МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

# ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

*Научный журнал*

2011 №2(12)

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, чл.-корр. Академии криптографии (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.

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

---

Подписано к печати 26.05.2011.
Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 13,4. Уч.-изд. л. 15. Тираж 300 экз.

---

Отпечатано в типографии ТПУ.

# СОДЕРЖАНИЕ

# CONTENTS

# ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

DOI 10.17223/20710410/12/1

УДК 519.7

## О НЕКОТОРЫХ МЕРАХ НЕЛИНЕЙНОСТИ БУЛЕВЫХ ФУНКЦИЙ[1]

Е. К. Алексеев

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

**E-mail:** geni-cmc@mail.ru

Рассматривается расстояние до алгебраически вырожденных функций как мера нелинейности булевых функций. Устанавливаются соотношения между этим расстоянием и некоторыми ранее предложенными мерами нелинейности булевых функций. Исследуется порядок алгебраической вырожденности тех функций, которые наилучшим образом аппроксимируют данную.

**Ключевые слова:** *нелинейность булевых функций, алгебраически вырожденные функции, пространство линейных структур, криптография.*

### Введение

Одним из приемов современного криптоанализа является использование близости (относительно некоторой метрики) криптографических отображений к отображениям, позволяющим свести исходную криптографическую задачу к менее трудоемкой. Яркими примерами этого являются корреляционный [1] и линейный [2] методы криптоанализа. Одна из возможных характеристик, показывающая эффективность этих методов, была названа нелинейностью булевой функции [3]. Однако могут быть использованы не только линейные (аффинные) приближения. В работе [3] также рассматривались меры близости к другим «слабым» с криптографической точки зрения классам функций. Множество алгебраически вырожденных функций в этом плане представляет определенный практический и теоретический интерес.

В данной работе рассматривается расстояние до алгебраически вырожденных функций. Устанавливаются соотношения между этим расстоянием и некоторыми ранее предложенными мерами нелинейности булевых функций. Рассматривается также степень алгебраической вырожденности тех функций, которые наилучшим образом аппроксимируют данную функцию.

### 1. Основные определения и обозначения

Пусть $F_2$ — конечное поле из двух элементов. Пусть $V_n = \underbrace{F_2 \times \ldots \times F_2}_{n}$ — векторное пространство наборов длины $n$ с компонентами из $F_2$. *Булевой функцией* от $n$ переменных будем называть отображение из $V_n$ в $F_2$. Множество всех булевых функций от $n$ переменных будем обозначать $\mathcal{F}_n$. *Носителем* функции $f \in \mathcal{F}_n$ называется множество $1_f = \{x \in V_n : f(x) = 1\}$. *Весом* $\mathrm{wt}(f)$ булевой функции $f \in \mathcal{F}_n$ называется мощность ее носителя $1_f$.

[1]Работа поддержана РФФИ (проект № 09-01-00653-а).

В дальнейшем будем придерживаться следующих обозначений: $x^{(i)}$ — $i$-й вектор из некоторой совокупности векторов; $x_j$ — $j$-я компонента вектора $x$. Если $n$ — натуральное число, а $c \in \{0,1\}$, то через $c^n$ будем обозначать вектор длины $n$, все компоненты которого равны $c$. Константные булевы функции будем обозначать через $\overline{0}$ и $\overline{1}$.

Пусть $L$ — подмножество пространства $V_n$. Тот факт, что $L$ является подпространством пространства $V_n$, будем обозначать так: $L < V_n$. Для произвольного линейного подпространства $L < V_n$ через $L^*$ будем обозначать множество $L \setminus \{0^n\}$. Плоскостями в $V_n$ будем называть смежные классы по подпространствам этого пространства.

*Индикаторной функцией* множества $S \subseteq V_n$ называется такая функция $I_S \in \mathcal{F}_n$, что $I_S(x) = 1$ тогда и только тогда, когда $x \in S$.

Для булевой функции $f$ от $n$ переменных и вектора $u \in V_n$ будем обозначать через $f^u$ функцию $f^u(x) = f(x \oplus u)$. *Производной булевой функции* $f$ по направлению $u \in V_n^*$ называется функция $D_u f = f \oplus f^u$. Через $J(f)$ будем обозначать подпространство $\{u \in V_n : f^u = f\}$ пространства $V_n$.

Пусть $A$ — $(n \times k)$-матрица над $F_2$, а $g$ — булева функция от $k$ переменных. Через $g^A$ будем обозначать функцию от $n$ переменных $g^A(x) = g(xA)$.

**Утверждение 1** [4]. Любая функция $f \in \mathcal{F}_n$ представима в виде

$$f = \bigoplus_{i=1}^{2^{n-\dim J(f)}} \varepsilon_i I_{J(f) \oplus z^{(i)}},$$

где $\varepsilon_i \in \{0,1\}$; $z^{(i)} \in V_n$.

**Определение 1** [5]. *Порядком алгебраической вырожденности* $AD(f)$ булевой функции $f \in \mathcal{F}_n$ называется максимально возможное значение $n-k$, где для целого числа $k$, $0 \leqslant k \leqslant n$, существуют такие функция $g \in \mathcal{F}_k$ и $(n \times k)$-матрица $D$ над $F_2$, что выполнено равенство $f = g^D$. Функции, для которых $AD(f) > 0$, называются *алгебраически вырожденными*. Множество всех алгебраически вырожденных функций от $n$ переменных обозначим через

$$DG(n) = \{f \in \mathcal{F}_n : AD(f) > 0\}.$$

**Определение 2.** Булева функция $f \in \mathcal{F}_n$ обладает нетривиальной *линейной структурой*, если существует такой вектор $u \in V_n^*$, что $D_u f \in \{\overline{0}, \overline{1}\}$. Подпространство $L_f = \{u \in V_n : D_u f \in \{\overline{0}, \overline{1}\}\}$ называется *пространством линейных структур* функции $f$.

Множество булевых функций $f \in \mathcal{F}_n$, имеющих нетривиальную линейную структуру, обозначается через $\mathcal{F}_n^0$ [6]. Легко показать, что если для $f \in \mathcal{F}_n^0$ существует такой вектор $u \in V_n^*$, что $D_u f = \overline{1}$, то $L_f = J(f) \cup (J(f) \oplus u)$.

**Определение 3.** *Преобразованием Уолша — Адамара* булевой функции $f$ называется целочисленная функция на $V_n$, определяемая следующим равенством:

$$W_f(u) = \sum_{x \in V_n} (-1)^{f(x) \oplus <x,u>}$$

(суммирование производится в действительной области). Для каждого $u \in V_n$ значение $W_f(u)$ называется *коэффициентом Уолша — Адамара*.

**Определение 4.** Пусть $f \in \mathcal{F}_n$. Целочисленная функция

$$\Delta_f(u) = \sum_{x \in V_n} (-1)^{f(x) \oplus f(x \oplus u)},$$

определенная на пространстве $V_n$, называется *автокорреляционной функцией*, или *автокорреляцией* булевой функции $f$.

Справедлива следующая теорема о связи автокорреляционной функции с преобразованием Уолша — Адамара.

**Теорема 1** [6]. Пусть $f \in \mathcal{F}_n$. Тогда для любого вектора $v \in V_n$ справедливы соотношения

$$W_f^2(v) = \sum_{u \in V_n} \Delta_f(u)(-1)^{<u,v>},$$
$$\Delta_f(v) = 2^{-n} \sum_{u \in V_n} W_f^2(u)(-1)^{<u,v>}.$$

**Определение 5.** Булева функция $f \in \mathcal{F}_n$ называется *корреляционно-иммунной порядка* $m$, $0 < m \leqslant n$, если для любого вектора $u \in V_n$, такого, что $1 \leqslant \mathrm{wt}(u) \leqslant m$, выполнено равенство $W_f(u) = 0$.

Корреляционно-иммунная порядка $m$ функция является корреляционно-иммунной любого меньшего порядка, поэтому введем обозначение

$$\mathrm{cor} f = \max\{m \in \mathbb{N} : f - \text{корреляционно-иммунна порядка } m\}.$$

Справедливо следующее соотношение между порядком корреляционной иммунности $\mathrm{cor} f$ булевой функции $f$ и ее алгебраической степенью $\deg f$.

**Теорема 2** (неравенство Зигенталера). Для любой функции $f \in \mathcal{F}_n$ справедливо неравенство $\deg f + \mathrm{cor} f \leqslant n$.

**Определение 6.** Функция $f \in \mathcal{F}_n$ называется *бент-функцией*, если все ее коэффициенты Уолша — Адамара равны $\pm 2^{n/2}$.

**Определение 7.** Булева функция $f$ от $n$ переменных называется *уравновешенной*, если $\mathrm{wt}(f) = 2^{n-1}$. Уравновешенная корреляционно-иммунная порядка $d$ функция $f \in \mathcal{F}_n$ называется *$d$-устойчивой*.

**Теорема 3** (критерий Ротхауза). Булева функция $f \in \mathcal{F}_n$ является бент-функцией тогда и только тогда, когда ее производная $D_u f$ по каждому ненулевому направлению $u \in V_n^*$ уравновешена.

**Определение 8.** Пусть $f \in \mathcal{F}_n$. Если существует такое натуральное число $r \in \mathbb{N}$, $0 \leqslant r \leqslant n$, что квадрат каждого коэффициента Уолша — Адамара равен либо $2^{2n-2r}$, либо 0, то функция $f$ называется *платовидной функцией порядка* $2r$.

**Лемма 1.** Пусть $f \in \mathcal{F}_n$, $\deg f = d \geqslant 1$. Тогда

$$2^{n-d} \leqslant \mathrm{wt}(f) \leqslant 2^n - 2^{n-d}.$$

## 2. Невырожденность булевых функций

Рассмотрим соотношение между параметром $AD(f)$ и подпространством $J(f)$.

**Утверждение 2.** Для любой булевой функции $f \in \mathcal{F}_n$ справедливо равенство

$$AD(f) = \dim J(f).$$

***Доказательство.*** Пусть $\dim J(f) = n - k$. Пусть $C = J(f)$ и $u^{(1)}, ..., u^{(k)}$ — базис пространства $C^{\perp}$. Пусть $D$ — $(n \times k)$-матрица, столбцами которой являются векторы

$u^{(1)}, ..., u^{(k)}$. По утверждению 1 функция $f$ может быть представлена следующим образом: $f = I_{C\oplus z^{(1)}} \oplus ... \oplus I_{C\oplus z^{(t)}}$, где $t \leqslant 2^k$. Пусть функция $g \in \mathcal{F}_k$ такова, что $g(x) = 1$ тогда и только тогда, когда существует номер $i = 1, ..., t$, что $x = z^{(i)}D$.

Покажем, что $f(x) = g(xD)$ для любого вектора $x \in V_n$. Поскольку множество $1_f$ представимо в виде объединения смежных классов $C \oplus z^{(1)}, ..., C \oplus z^{(t)}$, то для векторов $x \in 1_f$ и только для них справедливо представление $x = z^{(i)} \oplus c$, где $i \in \{1, ..., t\}$, $c \in C$. Поэтому верно соотношение $g(xD) = g(z^{(i)}D \oplus cD) = g(z^{(i)}D) = 1 = f(x)$. Таким образом, $AD(f) \geqslant \dim J(f)$.

Пусть $AD(f) = n-k$ и $D$ — такая $(n\times k)$-матрица ранга $k$, столбцами которой являются векторы $u^{(1)}, u^{(2)}, ..., u^{(k)}$ пространства $V_n$, что $f(x) = g(xD)$. Пусть $C$ — линейное подпространство пространства $V_n$ с базисом $u^{(1)}, ..., u^{(k)}$. Тогда для любого $u \in C^\perp$ и любого $x \in V_n$ выполнено $f(x \oplus u) = g((x \oplus u)D) = g(xD \oplus uD) = g(xD) = f(x)$. Таким образом, $C^\perp \subseteq J(f)$ и $AD(f) = \dim C^\perp \leqslant \dim J(f)$.

Учитывая ранее доказанное обратное неравенство, получаем $AD(f) = \dim J(f)$. ■

В работе [3] были введены следующие меры нелинейности булевых функций:

$$\mathrm{nl}(f) = \mathrm{dist}(f, \mathcal{A}_n) = 2^{n-1} - \frac{1}{2}\max_{u\in V_n} |W_f(u)|;$$
$$\delta(f) = \mathrm{dist}(f, \mathcal{F}_n^0) = 2^{n-2} - \frac{1}{4}\max_{u\in V_n^*} |\Delta_f(u)|.$$

Будем называть *невырожденностью* функции $f \in \mathcal{F}_n$ значение

$$\rho(f) = \mathrm{dist}(f, DG(n)).$$

Заметим, что параметры $AD(f)$, $\mathrm{nl}(f)$, $\delta(f)$ и $\rho(f)$ являются аффинными инвариантами.

**Утверждение 3.** Пусть $f \in \mathcal{F}_n$ и $u$ — произвольный вектор из $V_n^*$. Если функция $g \in DG(n)$ такова, что $u \in J(g)$ и $\mathrm{dist}(f, g) = \min\limits_{g':u\in J(g')} \mathrm{dist}(f, g')$, то $g$ представима в виде $g = f \cdot f^u \oplus h$, где для функции $h \in \mathcal{F}_n$ выполняется условие $D_u f \cdot h = h$ и $u \in J(h)$.

***Доказательство.*** Покажем, что $1_g \subseteq 1_{f\vee f^u}$ и $1_{f\cdot f^u} \subseteq 1_g$.

Пусть $h_1 = g((f \vee f^u) \oplus \overline{1})$. Тогда $u \in J(h_1)$. Но $\mathrm{dist}(g \oplus h_1, f) \leqslant \mathrm{dist}(g, f)$ и $u \in J(g \oplus h_1)$. Поэтому $h_1 = \overline{0}$, т. е. $1_g \subseteq 1_{f\vee f^u}$.

Пусть $h_2 = (g \oplus \overline{1})(f \cdot f^u)$. Тогда $u \in J(h_2)$ и $u \in J(g \oplus h_2)$. Но $\mathrm{dist}(g \oplus h_2, f) \leqslant$ $\leqslant \mathrm{dist}(g, f)$. Поэтому $h_2 = \overline{0}$, т. е. $1_{f\cdot f^u} \subseteq 1_g$.

Поскольку $1_{f\cdot f^u} \subseteq 1_g$, то $g$ представима в виде $g = f \cdot f^u \oplus h$, где $h \cdot f \cdot f^u = \overline{0}$ и $u \in J(h)$. Поскольку $1_g \subseteq 1_{f\vee f^u}$, а $f \vee f^u = f \cdot f^u \oplus D_u f$, то для функции $h$ выполнено $h \cdot D_u f = h$. Показанные соотношения доказывают утверждение. ■

**Следствие 1.** Для любой булевой функции $f \in \mathcal{F}_n$ справедливо соотношение

$$\rho(f) = \min_{u\in V_n^*} \frac{\mathrm{wt}(D_u f)}{2} = 2^{n-2} - \frac{1}{4}\max_{u\in V_n^*} \Delta_f(u).$$

***Доказательство.*** Из определения параметра $\rho$ следует, что справедливо равенство $\rho(f) = \min\limits_{u\in V_n^*} \min\limits_{g':u\in J(g')} \mathrm{dist}(f, g')$. Вычислим $\mathrm{dist}(f, g)$ для тех функций $g$, для которых существует ненулевой вектор $u \in J(g)$ и $\mathrm{dist}(f, g) = \min\limits_{g':u\in J(g')} \mathrm{dist}(f, g')$. Из утверждения 3 следует, что $g = f \cdot f^u \oplus h$, где $u \in J(h)$ и $h \cdot D_u f = h$. Для функции $h$

выполняется равенство $h = h_1 \oplus h_2$, где $h_1 \cdot h_2 = \overline{0}$, $h_1^u = h_2$ и справедливы соотношения $h_1(f \oplus f \cdot f^u) = h_1$ и $h_2(f \oplus f \cdot f^u) = \overline{0}$. Поэтому $\operatorname{dist}(f, g) = \operatorname{wt}(f \oplus f \cdot f^u \oplus h_1 \oplus h_2) = \operatorname{wt}(f) - \operatorname{wt}(f \cdot f^u) - \operatorname{wt}(h_1) + wh(h_2) = \operatorname{wt}(D_u f)/2$. Это соотношение доказывает первое равенство. Второе равенство следует из первого и из того, что

$$\Delta_f(u) = W_{D_u f}(0^n) = 2^n - 2\operatorname{dist}(D_u f, \overline{0}) = 2^n - 2\operatorname{wt}(D_u f).$$

Следствие доказано. ■

**Следствие 2.** Пусть $f \in \mathcal{F}_n$. Если $g \in DG(n)$ и $\operatorname{dist}(f, g) = \rho(f)$, то для любого вектора $u \in J^*(g)$ справедливо равенство $\Delta_f(u) = \max\{\Delta_f(v) : v \in V_n^*\}$.

***Доказательство.*** Пусть $u$ — произвольный вектор из $J^*(g)$. Тогда, по утверждению 3, функция $g$ представима в виде $g = f \cdot f^u \oplus h_u$. Отсюда и из следствия 1 получаем, что

$$\operatorname{dist}(f, g) = \frac{\operatorname{wt}(D_u f)}{2} = 2^{n-2} - \frac{1}{4}\Delta_f(u) = \rho(f) = 2^{n-2} - \frac{1}{4}\max_{v \in V_n^*}\Delta_f(v).$$

Следовательно, $\Delta_f(u) = \max\limits_{v \in V_n^*}\Delta_f(v)$. ■

**Утверждение 4.** Для любой булевой функции $f$ от $n \geqslant 2$ переменных справедливо неравенство $\rho(f) \leqslant 2^{n-2}$.

***Доказательство.*** Предположим, что для некоторой функции $f \in \mathcal{F}_n$ и для любого ненулевого вектора $u \in V_n$ справедливо $\Delta_f(u) < 0$, а поскольку числа $\Delta_f(u)$ четны, то $\Delta_f(u) \leqslant -2$.

Из равенства $\sum\limits_{u \in V_n}\Delta_f(u) = W_f^2(0^n)$ следует $\sum\limits_{u \in V_n}\Delta_f(u) \geqslant 0$. Но $\Delta_f(0^n) = 2^n$, а по предположению $\Delta_f(u) \leqslant -2$, поэтому

$$\sum_{u \in V_n}\Delta_f(u) \leqslant 2^n - 2(2^n - 1) = 2 - 2^n.$$

Следовательно, $\sum\limits_{u \in V_n}\Delta_f(u) < 0$ при $n \geqslant 2$. Полученное противоречие показывает, что существует такой ненулевой вектор $u \in V_n$, что $\Delta_f(u) \geqslant 0$, а значит, $\rho(f) \leqslant 2^{n-2}$. ■

Заметим, что поскольку выполнены включения $\mathcal{A}_n \subset DG(n) \subset \mathcal{F}_n^0$, то

$$\operatorname{nl}(f) \geqslant \rho(f) \geqslant \delta(f).$$

Следующий пример показывает, что высокая нелинейность не гарантирует высокой невырожденности булевой функции.

**Пример 1.** Рассмотрим функцию $f_{10,2}^4 \in \mathcal{F}_{10}$, построенную в работе [7]:

$$\begin{aligned} f_{10,2}^4(x) \;=\; & (x_9 \oplus x_{10} \oplus 1)(x_1x_3 \oplus x_1x_4 \oplus x_2x_3 \oplus x_2x_4 \oplus x_1 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_8) \oplus \\ & \oplus (x_9 \oplus x_{10})(x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5x_7 \oplus x_5x_8 \oplus x_6x_7 \oplus x_6x_8 \oplus x_5 \oplus x_7) \oplus x_9. \end{aligned}$$

Функция $f_{10,2}^4$ является 6-устойчивой булевой функцией, нелинейность которой равна $2^9 - 2^7 = 384$. Однако $\rho(f_{10,2}^4) = \delta(f_{10,2}^4) = 0$, т. е. функция является алгебраически вырожденной, причем $AD(f_{10,2}^4) = 3$, а $\dim L_{f_{10,2}^4} = 4$.

Рассмотрим вопрос достижимости параметром $\rho$ своего максимального значения $2^{n-2}$. Из критерия Ротхауза следует, что для любой бент-функции $f$ справедливо соотношение $\rho(f) = 2^{n-2}$. Для нечетного числа переменных справедлива

**Теорема 4.** Пусть $n$ нечетно, функция $f \in \mathcal{F}_n$ является платовидной функцией порядка $n-1$ и $\{u \in V_n : W_f(u) \neq 0\} = \{u \in V_n : <u,w> = 1\}$ для некоторого ненулевого вектора $w$. Тогда $\rho(f) = 2^{n-2}$.

***Доказательство.*** Рассмотрим соотношение между автокорреляционными коэффициентами и коэффициентами Уолша — Адамара функции $f$ для ненулевого вектора $u$. Поскольку функция $f$ является платовидной функцией порядка $n-1$, то в том случае, когда $W_f^2(u) \neq 0$, справедливо равенство $W_f^2(u) = 2^{2n-(n-1)} = 2^{n+1}$. Следовательно,

$$\Delta_f(u) = \frac{1}{2^n}\sum_{v\in V_n} W_f^2(v)(-1)^{<u,v>} = \frac{2^{n+1}}{2^n}\sum_{v:<v,w>=1}(-1)^{<u,v>} = \begin{cases} 0, & \text{если } u \neq w, \\ -2^n, & \text{если } u = w. \end{cases}$$

Таким образом, $\max\limits_{u\in V_n^*} \Delta_f(u) = 0$ и $\rho(f) = 2^{n-2}$. ■

В работе [8] было показано, что функции, удовлетворяющие условиям теоремы 4, представимы в виде $f = g^B$, где $B$ — некоторая невырожденная $(n\times n)$-матрица над $F_2$ и $g(x) = x_n \oplus h(x_1, ..., x_{n-1})$, где $h$ — бент-функция от $n-1$ переменной.

Заметим, что проблема исчерпывающего описания множества функций, для которых параметр $\rho$ достигает своего максимума, пока не решена.

## 3. Порядок алгебраической вырожденности наилучших аппроксимаций

Будем обозначать множество алгебраически вырожденных функций, наилучшим образом приближающих некоторую булеву функцию $f$, через

$$DG(f) = \{g \in DG(n) : \operatorname{dist}(f,g) = \rho(f)\}.$$

Множество функций, имеющих нетривиальную линейную структуру и наилучшим образом приближающих функцию $f$, обозначим через

$$\mathcal{F}_n^0(f) = \{g \in \mathcal{F}_n^0 : \operatorname{dist}(f,g) = \delta(f)\}.$$

Рассмотрим следующие функционалы:

$$\pi_\rho(f) = \max_{g\in DG(f)} \dim J(g) = \max_{g\in DG(f)} AD(g);$$
$$\pi_\delta(f) = \max_{g\in \mathcal{F}_n^0(f)} \dim L_g.$$

Алгебраически вырожденные функции, наилучшим образом аппроксимирующие данную функцию $f$, находятся от нее на расстоянии $\rho(f)$. Однако на этом расстоянии могут находиться функции с разными значениями порядка алгебраической вырожденности. С точки зрения криптоанализа нужно аппроксимировать функцию $f$ с помощью наиболее алгебраически вырожденной функции. Максимальное значение порядка алгебраической вырожденности тех функций, которые находятся от $f$ на расстоянии $\rho(f)$, отражает параметр $\pi_\rho(f)$. Содержательный смысл параметра $\pi_\delta(f)$ аналогичен.

**Определение 9.** Пусть $f \in \mathcal{F}_n$; *$\pi_\rho$-функцией для $f$* назовем функцию $g_\rho \in DG(f)$, для которой $AD(g_\rho) = \pi_\rho(f)$.

Аналогично определяется понятие *$\pi_\delta$-функции для $f$*.

Используя введенные понятия, можно следующим образом описать связь параметров $\delta$, $\pi_\delta$, $\rho$, $\pi_\rho$.

**Утверждение 5.** Если $\rho(f) = \delta(f)$, то $\pi_\delta(f) \in \{\pi_\rho(f), \pi_\rho(f)+1\}$. Если $\rho(f) > \delta(f)$, то $\pi_\delta(f) = 1$.

***Доказательство.*** Пусть функции $g_\rho$ и $g_\delta$ являются соответственно $\pi_\rho$-функцией и $\pi_\delta$-функцией для $f$.

Пусть $\rho(f) = \delta(f)$. Тогда $\pi_\delta(f) \geqslant \pi_\rho(f)$, так как иначе $\dim L_{g_\rho} > \dim L_{g_\delta}$, что противоречит условию. Заметим также, что $\dim J(g_\delta) \leqslant \dim J(g_\rho)$. Тот факт, что для любой функции $h$ справедливо $\dim L_h \in \{\dim J(h), \dim J(h)+1\}$, показывает справедливость первой импликации.

Если же $\rho(f) > \delta(f)$, то $\dim J(g_\delta) = 0$, а $L_{g_\delta} = \{0^n, u\}$, где $D_u\, g_\delta = \overline{1}$, т. е. $\pi_\delta(f) = 1$. Утверждение доказано. ■

Заметим, что для функции $f^4_{10,2}$ из примера 1 справедливы соотношения

$$\rho(f^4_{10,2}) = \delta(f^4_{10,2}) \quad \text{и} \quad \pi_\delta(f^4_{10,2}) = \pi_\rho(f^4_{10,2}) + 1,$$

а для функции $f$, удовлетворяющей условиям теоремы 4, справедливы равенства

$$\delta(f) = 0 \quad \text{и} \quad \pi_\delta(f) = 1.$$

Исходя из доказанного утверждения, можно сказать, что при равенстве значений $\rho(f)$ и $\delta(f)$, вычислив один из параметров $\pi_\delta$ и $\pi_\rho$, можно достаточно точно оценить значение второго. Однако когда $\rho(f) \neq \delta(f)$, значение $\pi_\delta(f)$ равно 1, тогда как $\pi_\rho(f)$ необходимо вычислять. В связи с этим далее большее внимание будет уделяться именно параметру $\pi_\rho$.

**Утверждение 6.** Пусть $f \in \mathcal{F}_n$. Если $|\mathrm{wt}(f) - 2^{n-1}| > \rho(f)$, то $\rho(f) = \delta(f)$ и $\pi_\rho(f) = \pi_\delta(f)$.

***Доказательство.*** Непосредственно следует из того факта, что любая функция $h$, для которой существует вектор $u \in V^*_n$, такой, что $D_u\, h = \overline{1}$, является уравновешенной, и из определений соответствующих функционалов. ■

**Теорема 5.** Для любой булевой функции $f \in \mathcal{F}_n$, для которой $\rho(f) > 0$, справедливо соотношение

$$\log_2 \rho(f) + \pi_\rho(f) \leqslant n.$$

***Доказательство.*** Пусть $g$ — некоторая $\pi_\rho$-функция для $f$. Тогда справедливо соотношение $f = g \oplus h$, где $\mathrm{wt}(h) = \rho(f)$. Предположим, что для некоторого вектора $u \in J(g)$ справедливо $h \cdot h^u \neq \overline{0}$. Тогда, переписав соотношение для функции $f$ в виде $f = (g \oplus h \cdot h^u) \oplus (h \oplus h \cdot h^u)$, можно заметить, что

$$\mathrm{dist}(f, g \oplus h \cdot h^u) = \mathrm{wt}(h \oplus h \cdot h^u) < \mathrm{wt}(h) = \rho(f).$$

Но это противоречит условию, поскольку функция $g \oplus h \cdot h^u$ является алгебраически вырожденной.

Получаем, что $\mathrm{wt}(D_u\, h) = 2\mathrm{wt}(h)$ для любого ненулевого вектора $u \in J(g)$. Но максимальный вес такой функции $h$ совпадает с мощностью фактор-пространства $V_n/J(g)$. Следовательно, $\rho(f) = \mathrm{wt}(h) \leqslant 2^{n-AD(g)} = 2^{n-\pi_\rho(f)}$. Утверждение теоремы получается путем логарифмирования обоих частей неравенства. ■

Заметим, что если сравнивать полученное неравенство с неравенством Зигенталера с точки зрения оптимизации криптографических параметров, то можно заметить следующее различие. С точки зрения стойкости криптографических примитивов нужно

максимизировать и $\deg(f)$, и $\mathrm{cor}(f)$. Неравенство Зигенталера показывает, что эти два параметра вступают в противоречие друг с другом. Исходя из тех же соображений, необходимо максимизировать $\rho(f)$, но минимизировать $\pi_\rho(f)$. Полученное в теореме 5 неравенство показывает, что эти два параметра в каком-то смысле согласованы.

**Следствие 3.** Пусть функция $f \in \mathcal{F}_n$ такова, что $\mathrm{nl}(f) = \rho(f)$. Тогда справедливо неравенство $\mathrm{nl}(f) = \rho(f) \leqslant 2$.

***Доказательство.*** Если $\mathrm{nl}(f) = \rho(f)$, то существует такая аффинная функция $g$, что $\mathrm{dist}(f, g) = \rho(f)$. Для этой функции справедливо соотношение $AD(g) \geqslant n - 1$. Отсюда $\pi_\rho(f) \geqslant n - 1$. Из утверждения теоремы 5 заключаем, что $\log_2 \rho(f) \leqslant 1$, т. е. $\rho(f) \leqslant 2$. ■

Следующие две леммы описывают некоторые классы булевых функций, для которых в неравенстве теоремы 5 достигается равенство.

**Лемма 2.** Пусть функция $f \in \mathcal{F}_n$ такова, что $\pi_\rho(f) = k > n/2$. Тогда для функции $f$ выполнено равенство $\log_2 \rho(f) + \pi_\rho(f) = n$ тогда и только тогда, когда $f$ представима в виде $f = g \oplus h$, где $AD(g) = k$, $\mathrm{wt}(h) = 2^{n-k}$ и для любых различных векторов $u, v \in 1_h$ справедливо $u \oplus v \notin J(g)$.

***Доказательство.*** Необходимость. Доказательство аналогично теореме 5.

Достаточность. Пусть $J(g) = L$. Заметим, что для любого вектора $u \in L^*$ справедливо $\mathrm{wt}\, D_u f = \mathrm{wt}\,(D_u g \oplus D_u h) = \mathrm{wt}\, D_u h = 2 \cdot 2^{n-k} < 2 \cdot 2^{\frac{n}{2}}$.

Пусть $u \in V_n \setminus L$. Тогда $\mathrm{wt}\, D_u f = \mathrm{wt}\,(D_u g \oplus D_u h) \geqslant \mathrm{wt}\, D_u g - \mathrm{wt}\, D_u h$. Поскольку $\mathrm{wt}\, D_u g \geqslant 2 \cdot 2^k$, а $\mathrm{wt}\, D_u h \leqslant 2 \cdot 2^{n-k}$, то

$$\mathrm{wt}\, D_u f \geqslant 2(2^k - 2^{n-k}) > 2(2 \cdot 2^{\frac{n}{2}} - 2^{\frac{n}{2}}) > 2 \cdot 2^{\frac{n}{2}}.$$

Учитывая неравенство, полученное для векторов из $L^*$, получаем, что для любых векторов $u \in L^*$ и $v \in V_n \setminus L$ справедливо $\mathrm{wt}\, D_u f < \mathrm{wt}\, D_v f$. Таким образом,

$$\rho(f) = \frac{1}{2}\mathrm{wt}\, D_u f = 2^{n-k} = \mathrm{dist}(f, g).$$

Тот факт, что $\pi_\rho(f) = AD(g) = k$, следует из неравенства $\log_2 \rho(f) + \pi_\rho(f) \leqslant n$. ■

**Лемма 3.** Пусть натуральные числа $n$ и $k$ удовлетворяют условиям $n \geqslant 3$, $2 \leqslant k \leqslant n/2$. Пусть $A$ — $(n \times (n-k))$-матрица над $F_2$ ранга $n-k$ и $L$ — подпространство пространства $V_n$, ортогональное к подпространству, порожденному набором из $n - k$ столбцов матрицы $A$. Пусть если число $(n-k)$ четно, то $g$ является бент-функцией от $n - k$ переменных, иначе пусть функция $g \in \mathcal{F}_{n-k}$ удовлетворяет условиям теоремы 4. Пусть функция $h \in \mathcal{F}_n$ такова, что $\mathrm{wt}(h) = 2^{n-k}$ и носитель $1_h$ является такой плоскостью пространства $V_n$, что для любых двух ненулевых векторов $u, v \in 1_h$ справедливо соотношение $u \oplus v \notin L$. Тогда для функции $f = g^A \oplus h$ справедливы соотношения

$$\rho(f) = 2^{n-k}, \quad \pi_\rho(f) = k.$$

***Доказательство.*** Пусть $(n - k)$ четно. Если $u \in L^*$, то

$$\mathrm{wt}\, D_u f = \mathrm{wt}\,(D_u g^A \oplus D_u h) = \mathrm{wt}\, D_u h = 2 \cdot \mathrm{wt}(h) = 2 \cdot 2^{n-k}.$$

Пусть теперь $u \in V_n \setminus L$. Если $u \in J(h)$, то

$$\mathrm{wt}\, D_u f = \mathrm{wt}\,(D_u g^A \oplus D_u h) = \mathrm{wt}\, D_u g^A = 2^{n-1}.$$

Последнее равенство справедливо, поскольку функция $g$ является бент-функцией от $n-k$ переменных и $D_u g^A = (D_v g)^A$ для некоторого $v \in V_{n-k}$.

Если же $u \notin J(h)$, то

$$\operatorname{wt} D_u f = \operatorname{wt}(D_u g^A \oplus D_u h) = \operatorname{wt} D_u g^A + \operatorname{wt} D_u h - 2 \cdot \operatorname{wt}(D_u g^A \cdot D_u h).$$

Заметим, что для любого $z \in V_n \setminus L$ справедливо равенство $\operatorname{wt}(I_{L \oplus z} \cdot D_u h) = 2$. Поэтому $\operatorname{wt}(D_u g^A \cdot D_u h) = 2^{n-1-k} \cdot 2$ и

$$\operatorname{wt} D_u f = 2^{n-1} + 2^{n-k+1} - 2 \cdot 2^{n-k} = 2^{n-1}.$$

Получили, что

$$\operatorname{wt} D_u f = \begin{cases} 2^{n-1}, \text{ если } u \notin L, \\ 2^{n-k+1}, \text{ если } u \in L^*. \end{cases}$$

Следовательно, $\rho(f) = 2^{n-k}$. А поскольку $\rho(f) = \operatorname{dist}(f, g)$ и $AD(g) = k$, то из теоремы 5 следует, что $\pi_\rho(f) = k$.

Пусть теперь число $(n-k)$ нечетно. Поскольку функция $g$ удовлетворяет условиям теоремы 4, то справедливо соотношение $\operatorname{wt}(D_u g^A) \in \{0, 2^{n-1}, 2^n\}$ для любого вектора $u \in V_n^*$. Пусть вектор $u \in V_n^*$ таков, что $\operatorname{wt}(D_u g^A) = 2^n$. Тогда $\operatorname{wt}(D_u f) \geqslant 2^n - 2^{n-k+1} \geqslant$ $\geqslant 2^{n-k+1}$. Рассмотрение оставшихся случаев аналогично случаю, когда $(n-k)$ четно. В результате получаем, что $\rho(f) = 2^{n-k}$ и $\pi_\rho(f) = k$. ■

**Пример 2.** Примером использования конструкции леммы 3 может служить функция $f_8$ от 8 переменных, которая задается в виде алгебраической нормальной формы следующим образом:

$$\begin{aligned} f_8(x) \;=\; & x_5x_6x_7x_8 \oplus x_7x_8 \oplus x_5x_6x_8 \oplus x_5x_6x_7 \oplus x_5x_8 \oplus x_5x_7x_8 \oplus x_6x_7 \oplus x_6x_7x_8 \oplus \\ & \oplus x_1x_2 \oplus x_3x_4 \oplus x_6x_8 \oplus x_5x_7 \oplus x_5x_6 \oplus x_3 \oplus x_8 \oplus x_7 \oplus x_5 \oplus x_6 \oplus 1. \end{aligned}$$

Параметры функции $f_8$ имеют значения

$$\deg(f_8) = 4, \quad \operatorname{nl}(f_8) = 100, \quad \rho(f_8) = \delta(f_8) = 16, \quad \pi_\rho(f_8) = 4, \quad \operatorname{wt}(f_8) = 100.$$

Функция $f_8$ представима в виде $f_8 = g^A \oplus h$. Функция $g = x_2 \oplus x_1x_2 \oplus x_3x_4$ является бент-функцией от 4 переменных. Носитель функции $h$ является подпространством пространства $V_8$ размерности 4 с базисом $u^{(1)} = (00000001)$, $u^{(2)} = (00000010)$, $u^{(3)} = (00000100)$, $u^{(4)} = (00001000)$. Эти же векторы являются столбцами $(8 \times 4)$-матрицы $A$.

Рассмотрим теперь некоторые соотношения, которые связывают параметры $\rho$ и $\pi_\rho$ булевой функции с другими ее важными характеристиками.

**Теорема 6.** Для любой функции $f \in \mathcal{F}_n$, такой, что $\pi_\rho(f) > n/2$, существует такое целое число $T$, $0 \leqslant T \leqslant 2^{n-\pi_\rho(f)}$, что справедливо неравенство

$$T\,(2^{\pi_\rho(f)} - 1) \leqslant \operatorname{wt}(f) \leqslant T\,(2^{\pi_\rho(f)} - 1) + 2^{n-\pi_\rho(f)}.$$

***Доказательство.*** Пусть $g$ — некоторая $\pi_\rho$-функция для $f$. Тогда для любого смежного класса $J(g) \oplus z$ справедливо $|1_f \cap J(g) \oplus z| \in \{0, 1, 2^{\pi_\rho(f)} - 1, 2^{\pi_\rho(f)}\}$. Вес функции $f$ получается суммированием мощностей множеств $|1_f \cap J(g) \oplus z|$ по всем различным смежным классам $J(g) \oplus z$. Утверждение теоремы следует из того, что количество различных смежных классов по подпространству $J(g)$ равно $2^{n-\pi_\rho(f)}$. ■

**Теорема 7.** Пусть $f \in \mathcal{F}_n$. Если $\pi_\rho(f) > \log_2 \rho(f)$, то $\deg f \geqslant \pi_\rho(f)$.

***Доказательство.*** Пусть функция $f \in \mathcal{F}_n$ представима в виде $f = g \oplus h$, где $AD(g) = \pi_\rho(f) > \log_2 \mathrm{wt}(h) = \log_2 \rho(f)$. Тогда справедливы неравенства

$$2^{n-\deg h} \leqslant \mathrm{wt}(h) \leqslant 2^n - 2^{n-\deg h} \quad \text{и} \quad \deg g \leqslant n - \pi_\rho(f).$$

Отсюда получаем $\deg h \geqslant n - \log_2 \rho(f)$ и $n - \pi_\rho(f) \geqslant \deg g$. Складывая эти два неравенства, получим, что $\deg h - \deg g \geqslant \pi_\rho(f) - \log_2 \rho(f) > 0$, т. е. $\deg h > \deg g$.

Поскольку $\deg f = \max(\deg g, \deg h)$ при $\deg g \neq \deg h$, то $\deg f = \deg h \geqslant \geqslant n - \log_2 \rho(f) \geqslant \pi_\rho(f)$. ■

Неравенство, полученное в теореме 6 и связывающее вес функции со значением параметра $\pi_\rho$, схоже с двусторонним неравенством из леммы 1. Комбинирование этих двух неравенств позволяет получать соотношения между различными параметрами булевой функции.

## 4. Свойства $\pi_\rho$-функций и вычисление параметра $\pi_\rho$

Рассмотрим свойства $\pi_\rho$-функций для некоторой функции $f$.

**Теорема 8.** Пусть $f \in \mathcal{F}_n$ и $g \in DG(f)$. Если существуют два таких различных вектора $u, v \in V_n^*$, что $u, v \in J(g)$, то функция $g$ представима в виде

$$g = f \cdot f^u \oplus f \cdot f^v \oplus f^u \cdot f^v.$$

***Доказательство.*** Поскольку $f^u \cdot f^v = (f \cdot f^{u \oplus v})^v$, то из следствия 2 следует, что $\mathrm{wt}(f \cdot f^u) = \mathrm{wt}(f \cdot f^v) = \mathrm{wt}(f^u \cdot f^v)$. Из утверждения 3 следует, что функция $g$ представима в виде

$$g = f \cdot f^u \oplus h_u \oplus h_u^u, \text{ где } h_u \cdot f = h_u \text{ и } h_u^u \cdot f = \overline{0},$$
$$g = f \cdot f^v \oplus h_v \oplus h_v^v, \text{ где } h_v \cdot f = h_v \text{ и } h_v^v \cdot f = \overline{0}.$$

Умножим обе части первого равенства на функцию $f \cdot f^v \oplus f \cdot f^u \cdot f^v$:

$$g\,(f \cdot f^v \oplus f \cdot f^u \cdot f^v) = f \cdot f^v \oplus f \cdot f^u \cdot f^v = (f \cdot f^v \oplus f \cdot f^u \cdot f^v)\, h_u.$$

Отсюда получаем, что $\mathrm{wt}(h_u) \geqslant \mathrm{wt}(f \cdot f^v \oplus f \cdot f^u \cdot f^v)$.

Воспользовавшись вторым представлением функции $g$, получим

$$g \cdot h_u^u = h_u^u = h_v \cdot h_u^u \oplus h_u^u \cdot h_v^v = h_v \cdot f \cdot h_u^u \oplus h_u^u \cdot h_v^v \cdot f^v = h_u^u \cdot f^v \cdot h_v^v.$$

Таким образом, справедливо равенство $h_u^u = h_u^u \cdot f^v$. Теперь умножим обе части первого представления функции $g$ на функцию $f^u \cdot f^v \oplus f \cdot f^u \cdot f^v$:

$$g\,(f^u \cdot f^v \oplus f \cdot f^u \cdot f^v) = h_u^u \cdot f^u \cdot f^v \oplus h_u^u \cdot f \cdot f^u \cdot f^v = h_u^u \cdot f^v = h_u^u.$$

Отсюда получаем $\mathrm{wt}(f^u \cdot f^v \oplus f \cdot f^u \cdot f^v) \geqslant \mathrm{wt}(h_u^u)$. Поскольку $\mathrm{wt}(h_u^u) = \mathrm{wt}(h_u)$ и $\mathrm{wt}(f^u \cdot f^v \oplus f \cdot f^u \cdot f^v) = \mathrm{wt}(f \cdot f^v \oplus f \cdot f^u \cdot f^v)$, то $\mathrm{wt}(h_u) = \mathrm{wt}(f \cdot f^v \oplus f \cdot f^u \cdot f^v)$, а значит, верны равенства $h_u = f \cdot f^v \oplus f \cdot f^u \cdot f^v$, $h_u^u = f^u \cdot f^v \oplus f \cdot f^u \cdot f^v$.

Переписав первое представление для функции $g$, учитывая полученные соотношения, получим $g = f \cdot f^u \oplus f \cdot f^v \oplus f^u \cdot f^v$. ■

Таким образом, вычисление значения $\pi_\rho(f)$ можно проводить по следующему принципу. Необходимо вычислить значение $\Delta_f^* = \max\limits_{u \in V_n^*} \Delta_f(u)$, одновременно сформировав

множество $A = \{u \in V_n^* : \Delta_f(u) = \Delta_f^*\}$. Вычислить значение $\pi_\rho(f)$ можно с помощью множества $B = \{g = f \cdot f^u \oplus f \cdot f^v \oplus f^u \cdot f^v : u, v \in A,\ u \neq v,\ \mathrm{dist}(f, g) = \rho(f)\}$ по формуле

$$\pi_\rho(f) = \max\{1, \max_{g \in B} \dim J(g)\}.$$

## 5. Вырожденность бент-функций

Рассмотрим свойства параметров $\rho$ и $\pi_\rho$ для случая, когда $f$ является бент-функцией.

**Лемма 4.** Пусть $f \in \mathcal{F}_n$ является бент-функцией и $g \in DG(f)$. Если существуют два различных ненулевых вектора $u, v \in V_n$, такие, что $u, v \in J(g)$, то функция $g$ представима в виде $g = D_u(f \cdot f^v)$.

***Доказательство.*** По теореме 8 справедливо равенство $g = f \cdot f^u \oplus f \cdot f^v \oplus f^u \cdot f^v$. По условию $g = g^u$. Используя представление функции $g$, получим равенство

$$f \cdot f^v \oplus f^u \cdot f^v = f^u \cdot f^{u \oplus v} \oplus f \cdot f^{u \oplus v}.$$

Преобразовав его, получаем $D_u f\,(D_u f)^v = \overline{0}$. Учитывая, что производная бент-функции по любому направлению уравновешена, это соотношение можно переписать в виде $D_u f \oplus \overline{1} = (D_u f)^v$. Умножая обе части на $f$, получаем $f\,(\overline{1} \oplus f \oplus f^u) = (f^v \oplus f^{u \oplus v})\, f$, $f \cdot f^u \oplus f \cdot f^v \oplus f \cdot f^{u \oplus v} = \overline{0}$. Используя представление функции $g$, запишем $g = f \cdot f^{u \oplus v} \oplus (f \cdot f^{u \oplus v})^u = D_u(f \cdot f^{u \oplus v})$. После переобозначения векторов имеем требуемое соотношение. ■

Теорема 8 и лемма 4 показывают, что $\pi_\rho$-функции для функции $f$ легко выражаются через функции вида $f \cdot f^u$. Некоторые свойства функций вида $f \cdot f^u$ для случая, когда функция $f$ является бент-функцией, отражены в следующем утверждении.

**Утверждение 7.** Для любой бент-функции $f \in \mathcal{F}_n$, где $n \geqslant 4$, и любого ненулевого вектора $u \in V_n$ справедливо соотношение $AD(f \cdot f^u) = 1$.

***Доказательство.*** По критерию Ротхауза, производная функции $f$ по любому ненулевому направлению уравновешена. Из соотношения $\mathrm{wt}(f \cdot f^u) = \mathrm{wt}(f) - \mathrm{wt}(D_u f) = 2^{n-2} \pm 2^{\frac{n}{2}-1}$ получаем, что для любых ненулевых векторов $u, v \in V_n$ справедливо равенство $\mathrm{wt}(f \cdot f^u) = \mathrm{wt}(f \cdot f^v)$.

Предположим, что утверждение не верно, т. е. $AD(f \cdot f^u) > 1$. Поскольку справедливо $AD(f \cdot f^u) = \dim J(f \cdot f^u)$, то это означает, что существует такой ненулевой вектор $v \in V_n$, отличный от $u$, что $(f \cdot f^u)^v = f \cdot f^u$. Учитывая равенство весов $\mathrm{wt}(f \cdot f^u) = \mathrm{wt}(f \cdot f^v)$, получаем $f \cdot f^u = f \cdot f^v = f \cdot f^{u \oplus v} = (f \cdot f^{u \oplus v})^u = f^u \cdot f^v$.

Используя полученные ранее равенства, имеем $f\,(f^u \oplus f^v) = \overline{0}$, откуда $f\,(D_{u \oplus v} f)^u = (f^u \cdot D_{u \oplus v} f)^u = \overline{0}$. Таким образом, выполнено соотношение $f^u \cdot D_{u \oplus v} f = \overline{0}$. Заменяя в рассуждениях вектор $u$ на $v$, получаем $f^v \cdot D_{u \oplus v} f = \overline{0}$.

Рассмотрим два случая. Если $\mathrm{wt}(f) = 2^{n-1} + 2^{\frac{n}{2}-1}$, то ни одно из полученных ранее соотношений не может быть выполнено, так как производная бент-функции по любому ненулевому направлению уравновешена.

