_2x_3x_6 \oplus x_2x_4x_6 \oplus x_2x_6 \oplus x_1x_2 \oplus x_2x_3 \oplus x_2x_4 \oplus x_2$, которая находится на минимальном расстоянии $2^{n/2} = 8$ от функции f_1 .

Следующий алгоритм нам потребуется для нахождения всех бент-функций, находящихся на минимальном расстоянии от заданной бент-функции, но мы приведем его в общем виде для произвольной булевой функции.

Алгоритм 2 (поиск всех многообразий размерности $n/2$, на которых функция аффинна).

- 1) На вход алгоритма подается булева функция $f(x)$ (в нашем случае это бент-функция).
- 2) Далее для всех x_0 из E^n получаем новую функцию: $g(x) = f_{x_0}(x) = f(x \oplus x_0)$.
- 3) После этого $g(x)$ нормируется: $g(x) = g(x) \oplus g(0)$.

Таким образом, мы перешли от поиска многообразий, на которых заданная функция аффинна, к поиску подпространств, на которых заданная функция линейна. Для каждой такой функции $g(x)$ будем рекуррентно строить все базисные матрицы размера $n/2 \times n$.

- 4) Пусть A_1 — матрица, содержащая одну строку.
- 5) Допустим, что построена матрица A_{i-1} размера $s - 1 \times n$, где $s - 1 < n/2$, и нужно построить матрицу A_i размера $s \times n$. Для этого добавим ещё один вектор в базис. Перебираем все векторы v в лексикографическом порядке² и приписываем их к матрице A_{i-1} , так, чтобы получившаяся матрица A_i удовлетворяла условиям 1–3 на с. 17. Для каждого вектора v , претендующего стать базисным, проверяются два условия:

а) $g(x \oplus v) = g(x) \oplus g(v)$;

- б) вектор x_0 лексикографически предшествует вектору $v \oplus x \oplus x_0$, где x пробегает множество всех ранее добавленных в подпространство векторов.

Если хотя бы одно условие не выполняется, вектор v отбрасывается. Таким образом, сразу отсекается большое количество ненужных векторов. Если ни один из векторов не подошел, значит, это тупиковая ветка рекурсии.

²Заметим, что в реализации удобнее нумеровать компоненты векторов справа налево.

Трудоёмкость данного алгоритма в самом худшем случае (если на вход алгоритма подать линейную функцию) не превышает $T = O(n/2 \cdot P_{n/2} \cdot 2^{n/2})$, где $n/2$ — количество операций, необходимое для построения одного подпространства; $P_{n/2} = \frac{(2^n - 1)(2^{n-1} - 1) \dots (2^{n/2+1} - 1)}{(2^{n/2} - 1)(2^{n/2-1} - 1) \dots (2^1 - 1)}$ — количество подпространств размерности $n/2$; $2^{n/2}$ — количество операций, затрачиваемых на проверку условий. Но на практике данный алгоритм работает значительно быстрее.

Приведем экспериментальные данные. В таблице показаны бент-функции (по утверждению 8 достаточно рассматривать представителей различных классов аффинной эквивалентности), количество многообразий, на которых данная бент-функция аффинна, и время работы программы в секундах (на процессоре с тактовой частотой 2,4 ГГц).

n	Бент-функция	$ L_{\text{all}} $	Время
4	$x_1x_2 \oplus x_3x_4$	60	0
6	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_4 \oplus$ $\oplus x_3x_5 \oplus x_3x_6 \oplus x_4x_5 \oplus x_4x_6$	376	0,001
6	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_2 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_5 \oplus x_4x_5$	440	0,001
6	$x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6$	568	0,001
6	$x_1x_2 \oplus x_3x_4 \oplus x_5x_6$	1080	0,002
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_1x_4x_7 \oplus x_3x_5 \oplus$ $\oplus x_2x_7 \oplus x_1x_5 \oplus x_1x_6 \oplus x_4x_8$	1392	0,035
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_2x_6 \oplus$ $\oplus x_2x_5 \oplus x_1x_7 \oplus x_4x_8$	2928	0,035
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_1x_3 \oplus$ $\oplus x_1x_4 \oplus x_2x_7 \oplus x_6x_8$	2928	0,035
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_2x_6 \oplus$ $\oplus x_2x_5 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1x_4 \oplus x_7x_8$	2928	0,036
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_1x_6 \oplus x_2x_7 \oplus x_4x_8$	2928	0,034
8	$x_1x_2x_7 \oplus x_3x_4x_7 \oplus x_5x_6x_7 \oplus x_1x_4 \oplus x_3x_6 \oplus$ $\oplus x_2x_5 \oplus x_4x_5 \oplus x_7x_8$	4464	0,036
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4 \oplus x_2x_6 \oplus x_1x_7 \oplus x_5x_8$	6000	0,037
8	$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_3 \oplus x_1x_5 \oplus x_2x_6 \oplus x_3x_4 \oplus x_7x_8$	6000	0,037
8	$x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6 \oplus x_7x_8$	12144	0,045
8	$x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \oplus x_7x_8$	36720	0,070
10	$x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \oplus x_7x_8 \oplus x_9x_{10}$	2424520	8,543

ЛИТЕРАТУРА

1. Токарева Н. Н. Бент-функции: результаты и приложения. Обзор работ // Прикладная дискретная математика. 2009. № 3. С. 15–37.
2. Paterson K. G. Sequences For OFDM and Multi-code CDMA: two problems in algebraic Coding Theory // Sequences and their applications. Seta 2001. Second Int. Conference (Bergen, Norway, May 13–17, 2001). Proc. Berlin: Springer, 2002. P. 46–71.
3. Логачев О. А., Сальников А. А., Яценко В. В. Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2004. 470 с.
4. Carlet C. Boolean Functions for Cryptography and Error Correcting Codes // Chapter of the monograph «Boolean Methods and Models», Cambridge Univ.

- Press / eds. P. Hammer, Y. Crama, to appear. Prelim. version is available at www-rocq.inria.fr/secret/Claude.Carlet/chap-fcts-Bool.pdf.
5. *McFarland R. L.* A family of difference sets in non-cyclic groups // J. Combin. Theory. Ser. A. 1973. V. 15. No. 1. P. 1–10.
 6. *Яценко В. В.* О критерии распространения для булевых функций и о бент-функциях // Пробл. передачи информации. 1997. Т. 33. Вып. 1. С. 75–86.
 7. *Dillon J. F.* Elementary Hadamard Difference sets // Ph. D. Thesis. Univ. of Maryland, 1974.
 8. *Dillon J. F.* A survey of bent functions // The NSA Technical J. 1972. Special Issue. P. 191–215.
 9. *Rothaus O.* On bent functions // J. Combin. Theory. Ser. A. 1976. V. 20. No. 3. P. 300–305.
 10. *Braeken A.* Cryptographic properties of Boolean functions and S-boxes // Ph. D. Thesis, Katholieke Univ. Leuven, 2006.

**ОБ ИНВАРИАНТАХ НЕКОТОРЫХ КЛАССОВ
КВАЗИМОНОТОННЫХ ФУНКЦИЙ НА ПОЛУРЕШЁТКЕ¹**

Н. Г. Парватов

*Томский государственный университет, г. Томск, Россия***E-mail:** parvatov@mail.tsu.ru

Рассматриваются классы квазимонотонных, монотонных, а также слабо существенных квазимонотонных и монотонных функций на конечной верхней полурешётке. В системах инвариантных для этих классов предикатов находятся порождающие множества, порождающие эти системы с использованием диагоналей и с помощью операций конъюнкции предикатов, отождествления и перестановки переменных, введения и удаления фиктивных переменных. Рассматриваемые вопросы связаны с проблемами полноты, выразимости и конечной порождаемости для этих классов.

Ключевые слова: *полурешётка, монотонная функция, квазимонотонная функция, мажоритарная функция.*

Введение: полурешётки и неполная информация

При описании управляющих систем средствами дискретных функций с операциями суперпозиции возникают трудности, обусловленные явлением состязаний [1]. В монографии Г. П. Агибалова [2] для преодоления этих трудностей предложено считать изменяющиеся (то есть в различной степени определённые) состояния дискретной управляющей системы элементами верхней полурешётки, интерпретируя полурешёточное отношение порядка как отношение сравнения состояний по степени их неопределённости и используя полурешёточное сложение для описания промежуточных состояний. При таком подходе функции состояний и выходов управляющей системы оказываются функциями на полурешётке, причём монотонными, ведь монотонность отражает очевидное свойство системы: её внутренние и выходные состояния уточняются (становятся более определёнными) при уточнении входного состояния. Среди монотонных функций выделяют функции, не допускающие дальнейшего монотонного уточнения. К синтезу таких функций, названных в работах Г. П. Агибалова *минимальными точечными*, сводятся задачи создания асинхронных дискретных управляющих систем. Представляют интерес также функции, в том числе и немонотонные, допускающие монотонное уточнение; такие функции называются *квазимонотонными*. Квазимонотонные функции удобно использовать для формулирования задачи синтеза дискретной управляющей системы с заданным динамическим поведением. Подобная задача может состоять в необходимости создания дискретной управляющей системы, функции состояний и выходов которой уточняют указанные заранее квазимонотонные функции. Отметим также, что идея рассматривать в разной степени неопределённые значения как элементы верхней полурешётки подмножеств лежит в основе понятия нечёткой информации, теория которой развита в работах Л. А. Шоломова (см., например, [3]).

¹Работа выполнена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы.

В связи со сказанным актуальны задачи синтеза функций в классах квазимонотонных, монотонных и минимальных точечных функций, возникающие на различных этапах проектирования асинхронных управляющих систем, а также проблемы полноты и выразимости функций в этих классах, рассматриваемые в [4, 5]. Точнее, в [4] получены критерии полноты в классах монотонных и квазимонотонных функций на трёхэлементной полурешётке непустых подмножеств двухэлементного множества, а также получен критерий выразимости минимальных точечных функций в классе монотонных функций на той же полурешётке. В [5] рассматривалась проблема полноты в классе квазимонотонных функций на произвольной полурешётке при суперпозиции со всеми так называемыми *слабо существенными квазимонотонными функциями*, допускающими, в соответствии с их определением, уточнение одноместными монотонными функциями.

В данной статье изучаются общие свойства замкнутых классов квазимонотонных и монотонных функций, а также классов слабо существенных квазимонотонных и монотонных функций на произвольной конечной верхней полурешётке. В том числе изучаются свойства предикатов, инвариантных для функций в этих классах. Рассматриваемые свойства связаны с вопросами конечной порождаемости этих классов и представляют интерес в связи с проблемами выразимости и полноты в них.

1. Верхняя полурешётка

Пусть в конечном множестве D , упорядоченном отношением \leq , для любых элементов a и b имеется точная верхняя грань $a + b$, а точная нижняя грань $a \cdot b$ существует не для любых элементов a и b . Иными словами, множество D вместе с указанным упорядочением является верхней полурешёткой, но не решёткой.

Отношение порядка \leq , определённое в полурешётке D , переносится на наборы в D^n естественным образом — покомпонентно, так, что выполнение неравенства $a \leq b$ для наборов $a = (a_1, \dots, a_n)$ и $b = (b_1, \dots, b_n)$ означает выполнение покомпонентных неравенств $a_i \leq b_i$ при $1 \leq i \leq n$. Таким образом, множество D^n становится полурешёткой с покомпонентными сложением и умножением.

Упорядочение \leq переносится и на функции $f : D^n \rightarrow D$, множество которых при всевозможных натуральных n обозначается через P_D . При этом неравенство $f \leq g$ означает для функций f и g , что они зависят от одинакового числа переменных и для любого набора a значений их переменных выполняется неравенство $f(a) \leq g(a)$. В этом случае функция f называется *минорантой* функции g .

Наибольший элемент верхней полурешётки будем обозначать через \top . Удобно верхнюю полурешётку D (это касается и любой другой верхней полурешётки) считать вложенной в решётку $D \cup \{\perp\}$ с наименьшим элементом \perp . Указанное вложение позволяет пользоваться произведениями ab для любых элементов a и b из D . В этом случае отсутствие произведения в полурешётке означает, что это произведение принимает значение \perp в решётке.

2. Основные классы квазимонотонных функций

Функции из P_D , сохраняющие полурешёточное отношение порядка \leq , называются *монотонными*, а их класс обозначается через M_D . Функция из P_D , имеющая монотонную миноранту, называется *квазимонотонной*. Среди монотонных минорант квазимонотонной функции f имеется наибольшая M_f , значение которой на любом наборе a из D^n (где n — число переменных функций f и M_f) можно найти так:

$$M_f(a) = \prod f(a'),$$

при этом произведение вычисляется в полурешётке D по всем наборам $a' \geq a$ из D^n .

Обозначим через $q(D)$ максимальное число элементов минимального по включению подмножества полурешётки D , не имеющего в ней нижней грани. Известный тест квазимонотонности из [2] утверждает, что функция $f : D^n \rightarrow D$ тогда и только тогда квазимонотонная, когда выполняется следующее свойство: если какие-то наборы (достаточно какие-то $q(D)$ наборов) имеют нижнюю грань в полурешётке D^n , то значения функции f на этих наборах имеют нижнюю грань в полурешётке D . В соответствии с этим класс квазимонотонных функций на полурешётке D , далее обозначаемый через Q_D , является классом сохранения системы предикатов

$$\varepsilon_r(x_1, \dots, x_r) \equiv \exists x(x \leq x_1 \wedge \dots \wedge x \leq x_r)$$

при всевозможных целых положительных r и даже является классом сохранения одного-единственного предиката $\varepsilon_{q(D)}$.

Функция из P_D , имеющая монотонную миноранту, существенно зависящую не более чем от одной переменной, называется *слабо существенной квазимонотонной функцией* (на полурешётке D). Слабо существенные функции введены в [5], где при суперпозиции с ними решена проблема полноты в классе квазимонотонных функций. В соответствии с [5] функция $f : D^n \rightarrow D$ тогда и только тогда является слабо существенной квазимонотонной, когда для некоторого числа $i, 1 \leq i \leq n$, выполняется следующее свойство: если i -е компоненты каких-то наборов (достаточно каких-то $q(D)$ наборов) из D^n имеют нижнюю грань в полурешётке D , то и значения функции f на этих наборах также имеют нижнюю грань в D .

Используя сформулированный критерий, покажем для дальнейшего использования, что класс слабо существенных квазимонотонных функций на верхней полурешётке D является классом сохранения системы предикатов

$$\varepsilon_{r,m} \equiv \varepsilon_r(x_1, \dots, x_r) \vee \varepsilon_r(x_{r+1}, \dots, x_{2r}) \vee \dots \vee \varepsilon_r(x_{(m-1)r+1}, \dots, x_{mr})$$

при всевозможных целых положительных r и m , и даже классом сохранения предикатов $\varepsilon_{q(D),m}$ при всевозможных целых положительных m . С этой целью условие сохранения n -местной функцией f предиката $\varepsilon_{r,m}$ сформулируем следующим образом. Если заданы mr наборов из множества D^n , разбитых на m r -элементных групп, и на наборах каждой группы значения функции f не имеют нижней грани, то найдётся компонента, значения которой в каждой группе наборов не имеют нижней грани. Это условие выполняется для слабо существенной квазимонотонной функции. Обратное, при достаточно большом m (но не превосходящем числа $|D|^{nr}$ всевозможных r -элементных групп наборов из D^n) mr наборов можно выбрать так, чтобы среди их групп присутствовала каждая r -элементная группа наборов из D^n , на которых функция f не имеет нижней грани. В связи с этим сформулированное выше условие приводит к наличию у функции f , сохраняющей предикат $\varepsilon_{r,m}$, такой переменной, пусть с номером i , что если на каких-то r наборах функция не имеет нижней грани, то i -е компоненты этих наборов не имеют нижней грани. Равносильно, если i -е компоненты каких-то r наборов имеют нижнюю грань, то и значения функции на этих наборах имеют нижнюю грань. Если сказанное выполняется для $r = q(D)$, то оно выполняется при любом r ; это следует из определения числа $q(D)$. Таким образом, желаемое свойство установлено.

Класс слабо существенных квазимонотонных функций станем обозначать через Φ_D .

Основная цель данной статьи состоит в выявлении некоторых общих свойств замкнутых классов квазимонотонных, монотонных и слабо существенных функций, а также систем инвариантов (то есть систем предикатов, сохраняемых функциями) этих классов. Результаты данной статьи иллюстрируют конструкции, введённые в [9] в связи с задачей выявления свойств, обеспечивающих конечную порождаемость замкнутого класса функций конечнозначной логики.

3. Клоны с мажоритарной функцией и их обобщения

Напомним некоторые необходимые для дальнейшего факты. При этом рассматриваемые здесь свойства выполняются для произвольного конечного множества D , не обязательно являющегося полурешёткой.

Мажоритарной называется функция $m(x_1, \dots, x_n)$, зависящая от $n \geq 3$ переменных и удовлетворяющая при любом $i = 1, \dots, n$ соотношению

$$x = m(x, \dots, x, x_i, x, \dots, x),$$

где переменная x_i находится на i -м месте под знаком функции m , переменная x — на остальных местах. Клоны с мажоритарной функцией конечно порождены [6].

Обратимся к вопросу о строении алгебры инвариантных предикатов клона, содержащего мажоритарную функцию. (Напомним, что *клоном* называют замкнутый суперпозицией класс функций, содержащий тождественную одноместную функцию.) Как известно [7], для любого множества K функций из P_D множество $\text{inv}_D(K)$ предикатов $p : D^n \rightarrow \{И, Л\}$, сохраняемых функциями из K , замкнуто операциями конъюнкции, проектирования, отождествления и перестановки переменных, введения и удаления фиктивных переменных и включает все диагонали. (Диагоналями называются тождественно истинные и тождественно ложные предикаты, а также предикаты, выражающиеся формулами вида $x_i = x_j \wedge \dots \wedge x_k = x_l$, где $\{i, j, \dots, k, l\} = \{1, \dots, n\}$ для натуральных n .) Множества предикатов, включающие диагонали и замкнутые операциями конъюнкции, отождествления и перестановки переменных, введения и удаления фиктивных переменных, назовём *и-классами*. Для любого множества A предикатов $p : D^n \rightarrow \{И, Л\}$ обозначим через $[A]_{\wedge}$ *и-класс*, порождённый множеством A , то есть наименьший по включению среди и-классов, включающих множество A ; через $A^{(d)}$ обозначим множество предикатов из A , зависящих не более чем от d переменных.

Клоны с мажоритарной функцией в терминах их инвариантных предикатов характеризует теорема Бейкера и Пиксли из [8]. С использованием этой теоремы в [9] установлено, что для клона K функций из P_D равносильны условия:

- (1) клон K содержит $(d + 1)$ -местную мажоритарную функцию;
- (2) и-класс $\text{inv}_D(K)$ порождается всеми своими предикатами, зависящими не более чем от d переменных, то есть имеет место равенство

$$\text{inv}_D(K) = [(\text{inv}_D(K))^{(d)}]_{\wedge}.$$

В качестве обобщений клонов с мажоритарной функцией в [9] были введены d -подклоны и (c, d) -клоны. Клон K_1 называется d -подклоном клона K_2 , если выполняются равносильные в силу [9] свойства:

- (1) имеет место включение $K_1 \subseteq K_2$, и для любой функции $f(x_1, \dots, x_n)$ из K_2 в K_1 найдётся функция $m_f(x_1, \dots, x_{n+d+1})$, удовлетворяющая всевозможным соотношениям

$$f(x) = m_f(x, f(x), \dots, f(x), x_{i+n}, f(x), \dots, f(x)),$$

где через x обозначен набор переменных x_1, \dots, x_n , переменная x_{i+n} находится на $(i+n)$ -м месте функции m_f и $1 \leq i \leq d+1$;

(2) имеет место равенство $\text{inv}_D(K_1) = [\text{inv}_D(K_2) \cup (\text{inv}_D(K_1))^{(d)}]_{\wedge}$.

Клон с конечно-порождаемым d -подклоном сам конечно-порождаемый [9].

Если функцию m_f в первом условии сделанного выше определения всегда удаётся выбрать зависящей не более чем от c переменных набора x (и тогда она зависит не более чем от $c+d+1$ переменных), то клон K_1 называется (c, d) -подклоном клона K_2 и оба клона K_1 и K_2 называются (c, d) -клонами. В соответствии с этим клоны с $(d+1)$ -местной мажоритарной функцией являются $(0, d)$ -подклонами клона P_D и $(0, d)$ -клонами. В статье [9] показано, что (c, d) -клоны (при натуральных n и d) конечно порождены.

Как видно, клоны с мажоритарной функцией и их обобщения — d -подклоны обладают двумя определениями: функциональным и предикатным. Функциональное определение характеризует указанные (под)клоны в терминах содержащихся в них функций, а предикатное — в терминах инвариантных предикатов. Для (c, d) -клонов в настоящее время известно только функциональное определение. Автор планирует исправить ситуацию в скором времени, охарактеризовав в одной из ближайших статей (c, d) -клоны в терминах сохраняемых предикатов.

Для нахождения порождающих множеств инвариантных и-классов полезной оказывается следующая

Лемма 1 (о продолжении частичной функции). Для клона K функций из P_D и множества A предикатов из $\text{inv}_D(K)$ следующие условия равносильны:

(1) $\text{inv}_D(K) = [A]_{\wedge}$;

(2) произвольную частичную функцию $f : A \rightarrow D$, где $A \subseteq D^n$, тогда и только тогда можно доопределить до функции в K , когда f сохраняет все предикаты из A .

Напомним, что сохранение частичной функцией $f : A \rightarrow D$, где $A \subseteq D^n$, m -местного предиката p означает, что для любых удовлетворяющих ему наборов $x^i = (x_1^i, \dots, x_m^i)$, $1 \leq i \leq n$, набор

$$f(x_1^1, \dots, x_1^n), \dots, f(x_m^1, \dots, x_m^n)$$

либо содержит неопределённую компоненту, либо также удовлетворяет предикату p . Доказывать здесь лемму о продолжении частичной функции не станем; сделаем это в одной из следующих статей.

4. Свойства основных классов квазимонотонных функций

В [10] показано, что классы M_D и Q_D содержат мажоритарную функцию

$$M(x_1, \dots, x_{q(D)+1}) = \prod_{i=1}^{q(D)} (x_1 + \dots + x_{i-1} + x_{i+1} + \dots + x_{q(D)+1}).$$

Значения этой функции определены при любых значениях переменных, несмотря на то, что операция умножения — частичная в полурешётке. Действительно, если в записанном соотношении произведение справа не определено при каких-то значениях переменных, то какие-то $q(D)$ из скобок-сомножителей этого произведения не имеют общей нижней грани в силу определения числа $q(D)$; но это невозможно, так как суммы в этих скобках имеют общее слагаемое. Функция M мажоритарная, в чём можно

убедиться с использованием полурешётчных тождеств следующим образом:

$$M(x, \dots, x, x_i, x, \dots, x) = (x + x_i) \cdots (x + x_i) \cdot x \cdot (x + x_i) \cdots (x + x_i) = (x + x_i)x = x.$$

Наконец, функция M монотонная, так как является композицией монотонных функций.

В силу сказанного и-классы $\text{inv}_D(M_D)$ и $\text{inv}_D(Q_D)$ порождаются множествами своих предикатов, зависящих не более чем от $q(D)$ переменных. С использованием леммы 1 можно получить более точные соотношения:

$$\text{inv}_D(M_D) = [\leq, \varepsilon_{q(D)}]_{\wedge}, \text{inv}_D(Q_D) = [\varepsilon_{q(D)}]_{\wedge}. \quad (1)$$

Действительно, частичную функцию $f : A \rightarrow D$, определённую на множестве $A \subseteq D^n$ и сохраняющую предикат $\varepsilon_{q(D)}$, можно доопределить значением \top до квазимонотонной функции, а если f сохраняет ещё и порядок \leq , то её можно доопределить до монотонной функции f' , такой, что $f'(d) = \prod f(d')$, где произведение вычисляется в полурешётке D по всем наборам d' из A , таким, что $d' \leq d$, и произведение пустого множества элементов считается равным элементу \top .

Из полученных соотношений (1) следует, что и-класс $\text{inv}_D(M_D)$ порождается множеством $\text{inv}_D(Q_D)$, пополненным 2-местным предикатом \leq . В частности, клон M_D является 2-подклоном клона Q_D . В справедливости этого можно убедиться иначе, воспользовавшись функциональным определением d -клона. Для этого достаточно заметить, что для квазимонотонной n -местной функции f , обладающей монотонной минорантой m' , функцию m_f , зависящую от $n+3$ переменных, можно выбрать следующим образом:

$$m_f(x_1, \dots, x_{n+3}) = m'(x_1, \dots, x_n) + (x_{n+1} + x_{n+2})(x_{n+1} + x_{n+3})(x_{n+2} + x_{n+3});$$

например, в качестве монотонной миноранты m' можно взять наибольшую M_f . Если функция f слабо существенная из Φ_D , то монотонную миноранту m' можно выбрать зависящей от некоторой одной переменной x_i при $1 \leq i \leq n$. В этом случае функция $m'(x_i)$ является минорантой функции m_f , принадлежащей классу $M_D \cap \Phi_D$, который является в силу сказанного (1, 2)-подклоном клона Φ_D .

Из сказанного видно, что порождающее множество и-класса $\text{inv}_D(M_D \cap \Phi_D)$ можно получить из порождающего множества и-класса $\text{inv}_D(\Phi_D)$, дополнив последнее какими-то 2-местными предикатами. С использованием леммы 1 можно получить более точные соотношения

$$\text{inv}_D(M_D \cap \Phi_D) = [\leq, \Upsilon_D]_{\wedge}, \text{inv}_D(\Phi_D) = [\Upsilon_D]_{\wedge},$$

где Υ_D — система всевозможных предикатов $\varepsilon_{q(D),m}$ при целых положительных m . Эти соотношения доказываются аналогично соотношениям (1). Таким образом, верна

Теорема 1. Имеют место следующие утверждения:

- (1) и-класс $\text{inv}_D(Q_D)$ порождается предикатом $\varepsilon_{q(D)}$;
- (2) и-класс $\text{inv}_D(M_D)$ порождается парой предикатов $\varepsilon_{q(D)}$ и \leq ;
- (3) и-класс $\text{inv}_D(\Phi_D)$ порождается системой предикатов Υ_D ;
- (4) и-класс $\text{inv}_D(\Phi_D \cap M_D)$ порождается системой предикатов $\Upsilon_D \cup \{\leq\}$.

В частности, клон M_D является 2-подклоном клона Q_D и клон $\Phi_D \cap M_D$ является 2-подклоном клона Φ_D . Более того, клон $\Phi_D \cap M_D$ является (1, 2)-подклоном клона Φ_D .

ЛИТЕРАТУРА

1. *Агибалов Г. П., Оранов А. М.* Лекции по теории конечных автоматов. Томск: Изд-во Том. ун-та, 1983. 185 с.
2. *Агибалов Г. П.* Дискретные автоматы на полурешётках. Томск: Изд-во Том. ун-та, 1993. 227 с.
3. *Шоломов Л. А.* Элементы теории недоопределённой информации // Прикладная дискретная математика. Приложение. 2009. №2. С. 18–42.
4. *Парватов Н. Г.* Функциональная полнота в замкнутых классах квазимонотонных и монотонных трехзначных функций на полурешетке // Дискретн. анализ и исслед. опер. Сер. 1. 2003. Т. 10. № 1. С. 1–78.
5. *Парватов Н. Г.* Теорема о функциональной полноте в классе квазимонотонных функций на конечной полурешетке // Дискрет. анализ и исслед. опер. Сер. 1. 2006. Т. 13. № 3. С. 62–82.
6. *Марченков С. С.* К существованию конечных базисов в замкнутых классах булевых функций // Алгебра и логика. 1984. Т. 23. № 1. С. 88–99.
7. *Боднарчук В. Г., Калужнин Л. А., Котов В. Н., Ромов Б. А.* Теория Галуа для алгебр Поста // Кибернетика. 1969. № 3. С. 1–10; № 5. С. 1–9.
8. *Baker K. A., Pixly A. F.* Polynomial interpolation and chinese remainder theorem for algebraic systems // Math. Zeiteschr. 1975. Bd. 143. No. 2. S. 165–174.
9. *Парватов Н. Г.* Замечания о конечной порождаемости замкнутых классов // Дискрет. анализ и исслед. операций. Сер. 1. 2004. Т. 11. № 3. С. 32–47.
10. *Парватов Н. Г.* Некоторые конструкции конечно-порождаемых клонов // Вестник Томского госуниверситета. Приложение. 2004. № 9. С. 26–28.

О ПРЕОБРАЗОВАНИЯХ ЦЕЙТИНА В ЛОГИЧЕСКИХ УРАВНЕНИЯХ¹

А. А. Семёнов

Институт динамики систем и теории управления СО РАН, г. Иркутск, Россия

E-mail: biclop@rambler.ru

В статье сообщается о применении преобразований Цейтина в различных областях пропозициональной логики, связанных с решением систем логических уравнений. Показывается, что преобразования Цейтина не изменяют числа решений системы логических уравнений, и строится биекция между множествами решений системы и результата её преобразований по Цейтину. Приводятся некоторые результаты по применению преобразований Цейтина к построению оценок сложности систем пропозиционального вывода. С использованием преобразований Цейтина строятся простейшие доказательства NP-полноты проблемы совместности системы логических уравнений степени 2 и #P-полноты проблемы подсчёта числа выполняющих наборов для хорновской КНФ. С использованием преобразований Цейтина показывается также, что ROBDD-граф для булевой функции в хорновской КНФ или в КНФ с двухбуквенными дизъюнктами нельзя построить за полиномиальное время, если $P \neq NP$.

Ключевые слова: логические уравнения, преобразования Цейтина, системы пропозиционального вывода, NP-полнота.

Введение

В 1968 г. в журнале «Записки научных семинаров ЛОМИ» вышла статья Григория Самуиловича Цейтина «О сложности вывода в исчислении высказываний» [1]. Сегодня можно с уверенностью сказать, что эта выдающаяся и совершенно новаторская на тот момент работа намного опередила время и предвосхитила целый спектр направлений в логике и теории алгоритмов.

Основным инструментом в [1] являются очень простые по своей природе преобразования выражений исчисления высказываний. Далее приведены две цитаты из [1], в которых описаны данные преобразования.

«Исчисления, которыми мы будем пользоваться, направлены на установление противоречивости систем дизъюнкций. Понятие противоречивой системы дизъюнкций служит здесь аналогом понятия тождественно истинной формулы в обычном исчислении высказываний. От вопроса о тождественной истинности заданной формулы исчисления высказываний можно перейти к вопросу о противоречивости некоторой системы дизъюнкций, приведя отрицание данной формулы к конъюнктивной нормальной форме. Однако при таком преобразовании может резко возрасти длина формулы, поэтому мы будем рассматривать другой способ перехода от формулы исчисления высказываний к системе дизъюнкций. Каждой подформуле данной формулы поставим в соответствие свою переменную; двум подформулам будут соответствовать сопряжённые переменные (переменная и её отрицание) в том и только том

¹Работа выполнена при поддержке гранта РФФИ № 07-01-00400-а и при поддержке гранта Президента РФ НШ-1676.2008.1. Результаты работы докладывались на Международной конференции с элементами научной школы для молодёжи, г. Омск, 7–12 сентября 2009 г.

случае, если одна из этих формул является отрицанием второй. Если некоторая подформула A представляет собой конъюнкцию подформул B и Γ и этим подформулам приписаны соответственно переменные α , β и γ , то припишем подформуле A следующую систему дизъюнкций: $\bar{\alpha}\beta$, $\bar{\alpha}\gamma$, $\alpha\bar{\beta}\bar{\gamma}$ (т.е. КНФ $(\bar{\alpha} \vee \beta) \cdot (\bar{\alpha} \vee \gamma) \cdot (\alpha \vee \bar{\beta} \vee \bar{\gamma})$). Аналогично припишем системы дизъюнкций подформулам, которые представляют собой дизъюнкции и импликации ($\alpha\bar{\beta}$, $\alpha\bar{\gamma}$, $\bar{\alpha}\beta\gamma$ для дизъюнкции и $\alpha\beta$, $\alpha\bar{\gamma}$, $\bar{\alpha}\bar{\beta}\gamma$ для импликации). Объединим все полученные таким способом системы дизъюнкций и добавим туда еще дизъюнкцию $\bar{\xi}$, где ξ — переменная, соответствующая всей данной формуле. Легко видеть, что полученная система дизъюнкций противоречива в том и только том случае, если данная формула тождественно истинна».

«Если α , β , γ — какие-нибудь переменные, причем ни α , ни $\bar{\alpha}$ не входят ни в одну из дизъюнкций системы, то систему можно дополнить следующим списком дизъюнкций: $\alpha\beta$, $\alpha\gamma$, $\bar{\alpha}\bar{\beta}\bar{\gamma}$ ». Подразумевается, что дополненная система будет противоречива тогда и только тогда, когда противоречива исходная.

Первоначальные идеи преобразований, описанных в первой цитате, принадлежат, по-видимому, к категории «фольклорных» — в качестве одного из наиболее ранних примеров Е. Я. Данциным (см. [2]) указывается работа [3]. Вторая цитата дает простейший пример преобразования из класса так называемых правил расширения. Отметим, что именно в [1] описанные выше преобразования (и первого, и второго типов) составили основу бесспорно нетривиальных результатов, по сути открывших новое направление в математической логике — теорию сложности формальных доказательств.

Перепечатку работы [1] в сборнике «Automation of reasoning» [4], выполненную через 15 лет после ее первого опубликования, можно считать заслуженной оценкой фундаментальности. Правда, здесь не обошлось без некоторых курьезных моментов. Так, авторы работы [5] рассматривают квантифицированный вариант приведенных выше преобразований, указывая в качестве первоисточника свою работу, датированную 1982 г., и буквально говоря, что «... пропозициональный вариант данных преобразований был предложен Цейтиным в 1983 г. (ссылка на [4])».

Сложно отследить момент, когда впервые в публикациях стал использоваться термин «преобразования Цейтина» (Tseitin transformation). Однако на сегодняшний день данный термин прочно укоренился в научной литературе (причем, главным образом, в отношении преобразований, описанных в первой цитате) и фигурирует в работах по сложности формальных доказательств, по верификации дискретных автоматов, по обращению дискретных функций (см., например, статьи [6–8] и многие другие). Далее термином «преобразования Цейтина» в соответствии со сложившимися традициями обозначаются преобразования первого типа, а для преобразований второго типа используется термин «правила расширения».

В настоящей работе делается попытка объединить несколько различных областей пропозициональной логики фактами использования в этих областях преобразований Цейтина. Далее приведен краткий план статьи.

В п. 1 преобразования Цейтина описываются в контексте общей проблемы приведения систем логических уравнений к нормальным формам. Рассматриваются логические уравнения вида $U = 1$, в левой части которых находится произвольная формула исчисления высказываний над множеством булевых переменных X . Результатом применения последовательности преобразований Цейтина к формуле U является формула U' над множеством X' , $X \subset X'$. Переход от произвольной U к U' в некоторой нормальной форме (КНФ, ДНФ, АНФ) осуществляется эффективно. Рассматривается логическое уравнение $U' = 1$. Естественным образом задается отображение ω множе-

ства решений уравнения $U' = 1$ на множество решений уравнения $U = 1$. Показано, что ω является биекцией. Здесь же рассматривается одно обобщение преобразований Цейтина для логических уравнений. Соответствующее обобщенным преобразованиям Цейтина отображение ω множества решений преобразованного уравнения на множество решений исходного уравнения в общем случае сюръективно.

П. 2 посвящён применению преобразований Цейтина и правил расширения в системах пропозиционального вывода. Приведен результат Г. С. Цейтина 1968 г., давший первый пример строгой аргументации большей мощности одной системы пропозиционального вывода в сравнении с другой (для этой цели использовались правила расширения). Здесь же приведены результаты, полученные А. Хакеном в 1985 г., в которых демонстрируется большая мощность «расширенной резолюции» (в терминологии А. Хакена) в сравнении с общей резолюцией на формулах Дирихле (в расширенной резолюции допускается использование правил расширения и преобразований Цейтина). Также рассматривается известная попытка сравнения мощности общей резолюции и системы пропозиционального вывода, базирующейся на двоичных диаграммах решений.

В п. 3 в очень сжатой форме рассмотрены вопросы эффективной сводимости различных по своей природе задач к задачам поиска решений логических уравнений в нормальных формах. Такого рода вопросы возникают в верификационных задачах микроэлектроники, задачах обращения дискретных функций, а также при осуществлении различных декомпозиционных представлений логических уравнений. Преобразования Цейтина при этом являются основным элементом соответствующих решений.

В п. 4 исследуется сложность проблемы подсчета числа наборов значений переменных, выполняющих произвольную хорновскую КНФ. Известно, что данная проблема $\#P$ -полна. Соответствующие доказательства, имеющиеся в работах Л. Дж. Валианта и С. П. Горшкова, весьма трудны технически. Предлагается новое доказательство данного факта, базирующееся на преобразованиях Цейтина. Полученное доказательство существенно проще известных.

Тезисное изложение результатов этой работы приведено в [9].

1. Преобразования Цейтина и отображения на множествах решений логических уравнений

Обозначим через $\{0, 1\}^n$ множество всех двоичных последовательностей (слов) длины n . Также в дальнейшем используем обозначение $\{0, 1\}^* = \bigcup_{n \in \mathbb{N}} \{0, 1\}^n$. Пусть $X = \{x_1, \dots, x_n\}$ — множество булевых переменных, а $U(x_1, \dots, x_n)$ — формула исчисления высказываний (ИВ), реализующая булеву функцию от этих переменных, определенную всюду на $\{0, 1\}^n$ [10].

Выражения вида $U(x_1, \dots, x_n) = 0$ или $U(x_1, \dots, x_n) = 1$ называются логическими (иногда булевыми) уравнениями. Решением логического уравнения $U(x_1, \dots, x_n) = \beta$, $\beta \in \{0, 1\}$, называется такой набор $(\alpha_1, \dots, \alpha_n)$ значений булевых переменных x_1, \dots, x_n , что $U(\alpha_1, \dots, \alpha_n) = \beta$. Если такого набора не существует, то говорят, что уравнение не имеет решений. Решением системы, состоящей из m логических уравнений, называется набор значений всех булевых переменных системы, который является решением каждого уравнения. Если такого набора не существует, то система называется несовместной. Говорят, что система уравнений $U_i(x_1, \dots, x_n) = \beta_i$, $i = 1, \dots, m$, представлена в нормальной форме, если для всех $i \in \{1, \dots, m\}$ значение β_i одно и то же и формула $U_i(x_1, \dots, x_n)$ одной и той же нормальной формы — КНФ, ДНФ или АНФ.

Литералом над X называется произвольная булева переменная из X либо ее отрицание. Литералы x и \bar{x} называются контрарными. Дизъюнктом над X называется произвольная дизъюнкция литералов, среди которых нет одинаковых и контрарных. Конъюнктивная нормальная форма над X — это конъюнкция различных дизъюнктов.

Пусть $C(x_1, \dots, x_n)$ — КНФ над $X = \{x_1, \dots, x_n\}$. Данная КНФ называется выполнимой, если логическое уравнение

$$C(x_1, \dots, x_n) = 1 \quad (1)$$

имеет решения. Решения уравнения (1) также называются наборами, выполняющими КНФ $C(x_1, \dots, x_n)$. Задача распознавания выполнимости (существования выполняющего набора) произвольной КНФ является исторически первой NP-полной задачей [11]. Задачи поиска решений логических уравнений вида (1) далее называются SAT-задачами.

Логические уравнения можно рассматривать как слова над конечными алфавитами. Эти слова при помощи взаимнооднозначных кодирований могут преобразовываться в двоичные слова [12]. При этом длина (число бит) получаемого слова считается объемом двоичного кода уравнения при фиксированной схеме кодирования. Далее предполагается, что все рассматриваемые схемы кодирования являются «разумными» (в соответствии с терминологией, используемой в [12]).

В работе [1] рассматривалось действие преобразований Цейтина на множестве формул ИВ. Для ряда приложений интерес представляет естественный перенос преобразований Цейтина на логические уравнения. Будем говорить, что преобразование Цейтина применяется к логическому уравнению U_1 , если оно применяется к формуле ИВ, стоящей в левой части U_1 ; в результате имеем новое логическое уравнение U_2 . Значительная часть дальнейшего материала посвящена исследованию взаимосвязей между множествами решений U_1 и U_2 с приложениями к различным теоретическим и прикладным областям математической логики.

Рассмотрим логическое уравнение следующего вида:

$$F(h_1(x_1^1, \dots, x_{r_1}^1), \dots, h_s(x_1^s, \dots, x_{r_s}^s)) = 1. \quad (2)$$

Здесь $F(h_1(x_1^1, \dots, x_{r_1}^1), \dots, h_s(x_1^s, \dots, x_{r_s}^s))$ — это произвольная формула ИВ, задающая некоторую булеву функцию от множества переменных

$$X = \{x_1, \dots, x_n\} = \cup_{j=1}^s \{x_1^j, \dots, x_{r_j}^j\},$$

где подформулы $h_i(x_1^i, \dots, x_{r_i}^i)$, $i \in \{1, \dots, s\}$, сами задают булевы функции от переменных в X . Решениями уравнения (2), если они существуют, являются векторы из $\{0, 1\}^n$. Множество решений (2) обозначим через Ω_1 .

Пусть $\mu(y_1, \dots, y_m)$ — произвольная булева функция от булевых переменных из множества $Y = \{y_1, \dots, y_m\}$, $Y \cap X = \emptyset$, которая принимает на $\{0, 1\}^m$ одинаковое число раз значения 0 и 1, и пусть $\Sigma(\mu)$ — произвольная формула ИВ, задающая данную функцию. Введем в рассмотрение булеву функцию

$$\varphi : \{0, 1\}^{r_1+m} \rightarrow \{0, 1\},$$

значение которой на произвольном наборе $(\alpha_1 \dots \alpha_{r_1} \lambda_1 \dots \lambda_m)$ значений переменных $x_1^1, \dots, x_{r_1}^1, y_1, \dots, y_m$ определяется следующим образом:

$$\varphi(\alpha_1, \dots, \alpha_{r_1}, \lambda_1, \dots, \lambda_m) = \begin{cases} 1, & \text{если } \mu(\lambda_1, \dots, \lambda_m) = h_1(\alpha_1, \dots, \alpha_{r_1}); \\ 0, & \text{если } \mu(\lambda_1, \dots, \lambda_m) = \neg h_1(\alpha_1, \dots, \alpha_{r_1}). \end{cases}$$

Рассмотрим следующее логическое уравнение над множеством булевых переменных $X \cup Y$:

$$\Sigma(\varphi(x_1^1, \dots, x_{r_1}^1, y_1, \dots, y_m)) F_{h_1 \rightarrow \Sigma(\mu)}(x_1, \dots, x_n, y_1, \dots, y_m) = 1, \quad (3)$$

где через $\Sigma(\varphi(x_1^1, \dots, x_{r_1}^1, y_1, \dots, y_m))$ обозначена произвольная формула ИВ, задающая функцию φ , а через $F_{h_1 \rightarrow \Sigma(\mu)}(x_1, \dots, x_n, y_1, \dots, y_m)$ обозначена формула ИВ, полученная из $F(h_1(x_1^1, \dots, x_{r_1}^1), \dots, h_s(x_1^s, \dots, x_{r_s}^s))$ заменой одного или нескольких (быть может, всех) вхождений формулы $h_1(x_1^1, \dots, x_{r_1}^1)$ формулой $\Sigma(\mu)$. Особо отметим, что решениями уравнения (3) (если они существуют) являются векторы из $\{0, 1\}^{n+m}$. Множество решений уравнения (3) обозначим через Ω_2 . Установим справедливость следующей теоремы.

Теорема 1.

- 1) $\Omega_1 = \emptyset$ тогда и только тогда, когда $\Omega_2 = \emptyset$.
- 2) В случае $\Omega_1 \neq \emptyset$, $\Omega_2 \neq \emptyset$ существует сюръективное отображение $\omega : \Omega_2 \rightarrow \Omega_1$, сопоставляющее каждому элементу Ω_2 некоторый элемент из Ω_1 , причем любой элемент из Ω_1 имеет 2^{m-1} прообразов в Ω_2 при отображении ω .
- 3) От любого решения уравнения (3) можно перейти к соответствующему (в смысле отображения ω) решению уравнения (2) в общем случае за линейное время.

Доказательство. Докажем второе утверждение. Предположим, что $\Omega_2 \neq \emptyset$, и покажем, что в этом случае любому решению уравнения (3) можно поставить в соответствие некоторое решение уравнения (2). Пусть $(x_1^0, \dots, x_n^0, y_1^0, \dots, y_m^0)$ — решение уравнения (3). Тогда $h_1(x_1^0, \dots, x_n^0) = \mu(y_1^0, \dots, y_m^0)$, $F_{h_1 \rightarrow \Sigma(\mu)}(x_1^0, \dots, x_n^0, y_1^0, \dots, y_m^0) = 1$ и, следовательно,

$$F(h_1(x_1^0, \dots, x_n^0), \dots, h_s(x_1^0, \dots, x_n^0)) = 1. \quad (4)$$

Таким образом, решению $(x_1^0, \dots, x_n^0, y_1^0, \dots, y_m^0)$ уравнения (3) соответствует решение уравнения (2), имеющее вид (x_1^0, \dots, x_n^0) . Тем самым определено отображение ω множества решений уравнения (3) на множество решений уравнения (2). Здесь под $h_i(x_1^0, \dots, x_n^0)$, $i \in \{1, \dots, s\}$, подразумевается результат подстановки соответствующих компонент вектора (x_1^0, \dots, x_n^0) в формулу $h_i(x_1^i, \dots, x_{r_i}^i)$.

Пусть теперь вектор (x_1^0, \dots, x_n^0) является решением уравнения (2), то есть имеет место (4). Поскольку функция $\mu(y_1, \dots, y_m)$ принимает одинаковое число раз значения 0 и 1 на $\{0, 1\}^m$, то на 2^{m-1} векторах из $\{0, 1\}^m$ ее значение совпадает с $\alpha = h_1(x_1^0, \dots, x_n^0)$. Обозначим через Y^0 такое подмножество в $\{0, 1\}^m$, что

$$\forall (y_1^0, \dots, y_m^0) \in Y^0 (\mu(y_1^0, \dots, y_m^0) = h_1(x_1^0, \dots, x_n^0)).$$

Несложно понять, что произвольный вектор вида

$$(x_1^0, \dots, x_n^0, y_1^0, \dots, y_m^0), (y_1^0, \dots, y_m^0) \in Y^0,$$

является решением уравнения (3). В силу сказанного выше $|Y^0| = 2^{m-1}$. Векторы вида $(x_1^0, \dots, x_n^0, y_1^0, \dots, y_m^0)$, $(y_1^0, \dots, y_m^0) \in \{0, 1\}^m \setminus Y^0$, решениями (3) быть не могут, поскольку на таких векторах функция φ принимает значение 0. Следовательно, каждое решение (2) имеет в точности 2^{m-1} прообразов при отображении ω (тем самым ω — сюръекция).

Справедливость первого утверждения теоремы 1 автоматически следует из сказанного. Справедливость третьего утверждения также очевидна, поскольку из любого

решения уравнения (3) можно эффективно выделить значения переменных из X , получив тем самым решение уравнения (2). ■

Переход от уравнения (2) к уравнению (3) представляет собой одну итерацию обобщенных преобразований Цейтина. Очевидным образом из теоремы 1 вытекает справедливость следующего факта.

Следствие 1. Для одноместных булевых функций $\mu(y)$, заданных формулами y или \bar{y} , отображение ω является биекцией между множествами Ω_1 и Ω_2 .

Дополнительно отметим, что функцию φ можно задать следующей формулой ИВ (здесь символ « \equiv » обозначает логическую эквивалентность):

$$\Sigma(\mu(y_1, \dots, y_m)) \equiv h_1(x_1^1, \dots, x_{r_1}^1).$$

В итоговом уравнении или системе функция φ может быть представлена в любой нормальной форме.

Несложно понять, что от любого уравнения вида (2) за полиномиальное от длины его двоичного кода время при помощи описанных выше преобразований можно перейти к системе уравнений в любой нормальной форме над множеством булевых переменных $Z = \{z_1, \dots, z_{q(n)}\}$, $X \subset Z$, $q(n)$ — некоторый полином. В получаемой при этом системе логические уравнения имеют вид $U(z_1, \dots, z_{q(n)}) = 1$, где $U(z_1, \dots, z_{q(n)})$ — формула ИВ в заранее оговоренной нормальной форме.

2. Преобразования Цейтина и сложность формальных доказательств

Приведен краткий обзор известных автору примеров использования преобразований Цейтина в теории сложности формального вывода. Изложим здесь некоторые ключевые понятия этой теории (см., например, [13]).

2.1. Системы пропозиционального вывода

Прежде всего отметим, что с точки зрения корректного определения понятия вычислительной сложности процедур логического вывода имеет смысл рассматривать лишь исчисление нулевого порядка, то есть исчисление высказываний, поскольку уже в исчислении предикатов первого порядка множество теорем не является рекурсивным.

В основе современной теории сложности пропозициональных доказательств лежит следующий формализм, восходящий к С. Куку и Р. Рекхау [14].

Пусть в исчислении высказываний дано некоторое натуральное семейство логических противоречий S , дополненное выделенным символом \emptyset . Рассмотрим алгоритмически вычислимую (вообще говоря, сюръективную) функцию

$$\pi : \{0, 1\}^* \rightarrow S \cup \{\emptyset\}, \tag{5}$$

которая произвольному двоичному слову $\sigma \in \{0, 1\}^*$ ставит в соответствие логическое противоречие из семейства S либо символ \emptyset . Если $\pi(\sigma) = s$, $s \in S$, говорим, что σ — это π -доказательство противоречивости s (ситуацию $\pi(\sigma) = \emptyset$ можно интерпретировать как абсурдность строки σ). Рассматриваются только такие функции вида (5), которые вычислимы за полиномиальное время (везде далее, кроме специально оговариваемых случаев, сложность того или иного алгоритма понимается как функция от объема двоичного кода входных данных). Особо отметим, что в определение функции π закладывается способ доказательства противоречивости формул из S — можно рассматривать функции, которые распознают именно резолютивные доказательства,

именно DPLL-доказательства и т. д., задавая тем самым вполне конкретные *системы пропозиционального вывода* (СПВ).

Произвольному противоречию $s \in S$ и фиксированной функции π вида (5) поставим в соответствие кратчайшее слово $\sigma_* \in \{0, 1\}^*$, такое, что $\pi(\sigma_*) = s$. Рассмотрим функцию, сопоставляющую каждому противоречию $s \in S$ длину соответствующего слова σ_* . Полученную функцию назовем сложностью СПВ π на семействе противоречий S . Если сложность π растёт как полином от длины двоичной записи противоречий из S (при некоторой разумной схеме кодирования), то π называем полиномиально ограниченной СПВ для семейства S .

Несложно убедиться в том, что СПВ для класса всех противоречий исчисления высказываний, определенные как функции вида (5), существуют. Для этого достаточно установить два факта. Во-первых, показать наличие полиномиальной по сложности процедуры, преобразующей произвольное противоречие исчисления высказываний в противоречие (возможно, над более широким множеством булевых переменных), представленное в КНФ. В контексте сказанного выше очевидно, что такого рода процедуру дают преобразования Цейтина. Во-вторых, предъявить любой из обширного семейства полных (то есть завершающих работу за конечное время) алгоритмов доказательства противоречивости КНФ.

Факт существования полиномиально ограниченных СПВ для класса всех противоречий исчисления высказываний совершенно неправдоподобен, поскольку несложно видеть, что наличие хотя бы одной такой СПВ влечет равенство $NP = co-NP$ [14].

Тем не менее можно пытаться строить оценки сложности конкретных СПВ на конкретных натуральных семействах логических противоречий. На первый взгляд такого рода задачи представляются малоинтересными. Однако при более детальном рассмотрении в этом направлении открывается целая область, насыщенная нетривиальными и практически значимыми результатами. Важнейшими в этом направлении являются результаты по аргументации большей мощности одних СПВ в сравнении с другими.

Допустим, что относительно двух полных СПВ π_1 и π_2 установлено, что любое π_2 -доказательство σ_2 противоречивости произвольной КНФ C можно за полиномиальное от $|\sigma_2|$ время преобразовать в π_1 -доказательство σ_1 противоречивости C . В этом случае говорим, что СПВ π_1 полиномиально моделирует СПВ π_2 . Если π_1 полиномиально моделирует π_2 , а π_2 полиномиально моделирует π_1 , то эти СПВ называются полиномиально эквивалентными.

Предположим, что существует такое натуральное семейство логических противоречий S , что сложность π_1 на S ограничивается полиномом, а сложность π_2 на S , напротив, полиномом ограничить нельзя. Очевидно, что в данном случае СПВ π_2 не может полиномиально моделировать СПВ π_1 . Если при этом π_1 полиномиально моделирует π_2 , то относительно π_1 логично сделать вывод о его большей мощности по сравнению с π_2 .

Работа [1] содержит исторически первый пример подобного сравнения мощности двух СПВ. Остановимся на данном моменте более подробно. Основным объектом изучения [1] являются СПВ, базирующиеся на методе резолюций. Данный метод впервые был предложен в работе [15] и долгое время считался одним из самых перспективных подходов к автоматическому доказательству теорем в исчислении предикатов первого порядка. Далее мы описываем и используем пропозициональный вариант метода резолюций.

Рассматривается произвольная КНФ $C = D_1 \cdot \dots \cdot D_m$, где $D_j, j \in \{1, \dots, m\}$, — дизъюнкты над множеством булевых переменных $X = \{x_1, \dots, x_n\}$. Ставится вопрос

о выполнимости C . Если дизъюнкты D_{k_1} и D_{k_2} , $k_1, k_2 \in \{1, \dots, m\}$, содержат контрарные литералы x и \bar{x} (например, первый дизъюнкт содержит x , а второй — \bar{x}), то говорят, что D_{k_1} и D_{k_2} контражны по переменной x . Если D — произвольный дизъюнкт, а a — литерал, входящий в D , то через $D \setminus \{a\}$ обозначается дизъюнкт, полученный из D вычеркиванием a . Пусть D_{k_1} и D_{k_2} , $k_1, k_2 \in \{1, \dots, m\}$, контражны по переменной x . Для определенности полагаем, что D_{k_1} содержит x , а D_{k_2} — \bar{x} . Дизъюнкт $D' = (D_{k_1} \setminus \{x\} \vee D_{k_2} \setminus \{\bar{x}\})$ называется резольвентой дизъюнктов D_{k_1} и D_{k_2} по переменной x . Несложно видеть, что КНФ $C' = C \cdot D'$, где D' — резольвента некоторой пары дизъюнктов из C , выполнима на тех и только тех наборах значений истинности переменных из X , на которых выполнима C . Далее ставится вопрос о выполнимости C' . Описанная процедура представляет собой одну итерацию метода резолюций. Дж. А. Робинсоном в [15] было показано, что C невыполнима тогда и только тогда, когда существует такая конечная последовательность итераций метода резолюций, итогом которой является дизъюнкт, не содержащий литералов и называемый пустым (пустой дизъюнкт есть резольвента единичных контрарных дизъюнктов вида a и \bar{a}). Данный факт известен также как теорема о полноте метода резолюций (пропозициональный вариант).

Рассмотрим функцию π вида (5), которая распознает двоичные описания резольтивных доказательств логических противоречий, заданных в КНФ. Полученную СПВ будем называть далее общей резолюцией (general resolution). Говоря о сложности резольтивного доказательства противоречивости некоторой КНФ, мы будем подразумевать число порожденных в ходе этого доказательства резольвент, поскольку именно этот параметр вносит основной вклад в длину двоичного описания доказательства (каждая резольвента — это дизъюнкт, включающий не более n литералов).

Сказанное выше демонстрирует недетерминированный характер метода резолюций: в общем случае после порождения каждой конкретной резольвенты существует много различных альтернатив порождения последующей, причем все эти альтернативы являются допустимыми в смысле общей резолюции. Снижения недетерминизма метода резолюций можно добиться за счет дополнения его специальными стратегиями, разрешающими строить резольвенты только в соответствии с определенными правилами, ограничивающими общую резолюцию. При этом основной является проблема сохранения полноты — ограниченная (в смысле конкретной стратегии) резолюция должна так же, как и общая, гарантировать доказуемость противоречивости КНФ за конечное число шагов. Большое число различных резольтивных стратегий проанализировано в [16].

В некотором смысле двойственным введению ограничивающих стратегий является увеличение выразительной силы СПВ за счет дополнительных правил вывода (расширение исходной СПВ). Здесь не возникает проблем с полнотой, поскольку полнота базовой системы гарантирует полноту расширенной. С другой стороны, получаемая система начинает выглядеть сложнее с точки зрения анализа ее предельных возможностей (по крайней мере, в отношении нижних границ сложности).

2.2. Сравнение мощностей различных СПВ, базирующихся на методе резолюций

Как уже отмечалось, исторически первый пример такого сравнения содержится в [1]. В данной работе рассматривались две СПВ. Первая — это ограниченный вариант общей резолюции, известный как регулярная резолюция (regular resolution), вторая

СПВ представляет собой регулярную резолюцию, дополненную возможностью применять к рассматриваемому противоречию простейшее правило расширения.

Далее поясняются некоторые ключевые моменты. Рассматривается логическое противоречие C , представленное в КНФ над множеством булевых переменных X . Процедуру опровержения C посредством общей резолюции удобно представлять в виде дерева (дерева вывода), корнем которого является пустой дизъюнкт, ветви помечаются литералами, а узлы (вершины) — дизъюнктами, по которым порождаются резольвенты (в том числе и собственно резольвентами). Очевидно, что при этом некоторые вершины дерева вывода могут быть помечены одинаковыми дизъюнктами, а ветви — одинаковыми переменными. Две ветви, выходящие из произвольного узла, соответствующего некоторой резольвенте, помечаются парой контрарных литералов, удаление которых привело к порождению данной резольвенты. В регулярной резолюции требуется, чтобы для любой переменной $x \in X$ каждый путь в дереве вывода из корня в лист содержал не более одного ребра, помеченного литералом из $\{x, \bar{x}\}$. Регулярная резолюция дает полную СПВ, поскольку произвольный вывод в смысле общей резолюции можно за конечное число шагов преобразовать в регулярный. Очевидным образом общая резолюция полиномиально моделирует регулярную, как свой частный случай.

В работе [1], помимо общей и регулярной резолюции, рассматриваются эти же СПВ, дополненные возможностью использовать в отношении исходной КНФ правило расширения, описанное во введении (вторая цитата из [1]). Данное правило предлагается применять в качестве схемы порождения аксиом, а правило резолюций — в качестве правила вывода. Несложно видеть, что одна итерация этого правила расширения соответствует вводу новой переменной α , а приписываемые дизъюнкты кодируют эквивалентность $\alpha \equiv (\bar{\beta} \vee \bar{\gamma})$. Ввод α можно рассматривать как обогащение исходной аксиоматики новой аксиомой, не влияющей на противоречивость опровергаемого утверждения.

Данного простейшего правила расширения Г. С. Цейтину оказалось вполне достаточно для демонстрации большей мощности расширенной СПВ по сравнению с исходной, в качестве которой выступала регулярная резолюция. Для этой цели в [1] вводится натуральное семейство логических противоречий S_T , в основе которого лежат системы линейных уравнений над $\text{GF}(2)$. Каждой КНФ C из S_T ставится в соответствие противоречивая КНФ C^\sim , полученная в результате (вообще говоря, многократного) применения к C правила расширения Цейтина.

Пусть C — невыполнимая КНФ. Через $N^*(C)$ обозначим наименьшее число резольвент, порождаемых регулярной резолюцией при доказательстве противоречивости C . Ставится вопрос о поведении $N^*(C)$ и $N^*(C^\sim)$, если C пробегает S_T . Далее приведен основной результат работы [1].

Теорема 2 [1]. Для величин $N^*(C)$, $N^*(C^\sim)$ и некоторой константы c справедливы следующие соотношения:

- 1) $N^*(C) \geq 2^c \sqrt{m(C)}$, где $m(C)$ — число дизъюнктов в $C \in S_T$;
- 2) $N^*(C) \geq 2^c \sqrt[3]{N^*(C^\sim)}$.

В первом пункте утверждается, что сложность регулярной резолюции не ограничивается сверху никаким полиномом. Второй пункт означает, что регулярная резолюция не моделирует полиномиально свой расширенный вариант.

В дополнение отметим, что задача доказательства противоречивости формул из S_T может быть решена за полиномиальное время, поскольку возможно ее сведение к задаче доказательства несовместности систем линейных уравнений над $\text{GF}(2)$ [2].

В зарубежных исследованиях некоторый интерес к проблемам сложности пропозиционального вывода начинает проявляться лишь с середины 70-х годов XX века. Настоящий бум результатов в этой области породила статья Армина Хакена 1985 г. «Труднорешаемость резолюций» [17]. В данной работе была установлена неполиномиальность общей резолюции на классе противоречий, известных как «формулы Дирихле».

Формулы Дирихле были введены С. Куком и Р. Рекхау в уже упоминавшейся работе [14]. Данные формулы представляют собой пропозициональные кодировки отрицания известного принципа Дирихле, в соответствии с которым при любом размещении m голубей по n , $n < m$, клеткам найдется клетка, в которой окажется более одного голубя. Таким образом, фраза: «существует такое размещение m голубей по n , $n < m$, клеткам, при котором в каждой клетке сидит не более одного голубя» является логическим противоречием для любых натуральных m и n . Принцип Дирихле и его отрицание допускают простые интерпретации в рамках исчисления высказываний. Введем для этой цели булевы переменные x_{ij} :

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-й голубь сидит в } j\text{-й клетке;} \\ 0, & \text{если } i\text{-й голубь не сидит в } j\text{-й клетке.} \end{cases}$$

Формула RHP_n^m [18] определяется следующим образом:

$$\left(\&_{i=1}^m \vee_{j=1}^n x_{ij} \right) \& \left(\&_{j=1}^n \&_{1 \leq i_1 < i_2 \leq m} (\bar{x}_{i_1 j} \vee \bar{x}_{i_2 j}) \right).$$

Часть $\&_{i=1}^m \vee_{j=1}^n x_{ij}$ означает, что каждый голубь сидит в некоторой клетке, а часть $\&_{j=1}^n \&_{1 \leq i_1 < i_2 \leq m} (\bar{x}_{i_1 j} \vee \bar{x}_{i_2 j})$ означает, что ни в одной клетке не сидит более одного голубя. Некоторое пояснение: тот факт, что никакие два голубя не сидят вместе в клетке с номером j , очевидно, можно записать так:

$$\overline{(x_{1j} \& x_{2j})} \& \overline{(x_{1j} \& x_{3j})} \& \dots \& \overline{(x_{1j} \& x_{mj})} \& \overline{(x_{2j} \& x_{3j})} \& \dots \& \overline{(x_{m-1j} \& x_{mj})}.$$

В силу сказанного выше при $m > n$ формула RHP_n^m представляет собой логическое противоречие.

Первоначальный результат А. Хакена состоял в том, что всякое доказательство противоречивости формулы RHP_n^{n+1} посредством общей резолюции потребует порождения экспоненциального от n числа резольвент. Три года спустя результат А. Хакена был усилен С. Бассом и Д. Тураном, которые в работе [18] показали, что всякое резольтивное опровержение формулы RHP_n^m , $m > n$, порождает не менее чем $1/2 \cdot (3/2)^{\frac{n^2}{50m}}$ резольвент. Обзор дальнейших результатов в данном направлении см. в работе [13].

Для наших целей, однако, важность представляют результаты А. Хакена, конспективно изложенные в заключительной части работы [17]. Речь идет об использовании в терминологии А. Хакена «расширенной резолюции». Основная идея А. Хакена восходит к [14] и состоит в возможности полиномиального сведения проблемы опровержения RHP_n^{n+1} к проблеме опровержения RHP_{n-1}^n . Многократное применение данного сведения тем не менее приводит к экспоненциальному разрастанию формулы и не может быть задействовано напрямую. Выходом из этой ситуации является использование преобразований Цейтина. А. Хакен описывает (правда, очень схематично) полиномиальную по сложности процедуру сведения проблемы опровержения RHP_n^{n+1} к проблеме опровержения RHP_1^2 . Данная процедура представляет собой итеративную последовательность преобразований Цейтина, в ходе которой задействуются $O(n^4)$ дополнительных переменных. Данный факт означает, что СПВ, в которой общая резолюция

дополнена возможностью применения преобразований Цейтина, является полиномиально ограниченной на семействе формул $\{RHP_n^{n+1} : n \in N\}$. Сказанное позволяет заключить, что формулы RHP_n^m являются легкими для такой СПВ и при любых $m > n$ («лишних голубей» можно не принимать во внимание).

Резюмируя сказанное, следует отметить, что и собственно преобразования Цейтина, и правила расширения, используемые как инструмент дополнения опровергаемого утверждения C новыми фактами, не влияющими на противоречивость C , могут приводить к значительному сокращению длины опровержения. Относительно предельной мощности таких расширенных СПВ в этом смысле мало что известно. Далее следует цитата из [17] по данному поводу.

«Одной из целей является доказательство того, что расширенные резолюции также неполиномиальны. Эта задача кажется очень трудной, поскольку, используя правила расширения, мы можем моделировать мета-рассуждения о том, что данная формула является противоречием. Возможно, вопрос о сложности расширенной резолюции будет решен только после решения проблемы равенства классов NP и co-NP».

2.3. Двоичные диаграммы решений и СПВ на их основе

Двоичные диаграммы решений (Binary Decision Diagrams, BDD) — класс ориентированных помеченных графов, используемых для работы с булевыми функциями. Первое описание BDD было приведено в работе [19], однако важные свойства BDD как структуры данных, используемой для манипулирования булевыми функциями, были описаны намного позже — в [20].

Стандартно BDD определяется как направленный ациклический граф, в котором выделена одна вершина с входной степенью 0, называемая корнем, и две вершины с выходной степенью 0, называемые терминальными. Терминальные вершины помечены константами 0 и 1. Все остальные вершины помечаются переменными из множества $X = \{x_1, \dots, x_n\}$. Из любой вершины, за исключением терминальных, выходят в точности 2 дуги. Одну дугу обычно рисуют пунктирной, а другую — сплошной линией. Дуга, обозначенная пунктиром, называется low-ребром, дуга, обозначенная сплошной линией, называется high-ребром. Наиболее нагляден процесс построения BDD из двоичных деревьев решений, представляющих булевы функции [21]. Здесь и далее подразумеваются всюду определенные булевы функции.

Если склеить в одну вершину все листья дерева решений некоторой булевой функции, помеченные 0, и то же самое проделать с листьями, помеченными 1, получится BDD.

Если произвольный путь π в BDD из корня в терминальную вершину не содержит вершин, помеченных одинаковыми переменными, и его прохождение подчинено общему для всех путей порядку (например, $x_1 \prec x_2 \prec \dots \prec x_{n-1} \prec x_n$), то такая BDD называется упорядоченной (Ordered Binary Decision Diagram, OBDD). В записи $x_1 \prec \dots \prec x_n$ здесь и далее подразумевается, что корень рассматриваемой OBDD помечен переменной x_1 . Зафиксированный указанным образом порядок на OBDD будем называть порядком означивания переменных.

При использовании BDD как структур данных, представляющих булевы функции, следует различать разные вершины BDD, помеченные одной и той же переменной. Далее для этой цели используем обозначения $v_1(x), v_2(x), \dots$. Вершины, соединенные с нетерминальной вершиной v исходящими из нее low- и high-ребрами, обозначаются

$\text{low}(v)$ и $\text{high}(v)$ соответственно. Также используем обозначение $\text{var}(v(x)) = x$ или более краткое $\text{var}(v) = x$.

В произвольной OBDD можно выделять фрагменты (подграфы), которые сами являются OBDD. Для этой цели достаточно объявить соответствующую нетерминальную вершину корнем BDD. Идея сокращенной OBDD (Reduced Ordered Binary Decision Diagram, ROBDD) заключается в склейке повторяющихся фрагментов: ROBDD-граф не должен содержать одинаковых OBDD-подграфов меньших размерностей. Таким образом, ROBDD можно рассматривать как наиболее сжатое графическое представление некоторой булевой функции. Сказанное означает, что ROBDD — это OBDD, в которой:

- 1) равенства $\text{var}(v) = \text{var}(u)$, $\text{high}(v) = \text{high}(u)$, $\text{low}(v) = \text{low}(u)$ означают, что $v = u$;
- 2) для любой нетерминальной вершины v имеет место $\text{high}(v) \neq \text{low}(v)$.

Р. Брайантом в 1986 г. [20] было показано, что любая всюду определенная булева функция при фиксированном порядке означивания переменных имеет единственное (с точностью до изоморфизма соответствующих графов) ROBDD-представление.

Двоичные диаграммы решений, а точнее ROBDD, можно использовать для решения систем логических уравнений. Подробным обзором на эту тему является работа [22].

Основным алгоритмом работы с ROBDD является описанный в [20] алгоритм APPLY. Данный алгоритм по паре ROBDD $B(f_1)$ и $B(f_2)$ булевых функций f_1 и f_2 над множеством булевых переменных $X = \{x_1, \dots, x_n\}$ строит ROBDD булевой функции $f_3 = f_1 * f_2$, где $*$ — произвольная бинарная логическая связка. При этом означивание переменных в $B(f_1)$ и $B(f_2)$ должно быть подчинено одному порядку. Факт построения посредством APPLY ROBDD $B(f_3)$ по известным $B(f_1)$ и $B(f_2)$ обозначается следующим образом:

$$B(f_3) = \text{APPLY}(B(f_1) * B(f_2)).$$

Сложность алгоритма APPLY построения $B(f_3)$ есть $O(|B(f_1)| \cdot |B(f_2)|)$, где через $|B|$ обозначено число вершин в ROBDD B .

Основа алгоритма APPLY чрезвычайно проста и заключается в одновременном прохождении обеих ROBDD в соответствии с выбранным порядком означивания переменных. Такому обходу $B(f_1)$ и $B(f_2)$ ставится в соответствие дерево $T(f_3)$, представляющее функцию f_3 . Каждая вершина в $T(f_3)$ определяется парой координат — соответствующими текущими вершинами в $B(f_1)$ и $B(f_2)$. Так как порядок означивания переменных в $B(f_1)$ и $B(f_2)$ совпадает, то при построении $T(f_3)$ не происходит возвратов, поэтому число вершин в нем не превосходит величины $|B(f_1)| \cdot |B(f_2)|$. После построения дерева $T(f_3)$ оно усекается до ROBDD. Процедура усекаения линейна от размерности $T(f_3)$. В целом, однако, возможна более эффективная схема, использующая в своей основе принцип динамического программирования. В соответствии с ней построение ROBDD $B(f_3)$ происходит, минуя этап построения $T(f_3)$. При этом $B(f_3)$ строится как динамически заполняемая таблица — новая вершина заносится в таблицу лишь тогда, когда таблица не содержит дубликата этой вершины.

Так как ROBDD является структурой, в рамках которой решаются вопросы совместности произвольных систем логических уравнений, то общую схему построения ROBDD-представлений булевых функций, заданных пропозициональными формулами, можно рассматривать как некоторую СПВ. Единственной известной нам работой, в которой подробно изучается такая система, является [6].

Основа подхода, предлагаемого в [6], состоит в том, что для опровержения или доказательства выполнимости произвольной пропозициональной формулы φ над множеством булевых переменных $X = \{x_1, \dots, x_n\}$ достаточно построить ROBDD булевой функции $f_\varphi : \{0, 1\}^n \rightarrow \{0, 1\}$, которую данная формула задает. Формула φ невыполнима тогда и только тогда, когда f_φ есть тождественный ноль, то есть когда ROBDD $B(f_\varphi)$ состоит из одной терминальной вершины 0. ROBDD-вывод для произвольной пропозициональной формулы φ над X в [6] определяется как рекурсивное применение алгоритма APPLY. Начальным (базовым) множеством является множество ROBDD, представляющих булевы функции вида x_i , $i \in \{1, \dots, n\}$. Если в формуле φ присутствует m логических связок, то APPLY вызывается m раз.

Таким образом, имеем СПВ, в которой APPLY-процедура используется в качестве итеративно применяемого правила вывода наподобие правила резолюции или правила единичного дизъюнкта в СПВ на базе алгоритма DPLL (см., например [23]). Итогом каждой итерации в рассматриваемой системе доказательств является некоторая ROBDD. Критическим параметром сложности вывода в данном случае является максимальный размер (число вершин) ROBDD, возникающей в процессе доказательства. Полнота описанной СПВ очевидна — в результате конечной последовательности итераций будет доказана выполнимость формулы φ или же данная формула будет опровергнута. Доказательства данного типа далее называем ROBDD-доказательствами (ROBDD-выводами, ROBDD-опровержениями).

Далее кратко остановимся на некоторых результатах работы [6]. Особо оговоримся, что направление, выбранное в [6], по-видимому, правильно и перспективно, но ключевые результаты выглядят не вполне убедительно и требуют дальнейшего совершенствования.

Основным результатом работы [6] является вывод о том, что ROBDD-доказательства не моделируют полиномиально резолютивные, а резолютивные доказательства не моделируют полиномиально ROBDD-доказательства.

Первый факт устанавливается при помощи формул Дирихле, а именно рассматриваются натуральные семейства формул следующего вида:

$$C_{m,n} = \&_{i=1}^m \vee_{j=1}^n x_{ij}; \quad Q_{m,n} = \&_{j=1}^n \&_{1 \leq i_1 < i_2 \leq m} (\bar{x}_{i_1 j} \vee \bar{x}_{i_2 j});$$

$$PHP_n^m = C_{m,n} \cdot Q_{m,n}.$$

Далее показывается, что при любом порядке означивания соответствующих булевых переменных в ходе ROBDD-доказательства применительно к формулам PHP_n^m ($m > n$) обязательно возникают ROBDD, функция числа вершин которых растет как $O(1,63^n)$. Данный факт в контексте результатов предыдущего пункта позволяет сказать о большей мощности расширенной резолюции в сравнении с СПВ на основе ROBDD.

Далее в [6] демонстрируется экспоненциальная сложность ROBDD-вывода на формулах $CR_{n,n} = C_{n,n} \cdot R_{n,n}$, где $R_{m,n} = \overline{Q_{m,n}}$ (порядок сложности аналогичен приведенному выше: $O(1,63^n)$). Затем рассматривается семейство логических противоречий вида $y \cdot (\bar{y} \cdot CR_{n,n})$ и отмечается, что, в силу сказанного, всякое ROBDD-опровержение формул данного вида содержит ROBDD-доказательство для формул $CR_{n,n}$ и поэтому экспоненциально. Перед построением резолютивного доказательства к формулам $y \cdot (\bar{y} \cdot CR_{n,n})$ предлагается применить преобразования Цейтина с целью приведения их к КНФ. После этого делается вывод о существовании линейного по сложности доказательства противоречивости полученных формул посредством использования только правила единичного дизъюнкта (являющегося частным случаем правила резолюции). Отметим, что данный результат выглядит весьма искусственным.

В работе [6] также устанавливается полиномиальная оценка сложности для ROBDD-доказательств так называемых бикондициональных формул (Biconditional Formula). Данный класс образован пропозициональными формулами, содержащими литералы над множеством булевых переменных, скобки, а также логическую эквивалентность. Бикондициональные формулы очень просто генерировать по словам над произвольными конечными алфавитами. Например, слову $abcd$ можно поставить в соответствие бикондициональные формулы типа

$$a \equiv (b \equiv (c \equiv (d \equiv a))), a \equiv (\neg b \equiv (c \equiv (\neg d \equiv a))), \dots,$$

являющиеся пропозициональными формулами над множеством булевых переменных $\{a, b, c, d\}$. В [6] показывается, что для всякой бикондициональной формулы φ существует ROBDD-вывод (в контексте данного выше определения), функция сложности которого ведет себя как $O(|\varphi|^3)$, где через $|\varphi|$ обозначено число встречающихся в φ символов.

Далее в [6] строится одно натуральное семейство бикондициональных формул. Сначала каждому $n \in N$ ставится в соответствие специальным образом построенное слово в некотором алфавите подходящей мощности

$$p_1 p_2 \dots p_{n \cdot 2^n}, \tag{6}$$

в котором каждая буква алфавита встречается дважды. Этому слову сопоставляется бикондициональная формула

$$S_n = p_1 \equiv (p_2 \equiv \dots \equiv (p_{n \cdot 2^n - 1} \equiv p_{n \cdot 2^n}) \dots).$$

Тем самым получается семейство формул $\{S_n : n \in N\}$. Конструкция слова (6) такова, что $\neg S_n$ является логическим противоречием. Затем каждая формула $\neg S_n$ переводится в КНФ при помощи преобразований Цейтина. Полученное так натуральное семейство КНФ обозначается через $\{C(\neg S_n) : n \in N\}$. Особо отметим тот факт, что порядок роста размера формул $C(\neg S_n)$ есть $O(n \cdot 2^n)$.

Второй основной результат работы [6] состоит в том, что во всяком доказательстве формул $C(\neg S_n)$ посредством общей резолюции функция числа порождаемых резольвент мажорирует величину $2^{O(2^n/n)}$. Данный факт устанавливается при помощи одного результата работы [24], выявляющего связь между сложностью резолютивного опровержения и длиной (то есть числом вхождений литералов) возникающих в ходе этого опровержения резольвент.

С другой стороны, сложность ROBDD-опровержений бикондициональных формул $\{\neg S_n : n \in N\}$, в силу сказанного выше, растет как полином от $n \cdot 2^n$.

Отметим, что «естественному» восприятию данного результата мешает тот факт, что $\{C(\neg S_n) : n \in N\}$ — это семейство формул, которое не порождается эффективно по своим натуральным индексам (в отличие, например, от семейства $\{PHP_n^{n+1} : n \in N\}$). Еще одним негативным моментом является то, что резолюция применяется к формулам $C(\neg S_n)$, а ROBDD-доказательства — к формулам $\neg S_n$. Сами авторы [6] отмечают, что ROBDD-доказательства в применении к формулам $C(\neg S_n)$ имеют экспоненциальную сложность.

3. Приведение систем логических уравнений к нормальным формам; смежные вопросы

3.1. Преобразования логических уравнений и систем

Одной из важнейших функций преобразований Цейтина, как следует из выше-сказанного, является приведение систем логических уравнений к некоторым форма-

там, удобным с точки зрения дальнейших исследований. Простейшим примером могут служить преобразования, при помощи которых доказывается NP-полнота задачи 3-SAT. Ключевым моментом здесь состоит в переходе от произвольного дизъюнкта вида $(z_1 \vee \dots \vee z_k)$, где $k > 3$ и $z_i, i \in \{1, \dots, k\}$, — литералы над множеством $X = \{x_1, \dots, x_n\}$, к КНФ

$$(z_1 \vee \dots \vee z_{k-2} \vee u) \cdot (u \vee \overline{z_{k-1}}) \cdot (u \vee \overline{z_k}) \cdot (\overline{u} \vee z_{k-1} \vee z_k)$$

применением преобразования Цейтина, в котором используется эквивалентность $u \equiv (z_{k-1} \vee z_k)$.

Далее при помощи преобразований Цейтина дадим очень простое доказательство известного результата об NP-полноте задачи определения совместности билинейных (т. е. степени 2) систем над полем GF(2).

Теорема 3. Задача определения совместности билинейных систем над полем GF(2) является NP-полной.

Доказательство. Сведем к задаче определения совместности билинейной системы над GF(2) задачу проверки выполнимости произвольной 3-КНФ C , то есть КНФ, составленной из трехлитеральных дизъюнктов. Данная задача NP-полна. Пусть C задана над множеством булевых переменных $X = \{x_1, \dots, x_n\}$. Для каждой переменной x_i , входящей в КНФ C без инверсии, введем новую переменную u_i , заменим каждое вхождение x_i на $\overline{u_i}$ и конъюнктивно припишем к C выражение $(x_i \vee u_i) \cdot (\overline{x_i} \vee \overline{u_i})$, кодирующее эквивалентность $\overline{u_i} \equiv x_i$. Прделаем аналогичные преобразования относительно всех переменных из X , входящих в C без инверсии. Итоговую КНФ обозначим через C^\sim . По теореме 1 существует биекция между множествами решений уравнений $C = 1$ и $C^\sim = 1$. В C^\sim присутствуют группы дизъюнктов двух видов: это трехлитеральные дизъюнкты, составленные только из переменных с инверсиями, и группы дизъюнктов вида $(x_i \vee u_i) \cdot (\overline{x_i} \vee \overline{u_i})$. Заметим, что уравнение $(x_i \vee u_i) \cdot (\overline{x_i} \vee \overline{u_i}) = 1$ эквивалентно линейному уравнению $x_i \oplus u_i \oplus 1 = 0$ над полем GF(2). Рассмотрим произвольный дизъюнкт, состоящий из трех переменных с инверсией: $(\overline{x} \vee \overline{y} \vee \overline{z})$. Введем новую переменную v , заменим формулу $(\overline{x} \vee \overline{y})$ формулой \overline{v} . По теореме 1 существует биекция между множествами решений уравнения $(\overline{x} \vee \overline{y} \vee \overline{z}) = 1$ и системы

$$\begin{cases} (\overline{v} \vee \overline{z}) = 1, \\ (\overline{v} \equiv (\overline{x} \vee \overline{y})) = 1. \end{cases}$$

Данная система эквивалентна системе

$$\begin{cases} v \cdot z = 0, \\ (v \equiv x \cdot y) = 1, \end{cases}$$

которая, в свою очередь, эквивалентна системе

$$\begin{cases} v \cdot z = 0, \\ x \cdot y \oplus v = 0. \end{cases} \quad (7)$$

Система (7) состоит из двух билинейных уравнений над полем GF(2). Таким образом, за линейное от объема двоичного кода КНФ C время можно при помощи преобразований Цейтина перейти от задачи выполнимости 3-КНФ C к задаче определения совместности системы уравнений над полем GF(2), в которой фигурируют линейные уравнения вида $x_i \oplus u_i \oplus 1 = 0$, а также билинейные уравнения, образующие подсистему вида (7). В соответствии с теоремой 1 полученная система совместна тогда и

только тогда, когда выполнима исходная КНФ. Тем самым установлена NP-полнота задачи определения совместности произвольной системы билинейных уравнений над полем $GF(2)$. ■

В работе [25] при помощи техники, похожей на технику доказательства теоремы 3, была исследована проблема декомпозиции системы логических уравнений вида $КНФ = 1$ на полиномиально разрешимые подсистемы с сохранением свойства консервативности.

Определение 1. Пусть существует алгоритмически вычислимая за полиномиальное время функция τ , преобразующая систему логических уравнений U в систему логических уравнений $\tau(U)$, причем

- 1) существует биекция ω между множествами решений систем U и $\tau(U)$;
- 2) от произвольного решения системы $\tau(U)$ за полиномиальное в общем случае время осуществим переход к соответствующему в смысле ω решению U .

В этом случае системы U и $\tau(U)$ называем полиномиально консервативно изоморфными (кратко консервативно изоморфными), а функцию τ — консервативным изоморфизмом.

Простейшие консервативные изоморфизмы, как показывает теорема 1, можно получить, используя преобразования Цейтина.