Если же $\mathrm{wt}(f) = 2^{n-1} - 2^{\frac{n}{2}-1}$, то для выполнения двух полученных равенств необходимо, чтобы выполнялось неравенство $\mathrm{wt}(f^u) + \mathrm{wt}(f^v) - \mathrm{wt}(f^u \cdot f^v) + 2^{n-1} \leqslant 2^n$. Так как $f^u \cdot f^v = (f \cdot f^{u \oplus v})^u$, то $\mathrm{wt}(f^u \cdot f^v) = 2^{n-2} - 2^{\frac{n}{2}-1}$. Подставляя значения весов функций в неравенство, получаем

$$2^n + 2^{n-1} - 2^{n-2} - 2^{\frac{n}{2}-1} \leqslant 2^n,$$

$$2^{n-2} - 2^{\frac{n}{2}-1} \leqslant 0.$$

Но для $n \geqslant 4$ это неравенство не выполняется. Получили противоречие, которое доказывает утверждение. ■

Для бент-функций параметр $\rho$ достигает своего максимального значения $2^{n-2}$. С помощью конструкции, описанной в лемме 3, можно строить бент-функции, для которых в неравенстве теоремы 5 достигается равенство, т. е. бент-функции с $\pi_\rho(f) = 2$. Для этого достаточно задать четное число переменных $n$ и положить $k = 2$.

**Пример 3.** Функция $b_8$ является бент-функцией от восьми переменных и задается следующим образом:

$$\begin{aligned} b_8(x) \;=\; & x_4x_5x_6 \oplus x_2x_3x_5 \oplus x_1x_3x_4 \oplus x_7x_8 \oplus x_1x_5 \oplus x_3x_4 \oplus x_2x_4 \oplus x_1x_4 \oplus x_2x_3 \oplus \\ & \oplus x_1x_3 \oplus x_3x_6 \oplus x_8 \oplus x_7 \oplus 1. \end{aligned}$$

Параметры функции $b_8$ имеют следующие значения:

$$\deg(b_8) = 3, \quad \mathrm{nl}(b_8) = 120, \quad \rho(b_8) = \delta(b_8) = 64, \quad \pi_\rho(b_8) = 2, \quad \mathrm{wt}(b_8) = 120.$$

Функция $b_8$ представима в виде $b_8 = g^A \oplus h$. Функция

$$g(x) = x_2x_6 \oplus x_3x_6 \oplus x_1x_2x_3 \oplus x_4x_6 \oplus x_1x_4 \oplus x_3x_4 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_4x_5 \oplus x_2x_4x_5$$

является бент-функцией от шести переменных. Векторы $u^{(1)} = (00000001)$, $u^{(2)} = (00000010)$, $u^{(3)} = (00000100)$, $u^{(4)} = (00001000)$, $u^{(5)} = (00010000)$, $u^{(6)} = (00100000)$ являются столбцами $(8 \times 6)$-матрицы $A$, а также образуют базис подпространства, которое является носителем функции $h$.

## ЛИТЕРАТУРА


1. *Siegenthaler T.* Decrypting a class of stream chipher using ciphertext only // IEEE Trans. Computers. 1985. V. C-34. No. 1. P. 81–85.
2. *Matsui M.* Linear cryptanalysis method for DES cipher // LNCS. 1993. V. 765. P. 386–397.
3. *Meier W. and Staffelbach O.* Nonlinearity criteria for cryptographic functions // LNCS. 1990. V. 434. P. 549–562.
4. *Алексеев Е. К.* О некоторых алгебраических и комбинаторных свойствах корреляционно-иммунных булевых функций // Дискретная математика. 2010. Т. 22. № 3. С. 110–126.
5. *Dawson E. and Wu C. K.* Construction of correlation immune boolean functions // LNCS. 1997. V. 1334. P. 170–180.
6. *Логачев О. А., Сальников А. А., Ященко В. В.* Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2004.
7. *Тарасников Ю. В.* О корреляционно-иммунных и устойчивых булевых функциях // Математические вопросы кибернетики. Вып. 11. М.: Физматлит, 2002. С. 91–148.
8. *Zhang X. M. and Zheng Y.* Characterizing the structures of cryptographic functions satisfying the propagation criterion for almost all vectors // Design Codes Cryptography. 1996. No. 7 (1/2). P. 111–134.

# МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

DOI 10.17223/20710410/12/2
УДК 004.94

## ФОРМИРОВАНИЕ СЛОВАРЯ ТЕРМИНОВ ТЕОРИИ МОДЕЛИРОВАНИЯ БЕЗОПАСНОСТИ УПРАВЛЕНИЯ ДОСТУПОМ И ИНФОРМАЦИОННЫМИ ПОТОКАМИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ[1]

П. Н. Девянин

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

**E-mail:** peter_devyanin@hotmail.com

Рассматривается проект словаря терминов, применяемых в теории моделирования управления доступом и информационными потоками в компьютерных системах. Словарь разработан на основе существующих стандартов в области компьютерной безопасности, а также с использованием определений основных элементов классических моделей и ДП-моделей безопасности компьютерных систем.

**Ключевые слова:** *компьютерная безопасность, управление доступом, словарь терминов.*

### Введение

Существующие отечественные и зарубежные стандарты, учебные и научные издания, другие работы, в которых приводится терминология в области защиты информации (например, в [1 – 7] и др.), обеспечивают основу для стандартизации понятийного аппарата в данной области. В то же время содержащиеся в этих работах определения терминов часто отражают только организационно-правовые или общетехнические вопросы защиты информации, иногда в разных работах определения одних и тех же терминов не соответствуют друг другу. При этом крайне мало приводится терминов в области теории моделирования управления доступом и информационными потоками в компьютерных системах (КС), а имеющиеся термины часто не соответствуют ее современному уровню развития.

Формирование словаря терминов, применяемых в теории моделирования управления доступом и информационными потоками в КС, может оказать благотворное влияние на развитие данной теории, а также на процессы стандартизации понятийного аппарата и совершенствования образовательного процесса по направлению информационной безопасности.

Предлагаемый словарь следует рассматривать как промежуточный. Он подготовлен на основе существующих стандартов в области компьютерной безопасности, а также с использованием определений основных элементов классических моделей и ДП-моделей безопасности компьютерных систем [8].

В словаре содержатся определения терминов, разработанные автором или заимствованные (полностью или частично) из других работ. Для некоторых терминов да-

[1]Работа выполнена при поддержке гранта МД-2.2010.10

ется несколько определений. При этом в тексте словаря курсивом в определениях терминов выделены слова, соответствующие терминам, также содержащимся в словаре. Если определения термина заимствовано, то указывается источник заимствования.

### Словарь терминов

**Актив** —
1) информация или ресурсы КС, подлежащие защите (на основе [2]);
2) подлежащие защите в КС *сущности*, *субъекты* и *информационные потоки*.

**Атрибут безопасности** — информация, связанная с *субъектами*, *пользователями* и/или *сущностями* и используемая для осуществления *политики безопасности управления доступом и информационными потоками* (на основе [2]).

**Атрибут способа доступа к содержимому контейнера (*CCR*)** — в рамках моделей КС с *мандатным управлением доступом* атрибут *контейнеров*, принимающий значения `true` или `false` и задающий порядок доступа к его содержимому (`true` — с учетом *уровня конфиденциальности* всего контейнера; `false` — с учетом только уровня конфиденциальности *сущности* в составе контейнера, к которой непосредственно осуществляется доступ).

**Базовая теорема безопасности** — теорема, обосновывающая в рамках *формальных моделей мандатного управления доступом*, построенных на основе *классической модели Белла — ЛаПадулы*, достаточные (в ряде случаев и необходимые) условия *безопасности системы*, заключающиеся в требовании *безопасности* всех *состояний* на *траекториях системы*. При этом в условии теоремы накладываются ограничения либо на множество действий системы (например, в *классической модели Белла — ЛаПадулы*), либо на *функцию переходов системы* (например, в *модели СВС)*.

**Безопасность в смысле Белла — ЛаПадулы** — выполнение в *системе* с *мандатным управлением доступом* $*$*-свойства* и *ss-свойства* безопасности или их аналогов (в *классической модели Белла — ЛаПадулы* дополнительно требуется выполнение *ds-свойства*).

**Безопасность информации (данных)** — состояние защищенности *информации* (*данных*), при котором обеспечены ее (их) *конфиденциальность*, *доступность* и *целостность* [3].

**Граф доступов [прав доступа, информационных потоков]** — конечный помеченный ориентированный без петель граф, описывающий *состояние системы* в рамках *формальной модели*. Вершинами графа являются *сущности* (*объекты*), *субъекты* или *роли*. Каждое ребро соответствует *доступу*, *праву доступа* или *информационному потоку* от вершины, соответствующей началу ребра, к вершине, соответствующей концу ребра.

**Граф [прав] доступов в классической модели *Take-Grant*** — *граф доступов* в рамках *классической модели Take-Grant*. Вершинами графа являются *объекты* и *субъекты* (обозначаются соответственно $\otimes$ и $\bullet$). Каждое ребро помечено непустым подмножеством множества видов *прав доступа* и задает наличие данных прав доступа у объекта или субъекта, соответствующих началу ребра, к объекту или субъекту, соответствующих концу ребра.

**Граф [прав] доступов и информационных потоков в расширенной модели *Take-Grant*** — *граф доступов* в рамках *расширенной модели Take-Grant*. Вершинами графа являются *объекты* и *субъекты* (обозначаются соответственно $\otimes$ и $\bullet$). Каждое «реальное» ребро (обозначается сплошной линией со стрелкой) помечено непустым подмножеством множества видов *прав доступа* и задает наличие данных прав досту-

па у объекта или субъекта, соответствующих началу ребра, к объекту или субъекту, соответствующих концу ребра. Каждое «мнимое» ребро (обозначается прерывистой линией со стрелкой) помечено непустым подмножеством множества видов информационных потоков (на чтение или на запись) и задает наличие данных информационных потоков от объекта или субъекта, соответствующих началу ребра, к объекту или субъекту, соответствующих концу ребра.

**Граф прав доступа, доступов и информационных потоков в ДП-моделях** — *граф доступов* в рамках *ДП-моделей*. Вершинами графа являются *сущности* и *субъекты* (обозначаются соответственно $\otimes$ и $\bullet$). Каждое ребро, соответствующее праву доступа *субъекта* к *сущности* (обозначается сплошной линией со стрелкой), помечено элементом множества видов *прав доступа*. Каждое ребро, соответствующее доступу субъекта к сущности (обозначается двойной сплошной линией со стрелкой), помечено элементом множества видов *доступов*. Каждое ребро, соответствующее *информационному потоку по памяти* (обозначается прерывистой линией со стрелкой и помечено $write_m$) или *по времени* (обозначается линией точек со стрелкой и помечено $write_t$), задает наличие информационного потока соответствующего вида от сущности или субъекта, являющегося началом ребра, к сущности или субъекту, являющемуся концом ребра. В описание графа также входит *функция иерархии сущностей*.

**Граф создания в модели ТМД** — ориентированный граф, вершины которого соответствуют *типам*, в котором ребро от вершины $u$ к вершине $v$ существует тогда и только тогда, когда в системе имеется *команда*, в которой $u$ является *родительским типом*, а $v$ — *дочерним типом*.

**Данные** —

1) факты, понятия или команды, представленные в формализованном виде и позволяющие осуществлять их передачу или обработку как вручную, так и с помощью средств автоматизации [3];

2) конечные последовательности бит.

**Доступ** — способ обращения *субъекта* к *сущности* для выполнения операции над нею.

**Доступ блокирующий [информационные потоки по времени]** — в рамках *ДП-моделей* любой *доступ доверенного субъекта* к *сущности* (такой доступ препятствует реализации *информационных потоков по времени* с использованием данной сущности и *иерархии сущностей*, в которую она входит).

**Доступ владения** — *доступ субъекта* (*субъект-сессии*) к субъекту, позволяющий первому субъекту получить полный контроль над вторым субъектом, заключающийся в возможности использования его *прав доступа*, *ролей* и *доступов*.

**Доступ запрещенный/разрешенный** — *доступ субъекта* к *сущности*, запрещенный/разрешенный априорно заданной *политикой безопасности управления доступом* и *информационными потоками*.

**Доступы разрешенные совместные к сущности** — задаваемое *субъектами*, обладающими *доступами* к данной *сущности* в некотором *состоянии системы*, множество *видов доступа*, которыми к данной сущности в этом состоянии одновременно могут обладать субъекты.

**Доступность информации** —

1) состояние информации, при котором *субъекты*, имеющие *права доступа*, могут реализовать их беспрепятственно [6];

2) свойство КС, в которой обрабатывается информация, характеризующееся способностью КС обеспечивать своевременный *доступ субъектов*, имеющих на это право, к запрашиваемой ими информации.

**Замкнутая программная среда** —

1) программная среда КС, в которой разрешено порождение *субъектов* только для фиксированного множества пар *активизирующих субъектов* и *сущностей источников* (на основе [7]);

2) программная среда КС, в которой функционирует *монитор безопасности субъектов* (на основе [7]).

**Замыкание графа доступов [прав доступа, информационных потоков]** — *граф доступов*, полученный из исходного графа доступов в результате применения к нему конечной последовательности *правил преобразования графов доступов*, принадлежащих некоторому подмножеству правил, заданных в рамках *формальной модели*. При этом дальнейшее применение к результирующему графу данных правил не приводит к появлению в нем новых элементов (вершин, соответствующих *сущностям*, или ребер, соответствующих *доступам*, *правам доступа* или *информационным потокам*).

**Замыкание *tg* графа доступов [и информационных потоков] в рамках расширенной модели *Take-Grant*** — *граф доступов*, полученный из исходного графа доступов в результате создания для каждого его *субъекта объекта* с ребром к нему, помеченным *правами доступа*: $t$ (*take*), $g$ (*grant*), $r$ (*read*), $w$ (*write*), а также применения к полученному графу конечной последовательности *де-юре правил преобразования графов доступов* вида *take* и *grant*, приводящих к добавлению в граф новых ребер, помеченных $t$ или $g$. При этом дальнейшее применение к результирующему графу правил вида *take* и *grant* не приводит к появлению в нем новых ребер, помеченных $t$ или $g$.

**Замыкание де-факто графа доступов [и информационных потоков] в рамках расширенной модели *Take-Grant*** — *граф доступов*, полученный из исходного графа доступов в результате создания для каждого его *субъекта объекта* с ребром к нему, помеченным *правами доступа*: $t$ (*take*), $g$ (*grant*), $r$ (*read*), $w$ (*write*), а также применения к полученному графу конечной последовательности *де-юре правил преобразования графов доступов* вида *take*, *grant* и *де-факто правил*. При этом дальнейшее применение к результирующему графу данных правил не приводит к появлению в нем новых ребер.

**Замыкание де-юре графа доступов [и информационных потоков] в рамках расширенной модели *Take-Grant*** — *граф доступов*, полученный из исходного графа доступов в результате создания для каждого его *субъекта объекта* с ребром к нему, помеченным *правами доступа*: $t$ (*take*), $g$ (*grant*), $r$ (*read*), $w$ (*write*), а также применения к полученному графу конечной последовательности *де-юре правил преобразования графов доступов* вида *take* и *grant*. При этом дальнейшее применение к результирующему графу правил вида *take* и *grant* не приводит к появлению в нем новых ребер.

**Идентификатор** —

1) представление уполномоченного *пользователя* (например, строка символов), однозначно его идентифицирующее [2];

2) некоторое представление (например, строкой символов или числом) *субъекта*, *сущности*, *роли*, *учетной записи пользователя* и др., однозначно их *идентифицирующее*.

**Идентификация** — присвоение *субъектам*, *сущностям*, *роли* или *учетной записи пользователя идентификатора* и (или) сравнение предъявляемого идентификатора с перечнем присвоенных идентификаторов (на основе [1]).

**Иерархия ролей (административных ролей)** — отношение *частичного порядка*, заданное на множестве *ролей*, удовлетворяющее условию: если две роли подчинены одна другой в иерархии ролей и старшая роль содержится в множестве авторизованных ролей некоторого *пользователя* (*учетной записи пользователя*), то младшая роль также содержится в этом множестве.

**Иерархия сущностей** — отношение *частичного порядка* на множестве *сущностей*, как правило, удовлетворяющее условию: если некоторая сущность $x$ подчинена сущностям $y$ и $z$, то из сущностей $y$ и $z$ одна подчинена другой (в ОС семейства *Unix* допускается размещение сущностей-объектов одновременно в нескольких сущностях-каталогах).

**Изолированная программная среда (ИПС)** — *замкнутая программная среда*, все *субъекты* которой из порождаемого множества, *монитор безопасности субъектов* и *монитор безопасности обращений попарно абсолютно корректны в смысле изолированной программной среды* (на основе [7]).

**Информационный поток [от сущности-источника к сущности-приемнику]** — преобразование *данных* в *сущности*-приемнике, реализуемое *субъектами системы*, зависящее от данных, содержащихся в сущности-источнике.

**Информационный поток запрещенный** —

1) *информационный поток*, запрещенный априорно заданной *политикой безопасности управления доступом и информационными потоками*;

2) в рамках КС с *мандатным управлением доступом* или их моделей информационный поток от *сущностей* с большим *уровнем конфиденциальности* к сущностям с меньшим уровнем конфиденциальности;

3) в рамках КС с *мандатным контролем целостности* или их моделей *информационный поток по памяти* от *сущностей* с *меньшим уровнем целостности* к сущностям с большим уровнем целостности;

4) в рамках *ДП-моделей информационные потоки по памяти* от *недоверенных субъектов* к *сущностям*, *функционально ассоциированным* с *доверенными субъектами*, или к недоверенным субъектам от сущностей, *параметрически ассоциированных* с доверенными субъектами.

**Информационный поток по памяти** — *информационный поток*, при реализации которого фактор времени является несущественным, и используются *сущности* (память), в которые передающий *субъект* (субъекты) записывает *данные*, а принимающий субъект (субъекты) считывает их (на основе [4]).

**Информационный поток по времени** — *информационный поток*, при реализации которого фактор времени является существенным, и передающий *субъект* (субъекты) модулирует передаваемые *данные* на некоторый изменяющийся во времени несущий процесс (последовательность событий) в КС, а принимающий субъект (субъекты) демодулирует передаваемые данные, наблюдая несущий процесс во времени (на основе [4]).

**Информационный поток разрешенный** — *информационный поток*, разрешенный априорно заданной *политикой безопасности управления доступом* и *информационными потоками*.

**Команда в моделях ХРУ или ТМД** — в рамках *моделей ХРУ* или *ТМД правило преобразования состояний* (задаваемых *матрицей доступов*), определяемое па-

раметрами (типизированными в модели ТМД), являющимися *субъектами* или *объектами*, условиями проверки у субъектов *прав доступа* к объектам и конечной последовательностью *примитивных операторов*.

**Команда преобразования состояний** — см. *правило преобразования состояний*.

**Контейнер** — *сущность*, которая содержит или получает информацию, над которой *субъекты* выполняют операции и могут получать к ней *доступ* или *права доступа* либо целиком, либо отдельно к входящим в ее состав сущностям (*объектам* или другим контейнерам).

**Конфиденциальность информации** — субъективно определяемая характеристика информации, указывающая на необходимость введения ограничений на множество *субъектов*, имеющих *права доступа* к данной информации.

**Кооперация субъектов** — взаимодействие *субъектов* КС при передаче *прав доступа* или реализации *информационных потоков*.

**Мандатный контроль целостности (управление доступом с мандатным контролем целостности)** — *управление доступом*, соответствующее следующим требованиям: все *сущности* должны быть *идентифицированы*; задана *решетка уровней целостности информации*; каждой сущности присвоен *уровень целостности*, задающий установленные ограничения на доступ к данной сущности; каждому *субъекту* системы присвоен *уровень целостности*, задающий уровень полномочий данного субъекта; субъект может получить *доступ* к сущности только в случае, когда уровень целостности субъекта позволяет предоставить ему данный доступ к сущности с заданным уровнем целостности, и реализация доступа не приведет к возникновению *запрещенных информационных потоков* от сущностей с низким уровнем целостности к сущностям с высоким уровнем целостности.

**Матрица доступов** — матрица, используемая при реализации *дискреционного управления доступом*, каждая строка которой соответствует *субъекту* КС, столбец — *сущности* КС, ячейка содержит список *прав доступа* субъекта к сущности, представляющий собой подмножество множества видов прав доступа, реализованных в КС.

**Модели безопасности логического управления доступом и информационными потоками (ДП-модели)** — семейство автоматных моделей КС с *дискреционным*, *мандатным* или *ролевым управлением доступом*, ориентированных на анализ условий передачи *прав доступа* к *сущностям* КС и возникновения между ними *информационных потоков по памяти* или по *времени* с учетом следующих существенных особенностей функционирования КС: возможности кооперации части *субъектов* при передаче прав доступа и создании информационных потоков, возможности реализации в КС *доверенных* и *недоверенных субъектов* с различными условиями функционирования, возможности противодействия доверенными субъектами КС передаче прав доступа или созданию информационных потоков недоверенными субъектами, различия условий реализации в КС информационных потоков по памяти и по времени, наличия в КС *иерархии сущностей* и возможности ее использования при создании информационных потоков по времени, а также изменения функциональности субъекта при реализации информационного потока по памяти на *функционально ассоциированные* с ним *сущности* или от *параметрически ассоциированных* с ним *сущностей*, необходимости в ряде случаев определения различных правил управления доступом и информационными потоками для распределенных компонент КС. Основные элементы: *граф прав доступа*, *доступов и информационных потоков* (включая *функцию иерархии сущностей*), задающий состояние системы, и фиксированное множество *правил преобразования состояний*.

**Модели безопасности логического управления доступом и информационными потоками дискреционные (дискреционные ДП-модели)** — базовые модели всего семейства *ДП-моделей*, ориентированные на анализ в КС с *дискреционным управлением доступом и информационными потоками* условий передачи *прав доступа* к *сущностям* и возникновения между ними *информационных потоков по памяти* или *по времени*. Основные модели семейства: ДП-модель с функционально или параметрически ассоциированными с субъектами сущностями (ФПАС ДП-модель), ДП-модель файловых систем (ФС ДП-модель) и ДП-модель управления доступом и информационными потоками в защищенных операционных системах (ЗОС ДП-модель).

**Модели безопасности логического управления доступом и информационными потоками мандатные (мандатные ДП-модели)** — модели семейства *ДП-моделей*, ориентированные на анализ в КС с *мандатным управлением доступом и информационными потоками* условий возникновения *запрещенных информационных потоков по времени* от *сущностей* с большим *уровнем конфиденциальности* к сущностям с меньшим уровнем конфиденциальности и запрещенных *информационных потоков по памяти* от *субъектов* с низким *уровнем доступа* к субъектам с высоким уровнем доступа или к сущностям, *функционально ассоциированным* с субъектами с высоким уровнем доступа. Основная модель семейства — мандатная ДП-модель.

**Модели безопасности логического управления доступом и информационными потоками ролевые (ролевые ДП-модели)** — модели семейства *ДП-моделей*, ориентированные на анализ в КС с *ролевым управлением доступом и информационными потоками* условий передачи *прав доступа ролей* и возникновения *информационных потоков*. Основные модели семейства: базовая ролевая ДП-модель (БР ДП-модель) и базовая ролевая ДП-модель управления доступом и информационными потоками в операционных системах (БРОС ДП-модель). Помимо элементов *дискреционных ДП-моделей* и *базовой ролевой модели* ($RBAC$) дополнительно используются следующие основные элементы: *доступы владения субъект-сессий* к субъект-сессиям, *фактические роли*, *права доступа* и *возможные действия* субъект-сессий, кроме того, в БРОС ДП-модель включены *фактические доступы* и *функции уровней целостности пользователей*, *сущностей*, *ролей* и *текущих уровней целостности субъект-сессий*.

**Модель администрирования ролевого управления доступом ($ARBAC$)** — автоматная модель КС с *ролевым управлением доступом и информационными потоками*, построенная на основе *базовой ролевой модели* ($RBAC$) и предназначенная для задания правил администрирования *иерархии ролей*, множеств авторизованных *ролей* пользователей и *прав доступа* ролей. Кроме элементов модели $RBAC$, основными элементами являются: *административные роли*, административные *права доступа*, функции авторизованных административных ролей *пользователей*, административных прав доступа административных ролей и иерархия административных ролей.

**Модель Белла — ЛаПадулы классическая** — автоматная базовая модель КС с *мандатным управлением доступом и информационными потоками*. Включает описание требований безопасности (*$*$-свойства*, *ss-свойства* и *ds-свойства*), направленных на предотвращение возможности реализации *запрещенных информационных потоков по памяти*. Основные элементы: *субъекты*, *объекты*, множество текущих *доступов* субъектов к объектам, *решетка многоуровневой безопасности*, *функции уровней доступа субъектов*, *текущих уровней доступа субъектов* и *уровней конфиденциальности объектов*, множество действий *системы*.

**Модель Белла — ЛаПадулы в интерпретации «безопасность переходов»** — автоматная модель КС с *мандатным управлением доступом и информаци-*

*онными потоками*, являющаяся интерпретацией *классической модели Белла — ЛаПадулы*, ориентированная на анализ условий реализации *информационных потоков по памяти* от *объектов* (*сущностей*) с высоким *уровнем конфиденциальности* к объектам с низким уровнем конфиденциальности. Основные элементы: *субъекты*, объекты, множество текущих *доступов* субъектов к объектам, *решетка многоуровневой безопасности*, *функции уровней доступа субъектов* и *уровней конфиденциальности объектов*, *функция переходов системы*.

**Модель Биба** — автоматная модель КС с *мандатным контролем целостности*, основанная на реализации требований *политики low-watermark*, направленных на предотвращение возможности возникновения *запрещенных информационных потоков по памяти* от *объектов* (*сущностей*) с низким *уровнем целостности* к *объектам* с высоким уровнем целостности. Основные элементы: *субъекты*, объекты, множество текущих *доступов* субъектов к объектам, *решетка уровней целостности*, *функции уровней целостности субъектов*, *текущих уровней целостности субъектов* и *уровней целостности объектов*, *функция переходов системы*.

**Модель политики безопасности КС** — структурированное представление *политики безопасности*, которая должна быть осуществлена в КС (на основе [2]).

**Модель ролевая базовая (*RBAC*)** — автоматная базовая модель КС с *ролевым управлением доступом и информационными потоками*, предоставляющая шаблон описания таких систем. Основные элементы: *пользователи*, *сессии*, *роли*, *права доступа*, *функции авторизованных ролей пользователей*, *прав доступа ролей* и *текущих ролей сессий*, *иерархия ролей* и *ограничения*.

**Модель ролевого мандатного управления доступом (*MRBAC*)** — автоматная модель КС с *ролевым управлением доступом и информационными потоками*, построенная на основе *базовой ролевой модели* (*RBAC*) и предназначенная для задания *мандатного управления доступом*. Кроме элементов модели *RBAC*, основными элементами являются: *решетка многоуровневой безопасности*, функции *уровней доступа пользователей* и *уровней конфиденциальности объектов* (*сущностей*), ограничения на *функции авторизованных ролей пользователей*, *прав доступа ролей* и *текущих ролей сессий*.

**Модель систем военных сообщений (СВС)** — автоматная модель КС с *мандатным управлением доступом и информационными потоками*, основанная на *классической модели Белла — ЛаПадулы*, изначально ориентированная на реализацию в электронных почтовых системах. Основные элементы: *пользователи*, *сущности* (в том числе сущности-«устройства вывода»), *косвенные* и *непосредственные ссылки на сущности*, *решетка многоуровневой безопасности*, *роли*, *функции иерархии сущностей*, *уровней доступа пользователей*, *уровней конфиденциальности сущностей*, максимальных уровней конфиденциальности информации, выводимой на сущностях-«устройствах вывода», *функция переходов системы*.

**Модель субъектно-ориентированная изолированной программной среды** — автоматная модель, основанная на *дискреционном управлении доступом*, предназначенная для анализа условий реализации в КС *изолированной программной среды*. Основные элементы: *субъекты*, в том числе *монитор безопасности обращений* и *монитор безопасности субъектов*, *объекты* (*сущности*), в том числе *функционально ассоциированные* с субъектами, *информационные потоки по памяти*.

**Модель типизированной матрицы доступов (ТМД)** — автоматная модель КС с *дискреционным управлением доступом*, развивающая *модель ХРУ*. Основные элементы: *матрица доступов*, *команды*, состоящие из типизированных параметров,

условий проверки наличия *прав доступа* в *матрице доступов* и *примитивных операторов*, изменяющих содержимое матрицы доступов. В рамках модели обосновывается *алгоритмическая разрешимость* задачи проверки безопасности *ациклических монотонных систем ТМД*.

**Модель Харрисона — Руззо — Ульмана (ХРУ)** — автоматная модель КС с *дискреционным управлением доступом*, предоставляющая шаблон описания таких систем. Основные элементы: *матрица доступов* и *команды*, состоящие из параметров, условий проверки наличия *прав доступа* в матрице доступов и *примитивных операторов*, изменяющих содержимое матрицы доступов. В рамках модели доказывается *алгоритмическая неразрешимость* задачи проверки безопасности КС с дискреционным управлением доступом, а также существование алгоритма проверки безопасности *монооперационных систем ХРУ*.

**Модель угроз [безопасности информации]** — физическое, математическое, описательное представление свойств или характеристик *угроз безопасности информации* [3].

**Модель *Take-Grant* классическая** — автоматная модель КС с *дискреционным управлением доступом*, ориентированная на анализ путей передачи *прав доступа* между *объектами* системы. Основные элементы: *граф доступов*, фиксированный набор *де-юре правил преобразования графов доступов* (*take*, *grant*, *create*, *remove*).

**Модель *Take-Grant* расширенная** — автоматная модель КС с *дискреционным управлением доступом и информационными потоками*, ориентированная на анализ путей реализации *информационных потоков* между *объектами* системы. Основные элементы: *граф доступов и информационных потоков*, фиксированный набор *де-юре* (*take*, *grant*, *create*, *remove*) и *де-факто правил преобразования графов доступов* (*find*, *pass*, *post*, *spy*). В рамках модели описывается также алгоритм построения *замыкания графа доступов* и информационных потоков и обосновывается корректность этого алгоритма.

**Монитор безопасности обращений (МБО)** — *монитор обращений*, допускающий возникновение в *системе* только *разрешенных информационных потоков* (на основе [7]).

**Монитор безопасности субъектов (МБС)** — *монитор порождения субъектов*, разрешающий порождение субъектов только для фиксированного множества пар *активизирующих субъектов* и *сущностей-источников* (на основе [7]).

**Монитор обращений** —

1) концепция абстрактной машины, реализующей *политику управления доступом* в КС (на основе [2]);

2) *субъект* (множество субъектов), активизирующийся при возникновении любого *информационного потока* между *сущностями* КС (на основе [7]).

**Монитор порождения субъектов** — *субъект* (множество субъектов), активизирующийся при любом порождении субъектов (на основе [7]).

**Мост в модели *Take-Grant*** — проходящий через вершины-*объекты* путь в *графе доступов*; концами пути являются вершины-*субъекты*, а его словарная запись имеет один из следующих видов: $\overrightarrow{t^*}$, $\overleftarrow{t^*}$, $\overrightarrow{t^*}\,\overrightarrow{g}\,\overleftarrow{t^*}$, $\overrightarrow{t^*}\,\overleftarrow{g}\,\overleftarrow{t^*}$, где $t$ — *take*, $g$ — *grant* и символ «$*$» означает многократное (в том числе нулевое) повторение.

**Нарушение безопасности системы** — переход КС в *состояние*, в котором либо получен *запрещенный доступ субъекта* к *сущности*, либо произошла *утечка запрещенного права доступа* субъекта к сущности, либо реализован *запрещенный информационный поток* между сущностями КС.

**Нарушитель [правил доступа, разграничения доступа]** —

1) *субъект доступа*, осуществляющий *несанкционированный доступ к информации* [5];

2) *недоверенный субъект*.

**Неиерархическая категория конфиденциальности информации** — элемент *решетки многоуровневой безопасности*, характеризующий субъективно принадлежность информации некоторому множеству *пользователей* КС (например, информация для отдела № 1, информация для отдела № 2) или указывающий на ее принадлежность определенной области, категории (например, информация военная, политическая, экономическая).

**Несанкционированный доступ к информации (НСД)** — *доступ* к информации, нарушающий *правила разграничения доступа* с использованием штатных средств, предоставляемых средствами вычислительной техники или автоматизированными системами [5].

**Неформальный** — выраженный на естественном языке [2].

**Объект** — *сущность*, которая содержит или получает информацию, над которой *субъекты* выполняют операции (на основе [2]) и могут получать к ней *доступ* или *права доступа* только целиком.

**Ограничение** — в рамках *моделей* КС с *ролевым управлением доступом* требование, которому должны удовлетворять *функции авторизованных ролей пользователей*, *прав доступа ролей* или *текущих ролей сессий* (*субъект-сессий*).

**Ограничение динамическое** — в рамках *моделей* КС с *ролевым управлением доступом ограничение* на *функцию текущих ролей сессий* (*субъект-сессий*).

**Ограничение инвариантное относительно немонотонных правил преобразования состояний** — *ограничение* в рамках *ролевых ДП-моделей*, обладающее следующим свойством: если в *системе* задано только данное ограничение, то для любой *траектории системы* применение или неприменение на ней любого *немонотонного правила* не влияет на выполнение ограничений у последующих за ним правил преобразования состояний.

**Ограничение статическое** — в рамках *моделей* КС с *ролевым управлением доступом ограничение* на *функцию авторизованных ролей пользователей* или *функцию прав доступа ролей*.

**Остров в модели *Take-Grant*** — максимальный *tg-связный* подграф, состоящий только из вершин *субъектов*.

**Отношение порядка строгого** — бинарное отношение «$<$» на конечном множестве, обладающее свойствами антирефлексивности (не $a < a$), транзитивности ($(a < b \,\&\, b < c) \Rightarrow a < c$) и антисимметричности (не одновременно $a < b$ и $b < a$).

**Отношение порядка частичного** — бинарное отношение «$\leqslant$» на конечном множестве, обладающее свойствами рефлексивности ($a \leqslant a$), транзитивности ($(a \leqslant b \,\&\, \&\, b \leqslant c) \Rightarrow a \leqslant c$) и антисимметричности ($(a \leqslant b \,\&\, b \leqslant a) \Rightarrow a = b$).

**Параметр команды дочерний** — в рамках *модели ТМД* параметр *команды*, соответствующий создаваемому в результате ее выполнения *субъекту* или *объекту*.

**Параметр команды родительский** — в рамках *модели ТМД* параметр *команды*, соответствующий *субъекту* или *объекту*, который не создается в результате ее выполнения.

**Политика безопасности *low-watermark*** — *политика управления доступом и информационными потоками* в КС с *мандатным контролем целостности*, направленная на предотвращение возможности реализации *запрещенных информационных*

*потоков по памяти* от *сущностей* с низким *уровнем целостности* к *сущностям* с высоким уровнем целостности.

**Политика безопасности КС** — совокупность правил, регулирующих управление *активами*, их защиту и распределение в КС (на основе [2]).

**Политика безопасности [управления] информационных потоков** — *политика безопасности*, основанная на разделении всех возможных *информационных потоков* между *сущностями* КС на два непересекающихся множества: множество *разрешенных информационных потоков* и множество *запрещенных информационных потоков* — и заключающаяся в обеспечении невозможности возникновения в КС запрещенных информационных потоков.

**Политика безопасности управления доступом** — совокупность правил управления *доступом субъектов* к *сущностям* КС.

**Политика безопасности управления доступом дискреционная/мандатная/ролевая** — *политика безопасности*, соответствующая требованиям *дискреционного/мандатного/ролевого управления доступом.*

**Политика изолированной программной среды** — *политика безопасности*, заключающаяся в задании порядка безопасного взаимодействия *субъектов* КС, обеспечивающего невозможность воздействия ими на систему защиты КС и модификации ее параметров или конфигурации, результатом которого могло бы стать изменение априорно заданной в КС *политики безопасности управления доступом и информационными потоками.*

**Полуформальный** — выраженный на языке с ограниченным синтаксисом и определенной семантикой [2].

**Пользователь** — любая *сущность* (человек-пользователь или внешний объект информационной технологии) вне КС, которая взаимодействует с КС (на основе [2]).

**Пользователь доверенный/недоверенный** — *пользователь системы в рамках формальной модели*, от имени которого функционируют *доверенные/недоверенные субъекты.*

**Потенциальная модификация сущности с источником** — способ задания в рамках *модели СВС* условий *возникновения информационного потока.* При этом предполагается, что информационный поток от *сущности*-источника к сущности-приемнику возникает при переходе *системы* из *состояния* в состояние по запросу *пользователя*, приводящему к изменению данных в сущности-приемнике в зависимости от данных в *сущности*-источнике.

**Похищение права доступа** — получение *субъектом права доступа* к *сущности* без активного участия субъектов, которые изначально этим правом обладали.

**Правило преобразования графов доступов и информационных потоков де-факто в расширенной модели *Take-Grant*** — *правило преобразования графов доступов и информационных потоков* (*find*, *pass*, *post*, *spy*), в результате применения которого в графе создаются ребра («мнимые» ребра), соответствующие *информационным потокам.*

**Правило преобразования графов доступов [и информационных потоков] де-юре в классической или расширенной моделях *Take-Grant*** — *правило преобразования графов доступов и информационных потоков* (*take*, *grant*, *create*, *remove*), в результате применения которого в графе создаются ребра («реальные» ребра), соответствующие правам доступа *субъектов* или *объектов* друг к другу.

**Правило (команда) преобразования состояний (графов доступов, матриц доступов)** — правило, задающее порядок перехода автомата (*системы в рамках формальной модели*) из *состояния* в состояние.

**Правило преобразования состояний монотонное** — *правило преобразования состояний*, в результате применения которого из *состояния системы* не происходит удаления любых его элементов (*субъектов*, *сущностей*, *прав доступа*, *доступов*, *информационных потоков* и др.).

**Правило преобразования состояний немонотонное** — *правило преобразования состояний*, в результате применения которого из *состояния системы* происходит удаление хотя бы одного его элемента (*субъекта*, *сущности*, *права доступа*, *доступа*, *информационного потока* и др.).

**Право доступа** —

1) совокупность правил доступа к защищаемой информации, установленных правовыми документами или собственником, владельцем информации [3];

2) совокупность правил *доступа*, задающих порядок и условия доступа *субъектов* к *сущностям* КС.

**Право доступа запрещенное/разрешенное** — *право доступа субъекта* к *сущности*, запрещенное/разрешенное априорно заданной *политикой безопасности управления доступом* и *информационными потоками*.

**Предварительное условие** — в рамках *модели администрирования ролевого управления доступом* (*ARBAC*) условие, которому должна удовлетворять *роль* или *право доступа* перед тем, как соответственно роль будет включена в множества *авторизованных ролей* некоторого *пользователя* или право доступа будет включено в множество прав доступа некоторой роли.

**Примитивный оператор** — используемое для задания *команд* в рамках *моделей ХРУ* или *ТМД* преобразование *матрицы доступов*, заключающееся в добавлении или удалении одного *права доступа субъекта* к *объекту*, создании или удалении одного субъекта или объекта.

**Проблема массовая алгоритмически неразрешимая/разрешимая** — массовая проблема, для которой не существует/существует алгоритм, вычисляющий ее характеристическую функцию [9].

**Пролет моста конечный в модели *Take-Grant*** — проходящий через вершины-объекты путь в *графе доступов* в рамках *модели Take-Grant*, началом которого является вершина-*субъект*, концом — вершина-*объект*; словарная запись пути имеет вид $\overrightarrow{t^*}$, где $t$ — *take* и символ «$*$» означает многократное ненулевое повторение.

**Пролет моста начальный в модели *Take-Grant*** — проходящий через вершины-объекты путь в *графе доступов* в рамках *модели Take-Grant*, началом которого является вершина-*субъект*, концом — вершина-*объект*; словарная запись пути имеет вид $\overrightarrow{t^*}\,\overrightarrow{g}$, где $t$ — *take*, $g$ — *grant* и символ «$*$» означает многократное (в том числе нулевое) повторение.

**Путь [*tg*-путь] в модели *Take-Grant*** — проходящий через вершины-*субъекты* путь в *графе доступов* без учета направления ребер, каждое ребро которого помечено *правами доступа* $t$ (*take*) или $g$ (*grant*).

**Путь [*own*-путь] в базовой ДП-модели** — проходящий через вершины-*субъекты* путь в *графе прав доступа*, *доступов и информационных потоков* (описывающем *состояние системы* в рамках *базовой ДП-модели*) без учета направления ребер, каждое ребро которого помечено *правом доступа* владения ($own_r$).

**Решетка уровней конфиденциальности [линейная]** — линейно упорядоченное конечное множество *уровней конфиденциальности* (например, несекретно $<$ конфиденциально $<$ секретно).

**Решетка многоуровневой безопасности (*MLS*-решетка)** — решетка $(X \times L, \leqslant)$ с бинарным отношением частичного порядка «$\leqslant$», где $(L, \leqslant)$ — *линейная решетка уровней конфиденциальности*, $(X, \leqslant)$ — решетка подмножеств множества *неиерархических категорий конфиденциальности*, и отношение «$\leqslant$» на $X \times L$ удовлетворяет следующему условию: для $(a, \alpha), (b, \beta) \in X \times L$ справедливо $(a, \alpha) \leqslant (b, \beta)$ тогда и только тогда, когда $a \subseteq b$, $\alpha \leqslant \beta$.

**Решетка уровней целостности** — аналог *решетки многоуровневой безопасности*, в которой вместо *уровней конфиденциальности* и *неиерархических категорий конфиденциальности* используются соответственно *уровни целостности* и неиерархические категории целостности.

**Роль** —

1) заранее определенная совокупность правил, устанавливающих допустимое взаимодействие между *пользователем* и КС (на основе [2]);

2) совокупность *прав доступа* к *сущностям*.

**Роль авторизованная пользователя (учетной записи пользователя, субъекта)** — *роль*, которую потенциально могут получить функционирующие от имени *пользователя сессии* (*субъект-сессии*).

**Роль административная** — *роль*, обладающая *правами доступа*, позволяющими изменять параметры *ролевого управления доступом*, например множества *авторизованных ролей пользователей*, множества прав доступа ролей, *иерархию ролей*.

**Роль текущая** — роль, которой обладает *сессия* (*субъект*, *субъект-сессия*) в некотором *состоянии системы*.

**Свойство безопасности $*$-свойство (свойство «звезда»)** — основное свойство безопасности моделей *систем с мандатным управлением доступом*, построенных на основе *классической модели Белла — ЛаПадулы*, задающее условия предоставления доступов *субъектов* к *сущностям* (*объектам*), направленные на предотвращение возможности реализации *запрещенных информационных потоков по памяти* от сущностей с высоким *уровнем конфиденциальности* к сущностям с низким уровнем конфиденциальности.

**Свойство безопасности [*ds*-свойство] (свойство дискреционной безопасности)** — свойство безопасности, задающее *дискреционное управление доступом* в рамках моделей *систем с мандатным управлением доступом*, построенных на основе *классической модели Белла — ЛаПадулы*.

**Свойство безопасности [*ss*-свойство] («простое» свойство безопасности)** — основное свойство безопасности моделей *систем с мандатным управлением доступом*, построенных на основе *классической модели Белла — ЛаПадулы*, задающее условия предоставления доступов *субъектов* к *сущностям* (*объектам*), направленные на обеспечение возможности предоставления *доступа* на чтение субъекта к сущности только в случае, когда *уровень доступа субъекта* не ниже *уровня конфиденциальности сущности*.

**Связность [*tg*-связность] в модели *Take-Grant*** — две вершины-*субъекта* *tg*-связны в *графе доступов*, когда они соединены *tg-путем*.

**Связность [*own*-связность] в базовой ДП-модели** — две вершины-*субъекта* *own*-связны в *графе прав доступа*, *доступов и информационных потоков* (описыва-

ющем *состояние системы* в рамках *базовой ДП-модели*), когда они соединены *output-путем*.

**Сессия** — см. субъект.

**Система** — специфическое воплощение информационной технологии с конкретным назначением и условиями эксплуатации [2].

**Система в рамках ДП-моделей** — автомат, каждое состояние которого задается *графом прав доступа, доступов и информационных потоков* (включая *функцию иерархии сущностей*) и элементами, специфичными для конкретных *ДП-моделей*, а *функция переходов — правилами преобразования состояний*.

**Система в рамках модели Белла — ЛаПадулы** —

1) автомат, каждое *состояние* которого задается множеством текущих *доступов субъектов* к *объектам* (*сущностям*), *матрицей доступов* (только в *классической модели Белла — ЛаПадулы*), *функциями уровней доступа субъектов*, *текущих уровней доступа субъектов* (только в *классической модели Белла — ЛаПадулы*) и *уровней конфиденциальности объектов*; *функция переходов* задается либо множеством действий системы (в *классической модели Белла — ЛаПадулы*), либо явно;

2) в *классической модели Белла — ЛаПадулы* множество всех возможных *траекторий* с заданным начальным *состоянием*, на каждой из которых переход из состояния в состояние осуществляется в соответствии с заданным множеством действий системы.

**Система в рамках модели Биба** — автомат, каждое *состояние* которого задается множеством текущих *доступов субъектов* к *объектам* (*сущностям*), *функциями уровней целостности субъектов*, *текущих уровней целостности* и *уровней целостности объектов*; *функция переходов* задается явно.

**Система в рамках модели СВС** — автомат, для которого задано начальное *состояние* и *функция переходов*, сопоставляющая каждому текущему состоянию системы, *идентификатору пользователя* (инициатора запроса) и запросу к *системе* последующее состояние системы.

**Система в рамках модели ТМД** — автомат, каждое состояние которого задается *матрицей доступов* и функцией *типов объектов*, *функция переходов — командами*.

**Система в рамках модели ТМД монотонная (МТМД)** — *система ТМД*, в *командах* которой отсутствуют *немонотонные примитивные операторы* вида «удалить»... и «уничтожить»...

**Система в рамках модели ТМД монотонная в канонической форме** — *система МТМД*, в которой *команды*, содержащие *примитивные операторы* вида «создать»..., не содержат условий и примитивных операторов вида «внести»...

**Система в рамках модели ТМД монотонная ациклическая (АМТМД)** — *система МТМД*, *граф создания* которой не содержит циклов.

**Система в рамках модели ТМД монотонная циклическая** — *система МТМД*, *граф создания* которой содержит хотя бы один цикл.

**Система в рамках модели ТМД тернарная** — *система ТМД*, каждая *команда* которой имеет не более трех *параметров*.

**Система в рамках модели ХРУ** — автомат, каждое состояние которого задается *матрицей доступов*; *функция переходов — командами*.

**Система в рамках модели ХРУ монооперационная** — *система* в рамках *модели ХРУ*, в каждой *команде* которой содержится один *примитивный оператор*.

**Система в рамках формальной модели** — заданный в рамках *формальной модели управления доступом* и *информационными потоками* автомат.

**Система в рамках базовой ролевой модели (*RBAC*)** — заданный в рамках *базовой ролевой модели* (*RBAC*) автомат, каждое состояние которого описывается *иерархией ролей*, *функциями авторизованных ролей пользователей*, *прав доступа ролей* и *текущих ролей сессий*; *функция переходов* задается явно.

**Система в рамках модели *Take-Grant* классической** — автомат, каждое состояние которого задается *графом доступов*, *функция переходов — де-юре правилами преобразования графов доступов*.

**Система в рамках модели *Take-Grant* расширенной** — автомат, каждое состояние которого задается *графом доступов и информационных потоков*, *функция переходов — де-юре* и *де-факто правилами преобразования графов доступов*.

**Состояние системы** — состояние автомата, задающего *систему* в рамках *формальной модели управления доступом* и *информационными потоками*.

**Состояние системы безопасное** — в рамках моделей КС с *мандатным управлением доступом состояние*, в котором все *доступы субъектов* к *сущностям* обладают $*$*-свойством* и *ss-свойством* (или их аналогами), дополнительно в *классической модели Белла — ЛаПадулы* обладают *ds-свойством*.

**Состояние системы в рамках ДП-моделей** — *граф прав доступа, доступов и информационных потоков* (включая *функцию иерархии сущностей*), а также элементы, специфичные для конкретных *ДП-моделей*.

**Состояние системы в рамках классической модели *Take-Grant*** — *граф доступов*.

**Состояние системы в рамках модели Белла — ЛаПадулы** — множество текущих *доступов субъектов* к *объектам* (*сущностям*), *матрица доступов* (только в *классической модели Белла — ЛаПадулы*), *функции уровней доступа субъектов*, *текущих уровней доступа субъектов* (только в *классической модели Белла — ЛаПадулы*) и *уровней конфиденциальности объектов*.

**Состояние системы в рамках модели Биба** — множество текущих *доступов субъектов* к *объектам* (*сущностям*), *функции уровней целостности субъектов*, *текущих уровней целостности субъектов* и *уровней целостности объектов*.

**Состояние системы в рамках модели СВС** — задается функциями *идентификаторов пользователей* (сопоставляет каждому идентификатору соответствующего ему *пользователя*), *ссылок на сущности* (сопоставляет каждой ссылке *сущность*, на которую она указывает) и входов пользователей (ставит в соответствие каждому идентификатору пользователя сущность-«устройство вывода», с которой вход в систему осуществил пользователь с данным идентификатором). При этом в каждом состоянии считаются заданными все другие элементы *модели СВС* (функции и множества).

**Состояние системы в рамках модели ТМД** — *матрица доступов* и *функция типов объектов*.

**Состояние системы в рамках модели ХРУ** — *матрица доступов*.

**Состояние системы в рамках базовой ролевой модели (*RBAC*)** — задается *иерархией ролей* и *функциями авторизованных ролей пользователей*, *прав доступа ролей* и *текущих ролей сессий*.

**Состояние системы в рамках расширенной модели *Take-Grant*** — *граф доступов и информационных потоков*.

**Состояние системы в рамках модели ХРУ или ТМД, безопасное относительно права доступа *r*** —

1) *состояние системы* в рамках *модели ХРУ* или *ТМД*, из которого невозможен переход системы в такое состояние, в котором возможна *утечка права доступа r*;

2) состояние системы в рамках модели ХРУ или *ТМД*, из которого невозможен переход системы в состояние, в котором право доступа $r$ появляется в ячейке *матрицы доступов*, до этого $r$ не содержавшей.

**Ссылка на сущность косвенная** — ссылка на *сущность*, являющуюся частью *контейнера*, через последовательность двух и более ссылок на сущности, в которой только первая ссылка есть *непосредственная ссылка*.

**Ссылка на сущность непосредственная** — ссылка на *сущность*, совпадающая с *идентификатором сущности*.

**Субъект, активизирующий субъекта** — *субъект*, в результате воздействия которого на *сущность-источник* в *системе* создается (активизируется) заданный субъект.

**Субъект (субъект доступа, сессия, субъект-сессия)** — *сущность*, которая инициирует выполнение операций (на основе [2]) или *правил преобразования состояний системы в рамках формальной модели*; действия субъекта регламентируются *политикой безопасности управления доступом* и*информационными потоками*.

**Субъект доверенный** —

1) *субъект системы в рамках формальной модели*, инициирующий выполнение *правил преобразования состояний*, реализация которых не приводит к нарушению в *системе* заданной *политики безопасности управления доступом* и *информационными потоками*. При моделировании реальных КС доверенным субъектам, как правило, соответствуют субъекты, реализующие механизмы безопасности или функционирующие от имени *доверенных* (привилегированных) *пользователей*;

2) в рамках *классической модели Белла — ЛаПадулы* субъект, которому разрешено нарушать $*$-*свойство безопасности*.

**Субъект (субъект-сессия), доверенный и корректный в смысле целостности относительно сущности** — *доверенный субъект*, не реализующий *информационный поток по памяти* к данной *сущности* от любой сущности с *уровнем целостности* ниже, чем у данной сущности.

**Субъект (субъект-сессия), доверенный и корректный относительно информационных потоков по времени** — *доверенный субъект*, не участвующий в реализации *информационных потоков по времени*.

**Субъект, доверенный и корректный параметрически относительно сущности** — *доверенный субъект*, не реализующий *информационный поток по памяти* к данной *сущности* (не являющейся доверенным субъектом) от любой сущности, *параметрически ассоциированной* с любым доверенным субъектом. При этом всегда каждый доверенный субъект корректен относительно другого доверенного субъекта (как сущности).

**Субъект (субъект-сессия), доверенный и корректный параметрически относительно доверенного субъекта и сущности** — *доверенный субъект*, не реализующий *информационный поток по памяти* к данной *сущности* от любой сущности, *параметрически ассоциированной* со вторым доверенным субъектом.

**Субъект (субъект-сессия), доверенный и корректный функционально** — *доверенный субъект*, во множество *функционально ассоциированных сущностей* которого не входят *недоверенные субъекты*.

**Субъект, доверенный и корректный функционально относительно сущности** — *доверенный субъект*, не реализующий *информационный поток по памяти* от данной *сущности* (не являющейся доверенным субъектом) к любой сущности, *функционально ассоциированной* с любым доверенным субъектом. При этом всегда каждый

доверенный субъект корректен относительно другого доверенного субъекта (как сущности).

**Субъект (субъект-сессия), доверенный и корректный функционально относительно доверенного субъекта и сущности** — *доверенный субъект*, не реализующий *информационный поток по памяти* от данной *сущности* к любой сущности, *функционально ассоциированной* со вторым доверенным субъектом.

**Субъект недоверенный** — *субъект системы в рамках формальной модели*, инициирующий выполнение любых заданных в *системе правил преобразования состояний*. При моделировании реальных КС недоверенным субъектам, как правило, соответствуют субъекты, функционирующие от имени *нарушителя* или *недоверенных* (непривилегированных) *пользователей*.

**Субъекты, попарно корректные в смысле изолированной программной среды** — множество *субъектов*, для каждой пары $s$ и $s'$ которых выполняется условие: в любом *состоянии системы* отсутствует *информационный поток по памяти* между любыми *сущностями* $e$ и $e'$, *функционально ассоциированными* соответственно с субъектами $s$ и $s'$ (на основе [7]).

**Субъекты, попарно корректные абсолютно в смысле изолированной программной среды** — множество *субъектов попарно корректных в смысле изолированной программной среды*, для каждой пары которых множества *функционально ассоциированных* с ними *сущностей* не имеют пересечения (на основе [7]).

**Сущности, ассоциированные параметрически с субъектом** — множество *сущностей*, содержащих *данные*, позволяющие идентифицировать вид преобразования данных, реализуемого *субъектом*.

**Сущности, ассоциированные параметрически с учетной записью пользователя** — множество *сущностей*, реализация от каждой из которых *информационного потока по памяти* к некоторому *субъекту* (*субъект-сессии*) позволяет ему создать субъекта от имени *данной учетной записи пользователя*.

**Сущности, ассоциированные параметрически с ролью** — множество *сущностей*, реализация от каждой из которых *информационного потока по памяти* к некоторому *субъекту* (*субъект-сессии*) является необходимым для получения или удаления им данной *роли* из множества его *текущих ролей*.

**Сущность** — *объект* или *контейнер*.

**Сущность (объект), ассоциированная функционально с субъектом** — *сущность*, *данные* в которой влияют на вид преобразования данных, реализуемого *субъектом*. Всегда субъект как сущность функционально ассоциирован сам с собой.

**Сущность (объект)-источник при порождении субъекта** — *сущность*, в результате воздействия на которую некоторым *субъектом* (*активизирующим субъектом*) в *системе* создается новый субъект.

**Тип сущности** — атрибут *сущности*, используемый при реализации, как правило, *дискреционного управления доступом*, как аналог *права доступа* (применяется в *моделях ТМД*, *СВС* и др., а также в некоторых ОС, например, ОС семейства *Linux* с пакетом безопасности *SELinux*).

**Тип параметра команды дочерний/родительский** — в рамках *модели ТМД тип дочернего/родительского параметра команды*.

**Траектория (история) системы** — конечная последовательность *состояний системы*, при этом на ней все переходы из состояния в состояние осуществлены в соответствии с заданными в системе *правилами преобразования состояний*.

**Траектория системы без кооперации доверенных и недоверенных субъектов для передачи прав доступа** — *траектория системы* в рамках *дискреционных* и *мандатных ДП-моделей*, при реализации которой используются *монотонные правила преобразования состояний* и *доверенные субъекты* не дают *недоверенным субъектам права доступа* к *сущностям* и не берут права доступа к сущностям.

**Траектория системы без кооперации доверенных и недоверенных субъект-сессий для передачи прав доступа** — *траектория системы* в рамках *ролевых ДП-моделей*, при реализации которой используются только *монотонные правила преобразования состояний* и *доверенные субъект-сессии* не берут *роли* во множество *текущих ролей*, не дают другим ролям *права доступа* к *сущностям*, не получают *доступ владения* к субъект-сессиям.

**Траектория системы без кооперации доверенных и недоверенных субъектов для передачи прав доступа и реализации информационных потоков** — в рамках *дискреционных* и *мандатных ДП-моделей траектория системы без кооперации доверенных и недоверенных субъектов для передачи прав доступа*, при реализации которой в случае инициирования *доверенными субъектами* применения *правил преобразования состояний* не создаются *информационные потоки по времени*.

**Траектория системы блокирующих доступов доверенных субъектов** — в рамках *дискреционных* и *мандатных ДП-моделей траектория системы без кооперации доверенных и недоверенных субъектов для передачи прав доступа*, при реализации которой все *доступы доверенных субъектов* являются *блокирующими* (*информационные потоки по времени*).

**Траектория системы простая в рамках ролевых ДП-моделей** — *траектория без кооперации доверенных и недоверенных субъект-сессий для передачи прав доступа*, при реализации которой *субъект-сессии* не получают *доступа владения* к субъект-сессиям с использованием *информационных потоков по памяти* к *сущностям, функционально ассоциированным*, или от сущностей, *параметрически ассоциированных* с этими субъект-сессиями.

**Угроза безопасности информации** — совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации [3].

**Угроза доступности информации** — потенциально возможное несанкционированное блокирование *доступа* к информации.

**Угроза конфиденциальности информации** — потенциально возможное нарушении *субъектами* установленных *политикой безопасности управления доступом* ограничений на *доступ* к информации.

**Угроза раскрытия параметров КС** — потенциально возможное несанкционированное выявление параметров, функций и свойств системы защиты КС.

**Угроза целостности информации** — потенциально возможное несанкционированное изменение информации *субъектами*.

**Управление доступом** — реализация в КС *политики безопасности управления доступом*.

**Управление доступом дискреционное** — *управление доступом*, соответствующее следующим требованиям: все *сущности* (в том числе *субъекты*) должны быть *идентифицированы*, т. е. каждой сущности должен быть присвоен уникальный *идентификатор*, задана *матрица доступов*, при этом субъект обладает *правом доступа* к сущности в том и только в том случае, когда в ячейке матрицы доступов, соответствующей субъекту и сущности, содержится данное право доступа.

**Управление доступом мандатное** — *управление доступом*, соответствующее следующим требованиям: все *сущности* должны быть *идентифицированы*; задана *решетка уровней конфиденциальности*; каждой сущности присвоен *уровень конфиденциальности*, задающий установленные ограничения на доступ к данной сущности; каждому *субъекту* системы присвоен *уровень доступа*, задающий уровень полномочий данного субъекта; субъект может получить *доступ* к сущности только в случае, когда уровень доступа субъекта позволяет предоставить ему данный доступ к сущности с заданным уровнем конфиденциальности и реализация доступа не приведет к возникновению *запрещенных информационных потоков* от сущностей с высоким уровнем конфиденциальности к сущностям с низким уровнем конфиденциальности.

**Управление доступом ролевое** — *управление доступом*, соответствующее следующим требованиям: все *сущности* должны быть *идентифицированы*; задано множество *ролей*, каждой из которых ставится в соответствие некоторое множество *прав доступа* к сущностям; каждый *субъект* (*сессия*, *субъект-сессия*) обладает некоторым множеством разрешенных (*авторизованных*) для данного субъекта ролей; субъект обладает правом доступа к сущности в случае, когда он обладает *текущей ролью* (принадлежащей множеству авторизованных ролей субъекта), которой соответствует множество прав доступа, содержащее данное право доступа к данной сущности.

**Управление информационными потоками** — реализация в КС *политики безопасности информационных потоков*.

**Уровень доступа пользователя** — элемент *решетки многоуровневой безопасности*, задающий максимальный *уровень конфиденциальности сущностей*, к которым функционирующие от имени *пользователя субъекты* могут получать *доступ* (как правило, на чтение).

**Уровень доступа субъекта** — элемент *решетки многоуровневой безопасности*, задающий максимальный *уровень конфиденциальности сущностей*, к которым *субъект* может получать *доступ* (как правило, на чтение).

**Уровень доступа субъекта текущий** — элемент *решетки многоуровневой безопасности*, задаваемый с целью предотвращения *запрещенных информационных потоков по памяти* от *сущностей* с высоким *уровнем конфиденциальности* к сущностям с низким уровнем конфиденциальности в зависимости от текущих (в некотором *состоянии системы*) *доступов субъекта* к *сущностям*. Всегда не превосходит *уровня доступа субъекта*.

**Уровень конфиденциальности информации (данных)** — элемент *решетки многоуровневой безопасности*, характеризующий *конфиденциальность информации*.

**Уровень конфиденциальности сущности (объекта)** — элемент *решетки многоуровневой безопасности*, соответствующий *уровню конфиденциальности данных*, содержащихся в *сущности*.

**Уровень целостности информации** — элемент *решетки уровней целостности*, характеризующий *целостность информации*.

**Уровень целостности пользователя (учетной записи пользователя)** — элемент *решетки уровней целостности*, задающий максимальный *уровень целостности сущностей*, к которым функционирующие от имени *пользователя субъекты* могут получать *доступ* на модификацию (запись, удаление и др.).

**Уровень целостности роли** — элемент *решетки уровней целостности*, задающий максимальный *уровень целостности сущностей*, к которым данная *роль* может обладать *правами доступа* владения или на модификацию (запись, удаление и др.).

**Уровень целостности субъекта** — элемент *решетки уровней целостности*, задающий максимальный *уровень целостности сущностей*, к которым *субъект* может получать *доступ* на модификацию (запись, удаление и др.).

**Уровень целостности субъекта текущий** — элемент *решетки уровней целостности*, задаваемый с целью предотвращения *запрещенных информационных потоков по памяти* от *сущностей* с низким *уровнем целостности* к сущностям с высоким уровнем целостности в зависимости от текущих (в некотором *состоянии системы*) *доступов субъекта* к сущностям. Всегда не превосходит *уровня целостности субъекта*.

**Уровень целостности сущности (объекта)** — элемент *решетки уровней целостности*, соответствующий *уровню целостности данных*, содержащихся в *сущности*.

**Утечка права доступа (права доступа роли)** — переход КС в *состояние*, в котором *субъект* получает *право доступа* к *сущности*, запрещенное априорно заданной *политикой безопасности управления доступом* и *информационными потоками*.

**Утечка права доступа $r$ в рамках модели ХРУ** — переход *системы* в рамках *модели ХРУ* из некоторого *состояния* в последующее состояние в результате применения некоторой *команды*, содержащей *примитивный оператор*, вносящий *право доступа* $r$ в ячейку *матрицы доступов*, до этого $r$ не содержавшую.

**Учетная запись пользователя** — *идентификатор пользователя* и множество *параметрически ассоциированных* с ним *сущностей*.

**Фактическое возможное действие субъекта (субъект-сессии)** — действие (применение *правила преобразования состояния системы*), которое *субъект* может инициировать либо сам (при наличии у него соответствующих *текущих прав доступа*, *текущих ролей* и *административных ролей*), либо от имени некоторого другого субъекта, *доступом владения* к которому обладает первый субъект.

**Фактическая роль/административная роль/доступ/уровень целостности субъекта (субъект-сессии)** — *текущая роль/административная роль/доступ/уровень целостности* либо самого *субъекта*, либо некоторого другого субъекта, *доступом владения* к которому обладает первый субъект.

**Фактическое право доступа субъекта (субъект-сессии)** — *право доступа* либо самого *субъекта* (или его *текущей роли*), либо некоторого другого субъекта (или его *текущей роли*), *доступом владения* к которому обладает первый субъект.

**Формальная модель** — модель, заданная на математическом или ином другом *формализованном* языке.

**Формальный** — выраженный на языке с ограниченным синтаксисом и определенной семантикой, основанной на установившихся математических понятиях [2].

**Функция авторизованных ролей пользователей** — функция, задающая для каждого *пользователя* множество *ролей*, которые потенциально могут получить функционирующие от его имени *сессии* (*субъект-сессии*).

**Функция администрирования иерархии ролей** — в рамках *модели администрирования ролевого управления доступом* ($ARBAC$) функция, задающая для каждой административной *роли* интервал ролей, в пределах которого она позволяет изменять *иерархию ролей*.

**Функция администрирования множеств авторизованных ролей пользователей** — в рамках *модели администрирования ролевого управления доступом* ($ARBAC$) функция, задающая для каждой административной *роли* множество ролей,

которые с ее использованием могут быть включены или удалены из множества *авторизованных ролей пользователя* при выполнении заданных *предварительных условий*.

**Функция администрирования прав доступа ролей** — в рамках моделей КС с *ролевым управлением доступом* функция, задающая для каждой административной *роли* множество ролей, для которых с ее использованием разрешено включать или удалять *права доступа* из множеств прав доступа этих ролей (дополнительно в рамках *модели администрирования ролевого управления доступом ARBAC* требуется выполнение предварительных условий).

**Функция иерархии ролей (административных ролей)** — функция, задающая для каждой *роли* (*административной роли*) роли, которые непосредственно подчинены ей в *иерархии ролей*.

**Функция иерархии сущностей** — функция, задающая для каждой *сущности* сущности, которые непосредственно подчинены ей в *иерархии сущностей*.

**Функция переходов системы** — функция, задающая в рамках модели *системы* правила ее перехода из *состояния* в состояние.

**Функция переходов системы безопасная** —

1) в рамках *интерпретации «безопасность переходов» модели Белла — ЛаПадулы функция переходов системы*, обладающая $*$*-свойством* и *ss-свойством* для *функции переходов*;

2) в рамках *модели СВС функция переходов системы*, безопасная во всех смыслах, заданных в данной модели.

**Функция переходов системы, безопасная в смысле администрирования** —

1) в рамках *интерпретации «безопасность переходов» модели Белла — ЛаПадулы функция переходов системы*, разрешающая инициировать переход *системы* из *состояния* в состояние, в результате которого происходит изменение для некоторого *субъекта* его *уровня доступа* или для некоторого *объекта* (*сущности*) его *уровня конфиденциальности* только специально заданному для данных субъекта или объекта множеству субъектов;

2) заданная в рамках *модели СВС функция переходов системы*, запрещающая выполнение запросов *пользователей* к *системе*, которые не обладают *ролью*, разрешающей им изменять множества *авторизованных ролей пользователей*, *уровни доступа пользователей* или максимальные *уровни конфиденциальности информации*, выводимой на сущностях-«устройствах вывода».

**Функция переходов системы, безопасная в смысле базовой теоремы безопасности** — заданная в рамках *модели СВС функция переходов системы*, запрещающая выполнение запросов к *системе*, по которым она осуществляет переходы в небезопасные *состояния*.

**Функция переходов системы, безопасная в смысле доступов к контейнерам с атрибутом *CCR*** — заданная в рамках *модели СВС функция переходов системы*, запрещающая выполнение запросов *пользователей* к *системе*, которые приводят к *потенциальной модификации сущности с сущностью-источником*, находящейся в *контейнере* со значением `true` *атрибута способа доступа к содержимому контейнера* (*CCR*), обладающем *уровнем конфиденциальности* выше, чем *уровень доступа пользователя*.

**Функция переходов системы, безопасная в смысле доступов к сущностям** — заданная в рамках *модели СВС функция переходов системы*, запрещающая выполнение запросов к *системе*, по которым она осуществляет переходы в *состояния*, не удовлетворяющие требованиям *дискреционного управления доступом*.

**Функция переходов системы, безопасная в смысле доступов по косвенной ссылке** — заданная в рамках *модели CBC функция переходов системы*, запрещающая выполнение запросов *пользователей* к *системе*, которые приводят к получению *непосредственной ссылки на сущность*, находящуюся в *контейнере* со значением `true` *атрибута способа доступа к содержимому контейнера* (*CCR*), обладающем *уровнем конфиденциальности* выше, чем *уровень доступа пользователя*.

**Функция переходов системы, безопасная в смысле модификации сущностей** — заданная в рамках *модели CBC функция переходов системы*, запрещающая выполнение запросов к *системе*, которые приводят к *потенциальной модификации сущности*-приемника, обладающей *уровнем конфиденциальности* ниже, чем уровень конфиденциальности *сущности*-источника.

**Функция пользователей сессий (субъект-сессий)** — функция, задающая для каждой *сессии* (*субъект-сессии*) *пользователя*, от имени которого она функционирует.

**Функция прав доступа ролей** — функция, задающая для каждой *роли* множество *прав доступа* к *сущностям* (*объектам*), которыми она обладает.

**Функция типов объектов в модели ТМД** — функция, задающая в каждом состоянии *системы ТМД* для каждого *объекта* (или *субъекта*) его *тип*.

**Функция текущих уровней доступа субъектов (пользователей)** — функция, задающая в рамках *моделей систем* с *мандатным управлением доступом* для каждого *субъекта* его *текущий уровень доступа*.

**Функция текущих уровней целостности субъектов (субъект-сессий)** — функция, задающая в рамках *моделей систем* с *мандатным контролем целостности* для каждого *субъекта* его *текущий уровень целостности*.

**Функция текущих ролей сессий (субъект-сессий)** — функция, задающая для каждой *сессии* (*субъект-сессии*) в каждом *состоянии системы* множество *ролей*, которыми она обладает. Всегда для каждой сессии данное множество является подмножеством множества *авторизованных ролей пользователя*, от имени которого она функционирует.

**Функция уровней доступа субъектов (пользователей)** — функция, являющаяся одним из основных элементов *моделей систем* с *мандатным управлением доступом* и задающая для каждого *субъекта* его *уровень доступа*.

**Функция уровней конфиденциальности сущностей (объектов)** — функция, являющаяся одним из основных элементов *моделей систем* с *мандатным управлением доступом* и задающая для каждой *сущности* ее *уровень конфиденциальности*.

**Функция уровней целостности ролей** — функция, являющаяся элементом *ролевых ДП-моделей* и задающая для каждой *роли* ее *уровень целостности*.

**Функция уровней целостности субъектов (субъект-сессий, пользователей)** — функция, являющаяся одним из основных элементов *моделей систем* с *мандатным контролем целостности* и задающая для каждого *субъекта* его *уровень целостности*.

**Функция уровней целостности сущностей (объектов)** — функция, являющаяся одним из основных элементов *моделей систем* с *мандатным контролем целостности* и задающая для каждой *сущности* ее *уровень целостности*.

**Целостность информации** — состояние информации, при котором отсутствует любое ее изменение, либо изменение осуществляется только преднамеренно *субъектами*, имеющими на него право [3].

## ЛИТЕРАТУРА

1. Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных // Методические материалы ФСТЭК. 2008.
2. Безопасность информационных технологий. Критерии оценки безопасности информационных технологий // Руководящий документ. ГОСТ Р ИСО/МЭК 15408-2002.
3. Защита информации. Основные термины и определения // Национальный стандарт Российской Федерации. ГОСТ Р 50922-2006.
4. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов // Национальный стандарт Российской Федерации. ГОСТ Р 53113.1-2008, ГОСТ Р 53113.2-2009.
5. Концепция защиты средств вычислительной техники и автоматизированных систем от несанкционированного доступа к информации // Гостехкомиссия России. Руководящий документ. 1992.
6. Техническая защита информации. Основные термины и определения // Рекомендации по стандартизации. Р 50.1.056-2005.
7. *Щербаков А. Ю.* Современная компьютерная безопасность. Теоретические основы. Практические аспекты. Учеб. пособие. М.: Книжный мир, 2009. 352 с.
8. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками. Учеб. пособие для вузов. М.: Горячая линия — Телеком, 2011. 320 с.
9. *Носов В. А.* Основы теории алгоритмов и анализа их сложности. Курс лекций. М.: МГУ, 1992. 140 с.

# МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

DOI 10.17223/20710410/12/3
УДК 510.52+004.051

## FPT-АЛГОРИТМЫ И ИХ КЛАССИФИКАЦИЯ НА ОСНОВЕ ЭЛАСТИЧНОСТИ

В. В. Быкова

*Институт математики Сибирского федерального университета, г. Красноярск, Россия*

**E-mail:** bykvalen@mail.ru

Приведены основные положения и проблемы параметризированной алгоритмики — нового направления теории сложности вычислений. Предложен новый показатель вычислительной сложности параметризированного алгоритма, с помощью которого можно измерять темп роста функции сложности многих переменных. Этим показателем является частная эластичность функции сложности. Предложена двумерная классификация параметризированных алгоритмов для мультипликативной формы представления функций сложности.

**Ключевые слова:** *сложность вычислений, параметризированные алгоритмы, анализ алгоритмов, эластичность алгоритмов.*

С позиции классической теории сложности большинство задач, имеющих важное практическое значение, NP-трудные [1]. Классическая теория сложности анализирует и классифицирует задачи и алгоритмы по объему ресурса (преимущественно времени), необходимого для достижения требуемого результата, и потому оперирует функциями вычислительной сложности $t(n)$, зависящими от одной переменной $n$ — длины входа алгоритма. Такая одномерность и ориентация в анализе алгоритмов на худший случай зачастую делает задачи сложнее, чем они есть на самом деле. Для практического решения NP-трудных задач были предложены многие подходы, среди которых приближенные и параметризированные алгоритмы. Цель приближенных алгоритмов — найти за полиномиальное время решение, близкое к оптимальному. Заметный класс приближенных алгоритмов составляют полностью полиномиальные схемы приближений [2]. Параметризированные алгоритмы направлены на поиск точных решений NP-трудных задач, когда параметр $k$ решаемой задачи мал по сравнению с длиной входа алгоритма. Роль этого параметра — учесть информацию о структуре входных данных алгоритма и выделить основной источник неполиномиальной сложности NP-трудной задачи. В параметризированной теории используются двумерные функции сложности алгоритмов (функции, зависящие от $n$ и $k$), поэтому ее называют двумерной теорией сложности вычислений [3]. Параметризированная теория сложности создает основу для анализа и детальной классификации задач, которые труднорешаемые в классическом понимании, а также для развития новых методов анализа и разработки эффективных алгоритмов решения NP-трудных задач. Идея параметризированного подхода к оценке сложности алгоритмов и задач была предложена в 90-х годах прошлого столетия [4]. За последние два десятилетия эта идея нашла применение в различных областях компьютерных наук, таких, как криптография [5], искусственный интеллект [6], вычислительная биология [7], теория баз данных [8].

## 1. Параметризированные задачи и алгоритмы

Приведем основные положения теории параметризированной сложности, используя обозначения и понятия, введенные в работах [3, 4]. Параметризированная задача $\Pi$ состоит в том, что для заданных языка $L(\Pi) \subseteq \Sigma^* \times \mathbb{N}$, где $\Sigma$ — некоторый конечный алфавит и $\mathbb{N}$ — множество всех неотрицательных целых чисел, и пары $(I, k) \in \Sigma^* \times \mathbb{N}$ требуется определить, является ли $(I, k)$ элементом $L(\Pi)$. Для алгоритма $\alpha$, решающего данную задачу, $(I, k)$ называют входом, $I$ — его основной частью, $n = |I|$ — длиной входа и число $k$ — параметром задачи. Алгоритм $\alpha$ называют параметризированным алгоритмом, если его вычислительная сложность, определяемая количеством времени исполнения, оценивается с точки зрения длины входа $|I| = n$ и значения параметра $k$. Таким образом, функция вычислительной сложности параметризированного алгоритма, определяющая время его работы, — функция двух переменных $t(n, k)$.

Параметризированная задача $\Pi$ считается разрешимой с фиксированным параметром, или FPT-разрешимой (*Fixed-Parameter Tractable*), если она может быть решена некоторым параметризированным алгоритмом за время

$$t(n, k) = \mathrm{O}(n^{\mathrm{O}(1)} \cdot f(k)) \tag{1}$$

для функции $f$, зависящей только от параметра $k$. Порядок роста функции $f(k)$ не ограничивается. Так, возможно $f(k) = 2^{\mathrm{o}(k)}$ или $f(k) = 2^{\mathrm{O}(k)}$. Важно, что исключаются функции вида $f(n, k)$, например $f(n, k) = n^k$. Класс всех разрешимых с фиксированным параметром задач обозначается FPT. Соответствующие параметризированные алгоритмы, решающие такие задачи, называются FPT-алгоритмами.

Примером FPT-разрешимой задачи может служить задача о вершинном покрытии графа, когда параметром выступает размер покрытия. В самом деле, вершинное покрытие размера $k$ в графе $G$ с $n$ вершинами можно определить за время $\mathrm{O}(n \cdot 2^k)$. Если в данной задаче в качестве параметра взять древовидную ширину $\mathrm{tw}(G)$ графа $G$, то метод динамического программирования способен отыскать наибольшее вершинное покрытие за время $\mathrm{O}(n \cdot 2^{\mathrm{tw}(G)})$ [9]. Следовательно, параметризация данной задачи относительно $\mathrm{tw}(G)$ вновь приводит к FPT-разрешимости. Примером задачи, которая не принадлежит FPT, является задача о $k$-раскраске $n$-вершинного графа с параметром $k$.

Очевидно, что в FPT лежат все полиномиально разрешимые задачи. Однако наибольший интерес с точки зрения теории и практики представляют FPT-разрешимые задачи, являющиеся NP-трудными. С теоретической точки зрения все эти задачи могут быть решены за полиномиальное время при каждом фиксированном значении параметра. Реально это удается осуществить в большинстве случаев лишь при малых значениях параметра (все зависит от вида функции $f(k)$). Уже имеются сборники параметризированных задач, включая FPT-разрешимые задачи. Наиболее конкретным на данный момент времени является сборник М. Чезати [10]. Теория параметризированной сложности развивается по нескольким направлениям: определение иерархии классов сложности параметризированных задач, установление условий FPT-разрешимости, выявление взаимосвязи между параметризированной сложностью и классами приближенных алгоритмов (в частности, полностью полиномиальными схемами приближений), развитие параметризированной алгоритмики (методов анализа и разработки параметризированных алгоритмов) [3]. Последнее направление — это преимущественно область прикладных исследований. Здесь имеется ряд открытых вопросов.

## 2. Некоторые вопросы параметризированной алгоритмики

Анализ сборника параметризированных задач [10] показывает, что одна и та же задача может иметь различные параметризации (различные параметризированные формулировки) относительно разнообразных параметров. Так, в задаче о вершинном покрытии параметрами могут быть размер покрытия, число вершин, число ребер, древовидная ширина, наибольшая степень вершин графа и др. Закономерны вопросы:

1) Что может служить в качестве параметра для NP-трудной задачи?
2) Как следует выбирать параметр, чтобы задача стала FPT-разрешимой?
3) Существуют ли специальные методы разработки FPT-алгоритмов?
4) Как сравнивать между собой различные FPT-алгоритмы, предназначенные для решения одной и той же NP-трудной задачи?

На некоторые из этих вопросов пока нет исчерпывающих ответов. Например, на первые два вопроса параметризированная теория сложности отвечает следующим образом.

Бывают параметры внутренние и внешние. Параметр считается явным внутренним, если он непосредственно появляется в основной части экземпляра $(I, k)$ параметризированной задачи П и его значение ограничено полиномом от $|I| = n$. Для задачи о вершинном покрытии подобными параметрами являются размер покрытия, число вершин или число ребер графа. Имеются стандартные параметры. Они наиболее популярны. Так, в задачах минимизации в роли стандартного параметра всегда выступает величина, которая минимизируется. Для задачи о вершинном покрытии — это размер покрытия. Если $k = n$, то $k$ становится тривиальным параметром, и параметризированный алгоритм вырождается в обычный алгоритм. Параметр является неявным внутренним, если его значение ограничено полиномом от $|I| = n$ и информация о нем неявно скрыта в $I$. Типичным примером неявного внутреннего параметра является древовидная ширина графа (значение этой величины никогда не превосходит числа вершин графа). В целом, все внутренние параметры характеризуют некоторым образом структуру входных данных алгоритма. Наконец, существуют внешние параметры, которые никак не связаны с $|I|$ или связаны с $|I|$ неполиномиальным образом. Информация о таких параметрах задается дополнительно к $I$, так как считается, что ее нет в $I$.

С теоретической точки зрения в качестве параметра может выступать любая часть входа алгоритма. На практике для получения хорошей параметризации (например, относящейся к FPT) рекомендуется выбирать ту часть входа, которая обычно принимает малые значения по сравнению с $|I| = n$. В конечном счете, все зависит от приложения. Для различных приложений могут быть приемлемы различные параметризации одной и той же NP-трудной задачи. Относительно ряда NP-трудных задач доказано, что некоторые их параметризации не могут быть реализованы FPT-алгоритмами. В частности, такой задачей является задача о $k$-раскраске $n$-вершинного графа с параметром $k$. Для некоторых NP-трудных задач найдены FPT-алгоритмы лишь применительно к отдельным параметрам. Например, для многих оптимизационных задач на графах древовидная ширина оказалась таким параметром, который, как правило, приводит к FPT-алгоритмам [9 – 11].

Что касается третьего из перечисленных выше вопросов, то в настоящее время уже предложены и апробированы несколько специальных методов конструирования FPT-алгоритмов. Прежде всего, это метод динамического программирования на основе дерева декомпозиции для графов с ограниченной древовидной шириной [11]. Другим

известным методом построения FPT-алгоритмов является параметрическая редукция (*kernelization*) [4]. Параметрическая редукция — это полиномиальное по времени преобразование каждого экземпляра $(I, k)$ параметризированной задачи $\Pi$ в экземпляр $(I', k')$ параметризированной задачи $\Pi'$ так, что $k' < k$. После выполнения редукции к экземпляру $(I', k')$, называемому ядром, применяется полный перебор. Например, для задачи о вершинном покрытии со стандартным параметром $k$ (размером покрытия) редукция сводится к многократному применению двух правил: удаление изолированной вершины без изменения значения параметра $k$; удаление вершины, степень которой больше $k$, и уменьшение параметра $k$ на единицу. Следует отметить, что метод параметрической редукции в сочетании с полным перебором не всегда приводит к FPT-алгоритму. Между тем доказано, что если параметризированная задача принадлежит к FPT, то данный метод неизменно дает FPT-алгоритм [4].

Четвертый вопрос теоретически открыт: в многочисленных работах по параметризированной теории сложности для анализа и сравнения параметризированных алгоритмов применяются методы классической (одномерной) теории сложности [12 – 14]. Между тем классическая одномерность ограничивает глубину анализа параметризированных алгоритмов, для которых вычислительная сложность описывается функцией от двух переменных $t(n, k)$. В настоящей работе предлагается мера вычислительной сложности алгоритма, с помощью которой можно исследовать темп роста функций многих переменных, анализировать степень влияния структуры входных данных параметризированного алгоритма на его вычислительную сложность, сравнивать между собой FPT-алгоритмы решения NP-трудных задач. Этой мерой является частная эластичность функции вычислительной сложности (далее просто функции сложности) алгоритма. Представленные в статье результаты являются обобщением и расширением результатов автора, опубликованных в [15 – 17] применительно к одномерной теории сложности вычислений.

## 3. Соглашения относительно функций сложности

Формальный подход к анализу параметризированных алгоритмов требует уточнения свойств их функций сложности $t(n, k)$. Относительно функций $t(n, k)$ мы сделаем несколько естественных допущений, которые выполнимы для большинства реальных алгоритмов:

- полагаем, что $t(n, k)$ — монотонно неубывающие функции по обоим аргументам, областью значений функций выступает множество неотрицательных действительных чисел, а областью определения — множество $\mathbb{N} \times \mathbb{N}$;
- допускаем возможность отступления от дискретности изменения $n$ и $k$ (с формальной заменой $n$ на $x$ и $k$ на $y$), т. е. считаем, что аргументы функции $t(n, k)$ непрерывны, а необходимые значения вычисляются в целочисленных точках $x = n$ и $y = k$. Руководствуясь данным допущением, ниже вместо $t(n, k)$ будем писать $z(x, y)$;
- предполагаем, что рассматриваемое множество функций ограничивается семейством $\mathfrak{L}$ «по-существу положительных», логарифмически-экспоненциальных функций. Функция $z(x, y)$ считается «по-существу положительной», если существуют такие значения $x_0$, $y_0$, что $z(x, y) > 0$ для всех $x > x_0$, $y > y_0$. Семейство $\mathfrak{L}$ определено рекурсивно, и всякая функция $z(x, y) \in \mathfrak{L}$ представима в виде $z(x, y) = e^{w(x,y)}$, где $w(x, y) \in \mathfrak{L}$ [18]. Известно, что каждая такая функция непрерывна и дифференцируема в той области, где она определена.

Данные предположения являются гарантией существования эластичности для функций сложности параметризированных алгоритмов.

## 4. Эластичность функции как мера вычислительной сложности алгоритма

Под эластичностью $E_x(z)$ функции $z = z(x)$ принято понимать предел отношения относительного приращения этой функции к относительному приращению аргумента:

$$E_x(z) = \lim_{\Delta x \to 0} \left( \frac{\Delta z}{z} : \frac{\Delta x}{x} \right) = \frac{dz}{z} : \frac{dx}{x} = \frac{dz}{dx} \cdot \frac{x}{z} = z' \frac{x}{z} = x(\ln z)'. \tag{2}$$

Отметим несколько важных особенностей эластичности $\mathfrak{L}$-функции. Для всякой $\mathfrak{L}$-функции $z = z(x)$ всегда существует эластичность и $E_x(z) \geqslant 0$. В [15] показано, что при $x \to \infty$ эластичность $\mathfrak{L}$-функции — тождественно постоянная, или бесконечно малая, или бесконечно большая логарифмически-экспоненциальная функция, для которой

$$E_x(z) = \mathrm{o}\left(\frac{z}{\ln x}\right), \quad E_x(cz^m) = \mathrm{O}(E_x(z)), \quad c > 0, m > 0.$$

Последняя оценка означает, что при $x \to \infty$ поведение эластичности инвариантно относительно полиномиального преобразования, заключающегося в умножении этой функции на константу $c > 0$ и возведении в положительную и отделенную от нуля степень $m > 0$. Хорошо известно [18], что две произвольные $\mathfrak{L}$-функции $z_1(x)$, $z_2(x)$ всегда сопоставимы между собой по порядку роста: если $z_1(x), z_2(x) \in \mathfrak{L}$, то при $x \to \infty$ верно одно из трех отношений:

$$z_1(x) \prec z_2(x), \quad z_2(x) \prec z_1(x), \quad z_1(x) = \mathrm{O}(z_2(x)).$$

Кроме того, доказано, что иерархия эластичностей порождает идентичную иерархию $\mathfrak{L}$-функций [17], т. е. если $E_x(z_1) \prec E_x(z_2)$, то $z_1(x) \prec z_2(x)$. Заметим, что здесь отношение $\prec$ интерпретируется так: $z_1(x) \prec z_2(x)$ тогда и только тогда, когда $z_1(x) = \mathrm{o}(z_2(x))$. О-большое обозначает асимптотическую пропорциональность функций: $z_1(x) = \mathrm{O}(z_2(x))$ тогда и только тогда, когда $z_1(x) \sim cz_2(x)$, $c > 0$. Согласно (2), при достаточно малых $\Delta x$ справедливо приближение

$$\frac{\Delta z}{z} \approx E_x(z) \frac{\Delta x}{x},$$

которое указывает, что $E_x(z)$ — коэффициент пропорциональности между темпами роста переменных $z$ и $x$. Поскольку $E_x(z) = E_{ax}(bz)$ для любых констант $a > 0$, $b > 0$, то эластичность — безразмерная величина.

Все перечисленные выше особенности эластичности позволяют использовать ее в качестве меры измерения вычислительной сложности алгоритма, поскольку она удовлетворяет ряду необходимых требований, таких, как сопоставимость, интерпретируемость и вычислимость. Сопоставимость обеспечивают безразмерность и инвариантность относительно модели вычислений: ведь переход от одной модели к другой меняет вычислительную сложность алгоритма лишь полиномиальным образом. Эластичность имеет отчетливую интерпретацию: если $z = z(x)$ — функция сложности алгоритма, то это означает, что при повышении значения $x$ (длины входа алгоритма) на один процент значение $z$ (время выполнения алгоритма) увеличивается приблизительно на $E_x(z)$ процентов. Вычислимость также имеет место: эластичность существует для всякой $\mathfrak{L}$-функции и всегда может быть определена непосредственно из (2) и свойств эластичности. Эластичности присущи свойства, сходные со свойствами операций логарифмирования и дифференцирования [16], поэтому она легко определяется для логарифмически-экспоненциальных функций. Простота вычисления — это основное достоинство эластичности по сравнению с другими мерами сложности вычислений.

Важно отметить, что эластичность — локальная характеристика функции: ее значения в общем случае изменяются при переходе от одного значения аргумента к другому. Между тем при больших значениях аргумента в поведении эластичности $\mathfrak{L}$-функций наблюдаются определенные закономерности: разным (по скорости роста) классам $\mathfrak{L}$-функций свойственно принципиально различное поведение эластичности, и наоборот, по асимптотике поведения эластичности можно однозначным образом определить класс, к которому относится $\mathfrak{L}$-функция. Это утверждает следующая теорема.

**Теорема 1** (о классификации $\mathfrak{L}$-функций) [15]. Разбиение семейства монотонно неубывающих «по-существу положительных» $\mathfrak{L}$-функций на классы SUBPOLY, POLY, SUBEXP, EXP, HYPEREXP в соответствии с порядком их роста эквивалентно надлежащему разбиению по асимптотике эластичности этих функций на бесконечности:

$$\text{SUBPOLY} = \{z(x) : z(x) \prec e^{\mathrm{O}(\ln x)}\} \equiv \{z(x) : E_x(z) = \mathrm{o}(1)\}; \tag{3}$$

$$\text{POLY} = \{z(x) : z(x) = \mathrm{O}\left(e^{\mathrm{O}(\ln x)}\right)\} \equiv \{z(x) : E_x(z) = \mathrm{O}(1)\}; \tag{4}$$

$$\text{SUBEXP} = \{z(x) : e^{\mathrm{O}(\ln x)} \prec z(x) \prec e^{\mathrm{O}(x)}\} \equiv \{z(x) : 1 \prec E_x(z) \prec x\}; \tag{5}$$

$$\text{EXP} = \{z(x) : z(x) = \mathrm{O}\left(e^{\mathrm{O}(x)}\right)\} \equiv \{z(x) : E_x(z) = \mathrm{O}(x)\}; \tag{6}$$

$$\text{HYPEREXP} = \{z(x) : e^{\mathrm{O}(x)} \prec z(x)\} \equiv \{z(x) : x \prec E_x(z)\}. \tag{7}$$

Данная теорема порождает пять сложностных классов алгоритмов (субполиномиальных, полиномиальных, субэкспоненциальных, экспоненциальных и гиперэкспоненциальных соответственно), которые полностью отвечают современной классификации алгоритмов. Класс быстрых алгоритмов составляют алгоритмы с функциями сложности $z(x) \in \text{SUBPOLY}$. Таким алгоритмам присуща тождественно нулевая или бесконечно малая эластичность. Класс полиномиальных алгоритмов — множество алгоритмов с $z(x) \in \text{POLY}$ и асимптотически постоянной эластичностью $E_x(z)$. Класс субэкспоненциальных алгоритмов — алгоритмы, для которых $z(x) \in \text{SUBEXP}$. Эластичность $E_x(z)$ субэкспоненциального алгоритма — бесконечно большая величина, такая, что $1 \prec E_x(z) \prec x$. Класс экспоненциальных алгоритмов — это алгоритмы, для которых $z(x) \in \text{EXP}$. Для них эластичность $E_x(z) = O(x)$ — бесконечно большая величина, асимптотически пропорциональная линейной функции. Класс гиперэкспоненциальных алгоритмов — это алгоритмы, для которых $z(x) \in \text{HYPEREXP}$ и $x \prec E_x(z)$. Эквивалентности (3) – (7) обеспечивают «прозрачность» сложностных классов алгоритмов, ведь эластичности согласно (3) – (7) представляются более простыми по виду функциями, чем соответствующие им $\mathfrak{L}$-функции. Понятие эластичности легко распространяется на функции многих переменных. Это дает возможность использовать данную дифференциальную характеристику функции в анализе параметризированных алгоритмов.

Под частной эластичностью $E_x(z)$ функции $z = z(x, y)$ по аргументу $x$ понимается эластичность переменной $z$, которая рассматривается как функция только от $x$ и при постоянных значениях $y$. Частная эластичность $E_x(z)$ связана с частной производной функции $z = z(x, y)$ по $x$ соотношением

$$E_x(z) = z'_x \frac{x}{z} = x(\ln z)'_x. \tag{8}$$

Аналогично определяется частная эластичность $E_y(z)$ функции $z = z(x, y)$ по аргументу $y$:

$$E_y(z) = z'_y \frac{y}{z} = y(\ln z)'_y. \tag{9}$$

Поскольку всякая $\mathfrak{L}$-функция $z = z(x, y)$ непрерывна и дифференцируема в той области, где она определена, то для нее в этой области всегда существуют частные эластичности. Выражения (8), (9) аналогичны формуле (2). Поэтому частные эластичности $E_x(z), E_y(z)$ обладают всеми основными свойствами эластичности функции одной переменной [16]. Укажем наиболее важные из этих свойств и запишем их применительно к $E_x(z)$:

1) Если $z = z(x, y)$ не зависит от $x$, то $E_x(z) = 0$.
2) Безразмерность: $E_x(z) = E_{ax}(bz)$ для любых констант $a > 0, b > 0$.
3) Частная эластичность произведения (отношения) функций $z_1 = z_1(x, y)$ и $z_2 = z_2(x, y)$ равна сумме (разности) их частных эластичностей:

$$E_x(z_1 \cdot z_2) = E_x(z_1) + E_x(z_2), \quad E_x(z_1/z_2) = E_x(z_1) - E_x(z_2).$$

Частным эластичностям свойственна отчетливая интерпретация. Пусть $z = z(n, k)$ является функцией сложности параметризированного алгоритма, где $n$ — длина входа, а $k$ — параметр этого алгоритма. После формальной замены $n$ на $x$ и $k$ на $y$ имеем $z = z(x, y) \in \mathfrak{L}$. Тогда $E_x(z)$ — коэффициент пропорциональности между темпом роста времени работы алгоритма и темпом роста его длины входа $x$. Аналогично $E_y(z)$ — коэффициент пропорциональности между темпом роста времени выполнения алгоритма и темпом роста параметра $y$.