Далее рассматриваются несколько классов логических уравнений. Во-первых, это линейные системы над $GF(2)$ вида

$$\begin{cases} x_{11} \oplus x_{12} = 1, \\ \dots \\ x_{s1} \oplus x_{s2} = 1. \end{cases}$$

Для систем данного типа будем также использовать обозначение $U(x_{11}, \dots, x_{s2}) = 1$. Второй тип уравнений — это уравнения вида $C^2(x_1, \dots, x_n) = 1$, где $C^2(x_1, \dots, x_n)$ — КНФ, каждый дизъюнкт которой содержит два литерала. Третий тип уравнений — это уравнения вида $H_-(x_1, \dots, x_n) = 1$ или $H_+(x_1, \dots, x_n) = 1$. Здесь $H_-(x_1, \dots, x_n)$ — это хорновская (используем также термин «негативно хорновская») КНФ, то есть КНФ, каждый дизъюнкт которой содержит не более одной переменной без инверсии (см., например, [26]), а $H_+(x_1, \dots, x_n)$ — позитивно хорновская КНФ, то есть КНФ, каждый дизъюнкт которой содержит не более одной переменной с инверсией. В некоторых источниках негативно и позитивно хорновские КНФ именуется соответственно слабо положительными и слабо отрицательными КНФ (см., например, [27]). Хорошо известно, что для задач поиска решений логических уравнений перечисленных классов существуют полиномиальные алгоритмы. В [25] установлена справедливость следующего утверждения.

Теорема 4 [25]. Для логического уравнения вида $C(x_1, \dots, x_n) = 1$, где $C(x_1, \dots, x_n)$ — произвольная КНФ над множеством булевых переменных $X = \{x_1, \dots, x_n\}$, существует консервативно изоморфная ему система логических уравнений любого из перечисленных ниже типов:

$$\begin{cases} U(x_{11}, \dots, x_{s(n)2}) = 1, \\ H_+(y_1, \dots, y_{p(n)}) = 1; \end{cases} \quad \begin{cases} U(x_{11}, \dots, x_{s(n)2}) = 1, \\ H_-(y_1, \dots, y_{p(n)}) = 1; \end{cases} \quad \begin{cases} H_-(y_1^1, \dots, y_{p(n)}^1) = 1, \\ H_+(y_1^2, \dots, y_{q(n)}^2) = 1; \end{cases}$$

$$\begin{cases} C^2(y_1^1, \dots, y_{p(n)}^1) = 1, \\ H_+(y_1^2, \dots, y_{q(n)}^2) = 1; \end{cases} \quad \begin{cases} C^2(y_1^1, \dots, y_{p(n)}^1) = 1, \\ H_-(y_1^2, \dots, y_{q(n)}^2) = 1. \end{cases}$$

Здесь $p(n) \leq 2n$; $q(n) \leq 2n$; $s(n) \leq n$.

3.2. Исследование свойств дискретных автоматов

Наблюдающееся в последние годы бурное развитие алгоритмической базы решения SAT-задач сделало возможным использование SAT-подхода в исследовании многих практически важных классов дискретных управляющих систем (в терминологии [28] — дискретных автоматов). Преобразования Цейтина при этом являются основным инструментом, реализующим переход от исходной задачи к SAT-задаче.

Следующая конструкция (относящаяся, по-видимому, к категории фольклорных) используется в задачах верификации логических микросхем (см., например, [29]). Предположим, что даны две схемы $S(f)$ и $S(g)$, реализующие булевы функции $f : \{0, 1\}^n \rightarrow \{0, 1\}$ и $g : \{0, 1\}^n \rightarrow \{0, 1\}$ в произвольном полном базисе B из функциональных элементов. Задача состоит в распознавании по схемам $S(f)$ и $S(g)$ эквивалентности функций f и g . Рассмотрим следующую схему $S(h)$ (рис. 1).

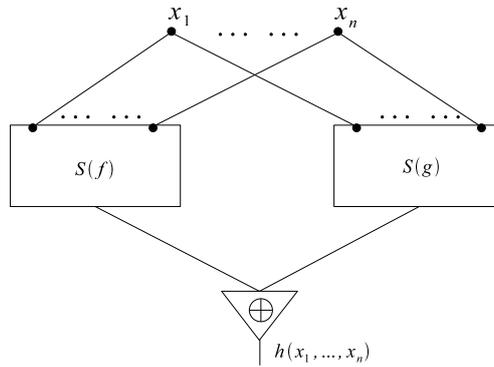


Рис. 1. Схема $S(h)$

Данная схема представляет булеву функцию $h = f \oplus g$. Очевидно, что функции f и g эквивалентны тогда и только тогда, когда функция h есть тождественный ноль на $\{0, 1\}^n$. Используя преобразования Цейтина, можно поставить в соответствие схеме $S(h)$ систему логических уравнений

$$\begin{cases} U_1(y_1, \dots, y_{p(n)}) = 1, \\ \dots \\ U_{q(n)}(y_1, \dots, y_{p(n)}) = 1 \end{cases} \quad (8)$$

над множеством булевых переменных $Y = \{y_1, \dots, y_{p(n)}\}$, $X \subseteq Y$, $p(n)$, $q(n)$ — некоторые полиномы. При этом строго показывается, что данная система несовместна тогда и только тогда, когда функция h есть тождественный ноль на $\{0, 1\}^n$. После этого, опять-таки при помощи преобразований Цейтина, можно перейти от задачи доказательства несовместности (8) к задаче доказательства невыполнимости некоторой КНФ.

В работе [11] С. А. Куком была доказана фундаментальная теорема о пропозициональном кодировании формальных вычислительных моделей, положившая начало

теории NP-полноты. Доказательство Кука было проведено в контексте машин Тьюринга и использовало преобразования, идейно схожие с преобразованиями Цейтина, но отличающиеся от них, вообще говоря, отсутствием консервативности. В [30] приведено использующее преобразования Цейтина доказательство варианта теоремы Кука в отношении задач обращения дискретных функций, вычислимых на двоичных аналогах машин с неограниченными регистрами в формализме Н. Катленда [31]. Эта формальная модель более близка современным компьютерам, чем машина Тьюринга, а развитая в [30] техника допускает перенос на пропозициональное кодирование алгоритмов вычисления дискретных функций, записанных на высокоуровневых языках программирования. Данный факт дает основу для разработки и программной реализации технологии обращения дискретных функций из класса, значимого различными практическими приложениями. Здесь кратко остановимся на основных результатах, полученных в данном направлении.

Обозначим через $f = \{f_n : n \in N\}$ натуральное семейство дискретных функций вида

$$f_n : \{0, 1\}^n \rightarrow \{0, 1\}^*,$$

определенных всюду на $\{0, 1\}^n$ ($\text{dom } f_n = \{0, 1\}^n$) и алгоритмически вычислимых за полиномиальное от n время. Проблемой обращения произвольной функции f_n из такого семейства называется следующая задача: по произвольному $y \in \text{range } f_n \subset \{0, 1\}^*$ и известному алгоритму вычисления f (программе для выбранной вычислительной модели) требуется найти такой $x \in \{0, 1\}^n$, что $f_n(x) = y$. Данную проблему будем называть проблемой обращения функции f_n в точке $y \in \text{range } f_n$.

Теорема 5 [30]. Для любого семейства f дискретных функций из определенного выше класса существует алгоритм с полиномиально от n ограниченной сложностью, который, получая на входе n и $y \in \text{range } f_n$, преобразует проблему обращения f_n в точке y в проблему поиска решений логического уравнения вида $C(x_1, \dots, x_{q(n)}) = 1$, где $q(n)$ — некоторый полином, а $C(x_1, \dots, x_{q(n)})$ — выполнимая КНФ над множеством булевых переменных $\{x_1, \dots, x_{q(n)}\}$.

Это утверждение вместе с теоремой 1 составляет основу пропозиционального подхода к обращению дискретных функций из рассматриваемого класса. В соответствии с данным подходом алгоритм вычисления f_n представляется в виде системы логических уравнений, от которой затем при помощи преобразований Цейтина осуществляется переход к одному уравнению вида КНФ = 1. Получаемая SAT-задача всегда имеет решение, от которого (в силу теоремы 1) можно эффективно перейти к искомому прообразу точки $y \in \text{range } f_n$, то есть к такому вектору $x \in \{0, 1\}^n$, что $f_n(x) = y$. Для решения SAT-задач можно использовать богатый арсенал наработанных методов и алгоритмов [32].

Данный подход оправдал себя по отношению к таким аргументированно трудным задачам обращения дискретных функций, как задачи криптоанализа некоторых поточных систем шифрования [8, 33–35].

4. Преобразования Цейтина в задачах подсчёта

Здесь преобразования Цейтина используются в исследовании сложности некоторых задач на подсчёт. Определяемый ниже класс #P-полных задач впервые был введен Л. Дж. Валиантом в работе [36].

Определение 2 (см., например, [37]). Класс #P образован такими функциями вида $f : \{0, 1\}^* \rightarrow N \cup \{0\}$, что для всякого $x \in \{0, 1\}^*$ и некоторой полиномиаль-

ной детерминированной машины Тьюринга M существует в точности $f(x)$ таких слов $y \in \{0, 1\}^*$, длина которых в общем случае ограничивается полиномом от $|x|$, что M , получив на входе слово $x|y$, останавливается в положении «да». Функция f , равно как и проблема вычисления её значения, называется $\#P$ -полной, если $f \in \#P$ и любая функция $g \in \#P$ вычисляется на полиномиальной машине Тьюринга с оракулом, выдающим значение функции f .

Примером $\#P$ -полной является функция, которая, получив на вход произвольную КНФ C , выдает число наборов, выполняющих C [36]. Несложно понять, что любая $\#P$ -полная проблема является также и NP -трудной.

4.1. Сложность проблемы подсчета числа выполняющих наборов хорновских КНФ

В работе [38] было показано, что проблемы подсчета числа решений некоторых систем логических уравнений являются $\#P$ -полными, и это при том, что проблемы совместности этих систем решаются за полиномиальное время. Наиболее ярким в этом направлении является результат о $\#P$ -полноте проблемы подсчета числа выполняющих наборов монотонной 2-КНФ, т. е. КНФ из двухлитеральных дизъюнктов, в которую все переменные входят с инверсиями или все — без инверсий. Прямым следствием данного факта является $\#P$ -полнота проблемы подсчета числа выполняющих наборов произвольной хорновской КНФ. К сожалению, детальное доказательство этого результата, являющееся итогом шести редукций, требуется извлекать из двух работ [36, 38]. Проблемы подсчета решений различных классов логических уравнений подробно исследовались также в [27, 39]. В работе [27] было дано еще одно доказательство $\#P$ -полноты проблемы подсчета числа наборов, выполняющих хорновские КНФ. Это доказательство также весьма сложно в техническом плане.

Далее приводится новое доказательство $\#P$ -полноты проблемы подсчета наборов, выполняющих произвольную хорновскую КНФ, использующее в своей основе преобразования Цейтина. Данное доказательство существенно проще доказательств, имеющих в упомянутых выше работах.

Рассмотрим произвольное логическое уравнение вида

$$\Phi(x_1, \dots, x_n) = 1 \quad (9)$$

над множеством булевых переменных $X = \{x_1, \dots, x_n\}$. Через $\#_X \Phi(X)$ обозначим число наборов значений переменных из X , являющихся решениями (9). Очевидным образом справедлив следующий факт.

Лемма 1. Рассматривается уравнение (9). Пусть $L(x_{i_1}, \dots, x_{i_r})$ — произвольная формула ИВ от булевых переменных x_{i_1}, \dots, x_{i_r} , $\{x_{i_1}, \dots, x_{i_r}\} \subseteq X$. Тогда имеет место соотношение

$$\#_X \Phi(X) = \#_X (\Phi(X) \cdot L(x_{i_1}, \dots, x_{i_r})) + \#_X (\Phi(X) \cdot \neg L(x_{i_1}, \dots, x_{i_r})).$$

Установим справедливость следующей теоремы.

Теорема 6. Проблема подсчета числа наборов, выполняющих произвольную хорновскую КНФ, является $\#P$ -полной.

Доказательство. Рассмотрим произвольное уравнение вида (1), т. е. уравнение $C(x_1, \dots, x_n) = 1$, где C — КНФ. Проблема подсчета числа решений (1) в общем случае $\#P$ -полна. Используем конструкцию, примененную при доказательстве теоремы 3.

Произвольному литералу $x_i, i \in \{1, \dots, n\}$, входящему в $C(x_1, \dots, x_n)$, сопоставим новую булеву переменную y_i , заменим все вхождения литерала x_i в C на вхождения литерала \bar{y}_i и припишем к C (через знак конъюнкции) выражение $(x_i \vee y_i) \cdot (\bar{x}_i \vee \bar{y}_i)$, кодирующее эквивалентность $\bar{y}_i \equiv x_i$. Повторив данную операцию не более n раз, перейдем от уравнения (1) к уравнению следующего вида:

$$(x_1 \vee y_1) \cdot \dots \cdot (x_t \vee y_t) \cdot H_-^1(X_1) = 1. \quad (10)$$

Здесь $H_-^1(X_1)$ — хорновская КНФ над множеством булевых переменных $X_1 = \{x_1, \dots, x_n\} \cup \{y_1, \dots, y_t\}$, $t \leq n$, а точнее, монотонная КНФ, в которую все переменные входят с инверсиями. В силу теоремы 1 уравнения (1) и (10) консервативно изоморфны. Если $t = 1$, то по лемме 1

$$\#_{X_1}((x_1 \vee y_1) H_-^1(X_1)) = \#_{X_1} H_-^1(X_1) - \#_{X_1}(\bar{x}_1 \cdot \bar{y}_1 H_-^1(X_1)).$$

Предположим, что $t \geq 2$. Обозначим левую часть (10) через $\Phi(X_1)$. В силу леммы 1 имеем

$$\#_{X_1} \Phi(X_1) = \#_{X_1} H_-^1(X_1) - \#_{X_1}((\bar{x}_1 \cdot \bar{y}_1 \vee \dots \vee \bar{x}_t \cdot \bar{y}_t) H_-^1(X_1)). \quad (11)$$

В отношении формулы $(\bar{x}_1 \cdot \bar{y}_1 \vee \dots \vee \bar{x}_t \cdot \bar{y}_t) \cdot H_-^1(X_1)$ осуществим преобразования Цейтина, используя следующие эквивалентности:

$$\bar{u}_i \equiv \bar{x}_i \cdot \bar{y}_i, \quad i \in \{2, \dots, t\}.$$

В результирующей формуле термы $\bar{x}_i \cdot \bar{y}_i$ заменяются термами \bar{u}_i , $i \in \{2, \dots, t\}$; кроме этого, появятся новые конъюнкции вида

$$(\bar{x}_i \vee u_i) \cdot (\bar{y}_i \vee u_i) \cdot (x_i \vee y_i \vee \bar{u}_i), \quad i \in \{2, \dots, t\}.$$

К дизъюнктам вида $(x_i \vee y_i \vee \bar{u}_i)$ снова применяем преобразования Цейтина, вводя эквивалентности $\bar{v}_i \equiv y_i$ и учитывая при этом появление в итоговой формуле конъюнкций вида $(y_i \vee v_i) \cdot (\bar{y}_i \vee \bar{v}_i)$, $i \in \{2, \dots, t\}$. Результатом перечисленных действий является формула

$$(y_2 \vee v_2) \cdot \dots \cdot (y_t \vee v_t) \cdot (\bar{x}_1 \cdot \bar{y}_1 \vee \bar{u}_2 \vee \dots \vee \bar{u}_t) \cdot H_-^{\sim}(X_2), \quad (12)$$

где $H_-^{\sim}(X_2)$ — хорновская КНФ над множеством булевых переменных

$$X_2 = X_1 \cup \{u_2, \dots, u_t\} \cup \{v_2, \dots, v_t\}.$$

С учетом того факта, что

$$(\bar{x}_1 \cdot \bar{y}_1 \vee \bar{u}_2 \vee \dots \vee \bar{u}_t) = (\bar{x}_1 \vee \bar{u}_2 \vee \dots \vee \bar{u}_t) \cdot (\bar{y}_1 \vee \bar{u}_2 \vee \dots \vee \bar{u}_t),$$

формула (12) преобразуется к виду

$$(y_2 \vee v_2) \cdot \dots \cdot (y_t \vee v_t) \cdot H_-^2(X_2),$$

где $H_-^2(X_2)$ — хорновская КНФ над X_2 . В силу теоремы 1 имеем

$$\#_{X_1}((\bar{x}_1 \cdot \bar{y}_1 \vee \dots \vee \bar{x}_t \cdot \bar{y}_t) \cdot H_-^1(X_1)) = \#_{X_2}((y_2 \vee v_2) \cdot \dots \cdot (y_t \vee v_t) \cdot H_-^2(X_2)).$$

Переходим к задаче вычисления величины

$$\#_{X_2}((y_2 \vee v_2) \cdot \dots \cdot (y_t \vee v_t) \cdot H_-^2(X_2)).$$

В общей сложности повторяем описанную процедуру $t - 1$ раз ($t \geq 2$).

В итоге имеем следующее соотношение, объединяющее все возможные случаи:

$$\#_{X_1} \Phi(X_1) = \sum_{i=1}^t (-1)^{i-1} \#_{X_i} (H_-^i(X_i)) + (-1)^t \#_{X_t} (\bar{r}_t \cdot \bar{s}_t \cdot H_-^t(X_t)). \quad (13)$$

Последнее слагаемое в (13) учитывает тот факт, что

$$\#_{X_t} ((r_t \vee s_t) \cdot H_-^t(X_t)) = \#_{X_t} H_-^t(X_t) - \#_{X_t} (\bar{r}_t \cdot \bar{s}_t \cdot H_-^t(X_t)).$$

Заметим, что переход от (11) к (13) требует времени, в общем случае ограниченного полиномом от объема двоичного кода КНФ $C(x_1, \dots, x_n)$, фигурирующей в исходном уравнении вида (1).

Все сказанное означает, что задачу подсчета числа решений произвольного уравнения вида (1) можно решить за полиномиальное время на оракульной машине Тьюринга, оракул которой, получая на входе произвольную хорновскую КНФ, выдает число выполняющих ее наборов. Тем самым задача подсчета числа наборов, выполняющих произвольную хорновскую КНФ, является $\#P$ -полной. Теорема 6 доказана. ■

4.2. К проблеме аргументации сложности задач построения ROBDD-представлений некоторых булевых функций

Здесь мы возвращаемся к проблеме сравнения эффективности SAT- и ROBDD-подходов к поиску решений логических уравнений. В п. 2.3 были приведены далеко не бесспорные результаты работы [6] по построению абсолютных сравнительных оценок эффективности данных подходов. Следующее утверждение дает условные оценки такого рода, базируясь на известных результатах о структурной сложности некоторых задач подсчёта.

Теорема 7. Проблемы построения ROBDD-представлений булевых функций, заданных хорновскими КНФ, а также КНФ, составленными из двухлитеральных дизъюнктов, не могут быть в общем случае решены за полиномиальное время, если $P \neq NP$.

Доказательство. В работе [20] описан алгоритм SAT-count, который по произвольной ROBDD $B(f)$, представляющей булеву функцию f , за линейное от числа вершин в $B(f)$ время выдает число наборов значений переменных, на которых функция f принимает значение 1. Далее можно использовать результаты Л. Валианта, а также результат теоремы 6 настоящей работы. Если бы существовала полиномиальная по сложности процедура построения ROBDD-представления произвольной булевой функции, заданной, например, в виде хорновской КНФ, то функция числа вершин в получаемом натуральном семействе ROBDD была бы ограничена сверху некоторым полиномом от объема двоичного кода исходных формул. Но тогда по полученной ROBDD можно было бы при помощи алгоритма SAT-count подсчитать число наборов, выполняющих исходную КНФ, за полиномиальное в общем время. Однако данный факт в силу теоремы 6 означал бы, что $P = NP$. Теорема 7 доказана. ■

Заключение

Рассмотрен ряд задач по проблемам вычислительной сложности вывода в исчислении высказываний, сравнительной эффективности различных систем пропозиционального вывода, приведения систем логических уравнений к нормальным формам с сохранением важных свойств, а также по вопросу аргументации вычислительной

сложности некоторых задач на подсчёт. Конструктивную основу большинства приведенных результатов составляют преобразования Цейтина [1].

Автор благодарит сотрудников лаборатории дискретного анализа и прикладной логики ИДСТУ СО РАН за активное обсуждение материала статьи.

ЛИТЕРАТУРА

1. *Цейтин Г. С.* О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. 1968. Т. 8. С. 234–259.
2. *Данцин Е. Я.* Алгоритмика задачи выполнимости // Вопросы кибернетики. Проблемы сокращения перебора. М.: АН СССР, 1987. С. 7–29.
3. *Waisberg M.* Untersuchungen uber den Aussagen kalkul von Heyting // Wiadom. Matemat. 1938. No. 46. P. 45–101.
4. *Tseitin G.* On the complexity of derivation in propositional calculus // Automat. Reasoning. 1983. V. 2. P. 466–483.
5. *Plaisted D., Greenbaum S.* A Structure-preserving Clause Form Translation // J. Symb. Comput. 1986. V. 2. P. 293–304.
6. *Groote J. F., Zantema H.* Resolution and binary decision diagrams cannot simulate each other polynomially // J. Discr. Appl. Math. 2003. No. 130:2. P. 157–171.
7. *Een N., Sorensson N.* Translating Pseudo-Boolean Constraints into SAT // J. Satisf., Boolean Mod. Comp. 2006. No. 2. P. 1–25.
8. *Семенов А. А., Заикин О. С., Беспалов Д. В., Ушаков А. А.* SAT-подход в криптоанализе некоторых систем поточного шифрования // Вычислительные технологии. 2008. Т. 13. № 6. С. 134–150.
9. *Семенов А. А.* О преобразованиях Цейтина в логических уравнениях // Прикладная дискретная математика. Приложение. 2009. № 1. С. 12–13.
10. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986. 384 с.
11. *Cook S. A.* The complexity of theorem-proving procedures // Proc. 3rd Ann. ACM Symp. on Theory of Computing. ACM, 1971. P. 151–159. [Пер.: *Кук С. А.* Сложность процедур вывода теорем // Кибернетический сборник. Новая серия. 1975. Вып. 12. С. 5–15.]
12. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
13. *Razborov A. A.* Proof Complexity of Pigeonhole Principles // LNCS. 2002.V. 2295. P. 100–116.
14. *Cook S. A., Reckhow R.* The relative efficiency of propositional proof systems // J. Symb. Logic. 1979. V. 44. P. 239–251.
15. *Robinson J. A.* A machine-oriented logic based on the resolution principle // J. ACM. 1965. V. 12. No. 1. P. 23–41. [Пер.: *Робинсон Дж. А.* Машинно-ориентированная логика, основанная на принципе резолюций // Кибернетический сборник. Новая серия. 1970. Вып. 7. С. 194–218.]
16. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983. 360 с.
17. *Haken A.* The intractability of resolution // Theor. Comp. Sci. 1985. No. 39. P. 297–308. [Пер.: *Хакен А.* Труднорешаемость резолюций // Кибернетический сборник. Новая серия. 1991. Вып. 28. С. 179–194.]
18. *Buss S. R., Turan G.* Resolution proofs of generalized pigeonhole principles // Theoret. Comp. Sci. 1988. No. 62. P. 311–317. [Пер.: *Басс С. Р., Туран Д.* Доказательство обобщенного принципа Дирихле методом резолюций // Кибернетический сборник. Новая серия. 1991. Вып. 28. С. 195–203.]

19. *Lee C. Y.* Representation of Switching Circuits by Binary-Decision Programs // Bell Syst. Techn. J. 1959. No. 38. P. 985–999.
20. *Bryant R. E.* Graph-Based Algorithms for Boolean Function Manipulation // IEEE Trans. Comp. 1986. No. 35(8). P. 677–691.
21. *Meinel Ch., Theobald T.* Algorithms and Data Structures in VLSI-Design: OBDD-Foundations and Applications. Berlin; Heidelberg; New York: Springer Verlag, 1998.
22. *Семёнов А. А., Игнатъев А. С.* Логические уравнения и двоичные диаграммы решений // Прикладные алгоритмы в дискретном анализе. Сер. Дискретный анализ и информатика. Вып. 2. Иркутск: Изд-во Ирк. ун-та, 2008. С. 99–126.
23. *Marques-Silva J. P., Sakallah K A.* GRASP: A search algorithm for propositional satisfiability // IEEE Trans. Comp. 1999. V. 48. No. 5. P. 506–521.
24. *Ben-Sasson E., Wigderson A.* Short proofs are narrow — resolution made simple // Proc. of 31st Ann. ACM Symposium on Theory of Computing. 1999. P. 517–526.
25. *Семёнов А. А.* Консервативные преобразования систем логических уравнений // Вестник Томского государственного университета. Приложение. 2007. № 23. С. 52–59.
26. *Тей А., Грибомон П., Луи Ж. и др.* Логический подход к искусственному интеллекту. М.: Мир, 1991. 429 с.
27. *Горшков С. П.* Применение теории NP-полных задач для оценки сложности решения систем булевых уравнений // Обзорение прикладной и промышленной математики. 1995. Т. 2. Вып. 3. С. 325–398.
28. *Агибалов Г. П.* Дискретные автоматы на полурешетках. Томск: Изд-во Том. ун-та, 1993.
29. *Черемисинова Л. Д., Новиков Д. Я.* Проверка схемной реализации частичных булевых функций // Вестник Томского государственного университета. Управление, вычислительная техника, информатика. 2008. № 4 (5). С. 102–111.
30. *Семёнов А. А.* Трансляция алгоритмов вычисления дискретных функций в выражения пропозициональной логики // Прикладные алгоритмы в дискретном анализе. Сер. Дискретный анализ и информатика. Вып. 2. Иркутск: Изд-во Ирк. ун-та, 2008. С. 70–98.
31. *Катленд Н.* Вычислимость. Введение в теорию рекурсивных функций. М.: Мир, 1983.
32. <http://www.satlive.org>
33. *Заикин О. С., Семёнов А. А.* Технология крупноблочного параллелизма в SAT-задачах // Проблемы управления. 2008. № 1. С. 43–50.
34. *Семёнов А. А., Заикин О. С.* Неполные алгоритмы в крупноблочном параллелизме комбинаторных задач // Вычислительные методы и программирование. 2008. Т. 9. С. 108–118.
35. *Семёнов А. А., Заикин О. С., Беспалов Д. В. и др.* Решение задач обращения дискретных функций на многопроцессорных вычислительных системах // Труды Четвертой Международной конф. РАСО'2008 (Москва, 26–29 октября 2008). М., 2008. С. 152–176.
36. *Valiant L. G.* The complexity of computing the permanent // Theor. Comp. Sci. 1979. V. 8. P. 189–202.
37. *Stockmeyer L.* Classifying of computational complexity of problems // J. Symb. Logic. 1987. V. 52. No. 1. P. 1–43. [Пер.: *Стокмейер Л.* Классификация вычислительной сложности проблем // Кибернетический сборник. Новая серия. 1989. Вып. 26. С. 20–83.]
38. *Valiant L. G.* The complexity of enumeration and reliability problems // SIAM J. Comp. 1979. V. 8. P. 410–421.
39. *Горшков С. П.* О сложности задачи нахождения числа решений систем булевых уравнений // Дискретная математика. 1996. № 8:1. С. 72–85.

РЕКУРСИВНЫЙ СПОСОБ ПОСТРОЕНИЯ СЕМЕЙСТВ БЕЗ ПЕРЕКРЫТИЙ¹

А. В. Черемушкин

Институт криптографии, связи и информатики, г. Москва, Россия

E-mail: avc238@mail.ru

Предлагается модификация рекурсивного способа построения семейств множеств без перекрытий, основанная на использовании ортогональных массивов. Показано, как с их помощью можно построить схемы предварительного распределения ключей на основе пересечений множеств.

Ключевые слова: *семейство множеств без перекрытий, схема предварительного распределения ключей.*

1. Семейства множеств без перекрытий

Если X — множество из v элементов, $|X| = v$, а \mathcal{F} — множество его подмножеств (блоков), $|\mathcal{F}| = b$, то (X, \mathcal{F}) называется (i, j) -семейством без перекрытий (*cover-free family*) и обозначается (i, j) -CFF(v, b), если для любых блоков $B_1, \dots, B_u \in \mathcal{F}$, $u \leq i$, и любых не совпадающих с ними блоков $A_1, \dots, A_w \in \mathcal{F}$, $w \leq j$, выполняется условие

$$\bigcap_{k=1}^u B_k \not\subseteq \bigcup_{s=1}^w A_s.$$

Матрица инцидентности системы множеств (X, \mathcal{F}) — это $(0,1)$ -матрица размера $b \times v$, в которой столбцы соответствуют элементам множества X , а строки — подмножествам из \mathcal{F} , причем единицы стоят на пересечении со столбцами, помеченными элементами подмножества, соответствующего строке.

Непосредственно из определения вытекает следующий критерий.

Лемма 1 [1]. Система множеств (X, \mathcal{F}) является семейством без перекрытий (i, j) -CFF(v, b) в том и только в том случае, когда в ее матрице инцидентности для любых двух непересекающихся наборов, состоящих из i и j строк, найдется столбец, на пересечении которого с первым набором строк стоят единицы, а на пересечении со вторым — нули.

Следствие. Для параметров семейства (i, j) -CFF(v, b) выполняются неравенства $b \geq i + j$ и $v \geq \binom{i+j}{i}$.

Действительно, в матрице инцидентности для любого набора из $i + j$ строк должны найтись столбцы, содержащие всевозможные расположения из i единиц и j нулей.

2. Основная теорема

Ортогональным массивом OA($n, k, 1$) называется $n^2 \times k$ -матрица с элементами из множества $\{1, \dots, n\}$, каждые два столбца которой содержат все различные пары элементов.

¹Работа выполнена при поддержке гранта Президента РФ НШ № 4.2008.10.

Лемма 2. Пусть $i \geq 1, j \geq 1$. В таблице ортогонального массива $OA(n, ij + 1, 1)$ для любых двух непересекающихся наборов, состоящих из i и j строк, найдется столбец, на пересечении с которым ни один из элементов, стоящих в строках из первого набора, не совпадает ни с одним из элементов, стоящих в строках из второго набора.

Доказательство. Рассмотрим два произвольных непересекающихся набора из i и j строк. Рассмотрим подтаблицу, состоящую из первой строки первого набора и всех строк второго набора. В каждом столбце этой подтаблицы выделим элементы первой строки, совпадающие с элементами этого же столбца, принадлежащими строкам второго набора. Из свойства ортогональности следует, что в строках второго набора совпадающие элементы не могут находиться в одинаковых строках, иначе найдутся два столбца с совпадающими парами элементов. Отсюда следует, что максимальное число столбцов с такими совпадениями равно j . Теперь в оставшихся столбцах элементы первой строки из первого набора не совпадают ни с одним из элементов строк второго набора. Удаляем в исходной таблице столбцы, в которых произошло совпадение элементов, и рассматриваем ортогональный массив из оставшихся столбцов. Повторяя рассуждения для оставшихся $i - 1$ строк из первого набора, получаем, что совпадения элементов, стоящих на пересечении со строками из первого набора, с элементами из строк второго набора могут быть не более чем в ij столбцах. Теперь в оставшихся не удаленными столбцах элементы, стоящие в строках из первого набора, не совпадают с элементами из строк второго набора. ■

Теорема 1. Пусть $i \geq 2, j \geq 1$. Если существуют семейство без перекрытий (i, j) -CFF(v, b) и ортогональный массив $OA(b, ij + 1, 1)$, то существует и семейство (i, j) -CFF($((ij + 1)v, b^2)$).

Доказательство. Рассмотрим матрицу инцидентности A семейства (i, j) -CFF(v, b). Построим $(0,1)$ -матрицу B размера $b^2 \times (ij + 1)v$ путем замены каждого элемента $a, 1 \leq a \leq v$, в матрице ортогонального массива $OA(b, ij + 1, 1)$ на строку матрицы A с номером a . В силу леммы 2 для любых двух непересекающихся наборов, состоящих из i и j строк ортогонального массива, найдется столбец, на пересечении с которым ни один из элементов, стоящих в строках из первого набора, не совпадает ни с одним из элементов, стоящих в строках из второго набора. Поэтому после замены всех элементов этого столбца на соответствующие строки матрицы A в построенной матрице B в силу леммы 1 найдется столбец, в котором на пересечении со строками из первого набора стоят единицы, а на пересечении со строками из второго набора — нули. По лемме 1 матрица B является матрицей инцидентности семейства (i, j) -CFF($((ij + 1)v, b^2)$). ■

3. Построение ортогональных массивов на основе разностных матриц

$(n, k; \lambda)$ -Разностной матрицей называется матрица (d_{st}) размера $k \times n\lambda$ над кольцом вычетов \mathbf{Z}_n , в которой при всех $x, y, 1 \leq x < y \leq k$, в мультимножестве

$$\{d_{xz} - d_{yz} \pmod n : 1 \leq z \leq n\lambda\}$$

каждый из элементов \mathbf{Z}_n встречается ровно λ раз.

В работе [1] разностные матрицы использовались для построения семейств без перекрытий. Предложенная там основная конструкция взята из работы [2]. Заметим, что она может быть интерпретирована в терминах ортогональных массивов. Для этого следует воспользоваться следующим способом построения ортогональных массивов

на основе разностных матриц. Если выполнено условие $(n, (k - 1)!) = 1$, то $(n, k; 1)$ -разностную матрицу $D = (d_{xz})$ можно построить, полагая при $1 \leq x \leq n$ и $1 \leq z \leq k$

$$d_{xz} = xz \bmod n.$$

По уже построенной $(n, k; 1)$ -разностной матрице $D = (d_{xz})$ ортогональный массив $OA(n, k, 1)$ с матрицей $B = (b_{(x,y),z})$ размера $n^2 \times k$ строится следующим образом: при $1 \leq x, y \leq n$ и $1 \leq z \leq k$ полагают

$$b_{(x,y),z} = d_{xz} + y \bmod n.$$

Разностные матрицы над произвольными абелевыми группами и способ построения на их основе ортогональных массивов описаны, например, в [3].

4. Рекурсивное построение семейств без перекрытий

В работе [1] предложен рекурсивный способ, позволяющий из семейств без перекрытий (i, j) -CFF(v, b) и $(b^{2^t}, ij + 1, 1)$ -разностных матриц, $t = 0, 1, 2, \dots$, которые существуют при условии $(b, (ij)!) = 1$ [2], строить новые семейства (i, j) -CFF($(ij + 1)^t v_0, b_0^{2^t}$). Вместе с тем условие $(b, (ij)!) = 1$ ограничивает возможность его применения.

Рассмотрим естественную модификацию этого способа, основанную на использовании теоремы 1 и позволяющую строить такие семейства для более широкого класса значений (i, j, v, b) .

Выберем b_0 так, чтобы при каждом $t = 0, 1, 2, \dots$ выполнялось условие существования ортогонального массива $OA(b^{2^t}, ij + 1, 1)$. Теперь, начиная с семейства (i, j) -CFF(v_0, b_0), будем последовательно применять теорему 1. В результате будет построена последовательность семейств

$$\{ (i, j)\text{-CFF}((ij + 1)^t v_0, b_0^{2^t}) : t = 1, 2, \dots \},$$

параметры v, b каждого из которых удовлетворяют условию

$$v = v_0 (\log_{b_0} b)^{\log_2(ij+1)}.$$

Заметим, что результат из работы [1] получается как частный случай теоремы 1. Хотя использование разностных матриц существенно упрощает саму процедуру построения, теорема 1 позволяет расширить по сравнению с [1] множество допустимых значений параметров (i, j, v, b) , для которых можно построить семейство без перекрытий.

В работе [1] для заданных (i, j) выбиралось такое минимальное число $b_0 \geq i + j$, что выполнялось условие $(b_0, (ij)!) = 1$, причем из него автоматически вытекало равенство $(b_0^{2^t}, (ij)!) = 1$ при всех $t = 1, 2, \dots$. При этом число b_0 не могло иметь малых делителей.

В то же время ортогональные массивы существуют и при других значениях параметров (i, j, v, b) . Например, так как при любых $p \geq 2$ и $m \geq 1$ ортогональные массивы $OA(p^m, p^m + 1, 1)$ легко строятся на основе поля из p^m элементов, то можно строить семейства без перекрытий (i, j) -CFF(v, b) при $i + j \leq p^m = b_0$.

В силу леммы 1 минимальным семейством без перекрытий для заданных значений i и j будет семейство (i, j) -CFF(v, b) при $i + j = b$, в матрице инцидентности которого каждый столбец имеет ровно i единиц. Если при данном значении b не найдется соответствующего ортогонального массива, то выбираем значение $b_0 \geq i + j$ так, чтобы при каждом $t = 0, 1, 2, \dots$ выполнялось условие существования ортогонального

массива $OA(b^{2t}, ij + 1, 1)$. Теперь в качестве семейства (i, j) -CFF(v_0, b_0) берем то, у которого $v_0 = \min \left\{ \binom{b}{i}, \binom{b}{j} \right\}$ и в матрице инцидентности каждый столбец имеет ровно i единиц при $i \leq j$ и $n - j$ единиц при $i > j$ соответственно.

В данном случае $v \geq b$. Применяя теорему, можно построить семейства с условием $v < b$. Например, при $b = 4$, последовательно применяя теорему 1, получаем семейства $(2, 2)$ -CFF(6, 4), $(2, 2)$ -CFF(30, 16), $(2, 2)$ -CFF(150, 256), $(2, 2)$ -CFF(750, 65536), $(2, 2)$ -CFF(3750, 4294967296) и т. д.

5. Схемы предварительного распределения ключей

Пусть $g \geq 2$. Под схемой предварительного распределения ключей для групп, состоящих не более чем из g участников, понимают два алгоритма: первый определяет значения распределяемых между n участниками наборов данных, которые будем называть ключевыми материалами, а второй позволяет каждой группе, состоящей не более чем из g участников, вычислить значение ключа для организации закрытого сеанса. При этом должно выполняться условие, гарантирующее, что никакая другая группа участников, не включающая первую в качестве подмножества, объединив свои ключевые материалы, не сможет получить никакой информации о ключе. Более подробно см. обзор [4].

Пусть $w \geq 1$. Схема предварительного распределения ключей для групп участников называется устойчивой к сговору w участников, если любая группа, состоящая не более чем из w участников, объединив свои ключевые материалы, не сможет определить ключи, применяемые группами из оставшихся участников.

Определение [5]. Устойчивая к сговору w участников схема распределения ключей на основе шаблонов для групп из g участников (g, w) -CRKDP(n, k) (*collusion-resistant key distribution patterns*) определяется набором подмножеств $\{S_1, \dots, S_n\}$ множества $\{1, \dots, k\}$, удовлетворяющим условию: если $i_1, \dots, i_g, p_1, \dots, p_w \in \{1, \dots, n\}$ и выполнено включение

$$\bigcap_{j=1}^g S_{i_j} \subseteq \bigcup_{j=1}^w S_{p_j},$$

то $\{i_1, \dots, i_g\} \cap \{p_1, \dots, p_w\} \neq \emptyset$.

Для ее применения надо сформировать множество из k секретных ключей и присвоить им номера $1, \dots, k$. Распределение ключевых материалов осуществляется путем передачи заранее по защищенному каналу каждому абоненту P_i всех ключей с номерами из множества S_i . Теперь для формирования общего ключа каждый участник из группы участников $\{P_{i_1}, \dots, P_{i_g}\}$ выбирает ключи, номера которых лежат в пересечении $S_{i_1} \cap \dots \cap S_{i_g}$, а затем вычисляет общий ключ как значение хеш-функции от строки, составленной из этих ключей.

Так как данное определение по сути совпадает с определением семейства без перекрытий (g, w) -CFF(k, n), то, переформулируя теорему 1, получаем следующий результат для схем распределения ключей на основе шаблонов.

Теорема 2. Пусть $g \geq 2$, $w \geq 1$. Если существует (g, w) -CRKDP(n, k)-схема и ортогональный массив $OA(n, gw + 1, 1)$, то существует и (g, w) -CRKDP($n^2, (gw + 1)k$)-схема.

Используя данный подход в сочетании с [1], можно строить различные схемы предварительного распределения ключей на основе шаблонов. Например, при $g = w = 2$

можно в зависимости от условий применять следующие схемы:

n	256	625	2401	4096	65536
k	150	250	525	700	750

ЛИТЕРАТУРА

1. *Stinson D. R., van Trung T., Wei R.* Secure frameproof codes, key distribution patterns, group testing algorithms and related structures // *J. Statist. Plan. Infer.* 2000. V.86. No.2. P. 595–617.
2. *Atici M., Magliveras M. M., Stinson D. R., Wei W.-D.* Some recursive constructions for perfect hash families // *J. Combinat. Designs.* 1996. V.44. P. 353–363.
3. *Beth T., Jungnickel D., Lenz H.* Design theory. Cambridge Univ. Press, 1989. 688 p.
4. *Черемушкин А. В.* Комбинаторно-геометрические подходы к построению схем предварительного распределения ключей (обзор публикаций) // *Прикладная дискретная математика.* 2008. № 1(1). С. 55–63.
5. *Mitchell C. J., Piper C.* Key storage in Secure Networks // *Discr. Appl. Math.* 1988. V.21. P. 215–228.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

DOI 10.17223/20710410/6/5

УДК 519.7

ДИФФЕРЕНЦИАЛЬНЫЙ КРИПТОАНАЛИЗ
БЛОЧНОГО ШИФРА MARS

А. И. Пестунов

*Институт вычислительных технологий СО РАН, г. Новосибирск, Россия***E-mail:** pestunov@gmail.com

Предлагается дифференциальная атака на шифр MARS, который является финалистом конкурса AES. Эта атака является более эффективной, чем ранее известные, и позволяет провести криптоанализ урезанной версии шифра MARS, использующей 752 бита подключей, в то время как лучшая из ранее известных атак позволяет провести криптоанализ урезанной версии шифра MARS, использующей только 682 бита подключей. Вероятность успеха предлагаемой атаки составляет более 99 %, а ее сложность меньше сложности полного перебора ключей.

Ключевые слова: *блочный шифр, дифференциальный криптоанализ, Advanced Encryption Standard, MARS.*

Введение

Блочные шифры являются важным элементом в современных системах защиты информации, поэтому в последние годы имели место несколько проектов, направленных на их исследование [1–3]. Одним из наиболее влиятельных проектов является Advanced Encryption Standard (AES) [1]. Типичный блочный шифр преобразует открытый текст, представленный в виде последовательности битов (нулей и единиц), по блокам фиксированной длины s , чаще всего s равно 64 или 128 бит. Секретный ключ также представляет собой битовую последовательность (обычно длиной 128 или 256), из которой по определенному для каждого шифра механизму получается набор из R *подключей*. Процедура шифрования носит итеративный характер и заключается в R -кратном выполнении некоторого относительно «слабого» преобразования, которое зависит от своего подключа и называется *раундом* шифрования. Раунды могут быть как одинаковыми, так и различными. Чем больше их выполнено, тем надежней, но медленней становится шифр, поэтому его создатель определяет такое количество раундов, которое обеспечивает и безопасность, и быстродействие алгоритма.