## 5. Двумерная классификация FPT-алгоритмов по сложности

В общем случае частные эластичности $E_x(z), E_y(z)$ являются функциями, зависящими от двух аргументов $x$ и $y$. Однако ситуация значительно упрощается, если учесть тот факт, что для большинства параметризированных алгоритмов время работы алгоритма описывается $\mathfrak{L}$-функцией вида

$$z(x, y) = q(x) \cdot f(y), \tag{10}$$

где $q(x) \in \mathfrak{L}$ — количественная компонента, а $f(y) \in \mathfrak{L}$ — параметрическая компонента функции $z(x, y)$. Мультипликативная форма представления (10) характерна для алгоритмов с параметрически зависимым циклом, выполняющим полный перебор всех возможных вариантов при различных значениях параметра, тогда как в теле этого цикла выполняется проверка текущего варианта и время проверки зависит только от длины входа алгоритма. Именно такие алгоритмы порождает метод параметрической редукции. Заметим, что формула (1), определяющая время работы FPT-алгоритма, также имеет мультипликативную форму записи:

$$z(x, y) = \mathrm{O}(x^{\mathrm{O}(1)} \cdot f(y)). \tag{11}$$

Пусть $z(x, y) = q(x) \cdot f(y) \in \mathfrak{L}$. Тогда, исходя из свойств эластичности 1 и 3, частные эластичности $E_x(z), E_y(z)$ вырождаются в обычные эластичности функции одного аргумента:

$$E_x(z) = E_x(q(x) \cdot f(y)) = E_x(q(x)) + E_x(f(y)) = E_x(q(x)); \tag{12}$$
$$E_y(z) = E_y(q(x) \cdot f(y)) = E_y(q(x)) + E_y(f(y)) = E_y(f(y)). \tag{13}$$

Теперь $E_x(z)$ зависит лишь от $x$, а $E_y(z)$ — только от $y$. Поскольку $q(x), f(y) \in \mathfrak{L}$, то каждая из этих функций принадлежит только одному сложностному классу из $\mathfrak{K} = \{\text{SUBPOLY, POLY, SUBEXP, EXP, HYPEREXP}\}$. Обозначим через $\mathfrak{F}_x$ класс

сложности функции $z(x,y)$ по аргументу $x$, а через $\mathfrak{F}_y$ — класс сложности по аргументу $y$. Тогда всякому параметризированному алгоритму с функцией сложности $z(x,y) = q(x)\cdot f(y) \in \mathfrak{L}$ соответствует пара

$$(\mathfrak{F}_x, \mathfrak{F}_y) \in \mathfrak{K} \times \mathfrak{K},$$

характеризующая сложность данного алгоритма как по длине входа $x$, так и по значению параметра $y$. Таким образом, мы приходим к двумерной классификации параметризированных алгоритмов по сложности. При двумерном подходе более отчетливую и общую формулировку получает определение FPT-алгоритма и условие (11): параметризированный алгоритм называется FPT-алгоритмом, если время его исполнения задается функцией $z(x,y) = q(x)\cdot f(y) \in \mathfrak{L}$, для которой

$$(\mathfrak{F}_x, \mathfrak{F}_y) \in \{\text{SUBPOLY, POLY}\} \times \mathfrak{K}.$$

Когда параметризированная задача не только FPT-разрешима, но и полиномиально разрешима, то для нее существуют FPT-алгоритмы, сложность которых соответствует парам

$$(\mathfrak{F}_x, \mathfrak{F}_y) \in \{\text{SUBPOLY, POLY}\} \times \{\text{SUBPOLY, POLY}\}. \tag{14}$$

Подобные параметризированные алгоритмы естественно назвать полиномиальными FPT-алгоритмами. Алгоритмы, сложность которых отвечает парам

$$(\mathfrak{F}_x, \mathfrak{F}_y) \in \{\text{SUBPOLY, POLY}\} \times \{\text{SUBEXP}\}, \tag{15}$$

$$(\mathfrak{F}_x, \mathfrak{F}_y) \in \{\text{SUBPOLY, POLY}\} \times \{\text{EXP}\}, \tag{16}$$

$$(\mathfrak{F}_x, \mathfrak{F}_y) \in \{\text{SUBPOLY, POLY}\} \times \{\text{HYPEREXP}\}, \tag{17}$$

целесообразно определить как субэкспоненциальные, экспоненциальные и гиперэкспоненциальные FPT-алгоритмы соответственно. Такие FPT-алгоритмы присущи NP-трудным задачам.

Введенная двумерная классификация FPT-алгоритмов не противоречит понятиям, используемым в настоящее время в параметризированной теории сложности применительно к FPT-алгоритмам, а лишь формально их уточняет и расширяет. При желании в декартовом произведении $\mathfrak{K}\times\mathfrak{K}$ всегда можно выделить более мелкие подмножества пар $(\mathfrak{F}_x, \mathfrak{F}_y)$ и, как следствие, определить более детальную классификацию параметризированных алгоритмов, чем (14) – (17). При тривиальной параметризации $y = x$ соотношения (12), (13) принимают единый вид

$$E_x(z) = E_x(q(x)\cdot f(x)) = E_x(q) + E_x(f).$$

В этом частном случае приходим к одномерной классификации алгоритмов. Аналогичный случай возникает, когда $f(y)$ — тождественная константа.

## Заключение

Параметризированная теория сложности дала толчок к разработке новых подходов конструирования алгоритмов (алгоритмов решения задач, разрешимых с фиксированным параметром). Одновременно с этим, данная теория породила ряд проблем. Одна из них — отсутствие простых и удобных инструментов анализа параметризированных алгоритмов, когда их функции сложности зависят от многих переменных. В настоящей работе предложен показатель, с помощью которого можно измерять темп роста

функций многих переменных, сравнивать между собой FPT-алгоритмы, анализировать степень влияния параметра на вычислительную сложность параметризированного алгоритма. Этим показателем является частная эластичность. Для мультипликативной формы представления функций сложности в работе предложена двумерная классификация параметризированных алгоритмов, точно определен класс FPT-алгоритмов, выделены подклассы полиномиальных, субэкспоненциальных, экспоненциальных и гиперэкспоненциальных FPT-алгоритмов. Допустимо дальнейшее развитие предложенного подхода в части увеличения числа параметров и анализа других форм представления функций сложности.

## ЛИТЕРАТУРА


1. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
2. *Ausiello G., Crescenzi P., Gambosi G., et al.* Complexity and approximation, combinatorial optimization problems and their approximability properties. New York: Springer Verlag, 1999.
3. *Flum J. and Grohe M.* Parameterized complexity theory. Berlin; Heidelberg: Springer Verlag, 2006.
4. *Downey R. and Fellows M.* Parameterized complexity. New York: Springer Verlag, 1999.
5. *Fellows M. and Koblitz N.* Fixed-parameter complexity and cryptography // Technical Report DCS-207-IR, Department of Comput. Science, University of Victoria, 1992.
6. *Gottlob G., Scarcello F., and Sideri M.* Fixed-parameter complexity in AI and Nonmonotonic reasoning // Artific. Intellig. 2002. V. 138. P. 55–86.
7. *Bodlaender H., Downey R., Fellows M., et al.* Parameterized complexity analysis in computational biology // Comput. Appl. Bioscien. 1995. V. 11. P. 49–57.
8. *Papadimitriou C. and Yannakakis M.* On the complexity of database queries // J. Comput. System Scien. 1999. V. 58. P. 407–427.
9. *Niedermeier R.* Invitation to fixed-parameter algorithms: Oxford Lecture series in mathematics and its applications. Oxford: University Press, 2006.
10. *Cesati M.* Compendium of parameterized problems. `http://bravo.ce.uniroma2.it/home/cesati/research/compendium` — 2006.
11. *Gottlob G., Pichler R., and Wei F.* Abduction with bounded treewidth: From theoretical tractability to practically efficient computation // Proc. of the Twenty-Third AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI Press, 2008. P. 1541–1546.
12. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. М.: МЦНМО, 1999.
13. *Грин Д., Кнут Д.* Математические методы анализа алгоритмов. М.: Мир, 1987.
14. *Быкова В. В.* Математические методы анализа рекурсивных алгоритмов // Журн. СФУ. Математика и физика. 2008. № 1(3). С. 236–246.
15. *Быкова В. В.* Метод распознавания классов алгоритмов на основе асимптотики эластичности функций сложности // Журн. СФУ. Математика и физика. 2009. № 2(1). С. 46–61.
16. *Быкова В. В.* Эластичность алгоритмов // Прикладная дискретная математика. 2010. № 2(8). С. 87–95.
17. *Bykova V. V.* Complexity and elasticity of the computation // Proc. of the 3-rd IASTED International Multi-Conference on Automation, Control, and Information Technology (ACIT-CDA 2010). Anaheim-Calgary-Zurich: ACTA Press, 2010. P. 334–340.
18. *Грэхем Р., Кнут Д., Поташник О.* Конкретная математика. М.: Мир; Бином. Лаборатория знаний, 2006.

# ПРИКЛАДНАЯ ТЕОРИЯ АВТОМАТОВ

DOI 10.17223/20710410/12/4

УДК 519.713

## О ПОГРЕШНОСТИ ПОЛИНОМИАЛЬНОГО ВЫЧИСЛЕНИЯ ОПТИМАЛЬНОЙ РАСКРАСКИ ГРАФА В СИНХРОНИЗИРУЕМЫЙ АВТОМАТ[1]

М. В. Берлинков

*Уральский государственный университет, г. Екатеринбург, Россия*

**E-mail:** berlm@mail.ru

Сильносвязный орграф называется допустимым, если исходящие степени всех вершин в нём одинаковы и длины циклов взаимно просты в совокупности. Раскраской допустимого графа $G$ назовем произвольный автомат $A$, граф которого совпадает с $G$. Слово называется синхронизирующим автомат $A$, если оно переводит его в одно и то же состояние вне зависимости от исходного состояния автомата $A$. Оптимальная раскраска допустимого графа — это раскраска с кратчайшим синхронизирующим словом среди всех синхронизирующих раскрасок. Длину соответствующего синхронизирующего слова назовем значением оптимальной раскраски. Доказано, что любой приближенный полиномиальный алгоритм для вычисления оптимальной раскраски или ее значения имеет относительную погрешность не меньше 2 в случае трехбуквенного алфавита при предположении $\mathcal{P} \neq \mathcal{NP}$. Также показано, как результат можно перенести на случай двухбуквенного алфавита.

**Ключевые слова:** *проблема раскраски дорог, поиск оптимальной раскраски, кратчайшее синхронизирующее слово, синхронизируемые автоматы.*

### Введение и история задачи

Введём определения синхронизируемого автомата и оптимальной раскраски. Пусть $A = \langle Q, \Sigma, \delta\rangle$ — полный детерминированный конечный автомат, где $Q$ — множество состояний; $\Sigma$ — входной алфавит автомата; $\delta : Q \times \Sigma \to Q$ — функция переходов. Функция $\delta$ продолжается единственным образом на множество $Q \times \Sigma^*$, где $\Sigma^*$ обозначает свободный моноид над $\Sigma$. Таким образом, каждое слово из $\Sigma^*$ действует на множестве $Q$ в соответствии с функцией переходов $\delta$. Через $S.v$ будем обозначать образ подмножества $S \subseteq Q$ под действием слова $v$, т. е. $S.v = \{\delta(q, v) : q \in S\}$.

Автомат $A$ называется *синхронизируемым*, если существует слово $w \in \Sigma^*$, действие которого «перезагружает» автомат $A$, т. е. переводит автомат в некоторое состояние вне зависимости от исходного состояния автомата: $q.w = p.w$ для всех $q, p \in Q$, что эквивалентно равенству $|Q.w| = 1$. Произвольное слово $w$ с таким свойством называется *синхронизирующим* автомат $A$, а длина кратчайшего синхронизирующего слова для $A$ называется значением функцией Черни и обозначается через $\mathfrak{C}(A)$. На рис. 1 в центре нарисован автомат Черни $C_4$ с кратчайшим синхронизирующим словом $ba^3ba^3b$ длины 9 и, следовательно, $\mathfrak{C}(C_4) = 9$.

[1]Работа выполнена при поддержке Федерального агентства по образованию России (грант №2.1.1/13995) и РФФИ (грант №10-01-00524).

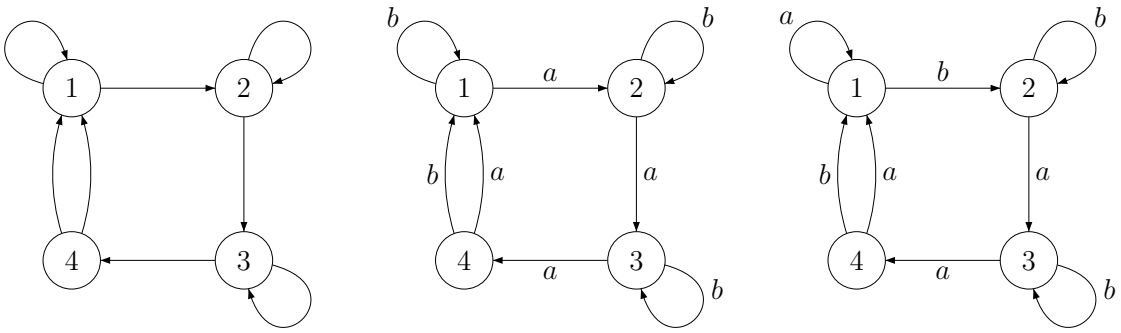


Рис. 1. Автомат Черни, его граф и оптимальная раскраска

Задачи, возникающие в теории синхронизируемых автоматов, являются не просто алгебраическими головоломками, но и имеют приложения в различных областях компьютерных наук, таких, как кодирование или построение вычислительных систем. Вычислительные системы зачастую могут быть смоделированы как конечные автоматы. Под действием *шума* или внутренних причин в системе могут происходить некорректные переходы. Системы, соответствующие синхронизируемым автоматам, более устойчивы к таким ошибкам, так как могут быть возвращены в корректное состояние применением перезагрузочной последовательности операций (синхронизирующего слова). Ясно, что важно не только наличие такой последовательности, но и то, насколько короткой ее можно выбрать. Таким образом, две ключевые задачи этой теории — это задача проверки заданного автомата на синхронизируемость и задача поиска кратчайшего синхронизирующего слова или его длины для заданного синхронизируемого автомата.

При моделировании систем конечными автоматами иногда встречаются такие, в которых определены возможные переходы между состояниями и набор допустимых операций, но не определено соответствие («раскраска») между операциями и переходами, либо соответствие задано, но его можно переопределить («перераскрасить»). При этом ключевые задачи остаются теми же, но ставятся с учетом возможности перераскраски: задача проверки возможности перераскраски в синхронизируемый автомат и задача поиска перераскраски с кратчайшим синхронизирующим словом среди всех синхронизируемых перераскрасок. Для перехода к этим задачам нам понадобится определение *графа автомата*.

Орграф $G = UG(A)$ называется *графом автомата* $A$, если его множество вершин совпадает с множеством состояний автомата $A$, а дуги соответствуют переходам автомата, т. е. из вершины $q$ есть стрелка в вершину $q'$ тогда и только тогда, когда $\delta(q, a) = q'$ для некоторой буквы $a \in \Sigma$. Тогда $G$ — орграф с одинаковыми степенями исхода вершин, и автомат $A$ называется *раскраской* графа $G$. Сильносвязный орграф $G$ называется *допустимым*, если исходящие степени всех вершин в нём одинаковы и длины циклов взаимно просты в совокупности. Условие одинаковых исходящих степеней вершин необходимо для возможности раскраски, а условия взаимной простоты длин циклов и сильной связности необходимы для разрешимости и неприводимости задачи нахождения синхронизирующей раскраски [1 – 3]. Очевидно, что любой допустимый орграф может быть получен как граф подходящего автомата и любая перераскраска автомата — это раскраска его графа.

Пусть $G$ — допустимый орграф. Раскраску $A$ назовем *оптимальной раскраской* $G$ и положим $\mathrm{OPT}(G) = \mathfrak{C}(A)$, если для любой другой раскраски $B$ графа $G$ выполняется $\mathfrak{C}(A) \leqslant \mathfrak{C}(B)$; в этом случае $\mathrm{OPT}(G)$ называется *значением оптимальной раскраски*. Другими словами, раскраска графа $G$ оптимальна, если она есть раскраска с кратчайшим синхронизирующим словом среди всех возможных синхронизируемых раскрасок графа $G$. На рис. 1 показаны один четырёхвершинный орграф и две его синхронизируемые раскраски. В центре нарисован автомат Черни с кратчайшим синхронизирующим словом $ba^3ba^3b$ длины 9, а справа — оптимальная раскраска заданного орграфа с синхронизирующим словом $a^3$ длины 3.

Вопрос проверки автомата на синхронизируемость полиномиально разрешим (см. для примера [4]). Гораздо более интересным является вопрос о том, можно ли перераскрасить любой автомат так, чтобы он стал синхронизируемым. Этот вопрос известен как *проблема раскраски дорог*. Задачу впервые поставили в 1970 г. Р. Л. Адлер и Б. Вэйс в работе [5], окончательно сформулировав её в следующем виде для допустимых графов в [1] (1977 г.).

**Гипотеза.** Любой допустимый граф имеет синхронизирующую раскраску.

Несмотря на многочисленные исследования, эта гипотеза оставалась открытой более 30 лет и была положительно решена А. Н. Трахтманом [2] только в 2008 г. Доказательство в [2] конструктивно и дает кубический (от числа состояний) алгоритм для нахождения синхронизирующей раскраски любого допустимого графа. В [6] было показано, как можно решать эту задачу квадратичным алгоритмом.

Рассмотрим теперь задачу минимизации длины синхронизирующего слова. Для автоматов ее можно поставить следующим образом.

**Задача** Min-Synch-Length. **Дан** синхронизируемый автомат $A$, **найти** $\mathfrak{C}(A)$.

Д. Эпштейн [7] показал, что эта задача NP- и co-NP-трудна, и, следовательно, даже недетерминированный полиномиальный алгоритм не может решать эту задачу в предположении $\mathcal{P} \neq \text{co-}\mathcal{NP}$. Из этого результата, однако, не следует, что не существует полиномиального алгоритма, находящего синхронизирующее слово на один символ длиннее кратчайшего. Тем не менее в [8] показано, что в предположении $\mathcal{P} \neq \mathcal{NP}$ не существует полиномиального алгоритма, решающего последнюю задачу даже для класса двухбуквенных автоматов.

Заметим, что для допустимых графов можно поставить две задачи, связанные с минимизацией длины синхронизирующего слова:

1) задачу вычисления значения оптимальной раскраски:

**Задача** OCV (Optimal Coloring Value). **Дан** допустимый граф $G$, **найти** $\mathrm{OPT}(G)$;

2) задачу построения оптимальной раскраски:

**Задача** OC (Optimal Coloring). **Дан** допустимый граф $G$, **найти** его оптимальную раскраску.

Заметим, что хотя задача Min-Synch-Length соответствует задаче OCV с зафиксированной раскраской, из её полиномиальной неаппроксимируемости не следует такой же результат для задачи вычисления значения оптимальной раскраски OCV, так как для нахождения $\mathrm{OPT}(G)$ не требуется искать значения $\mathfrak{C}(A(G))$ для всех раскрасок $A(G)$ графа $G$.

Далее докажем, что в предположении $\mathcal{P} \neq \mathcal{NP}$ любой приближенный полиномиальный алгоритм решения задачи OCV в трёхбуквенном алфавите имеет относительную погрешность не меньше 2, и покажем, как этот результат перенести на двухбуквенный случай и применить к задаче OC.

В п. 1 вводятся вспомогательные понятия и обозначения. В п. 2 доказано, что в предположении $\mathcal{P} \neq \mathcal{NP}$ любой полиномиальный алгоритм для вычисления значения оптимальной раскраски имеет относительную погрешность не меньше 2 в случае трехбуквенного алфавита, и показано, как перенести результат на задачу поиска оптимальной раскраски. В п. 3 показано, что при помощи некоторых изменений в структуре графа можно доказать такой же результат для раскраски в двухбуквенном алфавите. Из этого, в частности, следует, что задача поиска оптимальной раскраски NP-трудна даже в случае двухбуквенного алфавита.

## 1. Вспомогательные определения

Пусть $A$ — сильносвязный синхронизируемый автомат, $Q$ — множество его состояний. Назовем подмножество $S \subseteq Q$ *занятым* после применения некоторого слова $v$, если $S \subseteq Q.v$. Будем обозначать $v(i \dots j)$ подслово слова $v$ с $i$-го по $j$-й символ включительно; $v(i)$ — $i$-й символ слова.

Пусть $S, T$ — два непустых множества состояний автомата $A$. Через $LRadius_A(S, T)$ обозначим минимальное число $r$, такое, что найдется $t \in T$ и для каждого элемента $s$ из $S$ есть путь из $s$ в $t$ с длиной, в точности равной $r$. В роли множества $T$ обычно будет выступать $Q$, поэтому для краткости обозначим $LRadius_A(S, Q)$ через $LRadius_A(S)$. Например, на рис. 1 для автомата Черни $LRadius_{C_4}(\{1, 2\}, \{2\}) = 1$. Через $LSynch_A(S)$ для непустого множества состояний $S$ обозначим длину кратчайшего слова $v$, синхронизирующего множество $S$ в автомате $A$, т. е. такого $v$, что $|S.v| = 1$. Заметим, что на рис. 1 для автомата Черни $LSynch_{C_4}(\{1, 2\}) = 4$, и $LSynch_A(Q)$ совпадает с $\mathfrak{C}(A)$ для любого автомата $A$.

Пусть $r$ — натуральное число; тогда через $Image_A(S, r)$ (соответственно через $Preimage_A(S, r)$) обозначим множество состояний $q \in Q$, таких, что в $UG(A)$ есть путь длины не больше $r$ из множества $S$ в $q$ (соответственно в множество $S$ из $q$). Другими словами, $Image_A(S, r)$ (соответственно $Preimage_A(S, r)$) — это полный образ (соответственно прообраз) $S$ под действием слов длины не более $r$ в автомате $A$. Отметим несколько простых свойств введенных выше функций.

**Лемма 1.**

1) Функция $LRadius$ везде определена.
2) $LSynch_A(S) \geqslant LRadius_A(S)$.
3) Пусть $T$ — объединение двух непустых множеств $T_1$ и $T_2$; тогда
$$LRadius_A(S, T) = \min(LRadius_A(S, T_1), LRadius_A(S, T_2)).$$
4) Пусть $S_1$ — непустое подмножество $S$; тогда $LRadius_A(S_1, T) \leqslant LRadius_A(S, T)$.
5) Функции $LRadius$, $Image$ и $Preimage$ зависят только от графа автомата.
6) Пусть $S \nsubseteq Preimage(T, r)$ для некоторого $r > 0$; тогда $LRadius_A(S, T) > r$.

***Доказательство.*** Пусть $w$ есть кратчайшее слово, синхронизирующее $S$ в автомате $A$; тогда $S.w = q_0$ для некоторого $q_0 \in Q$. Так как автомат сильносвязный, то существует слово $v$, такое, что $q_0.v \in T$. Тогда $S.wv$ — это одноэлементное множество в $T$, поэтому $LRadius_A(S, T)$ определен и не превосходит $|wv|$. Таким образом, п. 1 леммы 1 доказан. Если в качестве $T$ выбрать множество всех состояний, то слово $v$ можно выбрать пустым. Следовательно, $LRadius_A(S) \leqslant |w| = LSynch_A(S)$, и п. 2 также доказан. Пункты 3–7 непосредственно следуют из определений. Действительно, проверим для примера п. 3. Заметим, что для функции $LRadius$ справедливо соотношение $LRadius_A(S, T) = \min\limits_{t \in T} LRadius_A(S, t)$. Следовательно, п. 3 следует из свойства минимума и условия $T = T_1 \cup T_2$. ■

Поскольку все автоматы, рассматриваемые далее в доказательстве теоремы, имеют один и тот же граф, то ввиду п. 5 леммы 1 можно не указывать автомат в записи функций $LRadius$, $Image$ и $Preimage$.

Используемые в работе определения из теории сложности вычислений можно найти в [9]. Приведем понятие аппроксимирующего алгоритма для нашей задачи. Пусть $\mathcal{K}$ — некоторый класс допустимых графов. Следуя определению погрешности алгоритма из [9], скажем, что алгоритм $Alg$ *приближенно находит* значение оптимальной раскраски в классе $\mathcal{K}$, если для любого допустимого графа $G \in \mathcal{K}$ алгоритм возвращает натуральное число $Alg(G) \geqslant \mathrm{OPT}(G)$. Тогда говорят, что алгоритм $Alg$ решает OCV с *относительной погрешностью* $k \in R$ для класса $\mathcal{K}$, если

$$\sup\left\{\frac{Alg(G)}{\mathrm{OPT}(G)} : G \in \mathcal{K}\right\} = k.$$

## 2. Случай трехбуквенного алфавита

**Теорема 1.** В предположении $\mathcal{P} \neq \mathcal{NP}$ любой полиномиальный алгоритм, решающий OCV для класса трехбуквенных автоматов, имеет относительную погрешность не меньше 2.

***Доказательство.*** План доказательства следующий. Покажем, что полиномиальный алгоритм $Alg$, решающий OCV с относительной погрешностью меньше 2, может быть использован для полиномиального алгоритма, решающего NP-полную задачу ВЫПОЛНИМОСТЬ (SAT). Таким образом, получим противоречие с предположением $\mathcal{P} \neq \mathcal{NP}$. Более точно, пусть полиномиальный алгоритм $Alg$ решает OCV с относительной погрешностью $2-\varepsilon$, где $0 < \varepsilon < 2$. Возьмем произвольный экземпляр задачи SAT — формулу $\psi$ с $n$ переменными и $m$ дизъюнктами (дизъюнкциями переменных и их отрицаний). Построим допустимый граф $G(\psi)$, имеющий полиномиальный размер от $m, n$, и полином $p(m, n)$, такие, что:

— если $\psi$ выполнима, то $\mathrm{OPT}(G(\psi)) \leqslant p(m, n)$ (этот случай назовем $GOODCASE$);
— если $\psi$ невыполнима, то $\mathrm{OPT}(G(\psi)) \geqslant (2 - 0{,}5\varepsilon)p(m, n)$ (случай $BADCASE$).

Из построения следует, что $\psi$ выполнима тогда и только тогда, когда $Alg$ возвращает значение, меньшее чем $(2 - 0{,}5\varepsilon)p(m, n)$. Следовательно, найдем решение NP-полной задачи SAT, используя полиномиальный алгоритм $Alg$.

Выполним теперь описанное сведение формально. Рассмотрим некоторый экземпляр $\psi$ задачи SAT с набором переменных $x_1, x_2, \ldots, x_n$ и набором дизъюнктов от них $c_1, c_2, \ldots, c_m$. Будем писать $x_i \in c_j$ или $\neg x_i \in c_j$, если дизъюнкт $c_j$ содержит в своей записи переменную $x_i$ или ее отрицание соответственно. Без ограничения общности можно предполагать, что $m > 3$, $n > 3$ и задача $\psi$ нормализована, т. е. выполняются следующие условия:

(C1) Для любой переменной $x_i$ есть дизъюнкт $c_j$, состоящий из переменной и ее отрицания, т. е. $c_j = (x_i \vee \neg x_i)$.
(C2) Переменная $x_n$ равна 0 в любом наборе, выполняющем $\psi$.
(C3) Для любой переменной $x_i$ и дизъюнкта $c_j$ если ($x_i \in c_j$ и $\neg x_i \in c_j$), то $c_j = x_i \vee \neg x_i$.

Для того чтобы выполнялись условия (C1), (C3), нужно добавить дизъюнкты вида $c_j = (x_i \vee \neg x_i)$, если их еще нет в формуле, и исключить все дизъюнкты, содержащие $x_i$, $\neg x_i$ и еще какие-то переменные. Если (C2) не выполняется, то можно добавить переменную $x_{n+1}$, дизъюнкт $c_{m+1} = (\neg x_{n+1})$, а также дизъюнкт $c_{m+2} = (x_{n+1} \vee \neg x_{n+1})$,

чтобы удовлетворять $(C1)$. Ясно, что такие преобразования формулы $\psi$ не влияют на ее выполнимость.

Вместо определения дуг графа $G(\psi)$ будем сразу определять переходы автомата $A$ этого графа так, что если $\psi$ выполнима, то $\mathfrak{C}(A) \leqslant p(m,n)$. Из этого следует корректность сведения для случая *GOODCASE*. Для случая *BADCASE*, т. е. когда $\psi$ невыполнима, сделаем предположение от противного о том, что есть некоторая раскраска $B$ графа $G(\psi)$, такая, что

$$\mathfrak{C}(B) < (2 - 0{,}5\varepsilon)p(m,n). \qquad (E_v)$$

Получив противоречие с этим предположением, докажем корректность сведения для случая *BADCASE*.

Таким образом, для случая *GOODCASE* достаточно показать, как можно за полиномиальное от $n, m$ время построить автомат $A$ графа $G(\psi)$, для которого есть синхронизирующее слово $u$ длины не больше $p(m,n)$. Для *BADCASE* нужно получить противоречие с тем, что найдется слово $v$ длины не больше $(2 - 0{,}5\varepsilon)p(m,n)$, синхронизирующее автомат $B$. Заметим также, что из доказательства будет видно, что граф $G(\psi)$ строится за полиномиальное от размера $\psi$ время.

Схема автомата $A$ показана на рис. 2. Множество состояний $Q$ автомата $A$ является объединением шести *компонент* (подмножеств состояний автомата, играющих определенную роль в доказательстве): $R_{init}$, $R_{sat}$, $T_{sat}$, $R_{fix}$, $R_{acc}$, $Q_{spec}$. Алфавит $\Sigma$ автомата $A$ состоит из трех букв $a, b, c$, а функция переходов будет определена по ходу доказательства. Заметим, что компонента $Q_{spec}$ состоит из подмножеств $C_0$, $C_1$, $C_2$, $R_0$ и состояния $z$ и не окружена рамкой. Компонента $R_{acc}$ состоит из набора подмножеств $D_1, D_2, \ldots, D_{n-1}$ и подкомпоненты $Row_{acc}$, также явно на рисунке не обозначенной.

Компоненты $R_{init}$ и $T_{sat}$ являются трехмерными, а компоненты $R_{sat}$ и $Row_{acc}$ — двумерными массивами состояний, т. е. каждое состояние из этих компонент может быть записано как $K(i_1, i_2, i_3)$, где $K$ — одна из компонент $R_{init}$, $R_{sat}$, $T_{sat}$, $Row_{acc}$ (индекс $i_3$ имеет смысл только для $R_{init}$ и $T_{sat}$). *Строкой* и *столбцом* этих компонент автомата будем называть подмножество состояний массива компоненты с фиксированным первым и вторым индексом соответственно. Например, на рис. 2 отмечены 1-я и 2-я строки $R_{init}$ ($R_{init}(1,*,*)$ и $R_{init}(2,*,*)$); 1-я и $(n+1)$-я строки $R_{sat}$ ($R_{sat}(1,*)$ и $R_{sat}(n+1,*)$); 4-й, 6-й и $(4n+2)$-й столбцы $Row_{acc}$ ($Row_{acc}(*,4)$, $Row_{acc}(*,6)$, $Row_{acc}(*,4n+2)$) и др. Одномерные массивы состояний $C_0$, $C_1$, $C_2$, $R_{fix}$ назовем столбцами, а $R_0$ — строкой. Заметим, что строкам и столбцам компонент на рис. 2 соответствуют горизонтальные и вертикальные наборы состояний соответственно.

Введем теперь полиномы, требующиеся для определения компонент:

$p_1 = p_1(m,n) = m + 3$ — число столбцов в $R_{init}$ и $R_{sat}$;

$p_2 = p_2(m,n) = 15n^2 + m$ — вспомогательный полином;

$p_3 = p_3(m,n) = 4(p_1 + p_2)$ — количество значений, которые может принимать третий индекс в $R_{init}$;

$p = p(m,n) = ((kn-1)p_3+2)+(n+1)+(18+(4n+13)(n-2))$ — полином, участвующий в оценках на $\mathrm{OPT}(G)$;

$p_{add} = p_{add}(m,n) = p_3(kn-1) = 4(15n^2+2m+2)(kn-1)$ — высота столбцов $C_0, C_1, C_2$ компоненты $Q_{spec}$.

Константа $k$ выбрана так, чтобы выполнялось неравенство $2p_{add} \geqslant (2 - 0{,}5\varepsilon)p$ для всех $m > 3$, $n > 3$. Заметим, что $k$ не зависит от задачи, т. е. от $\psi$, $m$ или $n$.

Перед тем как переходить к формальному определению компонент автомата и рассмотрению их свойств, объясним, для чего необходима каждая из компонент на соответствующем шаге доказательства.

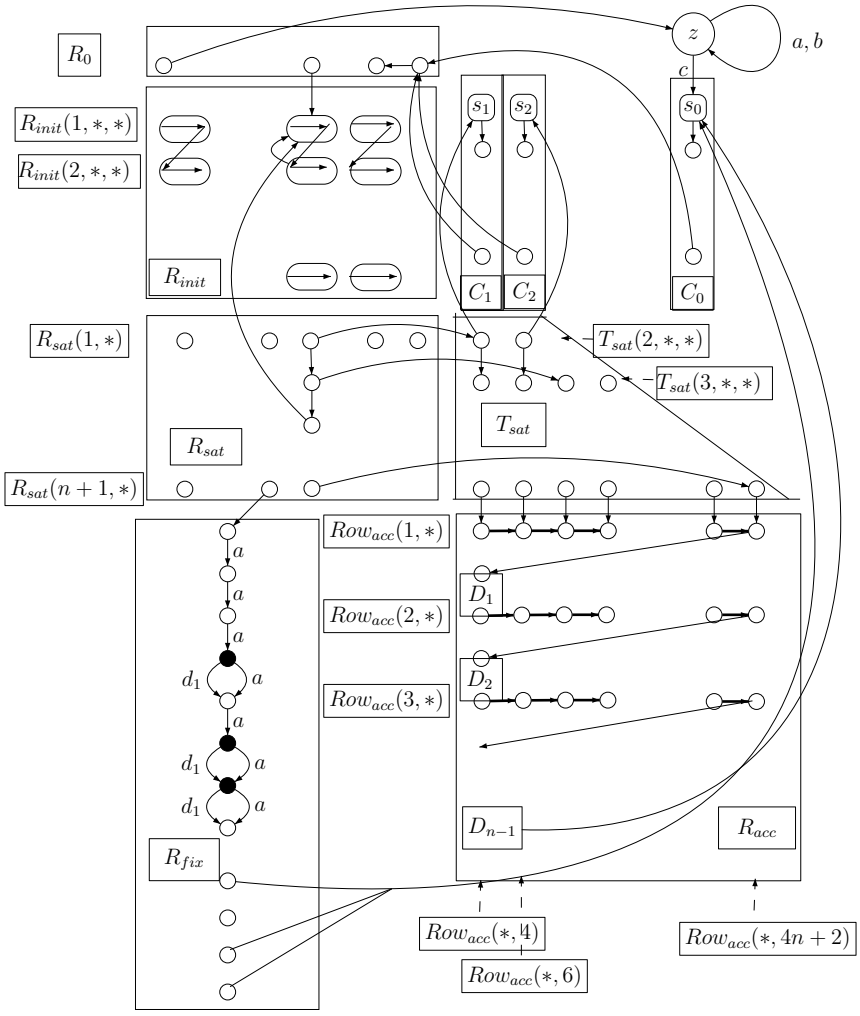


Рис. 2. Схема автомата $A$

1) При помощи свойств компоненты $R_{init}$ построим слово $u_{init}$ длины $(kn-1)p_3+$ $+2$ для автомата $A$, отображающее $Q \setminus Q_{spec}$ в первую строку $R_{sat}$ $(R_{sat}(1,*))$. Таким образом, фактически после этого шага потребуется отслеживать образ только одной строки. Для случая *BADCASE* докажем, что можно найти префикс $v_{init}$ слова $v$ со свойством $R_{sat}(1,*) \subseteq R_{init}.v_{init}$ и длиной не меньше $(kn-1)p_3$.
2) Компоненты $R_{sat}$ и $T_{sat}$ построены для определения выполнимости $\psi$. В случае *GOODCASE* для автомата $A$ найдется слово $u_{sat}$ длины $n+1$, такое, что ровно $n$ состояний заняты в строке $Row_{acc}(1,*)$ после применения этого слова к первой строке $R_{sat}$, полученной после шага 1. В случае *BADCASE* для автомата $B$ докажем, что в $Row_{acc}(1,*)$ занято более $n$ состояний после применения любого слова длины $n+1$ к первой строке $R_{sat}$, полученной после шага 1.
3) Компоненты $R_{fix}$ и $R_{acc}$ требуются для подсчета числа занятых состояний в строке $Row_{acc}(1,*)$ после предыдущего шага. Компонента $R_{fix}$ нужна для «фиксации» (наложения ограничений на структуру) слова, используемого на этом шаге для *BADCASE*.

4) Компонента $Q_{spec}$ играет специальную роль; в частности, с её помощью проверяется, что после предыдущего шага нет занятых состояний в $Q \setminus Q_{spec}$. Кроме того, компонента $Q_{spec}$ помогает сделать граф сильносвязным.

Определим формально множества состояний компонент автомата $A$:

$$R_{init} = \{R_{init}(i_r, i_c, i_f) : i_r \in [1, kn], i_c \in [-2, m], i_f \in [0, p_3 - 1]\},$$

где $i_r$ — номер строки $R_{init}$; $i_c$ — номер столбца $R_{init}$, при $i_c > 0$ соответствующий номеру дизъюнкта; $i_f$ — номер элемента в $R_{init}(i_r, i_c, *)$.

Следующие компоненты:

$$T_{sat} = \{T_{sat}(i_r, i_v, i_{vv}) : i_r \in [2, n+1], i_v \in [1, i_r], i_{vv} \in \{0, 1\}\},$$

где $i_r$ — номер строки $T_{sat}$, при $i_r \leqslant n$ соответствующий номеру переменной плюс 1, а $i_v$ (при $i_v \neq n+1$) и $i_{vv}$ соответствуют номеру переменной и ее значению.

Компонента $R_{sat}$ состоит из подкомпонент $R^+_{sat}$ и $R^-_{sat}$, где

$$R^-_{sat} = \{R_{sat}(i_v, i_c) : i_v \in [1, n+1], i_c \in [-2, 0]\},$$
$$R^+_{sat} = \{R_{sat}(i_v, i_c) : i_v \in [1, n], i_c \in [1, m], i_v = \max(i \in [1, n] : x_i \in c_{i_c} \text{ или } \neg x_i \in c_{i_c})\}.$$

В этом определении $i_v$ — это номер строки $R_{sat}$, при $i_v \neq n+1$ соответствующий номеру переменной, а $i_c$ — номер столбца $R_{sat}$, при $i_c > 0$ соответствующий номеру дизъюнкта.

Пусть $h = (4n + 13)(n - 2) + 20$, тогда $R_{fix} = \{R_{fix}(i_r) : i_r \in [1, h]\}$, т. е. компонента $R_{fix}$ состоит из одного столбца высоты $h$.

Компонента $R_{acc}$ состоит из $n - 1$ строки $Row_{acc}(i, *)$, и любой путь в $R_{acc}$ между двумя последовательными строками будет проходить через подкомпоненту $D_i$. Формально:

$$Row_{acc} = \{Row_{acc}(i_v, i_c) : i_v \in [1, n-1], i_c \in [4, 4n+2]\},$$
$$\forall i_v \in [1, n-1] \quad D_{i_v} = \{D_{i_v}(i_e) : i_e \in [1, 35]\},$$

где $i_v$ — номер строки $Row_{acc}$, $i_c$ — номер столбца $Row_{acc}$ и $i_e$ — номер элемента в $D_i$. Для $i \in [1, n-2]$ положим $D_i(35) = Row_{acc}(i+1, 4)$,

$$R_{acc} = Row_{acc} \cup D_1 \cup D_2 \cup \ldots \cup D_{n-1},$$

$h_s = p_{add}$ и определим $Q_{spec}$ следующим образом: $Q_{spec} = \{z\} \cup R_0 \cup C_0 \cup C_1 \cup C_2$, где $C_i = \{C_i(i_r) : i_r \in [1, h_s]\}$ для $i \in \{0, 1, 2\}$ — это столбец $Q_{spec}$ из $h_s$ элементов и $R_0 = \{R_0(i_c) : i_c \in [-2, m]\}$ — строка $Q_{spec}$ из $m + 3$ элементов. Для $i$ из $\{0, 1, 2\}$ обозначим $s_i = C_i(1)$.

При определении переходов в автомате $A$ будем использовать метки $LD_XY$, где $X$ — метка компоненты, а $Y$ — порядковый номер метки. Каждое такое определение переходов в автомате $A$ однозначно задает определение дуг графа $G(\psi)$, которое будем обозначать, заменяя первую букву $L$ на $E$ (например, $LD_{init}1$ и $ED_{init}1$). Будем говорить, что слово $w$ перемещает/отображает состояние $q$ в состояние $q'$, если $q.w = q'$, и слово $w$ «сжимает», или «синхронизирует» множество $S$, если $|S.w| = 1$. Термины направления перемещения следует понимать в соответствии с рис. 2, и они будут использоваться только для визуального представления.

Переходы из состояний $R_{init}$ в автомате определены следующим образом: буква $a$ перемещает состояния $R_{init}$ внутри столбцов по «спирали» относительно номера строки и третьего индекса. Формально:

$$R_{init}(i_r, i_c, i_f).a = \begin{cases} R_{init}(i_r, i_c, (i_f+1) \bmod p_3), \text{ если } i_f + 1 < p_3, & LD_{init}1 \\ R_{init}(i_r+1, i_c, (i_f+1) \bmod p_3), \text{ если } i_f + 1 = p_3 \text{ и } i_r < kn, & LD_{init}2 \\ R_{init}(1, i_c, (i_f+1) \bmod p_3), \text{ если } i_f + 1 = p_3 \text{ и } i_r = kn. & LD_{init}3 \end{cases}$$

Буква $b$ отображает состояния последней строки $R_{init}$ в первую строку $R_{sat}$ (т. е. $R_{init}(kn, *, *).b \subseteq R_{sat}(1, *)$). Более того, состояния нулевого столбца последней строки компоненты $R_{init}$ отображаются на первую строку $R_{sat}$ (т. е. $R_{init}(kn, 0, *).b = R_{sat}(1, *)$). Другие строки перемещаются в первую строку $R_{init}$ по этой букве:

$$R_{init}(i_r, i_c, i_f).b = \begin{cases} R_{init}(1, i_c, (i_f+1) \bmod p_3), \text{ если } i_r < kn, & LD_{init}4 \\ R_{sat}(1, i_c), \text{ если } i_r = kn, i_c \neq 0, & LD_{init}5 \\ R_{sat}(1, ((i_f+1) \bmod p_1) - 1), \text{если } i_r = kn, i_c = 0. & LD_{init}6 \end{cases}$$

Буква $c$ перемещает все состояния $R_{init}$ вверх, в первую строку $R_{init}$:

$$R_{init}(i_r, i_c, i_f).c = R_{init}(1, i_c, (i_f+1) \bmod p_3). \qquad LD_{init}7$$

Следующее замечание следует из определений дуг в $R_{init}$.

**Замечание 1.**

(A) Все буквы в автоматах $A$ и $B$ сохраняют «полное» множество остатков от деления третьего индекса в $R_{init}$ на $p_3$, пока состояния находятся в $R_{init}$, т. е. если для $x \in \Sigma$, $S \subseteq R_{init}$ выполняется $S.x \subseteq R_{init}$ и $\forall i \in [0, p_3 - 1] \exists q \in S \cap R_{init}(*, *, i)$, то также выполняется $\forall i \in [0, p_3 - 1] \exists q \in S.x \cap R_{init}(*, *, i)$.

(B) $Image(R_{init}(1, *, 0), p_{add}) \subseteq R_{init}$; другими словами, кратчайший путь, ведущий из $R_{init}(1, *, 0)$ вне $R_{init}$, имеет длину больше $p_{add} = p_3(kn - 1)$.

Рассмотрим теперь компоненты $R_{sat}$ и $T_{sat}$, которые «кодируют» $\psi$. Заметим, что образ множества $\{-2, -1, 0\}$ под действием функции $\varphi(x) = -0{,}5x^{2\smile}2{,}5x + 1$ равен $\{1, 3, 4\}$, и определим:

$$R_{sat}(i_v, i_c).x = \begin{cases} R_{init}(1, i_c, 0), \text{ если } x = c, & LD_{sat}1 \\ R_{fix}(\varphi(i_c)), \text{ если } x \in \{a, b\}, i_c \leqslant 0, i_v = n+1, & LD_{sat}2 \\ R_{sat}(i_v + 1, i_c), \text{ если } x \in \{a, b\}, i_c \leqslant 0, i_v \leqslant n. & LD_{sat}3 \end{cases}$$

Для $i_v \leqslant n$ и $i_c > 0$:

$$R_{sat}(i_v, i_c).x = \begin{cases} T_{sat}(i_v + 1, i_v, 0), \text{ если } x = a \text{ и } \neg x_{i_v} \in c_{i_c}, & LD_{sat}4 \\ T_{sat}(i_v + 1, i_v, 1), \text{ если } x = b \text{ и } x_{i_v} \in c_{i_c}, & LD_{sat}5 \\ R_{sat}(i_v + 1, i_c), \text{ если } x = a \text{ и не выполняется } (\neg x_{i_v} \in c_{i_c}), & LD_{sat}6 \\ R_{sat}(i_v + 1, i_c), \text{ если } x = b \text{ и не выполняется } (x_{i_v} \in c_{i_c}); & LD_{sat}7 \end{cases}$$

$$T_{sat}(i_r, i_v, i_{vv}).x = \begin{cases} s_{i_{vv}+1}, \text{ если } x = c, & LD_{sat}8 \\ T_{sat}(i_r + 1, i_v, i_{vv}), \text{ если } x \in \{a, b\}, i_r \leqslant n, & LD_{sat}9 \\ Row_{acc}(1, 2(2i_v + i_{vv})), \text{ если } x \in \{a, b\}, i_r = n + 1. & LD_{sat}10 \end{cases}$$

Будем писать $S \mapsto^r T$, если любой путь из подмножества $S$ длины не меньше $r$ содержит вершины из подмножества $T$. Заметим, что

$$\text{если} \quad S \mapsto^{r_1} T_1 \mapsto^{r_2} T_2, \ \text{ то} \quad S \mapsto^{r_1+r_2} T_2. \tag{1}$$

Следующее замечание непосредственно следует из определений дуг в $R_{sat}$ и $T_{sat}$ и свойств введенного отношения.

**Замечание 2.** Пусть $i_c \in [-2, m]$; тогда

(A) Если $i_c \leqslant 0$, то $R_{sat}(*, i_c) \mapsto^{n+1} R_{init}(1, i_c, 0) \cup R_{fix}$ (по $ED_{sat}1, 2, 3$).

(B) Если $i_c > 0$, то $R_{sat}(*, i_c) \mapsto^{n+1} R_{init}(1, i_c, 0) \cup \{s_1, s_2\} \cup R_{acc}$ (по $ED_{sat}1, 4, \ldots, 7$).

Введем линейный порядок между состояниями, находящимися в одной строке $Row_{acc}$ в соответствии с номерами столбцов компоненты $Row_{acc}$. Пусть $S$ — некоторое множество. Назовем две вершины из $S$, лежащие в одной строке $Row_{acc}$, *последовательными*, если между ними (в соответствии с порядком) нет других вершин из $S$. Назовем *$d$-переходами* в графе $G(\psi)$ переходы, помеченные буквой $d \in \Sigma$ в автомате $A$. Следующая лемма показывает, как компоненты $R_{sat}$ и $T_{sat}$ кодируют формулу $\psi$.

**Лемма 2.**

*GOODCASE*: Существует слово $u_{sat}$ длины $n+1$, такое, что $R_{sat}(1, *).u_{sat} \cap R_{fix} = = R_{fix}(\{1, 3, 4\})$, и если положить $S = R_{sat}(1, *).u_{sat} \cap R_{acc}$, то для $S$ верны следующие свойства:

(G0) $S \subseteq Row_{acc}(1, *)$, т. е. $S$ полностью содержится в первой строке;

(G1) номера столбцов состояний из $S$ четны;

(G2) $|S| = n$;

(G3) расстояние между любыми двумя последовательными состояниями из $S$ в строке равно $2, 4$ или $6$;

(G4) $Row_{acc}(1, 4n) \in S$, $Row_{acc}(1, 4n + 2) \notin S$.

*BADCASE*: Пусть при применении слова $v_{sat}$ длины $n+1$ к $R_{sat}(1, *)$ используются только $a$- и $b$-переходы; тогда $R_{sat}(1, *).v_{sat} \cap R_{fix} = R_{fix}(\{1, 3, 4\})$, и если положить $S = R_{sat}(1, *).v_{sat} \cap R_{acc}$, то для $S$ верны следующие свойства:

(B0) $S \subseteq Row_{acc}(1, *)$, т. е. $S$ полностью содержится в первой строке;

(B1) номера столбцов состояний из $S$ четны;

(B2) $|S| > n$.

***Доказательство.*** Рассмотрим сначала ситуацию *GOODCASE*. Так как $\psi$ выполнима, то существует набор значений переменных $x_1 = b_1$, $x_2 = b_2$, …, $x_n = b_n$, такой, что $\psi(b_1, b_2, \ldots, b_n)$ истинна, или, другими словами, существует минимальный номер переменной $j = j(i)$ для каждого дизъюнкта $c_i$, такой, что либо $b_j = 1$, $x_j \in c_i$, либо $b_j = 0$, $\neg x_j \in c_i$. Сопоставим слово $u_{sat}$ этому набору значений переменных следующим образом. Положим $u_{sat}(n + 1) = a$; $u_{sat}(i) = a$, если $b_i = 0$, и $u_{sat}(i) = b$ в противном случае.