Исследовать безопасность криптоалгоритмов призван раздел криптологии, называемый *криптоанализом*. Криптоанализ блочных шифров подобен соревнованию между криптоаналитиками, которые стараются найти неизвестный секретный ключ для упрощенных версий шифра, по возможности более близких к исходному варианту. Например, для шифра, состоящего из 20 раундов, криптоанализ 12 раундов считается продвижением в анализе шифра, если ранее удавалось найти ключ только для версии шифра, состоящей из 10 раундов. Заметим, что знание всех подключей блочного шифра эквивалентно знанию секретного ключа, поскольку в шифровании участвуют именно подключи, а секретный ключ используется только для их вычисления. Алгоритм нахождения ключа или подключей называется *атакой*. Даже если атака такова,

что она не осуществима на практике, но ее сложность меньше сложности полного перебора ключей, это является важным сертификационным недостатком шифра [4].

На сегодняшний день не существует цельной теории, в рамках которой можно было бы гарантировать безопасность того или иного блочного шифра, поэтому силу шифра можно оценить только на основе предложенных атак на него. Например, 20-раундовый шифр, имеющий атаку на 5 раундов, может быть признан безопасным, а имеющий атаку на 19 раундов вряд ли выдержит атаки криптоаналитиков в ближайшие годы.

Для финалиста конкурса AES шифра MARS [5] подсчитывать количество раундов не совсем корректно, поскольку половина раундов этого 32-раундового шифра — раунды *перемешивания* — не снабжаются подключами и, следовательно, не несут в себе такой криптографической силы, как раунды *ядра*, снабженные двумя подключами. Кроме того, в шифре присутствуют процедуры *пред-отбеливания* (pre-whitening) и *пост-отбеливания* (post-whitening), которые также используют подключи. По этим причинам при сравнении атаки, представленной в этой работе, с ранее известными результатами подсчитывается не количество раундов, а количество раскрытых битов подключей.

В таблице показано, что новая атака позволяет атаковать такой вариант шифра MARS, который использует 752 бита подключей, в то время как лучшая из ранее известных атак — только 682. Отметим, что шифр MARS имеет 256-битовый секретный ключ, и все атаки, приведенные в таблице, действуют быстрее, чем полный перебор ключей такой длины.

Лучшие результаты криптоанализа шифра MARS

Найденные биты подключей	Раунды			Отбеливания		Сложность			Источник
	Всего	Ядра	Перемешивающие	Пред-	Пост-	Блоки	Память, байт	Шифрования	
566	21	5	16	+	+	2^3	2^{236}	2^{232}	[6]
566	21	5	16	+	+	2^{50}	2^{197}	2^{247}	[6]
628	12	6	6	+	+	2^{69}	2^{73}	2^{197}	[6]
682	11	11	0	–	–	2^{65}	2^{69}	2^{229}	[7]
752	16	8	8	+	+	2^{105}	2^{109}	2^{231}	Данная работа

Используемые обозначения:

- $A := B$ — A присвоить B
- $a \oplus b$ — побитовый хог
- $a \boxplus b$ — сложение по модулю 2^{32}
- $a \boxtimes b$ — умножение по модулю 2^{32}
- $a \boxminus b$ — вычитание по модулю 2^{32}
- $a \lll n$ — поворот a влево на n бит
- δ_n — 32-битовое слово, которое имеет 1 на n -й позиции и нули на остальных
- (a_0, a_1, a_2, a_3) — 128-битовый блок, поделенный на четыре 32-битовых слова
- $(a_0, ?, ?, a_3)$ — блок, где некоторые слова неизвестны
- 31-й бит — старший бит в слове

1. Краткое описание дифференциального криптоанализа

Основной объект, который исследуется в дифференциальном криптоанализе, — это пары блоков текста A и B с определенной *разностью* $A \oplus B$ [8]. Если информация о том, как связаны входная разность (между блоками открытого текста) и выходная разность (между блоками шифртекста), отсутствует, то все выходные разности равновероятны. Однако если (как-то) удастся установить, что некоторая входная разность Δ_{inp} приводит к некоторой выходной разности Δ_{out} с вероятностью p большей, чем остальные, то это может быть использовано для отыскания подключей шифра. Пара $(\Delta_{\text{inp}}, \Delta_{\text{out}})$ называется *дифференциалом*, а совокупность дифференциалов на различных раундах

называется *характеристикой*. Если Δ_{out} содержит неизвестные биты, то дифференциал называется *усеченным* [9]. Далее дифференциал обозначается так:

$$\Delta_{\text{inp}} \xrightarrow{p} \Delta_{\text{out}}.$$

Например, для совершенного шифра со 128-битовым блоком при любой входной разности выходная разность примет некоторое фиксированное значение с вероятностью 2^{-128} . Таким образом, если в процессе анализа шифра обнаружится, что определенная входная разность приводит к определенной выходной разности с вероятностью больше чем 2^{-128} (например, 2^{-100}), то эта информация может быть использована с целью отыскания его подключей. Количество текстов, требуемых для реализации атаки, пропорционально $1/p$.

2. Описание шифра MARS

Опишем шифр MARS в объеме, достаточном для понимания статьи (см. полную спецификацию в [5]).

Общая структура шифра MARS. MARS оперирует со 128-битовыми блоками, секретный пользовательский ключ имеет длину до 256 бит. Перед шифрованием этот ключ преобразуется в массив из сорока 32-битовых подключей K_0, \dots, K_{39} . Каждый блок открытого текста A представляется в виде четырех 32-битовых слов (a_0, a_1, a_2, a_3) и подвергается следующим преобразованиям:

- пред-отбеливание: $a_0 := a_0 \boxplus K_0$; $a_1 := a_1 \boxplus K_1$; $a_2 := a_2 \boxplus K_2$; $a_3 := a_3 \boxplus K_3$;
- 8 *прямых* раундов перемешивания;
- 8 прямых раундов ядра;
- 8 *обратных* раундов ядра;
- 8 обратных раундов перемешивания;
- пост-отбеливание: $a_0 := a_0 \boxminus K_{36}$; $a_1 := a_1 \boxminus K_{37}$; $a_2 := a_2 \boxminus K_{38}$; $a_3 := a_3 \boxminus K_{39}$.

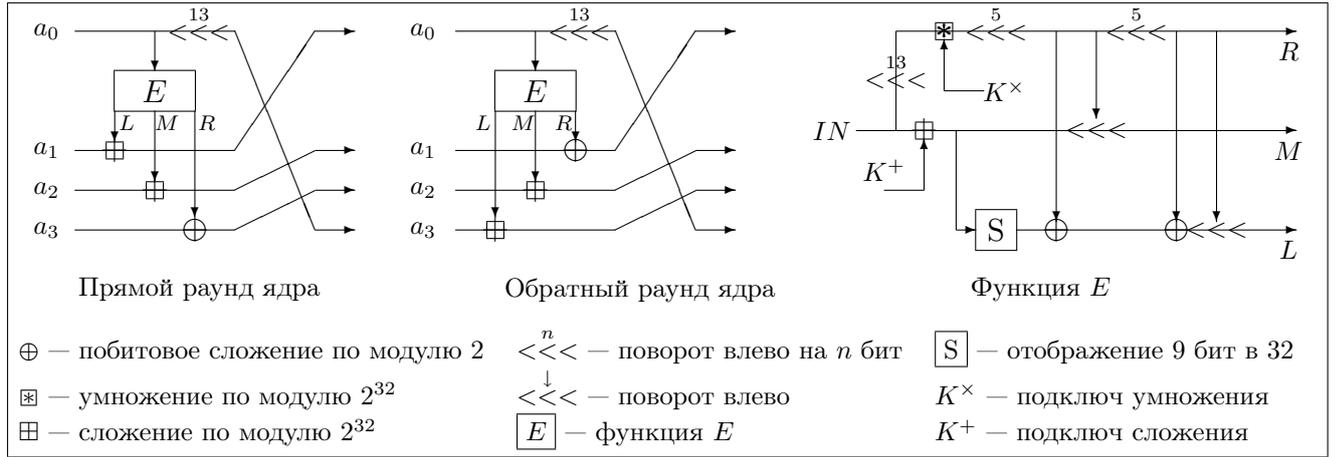
Описание функции E . Основную криптографическую силу шифру MARS придает функция E , которая в качестве аргументов принимает одно 32-битовое слово IN и два подключа: K^+ и K^\times . На выходе эта функция дает три 32-битовых слова: L , M и R (рис. 1). Важно заметить, что подлючи K^\times на всех раундах имеют только 30 неизвестных бит, поскольку согласно алгоритму их получения два младших бита этих подключей всегда равны 1. Слова L , M и R в течение выполнения функции E преобразуются следующим образом:

$$\begin{aligned} R &:= ((IN \lll 13) \boxtimes K^\times) \lll 5; \\ M &:= IN \boxplus K^+; L := M; \\ M &:= M \lll (R \bmod 32); M \text{ — выход}; \\ L &:= S(L \bmod 512) \oplus R; (S \text{ — это S-бокс, отображающий 9 бит в 32 бита}); \\ R &:= R \lll 5; L := (L \oplus R) \lll (R \bmod 32); L \text{ — выход}; R \text{ — выход}. \end{aligned}$$

Раунды ядра шифра MARS. Каждый из 16 раундов ядра (рис. 1) осуществляет следующее преобразование блока: слово a_0 вводится в функцию E (как IN) и затем поворачивается на 13 бит влево. Три выходных слова этой функции (R , M и L) изменяют оставшиеся три слова блока:

$$\begin{aligned} a_1 &:= a_1 \boxplus L; a_2 := a_2 \boxplus M; a_3 := a_3 \oplus R; (\text{прямой раунд ядра}); \\ a_1 &:= a_1 \oplus R; a_2 := a_2 \boxplus M; a_3 := a_3 \boxplus L; (\text{обратный раунд ядра}); \end{aligned}$$

завершается раунд перестановкой слов: $(a_0, a_1, a_2, a_3) := (a_1, a_2, a_3, a_0)$.


 Рис. 1. Раунды ядра и функция E шифра MARS

3. Дифференциальная характеристика для восьми раундов

Опишем построение дифференциальной характеристики

$$\begin{aligned}
 (0, 0, 0, 2^{18}) &\xrightarrow[p=1/2]{pre-whitening + 3 rounds} (2^{18}, 0, 0, 0) \xrightarrow[p=1]{1 round} (2^9, ?, ?, 2^{31}) \xrightarrow[p=1]{1 round} \\
 & \quad (?, ?, ?, 2^{22}) \underset{p=2^{-96}}{=} (0, 0, 0, 2^{22}) \xrightarrow[p=1/2]{3 rounds} (2^{22}, 0, 0, 0),
 \end{aligned} \tag{1}$$

которая покрывает 8 раундов ядра шифра MARS с вероятностью 2^{-98} . Первые три раунда могут быть как прямыми, так и обратными, последние пять — только обратными. Эта характеристика получается путем объединения нескольких дифференциалов.

Дифференциал 1. Построим дифференциал

$$(0, 0, 0, 2^{18}) \xrightarrow[p=1/2]{pre-whitening + 3 rounds} (2^{18}, 0, 0, 0), \tag{2}$$

который покрывает пред-отбеливание и три раунда ядра (прямые или обратные — значения не имеет). Для этого рассмотрим пару блоков A и B с разностью $(0, 0, 0, 2^{18})$ на входе шифра перед пред-отбеливанием, т. е. $a_3 \oplus b_3 = 2^{18}$. В этом случае выходная разность после пред-отбеливания и трех раундов ядра будет $((a_3 \boxplus z) \oplus (b_3 \boxplus z), 0, 0, 0)$, где z — это сумма первого подключа и выходов функции E на трех рассматриваемых раундах. Если $a_3 \oplus b_3 = 2^{18}$, то вероятность того, что $(a_3 \boxplus z) \oplus (b_3 \boxplus z) = 2^{18}$, равна $1/2$ (данный вопрос рассматривался, например, в [10]). Этот дифференциал является начальным в (1).

Дифференциал 2. Аналогичные рассуждения позволяют построить следующий дифференциал:

$$(0, 0, 0, 2^{22}) \xrightarrow[p=1/2]{3 rounds} (2^{22}, 0, 0, 0). \tag{3}$$

Этот дифференциал является заключительным в (1).

Дифференциал 3. Дифференциал

$$(2^{18}, 0, 0, 0) \xrightarrow[p=1]{1 round} (2^9, ?, ?, 2^{31}) \tag{4}$$

начинается с выходной разности $(2^{18}, 0, 0, 0)$ первого дифференциала и покрывает один обратный раунд ядра.

Для его построения заметим, что разность 2^{18} находится в первом слове, которое попадает в функцию E . Эта разность поворачивается на 13 бит влево и становится равной 2^{31} . Далее она проходит через умножение на подключ с вероятностью 1, т. е. после умножения на подключ двух слов с разностью 2^{31} они будут иметь такую же разность с вероятностью 1. За умножением следуют два поворота на 5 бит, т. е. суммарный поворот осуществляется на 10 бит, что приводит к разности $(?, ?, 2^9)$ на выходе функции E .

Выходные разности для двух оставшихся слов M и L могут быть частично установлены, но после следующего раунда уже невозможно будет сказать ничего определенного, их точные значения в дальнейшем не требуются, поэтому считаем их неизвестными. В итоге после рассмотренного одного обратного раунда ядра получим выходную разность $(2^9, ?, ?, 2^{31})$.

Дифференциал 4. Дифференциал

$$(2^9, ?, ?, 2^{31}) \xrightarrow[p=1]{1 \text{ round}} (?, ?, ?, 2^{22}) \quad (5)$$

начинается с выходной разности $(2^9, ?, ?, 2^{31})$ третьего дифференциала и покрывает один обратный раунд ядра. Этот дифференциал получается, если проследить за перестановкой слов в обратном раунде ядра.

Объединение дифференциалов. Чтобы получить характеристику (1), необходимо объединить построенные дифференциалы. Перед этим заметим, что вероятность того, что $(?, ?, ?, 2^{22}) = (0, 0, 0, 2^{22})$, равна 2^{-96} . Данный факт следует из того, что информации о неизвестной 96-битовой разности нет, поэтому можно лишь подразумевать, что она принимает любое значение с вероятностью 2^{-96} (подобные идеи использовались в [11] при построении атаки на AES).

Таким образом, вероятность характеристики (1) равна произведению вероятностей входящих в нее дифференциалов и 2^{-96} , что составляет 2^{-98} .

4. Атака на MARS

Опишем атаку на шифр MARS, состоящий из пред- и пост-отбеливаний, 8 обратных раундов ядра и 8 обратных раундов перемешивания. Таким образом, атакуется версия шифра, состоящая из отбеливаний и 16 раундов (с 17-го по 32-й). Данная версия шифра MARS использует 752 бита подключей. Атака, направленная на их отыскание, основана на дифференциальной характеристике (1).

Атака выполняется в два этапа: сначала — с целью определения подключей пост-отбеливания, затем — с целью определения подключей раундов ядра.

4.1. Атака на подключи пост-отбеливания

Для удобства рассмотрим четыре 32-битовых подключа для пост-отбеливания как один 128-битовый подключ. Первым и самым трудоемким этапом атаки является определение этого подключа. Положим также, что $X = \text{MARS}(A)$ — это шифрование блока A в блок X с помощью шифра MARS, состоящего из пред-отбеливания, восьми раундов ядра, которые покрываются характеристикой, восьми обратных раундов перемешивания и пост-отбеливания.

Описание атаки. Атака работает наподобие шестислойного фильтра. Перебираются все возможные 2^{128} подключей пост-отбеливания и пропускаются через этот

фильтр. Решающее свойство фильтра заключается в том, что правильный подключ успешно проходит все шесть слоев, а ни один из неправильных подключей-кандидатов этого сделать не может. Атака реализуется в несколько шагов:

1. Сформировать 6 групп различных пар блоков A_i^b, B_i^b ($b = 1, \dots, 6; i = 1, \dots, 2^{101}$) с фиксированной разностью $A_i^b \oplus B_i^b = (0, 0, 0, 2^{18})$. Эта разность является входной разностью характеристики (1).
2. Для каждой из пар A_i^b, B_i^b запросить пару шифртекстов $X_i^b = \text{MARS}(A_i^b)$ и $Y_i^b = \text{MARS}(B_i^b)$, полученные $6 \cdot 2^{101}$ пар шифртекстов сохранить в памяти.
3. Перебрать все возможные значения искомого подключа $\text{key} \in \{0, \dots, 2^{128} - 1\}$ и для каждого из них выполнить следующие действия:
 - а) $b := 1$;
 - б) с помощью пост-отбеливания подключом key и 8 обратных раундов перемешивания расшифровать сохраненные в памяти пары шифртекста из группы b и получить пары P_i^b и Q_i^b ;
 - в) если $P_i^b \oplus Q_i^b \neq (0, 0, 0, 2^{22})$ для всех $i = 1, \dots, 2^{101}$, то key — это неправильный подключ-кандидат. Отбросить его и перейти к п. 3, где выбрать следующий подключ-кандидат; если хотя бы одна из пар обеспечивает условие $P_i^b \oplus Q_i^b = (0, 0, 0, 2^{22})$, то перейти к п. г;
 - г) если $b < 6$, то $b := b + 1$ и перейти к п. б; иначе — к п. д;
 - д) $b = 6$, и подключ-кандидат прошел все шесть слоев фильтра. Это правильный подключ.

Вероятность успеха атаки. Вычислим вероятность того, что все неправильные подлючи-кандидаты отброшены, а правильный 128-битовый подключ остался.

Поскольку пара X_i^b, Y_i^b , частично расшифрованная с неправильным подключом, не подчиняется дифференциалу

$$(0, 0, 0, 2^{18}) \xrightarrow[p=2^{-98}]{\text{pre-whitening} + 8 \text{ rounds}} (2^{22}, 0, 0, 0),$$

то разность $P_i^b \oplus Q_i^b$ принимает любое значение (включая и $(2^{22}, 0, 0, 0)$) равновероятно, т. е. с вероятностью 2^{-128} . Вероятность получить такую разность среди 2^{101} пар приблизительно равна 2^{-27} , а вероятность получить такую разность сразу в шести группах — 2^{-162} . Значит, вероятность того, что хотя бы один неправильный подключ-кандидат из $2^{128} - 1$ пройдет шесть слоев фильтра, составляет примерно 2^{-34} .

Пара X_i^b, Y_i^b , частично расшифрованная с правильным подключом, подчиняется указанному дифференциалу, следовательно, $P_i^b \oplus Q_i^b = (0, 0, 0, 2^{22})$ с вероятностью 2^{-98} , и вероятность получить такую разность хотя бы раз среди 2^{101} пар равна примерно 0,999665 (см. распределение Пуассона [12]). Значит, вероятность получить такую разность во всех шести группах сразу приблизительно равна 0,99799.

Сложность атаки на подлючи пост-отбеливания. Для реализации описанной части атаки необходимо примерно 2^{233} частичных 8-раундовых операций расшифрования, что эквивалентно 2^{231} операциям шифрования полным шифром MARS. Также требуются 2^{105} выбранных открытых текстов и 2^{109} байт памяти, чтобы их хранить.

4.2. Атака на подлючи раундов ядра

После того как 128-битовый подключ пост-отбеливания установлен, все хранящиеся в памяти шифртексты необходимо расшифровать на пост-отбеливании и на 8 не

снабженных подключами обратных раундах перемешивания. После этого шифртексты оказываются зашифрованными с помощью шифра MARS, состоящего из пред-отбеливания и восьми раундов ядра. Применяя дифференциалы, образующие характеристику (1), нужно реализовать аналогичную атаку, перебирая подключи раундов ядра один за другим. Для определения подключа последнего раунда необходимо использовать дифференциал

$$(0, 0, 0, 2^{18}) \xrightarrow[p=2^{-98}]{\text{pre-whitening} + 7 \text{ rounds}} (0, 2^{22}, 0, 0),$$

для определения подключа предпоследнего раунда необходимо использовать дифференциал

$$(0, 0, 0, 2^{18}) \xrightarrow[p=2^{-98}]{\text{pre-whitening} + 6 \text{ rounds}} (0, 0, 2^{22}, 0),$$

затем —

$$(0, 0, 0, 2^{18}) \xrightarrow[p=1/2]{\text{pre-whitening} + 5 \text{ rounds}} (?, ?, ?, 2^{22}),$$

$$(0, 0, 0, 2^{18}) \xrightarrow[p=1/2]{\text{pre-whitening} + 4 \text{ rounds}} (2^9, ?, ?, 2^{31}),$$

$$(0, 0, 0, 2^{18}) \xrightarrow[p=1/2]{\text{pre-whitening} + 3 \text{ rounds}} (2^{18}, 0, 0, 0),$$

$$(0, 0, 0, 2^{18}) \xrightarrow[p=1/2]{\text{pre-whitening} + 2 \text{ rounds}} (0, 2^{18}, 0, 0),$$

$$(0, 0, 0, 2^{18}) \xrightarrow[p=1/2]{\text{pre-whitening} + 1 \text{ round}} (0, 0, 2^{18}, 0),$$

$$(0, 0, 0, 2^{18}) \xrightarrow[p=1/2]{\text{pre-whitening}} (0, 0, 0, 2^{18}).$$

После определения всех раундовых подключей подключи для пред-отбеливания вычисляются путем решения системы линейных уравнений.

Сложность атаки на подключи раундов ядра и сложность всей атаки.

В каждый раунд ядра вводятся два 32-битовых подключа, в одном из которых два младших бита всегда равны 1, следовательно, перебирать нужно 62-битовые подключи. Это существенно меньше, чем перебор 128-битовых подключей пост-отбеливания, поэтому сложность поиска раундовых подключей ничтожно мала по сравнению с поиском подключа пост-отбеливания, и суммарная сложность атаки определяется сложностью отыскания подключей пост-отбеливания.

Количество пар текстов, необходимых для отыскания подключей раундов ядра, также не превзойдет того количества, которое необходимо для атаки на ключ пост-отбеливания. Данный факт следует из того, что вероятности дифференциалов, используемых в атаке на подключи раундов ядра, меньше либо равны вероятности дифференциала первой части атаки.

Заключение

В данной работе описана атака на шифр MARS, которая является более эффективной, чем ранее известные. Эта атака основана на 8-раундовой дифференциальной характеристике.

ЛИТЕРАТУРА

1. <http://csrc.nist.gov/encryption/aes> — Advanced Encryption Standard (AES) project 1997–2000.
2. <https://www.cosic.esat.kuleuven.be/nessie> — New European Schemes for Signatures, Integrity, and Encryption Deliverables of the NESSIE project 2003.
3. <http://www.cryptrec.go.jp/english/> — CRYPTREC project 2000–2002.
4. *Schneier B.* A self-study course in block-cipher cryptanalysis // *Cryptologia*. 2000. V.24. No.1. P.18–34.
5. *Burwick C. et al.* MARS — a candidate cipher for AES // AES submission. 1999. <http://www.research.ibm.com/security/mars.pdf>.
6. *Kelsey J., Schneier B.* MARS attacks! Preliminary cryptanalysis of reduced-round MARS variants // Proc. of the Third AES Candidate Conf. 2000. <http://www.schneier.com/paper-mars-attacks.pdf>.
7. *Kelsey J., Kohno T., Schneier B.* Amplified boomerang attacks againts reduced-round MARS and Serpent // LNCS. 2001. V.1978. P.75–93.
8. *Biham E., Shamir A.* Differential cryptanalysis of DES-like cryptosystems // *J. Cryptol.* 1991. V.4. P.3–72.
9. *Knudsen L.* Truncated and higher order differentials // LNCS. 1995. V.1008. P.196–211.
10. *Biryukov A., Kushilevitz E.* Improved cryptanalysis of RC5 // LNCS. 1998. V.1403. P.85–99.
11. *Biryukov A.* The boomerang attack on 5 and 6-Round reduced AES // LNCS. 2005. V.3373. P.11–15.
12. *Боровков А. А.* Теория вероятностей. М.: Наука, 1976. 352 с.

**ПРОТОКОЛ АРГУМЕНТА ЗНАНИЯ СЛОВА КОДА ГОППЫ
И ОШИБКИ ОГРАНИЧЕННОГО ВЕСА¹**

В. Е. Федюкович

*Интропро, г. Киев, Украина***E-mail:** vf@unity.net

Предложен новый протокол, позволяющий показать знание кодового слова кода Гоппы и полинома Гоппы, таких, что хэмминг-вес ошибки в искажённом кодовом слове не превышает заданный порог. Протокол является аргументом в предположении о сложности задачи поиска логарифма в используемой конечной группе и имеет специальное свойство нулевого разглашения в модели с честным Проверяющим.

Ключевые слова: *интерактивная система доказательства, аргумент, нулевое разглашение, схема привязки, код Гоппы.*

Введение

Рассматривается задача проверки утверждения об ошибке в искажённом кодовом слове кода Гоппы. Дополнительным условием является предоставление проверяющей стороне минимально необходимой информации о структуре кода, кодовом слове и весе ошибки. Рассматриваются интерактивные протоколы аргумента и протоколы аргумента знания значений, из которых получены экземпляры привязки. Выполняется вероятностная проверка утверждения о кодовом слове и весе ошибки. Произвольный Доказывающий, который не знает компонентов кодового слова и коэффициентов полинома Гоппы, удовлетворяющих проверяемым условиям, имеет только ничтожную вероятность успешно завершить протокол с честным Проверяющим. Рассматривается схема запрос-ответ, в которой запросом является значение переменной некоторого полинома над конечным полем, а проверяемым условием — равенство нулю этого полинома. Также рассматриваются полиномы, в которых компоненты кодового слова и коэффициенты полинома Гоппы заменяются ответами Проверяющего.

Полученный протокол имеет свойство полноты, свойство корректности в предположении о сложности задачи поиска логарифма в конечной группе, является аргументом знания кодового слова и полинома Гоппы, имеет специальное свойство нулевого разглашения в модели с честным Проверяющим. Ошибка корректности протокола экспоненциально мала, параметром безопасности является количество бит в двоичном представлении порядка группы.

Тезисное изложение результатов этой работы приведено в [1].

1. Обозначения и определения**1.1. Интерактивные системы и их свойства**

Участника протокола называют *честным* (honest), если он следует спецификации протокола, либо *произвольным* (any) иначе. Функцию называют *ничтожной*, если она убывает быстрее любой степени своего аргумента. Будем говорить, что событие

¹Результаты работы докладывались на Международной конференции с элементами научной школы для молодёжи, г. Омск, 7–12 сентября 2009 г.

происходит *почти всегда* (overwhelming probability), если вероятность того, что событие не произойдёт, ничтожна. Будем называть *преимуществом* разность вероятностей успешного завершения некоторого алгоритма распознавания, обычно выполняемого соперником, в случае предоставления сопернику возможности выбора входных данных и для случайно выбранных входных данных. *Алфавитом* называют конечное множество символов (букв алфавита); *языком* \mathbb{L} называют некоторое подмножество множества всех слов в данном алфавите. Задачу принятия решения $x \in \mathbb{L}$ называют *распознаванием языка*. Задачу распознавания языка относят к классу сложности NP , если для любого слова x существует NP -свидетельство (witness) w , знание которого позволяет решить задачу за полиномиальное время; в этом случае язык называется NP -языком.

Определение 1. *Интерактивной системой аргумента* (interactive argument system) для NP -языка \mathbb{L} и для некоторого предположения (assumption) будем называть интерактивную пару машин Тьюринга, работающих полиномиальное время, со словом x на общей входной ленте и свидетельством w на входной ленте Доказывающего, такую, что машина Проверяющего всегда выдает бинарный ответ (решение) и останавливается, а также имеются свойства полноты и корректности. *Полнота* (completeness): честный Проверяющий всегда принимает положительное решение (ассерт) для честного Доказывающего и $x \in \mathbb{L}$ (нулевая *ошибка полноты*). *Корректность* (soundness): честный Проверяющий принимает положительное решение для произвольного Доказывающего и $x \notin \mathbb{L}$ с ничтожной вероятностью при условии, что Доказывающий не может нарушить используемое предположение (ничтожная *ошибка корректности*).

Заметим, что определение интерактивной системы доказательства в [2] предусматривает неограниченные ресурсы машины Доказывающего, ненулевую ошибку полноты и ошибку корректности, превышающую ошибку полноты на не-ничтожную функцию; параметром является $|x|$. В [3] рассматривались протоколы, свойство корректности которых выполняется при дополнительных предположениях о вычислительной сложности некоторых задач (computationally-sound proofs); для них был предложен термин «*протокол аргумента*». Протокол аргумента обычно имеет параметр, определяющий степень сложности задачи, содержащейся в используемом предположении.

Определение 2. Будем говорить, что протокол аргумента имеет свойство *знания* (is of knowledge), если существует алгоритм *extractor*, предполагающий управление машиной Доказывающего с возможностью возврата (rewind), работающий полиномиальное время, получающий на входе стенограмму протокола и почти всегда выдающий свидетельство Доказывающего при условии выполнимости предположения (ничтожная *ошибка знания*).

Введённое так понятие протокола аргумента со свойством знания отличается от понятия протокола доказательства знания [4] тем, что последний предусматривает неограниченные ресурсы Доказывающего и не-ничтожную ошибку корректности. Рассматривались [5] также протоколы с алгоритмом *extractor*, которые либо дают свидетельство, либо нарушают используемое предположение.

Определение 3 [2, 6, 7]. Говорят, что протокол имеет свойство *нулевого разглашения* (is zero knowledge), если существует *моделирующий алгоритм* (simulator), работающий полиномиальное время и в случае $x \in \mathbb{L}$ выдающий *моделируемую стенограмму* (simulated transcript), неотличимую от стенограммы протокола с Доказывающим. Протокол имеет свойство *нулевого разглашения в модели с честным Проверяющим* (is honest Verifier zero knowledge), если моделируемая стенограмма неотличима от всех

стенограмм протокола, в которых Проверяющий следует спецификации протокола. Протокол имеет свойство *специального нулевого разглашения* (is special honest Verifier zero knowledge), если моделирующий алгоритм получает на входе запросы Проверяющего и дает моделируемую стенограмму, неотличимую от всех стенограмм протокола, в которых запросы Проверяющего совпадают с запросами на входе моделирующего алгоритма.

Близкими к понятию нулевого разглашения, но не совпадающими с ним, являются следующие встречающиеся в литературе определения. Протоколы, дающие только ничтожное преимущество любому алгоритму Соперника (Adversary) найти свидетельство Доказывающего, называют *witness hiding*. Протоколы, дающие только ничтожное преимущество произвольному алгоритму Соперника, определяющему, какое из двух возможных значений является свидетельством, называют *witness indistinguishable*.

Способ Фиата — Шамира преобразования протокола в схему подписи [8] предусматривает произвольный алгоритм Проверяющего, выбирающий значение хэш-функции входного слова и имеющейся части стенограммы в качестве запроса, и в остальном следующий спецификации протокола; экземпляром подписи при этом является стенограмма протокола. Известен русскоязычный обзор протоколов и некоторых их свойств [9].

1.2. С х е м а п р и в я з к и

Определение 4. *Схемой привязки* (commitment scheme) называют тройку алгоритмов ($Generate()$, $Commit(w)$, $Open(C, w)$), такую, что экземпляр привязки C всегда успешно *раскрывается* значением w , из которого он был *создан*, а также имеются свойства связывания и скрытия. *Связывание* (binding): задача поиска альтернативного значения w' , такого, что $Commit(w') = Commit(w)$, является сложной. *Скрытие* (hiding): экземпляр привязки дает только ничтожное преимущество любому алгоритму Соперника, определяющему, какое из двух значений использовалось для создания экземпляра привязки.

Определение 5. Схема привязки имеет свойство *гомоморфизма* (homomorphic) относительно некоторых операций (\oplus, \otimes) , если

$$\forall(x_1, x_2) \quad Commit(x_1 \oplus x_2) = Commit(x_1) \otimes Commit(x_2).$$

Например, алгоритм инициализации схемы привязки Педерсена [10] к элементу x конечного поля \mathbb{Z}_q предусматривает выбор циклической конечной группы \mathbb{G} порядка q , а также двух элементов группы h, f так, что $\log_f(h)$ неизвестен и задача поиска логарифма в группе является сложной. Алгоритм создания экземпляра привязки выбирает случайное значение $r \in_R \mathbb{Z}_q$, создаёт элемент группы $C = h^x f^r$. Алгоритм раскрывания проверяет $C \stackrel{?}{=} h^x f^r$. Схема Педерсена имеет свойство аддитивного гомоморфизма и является протоколом доказательства знания значений, из которых был получен экземпляр привязки.

1.3. П р о т о к о л Ш н о р р а

Ряд протоколов предусматривают бинарные запросы Проверяющего, так что вероятность для произвольного Доказывающего случайно предложить правильный ответ равна $1/2$. Известны также протоколы с запросами, выбранными из множества S с большим количеством элементов, так что вероятность случайно предложить правильный ответ равна $1/|S|$. В протоколе Шнора [11] запросы выбираются из конечного поля \mathbb{Z}_q , так что ошибка корректности ничтожна (параметром является длина битового представления q). Общей информацией сторон является генератор g циклической конечной группы \mathbb{G} порядка q для некоторого большого простого q , а также

некоторый элемент $y \in \mathbb{G}$. Неформально, y играет роль публичного ключа проверки. Информацией Доказывающего является число b , такое, что $y = g^b$.

- 1) Доказывающий выбирает некоторое случайное $\alpha \in \mathbb{Z}_q$, получает элемент группы $A_0 = g^\alpha$ и пересылает A_0 Проверяющему.
- 2) Проверяющий выбирает некоторый запрос (challenge) $c \in \mathbb{Z}_q$ случайным образом и пересылает его Доказывающему.
- 3) Доказывающий вычисляет и пересылает Проверяющему ответ

$$B = cb + \alpha \pmod{q}. \quad (1)$$

- 4) Проверяющий принимает положительное решение, если

$$g^B y^{-c} A_0^{-1} = 1. \quad (2)$$

Протокол доказательства знания логарифма в конечной группе с бинарными запросами и ответами Доказывающего, полученными как значения линейного полинома, был предложен в работе Chaum, Evertse и van de Graaf [12].

1.4. Код Гоппы

Определение 6. Кодовым словом [13, 14] будем называть вектор $b = b_1 \dots b_N$ элементов поля, таких, что для некоторого полинома $g(z)$ и для дополнительных значений $a_j, j = 1, \dots, N$, выполняются условия $g(a_j) \neq 0$ и

$$\sum_{j=1}^N \frac{b_j}{z-a_j} \equiv 0 \pmod{g(z)}. \quad (3)$$

Множество всех таких кодовых слов называют *кодом Гоппы*, а $g(z)$ — *полиномом Гоппы*.

Двоичный код такого вида был предложен в работе [13], в [14] рассматривается код над полем, состоящим из $q = p^l$ элементов для некоторого простого p . В этой работе используется стандартная схемы привязки, в связи с чем рассматривается только случай $l = 1$.

Определение 7. Ошибкой в искажённом входном слове $w = w_1 \dots w_N$ будем называть вектор $w - b$, где b — кодовое слово; *весом ошибки* — количество ненулевых компонент в ней.

Пусть $g(z) = \sum_{k=0}^T z^k g_k$. В этой работе проверяется справедливость утверждения, эквивалентного (3):

$$W^*(z) \equiv 0, \quad W^*(z) = \sum_{j=1}^N b_j \sum_{k=1}^T g_k \frac{z^k - a_j^k}{z - a_j} \prod_{\substack{i=1, m=0 \\ i \neq j}}^N \sum g_m a_i^m. \quad (4)$$

Значения $b_j, j = 1, \dots, N$, и $g_k, k = 1, \dots, T$, (свидетельство Доказывающего) доступны Проверяющему только в виде экземпляров привязки и ответов (1) Доказывающего.

Определение 8. Проверочным полиномом кодового слова для некоторого значения аргумента $z = d$ и для некоторых начальных случайных значений β_1 , для $j = 1, \dots, N$ и α_k для $k = 1, \dots, T$ протокола Шнора будем называть

$$R^*(y) = \sum_{j=1}^N B_j(y) \sum_{k=1}^T G_k(y) \frac{d^k - a_j^k}{d - a_j} \prod_{\substack{i=1, m=0 \\ i \neq j}}^N \sum G_m(y) a_i^m,$$

$$B_j(y) = y b_j + \beta_j, \quad G_k(y) = y g_k + \alpha_k.$$

Определение 9. Проверочным полиномом ошибки для некоторых начальных случайных значений β_j , $j = 1, \dots, N$, протокола Шнора будем называть

$$E^*(y) = \prod_{j=1}^N (B_j(y) - yw_j), \quad B_j(y) = yb_j + \beta_j.$$

Нетрудно убедиться, что $\deg(R^*(y)) = N + 1$, старший коэффициент $R^*(y)$ равен $W^*(d)$, а степень $E^*(y)$ равна весу ошибки.

Алгебраическое декодирование искажённого кодового слова в этой работе не рассматривается.

1.5. Вероятностная проверка утверждений о полиномах

Пусть $f(z)$ — некоторый ненулевой полином с коэффициентами из конечного поля порядка q . Количество корней такого полинома не превышает $\deg(f(z))$. Вероятность выбрать корень случайно путём выбора произвольного значения аргумента из поля при условии отсутствия информации о коэффициентах не превышает $\deg(f(z))/q$.

2. Протокол для слова кода Гоппы и ошибки ограниченного веса

Протокол P

Общая информация Проверяющего и Доказывающего: порядок q и пара элементов группы (h, f) , дополнительные значения a_j , $j = 1, \dots, N$, искажённое кодовое слово $w_1 \dots w_N$, экземпляры привязки к коэффициентам полинома V_k , $k = 0, \dots, T$, и к компонентам кодового слова W_j , $j = 1, \dots, N$, верхняя граница веса ошибки S .

Информация Доказывающего: коэффициенты полинома g_k и дополнительные значения θ_k , компоненты кодового слова b_j и дополнительные значения φ_j , такие, что $V_k = h^{g_k} f^{\theta_k}$, $W_j = h^{b_j} f^{\varphi_j}$ и $g(z) = \sum_{k=0}^T z^k g_k$, $\sum_{j=1}^N \frac{b_j}{z^{-a_j}} \equiv 0 \pmod{g(z)}$, $|\{j \mid b_j \neq w_j\}| \leq S$.

Доказывающий демонстрирует знание значений b_j и g_k , из которых были получены экземпляры привязки W_j и V_k , а также справедливость утверждения о весе ошибки.

- 1) Проверяющий выбирает запрос $d \in \mathbb{Z}_q$ и пересылает его Доказывающему.
- 2) Доказывающий выбирает в \mathbb{Z}_q значения α_k, ζ_k , $k = 0, \dots, T$; β_j, η_j , $j = 1, \dots, N$; μ_t , $t = 0, \dots, N$; τ_s , $s = 0, \dots, S$; получает коэффициенты r_t, p_s , экземпляры привязки U_k, Q_j, R_t, P_s :

$$\sum_{j=1}^N (yb_j + \beta_j) \sum_{k=1}^T (yg_k + \alpha_k) \frac{d^k - a_j^k}{d - a_j} \prod_{\substack{i=1, m=0 \\ i \neq j}}^N \sum_{m=0}^T (yg_m + \alpha_m) a_i^m = \sum_{t=0}^N y^t r_t,$$

$$\prod_{j=1}^N (yb_j + \beta_j - yw_j) = \sum_{s=0}^S y^s p_s,$$

$$U_k = h^{\alpha_k} f^{\zeta_k}, \quad Q_j = h^{\beta_j} f^{\eta_j}, \quad R_t = h^{r_t} h^{\mu_t}, \quad P_s = h^{p_s} h^{\tau_s}.$$

Доказывающий пересылает U_k, Q_j, R_t, P_s Проверяющему.

- 3) Проверяющий выбирает запрос $c \in \mathbb{Z}_q$ и пересылает его Доказывающему.
- 4) Доказывающий получает ответы и пересылает их Проверяющему:

$$\begin{aligned} \Psi_k &= cg_k + \alpha_k, & \Theta_k &= c\theta_k + \zeta_k, & \Omega_j &= cb_j + \beta_j, & \Phi_j &= c\varphi_j + \eta_j, \\ \Delta &= \sum_{t=0}^N c^t \mu_t, & \Delta' &= \sum_{s=0}^S c^s \tau_s. \end{aligned}$$

5) Проверяющий получает значения проверочного полинома:

$$\Gamma = \sum_{j=1}^N \Omega_j \sum_{k=1}^T \Psi_k \frac{d^k - a_j^k}{d - a_j} \prod_{\substack{i=1, m=0 \\ i \neq j}}^N \Psi_m a_i^m, \quad \Gamma' = \prod_{j=1}^N (\Omega_j - c w_j).$$

Проверяющий рассматривает демонстрацию как убедительную, если

$$h^{\Psi_k} f^{\Theta_k} V_k^{-c} = U_k \wedge h^{\Omega_j} h^{\Phi_j} W_j^{-c} = Q_j \wedge h^{\Gamma} f^{\Delta} \prod_{t=0}^N R_t^{-c^t} = 1 \wedge h^{\Gamma'} f^{\Delta'} \prod_{s=0}^S P_s^{-c^s} = 1.$$

Нетрудно проверить, что протокол P обладает свойством полноты: честный Доказывающий всегда принимает положительное решение, если вес ошибки в искажённом кодовом слове не превышает порог.

Лемма 1. Любой Доказывающий, получающий приемлемые ответы, не совпадающие со значениями линейных полиномов, нарушает предположение о вычислительной сложности задачи поиска логарифма.

Доказательство. Пусть имеется Доказывающий, способный с более чем ничтожной вероятностью находить приемлемые ответы Ψ_k, Θ_k на запрос Проверяющего c , отличные от значений линейных полиномов $cg_k + \alpha_k, c\theta_k + \zeta_k$. Для любой пары элементов циклической группы (h, f) , $f \neq 1$, найдется некоторое значение $\chi \in \mathbb{Z}_q$, такое, что $h = f^\chi$. Тогда $V_k = f^{\chi g_k + \theta_k}$, $U_k = f^{\chi \alpha_k + \zeta_k}$, проверяемое условие $\chi \Psi_k + \Theta_k - c(\chi g_k + \theta_k) - (\chi \alpha_k + \zeta_k) = 0$. Предположим, что $\Psi_k - cg_k - \alpha_k \neq 0$. Тогда $\log_f(h) = -(\Theta_k - c\theta_k - \zeta_k) / (\Psi_k - cg_k - \alpha_k)$. Аналогично для ответов (Ω_j, Φ_j) . ■

Теорема 1. Протокол P является аргументом со свойством знания в предположении сложности задачи поиска дискретного логарифма.

Доказательство. Алгоритм *extractor* выполняет возврат Доказывающего после получения ответов $\Psi_k, \Theta_k, \Omega_j, \Phi_j, \Delta, \Delta'$ на запрос c к состоянию ожидания запроса Проверяющего и получает от Доказывающего повторные ответы $\bar{\Psi}_k, \bar{\Theta}_k, \bar{\Omega}_j, \bar{\Phi}_j, \bar{\Delta}, \bar{\Delta}'$ на повторный запрос $\bar{c} \neq c$ без изменения начальных случайных значений α_k, β_j . *Extractor* вычисляет свидетельство из любой такой пары приемлемых ответов: $g_k = (\Psi_k - \bar{\Psi}_k) / (c - \bar{c})$, $b_j = (\Omega_j - \bar{\Omega}_j) / (c - \bar{c})$. ■

Теорема 2. Протокол P является аргументом в предположении сложности задачи поиска дискретного логарифма.

Доказательство. Рассмотрим произвольный алгоритм Доказывающего с искажённым кодовым словом, содержащим ошибку веса, превышающего S . Тогда полином $\hat{E}(y) = E^*(y) - \sum_{s=0}^S y^s p_s$ ненулевой для любых p_s . Вероятность случайного выбора корня c полинома $\hat{E}(y)$ не превышает N/q . Пусть $\hat{E}(c) \neq 0$. Тогда такой Доказывающий также получает нетривиальный логарифм $\log_f(h) = (-\Delta' + \sum_{s=0}^S c^s \tau_s) / \hat{E}(c)$. Таким образом, вероятность для честного Проверяющего ошибочно принять положительное решение на этапе проверки веса ошибки не превышает N/q .

Предположим, Доказывающий создал экземпляры привязки к некоторым значениям b_j и g_k , которые не являются кодовым словом. Тогда полином $W^*(z)$ (4) ненулевой, $\deg(W^*(z)) = T - 1$, вероятность выбрать корень d полинома $W^*(z)$ не превышает $(T - 1)/q$. Пусть $W^*(d) \neq 0$. Тогда старший коэффициент полинома

$\hat{R}(y) = R^*(y) - \sum_{t=0}^N y^t r_t$ ненулевой для любых r_t , вероятность выбрать корень c полинома $\hat{R}(y)$ не превышает $(N+1)/q$. Пусть $\hat{R}(c) \neq 0$. Тогда такой Доказывающий также получает нетривиальный логарифм $\log_f(h) = (-\Delta + \sum_{t=0}^N c^s \mu_t) / \hat{R}(c)$. Таким образом, вероятность для честного Проверяющего ошибочно принять положительное решение на этапе проверки кодового слова не превышает $(N+T)/q$.

Таким образом, вероятность ошибочно принять положительное решение в рамках протокола для честного Проверяющего ничтожна, параметром является длина битового представления порядка группы q . ■

Теорема 3. Протокол P имеет свойство специального нулевого разглашения.

Доказательство. Рассмотрим следующий моделирующий алгоритм.

Входные параметры: запросы $(d, c) \in \mathbb{Z}_q$ Проверяющего.

- 1) Проверяющий выбирает в \mathbb{Z}_q случайные элементы $\Psi_k, \Theta_k, k = 0, \dots, T; \Omega_j, \Phi_j, j = 1, \dots, N; \Delta, \Delta'$.
- 2) Проверяющий выбирает в группе \mathbb{G} случайные элементы $U_k, k = 0, \dots, T; Q_j, j = 1, \dots, N; R_t, t = 1, \dots, N; P_s, s = 1, \dots, S$.
- 3) Проверяющий получает

$$\Gamma = \sum_{j=1}^N \Omega_j \sum_{k=1}^T \Psi_k \frac{d^k - a_j^k}{d - a_j} \prod_{\substack{i=1, m=0 \\ i \neq j}}^N \sum \Psi_m a_i^m, \quad \Gamma' = \prod_{j=1}^N (\Omega_j - c w_j),$$

$$U_k = h^{\Psi_k} f^{\Theta_k} V_k^{-c}, \quad Q_j = h^{\Omega_j} h^{\Phi_j} W_j^{-c}, \quad R_0 = h^{\Gamma} f^{\Delta} \prod_{t=1}^N R_t^{-c^t}, \quad P_0 = h^{\Gamma'} f^{\Delta'} \prod_{s=1}^S P_s^{-c^s}.$$

Моделируемой стенограммой является строка из элементов $d, U_k, Q_j, R_t, P_s, c, \Psi_k, \Theta_k, \Omega_j, \Phi_j, \Delta, \Delta'$.

Компоненты стенограммы имеют равномерное распределение; зависимости между компонентами моделируемой стенограммы, использованные на шаге 3 моделирующего алгоритма для расчёта U_k, Q_j, R_0, P_0 , совпадают с зависимостями между компонентами стенограммы протокола с Доказывающим. ■

Заключение

Предложен протокол для проверки утверждения о весе ошибки в искажённом кодовом слове и схема запрос-ответ для проверки справедливости утверждений о полиномах. Предложенная схема запрос-ответ также использовалась для протоколов проверки условий о множествах [15], графах [16, 17] и строках [18].

ЛИТЕРАТУРА

1. Федюкович В. Е. Протокол аргумента знания слова кода Гошпы и ошибки ограниченного веса // Прикладная дискретная математика. Приложение. 2009. №1. С. 30–32.
2. Goldreich O., Micali S., Wigderson A. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems // J. ACM. 1991. V. 38. No. 3. P. 691–729.
3. Brassard G., Chaum D., Crépeau C. Minimum disclosure proofs of knowledge // J. Comput. Syst. Sci. 1988. V. 37. No. 2. P. 156–189.
4. Bellare M., Goldreich O. On defining proofs of knowledge // CRYPTO. 1992. P. 390–420.
5. Computationally convincing proofs of knowledge / G. Brassard, C. Crépeau, S. Laplante, C. Léger // STACS. 1991. P. 251–262.

6. *Bellare M., Micali S., Ostrovsky R.* The (true) complexity of statistical zero knowledge // STOC. 1990. P. 494–502.
7. *Cramer R., Damgård I., Schoenmakers B.* Proofs of partial knowledge and simplified design of witness hiding protocols // CRYPTO. 1994. P. 174–187.
8. *Fiat A., Shamir A.* How to prove yourself: Practical solutions to identification and signature problems // CRYPTO. 1986. P. 186–194.
9. *Варновский Н. П.* Типы нулевого разглашения // XI Междунар. школа-семинар «Синтез и сложность управляющих систем». Нижний Новгород, 2000.
10. *Pedersen T. P.* Non-interactive and information-theoretic secure verifiable secret sharing // CRYPTO. 1991. P. 129–140.
11. *Schnorr C. P.* Efficient identification and signatures for smart cards // CRYPTO. 1989. P. 239–252.
12. *Chaum D., Evertse J. H., van de Graaf J.* An improved protocol for demonstrating possession of discrete logarithms and some generalizations // EUROCRYPT. 1987. P. 127–141.
13. *Гонна В. Д.* Новый класс линейных корректирующих кодов // Проблемы передачи информации. 1970. Т. 6. № 3. С. 24–30. <http://mi.mathnet.ru/eng/ppi1748>.
14. *Гонна В. Д.* Рациональное представление кодов и (L, g) -коды // Проблемы передачи информации. 1971. Т. 7. № 3. С. 41–49. <http://mi.mathnet.ru/eng/ppi1647>.
15. *Федюкович В. Е.* Изменчивые ключи подписи // Информационные технологии и системы (ИТиС'09): сборник трудов конференции. [Электронный ресурс] М.: ИППИ РАН, 2009. С. 396–400. <http://www.iitp.ru/ru/conferences/539.htm> (3.12.2009).
16. *Федюкович В. Е.* Протокол аргумента для цикла Гамильтона // Препринт IACR. 2008. <http://eprint.iacr.org/2008/363> (23.08.2008).
17. *Fedyukovich V.* Protocols for graph isomorphism and hamiltonicity // 9th Central European Conference on Cryptography — Trebic'09, June 23–26, 2009. The proceedings will be published as a special issue of Tatra Mountains Mathematical Publications.
18. *Федюкович В. Е., Шаранов В. Г.* Протокол демонстрации K -кратного вхождения строки // Информационные технологии и системы (ИТиС'08): сборник трудов конференции. [Электронный ресурс] М.: ИППИ РАН, 2008. С. 459–466. http://www.iitp.ru/upload/content/340/itas08_proceedings.pdf (9.09.2008).

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

DOI 10.17223/20710410/6/7

УДК 621.391.1:004

КЛЕТОЧНО-АВТОМАТНОЕ МОДЕЛИРОВАНИЕ ДИФФУЗИОННЫХ ПРОЦЕССОВ НА ТРИАНГУЛЯЦИОННЫХ СЕТКАХ

А. А. Евсеев, О. И. Нечаева

*Новосибирский государственный университет, г. Новосибирск, Россия***E-mail:** evseev.alexei@gmail.com

Работа посвящена построению аппарата клеточно-автоматного моделирования для триангуляционных сеток на плоских и криволинейных поверхностях. Исследования возможностей аппарата проводились на примере клеточных автоматов, моделирующих процесс диффузии, диффузионного распространения фронта и агрегации, ограниченной диффузией.

Ключевые слова: *клеточный автомат, триангуляция, диффузия, распространение фронта.*

Введение

Большинство клеточно-автоматных (КА) моделей создаются для прямоугольных сеток на плоскости [1, 2]. Целью данной работы является построение и изучение клеточных автоматов на различных триангуляционных сетках, что сделало возможным реализацию клеточных автоматов и наблюдение процессов на криволинейных поверхностях в трёхмерном пространстве. Кроме того, появляется возможность КА-моделирования на «реальных» моделях объектов любой формы, например на адаптивных неструктурированных сетках. При таком подходе сразу же учитывается геометрия объектов. Нельзя не обратить внимание на широкое распространение триангуляционных сеток и их доступность с появлением технологии лазерного сканирования.

1. Клеточные автоматы на триангуляции

1.1. Постановка задачи

Построение клеточных автоматов на триангуляционных сетках связано с несколькими существенными преимуществами триангуляции. Во-первых, любая поверхность может быть аппроксимирована с необходимой точностью сеткой из треугольников. Во-вторых, вычислительная сложность алгоритмов разбиения на треугольники существенно меньше, чем при использовании других полигонов. В-третьих, в настоящее время имеется тенденция повсеместно задавать объекты триангуляцией.

Такой подход даёт возможности строить КА на произвольных криволинейных поверхностях и наблюдать за их эволюцией непосредственно на исследуемой поверхности. Целью данной работы является:

- 1) построение методов КА-моделирования на триангуляционных сетках;
- 2) исследование влияния параметров сетки на результат работы КА;
- 3) сравнение КА на триангуляционных сетках с КА на прямоугольных сетках;

- 4) построение КА для процесса диффузии, диффузионного распространения фронта концентрации вещества и агрегации, ограниченной диффузией на триангуляции;
- 5) разработка программного пакета для моделирования.

1.2. Основные определения

Для КА-моделирования на триангуляции требуется ввести несколько понятий. Пусть

A — алфавит состояний, например, $A_B = \{0, 1\}$ — булев алфавит, $A_R = [0, 1]$ — вещественный алфавит;

M — множество имён клеток, например, $M = \{m_i : i = 1, \dots, N\}$;

клетка — пара (a, m) , где $a \in A, m \in M$, a называется состоянием клетки (обозначается $a(m)$);

клеточный массив — множество клеток $\Omega = \{(a(m), m) : m \in M\}$.

Каждому треугольнику из триангуляции соответствует клетка. Таким образом, всей триангуляции соответствует клеточный массив.

Шаблон соседства для клетки (a, m) — это множество имён клеток, обычно лежащих к данной клетке. Например, для триангуляционной сетки две клетки будут считаться соседними, если соответствующие им треугольники имеют общую сторону. Таким образом, у каждого треугольника может быть не более трёх соседей (рис. 1).

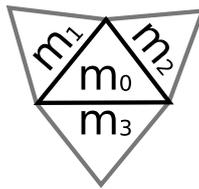


Рис. 1. Шаблон соседства для триангуляционной сетки $T = \{m_0, m_1, m_2, m_3\}$

Чтобы не задавать шаблон перебором, вводится именуемая функция на множестве M . Каждая именуемая функция клетки (a, m_0) указывает на имя одного из соседей этой клетки:

$$\begin{aligned} m_1 &= \varphi_1(m_0), \\ m_2 &= \varphi_2(m_0), \\ m_3 &= \varphi_3(m_0). \end{aligned}$$

Тогда шаблон задается так: $T = \{m_0, \varphi_1(m_0), \varphi_2(m_0), \varphi_3(m_0)\}$.

Правило перехода — это некоторая функция, которая определяет новое состояние клетки в зависимости от её текущего состояния, от клеток с именами из T или от каких-либо других величин (например, вероятностей), причём для всех клеток эта функция одинакова.

Синхронный режим функционирования КА — правило перехода применяется одновременно ко всем клеткам клеточного массива.

Асинхронный режим функционирования КА — из клеточного массива случайным образом выбирается клетка, и к ней применяется правило перехода.

Глобальная итерация — применение правила перехода к $|M|$ клеткам клеточного массива M . Последовательность клеточных массивов, каждый из которых получается из предыдущего в результате глобальной итерации, называется эволюцией КА.

Операция осреднения очень часто применяется для КА над булевым алфавитом для получения вещественных значений — вычисляется среднее значение по некоторой окрестности каждой клетки

$$\langle a \rangle = \frac{1}{|T_{\text{av}}|} \sum_{m_i \in T_{\text{av}}} a(m_i), \quad (1)$$

где $T_{\text{av}} \subseteq M$ — некоторая область осреднения.

Дискретизация является обратной операцией к осреднению: по вещественным значениям строится булев массив, при этом полагается

$$a' = \begin{cases} 1, & \text{если } \text{rand} < a, \\ 0, & \text{если } \text{rand} \geq a, \end{cases} \quad (2)$$

где rand — случайное число из $[0, 1]$, $a \in [0, 1]$.

1.3. Особенности использования триангуляции в КА

Следует заметить, что треугольники в триангуляции могут быть разные по площади и не обязательно равносторонние, в отличие от квадратов в прямоугольных сетках. Это необходимо учитывать при исследовании некоторых клеточных автоматов. Например, в асинхронном КА на прямоугольной сетке из квадратов выбор случайной клетки на одной итерации имеет равномерное распределение. Для получения независимости от используемой сетки выбор случайного треугольника на триангуляции нужно производить, основываясь на его площади; более мелкие частицы (треугольники с меньшей площадью) быстрее вступают в процесс, таким образом, чем меньше площадь, тем с большей вероятностью этот треугольник должен выбираться в качестве случайного.

Ещё один аспект, который хотелось бы осветить, — это замыкание границ в прямоугольных сетках. Ввиду простой структуры, края прямоугольных сеток очень часто замыкают, тем самым образуя тор. На триангуляции (в плоском случае) такой возможности нет ввиду возможной сложности области. При рассмотрении конкретных КА на триангуляции в данной работе возможность замыкания границ игнорируется. Это объясняется тем, что плоский случай обычно не представляет интереса, а для криволинейных поверхностей в трёхмерном пространстве очень часто рассматривается замкнутая триангуляция (у которой отсутствуют граничные треугольники). Треугольник считается граничным, если он имеет менее трёх соседей, иначе — внутренним.

Кроме того, существенную роль играет то, что у каждой внутренней клетки три соседа, в отличие от четырёх в прямоугольных сетках.

2. Клеточно-автоматная диффузия на прямоугольных сетках

2.1. Основной алгоритм КА-диффузии на прямоугольных сетках

Диффузия — процесс беспорядочного блуждания частиц, который приводит к выравниванию концентрации вещества в пространстве. В двумерном непрерывном случае при постоянном коэффициенте диффузии d процесс описывается уравнением Лапласа

$$\frac{\partial}{\partial t} u(x, y, t) = d \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right),$$

где $u(x, y, t)$ — концентрация вещества в точке декартова пространства с координатами x, y в момент времени t . Классические КА-модели диффузии имеют булев алфавит

и эволюцию в виде последовательности булевых массивов. Состояния клеток (0 или 1) определяют наличие или отсутствие единицы массы, которая не наделена скоростью. КА-диффузия на прямоугольных сетках уже довольно хорошо изучена [3].

Одной из КА-моделей диффузии является *наивная диффузия* [4]. Это наиболее примитивная модель диффузии, которая непосредственным образом отображает представление о процессе как о блуждании частиц в стремлении выровнять концентрацию вещества в пространстве. Режим функционирования КА — асинхронный, что вполне соответствует природе процесса. Окрестностью клетки являются её ближайшие четыре соседа. Правило функционирования таково, что на каждой итерации выбирается случайная ячейка, которая меняется своим значением равновероятно с одним из своих соседей. При таком правиле видно выполнение закона сохранения масс, а случайный выбор одного из соседей соответствует беспорядочному блужданию частиц в соответствии с определением процесса диффузии.

Ещё одна из моделей КА-диффузии — *КА-диффузия с окрестностью Марголуца* [4]. Для краткости будем называть её ТМ-диффузией по первым буквам фамилий её авторов, как это принято в западной литературе. ТМ-диффузия является более популярной, чем наивная диффузия, по двум причинам. Во-первых, потому, что она обладает свойством, которое в математике называют «элегантностью», то есть сочетанием простоты и эффективности. Во-вторых, существует строгое математическое доказательство, что она аппроксимирует оператор Лапласа [5]. Клеточный массив разбивают на два подмножества, каждое из этих подмножеств состоит из блоков, содержащих четыре клетки. Функционирование КА происходит в двухтактном синхронном режиме. Каждая итерация делится на два такта. На чётных тактах правила перехода применяются к чётным блокам, на нечётных — к нечётным. Правила перехода таковы, что выполняют сдвиг состояний в клетках блока равновероятно по часовой стрелке или против часовой стрелки. Уменьшая p и манипулируя значениями шагов во времени и пространстве, можно моделировать процесс диффузии с коэффициентом в широком диапазоне [1].

Проверить тот факт, что КА действительно моделирует диффузию, можно двумя путями: аналитически и экспериментально. Аналитическое доказательство построено только для одной КА-модели — ТМ-диффузии. Экспериментальное доказательство состоит в выполнении процесса моделирования и сравнении эволюции КА с решением уравнения на некотором наборе итераций.

2.2. Д и ф ф у з и о н н о е р а с п р о с т р а н е н и е ф р о н т а н а п р я м о у г о л ь н ы х с е т к а х

Распространение фронта — это такой процесс, при котором происходит равномерное распространение частиц, со временем заполняющих всю область. Распространение фронта может моделироваться *композиционным клеточным автоматом*. Это означает, что на каждой итерации к клеточному массиву последовательно применяется несколько правил. Композиционные клеточные автоматы хорошо отражают реальные физические процессы, так как в большинстве случаев они включают в себя несколько явлений [6].

Последовательность правил в КА распространения фронта:

- 1) проводится одна глобальная итерация диффузии;
- 2) полученный массив осредняется по формуле (1);
- 3) добавляется поток: в каждой клетке значение концентрации u заменяется на значение $0,5u(1 - u)$;

4) производится дискретизация по формуле (2).

Хотелось бы отметить, что добавление потока и дискретизация являются операциями, не зависящими от типа сетки, поэтому они могут быть легко распространены на трёхмерный случай на триангуляционной сетке.

3. КА-диффузия на триангуляции в плоскости

3.1. Основной алгоритм КА-диффузии на триангуляции в плоскости

Пусть сетка состоит из равносторонних треугольников. Воспользуемся введёнными основными определениями относительно клеточных автоматов на триангуляции. Правило функционирования КА будет аналогичным случаю на прямоугольной сетке, только вероятность выбора одного из соседей станет $1/3$. Тестирование работы клеточного автомата проводится над булевым алфавитом. В начальный момент времени в середине сетки вводится несколько рядом расположенных частиц. На рис. 2 представлена работа автомата (плоскость отмечена серым цветом). Как видно из рисунков, частицы распространились равномерно во все стороны. Таким образом, наблюдается визуальная аналогия с КА на прямоугольной сетке (более формальное сравнение будет произведено ниже). Такой эффект был достигнут благодаря тому, что все треугольники одинаковые (равносторонние). Но так как поставленная задача состоит в том, чтобы отойти от ограничений на используемую сетку, в ходе работы были подобраны правила функционирования клеточного автомата для произвольной сетки. Сосед по меньшей длине стороны треугольника выбирается вероятнее, чем по большей. Таким образом, вероятности выбора соседа относительно длины стороны будут выглядеть следующим образом:

$$\frac{1/\text{length}_i}{\sum_{i=1}^3 1/\text{length}_i}, \quad (3)$$

где length_i — длина соответствующей стороны в треугольнике.

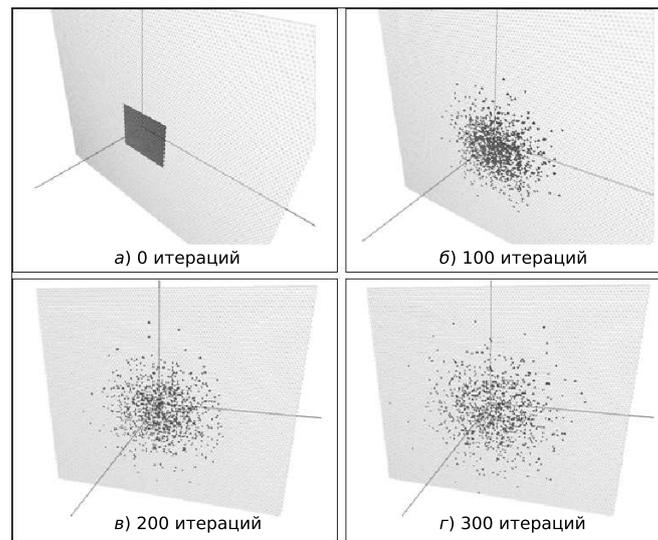


Рис. 2. Процесс диффузии на сетке из равносторонних треугольников

В качестве примера рассмотрим КА-диффузию на неструктурированной сетке (рис. 3).

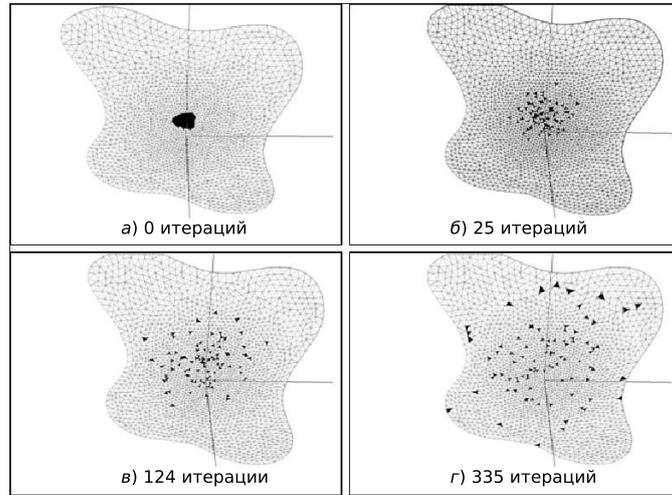


Рис. 3. Процесс диффузии на адаптивной сетке

Несмотря на относительно небольшое число треугольников в триангуляции (4129), процесс диффузии продолжительное время распространяет частицы лишь в середине области, так как размеры треугольников в этом месте значительно меньше, чем на краях. Но по истечении достаточно длительного времени частицы выходят за границы внутренней окружности с большой плотностью сетки.

3.2. Осреднение в КА-диффузии на триангуляции в плоскости

Опишем алгоритм *осреднения по окружности* для клеточных автоматов на триангуляционных сетках. Рассматривается центр каждого треугольника из триангуляции и проводится окружность некоторого радиуса из этого центра. Затем подсчитывается общее количество клеток, попавших в эту окружность, — `numOfTrianglesInCircle` и количество клеток в состоянии 1 — `numOfParticles`. Таким образом, значение концентрации будет определяться формулой

$$u = \frac{\text{numOfParticles}}{\text{numOfTrianglesInCircle}}, \quad (4)$$

где `numOfTrianglesInCircle` $\neq 0$ вне зависимости от значения радиуса, так как рассматриваемый треугольник заведомо лежит в этой окружности. Если соответствующим образом подобрать значение радиуса окружности, проводимой для осреднения, то получается наглядная картина (рис. 4).

Недостаток этого алгоритма в его вычислительной сложности, составляющей $O(N^2)$, где N — количество треугольников в триангуляции. В связи с этим рассматривается ещё один алгоритм осреднения с гораздо меньшей вычислительной трудоёмкостью — *осреднение по ближайшим соседям*. Суть этого алгоритма заключается в рассмотрении лишь соседних клеток и проведении среди них подсчёта клеток, находящихся в состоянии 1. Состояние самой клетки, для которой проводится осреднение, тоже влияет на значение `numOfParticles`:

$$u = \frac{\text{numOfParticles}}{\text{numOfNeighbors} + 1}, \quad (5)$$

где `numOfNeighbors` — количество соседей, `numOfNeighbors` $\in \{0, 1, 2, 3\}$.

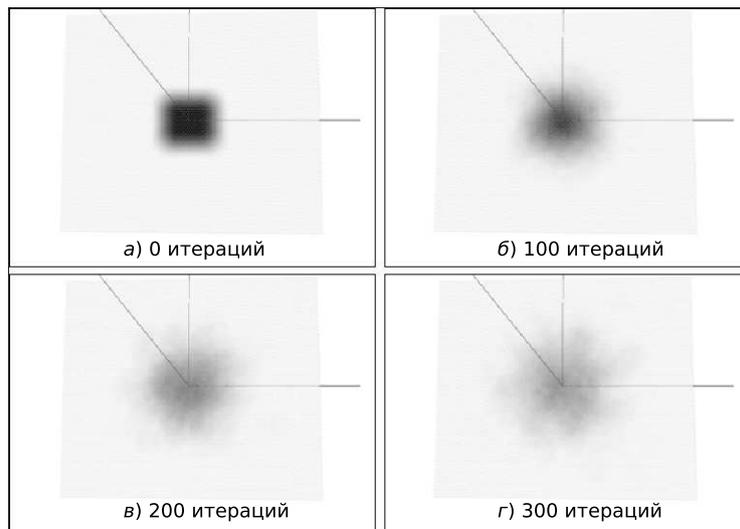


Рис. 4. Процесс диффузии с осреднением по окружности

При таком подходе возможных значений концентрации становится всего лишь 7: 0, 1/4, 1/3, 1/2, 2/3, 3/4, 1. Зато вычислительная сложность этого алгоритма $O(N)$, где N — количество треугольников в триангуляции. Такого выигрыша удалось добиться за счёт введения именуемой функции, которая по данному треугольнику возвращает трёх его соседей.

Несмотря на малое количество возможных принимаемых значений, картина всё же получается наглядной. В центре начального скопления концентрация сохраняет большие значения продолжительное время, но со временем наступает её равномерное распределение по всей области (рис. 5).

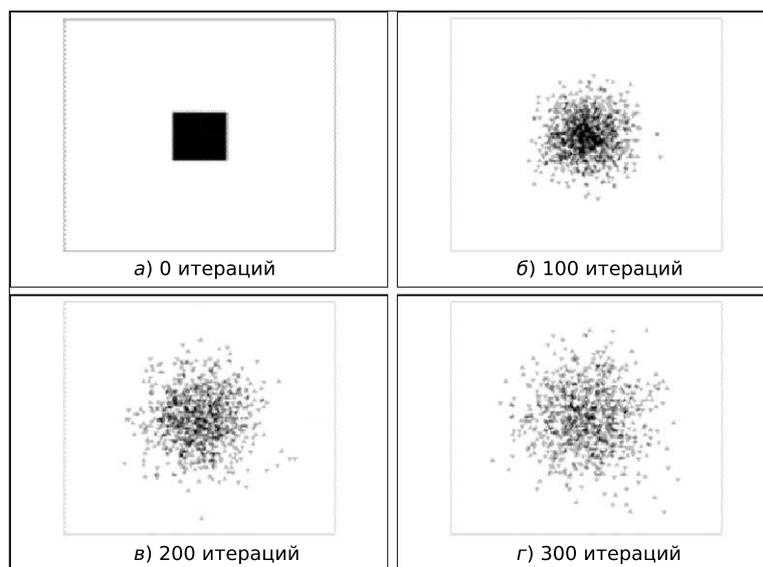


Рис. 5. Процесс диффузии с осреднением по ближайшим соседям

Этот алгоритм можно распространить на соседство порядка 2, то есть включать в рассмотрение соседей второго уровня. Тогда получится более гладкая картина. Аналогичным образом можно распространить алгоритм на соседство порядка n .

3.3. Сравнение КА-диффузии на триангуляции с КА-диффузией на прямоугольных сетках

Введём критерий соответствия построенного КА клеточным автоматам на прямоугольных сетках. Ввиду существования модели КА-диффузии лишь для плоского случая, сравнение возможно только с ним. В качестве сетки рассматривается квадратная область, заданная равносторонними треугольниками. Так как диффузия — процесс беспорядочного блуждания частиц, то сравнение автоматов над булевым алфавитом не представляет интереса. Рассматриваются автоматы с осреднением по окружности. На рис. 6 иллюстрируется сравнение вещественных значений концентрации

$$u = \frac{u_1 + u_2}{2} \longleftrightarrow u_0, \quad (6)$$

где u_0, u_1, u_2 — значения концентрации, соответствующие клетке.

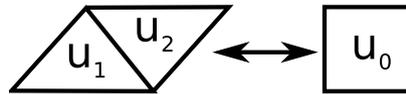


Рис. 6. Сравнение концентраций на прямоугольной и квадратной сетках

Таким образом, одному значению концентрации в квадратной сетке ставится в соответствие среднее арифметическое концентраций двух соседних клеток в триангуляционной сетке. При таком подходе количество клеток в клеточном массиве на триангуляционной сетке в два раза больше, чем на квадратной. Так как клеток больше, то очень часто в качестве случайной клетки выбирается пустая, которая впоследствии, с большой вероятностью, также меняется с пустой клеткой. Это приводит к тому, что непустые клетки выбираются реже, следовательно, процесс диффузии идёт медленнее, а концентрация вещества выше. Для соответствия значений концентраций на триангуляционной сетке проводится в два раза больше итераций. Сравнение проводится по клеткам, расположенным на горизонтальной средней линии в клеточных массивах. Для наглядности на рис. 7 приводятся также графики нормального распределения, отражающие аналитическое решение уравнения Лапласа.

Критерием соответствия КА на триангуляции и на прямоугольных сетках будем считать аналогию диаграмм срезов по горизонтальной средней линии.

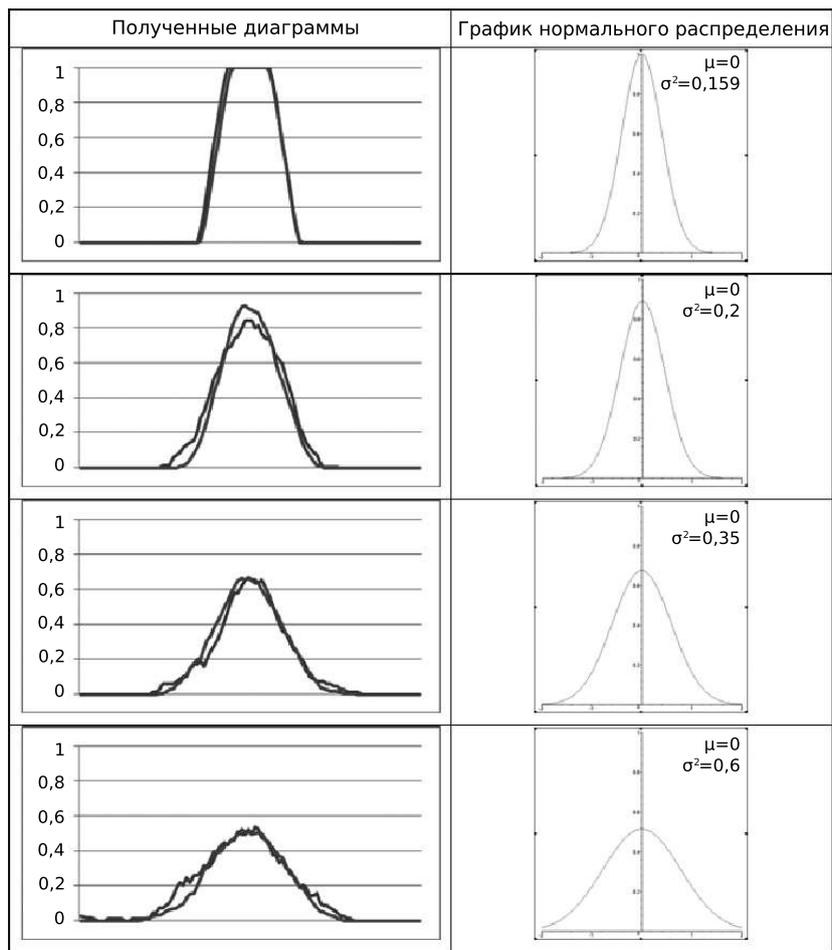


Рис. 7. Результаты сравнения КА-диффузии на триангуляции (число итераций увеличено в два раза) с КА на квадратной сетке и графики нормального распределения

4. КА-диффузия на триангуляции (криволинейная поверхность)

4.1. Основной алгоритм КА-диффузии на триангуляции (криволинейная поверхность)

Имея полный инструментарий для дальнейшего исследования, приступим к изучению поведения КА на криволинейных поверхностях в трёхмерном пространстве. Единственную трудность в этом случае представляет операция осреднения по окружности. Если проводить сферу из центра треугольника, то соответствия не получится: в сферу могут попасть клетки, расположенные на противоположной стороне сетки, что неприемлемо. Для получения вещественных значений концентрации вещества можно использовать осреднение по ближайшим соседям. В качестве сетки рассматривается поверхность кости, заданная триангуляцией [7]. В начальный момент времени вносится концентрация в небольшой участок кости (рис. 8). Как видно из рисунков, по прошествии некоторого времени частицы распространились равномерно по всей поверхности кости. Причём в ходе работы клеточного автомата частицы расходились во все стороны одинаково. Построенный автомат естественно отразил физический процесс на сложной поверхности.

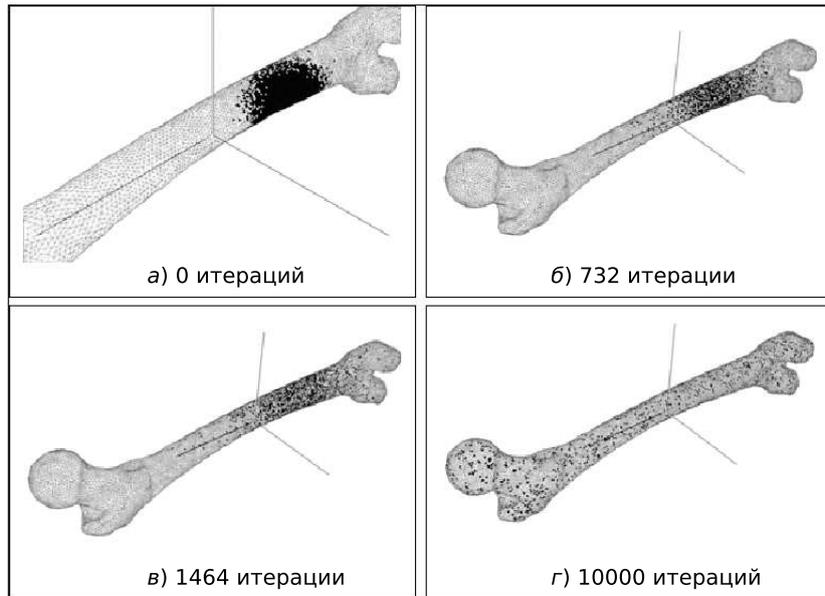


Рис. 8. Процесс диффузии на кости

4.2. Диффузионное распространение фронта на криволинейной поверхности

Далее рассматривается клеточный автомат распространения фронта на триангуляции. В качестве сетки вновь берётся поверхность кости с начальным состоянием, как и в п. 4.1. Процесс диффузионного распространения фронта на такой сетке можно интерпретировать как распространение некоего воспаления из локального участка на всю поверхность кости (рис. 9).

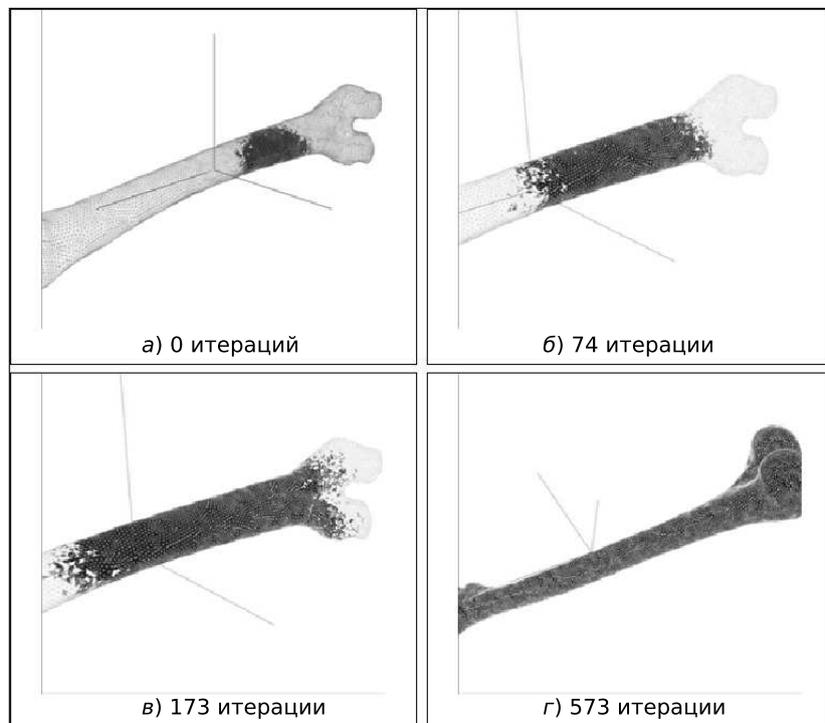


Рис. 9. Процесс распространения фронта на кости

Фронт распространился равномерно во все стороны и со временем охватил всю поверхность кости, что соответствует определению процесса.

4.3. Агрегация, ограниченная диффузией, на триангуляции (криволинейная поверхность)

Последним из рассматриваемых в данной работе клеточных автоматов является КА агрегации, ограниченной диффузией.

В начальном состоянии 10% частиц случайным образом разбросаны по всей сетке. Одна клетка является «источником», она имеет другой цвет и обездвижена. Необездвиженные частицы двигаются в случайных направлениях, но при соприкосновении с обездвиженной они перестают двигаться и перекрашиваются. Результат работы КА приведён на рис. 10.

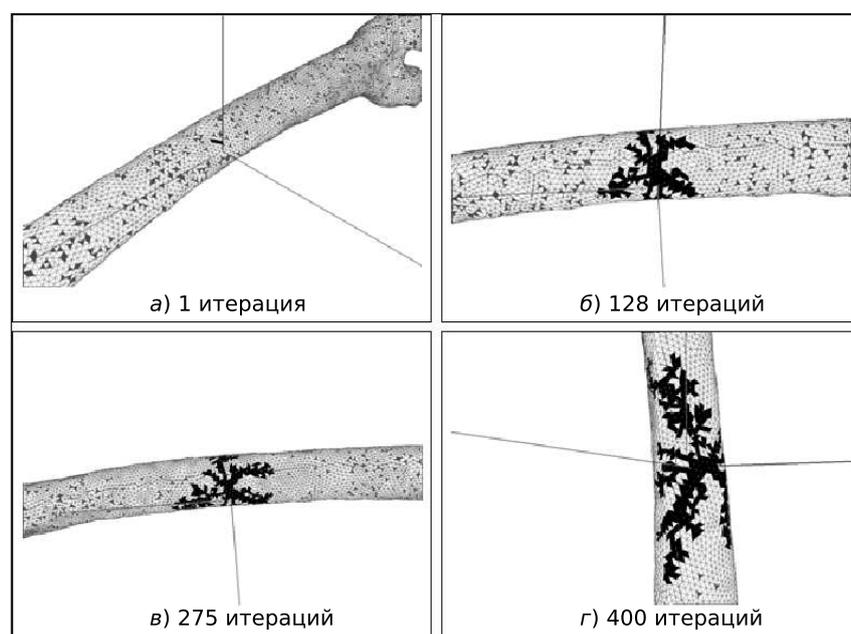


Рис. 10. Процесс агрегации, ограниченной диффузией, на кости

4.4. Области применения

Как отмечалось выше, область применения не ограничивается моделированием чисто физических процессов. КА-диффузия может быть использована во многих областях, вплоть до компьютерных игр и обработки изображений. Отличие может быть только в интерпретации данного процесса в конкретном случае его применения. В моделировании распространения каких-либо жидкостей или газов очень часто одной из составляющей процесса является именно диффузия. Таким образом, полученные результаты могут быть применены для построения более сложных композиционных КА, в которых одним из правил перехода будет диффузия.