Из определений $LD_{sat}2, 3$ и условия $u_{sat} \in \{a, b\}^{n+1}$ следует, что $R^-_{sat}(1, *).u_{sat} = = R_{fix}(\{1, 3, 4\})$. Осталось доказать, что выполняются свойства множества $S = = R_{sat}(1, *).u_{sat} \cap R_{acc}$. Для этого рассмотрим образ некоторого состояния $q = R_{sat}(1, i_c)$ под действием $u_{sat}$, где $i_c > 0$:

$$\begin{aligned} q.u_{sat} &= R_{sat}(1, i_c).u_{sat}(1 \ldots j(i_c) \ldots n + 1) = \\ &= T_{sat}(j(i_c) + 1, j(i_c), b_{j(i_c)}).u_{sat}(j(i_c) + 1 \ldots n + 1) = \\ &= T_{sat}(j(i_c) + (n + 1) - j(i_c), j(i_c), b_{j(i_c)}).u_{sat}(n + 1) = \\ &= T_{sat}(n + 1, j(i_c), b_{j(i_c)}).u_{sat}(n + 1) = Row_{acc}(1, 2(2j(i_c) + b_{j(i_c)})). \end{aligned} \tag{$EQ_1$}$$

Из условия нормализации (C1) и равенства $R_{sat}(1,i_c).u_{sat} = R_{acc}(1, 2(2j(i_c)+b_{j(i_c)}))$ следует свойство (G2), потому что $\{j(i_c) : i_c \in [1,m]\} = [1,n]$. Свойства (G1), (G3) выполняются, потому что разность между номерами столбцов $2(2j+b_j)$ и $2(2(j+1)+$ $+b_{j+1})$ четна и меньше либо равна 6 и $\{j(i_c) : i_c \in [1,m]\} = [1,n]$. Свойство (G0) выполняется, так как $Row_{acc}(1, 2(2j(i_c)+b_{j(i_c)})) \in Row_{acc}(1,*)$, а свойство (G4) следует из равенства $(EQ_1)$ и того, что $b_n = 0$ по (C2). Итак, случай *GOODCASE* доказан.

Рассмотрим теперь *BADCASE*. Так как при применении слова $v_{sat}$ к $R_{sat}(1,*)$ нет использования $c$-переходов, то по определениям $ED_{sat}2,\ldots,7,10$ получаем, что

$$R^-_{sat}(1,*).v_{sat} = R_{fix}(\{1,3,4\}) \quad \text{и} \quad R^+_{sat}(1,*).v_{sat} \subseteq R_{acc}(1,*).$$

Осталось доказать свойства (B1) и (B2) для $S = R_{sat}(1,*).v_{sat} \cap R_{acc}$. В силу (C1) для любого $j \in [1,n]$ существует номер $i_c = i_{c(j)}$, такой, что $c_{i_c} = (x_j \vee \neg x_j)$. Следовательно,

$$\begin{aligned} R_{sat}(1,i_{c(j)}).v_{sat} &= R_{sat}(j,i_{c(j)}).v_{sat}(j\ldots n+1) = \\ &= T_{sat}(j+1,j,a_1).v_{sat}(j+1\ldots n+1) = \\ &= T_{sat}(n+1,j,a_1).v_{sat}(n+1) = \\ &= R_{acc}(1,2(2j+a_1)),\ \text{где } a_1 \in \{0,1\}, j \in [1,n]. \end{aligned} \qquad (EQ_2)$$

Из этого равенства следуют (B1) и неравенство $|R_{sat}(1,*).v_{sat} \cap R_{acc}(1,*)| \geqslant n$. Чтобы показать, что неравенство строгое, ввиду равенства $(EQ_2)$ достаточно доказать, что существует номер $j \in [1,n]$, такой, что

$$R_{acc}(1,2(2j)) \in R_{sat}(1,*).v_{sat} \quad \text{и} \quad R_{acc}(1,2(2j+1)) \in R_{sat}(1,*).v_{sat}. \qquad (EQ_3)$$

Так как $R^+_{sat}(1,i).v_{sat} \subseteq R_{acc}(1,*)$, то для любого номера дизъюнкта $i_c \in [1,m]$ существует номер переменной $i_v(i_c) \in [1,n]$, такой, что

$$R_{sat}(1,i_c).v_{sat}(1\ldots i_v(i_c)) = R_{sat}(i_v(i_c),i_c).v_{sat}(i_v(i_c)) = T_{sat}(i_v(i_c)+1, i_v(i_c), i_{vv}(i_c)).$$

Предположим, что для всех $i_{c_1}, i_{c_2} \in [1,m]$ равенство номеров переменных $i_v(i_{c_1}) =$ $= i_v(i_{c_2})$ влечет равенство значений переменных $i_{vv}(i_{c_1}) = i_{vv}(i_{c_2})$. Для $j \in [1,n]$ положим $b_j = 1$, если $i_v(i_c) = j$, $i_{vv}(i_c) = 1$, и $b_j = 0$ в противном случае; тогда $\psi(b_1,b_2,\ldots,b_n) = 1$. Это верно, потому что для любого номера дизъюнкта $i_c$ выполняется $b_{i_v(i_c)} = i_{vv}(i_c)$. Следовательно, предположение противоречит тому, что $\psi$ невыполнима в *BADCASE*. Поэтому существуют некоторые числа $j, i_{c_1}, i_{c_2}$, такие, что $i_v(i_{c_1}) = i_v(i_{c_2}) = j$, $i_{vv}(i_{c_1}) = 0$, $i_{vv}(i_{c_2}) = 1$. Из этих равенств и определений дуг $ED_{sat}4,5,9$ непосредственно следует $(EQ_3)$, а следовательно, и доказательство леммы 2 в случае *BADCASE*. ■

Перейдем теперь к определению переходов в компонентах $R_{fix}$ и $R_{acc}$. Дуги $D_i$ одинаковы для всех $i$ и изображены на рис. 3. Символ $E$ на рисунке обозначает алфавит $\{a,b\}$, а метки $E^k$ на дугах заменяют $k-1$ промежуточных вершин и соответствующих переходов по $a,b$ между начальной и конечной вершинами соответствующей дуги. Символы $e_1(i)$ и $e_2(i)$ могут быть одной из букв $a,b$ и будут определены в процессе доказательства.

Переходы для $R_{fix}$ и $R_{acc}$ удобно представить, используя длины некоторых префиксов слова, фиксируемого при помощи компоненты $R_{fix}$ на шаге 3. По определению положим $h_1 = 3$, $h_2 = h_3 = \ldots = h_{n-1} = 12+4n$, $h_n = 13$ и обозначим

$w_i = w_i(d_1, d_2, \ldots, d_i) = a^3 d_1 a^{h_2} d_2 \ldots d_{i-1} a^{h_i} d_i$ для $i \in [1, n]$ и некоторых букв $d_1$, $d_2$, …, $d_n$. Заметим, что высота $h$ столбца $R_{fix}$ равна $|w_n| + 3$.

Обозначим $R^2_{fix} = \{R_{fix}(\{|w_i|, |w_i| + 2, |w_i| + 3\}) : i \in [1, n-1]\}$ и определим

$$q.x = \begin{cases} R_{init}(1,0,0), \text{ если } x = c \text{ и } q \in R_{fix} \setminus R_{fix}(\{h-3, h-1, h\}), & LD_{acc}1 \\ R_{fix}(i_r + 1), \text{ если } q = R_{fix}(i_r), i_r < h \text{ и} & \\ \qquad (x = a \text{ или } (x = b \text{ и } q \in R^2_{fix})), & LD_{acc}2 \\ R_{init}(1,0,0), \text{ если } x = b \text{ и } q \in R_{fix} \setminus R^2_{fix}, & LD_{acc}3 \\ s_0, \text{ если } x = c \text{ и } q \in R_{fix}(\{h-3, h-1, h\}) \cup D_{n-1}(35), & LD_{acc}4 \\ s_1, \text{ если } x = c \text{ и } q \in R_{acc} \setminus D_{n-1}(35), & LD_{acc}5 \\ Row_{acc}(i_v, i_c + 1), \text{ если } q = Row_{acc}(i_v, i_c), i_c < 4n + 2 \text{ и } x \neq c, & LD_{acc}6 \\ D_{i_v}(1), \text{ если } q = Row_{acc}(i_v, 4n + 2) \text{ и } x \neq c. & LD_{acc}7 \end{cases}$$

Переходы в $D_i$ по $a, b$ (кроме помеченных $e_1(i)$, $e_2(i)$, $E - e_1(i)$, $E - e_2(i)$) определяются по рис. 3. Для возможности сослаться на определения дуг в $D_i$ будем использовать метку $ED_{acc}8$. Следующее замечание говорит о назначении компоненты $R_{fix}$.

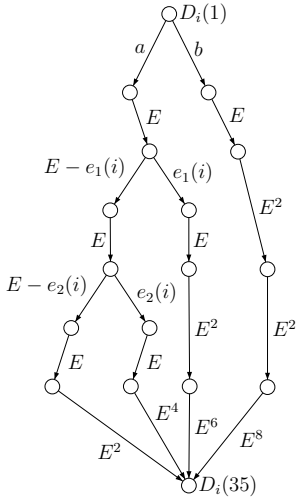


Рис. 3. Подкомпонента $D_i$

**Замечание 3.**

*GOODCASE*: Пусть $w = w_n(b, b, \ldots, b, c)$; тогда $w$ имеет следующие свойства: $w(1) = a$, $R_{fix}(\{1,3,4\}).w = s_0$ и для любого собственного префикса $w'$ слова $w$ справедливо $R_{fix}(\{1,3,4\}).w' \subseteq R_{fix}$.

*BADCASE*: Пусть слово $w$ обладает свойствами, указанными в утверждении для *GOODCASE*; тогда $w = w_n(d_1, d_2, \ldots, d_n)$ для некоторых $d_1, d_2, \ldots, d_n \in \Sigma$.

***Доказательство.*** В случае *GOODCASE* докажем по индукции для $i \in [0, |w|-1]$, что $R_{fix}(\{1,3,4\}).w(1 \ldots i) = R_{fix}(\{i+1, i+3, i+4\})$. База индукции для $i = 0$ очевидна. Докажем шаг индукции для $i$. Если $w(i) = a$, то по $LD_{acc}2$ получаем $R_{fix}(\{i, i+2, i+3\}).a = R_{fix}(\{i+1, i+3, i+4\})$. В противном случае $w(i) = b$; тогда $i = |w_j|$ для некоторого $j$, поэтому $R_{fix}(\{i, i+2, i+3\}) \subseteq R^2_{fix}$, и по $LD_{acc}2$ снова получаем

$R_{fix}(\{i, i+2, i+3\}).b = R_{fix}(\{i+1, i+3, i+4\})$. Следовательно, шаг индукции доказан. По заключению индукции получаем, что

$$R_{fix}(\{1,3,4\}).w = R_{fix}(\{1,3,4\}).w(1\ldots|w|-1)c = R_{fix}(\{h-3, h-1, h\}).c = s_0$$

и при применении слова $w(1\ldots|w|-1)$ состояния из $R_{fix}(\{1,3,4\})$ перемещаются внутри $R_{fix}$.

Рассмотрим теперь случай *BADCASE*. Так как только состояния из $R_{fix}(\{h-3, h-1, h\})$ могут перейти в $s_0$ из $R_{fix}$ и по условию $R_{fix}(\{1,3,4\}).w(1\ldots|w|-1) \subseteq R_{fix}$, то $R_{fix}(\{1,3,4\}).w(1\ldots|w|-1) \subseteq R_{fix}(\{h-3, h-1, h\})$. Все переходы внутри $R_{fix}$ (по $ED_{acc}2$) увеличивают на 1 индекс состояния в столбце $R_{fix}$, поэтому $R_{fix}(\{1,3,4\}).w(1\ldots|w|-1) = R_{fix}(\{h-3, h-1, h\})$ и для любого $i \in [0, h-4]$ и $j \in \{1,3,4\}$ должно выполняться $R_{fix}(j).w(1\ldots i) = R_{fix}(i+j)$ и $|w| = |w_n| = h-3$.

Покажем индукцией по $k < |w_n|$, что если для $i \in [1, k-1]$ слово $w(1\ldots k-1)$ является префиксом $w_n$ для некоторых $d_1, d_2, \ldots, d_n$, то и $w(1\ldots k)$ — также префикс $w_n$. База индукции очевидна, так как по условию $w(1) = a$.

Если для некоторого $k \in [3, h-2]$ из состояния $R_{fix}(3).w(1\ldots k-3) = R_{fix}(k)$ есть только один переход внутри $R_{fix}$, то $w(k) = w(k-2)$, так как $R_{fix}(k)$ также равно $R_{fix}(1).w(1\ldots k-1)$ (т. е. $R_{fix}(1)$ также проходит через эту вершину под действием перехода $w(k)$ и должно остаться в $R_{fix}$). Аналогично, если для $k \in [3, h-3]$ из состояния $R_{fix}(4).w(1\ldots k-4) = R_{fix}(k)$ есть только один переход внутри $R_{fix}$, то $w(k) = w(k-3)$. Следовательно, для всех $k \in [2, h-2]$, таких, что $R_{fix}(k) \notin R^2_{fix}$, выполняется $w(k) = w(k-2)$, и для всех $k \in [4, h-3]$, таких, что $R_{fix}(k) \notin R^2_{fix}$, выполняется $w(k) = w(k-3)$.

Заметим, что для $k \in \{2,3\}$ утверждение следует из соотношений $a = w(1) = w(3)$, $w(2) = w(3) = w(5)$. Пусть $k > 3$; если $k$ совпадает с длиной одного из $w_i$, то шаг индукции выполнен для $d_i = w(k)$. В противном случае, так как $k > 3$ и длины $w_i$ отличаются более чем на 1, то или $k-2$, или $k-3$ не совпадает с длиной ни одного из $w_i$ и, следовательно, выполняется одно из соотношений $w(k) = w(k-2) = a$ или $w(k) = w(k-3) = a$, что влечет шаг индукции. Заключение индукции дает необходимый результат. ■

Отметим также несколько несложных свойств компонент $R_{fix}, R_{acc}$.

**Замечание 4.**

(A) $R_{fix} \mapsto^h \{R_{init}(1,0,0), s_0\}$ (по $ED_{acc}1\ldots4$).
(B) $Row_{acc}(k,*) \mapsto^{4n-2} Row_{acc}(k, 4n+2)$ (по $ED_{acc}5, 6$).
(C) $Row_{acc}(k, 4n+2) \mapsto^{15} \{D_k(35), s_1\}$ (по $ED_{acc}7, 8$).
(D) $Row_{acc}(1,*) \mapsto^{h+4n-7} \{s_0, s_1\}$ (по B, C и равенству $(4n-2)(n-1) + 15(n-1) = h + 4n - 7$).
(E) $R_{sat}(1, i_c) \mapsto^{h+5n-6} \{R_{init}(1,0,0), R_{init}(i_c,0,0), s_0, s_1, s_2\}$ (по свойству $\mapsto$, пунктам A, B замечания 2 и пунктам A, D этого замечания).
(F) $Row_{acc}(k, 4n+2) \mapsto^{4n+13} \{Row_{acc}(k+1, 4n+2), s_1\}$ (по B и C).
(G) $Row_{acc}(k, 4n+2) \mapsto^{(n-k-1)(4n+13)+15} \{s_0, s_1\}$ (по F и C).

В замечании 3 показано, как при помощи $R_{fix}$ фиксируется шаблон слова. Далее отметим, как могут перемещаться состояния в $R_{acc}$ под действием таких слов в обоих случаях.

**Замечание 5.** Пусть $S$ — такое подмножество состояний, что для $k = 1$ выполняются следующие свойства.

*GOODCASE*:

(G0) $S \subseteq Row_{acc}(k, *)$, т. е. $S$ полностью содержится в $k$-й строке;
(G1) номера столбцов состояний из $S$ четны;
(G2) $|S| \leqslant n - k + 1$;
(G3) расстояние между любыми двумя последовательными состояниями из $S$ в строке равно $2, 4$ или $6$;
(G4) $Row_{acc}(k, 4n) \in S, Row_{acc}(k, 4n + 2) \notin S$.

Тогда существует дораскраска компоненты $R_{acc}$ (т. е. определение $e_1(i), e_2(i)$), такая, что $S.w_n(b, b, \ldots, b, c) = s_0$.

*BADCASE*:

(B0) $S \subseteq Row_{acc}(k, *)$, т. е. $S$ полностью содержится в $k$-й строке;
(B1) номера столбцов состояний из $S$ четны;
(B2) $|S| > n - k + 1$.

Тогда для всяких таких $d_1, d_2, \ldots, d_n \in \Sigma$, что любой собственный префикс $w'$ слова $w_n(d_1, d_2, \ldots, d_n)$ оставляет $S$ внутри $R_{acc}$ (т. е. $S.w' \subseteq R_{acc}$), выполняется $S.w_n(d_1, d_2, \ldots, d_n) \neq s_0$.

***Доказательство.*** Рассмотрим случай *GOODCASE*.

Обозначим $w = w_n(b, b, \ldots, b, c)$ и докажем индукцией по $k \in [1, n]$ утверждение о том, что $D_1, D_2, \ldots, D_{k-1}$ можно раскрасить так, что для множества $T_k = S.w(1 \ldots |w_k| - 4)$ выполнены свойства (G0) – (G4) для $k$. База индукции следует из условий замечания. Докажем шаг индукции для $k + 1$. Пусть $t$ обозначает разность номеров столбцов между максимальным (равным $Row_{acc}(k, 4n)$ по (G4)) и предшествующим состоянием $Row_{acc}(k, 4n - t)$ из $T_k$ в $k$-й строке. Раскрасим $D_k$ в зависимости от значения $t$ так, чтобы путь, помеченный $a^{14-t}$, помечал путь из $D_k(1)$ в $D_k(35)$. Это можно сделать, положив

$$\begin{aligned} &e_1(k) = a, \quad e_2(k) = a, \quad \text{если} \quad t = 2; \\ &e_1(k) = b, \quad e_2(k) = a, \quad \text{если} \quad t = 4; \\ &e_1(k) = b, \quad e_2(k) = b, \quad \text{если} \quad t = 6. \end{aligned}$$

Тогда под действием слова $w(|w_k| - 3 \ldots |w_{k+1}| - 4) = a^3 b a^{9+4n}$ (при $k < n - 1$) состояние $Row_{acc}(k, 4n)$ будет двигаться по пути

$$\begin{aligned} &Row_{acc}(k, 4n) \to Row_{acc}(k, 4n + 1) \to Row_{acc}(k, 4n + 2) \to \\ &\to D_k(1) \to \ldots \to D_k(35) = Row_{acc}(k + 1, 4) \to Row_{acc}(k + 1, 4n), \end{aligned}$$

где путь из $D_k(1)$ в $D_k(35)$ помечен $ba^{13}$, а все остальные состояния из $Row_{acc}(k, i) \in T_k$ будут двигаться по пути

$$\begin{aligned} &Row_{acc}(k, i) \to Row_{acc}(k, i + 1) \to \ldots \to D_k(1) \to \\ &\to \ldots \to D_k(35) = Row_{acc}(k + 1, 4) \to Row_{acc}(k + 1, i + t), \end{aligned}$$

где путь из $D_k(1)$ в $D_k(35)$ помечен $a^{14-t}$. Таким образом, оба состояния $Row_{acc}(k, 4n - t)$ и $Row_{acc}(k, 4n)$ перейдут в состояние $Row_{acc}(k + 1, 4n)$, а разность номеров столбцов между остальными состояниями не изменится. Из этого следует, что все свойства (G0) – (G4) выполнены для $k + 1$ и множества $T_{k+1} = S.w(1 \ldots |w_{k+1}| - 4)$. Заметим, что для случая $k = n - 1$ доказательство такое же, но рассматриваются пути только до

$D_k(35)$. В конечном итоге получаем, что $S.w_n(b,b,\ldots,b,c) = D_{n-1}(35).c = s_0$, и случай *GOODCASE* доказан.

Докажем *BADCASE* аналогичным способом, добавив к утверждению свойство (B5): $Row_{acc}(k, 4n{+}2) \notin S$. Обозначим $w = w_n(d_1, d_2, \ldots, d_n)$ и докажем индукцией по $k \in [1, n]$ свойства (B0, B1, B2, B5) для $T_k = S.w(1 \ldots |w_k| - 4)$. База индукции снова следует из условий, кроме свойства (B5), которое докажем на шаге индукции без использования предположения. Докажем шаг индукции для $k+1$. Рассмотрим, как перемещаются вершины из $T_k$ под действием слова $f = w(|w_k| - 3 \ldots |w_{k+1}| - 4) = a^3 d_k a^{9+4n}$. Докажем (B0) и (B5). Так как нет переходов из $Row_{acc}(k+1, *)$ в $Row_{acc}(k, *) \cup D_k$, то ввиду свойств B, C замечания 4 получаем, что $T_{k+1} \subseteq Row_{acc}([k{+}1, n{-}1], *) \cup D_{k+1} \ldots \cup D_{n-1}$. Пусть некоторое состояние $q \in T_{k+1}$ совпадает с $Row_{acc}(k{+}1, 4n{+}2)$ или не содержится в $Row_{acc}(k{+}1, *)$. Ввиду условия G замечания 4 максимальный путь из $q$ внутри $R_{acc}$ имеет длину меньше $(n-k-2)(4n+13)+15$. Однако по условиям путь из $q$ под действием слова $w(|w_{k+1}| - 3 \ldots |w| - 1)$ длины $(n-k-2)(4n+13)+17$ должен лежать в $R_{acc}$. Получили противоречие, и свойства (B0) и (B5) доказаны. Заметим, что для доказательства (B5) не использовалось предположение индукции. Следовательно, оно выполнено и для базы индукции.

Так как только на 4-й позиции слова $f$ стоит буква, отличная от $a$, и состояние $Row_{acc}(k, 4n+2) \notin T_k$, то по (B1) $Row_{acc}(k, 4n+1) \notin T_k$, поэтому длина кратчайшего пути из $T_k$ в состояние $D_k(1)$ не меньше 4. Следовательно, все состояния из $T_k$, кроме, быть может, одного ($Row_{acc}(k, 4n+2)$), перемещаются по пути, помеченному $a^{|f|}$ (так как по $ED_{acc}6$ для состояний из $Row_{acc}$ $a$- и $b$-переходы совпадают). Поскольку этот путь имеет четную длину и не может содержать циклов, получается, что разница номеров столбцов состояний из $T_{k+1}$ оставшихся состояний такая же, как и у прообразов из $T_k$. Отсюда следуют свойства (B1) и (B2). ■

Рассмотрим компоненту $Q_{spec}$. Пусть $q \in Q_{spec}$ и $x \in \Sigma$. Тогда

$$q.x = \begin{cases} R_{init}(1, i_c, 0), \text{ если } q = R_0(i_c) \text{ и } x = c, & LD_{spec}1 \\ R_0(i_c - 1), \text{ если } q = R_0(i_c) \text{ и } i_c > 1, x \neq c, & LD_{spec}2 \\ z, \text{ если } q = R_0(1) \text{ и } i_c > 1, x \neq c, & LD_{spec}3 \\ s_0, \text{ если } q = z \text{ и } x = c, & LD_{spec}4 \\ C_i(i_r + 1), \text{ если } q = C_i(i_r), i_r < h_s, & LD_{spec}5 \\ R_0(m), \text{ если } q = C_i(h_s). & LD_{spec}6 \end{cases}$$

Заметим, что граф $G(\psi)$ теперь полностью определен, хотя раскраска $A$ еще задана не до конца (не определены $e_i(1), e_i(2)$).

Для доказательства сильной связности графа рассмотрим рис. 4. Здесь вершины графа соответствуют подмножествам графа $G(\psi)$, а переход из подмножества $S_1$ в подмножество $S_2$, помеченный некоторым числом $k$ (если не указано, то предполагается 1) и, возможно, некоторым условием (в круглых скобках) на индексы вершин множеств $S_1, S_2$, соответствует тому, что $Image(S'_1, k) \supseteq S'_2$ и из любой вершины множества $S'_1$ можно перейти в вершину из $S'_2$, где множества $S'_1$ и $S'_2$ получены из $S_1$ и $S_2$ соответственно наложением указанного на переходе условия (если множество содержит индекс, на который накладывается условие). Например, переход из $R_{sat}(*, i_c)$ в множество $T_{sat}$, помеченный числом $n+1$ и условием $i_c > 0$, означает, что $Image(R^+_{sat}, n+1) \supseteq T_{sat}$ и из любой вершины из $R^+_{sat}$ можно перейти в некоторое состояние $T_{sat}$. Достижимость $T_{sat}$ для этого перехода следует из условий (C1)

и $ED_{sat}4, 5, 9$, а существование перехода из $R^+_{sat}$ в $T_{sat}$ следует из определения $R^+_{sat}$ и $ED_{sat}4, 5$. Остальные переходы в графе подмножеств следуют непосредственно из определений переходов графа $G(\psi)$.

Заметим также, что любая вершина $q$ графа $G(\psi)$ принадлежит, по крайней мере, одному подмножеству $S$ на рис. 4, причем множество $Image(\{q\}, 1)$ содержится в объединении множеств, в которые ведут переходы из множества $S$, и самого множества $S$ (кроме вершин, окруженных пунктирной линией — $s_0, s_1, s_2$). Таким образом, по рисунку для любой вершины можно найти множество, в котором содержится $Preimage(\{q\}, 1)$.

Сильная связность графа $G(\psi)$ следует из того, что граф подмножеств на рис. 4 является сильносвязным, так как из $z$ есть путь в любую вершину графа подмножеств и из любой вершины графа подмножеств есть путь в $z$.

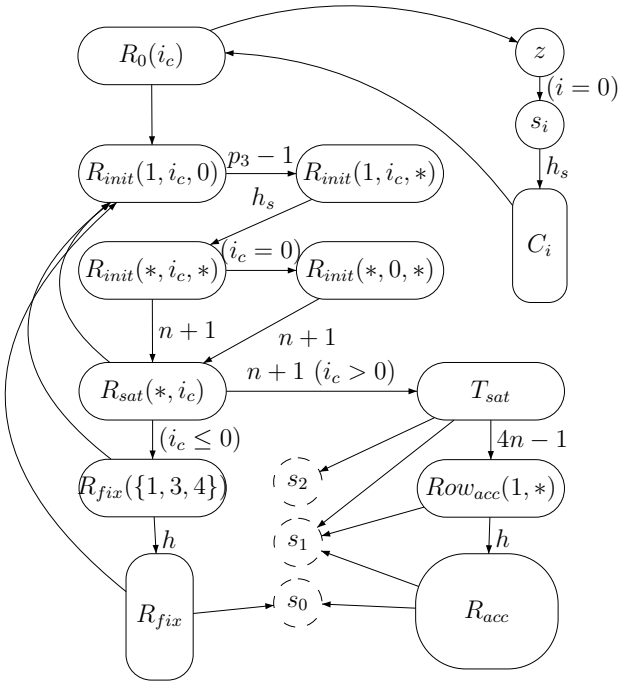


Рис. 4. Граф подмножеств $G(\psi)$

В следующем замечании показаны основные свойства $Q_{spec}$.

**Замечание 6.**

(A) Для $i \in \{0, 1, 2\}$ любой путь, начинающийся вне $C_i$ и проходящий через $C_i$, содержит $s_i$ (так как только переходы из $ED_{spec}4$ оканчиваются в $C_i(j)$ для $j > 1$).

(B) Для $r \in [0, h_s - 1]$ и $i \in \{0, 1, 2\}$ любой путь длины $r$ из $s_i$ оканчивается в $C_i(r+1)$ и любой путь длины $r$, оканчивающийся в $C_i(r + 1)$, начинается в $s_i$ (по (A) и $ED_{spec}4$).

(C.1) $Preimage(R_0, h_s) = Q_{spec} \setminus \{z\}$.

(C.2) $Preimage(R_{init}(*, i, *), 1) \subseteq R_{init}(*, i, *) \cup R_{sat}(*, i) \cup R_0$ для $i \neq 0$.

(C.3) $Preimage(R_{init}(*, 0, *), 1) \subseteq R_{init}(*, 0, *) \cup R_{sat}(*, 0) \cup R_0 \cup R_{fix}$.

(C.4) $Preimage(R_{sat}(*, i), 1) \subseteq R_{init}(*, \{0, i\}, *) \cup R_{sat}(*, \{0, i\})$.

(D) $Preimage(R_{init}(*,i,*),h_s) \subseteq R_{init}(*,\{0,i\},*) \cup R_{sat}(*,\{0,i\}) \cup R_{fix} \cup Q_{spec}$ для $i \in [-2,m]$ по пунктам (C.1 ... 4).
(E.1) $Image(R_{init}(*,i_c,*),1) \subseteq R_{init}(*,i_c,*) \cup R_{sat}(*,i_c)$ для $i_c > 0$ в силу $ED_{init}5$.
(E.2) $Image(R_{sat}(*,i_c),1) \subseteq R_{init}(*,i_c,*) \cup R_{sat}(*,i_c) \cup T_{sat}$ для $i_c > 0$ в силу $ED_{sat}1, 4 \dots 7$.
(E.3) $Image(T_{sat},1) \subseteq T_{sat} \cup R_{acc} \cup \{s_0,s_1,s_2\}$ в силу $ED_{sat}8,9,10$.
(E.4) $Image(R_{acc},1) \subseteq R_{acc} \cup \{s_0,s_1,s_2\}$ в силу $ED_{acc}5,6,7$.
(E.5) $Image(\{s_0,s_1,s_2\},h_s) \subseteq C_0 \cup C_1 \cup C2$ в силу $ED_{spec}4$.
(F) $Image(R_{init}(*,i_c,*),h_s) \subseteq R_{init}(*,i_c,*) \cup R_{sat}(*,i_c) \cup T_{sat} \cup R_{acc} \cup C_0 \cup C_1 \cup C2$ для $i_c > 0$ в силу $(E.1 \dots 5)$.

Доказательство оценок для длин синхронизирующих слов $u$ и $v$ автоматов $A$ и $B$ соответственно проведем следующим образом: для *GOODCASE* пошагово построим синхронизирующее слово. При этом на каждом шаге будем отслеживать образ множества состояний автомата после применения уже построенного слова. Для *BADCASE* также пошагово докажем, что синхронизирующее слово $v$ должно перемещать состояния аналогично тому, как перемещаются состояния в *GOODCASE* под действием слова $u$, либо можно выбрать несколько состояний в образе соответствующего префикса $v$, для которых значение *LRadius* по следующей лемме будет достаточно большим для получения требуемой оценки длины оставшегося суффикса $v$ (по п. 2 леммы 1).

**Лемма 3.** Справедливы следующие неравенства для функции *LRadius*:

(A) $LRadius(\{q_1,q_2\}) \geqslant p_3(kn-1)$, где $q_1 \in \{s_0,s_1,s_2\}, q_2 \neq q_1$.
(B) $LRadius(\{q_1,q_2\}) \geqslant p_3(kn-1)$, где $q_1 = R_{init}(1,i_{c_1},i_{f_1})$, $i_{c_1} \neq 0$, $q_2 \in R_{init}(*,i_{c_2},*) \cup \cup R_{sat}(*,i_{c_2}) \cup T_{sat} \cup R_{fix} \cup R_{acc}$, $0 \neq i_{c_2} \neq i_{c_1}$.
(C) $LRadius(\{q_1,q_2\}) \geqslant p_3(kn-1)$, где $q_1 = R_{init}(1,i_{c_1},i_{f_1})$, $i_{c_1} = 0$, $q_2 \in R_{init}(*,i_{c_2},*) \cup \cup R_{sat}(*,i_{c_2}) \cup T_{sat} \cup R_{acc}$, $0 < i_{c_2}$.
(D) $LRadius(\{q_f,q_2,q_3\}) \geqslant p_3(kn-1)$, где $q_f = R_{sat}(1,i_{c_1})$, $q_2 = R_{init}(i_{r_2},0,i_{f_2})$, $q_3 \in R_{init}(*,i_{c_3},*) \cup R_{sat}(1,i_{c_3})$, $i_{r_2} < kn$, $0 < i_{c_3} \neq i_{c_1}$ и ($i_{f_2} \leqslant 3p_2+1$ или $i_{r_2} = 1$).
(E) $LRadius(\{q_1,q_2\}) \geqslant p_3(kn-1)$, где $q_1 \in Image(R_{fix},1)$, $q_2 \in Image(R_{acc},1)$, $\{q_1,q_2\} \neq \{s_0\}$, $q_1 \notin (R_{fix} \cup R_{acc})$ или $q_2 \notin (R_{fix} \cup R_{acc})$.

***Доказательство.*** Пункт (A) непосредственно следует из пункта (B) замечания 6, так как для $r \in [0, h_s-1]$ путь из $q_2$ длины $r$ не может заканчиваться в $C_i(r+1)$, где заканчивается путь длины $r$ из $q_1$.

По пункту (B) замечания 1 кратчайший путь из $R_{init}(1,i_{c_1},*)$ вне $R_{init}$ не меньше $h_s = p_3(kn-1)$. Поэтому для пунктов (B), (C) выполняется $LRadius(q_1, Q \setminus R_{init}(1,i_{c_1},*)) \geqslant h_s$. Следовательно, по п. 3 леммы 1

$$LRadius(\{q_1,q_2\},Q) \geqslant \min(h_s, LRadius(\{q_1,q_2\}, R_{init}(1,i_{c_1},*))),$$

т. е. достаточно доказать, что $LRadius(\{q_1,q_2\}, R_{init}(1,i_{c_1},*)) \geqslant h_s$. По п. 6 леммы 1 для этого достаточно показать, что $q_2 \notin Preimage(R_{init}(1,i_{c_1},*),h_s)$. А это, в свою очередь, следует из пункта (D) замечания 6.

Для доказательства пункта (D) предположим от противного, что есть пути $P_f$, $P_2$, $P_3$ одинаковой длины $r$ меньше $p_3(kn-1)$, ведущие в одну из вершин $q_f$, $q_2$, $q_3$ соответственно. Из выбора $q_2$ и $ED_{init}1,2,4,7$ следует, что любой путь из $q_2$ длины не больше $p_3 - 3p_2 > 15n^2$ заканчивается в $R_{init}(*,0,*)$. Если для некоторого $r \leqslant 15n^2$ путь $P_f(1 \dots r)$ или $P_3(1 \dots r)$ оканчивается в $\{s_0,s_1,s_2\}$, то мы получим противоречие

нашего предположения с пунктом (A), примененного для вершин $P_2(r)$, $P_f(r)$ или $P_2(r)$, $P_3(r)$ соответственно.

Из пункта (E) замечания 4 следует, что для некоторого $r_0 \leqslant h+5n-6 \leqslant 15n^2$ путь $P_f(1 \dots r_0)$ оканчивается в $\{R_{init}(1,0,0), R_{init}(1,i_c,0), s_0, s_1, s_2\}$. Так как $h+5n-6 \leqslant \leqslant 15n^2$, то путь $P_f(1 \dots r_0)$ должен оканчиваться в $R_{init}(1,i_c,0)$.

Из того, что $i_{c_3} > 0$ и путь $P_3(1 \dots 15n^2)$ не проходит через $\{s_0, s_1, s_2\}$, следует, что $P_3(1 \dots 15n^2)$ целиком лежит в $R_{init}(*, i_{c_3}, *) \cup R_{sat}(*, i_{c_3}) \cup T_{sat} \cup R_{acc}$. Следовательно, мы получаем противоречие с пунктом (C) для вершин $P_f(r_0) = R_{init}(1,i_c,0)$ и $P_3(r_0)$. Таким образом, пункт (D) доказан.

Докажем теперь пункт (E). По определению переходов $ED_{acc}$, $ED_{fix}$ получаем, что $q_1 \in R_{fix} \cup \{R_{init}(1,0,0), s_0\}$, а $q_2 \in R_{acc} \cup \{s_1, s_0\}$. Тогда из условия

$$q_1 \notin (R_{fix} \cup R_{acc}) \text{ или } q_2 \notin (R_{fix} \cup R_{acc})$$

следует, что или $q_1 \in \{R_{init}(1,0,0), s_0\}$, или $q_2 \in \{s_1, s_0\}$. Если $q_1$ или $q_2$ совпадает с $s_0$, то другая вершина по условию не совпадает с $s_0$, и по пункту (A) получим требуемое неравенство. Если $q_2$ совпадает с $s_1$, то из $q_1 \in R_{fix} \cup \{R_{init}(1,0,0), s_0\}$ следует $q_1 \neq s_1$, и снова по пункту (A) получаем требуемое неравенство. Осталось рассмотреть случай, когда $q_1 = R_{init}(1,0,0)$ а $q_2 \in R_{acc}$. В этом случае получаем нужное неравенство по пункту (C). ■

В следующей лемме полностью доказывается корректность сведения для *GOOD-CASE* при помощи утверждений о свойствах компонент графа $G(\psi)$.

**Лемма 4.** Существуют дораскраска автомата $A$ (определение $e_1(i)$, $e_2(i)$) и слово $u$ длины не больше $p(m,n)$, синхронизирующее автомат $A$.

***Доказательство.*** Положим $u_{init} = ca^{p_3(kn-1)}b$. Тогда $|u_{init}| = (kn-1)p_3 + 2$ и по определениям переходов $LD_{init}$ получаем, что $R_{init}.u_{init} = R_{sat}(1,*)$. По замечанию 2, существует слово $u_{sat}$ длины $n+1$, такое, что множество $S = R_{sat}(1,*).u_{sat} = = R_{init}.u_{init}u_{sat}$ удовлетворяет замечанию 5, поэтому можно раскрасить $R_{acc}$ (определить $e_1(i)$, $e_2(i)$) так, чтобы для $u_{acc} = w_n(b,b,\dots,b,c)$ выполнялось равенство $s_0 = S.u_{acc} = R_{init}.u_{init}u_{sat}u_{acc}$. Заметим также, что $|u_{acc}| = |w_n| = (4n+13)(n-2)+18$, поэтому

$$|u| = |u_{init}| + |u_{sat}| + |u_{acc}| = ((kn-1)p_3 + 2) + (n+1) + (18 + (4n+13)(n-2)) \leqslant p.$$

Осталось доказать, что $Q.u = s_0$. Так как $u(1) = c$, то $Q.u(1) = Q.c \subseteq R_{init}(1,0,0) \cup \cup Q_{spec}$. Поскольку $R_{init}(1,*,0).u(2 \dots |u|) = s_0$, то нужно показать, что $Q_{spec}.u(2 \dots |u|) = = s_0$. Так как высота любого столбца $C_i$ плюс ширина строки $R_0$ меньше, чем $|u(2 \dots |u|-1)| = |u|-2$, и алфавит слова $u(2 \dots |u|-1)$ состоит из букв $\{a,b\}$, то по определениям $LD_{spec}1 \dots 6$ получаем, что $Q_{spec}.u(2 \dots |u|-1) = z$. Следовательно, $Q_{spec}.u(2 \dots |u|) = s_0$ и слово $u$ синхронизирующее. ■

Рассмотрим теперь *BADCASE*. Следующая лемма соответствует шагу 1 и использует свойства компоненты $R_{init}$.

**Лемма 5.** Существует префикс $v'_{init}$ слова $v$, такой, что $R_{init}(1,*,*).v'_{init} \cap R_{sat} \neq \neq \varnothing$. Если $v'_{init}$ — кратчайший префикс с этим свойством, тогда $|v'_{init}| > (kn-1)p_3$ и существует слово $v_{dop}$, такое, что $v_{init} = v'_{init}v_{dop}$ — префикс слова $v$ и $R_{sat}(1,*) \subseteq \subseteq R_{init}.v'_{init}v_{dop}$.

***Доказательство.*** Прежде всего, такой префикс $v'_{init}$ у слова $v$ существует, потому что любые два состояния из различных столбцов $R_{init}$ не могут быть сжаты внутри $R_{init}$ и все исходящие дуги из $R_{init}$ заканчиваются либо в $R_{init}$, либо в $R_{sat}$. Заметим также, что $|v'_{init}| > p_{add} = p_3(kn-1)$ по пункту (B) замечания 1.

Для доказательства леммы осталось показать, что найдется слово $v_{dop}$, такое, что $R_{sat}(1,*) \subseteq R_{init}.v'_{init}v_{dop}$.

Можно предполагать, что $R_{sat}(1,t) \not\subseteq R_{init}.v'_{init}$ для некоторого $t \in [-2,m]$. В противном случае $R_{sat}(1,*) \subseteq R_{init}(1,*,*).v'_{init}$ и в качестве $v_{dop}$ можно взять пустое слово. Во всех остальных случаях покажем, что можно либо найти подходящее слово $v_{dop}$, либо выбрать несколько состояний из $Q.v'_{init}$, для которых значение $LRadius$ не меньше $p_3(kn-1)-1$, и тем самым получить противоречие с $(E_v)$, используя неравенства

$$|v| \geqslant |v'_{init}| + LSynch_B(Q.v'_{init}) > p_3(kn-1) + LRadius(Q.v'_{init}) \geqslant \\ \geqslant 2p_3(kn-1) \geqslant 2p_{add} \geqslant (2-0{,}5\varepsilon)p. \qquad (EN_1)$$

Так как по условию множество $R_{init}(1,*,*).v'_{init} \cap R_{sat}$ не пустое, то из него можно выбрать состояние $q_f = R_{sat}(1,i_{c_1})$ для некоторого $i_{c_1} \in [-2,m]$.

Без ограничения общности предположим, что $v'_{init}(|v'_{init}|) = a$, т. е. $v'_{init} = wa$ для подходящего слова $w$. Тогда по выбору $w$ и по пункту (A) замечания 1 для любого номера столбца $i_c$ любой префикс слова $w$ сохраняет полное множество остатков от деления на $p_3$ координат вектора множества $R_{init}(1,i_c,*)$. Формально:

$$\forall i_c \in [-2,m],\ i_f \in [0,p_3-1],\ l_1 \in [1,|w|] \\ \exists i_r \in [1,kn]\ R_{init}(i_r,i_c,i_f) \in R_{init}(1,*,*).w(1\ldots l_1). \qquad (E_0)$$

Вспомним, что $v'_{init} = wa$, и обозначим $S = R_{init}(1,*,*).w$. Рассмотрим все возможные случаи.

С л у ч а й 1. $\exists j \in [0,3p_2]\ R_{init}(kn,0,j) \notin S$. Тогда по $(E_0)$ существует номер $i < kn$, такой, что $q'_2 = R_{init}(i,0,j) \in S$. Далее, так как $m > 3$, то по $(E_0)$ для $0 < i_{c_3} \neq i_{c_1}$ можем выбрать состояние $q'_3 = R_{init}(i_{r_3},i_{c_3},i_{f_3})$ из множества $S$ и положить $q_2 = q'_2.a$, $q_3 = q'_3.a$. Таким образом, есть три занятых состояния $q_f,q_2,q_3 \in S.a = R_{init}(1,*,*).v'_{init}$, такие, что $q_f = R_{sat}(1,i_{c_1})$, $q_2 = R_{init}(i,0,j).a = R_{init}(i_{r_2},0,i_{f_2})$, $q_3 \in R_{init}(*,i_{c_3},*) \cup \cup R_{sat}(1,i_{c_3})$, где $i_{r_2} < kn$, $0 < i_{c_3} \neq i_{c_1}$ и ($i_{f_2} \leqslant 3p_2+1$ или $i_{r_2} = 1$). По пункту (D) леммы 3 $LRadius(\{q_f,q_2,q_3\}) \geqslant (kn-1)p_3$, и получаем противоречие по $(EN_1)$.

С л у ч а й 2. $\forall j \in [0,3p_2]\ R_{init}(kn,0,j) \in S$. Обозначим через $x$ букву $w(|w|)$. Тогда

$$\forall j \in [0,3p_2-1]\ R_{init}(kn,0,j).x = R_{init}(kn,0,j+1), \qquad (E_2)$$

потому что только $R_{init}(kn,0,j)$ может перейти по $x$ в $R_{init}(kn,0,j+1)$ и какое-то состояние из $S.x^{-1}$ переходит по $x$ в $R_{init}(kn,0,j+1)$.

Пусть $t$ — максимальное число, такое, что $v = wx^tv_2$ и $v_2(1) \neq x$. Заметим, что если $t > 0$, то $x = a$, потому что $v'_{init} = wa$.

С л у ч а й 2.1. $t > p_2$. Тогда $x = a$ и $R_{init}(kn,0,0) \in S$. По $(E_0)$ выберем состояние в множестве $S$ из столбца $R_{init}$ с положительным номером, т. е. $q_3 = R_{init}(i_{r_3},i_{c_3},i_{f_3}) \in S$, где $i_{c_1} \neq i_{c_3} > 0$. Хотя кратчайший путь из $q_2 = R_{init}(kn,0,0)$ имеет длину меньше $p_2$, но так как $wa^t$ — префикс $v$, $q_2 \in Q.w$ и выполняется равенство $(E_1)$, то $q_2$ остается в $R_{init}(kn,0,*)$ при применении $a^t$ и $t > p_2$. Следовательно, оценка на $LRadius$ для состояний $q_f,q_2,q_3$ получается так же, как и в пункте (D) леммы 3, с той разницей, что в этом случае путь из $q_2$ длины $p_2$ зафиксирован. Следовательно,

$$LRadius(q_f.x,q_2.x,q_3.x) \geqslant LRadius(q_f,q_2,q_3) - 1 \geqslant p_3(kn-1) - 1,$$

и получаем противоречие по $(EN_1)$.

С л у ч а й 2.2. $t \leqslant p_2$. Так как $3p_2 \geqslant p_1$ и выполняются условия случая 2, то $R_{init}(kn, 0, j) \in S$ для $j \in [0, p_1]$. По $(E_2)$ получаем, что $R_{init}(kn, 0, j).x^t v_2(1) = R_{init}(kn, 0, j + t).v_2(1)$, потому что $j + t < p_2 + p_1 < 3p_2 - 1$. Так как $R_{init}(kn, 0, j + t).x = R_{init}(kn, 0, j + t + 1)$ и $v_2(1) \neq x$, то из определений дуг $ED_{1,6,7}$ следует, что $R_{init}(kn, 0, j+t).v_2(1)$ совпадает с $R_{sat}(1, ((j+t+1) \bmod p_1)-1)$ либо с $R_{init}(1, 0, j+t+1)$.

Если для любого $j \in [0, p_1]$ состояние $R_{init}(kn, 0, j).x^t v_2(1)$ совпадает с $R_{sat}(1, ((j + t + 1) \bmod p_1) - 1)$, то $R_{sat}(1, *) \subseteq S.x^t v_2(1)$, и в качестве $v_{dop}$ можно выбрать $x^{t-1}v_2(1)$.

В противном случае для некоторого $j \in [0, p_1]$ состояние $R_{init}(kn, 0, j).x^t v_2(1)$ совпадает с $R_{init}(1, 0, j + t + 1)$. Обозначим это состояние через $q_1$. Используя $(E_2)$, выберем $q'_2 \in R_{init}(*, i_{c_2}, *) \cap S$, где $i_{c_2} > 0$. Так как $t \leqslant p_2 < h_s$ и $i_{c_2} > 0$, то по пункту (F) замечания 6 получаем $q_2 = q'_2.x^t \in R_{init}(*, i_{c_2}, *) \cup R_{sat}(*, i_{c_2}) \cup R_{acc} \cup C_0 \cup C_1 \cup C_2$. Если $q_2 \in C_0 \cup C_1 \cup C_2$, то можно выбрать $t_1 \leqslant t$ так, чтобы $\widetilde{q_2} = q'_2.x^{t_1} \in \{s_0, s_1, s_2\}$. Тогда если положить $\widetilde{q_1} = R_{init}(kn, 0, j).x^{t_1}$, то $\widetilde{q_1} \notin \{s_0, s_1, s_2\}$, так как $\widetilde{q_1}.x^{t-t_1} = q_1 \in R_{init}(1, *, *)$ и $Image(\{s_0, s_1, s_2\}, h_s) \subseteq Q_{spec}$ в силу $ED_{spec}5$. Следовательно, $\widetilde{q_1} \neq \widetilde{q_2}$ и $\widetilde{q_2} \in \{s_0, s_1, s_2\}$, и получаем, что $LRadius(\{\widetilde{q_1}, \widetilde{q_2}\}) \geqslant p_3(kn - 1)$ по пункту (A) леммы 3 и $\widetilde{q_1}, \widetilde{q_2} \in Q.v(1 \dots |w| + t_1)$.

Иначе $q_2 \in R_{init}(*, i_{c_2}, *) \cup R_{sat}(*, i_{c_2}) \cup R_{acc}$, и тогда $LRadius(\{q_1, q_2\}) \geqslant p_3(kn - 1)$ по пункту (C) леммы 3 и $q_1, q_2 \in Q.v(1 \dots |w| + t)$.

Следовательно, в случае 2.2 после применения более длинного (с длиной больше $|v'_{init}|$) префикса $v$ получаем несколько занятых состояний, значение $LRadius$ для которых не меньше $p_3(kn - 1)$, и тем самым получаем противоречие с $(E_v)$ аналогично $(EN_1)$. Таким образом, рассмотрены все возможные случаи, и лемма доказана. ■

Таким образом, шаг 1 завершен. Следующая лемма соответствует шагу 2 и позволяет отделить случай *BADCASE* от *GOODCASE*.

**Лемма 6.** Существует разложение $v = v_{init}v_{sat}v_2$, такое, что

$$|v_{sat}| = n + 1,\ R_{sat}(1, *).v_{sat} \cap R_{fix} = R_{fix}(\{1, 3, 4\})$$

и для $S = R_{sat}(1, *).v_{sat} \cap R_{acc}$ верны следующие свойства:

(B0) $S \subseteq Row_{acc}(1, *)$, т. е. полностью содержится в 1-й строке;
(B1) номера столбцов состояний из $S$ четны;
(B2) $|S| > n$.

***Доказательство.*** Так как $LRadius(R_{sat}(1, *)) > n + 1$ и по лемме 5 $R_{sat}(1, *) \subseteq R_{init}.v_{init}$, то существует разложение $v = v_{init}v_{sat}v_2$, где $|v_{sat}| = n + 1$. Если при применении $v_{sat}$ к $R_{sat}(1, *)$ нет использования $c$-переходов, то утверждение доказано по лемме 2. Предположим от противного, что при применении $v_{sat}$ используется $c$-переход; тогда существуют состояние $q'_1 = R_{sat}(1, i_{c_1})$ и минимальное натуральное число $t_1 \leqslant n + 1$, такие, что если $q_1 = q'_1.v_{sat}(1 \dots t_1)$, то

– $q_1 \in \{R_{init}(1, i_{c_1}, 0), s_1, s_2\}$, если $i_{c_1} > 0$; выберем $q_2 = R_{sat}(1, 0).v_{sat}(1 \dots t_1)$, тогда $q_2 \in R_{sat}(*, 0) \cup R_{fix} \cup R_{init}(1, 0, *)$;
– $q_1 = R_{init}(1, i_{c_1}, 0)$, если $i_{c_1} \leqslant 0$; выберем $q_2 = R_{sat}(1, 1).v_{sat}(1 \dots t_1)$, тогда $q_2 \in R_{sat}(*, 1) \cup R_{acc} \cup R_{init}(1, 1, *) \cup \{s_0, s_1, s_2\}$.

В обоих случаях $q_1, q_2 \in R_{init}.v_{init}v_{sat}(1 \dots t_1)$ и по пунктам (A),(B) или (C) леммы 3 получаем $|v| \geqslant |v_{init}| + LRadius(\{q_1, q_2\}) \geqslant 2(kn - 1)p_3 \geqslant 2p_{add} \geqslant (2 - 0{,}5\varepsilon)p$. Таким образом, получаем противоречие с $(E_v)$, и лемма доказана. ■

Следующая лемма соответствует шагу 3 для *BADCASE* и завершает доказательство для этого случая.

**Лемма 7.** *BADCASE*: Предположим, что $v = v_{init}v_{sat}v_{acc}$ и слово $v$ — синхронизирующее для $B$. Тогда $|v| > (2 - 0{,}5\varepsilon)p(m, n)$

***Доказательство.*** Заметим, что по лемме 6 после применения $v_{init}v_{sat}$ занятые состояния остались как в $R_{fix}$, так и в $R_{acc}$. Так как между $R_{fix}$ и $R_{acc}$ нет дуг, а $v_{acc}$ должно сжимать оставшиеся состояния, то у слова $v_{acc}$ есть префикс, который перемещает некоторое состояние из $(R_{fix} \cup R_{acc}) \cap Q.v_{init}v_{sat}$ вне $R_{fix} \cup R_{acc}$. Пусть $w_{acc}x$ — разложение кратчайшего префикса $v_{acc}$ с этим свойством.

По определению $w_{acc}x$ можно выбрать состояние $q'_1$ из $(R_{acc} \cup R_{fix}) \cap Q.v_{init}v_{sat}w_{acc}$ так, что если положить $q_1 = q'_1.x$, то $q'_1.x \notin (R_{acc} \cup R_{fix})$. Заметим, что по выбору префикса множество $Q.v_{init}v_{sat}w_{acc}$ пересекается как с $R_{fix}$, так и с $R_{acc}$. Следовательно, можно выбрать $q'_2 \in (R_{fix} \cup R_{acc}) \cap Q.v_{init}v_{sat}w_{acc}$ так, что если $q'_1 \in R_{fix}$, то $q'_2 \in R_{acc}$, или наоборот — если $q'_1 \in R_{acc}$, то $q'_2 \in R_{fix}$.

Если $((R_{acc} \cup R_{fix}) \cap Q.v_{init}v_{sat}w_{acc}).x \neq s_0$, то можно считать, что $\{q'_1.x, q'_2.x\} \neq \neq \{s_0\}$. Тогда по пункту (E) леммы 3 получаем, что $LRadius(\{q'_1.x, q'_2.x\}) \geqslant (kn-1)p_3$. Следовательно, $|v_{init}v_{sat}w_{acc}x| \geqslant |v_{init}v_{sat}w_{acc}x| + Lradius(\{q'_1.x, q'_2.x\}) \geqslant 2(kn-1)p_3$, и утверждение леммы доказано.

Осталось рассмотреть случай, когда $((R_{acc} \cup R_{fix}) \cap Q.v_{init}v_{sat}w_{acc}).x = s_0$. Без ограничения общности предположим, что $w_{acc}(1) = a$ (в противном случае доказательство идентично с точностью до подстановки на алфавите). Тогда по выбору $w_{acc}$ и равенству $((R_{acc} \cup R_{fix}) \cap Q.v_{init}v_{sat}w_{acc}).x = s_0$ оказываемся в условиях замечания 3 и получаем, что слово $w_{acc}x$ равно $w_n$ для некоторых $d_1, d_2, \ldots, d_n \in \Sigma$, т. е.

$$w_{acc}x = a^3 d_1 a^{12+4n} d_2 a^{12+4n} d_3 \ldots a^{12+4n} d_{n-1} a^{13} d_n.$$

Объединяя это с леммой 6 и выбором $w_{acc}x$, оказываемся в условиях замечания 5 для $S = Q.v_{init}v_{sat} \cap Row_{acc}(1, *)$ и получаем противоречие с равенством $((R_{acc} \cup R_{fix}) \cap \cap Q.v_{init}v_{sat}w_{acc}).x = s_0$. ■

Таким образом, теорема полностью доказана в обоих случаях. ■

Заметим, что из теоремы 1 напрямую не следует, что в предположении $\mathcal{P} \neq \mathcal{NP}$ не существует полиномиального алгоритма, находящего оптимальную раскраску, так как для её поиска не требуется находить длину кратчайшего синхронизирующего слова.

Если бы задача поиска кратчайшего синхронизирующего слова для автомата принадлежала классу $\mathcal{P}$, то полиномиальный алгоритм, находящий оптимальную раскраску, можно было бы достроить до полиномиального алгоритма, решающего задачу OCV, и получить противоречие с теоремой. Однако ввиду результата [8] для этой задачи не существует даже приближенного полиномиального алгоритма с любой конечной относительной погрешностью.

Тем не менее можно получить аналогичный результат для задачи поиска оптимальной раскраски. Заметим, что алгоритм, решающий OC с относительной погрешностью $\beta < 2$, должен для произвольного допустимого орграфа $G$ возвращать его синхронизирующую раскраску $A$, такую, что $\mathfrak{C}(A) \leqslant \beta \cdot \mathrm{OPT}(G)$.

**Следствие 1.** В предположении $\mathcal{P} \neq \mathcal{NP}$ любой полиномиальный алгоритм, решающий OC для класса графов со степенью исхода вершин 3, имеет относительную погрешность не меньше 2.

***Доказательство.*** Приведем здесь только план доказательства, опуская технические детали. Предположим от противного, что существует алгоритм $Alg$, решающий с относительной погрешностью $2-\varepsilon$ задачу OC. Используя такое же построение графа $G(\psi)$ по $\psi$, что и в теореме 1, для получения противоречия достаточно показать, как по раскраске $B$, возвращаемой алгоритмом $Alg$ на графе $G(\psi)$, определить выполнимость $\psi$.

Выполнимость $\psi$ определяется следующим образом. Без ограничения общности можно считать, что дуга из $R_{fix}(1)$ в $R_{fix}(2)$ помечена буквой $a$. Далее, для $i \in [1, n-1]$ рассчитаем длины путей внутри $D_i$, помеченных степенью буквы $a$, и в соответствии с этими значениями восстановим, какие состояния из $Row_{acc}(1,*)$ могли быть сжаты под действием слова $w_n(d_1, d_2, \ldots, d_n)$, где $d_i$ равен символу на $b$-переходе из $D_i(1)$. Это можно сделать, используя рассуждения из замечания 5 или рассмотрев прообраз $D_{n-1}(35)$ под действием слова $w_n$ без последнего символа. Заметим, что если для некоторого $j$ в $D_j$ нет пути, помеченного степенью $a$, то нужно доказать, что $\psi$ невыполнима. Это можно сделать аналогично доказательству $BADCASE$ в лемме 7.

Далее, из определения переходов в $R_{sat}$ и $T_{sat}$ однозначно восстанавливаются значения переменных $b_1, b_2, \ldots, b_n$, которые должны удовлетворять $\psi$ (так же, как это делалось для $BADCASE$ в замечании 2). Если эти значения действительно удовлетворяют $\psi$, то ясно, что формула $\psi$ выполнима. В противном случае нужно показать, что $\psi$ невыполнима. Для этого по построению достаточно доказать, что $\mathfrak{C}(B) \geqslant (2-0{,}5\varepsilon)p$.

Это утверждение доказывается аналогично случаю $BADCASE$, с той разницей, что вместо аргумента о том, что $\psi$ невыполнима, в замечании 2 используем аргумент о том, что $\psi$ должна быть выполнима именно на наборе $b_1, b_2, \ldots, b_n$, так как в противном случае получим такое множество занятых состояний после шага 2 внутри $Row_{acc}(1,*)$, которое ввиду раскраски $R_{acc}$ не может быть сжато словом $w_n(d_1, d_2, \ldots, d_n)$. ■

## 3. Случай двухбуквенного алфавита

Теорема доказана для трехбуквенного алфавита. Очевидно, что для однобуквенного алфавита задача не имеет смысла, так как единственный сильносвязный автомат с одной буквой является циклом по этой букве и не может быть синхронизируемым. Несложно также проверить, что если в алфавите больше трех букв, то, продублировав необходимое количество раз переходы по букве $c$ в автомате $A$ из теоремы, можно доказать такой же результат и в этом случае. Возникает естественный вопрос: верно ли утверждение для двухбуквенного алфавита? Следующее следствие показывает справедливость результата и в этом случае.

**Следствие 2.** В предположении $\mathcal{P} \neq \mathcal{NP}$ любой приближенный полиномиальный алгоритм, решающий задачу OCV для подкласса автоматов с двухбуквенным алфавитом, имеет относительную погрешность не меньше 2.

***Доказательство.*** Некоторые технические детали доказательства этого следствия опущены для краткости. Представим только изменения в структуре автомата $A$, которые показаны на рис. 5. Каждое состояние $q \in Q$, кроме $s_0$, заменяется тремя состояниями $q^{top}, q^a, q^b$, и переходы переопределяются следующим образом:

$$q^{top}.x = q^x, \quad q^x.a = (q.c)^{top}, \quad q^x.b = (q.x)^{top}.$$

Тогда буква $c$ соответствует $xa$, а буква $y \in \{a, b\} - yb$.

Состояние $s_0$ заменяется двумя состояниями ${s_0}^{top}, {s_0}^{bot}$. Обозначим состояние $C_0(2)$ через $g$. Тогда

$${s_0}^{top}.a = {s_0}^{bot}, \quad {s_0}^{top}.b = g^{top}, \quad {s_0}^{bot}.a = {s_0}^{bot}.b = g^{top}.$$

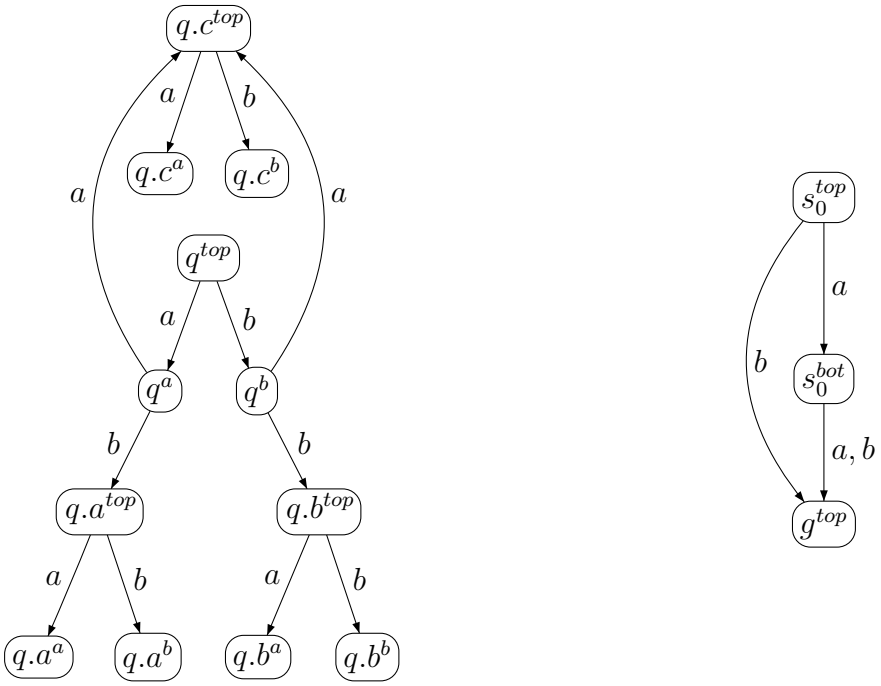


Рис. 5. Замены в автомате $A$

Легко проверить, что в случае *GOODCASE* $|u_{new}| = 2|u_{old}| + 1 \leqslant 2p(m,n)$, а в случае *BADCASE* $|v_{new}| \geqslant 2|v_{old}| \geqslant 2((2-0{,}5\varepsilon)p(m,n)) = (4-\varepsilon)p(m,n)$. Здесь $u_{old}, v_{old}$ соответствуют синхронизирующим словам для старых автоматов $A$ и $B$. Для завершения доказательства заметим, что отношение выражений $(4-\varepsilon)$ и $2p(m,n)$ стремится к 2 при $\varepsilon \to 0$. ■

Итак, мы доказали главный результат. Более того, мы показали, что он верен для случая общей степени исхода, равной 2.

В связи с доказанными результатами можно выделить два естественных открытых вопроса: о существовании эффективного алгоритма для некоторого подкласса допустимых графов и о минимальной возможной погрешности полиномиального алгоритма для задачи вычисления значения оптимальной раскраски.

## ЛИТЕРАТУРА

1. *Adler R. L., Weiss B., and Goodwyn L. W.* Equivalents of topological Markov shifts // Isr. J. Math. 1977. No. 27. P. 49–63.
2. *Trahtman A. N.* The Road Coloring Problem // arxiv:0709.0099. 2007.
3. *Volkov M.* Synchronizing Automata and the Cerny conjecture // LNCS. 2008. V. 5196. P. 11–27.
4. *Černý J.* Poznámka k homogénnym eksperimentom s konečnými automatami // Matematicko-fyzikalny Časopis Slovensk. Akad. Vied (in Slovak). 1964. V. 14. No. 3. P. 208–216.
5. *Adler R. L. and Weiss B.* Similarity of automorphisms of the torus // Mem. Amer. Math. Soc. 1970. V. 98. P. 1–43.
6. *Beal M. and Perrin D.* A quadratic algorithm for road coloring // arXiv:0803.0726. 2008.

7. *Eppstein D.* Reset sequences for monotonic automata // SIAM J. Comput. 1990. V. 19. P. 500–510.
8. *Berlinkov M.* Approximating the Minimum Length of Synchronizing Words is Hard // LNCS. 2010. V. 6072. P. 37–47.
9. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

2011 Прикладная теория автоматов №2(12)

DOI 10.17223/20710410/12/5

УДК 512.5

# СКЕЛЕТНЫЕ АВТОМАТЫ

В. Н. Салий

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

**E-mail:** SaliiVN@info.sgu.ru

Автомат (без выходов) называется сильносвязным, если любые два его состояния взаимно достижимы. Скелетный автомат характеризуется противоположным свойством: в нем никакие два различных состояния не являются взаимно достижимыми. Показано, что автомат тогда и только тогда допускает правильную нумерацию состояний, когда он скелетный (теорема 1). Предлагается метод, позволяющий для заданного автомата построить автомат с наименьшим возможным числом состояний, имеющий такую же решетку подавтоматов, при этом полученный автомат оказывается скелетным (теорема 2). Описывается процедура, позволяющая получить из данного автомата скелетный автомат путем удаления (замены петлями) минимального числа дуг в диаграмме переходов исходного автомата.

**Ключевые слова:** *автомат, сильносвязный автомат, скелетный автомат, правильная нумерация состояний, решетка подавтоматов.*

Под автоматом понимается тройка $\mathbf{A} = (S, X, \delta)$, где $S$ и $X$ — конечные непустые множества (соответственно множество состояний и множество входных сигналов), а $\delta : S \times X \to S$ — отображение, называемое функцией переходов. Запись $\delta(s, x) = s'$ для $s, s' \in S$, $x \in X$ означает, что автомат $\mathbf{A}$, находящийся в состоянии $s$, под действием входного сигнала $x$ переходит в следующий момент дискретного времени в состояние $s'$. Функция переходов естественным образом продолжается на множество $S \times X^*$, где $X^*$ — совокупность всех конечных слов над алфавитом $X$: по определению, $\delta(s, e) = s$ для любого $s \in S$ и пустого слова $e$, и $\delta(s, px) = \delta(\delta(s, p), x)$ для любых $s \in S$, $x \in X$, $p \in X^*$.

Автомату $\mathbf{A}$ сопоставляется диаграмма — мультиграф $G(\mathbf{A})$, вершинами которого являются элементы множества $S$ и дуги помечены элементами из $X$: из вершины $s$ в вершину $s'$ ведет дуга с меткой $x$, если $\delta(s, x) = s'$. Другим объектом, связанным с автоматом $\mathbf{A}$, является матрица переходов $M(\mathbf{A})$ — матрица размерности $m \times m$, где $m = |S|$, элементами которой являются подмножества множества $X$, при этом $[M(\mathbf{A})]_{i,j} = \{x \in X : \delta(s_i, x) = s_j\}$.

Говорят, что состояние $s'$ достижимо в автомате $\mathbf{A}$ из состояния $s$, если существует входное слово $p \in X^*$, такое, что $\delta(s, p) = s'$. В этом случае пишут $(s, s') \in \tau$, и так определенное отношение $\tau \subseteq S \times S$ называют отношением достижимости в автомате $\mathbf{A}$. Понятно, что $\tau$ рефлексивно и транзитивно, т. е. является квазипорядком. Его симметричная часть $\sigma = \tau \cap \tau^{-1}$ называется отношением взаимной достижимости в автомате $\mathbf{A}$. Классы эквивалентности $\sigma$ называют слоями автомата $\mathbf{A}$. В [1] было введено понятие каркаса автомата. Каркас автомата $\mathbf{A}$ — это упорядоченное множество $(F(\mathbf{A}), \leqslant)$, элементами которого являются слои автомата $\mathbf{A}$, а порядком на множестве слоев $F(\mathbf{A}) = S/\sigma$ — отношение, обратное отношению достижимости $\tau$, именно $\sigma(s') \leqslant \sigma(s) :\Longleftrightarrow (s', s) \in \tau^{-1} \Longleftrightarrow (s, s') \in \tau$. Очевидно, что для любого автомата $\mathbf{A}$

выполняются неравенства $1 \leqslant |F(\mathbf{A})| \leqslant |S|$. Если $|F(\mathbf{A})| = 1$, т. е. отношение $\sigma$ универсально, $\sigma = S \times S$, автомат $\mathbf{A}$ называется сильносвязным. Если же $|F(\mathbf{A})| = |S|$, т. е. отношение $\sigma$ тождественно, $\sigma = \Delta$, автомат $\mathbf{A}$ называется скелетным. В скелетном автомате отношение достижимости $\tau$ является порядком на множестве состояний $S$.

Поскольку в скелетном автомате $\mathbf{A}$ все слои одноэлементны, можно считать, что в этом случае каркасом является множество $S$, упорядоченное обратной достижимостью $\tau^{-1}$, так что $s \geqslant s'$ означает, что состояние $s'$ достижимо из состояния $s$.

Нумерацией состояний автомата называется биективное отображение множества его состояний $S$ на начальный отрезок $[1, m]$ натурального ряда. Нумерация, по определению, является правильной, если $(s_i, s_j) \in \tau \Longrightarrow i \geqslant j$, т. е. если состояния, достижимые из данного состояния, имеют меньшие, чем у него, номера.

**Теорема 1.** В автомате $\mathbf{A}$ существует правильная нумерация состояний тогда и только тогда, когда $\mathbf{A}$ — скелетный автомат.

***Доказательство.*** Пусть в автомате $\mathbf{A}$ существует правильная нумерация состояний, но $\mathbf{A}$ не является скелетным, т. е. $\sigma \neq \Delta$. Тогда в $\mathbf{A}$ найдутся два различных взаимно достижимых состояния, скажем $s_i$ и $s_j$. Так как $(s_i, s_j) \in \tau$, то $i > j$, а так как $(s_j, s_i) \in \tau$, то $j > i$, что невозможно. Значит, $\sigma = \Delta$ и $\mathbf{A}$ — скелетный автомат.

Пусть теперь, напротив, дан скелетный автомат $\mathbf{A}$ и нужно правильно пронумеровать его состояния. Построим диаграмму каркаса $F(\mathbf{A})$, отождествляя его элементы с соответствующими состояниями автомата $\mathbf{A}$. Под высотой элемента $s$ в каркасе $F(\mathbf{A})$ понимается наибольшая из длин убывающих цепей, начинающихся с $s$. Диаграмму строим следующим образом: если $l$ — наибольшая из высот элементов в каркасе, то на $l + 1$ горизонталях, пронумерованных снизу вверх числами $0, 1, \ldots, l$, расположим элементы каркаса в соответствии с их высотами, а затем, двигаясь снизу вверх, будем соединять прямолинейными отрезками каждый элемент $s$ с его нижними соседями (т. е. с теми элементами, за которыми $s$ непосредственно следует в смысле порядка в $F(\mathbf{A})$). Построив диаграмму каркаса $F(\mathbf{A})$, пронумеруем его элементы (т. е. состояния автомата $\mathbf{A}$) слева направо и снизу вверх. Состояние $s'$ достижимо из состояния $s$ в точности тогда, когда в диаграмме каркаса $F(\mathbf{A})$ есть восходящая ломаная от $s'$ к $s$. Но это означает, что в построенной нумерации $s$ имеет больший номер, чем $s'$, т. е. нумерация — правильная. ■

Подмножество $S' \subseteq S$ называется устойчивым в автомате $\mathbf{A}$, если $\delta(s, x) \in S'$ для любых $s \in S'$ и $x \in X$. Если $S'$ устойчиво в $\mathbf{A}$, то, ограничивая функцию переходов $\delta$ на $S' \times X$, получают подавтомат $\mathbf{A}' = (S', X, \delta)$. Совокупность $\mathrm{Sub}\mathbf{A}$ всех подавтоматов автомата $\mathbf{A}$, упорядоченная отношением $\mathbf{A}_1 \leqslant \mathbf{A}_2 \Longleftrightarrow S_1 \subseteq S_2$, где $\mathbf{A}_i = (S_i, X, \delta)$, $i = 1, 2$, является дистрибутивной решеткой. Это — решетка подавтоматов автомата $\mathbf{A}$.

Известно (см. [1]), что для каждого автомата $\mathbf{A}$ существует автомат $\mathbf{B}$ с двумя входными сигналами, такой, что $\mathrm{Sub}\mathbf{A} \cong \mathrm{Sub}\mathbf{B}$, и что не всегда найдется автономный (т. е. с $|X| = 1$) автомат $\mathbf{B}$ с такой же, как у $\mathbf{A}$, решеткой подавтоматов. Минимизацию по числу состояний дает следующее предложение.

**Теорема 2.** Пусть $\mathbf{A} = (S, X, \delta)$ и $\mathbf{B} = (T, Y, \gamma)$ — автоматы, такие, что $\mathrm{Sub}\mathbf{A} \cong \mathrm{Sub}\mathbf{B}$. Тогда $|T| \geqslant |F(\mathbf{A})|$. При этом существует скелетный автомат $\mathbf{B}$, такой, что $\mathrm{Sub}\mathbf{A} \cong \mathrm{Sub}\mathbf{B}$ и $|T| = |F(\mathbf{A})|$.

***Доказательство.*** Как показано в [2], $\mathrm{Sub}\mathbf{A} \cong \mathrm{Sub}\mathbf{B}$ тогда и только тогда, когда $F(\mathbf{A}) \cong F(\mathbf{B})$. Так как $|F(\mathbf{A})| \leqslant |S|$, то $|T| \geqslant |F(\mathbf{B})| = |F(\mathbf{A})|$. Определим автомат $\mathbf{B} = (T, Y, \gamma)$ следующим образом. Положим $T = F(\mathbf{A})$. Ветвлением элемента $a$ в конечном упорядоченном множестве называется количество $b(a)$ его нижних со-

седей. Пусть в каркасе $F(\mathbf{A})$ наибольшее из ветвлений элементов равно $t$. Возьмем $Y = \{y_1, y_2, \ldots, y_t\}$. Построим диаграмму переходов автомата $\mathbf{B}$. Ее вершинами будут элементы каркаса $F(\mathbf{A})$. Все ребра диаграммы каркаса $F(\mathbf{A})$ ориентируем сверху вниз. Пусть для $\sigma(s) \in F(\mathbf{A})$ будет $b(\sigma(s)) = k$. Присвоим дугам, исходящим из $\sigma(s)$, последовательно слева направо метки $y_1, y_2, \ldots, y_k$. Если $k < t$, то вершине $\sigma(s)$ соотнесем еще $t-k$ петель с метками $y_{k+1}, y_{k+2}, \ldots, y_t$. Проделав это для каждой вершины, получим мультиграф, из любой вершины которого исходят $t$ дуг с метками из $Y$. Это и есть диаграмма переходов автомата $\mathbf{B}$. Очевидно, что этот автомат скелетный и что $F(\mathbf{B}) = F(\mathbf{A})$. Из последнего равенства следует, что $\mathrm{Sub}\mathbf{A} \cong \mathrm{Sub}\mathbf{B}$ и $|T| = |F(\mathbf{A})|$. ■

Доказанная теорема соотносится со следующими известными фактами: для любого автомата $\mathbf{A}$ существуют сильносвязные (т. е. с универсальным отношением $\sigma$) автоматы $\mathbf{B}$ и $\mathbf{C}$, такие, что у $\mathbf{A}$ и $\mathbf{B}$ изоморфны решетки конгруэнций ($\mathrm{Con}\mathbf{A} \cong \mathrm{Con}\mathbf{B}$), у $\mathbf{A}$ и $\mathbf{C}$ изоморфны группы автоморфизмов ($\mathrm{Aut}\mathbf{A} \cong \mathrm{Aut}\mathbf{C}$). Теорема 2 показывает, что для любого автомата $\mathbf{A}$ существует скелетный (т. е. с тождественным отношением $\sigma$) автомат $\mathbf{B}$ с такой же, как у $\mathbf{A}$, решеткой подавтоматов ($\mathrm{Sub}\mathbf{A} \cong \mathrm{Sub}\mathbf{B}$).

Пусть $K$ — некоторый класс автоматов и $\mathbf{A} \notin K$. Как можно путем в том или ином смысле минимальных изменений в структуре автомата $\mathbf{A}$ получить из него автомат $\mathbf{A}'$ из класса $K$? В числе допустимых приемов реконструкции автомата можно рассматривать, например, отождествление некоторых вершин (факторизация), введение дополнительных состояний и/или входных сигналов (расширения), перенаправление дуг диаграммы переходов, удаление дуг (замена их петлями) и т. п.

Напомним, что конгруэнцией автомата $\mathbf{A} = (S, X, \delta)$ называется эквивалентность $\theta$ на множестве его состояний, согласованная с функцией переходов в том смысле, что $(\forall s, s' \in S)(x \in X)((s, s') \in \theta \Longrightarrow (\delta(s, x), \delta(s', x)) \in \theta)$. Если $\theta \in \mathrm{Con}\mathbf{A}$, то фактор-автомат автомата $\mathbf{A}$ по конгруэнции $\theta$ определяется как $\mathbf{A}/\theta = (S/\theta, X, \delta)$, где $\delta(\theta(s), x) = \theta(\delta(s, x))$ для любых $s \in S$, $x \in X$.

Одной из задач об оптимальных реконструкциях автоматов является построение для данного автомата сильносвязного фактор-автомата с наибольшим возможным числом состояний. Частный случай — факторизация на максимальный по числу состояний циклический автомат — рассматривался в [3] (циклический автомат — это такой автомат, в котором все входные сигналы действуют одинаково и все состояния образуют при этом цикл). В этом контексте покажем, как путем удаления минимального числа дуг в диаграмме $G(\mathbf{A})$ получить из данного автомата $\mathbf{A}$ скелетный автомат.

Из теоремы 1 непосредственно вытекает

**Следствие 1.** Автомат $\mathbf{A}$ тогда и только тогда является скелетным, когда при некоторой нумерации его состояний матрица переходов $M(\mathbf{A})$ будет нижней треугольной матрицей.

***Доказательство.*** То, что матрица $M(\mathbf{A})$ является нижней треугольной матрицей, равносильно тому, что соответствующая нумерация состояний автомата — правильная. ■

Прямолинейный подход к решению рассматриваемой задачи состоит в том, чтобы для каждой из $m!$ возможных нумераций состояний автомата $\mathbf{A}$ взять в соответствующей матрице $M(\mathbf{A})$ множество $\bigcup\limits_{i>j}[M(\mathbf{A})]_{ij} \subseteq X$, выбрать один из вариантов, при котором это множество будет минимальным по мощности, и построить матрицу $M'(\mathbf{A})$, полагая $[M'(\mathbf{A})]_{ij} = [M(\mathbf{A})]_{ij}$ при $i > j$ и $[M'(\mathbf{A})]_{ii} = \bigcup\limits_{j \geqslant i}[M(\mathbf{A})]_{ij}$. Автомат $\mathbf{A}'$ с матри-

цей переходов $M'(\mathbf{A})$ является скелетным, и он получен путем удаления из диаграммы $G(\mathbf{A})$ минимально возможного числа дуг.

Понятно, что предложенный способ ни в коей мере не является оптимальным для исполнения. Более эффективный способ дает обращение к известной в теории графов проблеме расконтуривания. Действуя в этом русле, удалим в диаграмме переходов $G(\mathbf{A})$ все метки дуг. Далее к полученному мультиграфу применим модификацию какой-либо процедуры удаления в графе минимального числа дуг, приводящей к бесконтурному графу (см., например, [4]). Затем в $G(\mathbf{A})$ каждую удаленную дугу заменим петлей в начале этой дуги и восстановим все метки дуг. Полученный мультиграф с помеченными дугами является диаграммой переходов скелетного автомата, дающего решение задачи.

## ЛИТЕРАТУРА

1. *Богомолов А. М., Салий В. Н.* Алгебраические основы теории дискретных систем. М.: Наука, 1997. 368 с.
2. *Салий В. Н.* Каркас автомата // Прикладная дискретная математика. 2010. №1(7). С. 63–67.
3. *Верзаков Г. Ф., Киншт Н. В., Рабинович В. И., Тимонен А. Г.* Введение в техническую диагностику. М.: Энергия, 1968. 224 с.
4. *Gill F. and Flexer R.* Periodic decomposition of sequential machines // J. Assoc. Comput. Mach. 1967. V. 15. P. 666–676.

# ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

DOI 10.17223/20710410/12/6

УДК 519.17

## МИНИМАЛЬНЫЕ РЕБЕРНЫЕ РАСШИРЕНИЯ НАПРАВЛЕННЫХ И ОРИЕНТИРОВАННЫХ ЗВЕЗД

М. Б. Абросимов

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

**E-mail:** mic@rambler.ru

Рассматриваются минимальные реберные $k$-расширения графов, которые получаются из звездного графа произвольной ориентацией ребер. Ранее было получено полное решение, описывающее минимальные вершинные и реберные $k$-расширения неориентированных звезд, а также минимальные вершинные $k$-расширения ориентированных звезд. В этой работе дается полное описание всех минимальных реберных $k$-расширений для ориентированных и направленных звезд.

**Ключевые слова:** *минимальное расширение, оптимальная отказоустойчивая реализация, отказоустойчивость, звездные графы.*

### Введение

*Ориентированным графом* (далее *орграфом*) называется пара $G = (V, \alpha)$, где $V$ — конечное непустое множество, называемое *множеством вершин*, а $\alpha$ — отношение на множестве вершин $V$, называемое *отношением смежности*. Орграф с антисимметричным отношением смежности называется *направленным графом*, или *графом*. Граф с симметричным и антирефлексивным отношением смежности называется *неориентированным графом*, или *неографом*. Основные определения даются по работе [1].

Граф с пустым отношением смежности называется *вполне несвязным* и обозначается $O_n$, где $n$ — число вершин. Полный диграф без петель называется *турниром*. Циклической тройкой называется 3-вершинный турнир, у которого в каждой вершине есть одна исходящая и одна входящая дуга.

*Симметризацией* орграфа $\overrightarrow{G} = (V, \alpha)$ называется неограф $G = (V, (\alpha \cup \alpha^{-1}) \backslash \Delta)$, то есть симметризация орграфа получается заменой дуг ребрами и удалением петель.

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

Два графа $G_1 = (V_1, \alpha_1)$ и $G_2 = (V_2, \alpha_2)$ называются *изоморфными*, если можно установить взаимно однозначное соответствие $\varphi : V_1 \to V_2$, сохраняющее отношение смежности: $(u, v) \in \alpha_1 \Leftrightarrow (\varphi(u), \varphi(v)) \in \alpha_2$ для любых $u, v \in V_1$. Изоморфизм графа на себя называется *автоморфизмом*. Две вершины $u$ и $v$ называются *подобными*, если существует автоморфизм $\varphi : \varphi(v) = u$. Аналогично, две дуги называются *подобными*, если существует автоморфизм, переводящий одну в другую.

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

1) $G^*$ является реберным $k$-расширением $G$, то есть граф $G$ вкладывается в каждый подграф графа $G^*$, получающийся удалением любых его $k$ ребер;
2) $G^*$ содержит $n$ вершин, то есть $|V^*| = |V|$;
3) $\alpha^*$ имеет минимальную мощность при выполнении условий 1 и 2.

Понятие минимального $k$-расширения введено на основе понятия оптимальной $k$-отказоустойчивой реализации, которое было предложено Дж. П. Хейзом в работе [2] для исследования отказоустойчивости технических систем. Позднее Дж. П. Хейз совместно с Ф. Харари в работах [3, 4] обобщили модель и распространили ее на случай реберной отказоустойчивости. В этих работах рассматривались неориентированные графы. Для ориентированных графов количество результатов существенно меньше. А. В. Киреева [5] рассматривала вершинную отказоустойчивость в ориентированных графах. Ею описана вершинная оптимальная 1-отказоустойчивая реализация произвольного функционального графа. Т. Санг и др. [6] использовали модель Хейза для нахождения вершинной и реберной оптимальной 1-отказоустойчивой реализации ориентированного цикла. В [7] доказывается, что задача проверки, является ли граф реберным (вершинным) $k$-расширением заданного графа, принадлежит классу NP-полных. Это означает, что едва ли появится эффективное решение задачи построения минимальных вершинных или реберных $k$-расширений для заданного произвольного графа, а новые результаты будут связаны с частным решением задачи для каких-либо семейств графов.

*Неориентированной звездой* (далее просто *звездой*) или *звездным графом* называется полный двудольный граф вида $K_{1,n}$, где $n > 0$ (то есть одна вершина смежна с $n$ несмежными вершинами). *Ориентированной звездой* будем называть орграф, симметризация которого является звездой. Ориентированную звезду будем обозначать $Z_{m,n,p}$, где $m$ и $n$ — число вершин с единственной соответственно исходящей и входящей дугой, а $p$ — число вершин с одной входящей и одной исходящей дугой. *Направленной звездой* будем называть ориентированную звезду, являющуюся диграфом, и обозначать $Z_{m,n}$ (у направленной звезды $p = 0$). Вершину, являющуюся концом или началом каждой дуги звезды, будем называть *центральной*. Вершину, соединенную со всеми остальными вершинами орграфа, будем называть *полной*. Полная вершина может быть соединена с другими вершинами исходящей дугой, входящей или парой встречных дуг. Центральная вершина является единственной полной вершиной звезды. Легко определить количество дуг звезды $Z_{m,n,p}$: $m + n + 2p$.

В работе [8] удалось найти общее решение для описания минимальных вершинных $k$-расширений предполных графов, то есть графов, которые имеют хотя бы одну вершину, смежную со всеми остальными. Звездный граф является частным случаем предполного графа, и в работе [9] дается полное описание минимальных вершинных и реберных $k$-расширений для неориентированных звезд. Дадим вспомогательное определение и приведем соответствующую теорему относительно минимальных реберных $k$-расширений.

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

**Теорема 1** [9]. При $k \leqslant n/2$ граф $K_{1+2k} + O_{n-2k}$ является единственным минимальным реберным $k$-расширением графа $K_1 + O_n$. При $k > n/2$ граф $K_1 + O_n$ не имеет минимальных реберных $k$-расширений.

## 1. Минимальные реберные 1-расширения

Обозначим через $ZK_{m,n,p}$ семейство графов, получающихся из звезды $Z_{m,n}$ добавлением $p-1$ «центральной» вершины, соединением их между собой и центральной вершиной звезды $Z_{m,n}$ парами встречных дуг. Каждая из добавленных центральных вершин соединяется $m$ входящими и $n$ исходящими дугами с произвольными источниками и стоками звезды $Z_{m,n}$. По описанной схеме для заданной звезды в общем случае может быть построено много графов (рис. 1).

Рис. 1. Звезда $Z_{2,2}$ и графы вида $ZK_{2,2,2}$

**Теорема 2.** Относительно минимальных реберных 1-расширений направленных звезд $Z_{m,n}$ справедливо следующее:

1) при $m = n = 1$ звезда $Z_{1,1}$ имеет единственное с точностью до изоморфизма минимальное реберное 1-расширение, которым является циклическая тройка;
2) при $mn > 1$ звезда $Z_{m,n}$ имеет минимальные реберные 1-расширения: $K_{1,m+n}$ и графы, построенные по схемам $ZK_{m-1,n,2}$ и $ZK_{m,n-1,2}$;
3) при $m > 0$, $n = 0$ звезда $Z_{m,0}$ имеет минимальные реберные 1-расширения вида $ZK_{m-1,0,2}$;
4) при $m = 0$, $n > 0$ звезда $Z_{0,n}$ имеет минимальные реберные 1-расширения вида $ZK_{0,n-1,2}$;
5) при $m = 2$, $n = 1$ звезда $Z_{2,1}$ имеет еще одно минимальное реберное 1-расширение — турнир, получающийся из циклической тройки добавлением одной вершины и дуг от нее во все остальные вершины;
6) при $m = 2$, $n = 1$ звезда $Z_{2,1}$ имеет еще одно минимальное реберное 1-расширение — турнир, получающийся из циклической тройки добавлением одной вершины и дуг в нее из всех остальных вершин.

***Доказательство.*** Убедимся, что орграфы $K_{1,m+n}$, $ZK_{m-1,n,2}$ и $ZK_{m,n-1,2}$ действительно являются реберными 1-расширениями звезды $Z_{m,n}$.

Для неориентированной звезды $K_{1,m+n}$ проверка вполне тривиальна. Пусть в звезде $Z_{m,n}$ есть и источники, и стоки, то есть $m > 0$ и $n > 0$. Рассмотрим удаление в графе $K_{1,m+n}$ любой дуги вида $(v, u)$, где $v$ — центральная вершина, а $u$ — любая другая. Вложение звезды $Z_{m,n}$ строится следующим образом: центральная вершина звезды $Z_{m,n}$ соответствует вершине $v$. Вершины $u$ и $m-1$ из оставшихся вершин графа $K_{1,m+n}$ будут соответствовать источникам, а остальные $n$ вершин — стокам. Аналогично — для случая удаления дуги вида $(u, v)$. Заметим, что количество дуг в графе $K_{1,m+n}$ в два раза больше, чем в звезде $Z_{m,n}$, то есть $2(m+n)$.

Пусть $m > 0$; рассмотрим орграф $ZK_{m-1,n,2}$. В этом графе две вершины имеют полустепени исхода и захода $n+1$ и $m$ соответственно, а у остальных вершин суммы полустепеней исхода и захода равны 2. Общее количество дуг равно $2(m+n)$. Обозначим две полные вершины через $v_1$ и $v_2$.

Все дуги рассматриваемого орграфа можно разделить на три группы подобных дуг:

1) дуги от вершин $v_1$ и $v_2$ в остальные вершины (если $n = 0$, то дуг этого типа может не быть);
2) дуги в вершины $v_1$ и $v_2$ из остальных вершин (если $m = 1$, то дуг этого типа может не быть);
3) дуги между вершинами $v_1$ и $v_2$.

Рассмотрим удаление из орграфа $ZK_{m-1,n,2}$ дуги каждого вида и укажем, каким образом в получившийся орграф может быть вложена звезда $Z_{m,n}$. Обозначим:

1) $G_1$ — граф, получающийся из орграфа $ZK_{m-1,n,2}$ удалением дуги первого типа; пусть для определенности это будет дуга $(v_1, u)$, где $u$ — произвольная вершина, которая в звезде $Z_{m,n}$ была стоком;
2) $G_2$ — граф, получающийся из орграфа $ZK_{m-1,n,2}$ удалением дуги второго типа; пусть для определенности это будет дуга $(w, v_1)$, где $w$ — произвольная вершина, которая в звезде $Z_{m,n}$ была истоком;
3) $G_3$ — граф, получающийся из орграфа $ZK_{m-1,n,2}$ удалением дуги третьего типа; пусть для определенности это будет дуга $(v_2, v_1)$.

Во всех трех случаях $m - 1$ вершин орграфов $G_1$, $G_2$, $G_3$, из которых идет дуга в вершину $v_2$, и вершина $v_1$ будут соответствовать источникам звезды $Z_{m,n}$, вершина $v_2$ — центральной вершине, а оставшиеся $n$ вершин орграфов $G_1$, $G_2$, $G_3$ — стокам звезды $Z_{m,n}$.

Аналогично рассмотрим орграф $ZK_{m,n-1,2}$ при $n > 0$. В этом графе две вершины имеют полустепени исхода и захода $n$ и $m + 1$ соответственно, а у остальных вершин суммы полустепеней исхода и захода равны 2. Общее количество дуг снова равно $2(m + n)$. Как и раньше, обозначим две полные вершины через $v_1$ и $v_2$. Все дуги рассматриваемого орграфа делятся на три группы подобных дуг:

1) дуги от вершин $v_1$ и $v_2$ в остальные вершины (если $n = 1$, то дуг этого типа может не быть);
2) дуги в вершины $v_1$ и $v_2$ из остальных вершин (если $m = 0$, то дуг этого типа может не быть);
3) дуги между вершинами $v_1$ и $v_2$.

Рассмотрим удаление из орграфа $ZK_{m,n-1,2}$ дуги каждого вида и укажем, каким образом в получившийся орграф может быть вложена звезда $Z_{m,n}$. Пусть

1) граф $G_1$ получается из орграфа $ZK_{m,n-1,2}$ удалением дуги первого типа; пусть для определенности это будет дуга $(v_1, u)$, где $u$ — произвольная вершина, которая в звезде $Z_{m,n}$ была стоком;
2) граф $G_2$ получается из орграфа $ZK_{m,n-1,2}$ удалением дуги второго типа; пусть для определенности это будет дуга $(w, v_1)$, где $w$ — произвольная вершина, которая в звезде $Z_{m,n}$ была истоком;
3) граф $G_3$ получается из орграфа $ZK_{m,n-1,2}$ удалением дуги третьего типа; пусть для определенности это будет дуга $(v_1, v_2)$.

Во всех трех случаях $m - 1$ вершин орграфов $G_1$, $G_2$, $G_3$, из которых идет дуга в вершину $v_2$, и вершина $v_1$ будут соответствовать источникам звезды $Z_{m,n}$, вершина $v_2$ — центральной вершине, а оставшиеся $n$ вершин орграфов $G_1$, $G_2$, $G_3$ — стокам звезды $Z_{m,n}$.

Исследуем, какие у звезды $Z_{m,n}$ могут быть минимальные реберные 1-расширения. Пусть $G^*$ — некоторое минимальное реберное 1-расширение звезды $Z_{m,n}$. Так как звезда $Z_{m,n}$ вкладывается в орграф $G^*$, то в $G^*$ есть как минимум одна полная вершина. Рассмотрим случаи, когда такая вершина одна, две, три и более.

I с л у ч а й. Предположим, что полная вершина одна, и обозначим ее через $v$. В этом случае вершина $v$ должна быть соединена со всеми остальными вершинами парой встречных дуг. Если бы это было не так и вершина $v$ была бы соединена с некоторой вершиной $u$ только одной дугой, например $(v,u)$, то вложение звезды $Z_{m,n}$ в граф $G^*-(v,u)$ было бы невозможно, так как ни одна вершина не будет соответствовать центральной вершине звезды $Z_{m,n}$. Граф $K_{1,m+n}$ имеет минимальное число дуг среди всех рассматриваемых графов, то есть графов с одной вершиной, соединенной парой встречных дуг со всеми остальными вершинами.

II с л у ч а й. Пусть есть две полные вершины. Обозначим их через $v_1$ и $v_2$. Какие дуги должны быть между этими вершинами? Если между ними будет только одна дуга, например $(v_1,v_2)$, то граф, получающийся из $G^*$ после удаления дуги $(v_1,v_2)$, не будет содержать ни одной вершины, которая могла бы соответствовать центральной вершине звезды $Z_{m,n}$. Следовательно, вершины $v_1$ и $v_2$ соединены парой встречных дуг.

Каковы могут быть полустепени исхода и захода вершин $v_1$ и $v_2$? Очевидно, что они не могут быть меньше, чем у центральной вершины звезды $Z_{m,n}$, то есть $n$ и $m$ соответственно. Общее количество дуг, входящих и исходящих из этих вершин, равно $m+n+1$ (две дуги между вершинами $v_1$ и $v_2$ и по одной дуге в или из оставшихся $m+n-1$ вершин). Таким образом, вершины $v_1$ и $v_2$ могут иметь либо полустепень исхода $m$, а полустепень захода $n+1$, либо наоборот. Можно заметить, что вершины $v_1$ и $v_2$ должны иметь одинаковые полустепени исхода и захода. В самом деле, если бы это было не так и вершины $v_1$ и $v_2$ имели бы разные полустепени, например вершина $v_1-(m,n+1)$, а вершина $v_2-(m+1,n)$, то удаление дуги $(v_1,v_2)$ привело бы к орграфу, в котором вершины $v_1$ и $v_2$ имеют полустепени исхода и захода $(m-1,n+1)$ и $(m+1,n-1)$ соответственно. Ясно, что вложение в получившийся граф звезды $Z_{m,n}$ невозможно. Итак, среди всех графов, имеющих две вершины с одинаковыми полустепенями исхода и захода, соединенные между собой парой встречных дуг, а с остальными вершинами хотя бы одной дугой, подходящими являются только графы семейств $ZK_{m-1,n,2}$ и $ZK_{m,n-1,2}$.