Заключение

Были получены аналоги клеточно-автоматной диффузии, ограниченной диффузии и диффузионного распространения фронта для произвольной криволинейной поверхности в трёхмерном пространстве, заданной триангуляцией. Полученные результаты качественно не уступают КА на прямоугольных сетках и могут служить хорошей основой для моделирования различных процессов. Из приведённых выше рассуждений

можно сделать вывод, что применение КА-диффузии не ограничивается физическими процессами, а может трактоваться по-разному в зависимости от исследуемой задачи. Так, например, приведённый пример с распространением воспаления по кости может быть использован в медицине. Отметим важность композиционных клеточных автоматов: в большинстве моделей пространственной динамики присутствует диффузионная составляющая (примером может служить рассмотренный процесс диффузионного распространения фронта). Приведённые алгоритмы могут быть применены для других клеточно-автоматных моделей.

ЛИТЕРАТУРА

1. *Бандман О. Л.* Клеточно-автоматные модели пространственной динамики // Системная информатика. 2006. № 10. С. 59–113.
2. *Wolfram S.* A new kind of science. Champaign, Ill., USA: Wolfram Media Inc., 2002.
3. *Bandman O.* Comparative Study of Cellular automata Diffusion Models // LNCS. 1999. No. 1662 (V. Malyshkin, ed.). P. 395–399.
4. *Тюффоли Т., Марголюс Н.* Машины клеточных автоматов. М.: Мир, 1991.
5. *Малинецкий Г. Г., Степанцов М. Е.* Моделирование диффузионных процессов клеточными автоматами с окрестностью Марголюса // Журн. вычислит. матем. и математич. физ. 1998. Т. 36. № 6. С. 1017–1021.
6. *Bandman O. L.* Cellular Automata composition techniques for spatial dynamics simulation // Bulletin of the Novosibirsk Computing Center. 2008. No. 27. С. 1–40.
7. *Helwig P., Faust G.* Finite element analysis of a bone-implant system with the proximal femur nail // Technol. Health Care. 2006. V. 14. No. 4–5. P. 411–419.

«ЛЕНТОЧНАЯ» ТЕОРЕМА И ЕЕ ПРИЛОЖЕНИЯ

В. В. Скобелев

Институт прикладной математики и механики НАН Украины, г. Донецк, Украина

E-mail: vv_skobelev@iamm.ac.donetsk.ua

Предлагается общий метод подсчета числа элементов конечных множеств, определенных в терминах классов вычетов, при помощи набора размеченных лент.

Ключевые слова: *классы вычетов, системы сравнений.*

1. Базовая конструкция

Объектом исследования является следующая «ленточная конструкция».

Под лентой будем понимать одностороннюю бесконечную (вправо) ленту, разбитую на идентичные клетки, занумерованные (слева направо) неотрицательными целыми числами (т.е. элементами множества \mathbb{Z}_+).

Зафиксируем число $n \in \mathbb{N}$ и расположим одну под другой $n+1$ лент, перенумеровав их сверху вниз числами $1, 2, \dots, n+1$. Ленты с номерами $1, 2, \dots, n$ назовем рабочими лентами, а ленту с номером $n+1$ — результирующей лентой.

Пусть a_1, \dots, a_n — попарно взаимно простые натуральные числа, а b_1, \dots, b_n — такие неотрицательные целые числа, что $b_i \leq a_i$ для всех $i = 1, \dots, n$.

Отметим клетки лент маркером в соответствии со следующими тремя правилами.

Правило 1. На i -й ($i = 1, \dots, n$) рабочей ленте среди первых a_i клеток отметим маркером произвольные b_i клеток.

Правило 2. На i -й ($i = 1, \dots, n$) рабочей ленте клетка с номером h ($h \geq a_i$) отмечена маркером тогда и только тогда, когда клетка с номером $h \bmod a_i$ отмечена маркером.

Правило 3. На результирующей ленте клетка с номером $j \in \mathbb{Z}_+$ отмечена маркером тогда и только тогда, когда клетка с номером j отмечена маркером на каждой рабочей ленте.

Обозначим через L_i ($i = 1, \dots, n+1$) начальный отрезок i -й ленты, состоящий из первых $\prod_{i=1}^n a_i$ клеток.

Назовем «ленточной конструкцией» упорядоченный набор лент

$$(L_1, \dots, L_{n+1}).$$

Покажем, что «ленточная конструкция» применима для подсчета числа элементов конечных множеств, определенных в терминах классов вычетов.

2. «Ленточная» теорема

Следующая теорема характеризует количество отмеченных клеток результирующей ленты в «ленточной конструкции» (L_1, \dots, L_{n+1}) .

Теорема 1. В точности $\prod_{i=1}^n b_i$ клеток результирующей ленты L_{n+1} отмечены маркером.

Доказательство. Из определения «ленточной конструкции» (L_1, \dots, L_{n+1}) вытекает, что процесс разметки клеток результирующей ленты L_{n+1} можно осуществить *методом решета*, состоящим из n этапов, причем на i -м этапе ($i = 1, \dots, n$) участвует только результирующая лента L_{n+1} и i -я рабочая лента L_i . Эти этапы имеют следующий вид.

На 1-м этапе на результирующей ленте L_{n+1} маркером отмечаются те и только те клетки, для которых соответствующие клетки 1-й рабочей ленты L_1 отмечены маркером.

На i -м этапе ($i = 2, \dots, n$) изменим разметку результирующей ленты L_{n+1} следующим образом: сотрем маркеры с тех и только тех клеток результирующей ленты L_{n+1} , для которых соответствующие клетки i -й рабочей ленты L_i не отмечены маркером.

Методом индукции подсчитаем число клеток, отмеченных маркером на результирующей ленте L_{n+1} .

Рассмотрим 1-й этап. На этом этапе участвуют только 1-я рабочая лента L_1 и результирующая лента L_{n+1} , у которой ни одна из клеток не отмечена маркером.

Из правил 1 и 2 вытекает, что после 1-го этапа:

1) на результирующей ленте L_{n+1} отмечено маркером в точности $b_1 \cdot \prod_{j=2}^n a_j$ клеток, причем отмечены те и только те клетки, номера которых имеют вид

$$r + a_1 \cdot h \quad (h = 0, 1, \dots, \prod_{j=2}^n a_j - 1),$$

где r ($0 \leq r \leq a_1 - 1$) — номер отмеченной маркером клетки 1-й ленты;

2) среди первых a_1 клеток результирующей ленты L_{n+1} маркером отмечены в точности b_1 клеток.

Предположим, что после $(i - 1)$ -го этапа ($i = 2, \dots, n$):

1) на результирующей ленте L_{n+1} отмечено маркером в точности $(\prod_{j=1}^{i-1} b_j) \cdot (\prod_{j=i}^n a_j)$ клеток, причем отмечены те и только те клетки, номера которых имеют вид

$$r_1 + (\prod_{j=1}^{i-1} a_j) \cdot h_1 \quad (h_1 = 0, 1, \dots, \prod_{j=i}^n a_j - 1),$$

где r_1 ($0 \leq r_1 \leq \prod_{j=1}^{i-1} a_j - 1$) — номер клетки, отмеченной маркером на каждой из лент L_1, \dots, L_{i-1} ;

2) среди первых $\prod_{j=1}^{i-1} a_j$ клеток результирующей ленты L_{n+1} маркером отмечены в точности $\prod_{j=1}^{i-1} b_j$ клеток.

Рассмотрим i -й этап ($i = 2, \dots, n$). На этом этапе участвуют только i -я рабочая лента L_i и результирующая лента L_{n+1} .

Из правил 1 и 2 вытекает, что число отмеченных маркером клеток на ленте L_i равно $b_i \cdot (\prod_{j=1}^{i-1} a_j) \cdot (\prod_{j=i+1}^n a_j)$, причем на ленте L_i отмечены маркером те и только те клетки, номера которых имеют вид

$$r_2 + a_i \cdot h_2 \quad (h_2 = 0, 1, \dots, (\prod_{j=1}^{i-1} a_j) \cdot (\prod_{j=i+1}^n a_j) - 1),$$

где r_2 ($0 \leq r_2 \leq a_i - 1$) — номер отмеченной маркером клетки ленты L_i .

Рассмотрим фрагменты лент L_i и L_{n+1} , состоящие из первых

$$a_i \cdot \prod_{j=1}^{i-1} a_j = \prod_{j=1}^i a_j$$

клеток.

Зафиксируем номер r_2 ($0 \leq r_2 \leq a_i - 1$) отмеченной маркером клетки ленты L_i .

Для каждого фиксированного номера r_1 ($0 \leq r_1 \leq \prod_{j=1}^{i-1} a_j - 1$) отмеченной маркером клетки ленты L_{n+1} числа

$$r_1 + \left(\prod_{j=1}^{i-1} a_j \right) \cdot h_1 \quad (h_1 = 0, 1, \dots, a_i - 1) \quad (1)$$

образуют полную систему вычетов по модулю a_i . Следовательно, только для одного из чисел (1) истинно сравнение

$$r_1 + \left(\prod_{j=1}^{i-1} a_j \right) \cdot h_1 \equiv r_2 \pmod{a_i}.$$

Итак, в результате выполнения i -го этапа каждая пара чисел

$$(r_1, r_2) \quad (0 \leq r_1 \leq \prod_{j=1}^{i-1} a_j - 1; 0 \leq r_2 \leq a_i - 1)$$

определяет на фрагменте результирующей ленты L_{n+1} , состоящем из первых $\prod_{j=1}^i a_j$ клеток, единственную отмеченную маркером клетку.

Отсюда вытекает, что после выполнения i -го этапа на фрагменте результирующей ленты L_{n+1} , состоящем из первых $\prod_{j=1}^i a_j$ клеток, число отмеченных маркером клеток равно

$$\left(\prod_{j=1}^{i-1} b_j \right) \cdot b_i = \prod_{j=1}^i b_j.$$

На оставшейся части результирующей ленты L_{n+1} эта разметка периодически повторяется. Следовательно, после выполнения i -го этапа общее число отмеченных маркером клеток результирующей ленты L_{n+1} равно

$$\left(\prod_{j=1}^i b_j \right) \cdot \left(\prod_{j=i+1}^n a_j \right). \quad (2)$$

Положив $i = n$ в (2), получим утверждение теоремы. ■

3. Приложения

Покажем, каким образом теорема 1 может быть применена при решении задач теории чисел, связанных с подсчетом количеств натуральных чисел, обладающих заданным свойством [1, 2].

Пример 1. Докажем свойство *мультипликативности* функции Эйлера $\varphi(k)$ ($k \in \mathbb{N}$), определяющей количество чисел, взаимно простых с числом k и не превосходящих k , а именно: для любых взаимно простых чисел $l_1, l_2 \in \mathbb{N}$ истинно равенство

$$\varphi(l_1 \cdot l_2) = \varphi(l_1) \cdot \varphi(l_2).$$

Положим $n = 2$ и рассмотрим «ленточную конструкцию»

$$(L_1, L_2, L_3),$$

где $a_i = l_i$ и $b_i = \varphi(l_i)$ для $i = 1, 2$.

Сформулируем правило 1 в следующем виде: на i -й ($i = 1, 2$) рабочей ленте среди первых a_i клеток маркером отмечены те и только те b_i клеток, номера которых — числа, взаимно простые с числом l_i .

Известно, что число a взаимно просто с числом $l_1 \cdot l_2$ тогда и только тогда, когда число a взаимно просто как с числом l_1 , так и с числом l_2 .

Отсюда вытекает, что в силу правила 3 клетка с номером r результирующей ленты L_3 отмечена маркером тогда и только тогда, когда число r взаимно просто с числом $l_1 \cdot l_2$.

Следовательно, число отмеченных маркером клеток результирующей ленты L_3 равно $\varphi(l_1 \cdot l_2)$.

Из теоремы 1 вытекает, что

$$\varphi(l_1 \cdot l_2) = b_1 \cdot b_2 = \varphi(l_1) \cdot \varphi(l_2),$$

что и требовалось доказать.

Пример 2. Докажем формулу Эйлера, а именно: если $m = p_1^{\alpha_1} \dots p_n^{\alpha_n}$, где p_1, \dots, p_n — попарно различные простые числа, то

$$\varphi(m) = m \cdot \prod_{i=1}^n (1 - p_i^{-1}).$$

Рассмотрим «ленточную конструкцию»

$$(L_1, \dots, L_{n+1}),$$

где $a_i = p_i^{\alpha_i}$ и $b_i = \varphi(p_i^{\alpha_i}) = p_i^{\alpha_i} - p_i^{\alpha_i-1}$ для всех $i = 1, \dots, n$.

Сформулируем правило 1 в следующем виде: на i -й ($i = 1, \dots, n$) рабочей ленте среди первых a_i клеток маркером отмечены те и только те b_i клеток, номера которых — числа, взаимно простые с числом p_i .

Отсюда вытекает, что в силу правила 3 клетка с номером r результирующей ленты L_{n+1} отмечена маркером тогда и только тогда, когда число r взаимно просто с каждым из чисел p_1, \dots, p_n .

Следовательно, число отмеченных маркером клеток результирующей ленты L_{n+1} равно $\varphi(m)$.

Из теоремы 1 вытекает, что

$$\varphi(m) = \prod_{i=1}^n b_i = \prod_{i=1}^n (p_i^{\alpha_i} - p_i^{\alpha_i-1}) = m \cdot \prod_{i=1}^n (1 - p_i^{-1}),$$

что и требовалось доказать.

Принимая во внимание, что в последнее время наблюдается тенденция к систематическому применению теории конечных колец и модулей линейных форм над конечными кольцами в процессе решения задач криптографии (по крайней мере, при решении задач анализа и синтеза поточных шифров), можно заключить, что теоретическое исследование этой общей комбинаторной схемы является актуальным как для комбинаторного анализа, так и с прикладной точки зрения. Анализ такой общей комбинаторной схемы является предметом дальнейших исследований.

В заключение автор выражает благодарность рецензенту, замечания которого позволили уточнить некоторые результаты.

ЛИТЕРАТУРА

1. Харин Ю. С., Берник В. И., Матвеев Г. В., Агиевич С. В. Математические и компьютерные основы криптологии. Минск: Новое знание, 2003. 382 с.
2. Лидл Р., Нидеррайтер Г. Конечные поля. Т. 1. М.: Мир, 1988. 430 с.

ПРИКЛАДНАЯ ТЕОРИЯ АВТОМАТОВ

DOI 10.17223/20710410/6/9

УДК 519.7

СИНТЕЗ УСЛОВНЫХ РАЗЛИЧАЮЩИХ ЭКСПЕРИМЕНТОВ
ДЛЯ АВТОМАТОВ С НЕДЕТЕРМИНИРОВАННЫМ ПОВЕДЕНИЕМ¹

М. Л. Громов, Н. В. Евтушенко

*Томский государственный университет, г. Томск, Россия***E-mail:** gromov@sibmail.com, ninayevtushenko@yahoo.com

Данная работа посвящена синтезу условных различающих экспериментов для трёх классов автоматов с недетерминированным поведением — детерминированных входо-выходных полуавтоматов, конечных недетерминированных автоматов и недетерминированных временных автоматов — без использования ограничения «всех погодных условий». Эти эксперименты строятся на основе пересечения различаемых автоматов и могут быть использованы при построении проверяющих и диагностических тестов.

Ключевые слова: *входо-выходной полуавтомат с молчанием, конечный недетерминированный автомат, временной автомат, различимость, совместимость, условный эксперимент.*

Введение

В современных исследованиях большое внимание уделяется тестированию дискретных управляющих систем, поведение которых описывается автоматами с недетерминированным поведением (см., например, [1–5]). Причины появления недетерминизма в спецификациях и реализациях систем, вообще говоря, различные; среди них можно выделить возможность различных опций при реализации, уровень абстракции описания, невозможность полной управляемости и наблюдаемости для реализации и так далее. В случае, когда поведение спецификации и реализации описывается автоматами с недетерминированным поведением, обычно требуется, чтобы поведение конформной реализации на допустимых входных последовательностях было частью поведения спецификации. При синтезе тестов с гарантированной полнотой необходимо уметь отличать неконформные реализации от спецификации, то есть уметь различать с помощью эксперимента два различимых автомата, имеющих недетерминированное поведение.

Одной из основных проблем при тестировании недетерминированных реализаций является тот факт, что в общем случае проверяемый автомат с недетерминированным поведением может реагировать различными выходными последовательностями на одну и ту же входную последовательность. Поэтому обычно при тестировании реализаций с недетерминированным поведением делается допущение о «всех погодных условиях» [1], то есть предполагается, что каждая тестовая входная последовательность подается на реализацию достаточно большое число раз в «различных погодных условиях», и поэтому можно считать, что тестер пронаблюдал все возможные выходные реакции реализации на каждую входную последовательность. Допущение о

¹Работа выполнена по ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2012 годы», гос.контракт № 02.514.12.4002.

«всех погодных условиях» является скорее теоретическим, так как никто не знает, сколько раз достаточно подать входную последовательность, чтобы дать возможность тестеру «увидеть» все выходные реакции реализации. Более того, такому предположению невозможно удовлетворить, если проверяемый автомат является только частично контролируемым, что имеет место, например, при удаленном тестировании реализаций телекоммуникационных протоколов. Известно, что если отсутствует предположение о «всех погодных условиях», то единственным отношением между проверяемым и эталонным автоматами, проверку которого можно гарантировать при проведении безусловного эксперимента с проверяемым автоматом, является отношение неразделимости между автоматами, когда реакции двух автоматов на любую входную последовательность пересекаются [6, 7]. В данной работе «все погодные условия» не предполагаются, но считается возможным проведение условного эксперимента [8] по различению двух автоматов, и для этой цели определяются необходимые и достаточные условия существования такого эксперимента.

Рассматриваются три класса автоматов с недетерминированным поведением, а именно: детерминированные входо-выходные полуавтоматы с молчанием [9], недетерминированные, возможно, частичные конечные автоматы [10] и недетерминированные временные автоматы с таймаутом [11]. Для каждого класса вводится понятие различимости автоматов в классе, которое сводится к существованию их общей конформной реализации, то есть к существованию такого автомата, поведение которого на допустимых входных последовательностях является частью поведения каждого из предъявленных автоматов, и показывается, что два автомата в классе различимы условным экспериментом, если и только если такой реализации не существует.

1. Различающий условный эксперимент с входо-выходными полуавтоматами

Под *детерминированным входо-выходным полуавтоматом* (или просто *полуавтоматом*) с входным алфавитом I и выходным алфавитом O будем понимать пятёрку $L = \langle L, I, O, \hat{l}, \lambda_L \rangle$, где L — непустое конечное множество состояний с выделенным начальным состоянием \hat{l} ; I и O — непересекающиеся конечные алфавиты входных и выходных действий (символов) соответственно, $I \cup O \neq \emptyset$. Отношение $\lambda_L \subseteq L \times (I \cup O) \times L$ есть отношение переходов, и если $\langle l, a, l' \rangle \in \lambda_L$ и $\langle l, a, l'' \rangle \in \lambda_L$, то $l' = l''$. Переход $\langle l, a, l' \rangle$ для удобства будем записывать как $l \xrightarrow{a} l'$.

Недетерминированность поведения полуавтомата проявляется в возможности совершения им в одном и том же состоянии разных выходных действий в отсутствие входного символа.

Состояние l полуавтомата L называется *молчащим*, если для любого действия a из O и любого состояния l' из L верно $\langle l, a, l' \rangle \notin \lambda_L$.

Предполагается, что все состояния полуавтомата достижимы из начального и что ситуацию, когда полуавтомат находится в молчащем состоянии, можно отличить от ситуации, когда полуавтомат находится в состоянии, в котором он готов произвести выходное действие. Для формального описания этой возможности вводится новое выходное действие δ , и если некоторое состояние l — молчащее, то подразумевается, что $l \xrightarrow{\delta} l$. Естественным образом отношение переходов распространяется на последовательности из алфавита $(I \cup O \cup \{\delta\})$.

Введём следующие обозначения:

- 1) $\llbracket L \rrbracket_\sigma$ — множество всех состояний, достижимых из состояния l по последовательности $\sigma \in (I \cup O \cup \{\delta\})^*$ (в детерминированном полуавтомате $\llbracket L \rrbracket_\sigma \leq 1$);

- 2) $\mathbf{out}(l)$ — множество всех выходных действий, включая δ , переходы по которым определены в состоянии l ;
- 3) $\mathbf{in}(l)$ — множество всех входных действий, переходы по которым определены в состоянии l ;
- 4) $s\text{-traces}_L$ — множество всех конечных последовательностей из $(I \cup O \cup \{\delta\})^*$, по которым возможны переходы из начального состояния полуавтомата L .

Полуавтомат L называется *полностью определённым по входам* (или просто *полностью определённым*), если для каждого состояния l справедливо $\mathbf{in}(l) = I$.

Полностью определённый полуавтомат L находится в *отношении \mathbf{ioco}* с полуавтоматом K (обозначение: $L \mathbf{ioco} K$), если для любой последовательности σ из $s\text{-traces}_K$ справедливо $\mathbf{out}(\llbracket \hat{l} \rrbracket_\sigma) \subseteq \mathbf{out}(\llbracket \hat{k} \rrbracket_\sigma)$; в этом случае полуавтомат L будем называть *\mathbf{ioco} -реализацией* полуавтомата K .

Суть отношения \mathbf{ioco} состоит в том, что поведение реализации в любом её состоянии определено на всех тех входных воздействиях, на которых определено поведение спецификации в соответствующем состоянии, а множество выходных реакций содержится в множестве выходных реакций спецификации.

Два полуавтомата L и K назовём *\mathbf{ioco} -совместимыми* (*\mathbf{ioco} -различимыми*), если для них существует (не существует) общая полностью определённая \mathbf{ioco} -реализация.

Пересечением полуавтоматов L и K назовём полуавтомат $J = L \cap K = \langle J, I, O \cup \{\Delta\}, \hat{j}, \lambda_J \rangle$, где $\Delta \notin I \cup O \cup \{\delta\}$; $\hat{j} = \langle \hat{l}, \hat{k} \rangle$; $J \subseteq L \times K$ — минимальное множество, полученное с использованием следующих правил для λ_J : если $\langle l, a, l' \rangle \in \lambda_L$ и $\langle k, a, k' \rangle \in \lambda_K$, то $\langle \langle l, k \rangle, a, \langle l', k' \rangle \rangle \in \lambda_J$; если l и k — молчащие состояния, то $\langle l, k \rangle \xrightarrow{\Delta} \langle l, k \rangle \in \lambda_J$.

Введём понятие *\mathbf{ioco} -различающего* полуавтомата.

Для этого рассмотрим такой полуавтомат $R = \langle R, I, O \cup \{\theta\}, \hat{r}, \lambda_R \rangle$, где $\theta \notin I \cup O \cup \{\delta\}$, в котором:

- 1) множество состояний R содержит три специальных тупиковых состояния \perp_L , \perp_K и \perp ;
- 2) для всякого $r \in R \setminus \{\perp, \perp_L, \perp_K\}$ либо $|\mathbf{in}(r)| = 1$, $\mathbf{out}(r) = \{\delta\}$ и $\llbracket r \rrbracket_i \cap \{\perp_L, \perp_K, \perp\} = \emptyset$, где $\{i\} = \mathbf{in}(r)$, либо $\mathbf{in}(r) = \emptyset$ и $\mathbf{out}(r) = O \cup \{\theta\}$;
- 3) граф переходов полуавтомата R ациклический.

Пересечением полуавтомата L и полуавтомата R назовём связный полуавтомат $J = L \cap_\theta R = \langle J, I, O \cup \{\theta\}, \hat{j}, \lambda_J \rangle$, где $\hat{j} = \langle \hat{l}, \hat{r} \rangle$, $J \subseteq L \times R$ — минимальное множество, полученное с использованием следующих правил для λ_J : если $\langle l, a, l' \rangle \in \lambda_L$ и $\langle r, a, r' \rangle \in \lambda_R$, то $\langle \langle l, r \rangle, a, \langle l', r' \rangle \rangle \in \lambda_J$; если l — молчащее и $\langle r, \theta, r' \rangle \in \lambda_R$, то $\langle l, r \rangle \xrightarrow{\theta} \langle l, r' \rangle \in \lambda_J$.

Говорят, что полуавтомат R *различает* полуавтоматы L и K , если в пересечении $J_{LR} = L \cap_\theta R$ для всех тупиковых состояний $\langle l, r \rangle$ верно $r = \perp_L$, в то время как в пересечении $J_{KR} = K \cap_\theta R$ для всех тупиковых состояний $\langle k, r \rangle$ верно $r = \perp_K$. Такой полуавтомат R будем называть *\mathbf{ioco} -различающим полуавтоматом* для полуавтоматов L и K и обозначать R_{LK} .

Предлагается способ построения \mathbf{ioco} -различающего автомата — алгоритм 1. В нём для $J = L \cap K$ состояние $\langle l, k \rangle \in J$ называется *1-недоопределённым*, если оно молчащее; состояние $\langle l, k \rangle \in J$ называется *z -недоопределённым*, если оно $(z-1)$ -недоопределённое или существует такое входное воздействие $i \in \mathbf{in}_J(\langle l, k \rangle)$, что все состояния в множестве $\llbracket \langle l, k \rangle \rrbracket_i$ являются $(z-1)$ -недоопределёнными, или для всех $o \in \mathbf{out}(\langle l, k \rangle)$ все состояния в $\llbracket \langle l, k \rangle \rrbracket_o$ являются $(z-1)$ -недоопределёнными.

Алгоритм 1. Построение **юсо**-различающего полуавтомата

Вход: Два детерминированных полуавтомата $L = \langle L, I, O, \hat{l}, \lambda_L \rangle$ и $K = \langle K, I, O, \hat{k}, \lambda_K \rangle$.

Выход: **юсо**-различающий полуавтомат R_{LK} , если L и K **юсо**-различимы.

- 1: $J_0 := \{\perp, \perp_L, \perp_K\}$; $\lambda_R := \emptyset$; $J := L \cap K$.
- 2: Построим Q_1 — множество 1-недоопределённых состояний, Q_2 — множество 2-недоопределённых состояний, и так далее, пока $Q_z \neq Q_{z+1}$.
- 3: **Если** $\forall z \langle \hat{l}, \hat{k} \rangle \notin Q_z$, **то**
- 4: Исходные полуавтоматы **юсо**-совместимы. **Конец.**
- 5: Пусть n — такое число, что $\langle \hat{l}, \hat{k} \rangle \in Q_n$, но $\langle \hat{l}, \hat{k} \rangle \notin Q_{n-1}$.
- 6: $z := n$; $J_n := \{\langle \hat{l}, \hat{k} \rangle\}$; $J_{n-1} := \emptyset$; $J_{n-2} := \emptyset$; ...; $J_1 := \emptyset$.
- 7: **Пока** ($z > 0$)
- 8: **Для всех** ($j \equiv \langle l, k \rangle \in J_z$)
- 9: **Если** $\text{out}_J(j) = \delta$ или $\llbracket j \rrbracket_{\text{out}_J(j)} \subseteq Q_{z-1}$, **то**
- 10: **Для каждого** $o \in (O \cup \{\delta\}) \setminus (\text{out}_L(l) \cup \text{out}_K(k))$ добавим в множество λ_R переходы $\langle j, o, \perp \rangle$, когда $o \neq \delta$, и переход $\langle j, \theta, \perp \rangle$, когда $o = \delta$.
- 11: **Для каждого** $o \in \text{out}_L(l) \setminus \text{out}_K(k)$ добавим в множество λ_R переходы $\langle j, o, \perp_L \rangle$, когда $o \neq \delta$, и переход $\langle j, \theta, \perp_L \rangle$, когда $o = \delta$.
- 12: **Для каждого** $o \in \text{out}_K(k) \setminus \text{out}_L(l)$ добавим в множество λ_R переходы $\langle j, o, \perp_K \rangle$, когда $o \neq \delta$, и переход $\langle j, \theta, \perp_K \rangle$, когда $o = \delta$.
- 13: **Для каждого** перехода $\langle j, o, j' \rangle$ из λ_J добавим в множество λ_R переход $\langle j, o, j' \rangle$, а в множество $J_{z'}$ — состояние j' , где $z' < z$ — такое число, что $j' \in Q_{z'}$ и $j' \in Q_{z'-1}$;
- 14: **иначе**
- 15: Выберем такое произвольное $i \in \text{in}_L(l) \cap \text{in}_K(k)$, что $\llbracket \langle l, k \rangle \rrbracket_i \subseteq Q_{z-1}$.
- 16: Добавим в множество λ_R переход $\langle j, i, j' \rangle$ из λ_J , а в множество $J_{z'}$ — состояние j' , где $z' < z$ — такое число, что $j' \in Q_{z'}$ и $j' \in Q_{z'-1}$.
- 17: $z := z - 1$.
- 18: $R := \cup_{z=0}^n J_z$.
- 19: Автомат $R_{LK} = \langle R, I, O \cup \{\theta\}, \hat{r}, \lambda_R \rangle$ — искомый автомат. **Конец.**

Условный эксперимент по различению двух полуавтоматов L и K , для которых существует **юсо**-различающий полуавтомат R_{LK} , строится следующим образом. В начальный момент полуавтомат R_{LK} находится в начальном состоянии \hat{r} . Пусть полуавтомат R_{LK} перешёл в состояние r , в котором определён переход по некоторому входному символу i . В этом случае на предъявленный для эксперимента полуавтомат подаётся входное воздействие i и вычисляется новое состояние $\{r'\} = \llbracket r \rrbracket_i$ полуавтомата R_{LK} . Если $\text{in}(r) = \emptyset$, то ожидается выходная реакция от исследуемого полуавтомата. Если полуавтомат реагирует выходным символом $o \in O$, то следующее состояние r' полуавтомата R_{LK} есть $\llbracket r \rrbracket_o$. Если реакцией исследуемого полуавтомата является молчание, то следующим состоянием полуавтомата R_{LK} является состояние из одноэлементного множества $\llbracket r \rrbracket_\theta$.

Эксперимент заканчивается, как только **юсо**-различающий полуавтомат переходит в одно из специальных состояний \perp_L , \perp_K или \perp . Если полуавтомат R_{LK} достигает состояния \perp_L , то предъявленным полуавтоматом является полуавтомат L ; если полуавтомат R_{LK} достигает состояния \perp_K , то предъявленным полуавтоматом является полуавтомат K . Если полуавтомат R_{LK} достигает состояния \perp , то предъявленный полуавтомат не является ни полуавтоматом L , ни полуавтоматом K .

Поскольку граф переходов полуавтомата R_{LK} ациклический, то условный эксперимент, описанный **іосо**-различающим полуавтоматом, является конечным.

Теорема 1. Два полуавтомата **іосо**-различимы тогда и только тогда, когда для них существует **іосо**-различающий полуавтомат.

Таким образом, два полуавтомата различимы условным экспериментом, если и только если эти автоматы являются **іосо**-различимыми.

2. Различающий условный эксперимент с недетерминированными автоматами

Под *наблюдаемым конечным автоматом* (или просто *автоматом*) с входным алфавитом I и выходным алфавитом O будем понимать пятёрку $F = \langle F, I, O, \hat{f}, \lambda_F \rangle$, где F — непустое конечное множество состояний с выделенным начальным состоянием \hat{f} ; I и O — непересекающиеся конечные алфавиты входных и выходных действий (символов) соответственно. Отношение $\lambda_F \subseteq F \times I \times O \times F$ есть отношение переходов, причём если $\langle f, i, o, f' \rangle \in \lambda_F$ и $\langle f, i, o, f'' \rangle \in \lambda_F$, то $f' = f''$. Отношение переходов естественным образом распространяется на входные и выходные последовательности, причём если $\langle f, \alpha, \beta, f' \rangle \in \lambda_F$, то пара $\langle \alpha, \beta \rangle$ называется *входо-выходной последовательностью* автомата в состоянии f , а β называется *выходной реакцией* автомата на *входную последовательность* α в состоянии f .

Введём следующие обозначения:

- 1) $\llbracket f \rrbracket_\sigma$ — множество всех состояний, достижимых из состояния f по входо-выходной последовательности σ (в наблюдаемом автомате $|\llbracket f \rrbracket_\sigma| \leq 1$);
- 2) $\mathbf{out}(f, i)$ — множество всех выходных действий, переходы по которым в паре с входным действием i определены в состоянии f ;
- 3) $\mathbf{in}(f)$ — множество всех входных символов, переходы по которым определены в состоянии f ;
- 4) Γ_F — множество всех конечных входо-выходных последовательностей в начальном состоянии автомата F .

Автомат F называется *полностью определённым*, если для каждого его состояния f справедливо $\mathbf{in}(f) = I$.

Полностью определённый автомат F называется *квазиредукцией* автомата G , если для любой входо-выходной последовательности σ из Γ_G и любого входного символа i из $\mathbf{in}(\llbracket \hat{g} \rrbracket_\sigma)$ верно следующее: $\mathbf{out}(\llbracket f \rrbracket_\sigma, i) \subseteq \mathbf{out}(\llbracket \hat{g} \rrbracket_\sigma, i)$.

Смысл отношения квазиредукции подобен смыслу отношения **іосо**: множество выходных реакций реализации на каждую допустимую входную последовательность содержится в соответствующем множестве выходных реакций спецификации.

Два автомата F и G называются *r-совместимыми* (*r-различимыми*) [10], если для них существует (соответственно не существует) общая полностью определённая квазиредукция.

Пересечением двух автоматов F и G называется автомат $H = F \cap G = \langle H, I, O, \hat{h}, \lambda_H \rangle$, где $\hat{h} = \langle \hat{f}, \hat{g} \rangle$, $H \subseteq F \times G$ — минимальное множество, полученное с использованием следующего правила для λ_H : если $\langle f, i, o, f' \rangle \in \lambda_F$ и $\langle g, i, o, g' \rangle \in \lambda_G$, то $\langle \langle f, g \rangle, i, o, \langle f', g' \rangle \rangle \in \lambda_H$.

Введём понятие *r-различающего автомата*, который будет представлять условный эксперимент по различению двух *r-различимых* автоматов.

Рассмотрим автомат $R = \langle R, I, O, \hat{r}, \lambda_R \rangle$, в котором:

- 1) множество состояний R содержит три специальных тупиковых состояния \perp_F , \perp_G и \perp ;
- 2) для всякого $r \in R \setminus \{\perp_F, \perp_G, \perp\}$ выполняется $|\mathbf{in}(r)| = 1$ и $\mathbf{out}(r, i) = O$, где $\{i\} = \mathbf{in}(r)$;
- 3) граф переходов автомата R — ациклический.

Говорят, что автомат R различает автоматы F и G , если в пересечении $H_{FR} = F \cap R$ для всех тупиковых состояний $\langle f, r \rangle$ верно $r = \perp_F$, в то время как в пересечении $H_{GR} = G \cap R$ для всех тупиковых состояний $\langle g, r \rangle$ верно $r = \perp_G$. Такой автомат R будем называть r -различающим автоматом для автоматов F и G и обозначать R_{FG} .

Алгоритм 2 строит r -различающий автомат R_{FG} для наблюдаемых недетерминированных, возможно частичных, автоматов F и G , если последние r -различимы, и выдаёт сообщение (п. 4) о том, что F и G не являются r -различимыми, в противном случае. В алгоритме используется следующее определение недоопределённых состояний из [12]. В автомате $H = F \cap G$ состояние $\langle f, g \rangle \in H$ называется 1-недоопределённым, если существует входное воздействие $i \in \mathbf{in}_F(f) \cap \mathbf{in}_G(g)$, такое, что $i \notin \mathbf{in}_H(\langle f, g \rangle)$; оно называется z -недоопределённым для $z > 1$, если оно $(z - 1)$ -недоопределённое или существует входное воздействие $i \in \mathbf{in}_F(f) \cap \mathbf{in}_G(g)$, такое, что все состояния в H , достижимые из $\langle f, g \rangle$ по входному символу i , являются $(z - 1)$ -недоопределёнными.

Алгоритм 2. Построение r -различающего автомата

Вход: Два наблюдаемых автомата $F = \langle F, I, O, \hat{f}, \lambda_F \rangle$ и $G = \langle G, I, O, \hat{g}, \lambda_G \rangle$.

Выход: R -различающий автомат R_{FG} , если F и G r -различимы.

- 1: $H_0 := \{\perp, \perp_F, \perp_G\}$; $\lambda_R := \emptyset$; $H := F \cap G$.
 - 2: Построим Q_1 — множество 1-недоопределённых состояний, Q_2 — множество 2-недоопределённых состояний, и так далее, пока $Q_j \neq Q_{j+1}$.
 - 3: **Если** $\forall j \langle \hat{f}, \hat{g} \rangle \notin Q_j$, **то**
 - 4: исходные автоматы не r -различимы. **Конец.**
 - 5: Пусть n — такое число, что $\langle \hat{f}, \hat{g} \rangle \in Q_n$, но $\langle \hat{f}, \hat{g} \rangle \notin Q_{n-1}$.
 - 6: $z := n$; $H_n := \{\langle \hat{f}, \hat{g} \rangle\}$; $H_{n-1} := \emptyset$; $H_{n-2} := \emptyset$; \dots ; $H_1 := \emptyset$.
 - 7: **Пока** ($z > 0$)
 - 8: **Для всех** ($h \equiv \langle f, g \rangle \in H_z$)
 - 9: Выберем такое произвольное $i \in \mathbf{in}_F(f) \cap \mathbf{in}_G(g)$, что для всех $o \in O$ выполняется $[[h]]_{i/o} \subseteq Q_{z-1}$.
 - 10: **Для каждого** $o \in O \setminus (\mathbf{out}_F(f, i) \cup \mathbf{out}_G(g, i))$ добавим в множество λ_R переходы $\langle h, i, o, \perp \rangle$.
 - 11: **Для каждого** $o \in \mathbf{out}_F(f, i) \setminus \mathbf{out}_G(g, i)$ добавим в множество λ_R переходы $\langle h, i, o, \perp_F \rangle$.
 - 12: **Для каждого** $o \in \mathbf{out}_G(g, i) \setminus \mathbf{out}_F(f, i)$ добавим в множество λ_R переходы $\langle h, i, o, \perp_G \rangle$.
 - 13: **Для каждого** перехода $\langle h, i, oh' \rangle$ из λ_H добавим в множество λ_R переход $\langle h, i, oh' \rangle$, а в множество $H_{z'}$ — состояние h' , где $z' < z$ — такое число, что $h' \in Q_{z'}$, но $h' \notin Q_{z'-1}$.
 - 14: $z := z - 1$.
 - 15: $R := \cup_{z=0}^n H_z$.
 - 16: Автомат $R_{FG} = \langle R, I, O, \hat{r}, \lambda_R \rangle$ — искомый автомат. **Конец.**
-

Пусть для различения предъявлен один из автоматов F или G , для которых существует r -различающий автомат R_{FG} , или какой-то другой автомат. Условный эксперимент с использованием автомата R_{FG} строится следующим образом. В начальный момент времени автомат R_{FG} находится в начальном состоянии \hat{r} . Если автомат R_{FG} находится в текущем состоянии r , в котором определён переход по входному символу $i \in I$, то на предъявленный для эксперимента автомат подаётся входной символ i . Предъявленный автомат реагирует на входной символ некоторым выходным символом $o \in O$, на основании которого вычисляется следующее состояние r -различающего автомата R_{FG} и следующий входной символ. Эксперимент заканчивается, как только r -различающий автомат переходит в одно из специальных состояний \perp_F , \perp_G или \perp . Если r -различающий автомат R_{FG} достигает состояния \perp_F , то предъявленным автоматом является автомат F ; если r -различающий автомат R_{FG} достигает состояния \perp_G , то предъявленным автоматом является автомат G . Если автомат R_{FG} достигает состояния \perp , то предъявленный автомат не является ни автоматом F , ни автоматом G . Поскольку граф переходов автомата R_{FG} ациклический, то условный эксперимент на основе r -различающего автомата является конечным.

Теорема 2. Два автомата r -различимы тогда и только тогда, когда для них существует r -различающий автомат.

Таким образом, два автомата различимы условным экспериментом, если и только если эти автоматы являются r -различимыми.

3. Различающий условный эксперимент с временными автоматами

Наблюдаемый временной автомат V (или просто *временной автомат*) есть шестёрка $\langle V, I, O, \hat{v}, \lambda_V, \Delta_V \rangle$, где V — конечное непустое множество состояний с выделенным начальным состоянием \hat{v} ; I и O — непересекающиеся конечные входной и выходной алфавиты соответственно; $\lambda_V \subseteq V \times I \times O \times V$ — отношение переходов, причём если $\langle v, i, o, v' \rangle \in \lambda_V$ и $\langle v, i, o, v'' \rangle \in \lambda_V$, то $v' = v''$; $\Delta_V : V \rightarrow V \times \mathbb{N} \cup \{\infty\}$, где \mathbb{N} — множество натуральных чисел, причём если $\Delta_V(v) = \langle v', \infty \rangle$, то $v = v'$. Как обычно, переход $\langle v, i, o, v' \rangle$ записывается как $v \xrightarrow{i/o} v'$, а равенство $\Delta_V(v) = \langle v'', t \rangle$ — как переход $v \xrightarrow{t} v''$, и этот переход будем называть *переходом по временной задержке* t .

С каждым временным автоматом V связаны часы (или таймер), отмеряющие количество (временных) тактов, прошедших после выдачи системой последнего выходного символа. Для измерения времени вводится специальная временная переменная. Предполагается, что изменения в системе происходят только по истечении некоторого целого количества тактов от начала работы системы, то есть в начальный момент времени, через один такт, через два такта и так далее. При этом, чтобы избежать неопределённости, входные воздействия разрешается подавать только в эти моменты времени. Если $v \xrightarrow{i/o} v'$, то временной автомат V , находясь в состоянии v , может принять входное воздействие i и отреагировать на него выходным действием (сигналом) o , в тот же момент временной автомат перейдёт в состояние v' и «сбросит» часы, то есть в состоянии v' время вновь начнёт отсчитываться с 0. Если $v \xrightarrow{t'} v''$ и ни одно из возможных входных воздействий не поступает на временной автомат в течение $(t' - 1)$ тактов после перехода временного автомата в состояние v , то в момент времени t' временной автомат перейдёт в состояние v'' и «сбросит» часы, то есть в состоянии v'' время начнёт отсчитываться с 0. Здесь мы предполагаем, что если $\Delta_V(v) = \langle v, \infty \rangle$, то система может оставаться в состоянии v бесконечно долго.