III с л у ч а й. Рассмотрим случай, когда есть три вершины, соединенные со всеми остальными $m+n-2$ вершинами хотя бы одной дугой. Обозначим эти полные вершины через $v_1$, $v_2$ и $v_3$. Подобное соединение возможно при $m+n\geqslant 2$. Между этими вершинами, очевидно, должна быть по крайней мере одна дуга. Общее количество дуг в таком графе будет не меньше чем $3+3(m+n-2)=3(m+n-1)$.

Если $m+n=2$, то число дуг будет меньше, чем в рассмотренных ранее случаях I и II, где количество дуг равно $2(m+n)$. Непосредственной проверкой убедимся, что звезда $Z_{1,1}$, которая является ориентированной цепью, имеет подходящее реберное 1-расширение, которое в данном случае будет циклической тройкой (рис. 2).

Рис. 2. Звезда $Z_{1,1}$ и ее единственное МР-1Р

Звезды $Z_{2,0}$ и $Z_{0,2}$ не имеют реберных 1-расширений, построенных по рассматриваемой схеме. Если же добавить еще одну дугу, то мы получим графы, изоморфные

графам семейств $ZK_{1,0,2}$ и $ZK_{0,1,2}$. При $m+n=3$ имеем $3(m+n-1)=2(m+n)$. То есть число дуг будет таким же, как и в рассмотренных ранее случаях. Подходящими звездами будут $Z_{3,0}$, $Z_{2,1}$, $Z_{1,2}$ и $Z_{0,3}$. Непосредственной проверкой убеждаемся, что звезды $Z_{3,0}$ и $Z_{0,3}$ не имеют реберных 1-расширений, построенных по рассматриваемой схеме, а звезды $Z_{2,1}$ и $Z_{1,2}$ имеют. Все их реберные 1-расширения изображены на рис. 3, причем первыми указаны расширения, укладывающиеся в рассматриваемую схему. Вершины $v_1$, $v_2$ и $v_3$ образуют в них контур. Последующие три расширения принадлежат семействам $ZK_{2,0,2}$ и $ZK_{1,1,2}$ для звезды $Z_{2,1}$; $ZK_{0,2,2}$ и $ZK_{1,1,2}$ для звезды $Z_{1,2}$. Последние расширения — это звезды $K_{1,3}$. Видно, что звезды $Z_{2,1}$ и $Z_{1,2}$ имеют три изоморфных минимальных реберных 1-расширения и по два различных.

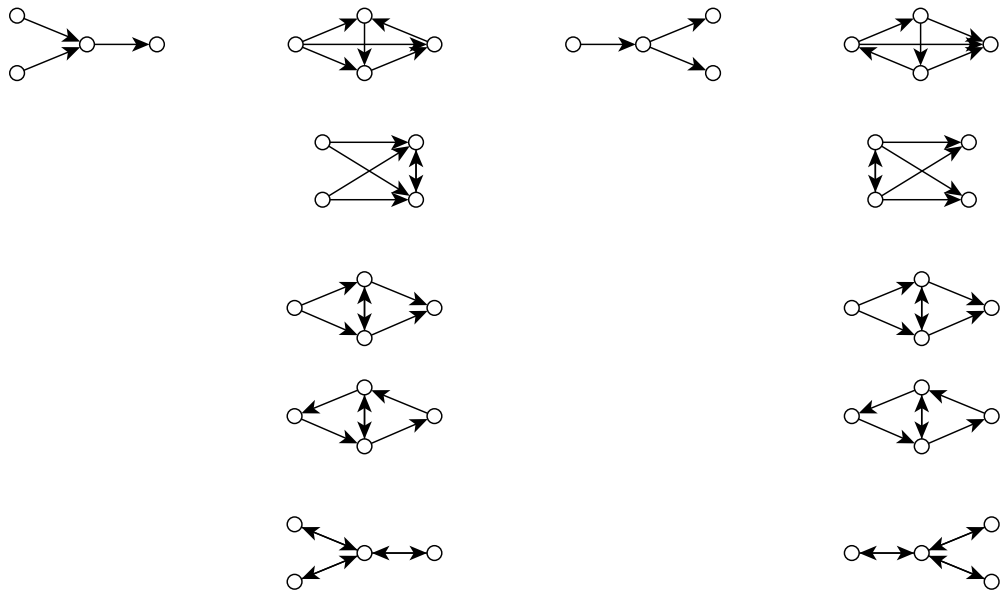

Рис. 3. Звезды $Z_{2,1}$ и $Z_{1,2}$ и все их МР-1Р

IV с л у ч а й. Рассмотрим случай, когда есть $p>3$ вершин, соединенных со всеми остальными $m+n-p+1$ вершинами хотя бы одной дугой. Подобное соединение возможно при $m+n\geqslant p-1$. Между этими вершинами, очевидно, должна быть по крайней мере одна дуга, то есть индуцированный ими подграф является турниром. Общее количество дуг в таком графе будет не менее чем

$$p(m+n-p+1)+p(p-1)/2=p(m+n)-p(p-1)/2.$$

Исследуем, в каких случаях эта величина может быть меньше, чем $2(m+n)$ — количество дополнительных ребер в случаях I и II. Получаем

$$p(m+n)-p(p-1)/2\leqslant 2(m+n),$$
$$(m+n)(p-2)-p(p-1)/2\leqslant 0.$$

Так как $m+n\geqslant p-1$, то $(p-1)(p-2)-p(p-1)/2\leqslant 0$, откуда $(p-1)(p-2-p/2)\leqslant 0$, а так как $p>3$, то остается $p/2-2\leqslant 0$, то есть $p\leqslant 4$.

Случаи, когда $p=1$, 2 и 3, были рассмотрены ранее, и, таким образом, еще только при $p=4$ возможно реберное 1-расширение, которое будет иметь не больше дуг, чем расширения из случаев I и II. Итак, при $p=4$ количество дуг есть $4(m+n)-6$, а у

расширений в случаях I и II — $2(m+n)$, причем $m+n \geqslant 3$. Сравнивая, получаем, что лишь при $m+n=3$ достигается равенство, а при остальных значениях количество дуг в расширении с $p=4$ будет больше, чем в случаях I и II. При $m+n=3$ имеем четыре звезды, которые мы уже рассматривали ранее: $Z_{3,0}$, $Z_{2,1}$, $Z_{1,2}$ и $Z_{0,3}$. Можно заметить, что эта ситуация идентична рассмотренной ранее и не дает новых реберных 1-расширений. ■

**Следствие 1.** Исходящая звезда, то есть направленная звезда вида $Z_{m,0}$, $m>1$, имеет два неизоморфных минимальных реберных 1-расширения — граф $K_{1,m}$ и орграф $ZK_{m-1,0,2}$.

**Следствие 2.** Входящая звезда, то есть направленная звезда вида $Z_{0,n}$, $n>1$, имеет два неизоморфных минимальных реберных 1-расширения — граф $K_{1,m}$ и орграф $ZK_{0,n-1,2}$.

Теорему 2 легко обобщить для ориентированных звезд. Обозначим через $ZK_{m,n,p,t}$ граф, получающийся из звезды $Z_{m,n}$ добавлением $t-1$ «центральной» вершины, соединением их между собой и центральной вершиной звезды $Z_{m,n}$ парами встречных дуг. Каждая из добавленных центральных вершин соединяется $m$ входящими, $n$ исходящими дугами и $p$ ребрами с произвольными источниками и стоками звезды $Z_{m,n}$.

**Теорема 3.** Ориентированные звезды $Z_{m,n,p}$ при $m>0$, $n>0$ и $p>0$ имеют единственное с точностью до изоморфизма минимальное реберное 1-расширение — звезду $K_{1,m+n+p}$.

***Доказательство.*** Пусть $G^*$ — некоторое минимальное реберное 1-расширение звезды $Z_{m,n,p}$. Так как звезда $Z_{m,n,p}$ вкладывается в орграф $G^*$, то в $G^*$ есть как минимум одна полная вершина. Рассмотрим случаи, когда полная вершина одна и более.

I с л у ч а й. Предположим, что полная вершина одна, и обозначим ее через $v$. Повторяя рассуждения из доказательства теоремы 2, получим, что вершина $v$ должна быть соединена со всеми остальными вершинами парой встречных дуг. Если бы это было не так и вершина $v$ была бы соединена с некоторой вершиной $u$ только одной дугой, например $(v,u)$, то вложение звезды $Z_{m,n,p}$ в граф $G^*-(v,u)$ было бы невозможно, так как ни одна вершина не будет соответствовать центральной вершине звезды $Z_{m,n,p}$. Граф $K_{1,m+n+p}$ имеет минимальное число дуг среди всех подходящих под этот случай графов, то есть графов с одной вершиной, соединенной парой встречных дуг со всеми остальными вершинами. Легко видеть, что звезда $K_{1,m+n+p}$ является реберным 1-расширением орзвезды $Z_{m,n,p}$. Рассмотрим удаление в графе $K_{1,m+n+p}$ любой дуги вида $(v,u)$, где $v$ — центральная вершина, а $u$ — любая другая. Вложение звезды $Z_{m,n,p}$ строится следующим образом: центральная вершина звезды $Z_{m,n,p}$ соответствует вершине $v$. Вершина $u$ и $m-1$ из оставшихся вершин графа $K_{1,m+n}$ будут соответствовать источникам, а остальные $n+p$ вершин — стокам и вершинам, которые соединены с центральной вершиной парой встречных дуг. Заметим, что количество дуг в графе $K_{1,m+n+p}$ есть $2(m+n+p)$.

II с л у ч а й. Предположим, что вершин, соединенных дугой с остальными вершинами, всего $t$ штук и $t>1$. Повторяя рассуждения из теоремы 2, получим, что центральные вершины должны быть соединены между собой, что дает не менее $t(t-1)/2$ дуг, каждая из них — с $p$ вершинами парой встречных дуг и с остальными $m+n-t+1$ вершинами по крайней мере одной дугой. Всего получаем дуг

$$t(t-1)/2+2pt+t(m+n-t+1)=t(m+n+2p-(t-1)/2).$$

Эта величина принимает минимальное (по $t > 1$) значение при $t = 2$, и это будет $2(m+n+p)+p-1$. Однако при $t = 2$, как мы установили при исследовании случая II в доказательстве теоремы 2, центральные вершины должны быть соединены парой встречных дуг. Поэтому минимально возможное количество дуг по данной схеме может быть $2(m+n+p)+p$, и при $p > 0$ это больше числа дуг в графе $K_{1,m+n+p}$. ■

## 2. Минимальные реберные $k$-расширения

Обобщим результаты двух предыдущих теорем.

**Теорема 4.** Относительно минимальных реберных $k$-расширений направленных звезд $Z_{m,n}$ справедливо следующее:

1) при $m = n = k$ звезда $Z_{m,n}$ имеет минимальным реберным $k$-расширением любой регулярный $(2k+1)$-вершинный турнир, и только их;
2) при $m = n+1 = k$ звезда $Z_{m+1,m}$ имеет минимальным реберным $k$-расширением любой $(2k+2)$-вершинный турнир, который получается из регулярного $(2k+1)$-вершинного турнира добавлением дополнительной вершины-источника;
3) при $m+1 = n = k$ звезда $Z_{m,m+1}$ имеет минимальным реберным $k$-расширением любой $(2k+2)$-вершинный турнир, который получается из регулярного $(2k+1)$-вершинного турнира добавлением дополнительной вершины-стока;
4) кроме случая $m = n = k$, звезда $Z_{m,n}$ при $k \leqslant m+n$ имеет минимальным реберным $k$-расширением графы вида $ZK_{m_1,n_1,k+1}$, где $m_1 + n_1 = m + n - k$, $\max\{0, m-k\} \leqslant m_1 \leqslant m$, $\max\{0, n-k\} \leqslant n_1 \leqslant m$. При $k > m+n$ звезда $Z_{m,n}$ не имеет минимальных реберных $k$-расширений.

***Доказательство.*** Пусть $k > 1$ и $G^*$ — некоторое минимальное реберное $k$-расширение звезды $Z_{m,n}$. Так как звезда $Z_{m,n}$ вкладывается в орграф $G^*$, то в $G^*$ есть как минимум одна вершина, соединенная дугой со всеми остальными. Обозначим через $p$ количество полных вершин; положим $N = m+n$. Очевидно, что $p \leqslant m+n+1$. Удаление $k$ любых дуг из графа $G^*$ должно оставить по крайней мере одну полную вершину.

I с л у ч а й. Если все полные вершины в графе $G^*$ соединены между собой одной дугой, то удаление такой дуги исключает две полные вершины. Поэтому количество полных вершин должно быть не менее $2k+1$. Тогда количество дуг в таком орграфе будет не менее чем $(2k+1)k + (2k+1)(N-2k) = (2k+1)N - 2k2 - k$, причем $2k \leqslant N$.

II с л у ч а й. Если все полные вершины в графе $G^*$ соединены между собой парой встречных дуг, то удаление такой пары дуг исключает две полные вершины. Поэтому, если $k$ четно, то количество полных вершин должно быть не менее $k+1$. При нечетном $k = 2k_1 + 1$ с удалением $k_1$ пар встречных дуг будут исключены $k-1$ полных вершин. Предположим, что останется еще лишь одна полная вершина. Если она будет соединена с некоторой вершиной только одной дугой, то, удалив ее, мы исключим последнюю полную вершину. Следовательно, эта вершина должна быть соединена с остальными вершинами также парой встречных дуг. Более того, удалив одну такую дугу, мы получим, что из единственной полной вершины либо исходит, либо входит в некоторую вершину единственная дуга. Следовательно, в звезде $Z_{m,n}$ должен быть хотя бы один источник и сток. В силу произвольности выбора исключаемых вершин можно сделать вывод, что при нечетном $k$, при $n > 0$, $m > 0$ и $k-1 \leqslant N$, орграф $G^*$ может иметь $k$ полных вершин, соединенных между собой и со всеми остальными вершинами парой встречных дуг. Определим количество дуг в этом случае:

$$k(k-1) + 2(N-k+1)k = 2Nk - k2 + k,$$

причем $k-1 \leqslant N$.

III с л у ч а й. Если количество полных вершин в орграфе $G^*$ есть $k+1$, то между собой они должны быть соединены парой встречных дуг, но с остальными вершинами они могут быть соединены одной дугой. Мы уже встречались с такими орграфами, например это семейства $ZK_{m_1,n_1,k+1}$. Минимальное количество дуг в этом случае равно $(k+1)k+(k+1)(N-k) = N(k+1)$, причем $k \leqslant N$.

Видно, что при больших значениях $N$ количество дуг в последнем случае будет меньше, чем в случаях I и II. Определим более точные соотношения по количеству дуг между этими случаями. Случаи II и III:

$$2Nk - k^2 + k \leqslant N(k+1),$$
$$N(k-1) - k(k-1) \leqslant 0,$$
$$(N-k)(k-1) \leqslant 0.$$

При $k=1$ имеем равенство количества дуг. Случай II при $k=1$ — это звезда $K_{1,N}$ из теоремы 2. При $k>1$ интерес представляет случай $N \leqslant k$. Но так как в случае II $k \leqslant N$, то остается единственная возможность $k=N$. Легко видеть, что при $k=N$ графы в случаях II и III изоморфны и представляют собой полный граф $K_n$. Таким образом, при $k>1$ случай II не представляет интереса.

Случаи I и III:

$$(2k+1)N - 2k^2 - k \leqslant N(k+1),$$
$$Nk - 2k^2 - k \leqslant 0.$$

Так как в случае I имеем ограничение $2k \leqslant N$, то интерес представляют всего два значения $N$. При $N=2k$ получаем, что граф, построенный по схеме I, будет иметь меньше дуг, чем граф из случая III, а при $N=2k+1$ количество дуг в случаях I и III будет одинаково. Видно, что при этих значениях графы из случая I будут турнирами. Исследуем эти возможности подробнее.

$N=2k$. В этом случае имеем $(2k+1)$-вершинный турнир. Удаляя $k$ произвольных дуг, соединяющих различные вершины турнира, получим граф с единственной полной вершиной, которая соединена с каждой из остальных вершин одной дугой. Значит, чтобы звезда $Z_{m,n}$ вкладывалась в получившийся граф, в эту вершину должны входить $m$ дуг и выходить $n$. В силу произвольности выбора это должно быть справедливо для всех вершин турнира, то есть все вершины должны иметь одинаковые степени исхода и захода. Это возможно лишь при $m=n=k$. Таким образом, только звезда вида $Z_{m,m}$ может иметь регулярный турнир минимальным реберным $k$-расширением при $k=m$. При $k=1$ имеем звезду $Z_{1,1}$, и ее минимальным реберным 1-расширением является циклическая тройка. На рис. 4 показана звезда $Z_{2,2}$ и ее единственное минимальное реберное 2-расширение, построенное по описанной схеме.

$N=2k+1$. В этом случае имеем $(2k+2)$-вершинный турнир. Удаляя $k$ произвольных дуг, соединяющих различные вершины турнира, получим граф с двумя полными вершинами, которые соединены с каждой из остальных вершин одной дугой. Значит, чтобы звезда $Z_{m,n}$ вкладывалась в получившийся граф, в одну из этих вершин должны входить $m$ дуг и выходить $n$. Так как число вершин в турнире четно, то он не может быть регулярным и все вершины не могут иметь одинаковые полустепени исхода и захода. Следовательно, хотя бы одна вершина будет иметь полустепени исхода и захода, отличные от $m$ и $n$. Однако такая вершина может быть только одна, потому что

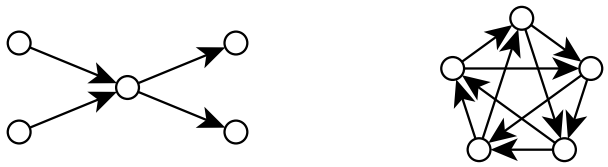

Рис. 4. Звезда $Z_{2,2}$ и ее единственное МР-2Р

в противном случае, подбирая соответствующим образом $k$ удаляемых дуг, мы могли бы получить орграф с двумя полными вершинами, ни одна из которых бы не имела $m$ входящих и $n$ исходящих дуг. Итак, в $(2k+2)$-вершинном турнире $2k+1$ вершин должны иметь одинаковые полустепени исхода и захода — $m$ и $n$ соответственно. Таким образом, имеем $(2k+1)m$ исходящих дуг, $(2k+1)n$ входящих и одну неучтенную вершину, обозначим ее $w$. Так как число входящих и исходящих дуг должно быть равно, а $m \neq n$, то можно сделать вывод, что либо $m = n+1$ и $w$ является стоком, либо $n = m+1$ и $w$ является источником. Таким образом, только звезды вида $Z_{m+1,m}$ и $Z_{m,m+1}$ могут иметь минимальным реберным $k$-расширением при $k = m$ турнир, получающийся из регулярного $(2m+1)$-вершинного турнира добавлением вершины и дуг из нее во все вершины турнира для звезды $Z_{m+1,m}$ и из всех вершин турнира в добавленную для звезды $Z_{m,m+1}$. При $k = 1$ имеем звезды $Z_{2,1}$ и $Z_{1,2}$ и их минимальные реберные 1-расширения, упомянутые в п. 5 и 6 теоремы 2. Этим завершается исследование случая I. Рассмотрим далее наиболее общий случай III.

Кроме случая $N = 2k$, графы, построенные по схеме случая III, являются кандидатами на минимальные реберные $k$-расширения. Убедимся, что эти графы действительно будут являться реберными $k$-расширениями. Пусть $G^*$ — произвольный граф вида $ZK_{m_1,n_1,k+1}$, где $m_1+n_1+k = m+n$. Рассмотрим удаление дуг между различными полными вершинами и остальными вершинами орграфа $G^*$. Удаление одной такой дуги исключает одну полную вершину. Удаление $k$ дуг оставит единственную полную вершину. Эта полная вершина будет соединена с $k$ вершинами парой встречных дуг, с $m_1$ вершинами исходящей дугой, а с $n_1$ вершинами входящей дугой. По предположению звезда $Z_{m,n}$ должна вкладываться в получившийся орграф. Если $m_1 > m$ или $n_1 > n$, то такое вложение будет невозможно. Аналогично, если $m_1 < m-k$ или $n_1 < n-k$. Если же все указанные неравенства выполняются, то недостающими вершинами для соответствия источникам и стокам звезды будут являться бывшие полные вершины орграфа $G^*$. Похожим образом можно рассмотреть удаления дуг между полными вершинами и убедиться, что графы вида $ZK_{m_1,n_1,k+1}$ действительно являются реберными $k$-расширениями звезды $Z_{m,n}$ при выполнении указанных ранее ограничений. ■

На рис. 5 показаны все минимальные реберные 2-расширения звезд $Z_{2,3}$ и $Z_{3,2}$.

**Теорема 5.** Относительно минимальных реберных $k$-расширений ориентированных звезд $Z_{m,n,p}$ справедливо следующее:

1) при четном $k$ звезда $Z_{m,n,p}$ имеет минимальные реберные $k$-расширения вида $ZK_{m_1,n_1,p,k+1}$, и только их;
2) при нечетном $k$, $m > 0$, $n > 0$ и $(m+n-k)(k-1) - 2p \leqslant 0$ звезда $Z_{m,n,p}$ имеет минимальное реберное $k$-расширение вида $K_k + O_{m+n+p-k+1}$;
3) при $k \leqslant m+n$ и $(m+n-k)(k-1) - 2p \geqslant 0$ звезда $Z_{m,n,p}$ имеет минимальные реберные $k$-расширения вида $ZK_{m_1,n_1,p,k+1}$;

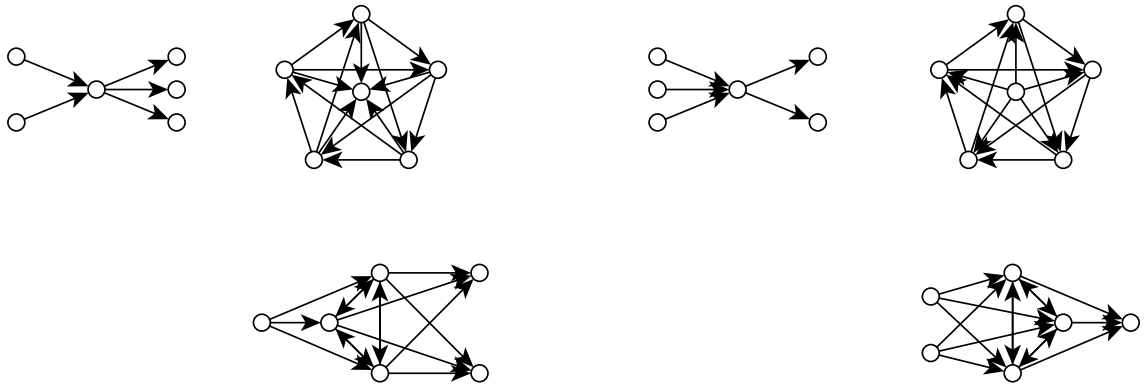

Рис. 5. Звезды $Z_{2,3}$ и $Z_{3,2}$ и их MP-2P: единственное в форме турнира и по одному представителю семейств $ZK_{1,2,3}$ и $ZK_{2,1,3}$

4) при $k > m + n$ звезда $Z_{m,n,p}$ не имеет минимальных реберных $k$-расширений.

***Доказательство.*** Пусть $k > 1$ и $G^*$ — некоторое минимальное реберное $k$-расширение звезды $Z_{m,n,p}$. Так как звезда $Z_{m,n,p}$ вкладывается в орграф $G^*$, то в $G^*$ есть как минимум одна полная вершина. Обозначим через $t$ количество полных вершин; положим $N = m + n$. Очевидно, что $t \leqslant m + n + 1$. Удаление $k$ любых дуг из графа $G^*$ должно оставить по крайней мере одну полную вершину.

I с л у ч а й. Если все полные вершины в графе $G^*$ соединены между собой одной дугой, то удаление одной такой дуги исключает две полные вершины. Поэтому количество полных вершин должно быть не менее $2k+1$. Каждая полная вершина должна быть соединена с $p$ вершинами парой встречных дуг, а с остальными по крайней мере одной дугой. Тогда количество дуг в таком орграфе будет не менее чем

$$(2k + 1)k + (2k + 1)(N - 2k) + 2(2k + 1)p = (2k + 1)N - 2k2 - k + 2(2k + 1)p,$$

причем $2k \leqslant N$.

II с л у ч а й. Если все полные вершины в графе $G^*$ соединены между собой парой встречных дуг, то удаление такой пары дуг исключает две полные вершины. Поэтому если $k$ четно, то количество полных вершин должно быть не менее $k+1$. При нечетном $k = 2k_1 + 1$ удаление $k_1$ пары встречных дуг исключает $k - 1$ полных вершин. Предположим, что останется еще лишь одна полная вершина. Если она будет соединена с некоторой вершиной только одной дугой, то, удалив ее, мы исключим последнюю полную вершину. Следовательно, эта вершина должна быть соединена с остальными вершинами также парой встречных дуг. Более того, удалив одну такую дугу, мы получим, что из единственной полной вершины либо исходит, либо входит в некоторую вершину единственная дуга. Следовательно, в звезде $Z_{m,n}$ должен быть хотя бы один источник и сток. В силу произвольности выбора исключаемых вершин можно сделать вывод, что при нечетном $k$, при $n > 0$, $m > 0$ и $k - 1 \leqslant N$ орграф $G^*$ может иметь $k$ полных вершин, соединенных между собой и со всеми остальными вершинами парой встречных дуг. Такой граф можно представить как $K_k + O_{m+n+p-k+1}$. Определим количество дуг в этом случае: $k(k - 1) + 2(N - k + 1 + p)k = 2Nk - k^2 + k + 2pk$, причем $k - 1 \leqslant N$.

III с л у ч а й. Если количество полных вершин в орграфе $G^*$ есть $k + 1$, то между собой они должны быть соединены парой встречных дуг, с $p$ вершинами также

парой встречных дуг, а с остальными вершинами они могут быть соединены одной дугой. Мы уже встречались с такими орграфами, например это семейства $ZK_{m_1,n_1,p,k+1}$. Определим минимальное количество дуг в этом случае:

$(k+1)k + (k+1)(N-k) + 2(k+1)p = N(k+1) + 2(k+1)p$, причем $k \leqslant N$.

Видно, что при больших значениях $N$ количество дуг в последнем случае будет минимальным. Определим более точные соотношения по количеству дуг между этими случаями. Случаи II и III:

$$2Nk - k^2 + k + 2pk \leqslant N(k+1) + 2(k+1)p,$$
$$N(k-1) - k(k-1) - 2p \leqslant 0,$$
$$(N-k)(k-1) - 2p \leqslant 0.$$

При $k = 1$ в теореме 3 мы получили, что графы из случая II всегда имеют меньшее число дуг, чем графы из случая III. При $k > 1$ ситуация представляется более сложной: при значениях $N$, много больших, чем $k$ и $p$, меньше дуг будет в схеме III, а при нечетных значениях $k$, близких к $N$ или больших, чем $p$, меньше дуг будет в схеме II.

Случаи I и III:

$$(2k+1)N - 2k2 - k + 2(2k+1)p \leqslant N(k+1) + 2(k+1)p,$$
$$Nk - 2k2 - k + 2kp \leqslant 0.$$

Так как в случае I имеем ограничение $2k \leqslant N$, то при $p > 0$ у графов из случая I будет всегда больше дуг, чем у графов из случая III. ■

Рассмотрим для примера поиск минимальных реберных 3-расширений звезды $Z_{2,2,1}$. Посчитаем величину $(m+n-k)(k-1) - 2p$ и получим

$$(2+2-3)(3-1) - 2*1 = 0.$$

Таким образом, по теореме 5 получаем, что звезда $Z_{2,2,1}$ будет иметь минимальные реберные 3-расширения двух видов: граф $K_3 + O_3$ и графы вида $ZK_{1,0,1,4}$ и $ZK_{0,1,1,4}$. Все эти графы содержат по 24 дуги. На рис. 6 показаны все минимальные реберные 3-расширения звезды $Z_{2,2,1}$. Для облегчения визуализации расширений пара встречных дуг заменена неориентированным ребром.

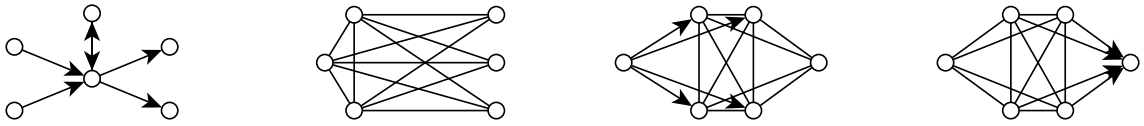

Рис. 6. Звезда $Z_{2,2,1}$ и все ее МР-3Р

## Заключение

В данной работе дано аналитическое решение задачи описания минимальных реберных $k$-расширений для направленных и ориентированных звезд. Ранее удалось найти полное решение задачи описания минимальных вершинных $k$-расширений для направленных звезд, а также минимальных вершинных и реберных $k$-расширений для неориентированных звезд. Полученные результаты расширяют класс графов, для которых удалось найти полное решение рассматриваемой задачи.

ЛИТЕРАТУРА

1. *Богомолов А. М., Салий В. Н.* Алгебраические основы теории дискретных систем. М.: Наука, 1997.
2. *Hayes J. P.* A graph model for fault-tolerant computing system // IEEE Trans. Comput. 1976. V. C25. No. 9. P. 875–884.
3. *Harary F. and Hayes J. P.* Edge fault tolerance in graphs // Networks. 1993. V. 23. P. 135–142.
4. *Harary F. and Hayes J. P.* Node fault tolerance in graphs // Networks. 1996. V. 27. P. 19–23.
5. *Киреева А. В.* Отказоустойчивость в функциональных графах // Упорядоченные множества и решетки. Саратов, 1995. Вып. 11. С. 32–38.
6. *Sung T. Y., Lin C. Y., Chuang Y. C., and Hsu L. H.* Fault tolerant token ring em-bedding in double loop networks // Inform. Process. Lett. 1998. V. 66. P. 201–207.
7. *Абросимов М. Б.* О сложности некоторых задач, связанных с расширениями графов // Матем. заметки. 2010. № 5(88). С. 643–650.
8. *Абросимов М. Б.* Минимальные $k$-расширения предполных графов // Изв. вузов. Математика. 2003. № 6(493). С. 3–11.
9. *Абросимов М. Б.* Минимальные расширения неориентированных звезд // Теоретические проблемы информатики и ее приложений. Саратов: СГУ, 2006. Вып. 7. С. 3–5.
10. *Абросимов М. Б.* Минимальные расширения транзитивных турниров // Вестник Томского госуниверситета. Приложение. 2006. № 17. С. 187–190.

2011 Прикладная теория графов №2(12)

DOI 10.17223/20710410/12/7

УДК 512.5

# АТТРАКТОРЫ ДИНАМИЧЕСКИХ СИСТЕМ, АССОЦИИРОВАННЫХ С ЦИКЛАМИ

А. В. Власова

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

**E-mail:** VAnastasiyaV@gmail.com

Доказывается критерий принадлежности состояний аттракторам и описывается формирование аттракторов динамических систем, ассоциированных с циклами, состояниями которых являются двоичные векторы заданной размерности, а эволюционная функция преобразует вектор с помощью одновременного выполнения следующих действий: начальный 0 и последняя 1 заменяются на 1 и 0 соответственно, каждая диграмма 10 — на 01.

**Ключевые слова:** *аттрактор, динамическая система, эволюционная функция.*

## Введение

В задачах, связанных с отказоустойчивостью компьютерных сетей, заметное место занимают графовые модели, в которых отказы процессоров интерпретируются как удаление соответствующих вершин, а отказы сетевых каналов — как удаление дуг. Здесь можно выделить следующую конструкцию, получившую самостоятельное значение в теории графов — бесконтурный граф с заданной структурой источников и стоков [1]. В модели [1] в качестве механизма восстановления работоспособности сети предлагается так называемая SER-динамика бесконтурных графов. Это позволяет использовать при изучении модельных графов идеи и методы теории конечных динамических систем и, в частности, динамических систем двоичных векторов (см., например, [2, 3]), когда имеется естественная двоичная кодировка графов рассматриваемого класса.

Под *конечной динамической системой* понимается пара $(S,\ \delta)$, где $S$ — конечное непустое множество, элементы которого называются *состояниями системы*, $\delta : S \to S$ — отображение множества состояний в себя, называемое *эволюционной функцией системы*.

Каждой конечной динамической системе сопоставляется *карта* — ориентированный граф с множеством вершин $S$ и дугами, проведенными из каждой вершины $s \in S$ в вершину $\delta(s)$. Этот граф является функциональным, т. е. из каждой вершины выходит точно одна дуга. Компоненты связности графа, задающего динамическую систему, называются её *бассейнами*.

Каждый бассейн представляет собой контур с входящими в него деревьями. Контуры называются предельными циклами или *аттракторами*.

Основными проблемами теории конечных динамических систем являются задачи отыскания эволюционных параметров состояний системы без проведения динамики, таких, как индекс (расстояние до аттрактора того бассейна, которому принадлежит состояние), период (длина соответствующего аттрактора), ветвление (количество непосредственных предшественников) [4] и другие. Программа [5] позволяет исследовать некоторые из эволюционных параметров определенных систем. Одной из нерешенных

в общем случае задач является вопрос о том, принадлежит ли данное состояние аттрактору, что обозначено, например, в [1, 3].

В данной работе представлены критерий принадлежности состояний аттракторам и описание формирования аттракторов динамических систем двоичных векторов, порожденных такими графами, как циклы.

## 1. Описание динамической системы

Пусть имеется $n$-звенный цикл $c$. Выберем в нём какую-либо вершину в качестве начальной и обозначим её $c_0$, тогда цикл можно записать как $c = c_0c_1..c_{n-1}c_n$, где $c_n = c_0$. Придадим каждому ребру цикла произвольную ориентацию. Дуги, имеющие направление по часовой стрелке, пометим символом 1, а дуги с противоположной ориентацией — символом 0.

Таким образом, каждой ориентации цикла сопоставляется $n$-мерный двоичный вектор. С другой стороны, каждый такой вектор однозначно определяет некоторую ориентацию цикла, так что между множеством $C^n$, $n > 2$, всевозможных ориентаций $n$-звенного цикла и множеством $B^n$ всех двоичных векторов размерности $n$ устанавливается взаимно однозначное соответствие.

Динамическая система $(C^n, \theta)$ вводится следующим образом: если дан некоторый граф $c$ из $C^n$, то его динамическим образом $\theta(c)$ является граф, получаемый из $c$ одновременным превращением всех стоков в источники (SER-динамика бесконтурных графов [1]).

Исходная динамическая система $(C^n, \theta)$ оказывается изоморфной динамической системе $(B^n, \theta)$, которая вводится следующим образом: пусть состоянием динамической системы $(B^n, \theta)$ в данный момент времени является вектор $v \in B^n$, тогда в следующий момент времени она окажется в состоянии $\theta(v)$, описываемом правилами: 1) если первой компонентой в $v$ является 0 и последней компонентой является 1, то первой компонентой в $\theta(v)$ будет 1, а последней компонентой — 0; 2) если в составе $v$ имеются диграммы вида 10, то в $\theta(v)$ каждая из них заменяется на 01; 3) других отличий между $v$ и $\theta(v)$ нет. Вышеперечисленные правила применяются одновременно. Будем полагать по определению, что контуры представляют собой аттракторы единичной размерности. На рис. 1 показана карта динамической системы $(B^6, \theta)$.

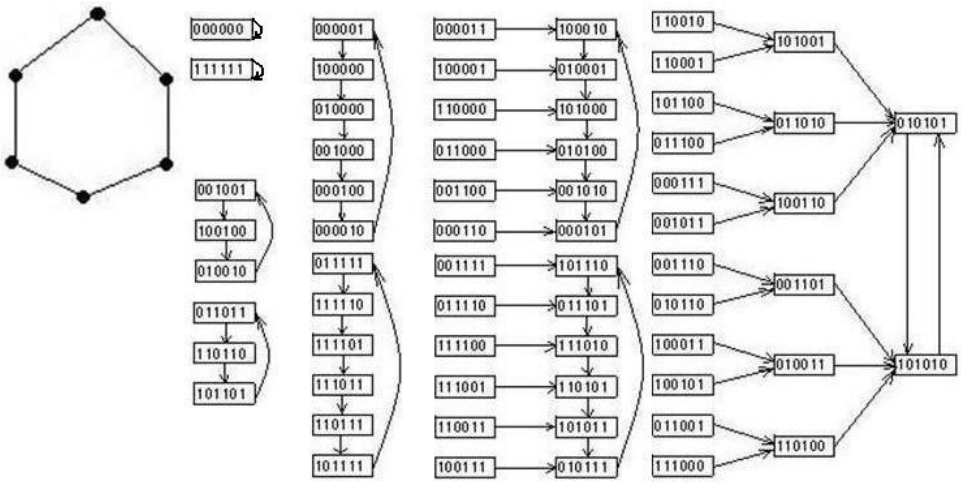


Рис. 1. Карта динамической системы $(B^6, \theta)$

## 2. Аттракторы динамической системы $(B^n, \theta)$

Через $p_c(v)$ обозначим *циклическую плотность вектора* $v$, т. е. количество пар совпадающих соседних компонент в нем с учётом циклического сдвига. Например, $p_c(111111) = 6$, $p_c(101010) = 0$, $p_c(111011) = 4$, $p_c(\theta(111011)) = p_c(110111) = 4$. Очевидно, что для состояния $v$ системы $(B^n, \theta)$ верно $0 \leqslant p_c(v) \leqslant n$. *Циклический блок* — это максимальное по включению множество подряд стоящих нулей (0-блок) или единиц (1-блок) в количестве $\geqslant 2$ с учетом циклического сдвига. При работе с динамической системой, ассоциированной с циклом, будем под понятием *блок* подразумевать понятие *циклический блок*. *Длина блока* — число нулей (единиц) в блоке, уменьшенное на 1. Обозначим через $p_c^0$, $p_c^1$ суммы длин с учетом циклического сдвига рассматриваемых 0-блоков и 1-блоков соответственно.

**Теорема 1** (критерий принадлежности состояния аттрактору). Вектор динамической системы $(B^n, \theta)$ тогда и только тогда принадлежит аттрактору, когда у него $p_c^0 = 0$ или $p_c^1 = 0$. При этом периоды равны делителям числа $n$, и если $p_c^0 = 0$, то аттрактор представляет собой цикл, в котором следующее состояние получается из предыдущего циклическом сдвигом влево на одну компоненту, а при $p_c^1 = 0$ — вправо.

***Доказательство.*** Рассмотрим состояния системы $(B^n, \theta)$ в зависимости от наличия и количества 0- и 1-блоков.

I. *Вектор не содержит ни 0-, ни 1-блоков, т. е.* $p_c^0 = p_c^1 = 0$. Это состояние вида $(01)^{\frac{n}{2}}$, которое при динамике переходит в состояние $(10)^{\frac{n}{2}}$, или вида $(10)^{\frac{n}{2}}$, которое при динамике переходит в состояние $(01)^{\frac{n}{2}}$, где $n$ — четное (при нечетном $n$ состояние содержит хотя бы один блок, образуемый первой и последней компонентами). Таким образом, эти состояния образуют аттрактор длины 2.

II. *Вектор содержит блоки, и все они состоят из 1.* Рассмотрим эволюцию таких состояний в зависимости от количества 1-блоков.

1) Вектор имеет единственный 1-блок. Не теряя общности, рассмотрим эволюцию такого состояния при расположении 1-блока в начале вектора. В табл. 1 приведена его схематичная эволюция.

Таким образом, эти состояния образуют аттрактор длины $n$, причём в нем каждое следующее состояние получается из предыдущего циклическим сдвигом влево на одну компоненту. Можно заметить, что длина 1-блока на каждом очередном шаге остаётся прежней.

2) Вектор имеет несколько блоков, и все они состоят из 1. Из рассмотрения п. II(1) можно заключить, что каждый 1-блок на очередном шаге смещается влево на одну компоненту с учётом циклического сдвига. Таким образом, такие состояния образуют аттракторы, причём в каждом аттракторе следующее состояние получается из предыдущего циклическим сдвигом влево на одну компоненту, и их периоды равны делителям числа $n$, отличным от 1 и 2.

III. *Вектор содержит блоки, и все они состоят из 0.* Рассмотрим эволюцию таких состояний в зависимости от количества 0-блоков.

1) Вектор имеет единственный 0-блок. Не теряя общности, рассмотрим эволюцию такого состояния при расположении 0-блока в начале вектора. В табл. 2 приведена его схематичная эволюция.

Таким образом, эти состояния образуют аттрактор длины $n$, причём в нем каждое следующее состояние получается из предыдущего циклическим сдвигом вправо на одну компоненту. Можно заметить, что длина 0-блока на каждом шаге остаётся прежней.

Таблица 1

**Эволюция вектора, содержащего единственный 1-блок**

| № шага | Состояние |
|---|---|
| 0 | $1^h01\ldots010$ |
| 1 | $1^{h-1}010\ldots101$ |
| 2 | $1^{h-2}0101\ldots011$ |
| ... | ... |
| $h-2$ | $110\ldots10101^{h-2}$ |
| $h-1$ | $101\ldots010101^{h-1}$ |
| $h$ | $010\ldots10101^h$ |
| $h+1$ | $101\ldots0101^h0$ |
| $h+2$ | $010\ldots101^h01$ |
| ... | ... |
| $n-1$ | $01^h01\ldots01$ |
| $n$ | $1^h010\ldots10$ |

Таблица 2

**Эволюция вектора, содержащего единственный 0-блок**

| № шага | Состояние |
|---|---|
| 0 | $0^h101\ldots0101$ |
| 1 | $10^h10\ldots1010$ |
| 2 | $010^h1\ldots0101$ |
| ... | ... |
| $n-h$ | $1010\ldots1010^h$ |
| $n-h+1$ | $0101\ldots010101^{h-1}$ |
| $n-h+2$ | $0010\ldots101010^{h-2}$ |
| ... | ... |
| $n-1$ | $0^{h-1}1010\ldots1010$ |
| $n$ | $0^h101\ldots0101$ |

2) Вектор имеет несколько блоков, и все они состоят из 0. Из рассмотрения п. III(1) можно заключить, что каждый 0-блок на очередном шаге смещается вправо на одну компоненту с учётом циклического сдвига. Таким образом, эти состояния образуют аттракторы, причём в каждом аттракторе следующее состояние получается из предыдущего циклическим сдвигом вправо, и их периоды равны делителям числа $n$, отличным от 1 и 2.

IV. *Вектор состоит из 0-блоков (1-блоков), после которых идут 1-блоки (0-блоки).* Так как блоки в состоянии рассматриваются с учётом циклического сдвига, то не имеет значения, идут ли сначала 0-блоки или 1-блоки. Не теряя общности, будем считать, что сначала идут 0-блоки, потом 1-блоки.

1. Рассмотрим эволюцию данных состояний, включающих в себя по одному 0-блоку и 1-блоку, в зависимости от их значений $p_c^0$ и $p_c^1$.

Исходя из рассуждений предыдущих пунктов, можно заключить, что в данном случае на каждом шаге эволюции одновременно 0-блок начнет движение вправо, 1-блок — влево с учётом циклического сдвига, пока не окажутся стоящими рядом. Рассмотрим эволюцию с этого шага в зависимости от сумм длин 0- и 1-блоков.

а) Состояния, для которых $p_c^0 < p_c^1$.

В табл. 3 приведена схематичная эволюция таких состояний.

Таблица 3

**Эволюция вектора, содержащего единственные 0-блок и 1-блок, с $p_c^0 < p_c^1$**

| № шага | Состояние |
|---|---|
| 0 | $01\ldots01010^{h_0}1^{h_1}0101\ldots01$ |
| 1 | $10\ldots101010^{h_0-1}1^{h_1-1}01010\ldots10$ |
| 2 | $01\ldots0101010^{h_0-2}1^{h_1-2}010101\ldots10$ |
| ... | ... |
| $h_0-2$ | $01\ldots010101001^{h_1-h_0+2}010101\ldots10$ |
| $h_0-1$ | $10\ldots101010101^{h_1-h_0+1}0101010\ldots01$ |

Таким образом, когда блоки оказываются стоящими рядом, то их длины начинают уменьшаться у каждого на единицу за счёт поглощения компонент друг друга, пока 0-блок полностью не поглотится. В результате остаётся 1-блок и чередующиеся

0 и 1; дальнейшая эволюция такого состояния описана в п. II(1) и показано, что оно принадлежит аттрактору.

б) Состояния, для которых $p_c^0 = p_c^1$.

Покажем, что такая ситуация возможна только при четном $n$, независимо от количества и размерности 0- и 1-блоков. Докажем это индукцией по $p_c^0$.

Базис индукции: $p_c^0 = 1$. Здесь длины 0- и 1-блоков равны 1 и в сумме дают чётное число; количество компонент между этими блоками с учетом циклического сдвига также чётно, так как это чередующиеся 0 и 1, а после 0-блока обязательно стоит 1 и перед 1-блоком стоит 0. В итоге получаем чётное количество компонент вектора.

Шаг индукции. Предположим, что при любом $p_c^0 \leqslant m \in \mathbb{N}$ количество компонент вектора чётно, и покажем, что это условие выполняется при $p_c^0 = m + 1$.

Рассмотрим несколько ситуаций.

1) Длины первого 0-блока и последнего 1-блока не равны 1. Тогда отсекаем у этих блоков по одному элементу, получаем случай $p_c^0 = m$, для которого требуемое условие выполняется, а отсекли мы две компоненты, т. е. в итоге получаем чётное число и выполнение нужного условия.

2) Длина первого 0-блока или длина последнего 1-блока равна 1. Например, длина первого 0-блока равна 1. Тогда отсекаем у этих блоков по одной компоненте, получаем случай $p_c^0 = m$, для которого требуемое условие выполняется. От 0-блока осталась одна компонента, до следующего 0-блока идёт нечетное количество компонент, а отсекли мы две компоненты, т. е. в итоге получаем чётное число и выполнение нужного условия.

3) Длины первого 0-блока и последнего 1-блока равны 1. Тогда отсекаем у этих блоков по одному элементу, получаем случай $p_c^0 = m$, для которого требуемое условие выполняется. От 0-блока и 1-блока осталось по одной компоненте, до следующего 0-блока идёт нечетное количество компонент, от предыдущего 1-блока идёт нечетное количество компонент, а отсекли мы две компоненты, т. е. в итоге получаем чётное число и выполнение нужного условия.

Заключаем, что действительно ситуация $p_c^0 = p_c^1$ возможна только для векторов четной размерности.

Из рассуждений п. IV(1,$a$) получаем, что блоки будут поглощать друг друга с каждым следующим шагом эволюции, пока от них не останется по одной компоненте, причём это уже будет вектор, полностью состоящий из чередующихся 0 и 1, а про его эволюцию было сказано в п. I, и это состояние принадлежит аттрактору.

в) Состояния, для которых $p_c^0 > p_c^1$.

Здесь ситуация аналогична п. IV(1,$a$), только в конце концов остаётся один 0-блок и чередующиеся 0 и 1, дальнейшая эволюция описана в п. III(1).

2. Рассмотрим состояния, включающие в себя несколько 0-блоков, после которых идут 1-блоки, также в зависимости от значений $p_c^0$ и $p_c^1$ в векторе.

а) Состояния, для которых $p_c^0 < p_c^1$.

Из п. IV(1,$a$) получаем, что здесь на каждом шаге эволюции 0-блоки начинают двигаться вправо, а 1-блоки — влево за счёт поглощения компонент, стоящих между блоками, с учетом циклического сдвига; когда они оказываются стоящими рядом, то блоки начинают уменьшаться на единицу каждый за счёт поглощения компонент друг друга, пока один из блоков полностью не поглотится, после чего опять продолжаются сдвиги блоков навстречу друг другу, пока не поглотится самый последний 0-блок. Таким образом, в состоянии останутся только 1-блоки и чередующиеся 0 и 1, дальнейшая эволюция описана в п. II, и это состояние принадлежит аттрактору.