Другими словами, если входное воздействие i подаётся на временной автомат в состоянии v в один из моментов времени $0, 1, 2, \dots, t' - 1$, то временной автомат выдаст выходной сигнал o и мгновенно изменит своё состояние на состояние v' . Однако, если воздействие i на временной автомат в состоянии v будет подано в момент t' или позже, то временной автомат примет его, находясь уже не в состоянии v , а в состоянии v'' , поскольку в момент времени t' временной автомат, руководствуясь переходом $v \xrightarrow{t'} v''$, перейдёт в состояние v'' . При этом временной автомат «сбросит» часы, и воздействие, поданное в момент времени t' , поступит на временной автомат в состоянии v'' в момент времени 0 .

Временным входным воздействием назовём пару $\langle i, t \rangle \in I \times \mathbb{Z}_0^+$, где \mathbb{Z}_0^+ — множество целых неотрицательных чисел. Пара $\langle i, t \rangle$ означает, что входное воздействие i поступает на временной автомат в состоянии v в момент времени t , считая от момента перехода временного автомата в состояние v .

Чтобы определить выходную реакцию временного автомата в состоянии v на входное временное воздействие $\langle i, t \rangle$, необходимо определить такое состояние v' , в которое временной автомат попадёт из состояния v за время t , используя переходы по временным задержкам. Затем, согласно отношению λ_V , найти переход из состояния v' по входному воздействию i , то есть переход $v' \xrightarrow{i/o} v''$. Выходная реакция o будет выходной реакцией временного автомата в состоянии v на временное входное воздействие $\langle i, t \rangle$, а состояние v'' — конечным состоянием перехода по временному входному воздействию $\langle i, t \rangle$. Понятие переходов по временным входным воздействиям естественным образом распространяется на временные входные и выходные последовательности, причём если $v \xrightarrow{\sigma} v'$, где $\sigma \in (I \times \mathbb{Z}_0^+ \times O)^*$, то σ называется *временной входо-выходной последовательностью* временного автомата в состоянии v .

Введём следующие обозначения:

- 1) $\llbracket v \rrbracket_\sigma$ — множество всех состояний, достижимых из состояния v по временной входо-выходной последовательности σ (в наблюдаемом временном автомате $|\llbracket v \rrbracket_\sigma| \leq 1$);
- 2) $\mathbf{out}(v, i)$ — множество всех выходных символов $o \in O$, для каждого из которых существует такое $v' \in V$, что $\langle v, i, o, v' \rangle \in \lambda_V$;
- 3) $\mathbf{out}(v, \langle i, t \rangle)$ — множество всех выходных реакций временного автомата на временное входное воздействие $\langle i, t \rangle$ в состоянии v ;
- 4) $\mathbf{in}(v)$ — множество всех входных действий, переходы по которым определены в состоянии v ;
- 5) $\mathbf{in}_\chi(v)$ — множество всех временных входных действий, переходы по которым определены в состоянии v ;
- 6) Γ_V — множество всех конечных временных входо-выходных последовательностей временного автомата V в начальном состоянии.

Временной автомат V называется полностью определённым, если для каждого его состояния v справедливо $\mathbf{in}(v) = I$.

Полностью определённый временной автомат U называется *квазиредукцией* временного автомата V , если для любой временной входо-выходной последовательности σ из Γ_V и всякого временного входного воздействия $\langle i, t \rangle$ из $\mathbf{in}_\chi(\llbracket \hat{v} \rrbracket_\sigma)$ верно следующее: если $\llbracket \hat{u} \rrbracket_\sigma \neq \emptyset$, то $\mathbf{out}(\llbracket \hat{u} \rrbracket_\sigma, \langle i, t \rangle) \subseteq \mathbf{out}(\llbracket \hat{v} \rrbracket_\sigma, \langle i, t \rangle)$.

Два временных автомата называются *r-совместимыми* (*r-различимыми*), если для них существует (соответственно не существует) общая полностью определённая квазиредукция.

Для построения условного эксперимента, различающего два временных автомата, определим понятие пересечения этих автоматов. Для этого введём следующее обозначение: $\Delta'_U = \{t \in \mathbb{Z}_0^+ \mid \Delta_U(u) = \langle u', t \rangle, u, u' \in U\}$. Пересечением временных автоматов U и V назовём временной автомат $W = \langle W, I, O, \hat{w}, \lambda_W, \Delta_W \rangle$, где $\hat{w} \equiv \langle \hat{u}, 0, \hat{v}, 0 \rangle$, $K = \{0, 1, \dots, k_0\}$, $k_0 = \min(\max(\Delta'_U), \max(\Delta'_V))$ и $W \subseteq U \times K \times V \times K$ — минимальное множество, полученное согласно следующим правилам для отношения λ_W и функции Δ_W : если $\langle u, i, o, u' \rangle \in \lambda_U$, $\langle v, i, o, v' \rangle \in \lambda_V$, $\langle s_u, d_u \rangle = \Delta_U(u)$, $\langle s_v, d_v \rangle = \Delta_V(v)$, $k_1 < d_u$, $k_2 < d_v$, $k = \min(d_u - k_1, d_v - k_2)$, то $\langle \langle u, k_1, v, k_2 \rangle, i, o, \langle u', 0, v', 0 \rangle \rangle \in \lambda_W$ и

$$\Delta_W(\langle u, k_1, v, k_2 \rangle) = \begin{cases} \langle \langle s_u, 0, s_v, 0 \rangle, k \rangle, & \text{если } d_u = \infty \vee d_v = \infty \vee d_u - k_1 = d_v - k_2; \\ \langle \langle s_u, 0, v, k_2 + k \rangle, k \rangle, & \text{если } d_u \in \mathbb{Z}_0^+, d_v \in \mathbb{Z}_0^+, (d_u - k_1) < (d_v - k_2); \\ \langle \langle u, k_1 + k, s_v, 0 \rangle, k \rangle, & \text{если } d_u \in \mathbb{Z}_0^+, d_v \in \mathbb{Z}_0^+, (d_u - k_1) > (d_v - k_2). \end{cases}$$

Рассмотрим произвольный временной автомат $R = \langle R, I, O, \hat{r}, \lambda_R, \Delta_R \rangle$, в котором:

- 1) множество состояний R содержит три специальных состояния \perp_U , \perp_V и \perp , таких, что из них не определено ни одного перехода по входо-выходным парам и $\Delta_R(\perp) = \langle \perp, \infty \rangle$, $\Delta_R(\perp_U) = \langle \perp_U, \infty \rangle$, $\Delta_R(\perp_V) = \langle \perp_V, \infty \rangle$;
- 2) для всякого $r \in R \setminus \{\perp_U, \perp_V, \perp\}$ выполняется либо $|\mathbf{in}(r)| = 1$ и $\mathbf{out}(r, i) = O$, где $\{i\} = \mathbf{in}(r)$, и в этом случае $\Delta_R(r) = \langle \perp, t \rangle$, $t \in \mathbb{N}$; либо $\mathbf{in}(r) = \emptyset$, и в этом случае $\Delta_R(r) = \langle r', t \rangle$, $t \in \mathbb{N}$, $r' \notin \{\perp, \perp_U, \perp_V\}$;
- 3) граф переходов автомата R ациклический, не считая петель $\perp \xrightarrow{\infty} \perp$, $\perp_U \xrightarrow{\infty} \perp_U$ и $\perp_V \xrightarrow{\infty} \perp_V$.

Пусть построены пересечения $W_{UR} = U \cap R$ и $W_{VR} = V \cap R$. Для первого из них подмножество состояний $\{\langle u, k_u, r, k_r \rangle \in W_{UR} \mid \mathbf{in}(\langle u, k_u, r, k_r \rangle) = \emptyset\}$ обозначим W' , а для второго — подмножество состояний $\{\langle v, k_v, r, k_r \rangle \in W_{VR} \mid \mathbf{in}(\langle v, k_v, r, k_r \rangle) = \emptyset\}$ обозначим W'' . Говорят, что временной автомат R различает временные автоматы U и V , если выполняются следующие условия:

- 1) в подмножестве W' найдутся состояния, для которых $r = \perp_U$;
- 2) для всех состояний из W' верно $r \neq \perp_V$;
- 3) если для какого-то состояния из W' верно $\Delta_W(\langle u, k_u, r, k_r \rangle) = \langle \langle u', k'_u, \perp, k'_\perp \rangle, t \rangle$ для некоторого t , то $r = \perp$;
- 4) все переходы, ведущие в состояния вида $\langle u, k_u, \perp, k_\perp \rangle$ пересечения W_{UR} , помечены только временными задержками;
- 5) в подмножестве W'' найдутся состояния, для которых $r = \perp_V$;
- 6) для всех состояний из W'' верно $r \neq \perp_U$;
- 7) если для какого-то состояния из W'' верно $\Delta_W(\langle v, k_v, r, k_r \rangle) = \langle \langle v', k'_v, \perp, k'_\perp \rangle, t \rangle$ для некоторого t , то $r = \perp$;
- 8) все переходы, ведущие в состояния вида $\langle v, k_v, \perp, k_\perp \rangle$ пересечения W_{VR} , помечены только временными задержками.

Такой временной автомат R будем называть r -различающим временным автоматом для временных автоматов U и V и обозначать R_{UV} .

Алгоритм 3 описывает построение r -различающего временного автомата по пересечению двух временных автоматов с использованием следующего понятия недоопределённых состояний.

Состояние $w = \langle u, k_u, v, k_v \rangle$ пересечения $W = U \cap V$ называется 1-недоопределённым, если найдётся такой входной символ i из $\mathbf{in}_U(u') \cap \mathbf{in}_V(v')$, по которому нет ни одного перехода из состояния w . Кроме того, состояние называется 1-недоопределённым,

если из него можно перейти в другое 1-недоопределённое состояние по временной задержке. Состояние w пересечения W называется z -недоопределённым, $z > 1$, если оно $(z - 1)$ -недоопределённое или найдётся такой входной символ $i \in \mathbf{in}(\langle u, k_u, v, k_v \rangle)$, что все состояния, достижимые из w по входному символу i , являются $(z - 1)$ -недоопределёнными. Кроме того, состояние w будет z -недоопределённым и в случае, когда из него по временной задержке достижимо некоторое z -недоопределённое состояние.

Алгоритм 3. Построение r -различающего временного автомата

Вход: Два наблюдаемых временных автомата $U = \langle U, I, O, \hat{u}, \lambda_U, \Delta_U \rangle$ и $V = \langle V, I, O, \hat{v}, \lambda_V, \Delta_V \rangle$.

Выход: R -различающий временной автомат R_{UV} , если U и V r -различимы.

- 1: $W_0 := \{\perp, \perp_U, \perp_V\}$; $\lambda_R := \emptyset$; $W := U \cap V$.
 - 2: Построим Q_1 — множество 1-недоопределённых состояний, Q_2 — множество 2-недоопределённых состояний, и так далее, пока $Q_j \neq Q_{j+1}$.
 - 3: **Если** $\forall j \langle \hat{u}, 0, \hat{v}, 0 \rangle \notin Q_j$, **то**
 - 4: исходные временные автоматы r -совместимы. **Конец.**
 - 5: Пусть n — такое число, что $\langle \hat{u}, 0, \hat{v}, 0 \rangle \in Q_n$, но $\langle \hat{u}, 0, \hat{v}, 0 \rangle \notin Q_{n-1}$.
 - 6: $z := n$; $W_n := \{\langle \hat{u}, 0, \hat{v}, 0 \rangle\}$; $W_{n-1} := \emptyset$; $W_{n-2} := \emptyset$; \dots ; $W_1 := \emptyset$.
 - 7: **Пока** ($z > 0$)
 - 8: **Для всех** ($w \equiv \langle u, k_u, v, k_v \rangle \in W_z$)
 - 9: **Если** для всех $i \in \mathbf{in}_U(u) \cap \mathbf{in}_V(v)$ и всех $o \in O$ выполняется $\llbracket w \rrbracket_{i/o} \not\subseteq Q_{z-1}$, **то**
 - 10: Положим $\Delta_R(w) := \langle w', t \rangle$, где $\langle w', t \rangle = \Delta_W(w)$.
 - 11: Добавим состояние w' в множество W_z ;
 - 12: **иначе**
 - 13: Выберем такое произвольное $i \in \mathbf{in}_U(u) \cap \mathbf{in}_V(v)$, что для всех $o \in O$ выполняется $\llbracket w \rrbracket_{i/o} \subseteq Q_{z-1}$.
 - 14: **Для каждого** $o \in O \setminus (\mathbf{out}_U(u, i) \cup \mathbf{out}_V(v, i))$ добавим в множество λ_R переходы $\langle w, i, o, \perp \rangle$.
 - 15: **Для каждого** $o \in \mathbf{out}_U(u, i) \setminus \mathbf{out}_V(v, i)$ добавим в множество λ_R переходы $\langle w, i, o, \perp_U \rangle$.
 - 16: **Для каждого** $o \in \mathbf{out}_V(v, i) \setminus \mathbf{out}_U(u, i)$ добавим в множество λ_R переходы $\langle w, i, o, \perp_V \rangle$.
 - 17: **Для каждого** перехода $\langle w, i, o, w' \rangle$ из λ_W добавим в множество λ_R переход $\langle w, i, o, w' \rangle$, а в множество $W_{z'}$ — состояние w' , где $z' < z$ — такое число, что $w' \in Q_{z'}$, но $w' \notin Q_{z'-1}$.
 - 18: **Если** $\Delta_W(w) = \langle w, \infty \rangle$, **то** положим $\Delta_R(w) := \langle \perp, 1 \rangle$, **иначе если** $\Delta_W(w) = \langle w'', t \rangle \in W \times \mathbb{N}$, **то** положим $\Delta_R(w) := \langle \perp, t \rangle$.
 - 19: $z := z - 1$.
 - 20: $R := \cup_{z=0}^n W_z$.
 - 21: Временной автомат $R_{UV} = \langle R, I, O, \hat{r}, \lambda_R, \Delta_R \rangle$ — искомый временной автомат.
- Конец.**
-

Пусть для различения предъявлен один из временных автоматов U или V , для которых существует r -различающий временной автомат R_{UV} . Условный эксперимент с использованием автомата R_{UV} строится следующим образом. В начальный момент значение переменной t равно 0, а R_{UV} находится в начальном состоянии \hat{r} . Пусть $t = D$

и временной автомат R_{UV} находится в состоянии r , в котором определён переход по входному символу $i \in I$. В этом случае на предъявленный для эксперимента временной автомат подаётся входной символ i в момент времени t' , $D \leq t' < D + d_r$, где d_r — значение временной задержки в состоянии r (время отсчитывается от момента получения последней выходной реакции предъявленного для эксперимента временного автомата). Предъявленный временной автомат реагирует на входной символ некоторым выходным символом o , на основании которого вычисляется следующее состояние r -различающего временного автомата R_{UV} , значение переменной t обнуляется. Если в текущем состоянии r -различающего временного автомата не определён переход по любому входному символу, то к значению D переменной t прибавляется величина временной задержки в текущем состоянии, и следующее состояние r -различающего временного автомата R_{UV} вычисляется по значению $\Delta_R(r)$. Эксперимент заканчивается, как только r -различающий временной автомат переходит в одно из специальных состояний \perp_U или \perp_V . Если r -различающий временной автомат R_{UV} достигает состояния \perp_U , то предъявленным временным автоматом является временной автомат U ; если r -различающий временной автомат R_{UV} достигает состояния \perp_V , то предъявленным временным автоматом является временной автомат V .

Поскольку граф переходов временного автомата R_{UV} ациклический (за исключением петель с пометками ∞) и в каждом состоянии, в котором не определён переход по любому входному символу, задержка является конечной, то условный эксперимент, описанный r -различающим временным автоматом, является конечным.

Теорема 3. Два временных автомата r -различимы тогда и только тогда, когда для них существует r -различающий временной автомат.

Таким образом, два временных автомата различимы условным экспериментом, если и только если они r -различимы.

Заключение

В данной работе установлены необходимые и достаточные условия существования условных экспериментов, различающих два входо-выходных полуавтомата, два недетерминированных, возможно частичных, конечных автомата или два недетерминированных временных автомата. Они сводятся к существованию надлежащих различающих автоматов. Приведены алгоритмы построения последних (в случае их существования) и условных различающих экспериментов на их основе. Полученные результаты могут быть использованы как при синтезе условных (адаптивных) тестов с гарантированной полнотой для дискретных управляющих систем, так и для диагностики неисправностей в таких системах.

ЛИТЕРАТУРА

1. *Milner R.* Communication and concurrency Upper Saddle River: Prentice-Hall, Inc., 1989.
2. *Hierons R. M.* Testing from a Nondeterministic Finite State Machine Using Adaptive State Counting // IEEE Trans. Comput. 2004. V. 53. No. 10. P. 1330–1342.
3. *Simão A., Petrenko A.* Generating Checking Sequences for Partial Reduced Finite State Machines // TestCom '08 / FATES '08. Berlin: Springer Verlag, 2008. P. 153–168.
4. *Shabdina N., El-Fakih K., Yevtushenko N.* Testing Nondeterministic Finite State Machines with Respect to the Separability Relation // Testing of Software and Communicating Systems. Berlin: Springer, 2007. P. 305–318.
5. *Merayo M., Núñez M., Rodríguez I.* Formal Testing from Timed Finite State Machines // Computer Networks. 2008. V. 52. No. 3. P. 432–460.

6. *Alur R., Courcoubetis C., Yannakakis M.* Distinguishing tests for nondeterministic and probabilistic machines // STOC'95. New York: ACM, 1995. P. 363–372.
7. *Gromov M., Willemse T.* Model-Based Testing Techniques for Diagnosis // Testing of Software and Communicating Systems. Berlin: Springer, 2007. P. 138–154.
8. *Gill A.* Introduction to the theory of Finite-State Machines. New York: McGraw-Hill, 1962.
9. *Tretmans J.* Test Generation with Inputs, Outputs and Repetitive Quiescence // Software—Concepts and Tools. 1996. V. 17. No. 3. P. 103–120.
10. *Евтушенко Н. В., Петренко А. Ф., Ветрова М. В.* Недетерминированные автоматы: Анализ и синтез. Ч. 1. Отношения и операции: учеб. пособие. Томск: Томский госуниверситет, 2006. 142 с.
11. *Gromov M., Popov D., Yevtushenko N.* Deriving test suites for timed Finite State Machines // Proceedings of IEEE East-West Design & Test Symposium'08, Kharkov: SPD FL Stepanov V. V., 2008. P. 339–343.
12. *Куфарева И. Б.* Применение недетерминированных автоматов в задачах синтеза проверяющих тестов для систем логического управления: автореф. дис. ... канд. техн. наук. Томск: Томский госуниверситет, 2000. 16 с.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

DOI 10.17223/20710410/6/10

УДК 519.7; 519.81

МОДИФИКАЦИЯ МЕТОДА АНАЛИЗА ИЕРАРХИЙ ДЛЯ ДИНАМИЧЕСКИХ НАБОРОВ АЛЬТЕРНАТИВ¹

С. И. Колесникова

*Томский государственный университет систем управления и радиоэлектроники, г. Томск,
Россия*

E-mail: skolesnikova@yandex.ru

Рассматривается проблема некорректного оценивания динамических наборов альтернатив в методе анализа иерархий, используемом в интеллектуальных системах поддержки принятия решения. Излагается модифицированная процедура корректного определения значимости альтернатив, доказываются ее свойства и приводятся иллюстративные примеры.

Ключевые слова: метод анализа иерархий, весовые коэффициенты альтернатив, принятие решений.

1. Введение в проблему

По ориентировочным оценкам число статей прикладного характера, в которых метод анализа иерархий (МАИ) [1] применяется к решению прикладных многокритериальных задач, еще 10 лет назад составляло более тысячи [2]. Широкому применению метода способствует и разработанный пакет зарубежных программ EXPERT CHOICE, реализующий МАИ. Однако, как известно, недостатком классического МАИ [1], отмеченным и самим автором метода, является противоречие, связанное с эффектом единичной нормировки, приводящей к тому, что предпочтения, выявленные на всем множестве альтернатив (признаков, объектов и пр.), могут не совпадать с «частными» предпочтениями на подмножестве альтернатив. Содержательно на примере это означает следующее. При оценивании двух проектов z_1 и z_2 несколькими экспертами по методу МАИ устанавливается, что z_1 предпочтительней z_2 ($z_1 \succ z_2$), знак « \succ » означает факт предпочтительности (доминирования). При появлении третьего проекта интеллектуальная экспертная система (на основе МАИ) их ранжирует по ценности заново, и в результате возможна ситуация: z_2 предпочтительней z_1 ($z_2 \succ z_1$).

В ряде работ (например, [3, 4]) приведены примеры нарушения предпочтений между альтернативами (в [4] — признаками) при включении в группу (удалении из группы) сравниваемых альтернатив (признаков) дополнительной альтернативы и описана конструктивная идея целесообразности оценивать относительную значимость каждой альтернативы по отношению ко всей совокупности альтернатив. В данной работе выясняются условия, при которых бинарное отношение (выражающее предпочтение

¹Работа частично поддержана РФФИ, проект №09-01-99014. Результаты работы докладывались на Международной конференции с элементами научной школы для молодежи, г. Омск, 7–12 сентября 2009 г.

альтернатив) сохраняется на измененном множестве альтернатив, излагается процедура модифицированного метода анализа иерархий (ММАИ), доказываются свойства ММАИ. Тезисное изложение результатов этой работы приведено в [5].

1.1. Основные понятия и определения

В общем виде постановка задачи, решаемой МАИ [1], включает цель, альтернативы и критерии оценки альтернатив; требуется выбрать наилучшую альтернативу. После построения иерархической структуры (цели — критерии — альтернативы) система парных сравнений элементов каждого уровня приводит к результату, который может быть представлен в виде обратно симметричной матрицы (матрицы парных сравнений альтернатив), элемент которой a_{ij} есть интенсивность проявления альтернативы (как элемента иерархии) i относительно альтернативы j в смысле выбранного фиксированного критерия. Главные собственные векторы матрицы A парных сравнений (МПС) интерпретируются как векторы приоритетов сравниваемых элементов. Допускается вычислять коэффициенты важности для каждой из альтернатив в виде среднего геометрического элементов соответствующей строки МПС в случае, когда оценки элементов заданы в виде отношения (мультипликативный метод анализа иерархий).

Обозначим совокупность альтернатив $\Theta = \{z_1, z_2, \dots, z_g\}$. Прежде чем сформулировать и доказать результат, в котором выясняются условия, при которых бинарное отношение (выражающее предпочтение альтернатив) $z_i \succ z_j$ ($z_j \succ z_i$) сохраняется на множествах $\Theta' = \{z_1, z_2, \dots, z_{g-1}\}$ ($\Theta' = \{z_1, z_2, \dots, z_{g+1}\}$), напомним требования к матрице относительных весов [1] $A = \|a_{ij}\|_{g \times g}$, $a_{ij} = w_i/w_j$, где w_i, w_j — компоненты весового вектора альтернатив $W = \{w_1, w_2, \dots, w_g\}^T$, g — количество сравниваемых альтернатив: 1) $a_{ij} \geq 0$; 2) $a_{ij} = a_{ji}^{-1}$; 3) $a_{ij} = a_{ik}a_{kj}$; 4) число g является максимальным собственным значением матрицы A , и для некоторого единственного (нормированного) вектор-столбца $W = \{w_1, w_2, \dots, w_g\}^T$ с положительными компонентами выполняется равенство: $A \cdot W = g \cdot W$.

Приведем пример негативной стороны стандартного применения МАИ из работы [3], способствующей неточности в принятии решения.

Пример 1. Пусть оцениваются две альтернативы (два признака) z_1, z_2 по двум равновесным мерам относительной важности со следующими МПС:

$$A_1 = \begin{bmatrix} 1 & 1/2 \\ 2 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 4 \\ 1/4 & 1 \end{bmatrix}.$$

В результате стандартного применения МАИ получим для альтернатив z_1, z_2 вектор весовых коэффициентов (оценок) альтернатив (ВКА) $w = (w_1, w_2) = (0,567, 0,433)$. Так как $w_1 > w_2$, то $z_1 \succ z_2$. При оценивании трех альтернатив z_1, z_2, z_3 по тем же двум мерам относительной важности со следующими МПС:

$$A_1 = \begin{bmatrix} 1 & 1/2 & 3 \\ 2 & 1 & 7 \\ 1/3 & 1/7 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 4 & 1/3 \\ 1/4 & 1 & 1/5 \\ 3 & 5 & 1 \end{bmatrix}$$

— после стандартного применения МАИ и последующей нормализации получаем $w = (w_1, w_2, w_3) = (0,286, 0,354, 0,360)$, т.е. $z_2 \succ z_1$.

2. Постановка задачи

Обозначим через ρ бинарное отношение предпочтения на множестве альтернатив. Пусть заданы множества (наборы) альтернатив $\Theta_1 = \{z_1, z_2, \dots, z_{g-1}\}$ и $\Theta_2 =$

$= \{z_1, z_2, \dots, z_{g-1}, z_g\}$. Требуется построить процедуру оценки значимости альтернатив по совокупности заданных критериев (назовем ее модифицированным МАИ), сохраняющей бинарные отношения $z_i \rho_1 z_j, z_i \rho_2 z_j, z_i, z_j \in \Theta_1 \subseteq \Theta_2, i \neq j$, индуцированные на множествах Θ_1 и Θ_2 применением МАИ.

3. Решение задачи

Нежелательную особенность стандартного применения МАИ в оценивании динамических наборов альтернатив, способствующей неточности в принятии решения в интеллектуальных системах, выразим в виде двух теорем.

Теорема 1. Бинарные отношения (предпочтения) $z_i \rho_1 z_j, z_i \rho_2 z_j$, где $z_i, z_j \in \Theta_1 \subseteq \Theta_2$ ($i, j \in \{1, 2, \dots, g\}, i \neq j$), индуцированные на множествах Θ_1, Θ_2 посредством применения стандартной процедуры МАИ, в общем случае не совпадают.

Доказательство. Обозначим МПС альтернатив множества Θ_1 через $A_1 = \|a_{ij}\|_{(g-1) \times (g-1)}$, где $a_{ij} = \frac{w_i}{w_j}$, и МПС альтернатив множества Θ_2 через $A_2 = \|a'_{ij}\|_{g \times g}$, где $a'_{ij} = \frac{w'_i}{w'_j}$. В качестве приближения собственных векторов МПС A_1 и A_2 в МАИ используют векторы $W_1 = \{w_1, w_2, \dots, w_{g-1}\}^T, W_2 = \{w'_1, w'_2, \dots, w'_g\}^T$, компоненты которых вычисляются как среднее геометрическое по элементам строк, т. е. $w_i = \left(\prod_{l=1}^{g-1} a_{il}\right)^{\frac{1}{g-1}}$ и $w'_i = \left(\prod_{l=1}^g a'_{il}\right)^{\frac{1}{g}}$ соответственно.

Пусть после применения МАИ весовые оценки альтернатив таковы, что $w_i > w_j$, то есть на множестве Θ_1 задано отношение предпочтения $z_i \succ z_j$. Покажем, что это отношение предпочтения в общем случае не сохраняется на множестве Θ_2 .

Для сохранения отношения $z_i \succ z_j$ на множестве Θ_2 необходимо должно выполняться неравенство $w'_i > w'_j$, или $\left(\prod_{l=1}^g a'_{il}\right)^{\frac{1}{g}} > \left(\prod_{l=1}^g a'_{jl}\right)^{\frac{1}{g}}$. Это эквивалентно требованию $D_{i,g-1} a'_{ig} > D_{j,g-1} a'_{jg}$, где $D_{s,g-1} = \prod_{l=1}^{g-1} a'_{sl}, s \in \{i, j\}$.

Преобразуем обе части неравенства $\frac{a'_{ig}}{a'_{jg}} > \frac{D_{j,g-1}}{D_{i,g-1}}$, осуществив соответствующие замены с применением свойств 2 и 3 МПС:

$$\frac{a'_{ig}}{a'_{jg}} = \frac{a'_{ij} a'_{jg}}{a'_{jg}} = a'_{ij}, \quad \frac{D_{j,g-1}}{D_{i,g-1}} = \prod_{l=1}^{g-1} \frac{a'_{jl}}{a'_{il}} = \prod_{l=1}^{g-1} \frac{a'_{ji} a'_{il}}{a'_{il}} = \prod_{l=1}^{g-1} \frac{1}{a'_{ij}} = (a'_{ij})^{1-g}.$$

Таким образом, для справедливости требуемого неравенства необходимо должно выполняться соотношение $a'_{ij} > (a'_{ij})^{1-g}$, или $(a'_{ij})^g > 1, i, j < g$, что, очевидно, будет иметь место только в частных случаях. ■

Теорема 2. Пусть для МПС A_1 и A_2 выполнены свойства 1–4 МПС. Тогда при выполнении условий: 1) $a'_{ij} = a_{ij} (i, j < g)$ и 2) $(a'_{ij})^g > 1$ ($(a'_{ij})^g < 1$) бинарные отношения (предпочтения) $z_i \succ z_j (z_j \succ z_i)$, индуцированные на множествах Θ_1 и Θ_2 посредством применения стандартной процедуры метода анализа иерархий, совпадают.

Доказательство теоремы 2 опирается на свойства 1–4 МПС и результат теоремы 1.

3.1. Модифицированный метод анализа иерархий

Изложим модифицированную процедуру определения ВКА, опираясь на [1, 3, 4].

1. Введем весовые коэффициенты мер относительной важности признаков, обозначенные через c_s , $\sum_{s=1}^{\nu} c_s = 1$.

Построим МПС на каждом из этапов МАИ (по числу мер относительной важности альтернатив). Результатом каждого s -го этапа ($s \in \{1, 2, \dots, \nu\}$) является g -компонентный вектор нормализованных значений ВКА $W_s = \{w_1^s, w_2^s, \dots, w_g^s\}^T$.

2. Сформируем всевозможные векторы $w_{ij}^s = (w_{ij}^{s1}, w_{ij}^{s2})$ локальных ВКА уровня 1 по формулам

$$w_{ij}^{s1} = \frac{w_i^s}{w_i^s + w_j^s}; \quad w_{ij}^{s2} = \frac{w_j^s}{w_i^s + w_j^s}, \quad s \in \{1, 2, \dots, \nu\}, i, j \in \{1, 2, \dots, g\}.$$

С содержательной точки зрения, первая компонента w_{ij}^{s1} вектора $w_{ij}^s = (w_{ij}^{s1}, w_{ij}^{s2})$ означает относительный вес альтернативы z_i по отношению к суммарному весу альтернатив z_i и z_j ; вторая компонента w_{ij}^{s2} вектора w_{ij}^s означает относительный вес альтернативы z_j .

3. Сформируем матрицу $W = \|w_{ij}\|$, где векторы $w_{ij} = (w_{ij}^i, w_{ij}^j)$ — локальные ВКА (уровня 2) альтернатив z_i, z_j относительно всей совокупности мер относительной важности признаков; компоненты векторов находим по формулам

$$w_{ij}^i = \frac{u_{ij}^i}{u_{ij}^i + u_{ij}^j}; \quad w_{ij}^j = \frac{u_{ij}^j}{u_{ij}^i + u_{ij}^j}, \quad \text{где } u_{ij}^i = \sum_{s=1}^{\nu} c_s w_{ij}^{s1}, \quad u_{ij}^j = \sum_{s=1}^{\nu} c_s w_{ij}^{s2}.$$

4. Вычислим глобальные значения ВКА по одной из формул:

$$V_i^{(1)} = \sum_{j=1}^g w_{ij}^i, \quad V_i^{(2)} = \left(\prod_{j=1}^g w_{ij}^i \right)^{1/g}, \quad i \in 1, 2, \dots, g.$$

Отметим, что в соответствии с модифицированной процедурой для исходных данных примера 1 получаем следующие ВКА: $w = (w_1, w_2, w_3) = (0,349, 0,325, 0,326)$. Таким образом, предпочтения между альтернативами z_1, z_2 не изменились: $z_1 \succ z_2$.

Процедура определения ВКА, несмотря на простоту, нуждается в численной иллюстрации.

3.2. Иллюстративный пример применения модифицированного метода анализа иерархий

1. Пусть МПС трех альтернатив $z_i, i = 1, 2, 3$, составлена по каждому из двух данных равновесных критериев ($c_1 = c_2 = 0,5$) и посчитаны весовые коэффициенты альтернатив. Последние два столбца в нижеприведенных расширенных МПС представляют собой собственные векторы W^1 (W^2) и нормализованные собственные векторы W_n^1 (W_n^2) МПС альтернатив по 1-му критерию (вычисления проведены с точностью до трёх знаков):

	z_1	z_2	z_3	W^1	W_n^1
z_1	1,000	0,500	4,000	1,260	0,308
z_2	2,000	1,000	8,000	2,520	0,615
z_3	0,250	0,125	1,000	0,315	0,077

и МПС альтернатив по 2-му критерию:

$$\begin{array}{c} z_1 \quad z_2 \quad z_3 \quad W^2 \quad W_n^2 \\ z_1 \left(\begin{array}{ccccc} 1,000 & 4,000 & 0,500 & 1,260 & 0,323 \\ 0,250 & 1,000 & 0,167 & 0,347 & 0,089 \\ 2,000 & 6,000 & 1,000 & 2,289 & 0,588 \end{array} \right). \end{array}$$

2. Формируем обобщенные матрицы относительных локальных ВКА по нижеприведенной схеме.

2.1. Формируем матрицу относительных ВКА по 1-му критерию ($w_{ij}^1 = w_i^1, w_j^1$):

$$\left(\begin{array}{ccc} (1,000; 1,000) & (0,308; 0,615) & (0,308; 0,077) \\ (0,615; 0,308) & (1,000; 1,000) & (0,615; 0,077) \\ (0,077; 0,308) & (0,077; 0,615) & (1,000; 1,000) \end{array} \right).$$

2.2. Нормализуем локальные ВКА по 1-му критерию, например,

$$w_{12}^1 = \left(\frac{0,308}{0,308 + 0,615}; \frac{0,615}{0,308 + 0,615} \right);$$

$$\left(\begin{array}{ccc} (0,500; 0,500) & (0,333; 0,667) & (0,800; 0,200) \\ (0,667; 0,333) & (0,500; 0,500) & (0,889; 0,111) \\ (0,200; 0,800) & (0,111; 0,889) & (0,500; 0,500) \end{array} \right).$$

2.3. Формируем всевозможные векторы локальных ВКА по 2-му критерию ($w_{ij}^2 = w_i^2, w_j^2$):

$$\left(\begin{array}{ccc} (1,000; 1,000) & (0,323; 0,089) & (0,323; 0,588) \\ (0,089; 0,323) & (1,000; 1,000) & (0,089; 0,588) \\ (0,588; 0,323) & (0,588; 0,089) & (1,000; 1,000) \end{array} \right).$$

2.4. Нормализуем локальные ВКА по 2-му критерию аналогично п. 2.2:

$$\left(\begin{array}{ccc} (0,500; 0,500) & (0,784; 0,216) & (0,355; 0,645) \\ (0,216; 0,784) & (0,500; 0,500) & (0,132; 0,868) \\ (0,645; 0,355) & (0,868; 0,132) & (0,500; 0,500) \end{array} \right).$$

3. Формируем обобщенную матрицу $W = \|w_{ij}\| = \|(w_{ij}^i, w_{ij}^j)\|$, компонентами которой являются векторы — локальные ВКА альтернатив относительно всей заданной совокупности критериев. Компоненты локальных ВКА находим по формулам п. 3 вышеприведенной процедуры ММАИ, например:

$$(u_{12}^1; u_{12}^2) = \left(\frac{0,333 + 0,784}{2}; \frac{0,667 + 0,216}{2} \right); w_{12} = (w_{12}^1; w_{12}^2) = (0,559; 0,441).$$

В итоге получаем обобщенную матрицу вида

$$\left(\begin{array}{ccc} (0,500; 0,500) & (0,542; 0,458) & (0,567; 0,433) \\ (0,458; 0,542) & (0,500; 0,500) & (0,515; 0,485) \\ (0,433; 0,567) & (0,485; 0,515) & (0,500; 0,500) \end{array} \right).$$

4. Глобальные значения ВКА определяем как линейную свертку [1]:

$$V_1^{(1)} = \sum_{j=1}^3 w_{1j} = 0,500 + 0,542 + 0,567; \quad V_2^{(1)} = \sum_{j=1}^3 w_{2j} = 0,458 + 0,500 + 0,515;$$

$$V_3^{(1)} = \sum_{j=1}^3 w_{3j} = 0,433 + 0,485 + 0,500.$$

Нормализованный итоговый вектор оценок значимости альтернатив имеет вид $V^{(1)} = (0,358; 0,327; 0,315)$.

Свойство данной процедуры сформулировано в теореме 3.

Теорема 3. Бинарные отношения (предпочтения) $z_i \rho_1 z_j$, $z_i \rho_2 z_j$, $z_i, z_j \in \Theta_1 \subseteq \Theta_2, i \neq j$, индуцированные посредством применения стандартной процедуры МАИ на множестве Θ_1 и модифицированного метода анализа иерархий на множестве Θ_2 , совпадают.

Доказательство.

При выполнении условия 1 теоремы 2 справедлива полезная формула, связывающая ВКА двух множеств (по каждому критерию) Θ_1 и Θ_2 : $w'_i = (w_i)^{\frac{g-1}{g}} (a'_{ig})^{\frac{1}{g}}, i < g$. Действительно, использование представления компонент весового вектора альтернатив в мультипликативном МАИ как среднего геометрического по строкам МПС приводит к справедливости соотношений

$$w'_i = \left(\prod_{l=1}^g a'_{il} \right)^{\frac{1}{g}} = \left(\prod_{l=1}^{g-1} a'_{il} \right)^{\frac{1}{g}} (a'_{ig})^{\frac{1}{g}} = \left(\prod_{l=1}^{g-1} a_{il} \right)^{\frac{1}{g}} (a'_{ig})^{\frac{1}{g}} = w_i^{\frac{g-1}{g}} (a'_{ig})^{\frac{1}{g}}.$$

Пусть для определенности на множестве Θ_1 задано отношение предпочтения $z_i \succ z_j$, то есть $w_i > w_j$. Требуется показать, что это отношение предпочтения сохранится на множестве Θ_2 при оценивании весовых коэффициентов альтернатив по модифицированному МАИ, т. е. необходимо должно выполняться неравенство $w'_i > w'_j$ ($i, j < g$).

Предположим сначала выполнение условия 1 теоремы 2. Применяя полученную зависимость, вышеприведенные формулы для ВКА по методу ММАИ, свойства 1–4 МПС (в частности, свойство 3 — транзитивность МПС), получаем

$$w'_i - w'_j = w_i^{\frac{g-1}{g}} (a'_{ig})^{\frac{1}{g}} - w_j^{\frac{g-1}{g}} (a'_{jg})^{\frac{1}{g}} = w_i^{\frac{g-1}{g}} (a'_{ij} a'_{jg})^{\frac{1}{g}} - w_j^{\frac{g-1}{g}} (a'_{jg})^{\frac{1}{g}} =$$

$$= (a'_{ig})^{\frac{1}{g}} \left[w_i^{\frac{g-1}{g}} (a'_{ij})^{\frac{1}{g}} - w_j^{\frac{g-1}{g}} \right] = (a'_{ig})^{\frac{1}{g}} \left[w_i^{\frac{g-1}{g}} (a_{ij})^{\frac{1}{g}} - w_j^{\frac{g-1}{g}} \right] \geq 0.$$

Справедливость последнего неравенства обеспечивается предположением $a_{ij} = \frac{w_i}{w_j}$ и $w_i > w_j$.

В общем случае используются непосредственно формулы для вычисления ВКА по методу ММАИ:

$$w_i = \frac{V_i}{\sum_{j=1}^g V_j}, \quad \text{где } V_i = \sum_{l=1}^g w_{il}^i = \sum_{l=1}^g \frac{u_{il}^i}{u_{il}^i + u_{il}^j}, \quad u_{il}^i = \sum_{s=1}^{\nu} c_s \frac{w_i^s}{w_i^s + w_l^s},$$

величины w_i^s, w_j^s являются компонентами собственного вектора МПС альтернатив на s -м уровне иерархии (по s -му критерию). Осуществляя последовательно несложные, но громоздкие преобразования, можно показать, что и в общем случае $w_i' \geq w_j'$. ■

Замечание 1. Прокомментируем условие $a'_{ij} = a_{ij}$ ($i, j < g$) (условие 1 теоремы 2). Данное условие означает, что при добавлении новой альтернативы в исследуемую группу альтернатив предпочтения относительно «старых» альтернатив не изменились, т. е. выполнено условие независимости совокупности альтернатив по предпочтению, а, как известно [6], этот факт является обоснованием использования линейной свертки для агрегирования многокритериальных решений (в том числе и МАИ).

Замечание 2. В практических задачах «групповой эффект» и «человеческий фактор» при экспертном оценивании альтернатив (признаков) нередко приводит к нарушению свойства 3 МПС, которое, в свою очередь, приводит к некорректному использованию МАИ, так как в этом случае собственный вектор такой (уже несовместной) МПС соответствует максимальному собственному значению, которое строго больше g , а не равно g (см. свойство 4 МПС). Следует отметить, что доказательства теорем 1–3 существенно опираются на свойства МПС, в частности на свойства 2 и 3, т. е. эти условия (поскольку они являются условиями применения МАИ) считаются по умолчанию выполненными.

4. Применение модифицированного МАИ в решении прикладной задачи

Основой одного из наиболее эффективных подходов к созданию интеллектуальных систем являются тестовые методы распознавания образов [7], использующие для принятия решений наборы (тесты), содержащие меньшее количество признаков и имеющие больший вес (под весом теста понимается сумма весовых коэффициентов признаков). Один из методов определения «весов» сравниваемых признаков предложен в работе [8]. Метод учитывает вклад признаков в распознающую способность теста с учетом их взаимозависимости и базируется на представлении совокупности всех различных пар объектов из разных классов (образов) для каждого признака в виде мультимножества [9] и применении МАИ с использованием парных сравнений признаков на основе специальным образом выбранных мер относительной важности признаков, учитывающих их особенности. Приведем пример [7], связанный с целесообразностью изменения состава тестового набора (удаления признака).

Пример 2. Пусть матрица описаний Q содержит описание шести объектов (строки) по четырём признакам (z_1, z_2, z_3, z_4) (столбцы), в матрице различений R указывается на соответствие номеров объектов (1-й столбец) и классов (2-й столбец), которым объекты принадлежат:

$$Q = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 2 \\ 1 & 1 & 1 & 2 \end{pmatrix}; R = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 2 \\ 5 & 2 \\ 6 & 2 \end{pmatrix}.$$

Для данного примера тупиковыми тестами являются наборы признаков $\tau_1 = (z_1, z_2, z_3)$, $\tau_2 = (z_1, z_2, z_4)$ и $\tau_3 = (z_2, z_3, z_4)$. Если использовать тестовый алгоритм непосредственно, то объект $S = (0, 1, 2, 1)$ не будет отнесен ни к одному из классов, однако фрагмент $(0, 1)$, порождаемый набором $\tau = (z_1, z_2)$, содержится в S и

соответствующих объектах из первого класса и не содержится в объектах из второго класса, что дает основание полагать, что распознаваемый объект более близок к первому классу. Непосредственное применение классического МАИ в методе из работы [8] может привести к ошибочному определению весовых коэффициентов признаков, а следовательно, и к неточности в принятии решения. Модифицированный МАИ корректно «взвесит» признаки в изменившемся (удалением признака из теста) наборе.

Заклучение

Дальнейшее развитие МАИ (и модифицированного МАИ) должно быть связано с выяснением условий, связанных с правомочностью применения линейной свертки при оценивании ВКА (шаг 4 вышеизложенного алгоритма), поскольку данный способ агрегирования приемлем при весьма ограничительных предположениях и может приводить к неточности в принятии решений (см., например, работу [2]).

ЛИТЕРАТУРА

1. Саати Т. Л. Принятие решений. Метод анализа иерархий. М.: Радио и связь, 1989. 311с.
2. Ногин В. Д. Упрощенный вариант метода анализа иерархий на основе нелинейной свертки критериев // Журн. вычислит. матем. и математич. физ. 2004. Т. 44. № 7. С. 1259–1268.
3. Самохвалов Ю. Я. Групповой учет относительного превосходства альтернатив в задачах принятия решений // Кибернетика и системный анализ. 2003. № 6. С. 141–145.
4. Колесникова С. И. Системный подход к оцениванию взаимного влияния признаков в тестовом распознавании // Кибернетика и системный анализ. 2009. № 3. С. 127–135.
5. Колесникова С. И. Метод корректного определения весовых коэффициентов альтернатив в процедуре анализа иерархий // Прикладная дискретная математика. Приложение. 2009. № 1. С. 107–109.
6. Кини Р. Л., Райфа Х. Принятие решений при многих критериях: предпочтения и замещения. М.: Радио и связь, 1981. 560 с.
7. Дюкова Е. В., Журавлев Ю. И. Дискретный анализ признаковых описаний в задачах распознавания большой размерности // Журн. вычислит. матем. и математич. физ. 2000. Т. 40. № 8. С. 1264–1278.
8. Колесникова С. И., Янковская А. Е. Оценка значимости признаков для тестов в интеллектуальных системах // Изв. РАН. Теория и системы управления. 2008. № 6. С. 135–148.
9. Петровский А. Б. Упорядочивание и классификация объектов с противоречивыми признаками // Новости искусственного интеллекта. 2003. № 4. С. 34–43.

ИНФОРМАЦИЯ ДЛЯ АВТОРОВ

DOI 10.17223/20710410/6/11

УДК 001.816, 655.253.3

ИНСТРУКЦИИ И РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ СТАТЕЙ
В ФОРМАТЕ \LaTeX ДЛЯ ЖУРНАЛА
«ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА»

Д. А. Стефанцов

*Томский государственный университет, г. Томск, Россия***E-mail:** dastephanstov@mail.tsu.ru

Излагаются инструкции и рекомендации авторам для подготовки статей в журнал «Прикладная дискретная математика» в формате \LaTeX с использованием стилевого файла adm.sty.

Ключевые слова: \LaTeX , правила оформления, стилевой файл.

Данный документ содержит инструкции и рекомендации авторам для подготовки статей в формате \LaTeX с использованием стилевого файла adm.sty, разработанного для вёрстки журнала «Прикладная дискретная математика» (ПДМ) на основе стилевого файла mmro.sty, созданного К. В. Воронцовым для вёрстки сборника докладов Всероссийской конференции «Математические методы распознавания образов» (ММРО) [1]. Инструкции и рекомендации по подготовке статей в журнал ПДМ и докладов в сборник ММРО в основном совпадают, но имеются и отличия, связанные со структурными отличиями журнала от сборника и предоставлением авторам статей и верстальщику журнала некоторых дополнительных удобств. Работоспособность стилевого файла adm.sty проверена в системах MiKTeX 2.7 под Windows и TeXLive 2007 и 2008 под Windows и Linux.

Предполагается знакомство авторов статей с правилами набора текстов в системе \LaTeX [2, 3].

1. Структура статьи

Текст статьи для ПДМ начинается со строк, приведённых в листинге 1. Команда `\usepackage` подключает стилевой файл adm.sty, который должен располагаться в той же директории, что и файл статьи.

```
1 \documentclass[a4paper,twoside,12pt]{article}
2 \usepackage{adm}
3 \begin{document}
```

Листинг 1. Строки начала файла статьи

Образец оформления заголовка статьи приведён в листинге 2. Все команды являются обязательными. Необязательный первый аргумент команды `\title` задаёт краткое название статьи для верхних колонтитулов, второй аргумент задаёт полное название статьи с элементами форматирования для заглавия статьи, третий аргумент — полное название статьи без форматирования для содержания на русском языке, четвёртый аргумент — полное название статьи без форматирования для содержания на

английском языке. Необязательный первый аргумент команды `\author` задаёт краткий список авторов для верхних колонтитулов, второй аргумент задаёт полный список авторов, в котором инициалы помещаются перед фамилиями, третий — полный список авторов, в котором инициалы помещаются после фамилий, четвёртый — полный список авторов на английском языке, инициалы помещаются после фамилий. Если необязательный первый аргумент не задан, соответствующую строку для колонтитулов команда `\title` берёт из третьего аргумента, а команда `\author` — из второго. Если статья написана на английском языке, то первые три аргумента команд `\title` и `\author` составляются на английском, а четвёртый — на русском.

Для того чтобы сослаться на проект или грант, при поддержке которого выполнена работа, необходимо поместить команду `\protect\footnotemark` в конец второго аргумента команды `\title`, а текст сноски передать в качестве единственного аргумента команде `\footnotetext` в тексте статьи после команды `\maketitle`. Использование команды `\footnote` в данном случае не рекомендуется, поскольку текст, в том числе и текст сноски, переданный команде `\title` во втором аргументе, будет напечатан прописными буквами.

```

1 \udk{XYZ}
2 \title[Краткое название статьи]%
3     {Полное название статьи с форматированием}%
4     {Полное название статьи без форматирования}%
5     {Full article title without formatting}
6 \author[Краткий список авторов]%
7     {Список авторов, инициалы в начале}%
8     {Список авторов, инициалы в конце}%
9     {List of the authors, initials at the end}
10 \organization{Название организации, город, страна}
11 \email{адрес электронной почты}
12 \maketitle

```

Листинг 2. Пример оформления заголовка статьи

После заголовка статьи размещается аннотация в окружении `abstract`. После текста аннотации следует указать список ключевых слов статьи с помощью команды `\keywords` (см. листинг 3). В аннотации не должно быть пустых строк, в том числе между текстом аннотации и списком ключевых слов.

```

1 \begin{abstract}
2 Текст аннотации.
3 \keywords{список ключевых слов через запятую.}
4 \end{abstract}

```

Листинг 3. Пример оформления аннотации

Текст статьи разбивается на разделы командой `\section` с единственным аргументом — названием раздела. Форма `\section*` этой команды позволяет сделать нумерованный раздел. Рекомендуется использовать `\section*` для оформления введения и заключения. Текст раздела делится на подразделы с помощью команды `\subsection`. Также имеется возможность выделения отдельных параграфов с помощью команды `\paragraph`.

После текста статьи в окружении `thebibliography` помещается список литературы. Каждый пункт библиографии начинается командой `\bibitem{метка}`. Метка позволяет

ссылаться на данный пункт в тексте командой `\cite{метка}`. Допустимо указывать несколько меток через запятую: `\cite{метка1, метка2}`. Русские буквы в именах меток недопустимы. Определённые таким образом метки действуют в пределах статьи.

Фамилии авторов выделяются командой `\BibAuthor`, названия статей в сборниках — командой `\BibTitle`, ссылки на ресурсы в Интернете — командой `\BibUrl`. Образец оформления библиографии приведён в листинге 4.

```

1 \begin{thebibliography}{1}
2 \bibitem{bibBook}
3   \BibAuthor{Автор~И.\,0.}
4   Название книги.
5   Город:~Издательство, 2009. 314~с.
6 \bibitem{bibProceedings}
7   \BibAuthor{Автор~И.\,0.}
8   \BibTitle{Название статьи}~//
9   Название конференции или сборника,
10  Город:~Издательство, 2009. С.\,5--6.
11 \bibitem{bibArticle}
12  \BibAuthor{Автор~И.\,0., Соавтор~И.\,0.}
13  \BibTitle{Название статьи}~//
14  Название журнала. 2009. Т.\,38. \No\,5. С.\,54--62.
15 \bibitem{bibUrl}
16  \BibUrl{http://www.site.ru/} "---
17  Название сайта. 2008.
18 \end{thebibliography}

```

Листинг 4. Пример оформления библиографии

После библиографии помещается аннотация статьи на английском языке, если статья написана на русском, или аннотация на русском языке, если статья написана на английском, с помощью команды `\enabstract` (см. листинг 5). В аннотацию включается список ключевых слов на том же языке.

```

1 \enabstract{%
2   Text of abstract in another language.
3   \protect%
4   \enkeywords{list of keywords separated by comma.}
5 }

```

Листинг 5. Пример оформления аннотации на другом языке

В конце статьи помещается список сведений об авторах, пример которого показан в листинге 6. Фамилии авторов должны быть набраны прописными буквами.

```

1 \begin{authors}
2   \item{ФАМИЛИЯ Имя Отчество}{звание, степень, должность,
3     организация, город}{author@site.ru}
4   \item{ФАМИЛИЯ Имя Отчество}{звание, степень, должность,
5     организация, город}{author@site.ru}
6 \end{authors}

```

Листинг 6. Пример оформления сведений об авторах

Текст статьи завершается командой `\end{document}`.

2. Использование стандартных средств

При подготовке статьи в журнал ПДМ с помощью стилевого файла `adm.sty` можно использовать все основные средства ЛАТ_EX для задания формул, таблиц, списков, рисунков, сносок и пр. Определения ссылок `\label`, команд `\command`, `\newcommand` и ссылок библиографии `\bibitem` действуют только внутри статьи и не вызовут конфликта с идентичными определениями в статьях других авторов.

В стилевом файле `adm.sty` подключены следующие пакеты: `inputenc`, `babel`, `amssymb`, `amsmath`, `mathrsfs`, `euscript`, `array`, `theorem`, `algorithm`, `algorithmic`, `listings`, `bp-diagram`, `xu`, `graphicx`, `color`, `url`, `ifthen`. Этими пакетами можно пользоваться, не вызывая команду `\usepackage`.

Рекомендуется использовать предоставляемые файлом `adm.sty` средства. В случае необходимости изменения стиля оформления какого-либо элемента желательно пояснить это в комментарии.

2.1. З а д а н и е ф о р м у л

Все формулы, используемые внутри текста, обрамляются знаками «\$». Например, функция $\phi(pq) = (p - 1) \cdot (q - 1)$, число $2{,}71$, переменная x . Выключные формулы без номера обрамляются скобками «\» и «\» либо окружаются с обеих сторон парами символов «\$\$». Выключные формулы с номером обрамляются командами `\begin{equation}` и `\end{equation}`. Команда `\label{метка}` задаёт метку, с помощью которой можно сослаться на формулу командой `\eqref{метка}`. Для задания метки необходимо использовать латиницу. Например, использование команды `\eqref{formula}` сделает ссылку на формулу (1).

Команды для задания формулы (1) приведены в листинге 7. Окружение `cases` используется для оформления многострочной конструкции. Знак «&» выравнивает текст, помещённый в строках после него, по вертикали, символы «\» разбивают строки. Русский текст в формуле записывается с помощью команды `\text`.

```

1 \begin{equation}
2   \label{formula}
3   n! =
4   \begin{cases}
5     n \cdot (n - 1)!, & \text{если } n > 0; \\
6     1, & \text{если } n = 0.
7   \end{cases}
8 \end{equation}

```

Листинг 7. Пример задания выключной формулы с номером

В результате трансляции фрагмента, приведённого в листинге 7, будет получена формула

$$n! = \begin{cases} n \cdot (n - 1)!, & \text{если } n > 0; \\ 1, & \text{если } n = 0. \end{cases} \quad (1)$$

Рекомендуется использование окружений `align`, `gather`, `multline` и `split` для разбиения длинных формул на несколько строк. Следующая формула является примером использования окружения `align*`:

$$b = \bigoplus_{i=0}^{n-1} a_i x_{k+i},$$

$$x_{m+n} = x_m \left(\bigoplus_{i=1}^{n-1} x_{m+i} \right).$$

В листинге 8 приведён текст, трансляцией которого была получена эта формула.

```

1 \begin{align*}
2 \label{eqalign}
3 b &= \bigoplus_{i=0}^{n-1} a_i x_{k+i}, \\
4 x_{m+n} &= x_m \left( \bigoplus_{i=1}^{n-1} x_{m+i} \right).
5 \end{align*}

```

Листинг 8. Пример использования окружения `align*`

Команды `\left` и `\right` использованы для того, чтобы сделать высоту скобок соответствующей высоте обрамляемой ими подформулы.

2.2. Оформление рисунков

Иллюстрации рекомендуется выполнять в векторных графических редакторах с последующим сохранением их в форматах SVG или EPS — при масштабировании векторная графика не теряет качества. EPS-файлы желательно конвертировать в PDF утилитой `epstopdf`. Созданные таким образом файлы рисунков могут быть подключены в тексте так же, как и прочие рисунки, а сам файл статьи может быть транслирован в PDF утилитой `pdflatex`. При создании файлов EPS или при конвертировании SVG в EPS необходимо включать в создаваемый файл используемые шрифты. Если нет возможности создать векторный рисунок (например, необходима вставка фотографии), то можно использовать растровые изображения в форматах BMP, PNG или JPG. Все рисунки должны быть выполнены в оттенках серого.

В листинге 9 приведены команды, использованные для оформления рис. 1.

```

1 \begin{figure}[ht]
2 \centering
3 \includegraphics[scale=0.2]{isc.pdf}
4 \caption{Значок кафедры защиты информации и криптографии}
5 \label{isc}
6 \end{figure}

```

Листинг 9. Пример оформления рисунка

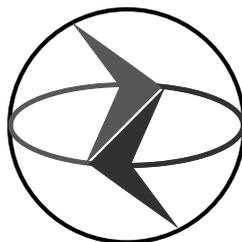


Рис. 1. Значок кафедры защиты информации и криптографии

Команда `\centering` используется для выравнивания рисунка по центру страницы. Команда `\caption` задаёт название рисунка. Команда `\label{метка}` позволяет сослаться на рисунок в тексте статьи с помощью команды `\ref{метка}`. Команда `\label` должна идти после команды `\caption`. В метках допустимы только латинские буквы и цифры, начинаться метка должна с буквы. Необязательный параметр окружения `ht` указывает, в какой части страницы следует разместить рисунок (в примере указано размещение либо в середине текста, либо в начале страницы). Название рисунка рекомендуется размещать под графическим изображением.

Графы можно изображать средствами стандартных пакетов ЛАТЭХ. Данные для изображения графа задаются в окружении `network`. Команда `\nnNode` задаёт имя и координаты вершины, команда `\nnLink` связывает две вершины. Внешний вид вершин и связей задаётся средствами пакета `xu`. Для того чтобы в названиях дуг и вершин можно было использовать математические формулы, окружение `network` помещается в выключную формулу скобками «`\[`» и «`\]`». Оформление графа рекомендуется помещать в окружение `figure`, для того чтобы граф был оформлен как рисунок. Как и для рисунков, необходимо использовать команды `\caption` и `\label`. Пример оформления графа приведён в листинге 10.

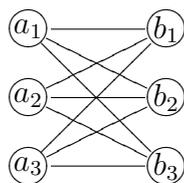
```

1 \begin{figure}[H]
2 \centering
3 \[
4   \begin{network}
5     \nnNode"a1"( 0, 10)    {+[o][F]{a_1}}
6     \nnNode"a2"( 0,  5)    {+[o][F]{a_2}}
7     \nnNode"a3"( 0,  0)    {+[o][F]{a_3}}
8     \nnNode"b1"(10, 10)    {+[o][F]{b_1}}
9     \nnNode"b2"(10,  5)    {+[o][F]{b_2}}
10    \nnNode"b3"(10,  0)    {+[o][F]{b_3}}
11    \nnLink"a1,b1"        {@{-}}
12    \nnLink"a1,b2"        {@{-}}
13    \nnLink"a1,b3"        {@{-}}
14    \nnLink"a2,b1"        {@{-}}
15    \nnLink"a2,b2"        {@{-}}
16    \nnLink"a2,b3"        {@{-}}
17    \nnLink"a3,b1"        {@{-}}
18    \nnLink"a3,b2"        {@{-}}
19    \nnLink"a3,b3"        {@{-}}
20  \end{network}
21 \]
22 \caption{Граф  $K_{3,3}$ }
23 \label{network}
24 \end{figure}

```

Листинг 10. Пример оформления графа

В результате компиляции данного набора команд получится граф, изображённый на рис. 2. Сослаться на такой рисунок можно командой `\ref{network}`.

Рис. 2. Граф $K_{3,3}$

2.3. Оформление таблиц

Таблицы оформляются с помощью окружения `table`. В этом окружении помещаются дополнительные команды такие, как `\centering`, `\caption` и `\label`, которые имеют то же значение, что и для рисунков. Также в окружение `table` помещается окружение `tabular`, в котором задаются табличные данные. Название рекомендуется размещать над таблицей. Пример оформления таблицы приведён в листинге 11.

```

1 \begin{table}[ht]
2 \centering
3 \caption{Название таблицы}
4 \label{tab1}
5 \begin{tabular}{|c|c|c|}
6 \hline
7 Один & Два & Три \\
8 \hline
9 Четыре & Пять & Шесть \\
10 \hline
11 Семь & Восемь & Девять \\
12 \hline
13 \end{tabular}
14 \end{table}

```

Листинг 11. Пример оформления таблицы

В результате будет получена табл. 1, а именно:

Таблица 1

Название таблицы

Один	Два	Три
Четыре	Пять	Шесть
Семь	Восемь	Девять

2.4. Оформление алгоритмов и программ

Алгоритмы могут быть оформлены в виде псевдокода с помощью окружения `Algorithm`, внутри которого определены такие ключевые слова, как `\STATE`, `\FOR`, `\FORALL`, `\ENDFOR`, `\IF`, `\ENDIF`, `\PRINT` и др. Пример оформления алгоритма:

Алгоритм 1. Вывод всех элементов множества

- 1: Для всех $a \in A$
 - 2: Вывести a
-

В листинге 12 приведены команды оформления алгоритма 1. Команды `\caption` и `\label` имеют то же значение, что и для рисунков. На этот алгоритм можно сослаться с помощью команды `\ref{algo}`, а на строку 2 — с помощью команды `\ref{printing}`.

```

1 \begin{Algorithm}[ht]
2 \caption{Вывод всех элементов множества}
3 \label{algo}
4 \FORALL{$a \in A$}
5     \PRINT $a$ \label{printing}
6 \ENDFOR
7 \end{Algorithm}

```

Листинг 12. Пример оформления алгоритма

Фрагменты программ можно приводить в окружении `verbatim`, вложенном в окружение `figure`, что позволит использовать команды `\label` для формирования метки и `\caption` для формирования подписи, которая должна размещаться под фрагментом программы. Рекомендуется применять средства пакета `listings`, подключенного в стилевом файле `adm.sty`. Пример включения текста программы на языке Python приведён в листинге 13.

```

1 \lstset{float=ht,
2     caption={Программа на языке Python}, label={pyprog}}
3 \begin{lstlisting}
4 from re import *
5
6 if __name__ == '__main__':
7     pat = compile('[0-9]+') (*@ \label{compiling} @*)
8     print search(pat, 'In a year 2009...').group()
9 \end{lstlisting}

```

Листинг 13. Пример оформления фрагмента программы

Параметр `float` команды `\lstset` задаёт расположение листинга в тексте статьи, параметры `caption` и `label` имеют такой же смысл, как и команды `\caption` и `\label` при оформлении рисунков. С помощью скобок «`*@`» «`@*`» и команды `\label` можно определять ссылки на строки фрагмента. В приведённом примере ссылка на строку 4 может быть сделана командой `\ref{compiling}`. В результате будет получен листинг 14, а именно:

```

1 from re import *
2
3 if __name__ == '__main__':
4     pat = compile('[0-9]+')
5     print search(pat, 'In a year 2009...').group()

```

Листинг 14. Программа на языке Python

Не рекомендуется с помощью команды `\lstset` менять значения параметров `numbers`, `frame`, `breaklines`, `captionpos`, `columns`, `flexiblecolumns`, `keepspace`, `basewidth`, `fontadjust`, `basicstyle`, `xleftmargin`, `xrightmargin`, `aboveskip`, `belowskip`. Значения для них определены в стилевом файле `adm.sty`.

3. Математические обозначения

В стилевом файле переопределены некоторые стандартные и введены новые команды для оформления формул и окружений типа теоремы.

Обозначения множеств чисел \mathbb{N} , \mathbb{Z} , \mathbb{R} выполняются с помощью команд `\NN`, `\ZZ` и `\RR` соответственно. Некоторые математические символы приведены в соответствие с традициями русской типографии: `\geq` (\geq), `\leq` (\leq), `\emptyset` (\emptyset), `\epsilon` (ϵ), `\kappa` (κ), `\phi` (φ). Определены математические операторы `\argmin`, `\argmax`, `\diag`, `\sign`, `\Tr`, `\const`, а математические операторы `\lim`, `\inf`, `\sup`, `\max`, `\min` переопределены так, что пределы ставятся под ними, а не сбоку. С помощью команд `\mylim` и `\myop` можно определять собственные математические операторы с пределом снизу и без пределов соответственно.

Для выделения векторных и матричных величин можно пользоваться командой `\vec{формула}`. Для набора формул теории вероятности предназначены следующие команды: `\Prob` (вероятность), `\Expect` (математическое ожидание), `\Var` (дисперсия), `\Normal` (нормальное распределение). В условных вероятностях вертикальная черта ставится командой `\cond`.

Окружения предложений типа теоремы следующие: `Theorem` — теорема, `Lemma` — лемма, `State` и `State-rm` — утверждение, `Corollary` — следствие, `Def` — определение, `Hypothesis` — гипотеза, `Problem` — задача, `Example` — пример, `Remark` — замечание, `Proof` — доказательство. Нумерация каждого из элементов сквозная. Доказательство завершается символом `\qed` автоматически. Ниже приведён пример использования этих окружений.

Утверждение 1. Текст утверждения.

Определение 1. Текст определения.

Теорема 1. Текст теоремы.

Следствие 1. Текст следствия.

Следствие 2. Текст другого следствия.

Лемма 1. Текст леммы.

Доказательство. Доказательство леммы 1. ■

Следствие 3. Следствие леммы 1.

Лемма 2. Текст другой леммы.

Теорема 2 (название или имя автора). Текст теоремы с названием.

Доказательство. Доказательство теоремы 2. ■

Пример 1. Текст примера.

Текст, компиляцией которого получен данный пример, приведён в листинге 15.

```

1 \begin{State}
2 Текст утверждения.
3 \end{State}
4 \begin{Definition}
5 Текст определения.
6 \end{Definition}
7 \begin{Theorem}
8 Текст теоремы.
9 \end{Theorem}
```

```

10 \begin{Corollary}
11 Текст следствия.
12 \end{Corollary}
13 \begin{Corollary}
14 Текст другого следствия.
15 \end{Corollary}
16 \begin{Lemma}
17 \label{lemma1}
18 Текст леммы.
19 \end{Lemma}
20 \begin{Proof}
21 Доказательство леммы~\ref{lemma1}.
22 \end{Proof}
23 \begin{Corollary}
24 Следствие леммы~\ref{lemma1}.
25 \end{Corollary}
26 \begin{Lemma}
27 Текст другой леммы.
28 \end{Lemma}
29 \begin{Theorem}[(название или имя автора)]
30 \label{theorem1}
31 Текст теоремы с названием.
32 \end{Theorem}
33 \begin{Proof}
34 Доказательство теоремы~\ref{theorem1}.
35 \end{Proof}
36 \begin{Example}
37 Текст примера.
38 \end{Example}

```

Листинг 15. Пример использования окружений предложений типа теоремы

4. Рекомендации по оформлению текста

При подготовке статей в журнал ПДМ рекомендуется придерживаться общих правил для подготовки печатных текстов на русском и английском языках с помощью системы \LaTeX .

В тексте на русском языке кавычки ставятся парами символов «<<» и «>>». Вложенные кавычки ставятся парами символов «,» и «‘», например «Крейсер „Варяг“». В тексте на английском языке кавычки ставятся парами символов ‘ и ’, например the “Applied Discrete Mathematics” journal.

Знаки препинания (точки, запятые и т. д.) набираются слитно с предшествующим текстом и отделяются пробельным символом от последующего. Скобки пишутся слитно с текстом, который они окружают. Тире в тексте на русском языке оформляется командой "---, а в английском — командой ---. Тире отделяется от предшествующего и последующего текста пробельными символами. Диапазоны чисел оформляются с помощью команды --, например «С. 50–64». В сложных словах дефис ставится командой "=", например визуально"=матричный, объектно"=ориентированный.

Неразрывный пробел ~ рекомендуется использовать для того, чтобы короткие слова и формулы в конце абзаца не переносились на новую строку, а также чтобы не

отрывать предлоги от следующих за ними слов. Короткий неразрывный пробел `\,` используется в инициалах и сокращениях типа `т.\,д.` и `т.\,п.`

Списки рекомендуется оформлять следующим образом:

- 1) после номера ставить скобку;
- 2) пункты завершать точкой с запятой;
- 3) последний пункт завершать точкой.

Для этих целей подходит окружение `enumerate*`.

5. Подготовка статей в кодировках, отличных от `cp1251`

Для вёрстки журнала ПДМ используется кодировка `cp1251`. При подготовке статьи в другой кодировке, например `koï8-r`, необходимо:

- 1) перекодировать файл `adm.sty` из кодировки `cp1251` в требуемую кодировку;
- 2) заменить параметр `\RequirePackage[cp1251]{inputenc}` в файле `adm.sty` с `cp1251` на название требуемой кодировки.

Поскольку пакет `listings` работает некорректно, если внутри находится текст в многобайтовой кодировке, например `utf8`, рекомендуется использовать кодировку `cp1251` или `koï8-r` для статей, содержащих листинги со словами на русском языке.

Файлы `adm.sty.koï8-r`, `adm.sty.cp866`, `adm.sty.utf8` являются версиями файла `adm.sty` для работы с кодировками `koï8-r`, `cp866` и `utf8` соответственно и распространяются вместе с `adm.sty`. При необходимости один из них можно переименовать в `adm.sty` и использовать вместо оригинального стилевого файла.

ЛИТЕРАТУРА

1. <http://www.ccas.ru/voron/latex.html> — Подготовка сборника трудов конференции в системе \LaTeX 2007.
2. Котельников И. А., Чеботаев П. З. \LaTeX 2_ε по-русски. Новосибирск: Сибирский хронограф, 2004. 489 с.
3. Балдин Е. М. Компьютерная типография \LaTeX . СПб.: БХВ-Петербург, 2008. 304 с.

ТЕМАТИКА ЖУРНАЛА

- 1) *Теоретические основы прикладной дискретной математики* — алгебраические структуры, дискретные функции, комбинаторный анализ, теория чисел, математическая логика, теория информации, системы уравнений над конечными полями и кольцами.
- 2) *Математические методы криптографии* — синтез криптосистем, методы криптоанализа, генераторы псевдослучайных последовательностей, оценка стойкости криптосистем, криптографические протоколы, математические методы квантовой криптографии.
- 3) *Математические методы стеганографии* — синтез стеганосистем, методы стеганоанализа, оценка стойкости стеганосистем.
- 4) *Математические основы компьютерной безопасности* — математические модели безопасности компьютерных систем (КС), математические методы анализа безопасности КС, математические методы синтеза защищенных КС.
- 5) *Математические основы надежности вычислительных и управляющих систем (ВиУС)* — математические модели функциональной устойчивости ВиУС (к отказам, неисправностям, сбоям, состязаниям, исследованию), математические методы анализа функциональной устойчивости ВиУС, математические методы синтеза функционально устойчивых ВиУС, математические методы верификации логических схем и программ, математические методы синтеза самопроверяемых и контролепригодных схем.
- 6) *Прикладная теория кодирования* — коды для сжатия данных и защиты информации, коды для обнаружения и исправления ошибок, построение оптимальных кодов, анализ свойств кодов.
- 7) *Прикладная теория автоматов* — автоматные модели сетевых протоколов, криптосистем и управляющих систем, автоматы без потери информации, эксперименты с автоматами, декомпозиция автоматов, автоматные уравнения, клеточные автоматы.
- 8) *Логическое проектирование дискретных автоматов* — математические модели и методы анализа, синтеза, оптимизации и оценки сложности дискретных автоматов, аппаратная реализация криптоалгоритмов.
- 9) *Математические основы информатики и программирования* — формальные языки и грамматики, алгоритмические системы, языки программирования, структуры и алгоритмы обработки данных, теория вычислительной сложности.
- 10) *Вычислительные методы в дискретной математике* — теоретико-числовые методы в криптографии, вычислительные методы в теории чисел и общей алгебре, комбинаторные алгоритмы, параллельные вычисления, методы дискретной оптимизации, дискретно-событийное и клеточно-автоматное моделирование.
- 11) *Математические основы интеллектуальных систем* — базы данных, базы знаний, логический вывод, экспертные системы, математическая лингвистика, формализация естественных языков, анализ текстов.

- 12) *Прикладная теория графов* — графовые модели в информатике и программировании, в компьютерной безопасности, вычислительных и управляющих системах, в интеллектуальных системах.
- 13) *Исторические очерки по дискретной математике и ее приложениям* — в криптографии, компьютерной безопасности, кибернетике, информатике, программировании и теории надежности.

TOPICS OF THE JOURNAL

- 1) *Theoretical foundations of applied discrete mathematics* — algebraic structures, discrete functions, combinatorial analysis, number theory, mathematical logic, information theory, systems of equations over finite fields and rings.
- 2) *Mathematical methods in cryptography* — synthesis of cryptosystems, methods for cryptanalysis, pseudorandom generators, appreciation of cryptosystem security, cryptographic protocols, mathematical methods in quantum cryptography.
- 3) *Mathematical methods in steganography* — synthesis of steganosystems, methods for steganoanalysis, appreciation of steganosystem security.
- 4) *Mathematical foundations of computer security* — mathematical models for computer system security, mathematical methods for the analysis of the computer system security, mathematical methods for the synthesis of protected computer systems.
- 5) *Mathematical foundations of computer and control system reliability* — mathematical models for functionally stable computer and control systems (that is for ones that are correctly function in the presence of faults, hazards and so on), mathematical methods for the analysis of the functional stability of computer and control systems, mathematical methods for the synthesis of functionally stable computer and control systems, mathematical methods for the verification of switching circuits and for program testing, mathematical methods for the synthesis of self checking and testable circuits.
- 6) *Applied coding theory* — data compressing and information protecting codes, error correcting and error detecting codes, construction of optimal codes, analysis of code properties.
- 7) *Applied theory of automata* — automaton models for net protocols, for cryptosystems and for control systems, information-lossless automata, experiments on automata, decomposition of automata, automaton equations, cellular automata.
- 8) *Logical design of discrete automata* — mathematical models and methods for analysis, synthesis, optimization and complexity appreciation of discrete automata, apparatus realization of cryptographic algorithms.
- 9) *Mathematical foundations of informatics and programming* — formal languages and grammars, algorithmic systems, programming languages, data structures, data processing algorithms, theory of computing complexity.

-
- 10) *Computing methods in discrete mathematics* — number theory methods in cryptography, computing methods in number theory and abstract algebra, combinatorial algorithms, parallel computations, methods for discrete optimization, discrete-event and cellular automaton models.
 - 11) *Mathematical foundations of intelligent systems* — data bases, knowledge bases, logical inference, expert systems, mathematical linguistics.
 - 12) *Applied graph theory* — graph models in informatics and programming, in computer security, in computer and control systems, in intellectual systems.
 - 13) *Historical records on discrete mathematics and its applications* — in cryptography, in computer security, in cybernetics, in informatics, in programming and in reliability theory.

СВЕДЕНИЯ ОБ АВТОРАХ

ГРОМОВ Максим Леонидович — магистр радиофизики, старший преподаватель Томского государственного университета, г. Томск. E-mail: gromov@sibmail.com

ЕВСЕЕВ Алексей Алексеевич — магистрант Новосибирского государственного университета, г. Новосибирск. E-mail: evseev.alexei@gmail.com

ЕВТУШЕНКО Нина Владимировна — профессор, доктор технических наук, зав. кафедрой ИТИДиС Томского государственного университета, г. Томск. E-mail: ninayevtushenko@yahoo.com

КОЛЕСНИКОВА Светлана Ивановна — кандидат физико-математических наук, доцент кафедры экономической математики, информатики и статистики Томского государственного университета систем управления и радиоэлектроники (ЭМИС ТУСУР), г. Томск. E-mail: skolesnikova@yandex.ru

КОЛОМЕЕЦ Николай Александрович — студент Новосибирского государственного университета, г. Новосибирск. E-mail: nkolomeec@gmail.com

НЕЧАЕВА Ольга Игоревна — кандидат физико-математических наук, зав. лабораторией НГУ-Интел, Новосибирский государственный университет, г. Новосибирск. E-mail: oinechaeva@gmail.com

ПАВЛОВ Андрей Владимирович — студент Новосибирского государственного университета, г. Новосибирск. E-mail: apavlov.nsk@gmail.com

ПАРВАТОВ Николай Георгиевич — кандидат физико-математических наук, доцент кафедры защиты информации и криптографии Томского государственного университета, г. Томск. E-mail: parvatov@mail.tsu.ru

ПЕСТУНОВ Андрей Игоревич — научный сотрудник Института вычислительных технологий СО РАН, г. Новосибирск. E-mail: pestunov@gmail.com

СЕМЁНОВ Александр Анатольевич — кандидат технических наук, заведующий лабораторией дискретного анализа и прикладной логики Института динамики систем и теории управления СО РАН (ИДСТУ СО РАН), г. Иркутск. E-mail: biclop@rambler.ru

СКОБЕЛЕВ Владимир Владимирович — аспирант Института прикладной математики и механики НАН Украины, г. Донецк. E-mail: vv_skobelev@iamm.ac.donetsk.ua

СТЕФАНЦОВ Дмитрий Александрович — аспирант Томского государственного университета, г. Томск. E-mail: dastephantsov@mail.tsu.ru

ФЕДЮКОВИЧ Вадим Евгеньевич — разработчик программного обеспечения, Интропро, г. Киев. E-mail: vf@unity.net

ЧЕРЕМУШКИН Александр Васильевич — доктор физико-математических наук, зав. кафедрой Института криптографии, связи и информатики, г. Москва. E-mail: avc238@mail.ru

АННОТАЦИИ СТАТЕЙ НА АНГЛИЙСКОМ ЯЗЫКЕ

Kolomeec N. A., Pavlov A. V. **PROPERTIES OF BENT FUNCTIONS WITH MINIMAL DISTANCE.** The minimal Hamming distance $2^{n/2}$ between distinct bent functions of n variables is obtained. We prove that two bent functions are at the minimal distance if and only if the set of vectors for which they differ is a linear manifold and both functions are affine ones on it. We give an algorithm for constructing all the bent functions being at the minimal distance from the given bent function. Some experimental data are presented for bent functions of the small number of variables.

Keywords: *bent function, CDMA, OFDM.*

Parvatov N. G. **ABOUT INVARIANTS FOR SOME CLASSES OF QUASI-MONOTONIC FUNCTIONS ON A SEMILATTICE.** Invariant predicates for some classes of quasimonotonic and monotonic functions on a finite semilattice are studied. Generating sets in the systems of such predicates are defined. For the purpose of generating, the operations of predicate conjunction and variable relabeling are used.

Keywords: *semilattice, monotonic function, quasimonotonic function, invariant predicates, generating sets.*

Semenov A. A. **ABOUT TSEITIN TRANSFORMATION IN LOGICAL EQUATIONS.** The paper is devoted to the applications of Tseitin transformations in some propositional logics areas connected with the systems of logical equations. It is shown that Tseitin transformations of the system of logical equations do not change the number of its solutions and a bijection is pointed between the solutions of the system and its transformation result. Some results related to the usage of Tseitin transformations in obtaining bounds for the complexity of the propositional proof systems are given. By using Tseitin transformations, the simplest proofs of the NP-completeness problem for the solvability of a 2-degree logical equations system and of the #P-completeness problem for counting the number of satisfying assignments for any Horne CNF are constructed too. By using Tseitin transformations, it is also shown that the ROBDD for any Boolean function given in a Horne CNF or in a CNF with 2-literal disjuncts may not be build for a polynomial time if $P \neq NP$.

Keywords: *logical equations, Tseitin transformations, propositional proof systems, NP-completeness.*

Cheremushkin A. V. **A RECURSIVE ALGORITHM FOR COVER-FREE FAMILY CONSTRUCTION.** A new recursive algorithm based on orthogonal arrays is proposed for cover-free family construction. The algorithm modifies the one suggested by Stinson D. R., van Trung T, and Wei R. As a consequence we obtain the method for recursive construction of collusion-resistant key distribution schemes.

Keywords: *cover-free family, key distribution scheme.*

Pestunov A. I. **DIFFERENTIAL CRYPTANALYSIS OF THE MARS BLOCK CIPHER.** In this work we present a differential attack on MARS which breaks 8 core and 8 mixing rounds with pre- and post-whitening. This attack is based on a new 8-core round differential characteristic with probability 2^{-98} and allows to recover more subkeys bits than previously published attacks (752 instead of 682) faster than exhaustive key search.

The success probability of the attack is more than 0,99. The attack requires 2^{105} chosen plaintexts, 2^{109} bytes of memory and 2^{231} encryptions.

Keywords: *block cipher, differential attack, Advanced Encryption Standard, MARS.*

Fedyukovich V. E. **ARGUMENT OF KNOWLEDGE PROTOCOL FOR A GOPPA CODEWORD AND FOR AN ERROR OF A BOUNDED WEIGHT.**

A new argument of knowledge protocol with honest verifier is proposed for the Goppa polynomial, codeword and the error of a bounded weight. The soundness of the protocol is based on the hardness assumption for the discrete logarithm problem.

Keywords: *interactive argument system, zero knowledge, commitment scheme, Goppa code.*

Evseev A. A., Nechaeva O. I. **CELLULAR AUTOMATA SIMULATION ON SURFACE TRIANGULATION FOR DIFFUSION PROCESSES.** This work is devoted to the development of techniques for cellular automata simulation on triangulation grids on flat and curved surfaces. Possibilities of the proposed techniques are shown on examples of cellular automata simulation of diffusion, front propagation and diffusion-limited aggregation.

Keywords: *cellular automata, triangulation, diffusion, front propagation.*

Skobelev V. V. **“STRINGS” THEOREM AND ITS APPLICATIONS.** A general method based on the set of marked strings is proposed for calculating the number of elements in a finite set defined in terms of residue classes.

Keywords: *residue classes, systems of congruences.*

Gromov M. L., Yevtushenko N. V. **ADAPTIVE TESTS DERIVATION FOR NON-DETERMINISTIC AUTOMATA.** The paper is devoted to the adaptive distinguishing experiments derivation for the following automata models: Labelled Transition Systems, nondeterministic, possibly partial, Finite State Machines and Timed Finite State Machines. The methods suggested here do not exploit “all weather condition” assumption and are based on the intersection of the corresponding automata. The experiments can be used for test derivation and diagnosis for discrete event systems.

Keywords: *Labelled Transition Systems, nondeterministic Finite State Machine, Timed Finite State Machine, distinguishing experiment, compatibility, adaptive experiment.*

Kolesnikova S. I. **MODIFICATION OF HIERARCHIES ANALYSIS METHOD FOR THE DYNAMIC SET OF ALTERNATIVES.** The problem of incorrect estimation for alternatives in a dynamic set by Saaty’s hierarchies analysis method used for regularities discovery and decision-making support is considered. A modified procedure for estimation of alternatives significance is suggested. The properties of this procedure are proved and illustrative examples are given.

Keywords: *hierarchies analysis method, alternatives significance, decision-making support.*

Stephantsov D. A. **INSTRUCTIONS AND RECOMMENDATIONS FOR AUTHORS TO PREPARE ARTICLES IN LATEX FORMAT FOR “APPLIED DISCRETE MATHEMATICS” JOURNAL.** The document contains instructions for authors preparing articles for the “Applied Discrete Mathematics” journal with \LaTeX typesetting system and special style file `adm.sty` created for this purpose.

Keywords: \LaTeX , *instructions for authors, style file.*