б) С о с т о я н и я, д л я к о т о р ы х $p_c^0 = p_c^1$.

В п. IV(1,*б*) было показано, что такая ситуация возможна только для четного $n$. Из рассуждений пп. IV(1,*б*) и IV(2,*а*) получаем, что при эволюции 0-блоки и 1-блоки начнут движение навстречу друг другу, пока не окажутся рядом, а затем начнут поглощать друг друга с каждым следующим шагом эволюции, пока не поглотится один из блоков, после чего продолжится аналогичное движение, пока рядом не окажутся последние оставшиеся 0-блок и 1-блок, которые начнут поглощать друг друга с каждым следующим шагом эволюции, пока от них не останутся по одной компоненте, причём это уже будет вектор, полностью состоящий из чередующихся 0 и 1, эволюция которого была описана в п. I, и он принадлежит аттрактору.

в) С о с т о я н и я, д л я к о т о р ы х $p_c^0 > p_c^1$.

Здесь ситуация аналогична п. IV(2,*а*), только в состоянии в конце концов остаётся один 0-блок и чередующиеся 0 и 1, дальнейшая эволюция состояния описана в п. II, и оно принадлежит аттрактору.

V. *Вектор состоит из 0-блоков и 1-блоков в произвольном порядке.*

Из доказательств предыдущих пунктов можно заключить, что при эволюции такого состояния 0-блоки будут циклически сдвигаться вправо, при этом если они будут встречаться с 1-блоками, то длина этого 0-блока будет уменьшаться с очередным шагом эволюции, т. е. если есть подряд стоящие 0-блоки, то они или все поглотятся, если следующие за ними подряд стоящие 1-блоки в сумме имеют равный или больший порядок, или поглотят сами следующие за ними подряд стоящие 1-блоки и продолжат сдвиг вправо, встречая очередные 1-блоки, если порядок 0-блоков будет больше порядка 1-блоков. С точки рассмотрения 1-блоков ситуация получается аналогичная. В итоге получим состояние, имеющее только 0-блоки, только 1-блоки или содержащее только чередующиеся нули и единицы, чьи эволюции описаны в пп. I–III, где показано, что они и только они (с учетом дальнейшего доказательства) принадлежат аттракторам.

Для полноты заметим, что контуры представляют собой аттракторы единичной длины; периоды равны делителям числа $n$. ■

## Заключение

Для динамических систем, ассоциированных с циклами, решена задача выяснения принадлежности состояния аттрактору; приведено описание аттракторов этих систем.

## ЛИТЕРАТУРА

1. *Barbosa V. C.* An atlas of edge-reversal dynamics. London: Chapman & Hall/CRC, 2001. 372 p.
2. *Салий В. Н.* Об одном классе конечных динамических систем // Вестник Томского госуниверситета. Приложение. 2005. № 14. С. 23–26.
3. *Colon-Reyes O., Laubenbacher R., and Pareigis B.* Boolean monomial dynamical systems // Ann. Combinat. 2004. V. 8. P. 425–439.
4. *Власова А. В.* Ветвления в конечной динамической системе $(B^n, \theta)$ // Научные исследования студентов Саратовского государственного университета: материалы итог. студ. науч. конф. Саратов: Изд-во Сарат. ун-та, 2008. С. 57–58.
5. *Власова А. В.* Исследование эволюционных параметров в динамических системах двоичных векторов // Свидетельство РОСПАТЕНТа №2009614409, зарегистрировано 20 августа 2009.

2011 Прикладная теория графов №2(12)

DOI 10.17223/20710410/12/8
УДК 512.2

# О КОНГРУЭНЦИЯХ ЦЕПЕЙ

Е. О. Карманова

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

**E-mail:** lkb@info.sgu.ru

Под конгруэнцией цепи понимается отношение эквивалентности на множестве ее вершин, все классы которого являются независимыми подмножествами. Показано, что любой связный граф является фактор-графом подходящей цепи. Найдены границы для минимальной длины цепи, факторизующейся на данный граф.

**Ключевые слова:** *цепь, конгруэнция, фактор-граф, дерево, звезда, обход.*

Под ориентированным графом (далее орграфом) понимается пара $G = (V, \alpha)$, где $V$ — конечное непустое множество вершин, а $\alpha$ — отношение на $V$, задающее множество дуг. Основные понятия приводятся в соответствии с [1].

Графовые модели широко используются во многих областях человеческой деятельности. Транспортные системы, информационные сети, компьютерные программы, отношение зависимости в социальных группах — все могут моделироваться графами. Существуют различные методы преобразования графовых систем для приложений к проблемам оптимизации в вышеупомянутых ситуациях. В качестве допустимых реконструкций данного графа обычно рассматриваются следующие [2]:

1) ориентация ребер данного неориентированного графа (например, известная теорема Оре — критерий ориентируемости графа в сильно связный орграф [3]);
2) добавление новых дуг (ребер) (эта реконструкция используется, например, для построения отказоустойчивых реализаций компьютерных сетей по Хейзу — Абросимову [4, 5]);
3) удаление некоторых дуг (ребер) (здесь общеизвестными результатами являются, например, алгоритмы построения минимального остовного дерева для связной сети, так называемые минимальные расконтуривания сетей в технической диагностике [6]);
4) отождествление некоторых вершин графа.

Последний вид реконструкций формализуется следующим образом.

Пусть $\varepsilon$ — некоторое отношение эквивалентности на множестве вершин $V$ орграфа $G$. Фактор-графом орграфа $G$ по эквивалентности $\varepsilon$ называется орграф $G/\varepsilon = (V/\varepsilon, \alpha_\varepsilon)$, где $V/\varepsilon$ — множество классов эквивалентности $\varepsilon$; $\alpha_\varepsilon = \{(\varepsilon(v_1), \varepsilon(v_2)) : \exists u_1 \in \varepsilon(v_1), u_2 \in \varepsilon(v_2)\ ((u_1, u_2) \in \alpha)\}$.

Пусть $K$ — некоторый класс орграфов. Конгруэнцией $K$-графа $G$ называется такое отношение эквивалентности $\theta$ на $V$, что фактор-граф $G/\theta$ является $K$-графом.

Известны результаты М. Р. Мирзаянова [7], который рассматривал случай, когда $K$ — класс сильносвязных орграфов, и предложил способ построения сильносвязной конгруэнции произвольного орграфа, наибольшей по числу вершин в фактор-графе. Им установлено также [8], что $n$-элементная ориентированная цепь имеет $2^{n-3}$ минимальных сильносвязных конгруэнций.

Возьмём в качестве класса $K$ класс неориентированных графов.

Путем в графе $G = (V, \alpha)$ называется последовательность ребер, в которой каждые два соседних ребра имеют общую вершину и никакое ребро не встречается более одного раза. Если начальная и конечная вершины пути совпадают, путь называется циклическим. Путь, каждая вершина которого принадлежит не более чем двум его ребрам, является простым. Простой циклический путь называется циклом, нециклический — цепью. Цепь с $m$ ребрами будем обозначать $P_m$.

Звезда — это граф, все ребра которого инцидентны одной и той же вершине. Звезду с $m$ ребрами будем обозначать $S_m$.

Показано, что любой связный граф является фактор-графом подходящей цепи, и получены оценки для минимальной длины цепи, факторизующейся на данный граф.

Множество вершин называется независимым, если любые две вершины из этого множества несмежны.

Очевидно, что отношение эквивалентности $\theta$ на множестве вершин графа $G$ тогда и только тогда будет конгруэнцией этого графа, когда каждый $\theta$-класс образует в $G$ независимое подмножество.

Известна следующая задача о факторизации: можно ли для заданного графа сказать, является ли он фактор-графом другого заданного графа? Эта задача является NP-полной.

Например, возьмем пятиреберную цепь $P_5$. Пятивершинный цикл $C_5$ будет её фактор-графом, а четырехреберная звезда $S_4$ — нет.

К числу нерешенных проблем комбинаторной теории графов относится следующая задача: сколько фактор-графов имеет $n$-элементная цепь? Сколько среди них неизоморфных?

В [9] доказана следующая теорема.

**Теорема 1.** $i(P_{n-1}) = F(n+2)$, где $i(G)$ — число независимых множеств в графе $G$, $F(n)$ — числа Фибоначчи.

Отсюда следует, что для цепи $P_{n-1}$ количество всех конгруэнций с одним нетривиальным классом равно $F(n+2) - 2$, так как в число всех независимых множеств в графе включается пустое множество и все одноэлементные подмножества множества его вершин.

В [10] представлена программа, генерирующая все конгруэнции заданной цепи и выделяющая среди них цепные конгруэнции, т. е. такие, фактор-графы по которым являются цепями. При этом выдается общее число конгруэнций данной цепи и количество её цепных конгруэнций. Например, у трехреберной цепи $P_3$ имеется четыре различных конгруэнции, которые дают три неизоморфных фактор-графа. У четырехреберной цепи $P_4$ всего имеется 14 фактор-графов, из них 7 попарно неизоморфных.

На основании результатов, полученных в [10], можно выдвинуть следующее предположение.

ГИПОТЕЗА. Количество конгруэнций $n$-элементной цепи равно $B(n-1)$, где $B(n)$ — число Белла.

Число Белла $B(n)$ — это число всевозможных эквивалентностей на $n$-элементном множестве.

Маршрут в связном графе называется обходом, если он содержит все ребра графа.

Следующий известный результат Тарри [3] показывает, что любой связный граф с $m$ ребрами имеет обход длины $2m$.

**Лемма 1.** Если граф связный, то можно построить циклический маршрут, содержащий все ребра графа в точности два раза, по одному в каждом направлении.

Пусть дана $m$-реберная цепь. Какие графы являются ее фактор-графами?

**Теорема 2.** Связный граф тогда и только тогда является фактор-графом $m$-реберной цепи, когда в нем есть обход длины $m$.

***Доказательство.*** Необходимость. Пусть связный граф $G$ является фактор-графом цепи $P_m$ по конгруэнции $\theta$. Покажем, что в нем есть обход длины $m$.

Пусть в цепи $P_m$ вершины пронумерованы натуральными числами $1, 2, \ldots, m, m+1$.

Каждая вершина графа $G$ является $\theta$-классом. При этом $\theta(i)$ и $\theta(j)$ смежны в $G$ тогда и только тогда, когда существуют $i' \in \theta(i)$ и $j' \in \theta(j)$, такие, что $|i' - j'| = 1$.

Рассмотрим в $G$ маршрут $M = \theta(1), \theta(2), \ldots, \theta(m+1)$. Покажем, что это обход, т. е. любое ребро графа $G$ входит в состав маршрута. В самом деле, пусть $\{\theta(k), \theta(l)\}$ — ребро в $G$. Так как $\theta(k)$ и $\theta(l)$ смежны в $G$, то найдутся $k' \in \theta(k)$ и $l' \in \theta(l)$, такие, что $k'$ и $l'$ смежны в цепи $P_m$, т. е. $|k' - l'| = 1$. Пусть $l' = k' + 1$. Тогда в $M$ встретится фрагмент $\ldots, \theta(k'-1), \theta(k'), \theta(l'), \ldots$, и значит, ребро $\{\theta(k), \theta(l)\}$ входит в состав $M$. Таким образом, $M$ — обход, и его длина равна $m$.

Достаточность. Пусть $G$ — произвольный связный граф и в нем есть обход длины $m$. Построим цепь, фактор-графом которой является граф $G$.

Так как в графе $G$ есть обход длины $m$, то каждая вершина при нумерации вершин вдоль обхода получит одну или более меток из натуральных чисел $1, 2, \ldots, m, m+1$. Наибольшая метка равна $m+1$. В цепи $P_m$, вершины которой пронумерованы натуральными числами $1, 2, \ldots, m, m+1$, рассмотрим отношение

$$(i, j) \in \theta \;\Leftrightarrow i \text{ и } j \text{ — метки одной и той же вершины графа } G.$$

Очевидно, что отношение $\theta$ — эквивалентность на множестве вершин цепи.

Пусть $(i, j) \in \theta$ и $i < j$, т. е. при прохождении обхода некоторая вершина графа $G$ встретилась на шаге $i$, а затем на шаге $j$. Понятно, что $j \geqslant i + 2$. Следовательно, вершины $i$ и $j$ не являются смежными в цепи $P_m$. Это означает, что все $\theta$-классы являются независимыми подмножествами, и значит, $\theta$ — конгруэнция цепи $P_m$.

Покажем, что граф $G$ изоморфен фактор-графу $P_m/\theta$.

Каждой вершине графа $G$ сопоставим множество всех её меток. Тем самым устанавливается взаимно однозначное соответствие между вершинами графа $G$ и $\theta$-классами. Покажем, что это соответствие сохраняет отношение смежности.

Пусть вершины $u$ и $v$ смежны в графе $G$. Это означает, что при обходе они были пройдены последовательно, например вершина $v$ после $u$. Тогда существуют такие метки $i$ у вершины $u$ и $j$ у вершины $v$, что $j = i + 1$. Отсюда следует, что классы $\theta(i)$ и $\theta(j)$ смежны в фактор-графе $P_m/\theta$.

С другой стороны, если классы $\theta(i)$ и $\theta(j)$ смежны в фактор-графе $P_m/\theta$, то в $P_m$ существуют вершины $i' \in \theta(i)$ и $j' \in \theta(j)$, такие, что $|i' - j'| = 1$. Это означает, что в графе $G$ вершины, соответствующие классам $\theta(i)$ и $\theta(j)$, являются смежными.

Таким образом, граф $G$ изоморфен фактор-графу $P_m/\theta$. ■

**Следствие 1.** Любой связный граф с $m$ ребрами является фактор-графом цепи $P_{2m-1}$.

Одной из открытых проблем является следующая: для данного связного графа $G$ найти цепь с минимальным числом ребер $p(G)$, фактор-графом которой является данный граф.

Напомним, что диаметром дерева называется максимальное расстояние между его вершинами.

**Теорема 3.** Если $T$ — дерево с $m$ ребрами, имеющее диаметр $d$, то $p(T) = 2m - d$.

***Доказательство.*** Согласно лемме, любой граф с $m$ ребрами имеет обход длины $2m$. Пусть $R$ — минимальный обход дерева $T$, и пусть его длина $r < 2m$. Значит, в $R$ есть ребро, проходимое один раз. Пусть таких ребер $k$ штук. Пронумеруем их в порядке обхода $R$. Покажем, что ребра, проходимые один раз, образуют в $T$ цепь.

Предположим, что это не так. Пусть ребра $1 = \{u_0, u\}$ и $2$ с началом $v$ не инцидентны. Рассмотрим в составе $R$ маршрут $R(u, v)$. Он содержит цепь $P(u, v)$. Пусть ее первым ребром будет $\{u, u'\}$.

Ребро $\{u, u'\}$ в обходе $R$ проходится больше одного раза.

Представим обход $R$ в виде $R = R_{in}u_0uR^1_uuu'R^1_{u'}u'uR^2_uuu'R^2_{u'}\dots uu'R_{fin}$, где $R_{in}$ — подмаршрут, соединяющий начальную вершину обхода $R$ с вершиной $u_0$; $R_{fin}$ — подмаршрут, соединяющий вершину $u'$ с конечной вершиной обхода $R$; $R^i_u$ — подмаршруты обхода $R$ с началом и концом в $u$; $R^i_{u'}$ — с началом и концом в $u'$.

Заметим, что второй раз ребро $\{u, u'\}$ не может быть пройдено от $u$ к $u'$, так как иначе после его прохождения нужно попасть из $u'$ в $u$ по некоторому подмаршруту $R(u', u)$, а тогда в составе маршрута $uu'R(u', u)$ появляется цикл, что невозможно для дерева. Таким образом, в составе $R$ ребро $\{u, u'\}$ второй раз проходится от $u'$ к $u$.

Теперь построим маршрут $R_{in}u_0uR^1_uR^2_u\dots R^s_uuu'R^1_{u'}R^2_{u'}\dots R^s_{u'}R_{fin}$. Этот маршрут является обходом, так как он содержит все ребра, пройденные в составе $R$, а значит, все ребра дерева $T$. Его длина меньше, чем у $R$, что невозможно, ибо $R$ минимален. Следовательно, предположение о том, что ребра, проходимые один раз, не образуют в $T$ цепь, неверно, и в составе обхода $R$ есть цепь $P$, состоящая из ребер, проходимых один раз.

Длина цепи $P$ равна $k$. Но $k \leqslant d$, так как $d$ — наибольшая длина цепи в дереве $T$.

Пусть $s_{m-k}$ — длина части обхода $R$, проходимая по ребрам кратности $\geqslant 2$ в $R$. Тогда $2m > r = s_{m-k} + k \geqslant 2(m-k) + k = 2m - k \geqslant 2m - d$. Итак, каждый обход дерева $T$ имеет длину не меньше чем $2m-d$. Следовательно, $2m-d$ — это минимальная возможная длина обхода дерева $T$. Покажем, что обход длины $2m - d$ существует.

Изобразим дерево $T$ следующим образом. Пусть $P_d$ — цепь длины $d$ в дереве $T$, $v_i \in P_d$, $i = \overline{0, d-1}$.

Выберем висячую вершину $v_0 \in P_d$ в качестве корневой и припишем уровень $i$ каждой вершине $v_i$ цепи $P_d$. Эти вершины могут быть смежны с некоторыми вершинами, не входящими в цепь $P_d$, они, в свою очередь, с другими вершинами и т. д. Таким образом, каждая вершина $v_i \in P_d$, $i = \overline{1, d-2}$, является корнем некоторого дерева $T_i$ (среди этих деревьев могут быть пустые).

Построим обход $v_0v_1T_1v_1v_2T_2\dots v_{d-3}v_{d-2}T_{d-2}v_{d-2}v_{d-1}$. Согласно лемме 1, каждое дерево $T_i$ имеет обход длины, равной удвоенному числу его ребер, причем начинается и заканчивается он в вершине $v_i \in P_d$ графа $T$.

Отсюда и из теоремы 2 следует доказываемое утверждение, поскольку длина построенного обхода равна $2m - d$, и этот обход минимален. ■

**Следствие 2.** Для звезды $S_m$ имеем $p(S_m) = 2m - 2$.

Докажем теорему о верхней и нижней оценках $p(G)$ для произвольного связного графа $G$ с $m$ ребрами.

**Теорема 4.** Для связного графа $G$ с $m$ ребрами $m \leqslant p(G) \leqslant 2m - 2$.

***Доказательство.*** Первое неравенство следует из того, что количество ребер в фактор-графе не может превышать количества ребер в исходном графе.

Покажем, что в каждом связном графе $G$ с $m$ ребрами существует обход длины $2m-2$. Рассмотрим два случая.

1) Граф $G$ имеет висячие вершины.

Возьмем произвольную висячую вершину $u$, пройдем исходящее из неё ребро. Граф $G^* = G \setminus \{u\}$ связный и имеет $m^* = m-1$ ребер. По следствию 1, существует обход графа $G^*$ длины $2m^*-1 = 2m-3$ и, следовательно, обход графа $G$ длины $2m-2$.

2) Граф не имеет висячих вершин.

Отметим произвольную вершину $u$, не являющуюся точкой сочленения. Возьмем некоторое исходящее из неё ребро, пусть другим его концом будет вершина $v$.

Граф $G^* = G \setminus \{u\}$ связный и имеет $m^* = m-d$ ребер, где $d$ — степень вершины $u$. По лемме 1 существует обход графа $G^*$ из вершины $v$ длины $2(m-d)$. Вершина $u$ с непройденными ребрами образует $d$-реберную звезду. Обойдем эту звезду, отправляясь из вершины $v$. Согласно следствию 2, этот обход имеет длину $2d-2$. Таким образом, получаем обход графа $G$ длины $2(m-d)+2d-2 = 2m-2$. Заметим, что если в $m$-реберном графе есть эйлеров путь, то этот граф является фактор-графом цепи длины $m$ (наименьшая возможная длина для цепи, факторизуемой на $m$-реберный граф). С другой стороны, звезда $S_m$, имеющая $m$ ребер, не может быть получена как фактор-граф цепи с длиной меньше чем $2m-2$. Следовательно, обе оценки являются точными. ■

## ЛИТЕРАТУРА


1. *Богомолов А. М., Салий В. Н.* Алгебраические основы теории дискретных систем. М.: Наука, Физматлит, 1997. 367 с.
2. *Салий В. Н.* Оптимальные реконструкции графов // Современные проблемы дифференциальной геометрии и общей алгебры. Саратов: Изд-во Сарат. ун-та, 2008. С. 59–65.
3. *Оре О.* Теория графов. 2-е изд. М.: Наука, 1980. 336 с.
4. *Hayes J. P.* A graph model for fault-tolerant computing systems // IEEE Trans. Comput. 1976. No. 9. P. 25.
5. *Абросимов М. Б.* Некоторые вопросы о минимальных расширениях графов // Изв. Сарат. ун-та. Сер. Математика. Механика. Информатика. 2005. Т. 5. Вып. 1. С. 86–91.
6. *Верзаков Г. Ф., Киншт Н. В., Рабинович В. И., Тимонен Л. С.* Введение в техническую диагностику. М.: Энергия, 1968.
7. *Мирзаянов М. Р.* Сильно связные конгруэнции ориентированных графов // Теоретические проблемы информатики и ее приложений. Саратов: Изд-во Сарат. ун-та, 2006. Вып. 7. С. 104–114.
8. *Мирзаянов М. Р.* О минимальных сильно связных конгруэнциях ориентированных цепей // Изв. Сарат. ун-та. Сер. Математика. Механика. Информатика. 2006. Т. 6. Вып. 1/2. С. 91–95.
9. *Prodinger H. and Tichy R. F.* Fibonacci numbers of graphs // Fibon. Quart. 1982. V. 20. No. 1. P. 16–21.
10. *Карманова Е. О.* О конгруэнциях цепей и циклов // Компьютерные науки и информационные технологии. Саратов: Изд-во Сарат. ун-та, 2009. С. 238.

2011 Прикладная теория графов №2(12)
DOI 10.17223/20710410/12/9
УДК 519.6

# ОЦЕНКИ ЭКСПОНЕНТОВ ПРИМИТИВНЫХ ГРАФОВ

В. М. Фомичев

*Институт проблем информатики РАН, г. Москва, Россия*

**E-mail:** fomichev@nm.ru

Уточнены оценки экспонентов для $n$-вершинных примитивных орграфов (неотрицательных матриц порядка $n$), содержащих два простых контура, длины которых взаимно просты. Получены достижимые оценки порядка $\mathrm{O}(\max\{l\lambda, f(l,\lambda,n)\})$, где $l$ и $\lambda$ — взаимно простые длины простых контуров в орграфе и $f(l,\lambda,n)$ — линейный полином. Описан полностью класс примитивных орграфов, на которых достигается абсолютная оценка экспонента $n^2-2n+2$ (H. Wielandt, 1950). Для экспонентов неориентированных $n$-вершинных примитивных графов доказаны уточняющие оценки. В частности, если $l$ — длина длиннейшего простого цикла нечетной длины в графе $\Gamma$, то экспонент графа $\Gamma$ не превышает $2n-l-1$. Описан полностью класс примитивных неориентированных графов, на которых достигается абсолютная оценка экспонента $2n-2$.

**Ключевые слова:** *примитивные графы, экспонент графа.*

## Введение

Рассмотрим неотрицательную (положительную) матрицу $A=(a_{i,j})$ порядка $n>1$ над полем действительных чисел, т. е. $a_{i,j}\geqslant 0$ ($a_{i,j}>0$) для всех $i,j\in\{1,\ldots,n\}$, свойство неотрицательности (положительности) матрицы $A$ обозначим так: $A\geqslant 0$ ($A>0$). Неотрицательную матрицу $A$ называют *примитивной*, если $A^t>0$ при некотором натуральном $t$, а наименьшее натуральное $\gamma$, при котором $A^\gamma>0$, называется *экспонентом*, или *показателем примитивности матрицы* $A$, и обозначается $\exp A$. Если такого $t$ не существует, то $\exp A=\infty$.

Важной задачей, в частности для криптографических приложений, является определение экспонентов матриц из различных классов. При исследовании экспонентов матриц часто используется эпиморфизм $\varphi$ мультипликативного моноида неотрицательных матриц порядка $n$ на моноид $n$-вершинных орграфов, где умножение орграфов[1] определено как умножение бинарных отношений [1, с. 212]. При эпиморфизме $\varphi$ матрице $A$ соответствует орграф $\Gamma$ с множеством вершин $\{1,\ldots,n\}$ и с множеством дуг $U$, где $(i,j)\in U\Leftrightarrow a_{i,j}>0$, при этом матрица $M$ смежности вершин графа $\Gamma$ называется *носителем матрицы* $A$. Очевидно, $\varphi(M)=\Gamma$. Для ограничения эпиморфизма $\varphi$ на подмоноид симметрических матриц (для них $a_{i,j}=a_{j,i}$ при всех допустимых $i,j$) областью значений является подмоноид $n$-вершинных графов. Для эпиморфизма $\varphi$ выполнено: $A>0\Leftrightarrow$ орграф $\Gamma=\varphi(A)$ полный. Отсюда неотрицательная матрица $A$ и орграф $\Gamma=\varphi(A)$ одновременно примитивны или не примитивны, в случае примитивности экспоненты их равны.

В данной работе используем преимущественно аппарат теории графов. Необходимым условием примитивности орграфа является его сильная связность. Критерий примитивности орграфа $\Gamma$ определяется длинами его простых контуров [2, с. 226] (контур простой, если проходит через любую вершину не более одного раза). Если $C_1,\ldots,C_k$

[1] Неориентированные графы будем называть просто графами.

суть все простые контуры орграфа $\Gamma$ длин соответственно $l_1, \ldots, l_k$, то сильносвязный орграф $\Gamma$ примитивный, если и только если НОД$(l_1, \ldots, l_k) = 1$. Отсюда $\exp\Gamma = \exp\Gamma'$, если примитивные орграфы (графы) $\Gamma$ и $\Gamma'$ изоморфны.

Достижимая абсолютная оценка экспонента любого примитивного $n$-вершинного орграфа $\Gamma$ получена Виландтом [3], $n > 1$:

$$\exp\Gamma \leqslant n^2 - 2n + 2.$$

Эта оценка для $n$-вершинного примитивного орграфа $\Gamma$ допускает уточнение [2, с. 227] с использованием длины $l$ кратчайшего простого контура в $\Gamma$:

$$\exp\Gamma \leqslant (n-2)l + n. \tag{1}$$

В частности, если орграф $\Gamma$ имеет петлю, то он примитивен и $\exp\Gamma \leqslant 2n - 2$.

Необходимым условием примитивности графа является его связность. Так как любое ребро графа можно рассматривать как цикл длины 2, то из критерия примитивности орграфа следует, что примитивность связного графа $\Gamma$ равносильна наличию в $\Gamma$ простого цикла нечетной длины. Следовательно, в соответствии с теоремой Кенига о двудольных графах связный граф примитивен, если и только если он не является двудольным. В [2, с. 409] приведена достижимая абсолютная оценка экспонентов примитивных $n$-вершинных графов:

$$\exp\Gamma \leqslant 2n - 2.$$

В данной работе уточнены оценки экспонентов для примитивных $n$-вершинных орграфов, имеющих два простых контура взаимно простых длин, $n > 3$, и оценки экспонентов графов, $n > 2$. Описаны все $n$-вершинные орграфы и графы, на которых достигаются абсолютные оценки $\exp\Gamma$.

## 1. Свойства экспонентов графов

Далее через $C$ обозначим контур в орграфе и через $C^*$ — мультимножество вершин контура $C$, в случае простого контура — множество вершин.

Множество $W(\Gamma)$ всех путей орграфа $\Gamma$ образует частичный моноид относительно операции конкатенации (обозначим ее $\bullet$), операция определена на паре путей $(u, v)$, если и только если конечная вершина пути $u$ совпадает с начальной вершиной пути $v$. Результат конкатенации пути $u$ с начальной вершиной $i$ и конечной вершиной $a$ и пути $v$ с начальной вершиной $a$ и конечной вершиной $j$ есть путь $w = u \bullet v$ с начальной вершиной $i$ и конечной вершиной $j$. При этом $\mathrm{len}(w) = \mathrm{len}(u) + \mathrm{len}(v)$, где $\mathrm{len}(w)$ — длина пути $w \in W(\Gamma)$, измеряемая числом дуг (или ребер), составляющих путь $w$. Единицей частичного моноида $W(\Gamma)$ является пустой путь $w_\varnothing$, где $\mathrm{len}(w_\varnothing) = 0$.

При обходе контура $C$ выделим его вершину $a$ как начальную, контур $C$ в этом случае обозначим $C(a)$. Для целого неотрицательного $q$ через $q{\cdot}C(a)$ обозначим контур, составленный из $q$-кратно пройденного контура $C(a)$, где $0 \cdot C(a) = w_\varnothing$.

Обозначим при $i \neq j$, где $i, j \in \{1, \ldots, n\}$, через $w(i, j)$ путь из $i$ в $j$; $[i, j]$ — кратчайший путь из $i$ в $j$; $[i, i]$ — кратчайший контур, проходящий через вершину $i$. Для $H, P \subseteq \{1, \ldots, n\}$ обозначим через $\mathrm{dist}(H, P)$ расстояние в графе $\Gamma$ между множествами $H$ и $P$:

$$\mathrm{dist}(H, P) = \begin{cases} \min\limits_{(i,j)\in H\times P} \mathrm{len}[i, j], & H \cap P = \varnothing, \\ 0, & H \cap P \neq \varnothing. \end{cases}$$

Множество $W$ путей из $i$ в $j$ (при $i = j$ — контуров), где $i, j \in \{1, \ldots, n\}$, назовем $(t, l)$-множеством путей, $t, l$ — натуральные, если в $W$ имеется $l$ путей (контуров), длины которых равны $t, t+1, \ldots, t+l-1$.

Для получения оценки экспонентов примитивных графов отметим ряд свойств. Из критерия примитивности орграфа следует достаточное условие.

**Утверждение 1.** Сильносвязный орграф примитивен, если содержит два простых контура, длины которых взаимно просты.

Пусть $M$ — матрица смежности вершин графа $\Gamma$ и $M^l = (m_{i,j}^{(l)})$, $l \geqslant 1$.

**Утверждение 2** [4, следствие 1 теоремы 2, с. 114]. Число путей длины $l$ из $i$ в $j$ в $n$-вершинном графе $\Gamma$ равно элементу $m_{i,j}^{(l)}$ матрицы $M^l$, где $l \geqslant 1$.

**Утверждение 3.**

а) Если $M^l > 0$ при натуральном $l$, то $M^t > 0$ при любом $t > l$.

б) Если в $n$-вершинном сильносвязном орграфе $\Gamma$ (связном графе $\Gamma'$), где $n > 1$, имеются пути из $i$ в $j$ длины $l > 0$ для любых $i, j \in \{1, \ldots, n\}$, то орграф $\Gamma$ (граф $\Gamma'$) примитивен и $\exp \Gamma \leqslant l$.

в) Если для некоторых $i, j \in \{1, \ldots, n\}$ и для натурального $\tau$ в примитивном орграфе $\Gamma$ (в графе $\Gamma'$) не имеется путей из $i$ в $j$ длины $\tau$, то $\exp \Gamma > \tau$.

***Доказательство.***

а) Если $M^l > 0$, то $m_{i,j}^{(l)} > 0$ для всех $i, j \in \{1, \ldots, n\}$. Вместе с тем в матрице $M$ каждая строка и каждый столбец ненулевые, иначе $M^l > 0$ не выполнено, отсюда $m_{s(j),j}^{(1)} > 0$ при любом $j = 1, \ldots, n$ и некотором $s(j) \in \{1, \ldots, n\}$. Тогда из равенства $M^{l+1} = M^l M$ следует, что $m_{i,j}^{(l+1)} = \sum\limits_{s=1}^{n} m_{i,s}^{(l)} m_{s,j}^{(1)} \geqslant m_{i,s(j)}^{(l)} m_{s(j),j}^{(1)} > 0$ для всех $i, j \in \{1, \ldots, n\}$. Значит, $M^{l+1} > 0$.

б) и в) следуют из утверждений 2 и 3,*а*. ■

**Утверждение 4.** Пусть в $n$-вершинном орграфе $\Gamma$ при $n > 1$ имеется контур $C$ длины $l$ и $(t, l)$-множество путей из $i$ в $j$, каждый из которых проходит через некоторую вершину контура $C$, тогда в $\Gamma$ имеются пути из $i$ в $j$ длины $\tau$ при любом $\tau \geqslant t$, где $i, j \in \{1, \ldots, n\}$.

***Доказательство.*** Пусть путь $w^{(s)}$ из $(t, l)$-множества путей графа $\Gamma$ есть путь из $i$ в $j$ длины $t + s$, проходящий через вершину $a$ контура $C$, $s = 0, 1, \ldots, l-1$. Тогда при $a \notin \{i, j\}$ путь $w^{(s)}$ может быть представлен конкатенацией двух путей: $w^{(s)} = w(i, a) \bullet w(a, j)$, где $\mathrm{len}(w(i, a)) + \mathrm{len}(w(a, j)) = \mathrm{len}(w^{(s)}) = t + s$. Построим пути $u_q^{(s)}$ из $i$ в $j$ длины $t + s + ql$, где $q = 0, 1, \ldots$:

$u_q^{(s)} = (q \cdot C(a)) \bullet w^{(s)}$, если $i = a$;

$u_q^{(s)} = w^{(s)} \bullet (q \cdot C(a))$, если $j = a$, $i \neq j$;

$u_q^{(s)} = w(i, a) \bullet (q \cdot C(a)) \bullet w(a, j)$ в остальных случаях.

По построению $\mathrm{len}(u_q^{(s)}) = ql + t + s$, где $q = 0, 1, \ldots$, $s = 0, 1, \ldots, l-1$. Значит, в семействе путей $\{u_q^{(s)}\}$ имеется путь из $i$ в $j$ длины $\tau$ при любом $\tau \geqslant t$. ■

## 2. Оценки экспонентов орграфов

Пусть в орграфе $\Gamma$ имеются простые контуры $C$ и $C'$ длины соответственно $l$ и $\lambda$, где, не теряя общности, положим $1 < \lambda < l \leqslant n$. Обозначим $\theta = (l-1)(\lambda-1)$.

Для вершин $i$ и $j$ контура (цикла) $C$ обозначим через $\rho(i, j)$ длину кратчайшего пути от $i$ до $j$, составленного только из дуг контура (ребер цикла) $C$, и положим

$\rho(i,i) = 0$. При $i \neq j$ имеем $\rho(i,j) \leqslant l-1$ для орграфа и $\rho(i,j) \leqslant \lfloor l/2 \rfloor$ для графа. Для вершины $i$ и подмножества $H$ вершин контура (цикла) $C$ обозначим через $\rho(i,H)$ длину кратчайшего пути от $i$ до ближайшей вершины множества $H$, составленного только из дуг контура (ребер цикла) $C$. Если $|H| = h$, то $\rho(i,H) \leqslant l-h$ для контура $C$ орграфа. Аналогичные расстояния на контуре $C'$ обозначим $\rho'$.

**Теорема 1.** Пусть $(l,\lambda) = 1$, $n > 2$, тогда:

а) если $C^* \cap C'^* = \varnothing$, то $\exp\Gamma \leqslant l\lambda - 2l - 3\lambda + 3n$;

б) если $C^* \cap C'^* = H$, где $|H| = h > 0$, то $\exp\Gamma \leqslant l\lambda - l - 3\lambda + h + 2n$.

***Доказательство.*** В условиях теоремы граф $\Gamma$ примитивен. Для получения оценки $\exp\Gamma$ построим $(t,\lambda)$-множество путей из $i$ в $j$ для любых $i,j \in \{1,\dots,n\}$, каждый из которых проходит через некоторую вершину контура $C'$.

Множество чисел $\{ql \bmod \lambda : q = 0,1,\dots,\lambda-1\}$ в силу взаимной простоты чисел $l$ и $\lambda$ образует полную систему неотрицательных вычетов по модулю $\lambda$, при этом числа $ql$ не превосходят $(\lambda-1)l$. Значит, для любого $q = 0,1,\dots,\lambda-1$ найдется неотрицательное целое число $z(q)$, такое, что $\theta \leqslant ql + z(q)\lambda \leqslant (\lambda-1)l$, при этом

$$\{ql + z(q)\lambda : q = 0,1,\dots,\lambda-1\} = N(\theta,\lambda), \tag{2}$$

где $N(\theta,\lambda) = \{\theta, \theta+1, \dots, \theta+\lambda-1\}$; равенство (2) известно как лемма Шора.

а) Пусть $C^* \cap C'^* = \varnothing$, $a,b \in C^*$, $a',b' \in C'^*$, где $\mathrm{len}[a,a'] = \mathrm{dist}[C^*,C'^*]$, $\mathrm{len}[b',b] = \mathrm{dist}[C'^*,C^*]$ (рис. 1). Определим пути $u_q(i,j)$ и $v_q(i,j)$ из $i$ в $j$, проходящие через вершины контура $C'$, $q = 0,1,\dots,\lambda-1$:

$$u_q(i,j) = q\cdot C(a) \bullet [a,a'] \bullet z(q)\cdot C'(a'), \text{ если } i = a, j = a';$$
$$u_q(i,j) = q\cdot C(a) \bullet [a,a'] \bullet z(q)\cdot C'(a') \bullet [a',j], \text{ если } i = a, j \neq a';$$
$$u_q(i,j) = [i,a] \bullet q\cdot C(a) \bullet [a,a'] \bullet z(q)\cdot C'(a'), \text{ если } i \neq a, j = a';$$
$$u_q(i,j) = [i,a] \bullet q\cdot C(a) \bullet [a,a'] \bullet z(q)\cdot C'(a') \bullet [a',j], \text{ если } i \neq a, j \neq a';$$
$$v_q(i,j) = z(q)\cdot C'(b') \bullet [b',b] \bullet q\cdot C(b), \text{ если } i = b', j = b;$$
$$v_q(i,j) = z(q)\cdot C'(b') \bullet [b',b] \bullet q\cdot C(b) \bullet [b,j], \text{ если } i = b', j \neq b;$$
$$v_q(i,j) = [i,b'] \bullet z(q)\cdot C'(b') \bullet [b',b] \bullet q\cdot C(b), \text{ если } i \neq b', j = b;$$
$$v_q(i,j) = [i,b'] \bullet z(q)\cdot C'(b') \bullet [b',b] \bullet q\cdot C(b) \bullet [b,j], \text{ если } i \neq b', j \neq b.$$

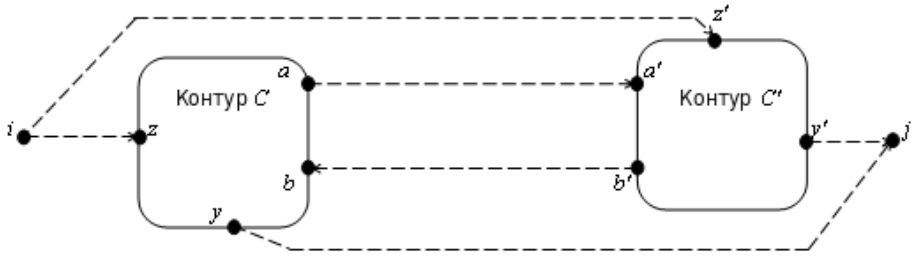


Рис. 1. Непересекающиеся контуры $C$ и $C'$ в орграфе

Длины путей $u_q(i,j)$ и $v_q(i,j)$ соответственно равны:

$$\operatorname{len}(u_q(i,j)) = ql + z(q)\lambda + \operatorname{len}[a,a'],\ \text{если } i=a, j=a'; \tag{3}$$
$$\operatorname{len}(u_q(i,j)) = ql + z(q)\lambda + \operatorname{len}[a,a'] + \operatorname{len}[a',j],\ \text{если } i=a, j\neq a'; \tag{4}$$
$$\operatorname{len}(u_q(i,j)) = ql + z(q)\lambda + \operatorname{len}[i,a] + \operatorname{len}[a,a'],\ \text{если } i\neq a, j=a'; \tag{5}$$
$$\operatorname{len}(u_q(i,j)) = ql + z(q)\lambda + \operatorname{len}[i,a] + \operatorname{len}[a,a'] + \operatorname{len}[a',j],\ \text{если } i\neq a, j\neq a'; \tag{6}$$
$$\operatorname{len}(v_q(i,j)) = ql + z(q)\lambda + \operatorname{len}[b',b],\ \text{если } i=b', j=b; \tag{7}$$
$$\operatorname{len}(v_q(i,j)) = ql + z(q)\lambda + \operatorname{len}[b',b] + \operatorname{len}[b,j],\ \text{если } i=b', j\neq b; \tag{8}$$
$$\operatorname{len}(v_q(i,j)) = ql + z(q)\lambda + \operatorname{len}[i,b'] + \operatorname{len}[b',b],\ \text{если } i\neq b', j=b; \tag{9}$$
$$\operatorname{len}(v_q(i,j)) = ql + z(q)\lambda + \operatorname{len}[i,b'] + \operatorname{len}[b',b] + \operatorname{len}[b,j],\ \text{если } i\neq b', j\neq b. \tag{10}$$

Из формул (3)–(6) следует, что в соответствии с леммой Шора для любых $i,j \in \in \{1,\dots,n\}$ множество путей $\{u_q(i,j) : q = 0,1,\dots,\lambda-1\}$ есть $(\theta+t_0(i,j),\lambda)$-множество путей из $i$ в $j$, где

$$t_0(i,j) = \operatorname{len}[a,a'],\ \text{если } i=a, j=a'; \tag{11}$$
$$t_0(i,j) = \operatorname{len}[a,a'] + \operatorname{len}[a',j],\ \text{если } i=a, j\neq a'; \tag{12}$$
$$t_0(i,j) = \operatorname{len}[i,a] + \operatorname{len}[a,a'],\ \text{если } i\neq a, j=a'; \tag{13}$$
$$t_0(i,j) = \operatorname{len}[i,a] + \operatorname{len}[a,a'] + \operatorname{len}[a',j],\ \text{если } i\neq a, j\neq a'. \tag{14}$$

Аналогично из формул (7)–(10) следует, что в соответствии с леммой Шора для любых $i,j \in \{1,\dots,n\}$ множество $\{v_q(i,j) : q = 0,1,\dots,\lambda-1\}$ есть $(\theta + t_1(i,j),\lambda)$-множество путей из $i$ в $j$, где

$$t_1(i,j) = \operatorname{len}[b',b],\ \text{если } i=b', j=b; \tag{15}$$
$$t_1(i,j) = \operatorname{len}[b',b] + \operatorname{len}[b,j],\ \text{если } i=b', j\neq b; \tag{16}$$
$$t_1(i,j) = \operatorname{len}[i,b'] + \operatorname{len}[b',b],\ \text{если } i\neq b', j=b; \tag{17}$$
$$t_1(i,j) = \operatorname{len}[i,b'] + \operatorname{len}[b',b] + \operatorname{len}[b,j],\ \text{если } i\neq b', j\neq b. \tag{18}$$

Следовательно, в соответствии с утверждениями 3,*б* и 4

$$\exp\Gamma \leqslant \theta + \max_{(i,j)}\{\min\{t_0(i,j), t_1(i,j)\}\}. \tag{19}$$

Оценим величины $t_0(i,j)$ и $t_1(i,j)$. Пусть $w_i$ — кратчайший путь от $i$ до ближайшей вершины контура $C$ ($z$ — конечная вершина) при $i \notin C^*$, не проходящий через вершины контура $C'$; $u_i$ — кратчайший путь от $i$ до ближайшей вершины контура $C'$ ($z'$ — конечная вершина) при $i \notin C'^*$, не проходящий через вершины контура $C$. В $W(\Gamma)$ содержится хотя бы один из путей $w_i$ и $u_i$ при любом $i \notin C^* \cup C'^*$. Тогда верны следующие оценки:

$$\operatorname{len}[i,a] \leqslant \operatorname{len}(w_i) + \rho(z,a),\ \text{если } w_i \in W(\Gamma) \text{ и } i \notin C^*; \tag{20}$$
$$\operatorname{len}[i,a] \leqslant \rho(i,a),\ \text{если } i \in C^*; \tag{21}$$
$$\operatorname{len}[i,b'] \leqslant \operatorname{len}(u_i) + \rho'(z',b'),\ \text{если } u_i \in W(\Gamma) \text{ и } i \notin C'^*; \tag{22}$$
$$\operatorname{len}[i,b'] \leqslant \rho'(i,b'),\ \text{если } i \notin C'^*; \tag{23}$$

$$\mathrm{len}[a'j,] \leqslant n-1, \text{ если } j \notin C'^*; \tag{24}$$
$$\mathrm{len}[a',j] \leqslant \rho'(a',j), \text{ если } j \in C'^*; \tag{25}$$
$$\mathrm{len}[b,j] \leqslant n-1, \text{ если } j \notin C^*; \tag{26}$$
$$\mathrm{len}[b,j] \leqslant \rho(b,j), \text{ если } j \in C^*. \tag{27}$$

В соответствии с (20) и (21) наибольшее значение оценка $\mathrm{len}[i,a]$ принимает при $i \notin C^*$, и в соответствии с (22) и (23) наибольшее значение оценка $\mathrm{len}[i,b']$ принимает при $i \notin C'^*$. Значит, при $i \notin C^* \cup C'^*$ независимо от того, какой из путей $w_i$, $u_i$ содержится в $W(\Gamma)$,

$$\min\{\mathrm{len}[i,a],\mathrm{len}[i,b']\} \leqslant \max\{\mathrm{len}(w_i),\mathrm{len}(u_i)\} + \max\{\rho(z,a),\rho'(z',b')\};$$

при $i \in C^* \cup C'^*$ оценка $\min\{\mathrm{len}[i,a],\mathrm{len}[i,b']\}$ ниже этой. При любом $i \notin C^* \cup C'^*$ в соответствии с определением путей $w_i$, $u_i$ верна оценка

$$\max\{\mathrm{len}(w_i),\mathrm{len}(u_i)\} \leqslant n-l-\lambda,$$

отсюда, учитывая, что $\rho(z,a) \leqslant l-1$ и $\rho'(z',b') \leqslant \lambda-1$, получаем для любого $i \in \in \{1,\dots,n\}$

$$\min\{\mathrm{len}[i,a],\mathrm{len}[i,b']\} \leqslant n-\lambda-1. \tag{28}$$

Кратчайший путь из $a$ в $a'$ (из $b'$ в $b$) по определению не содержит вершин контуров $C$ и $C'$, исключая начальную и конечную вершины, поэтому

$$\max\{\mathrm{len}[a,a'],\mathrm{len}[b,b']\} \leqslant n-l-\lambda+1. \tag{29}$$

Для $\mathrm{len}[a',j]$ и $\mathrm{len}[b,j]$ в соответствии с (24)–(27) при любом $j \in \{1,\dots,n\}$ верна оценка

$$\max\{\mathrm{len}[a',j],\mathrm{len}[b,j]\} \leqslant n-1. \tag{30}$$

Из формул (11)–(14) и (15)–(18) следует, что для любых $i,j \in \{1,\dots n\}$

$$t_0(i,j) \leqslant \mathrm{len}[i,a] + \mathrm{len}[a,a'] + \mathrm{len}[a',j],$$
$$t_1(i,j) \leqslant \mathrm{len}[i,ba'] + \mathrm{len}[b',b] + \mathrm{len}[b,j],$$

где первая оценка достигается при $i \neq a$, $j \neq a'$, а вторая — при $i \neq b'$, $j \neq b$ (см. рис. 1). В остальных случаях оценка $t_0(i,j)$ ниже этой. Следовательно,

$$\max_{(i,j)}\{\min\{t_0(i,j),t_1(i,j)\}\} \leqslant$$
$$\leqslant \min\{\mathrm{len}[i,a],\mathrm{len}[i,b']\} + \max\{\mathrm{len}[a,a'],\mathrm{len}[b',b]\} + \max\{\mathrm{len}[a',j],\mathrm{len}[b,j]\},$$

отсюда в соответствии с (28)–(30) получаем

$$\max_{(i,j)}\{\min\{t_0(i,j),t_1(i,j)\}\} \leqslant 3n-l-2\lambda-1.$$

Из этой оценки и из (19) получаем оценку теоремы 1,*а*.

б) Пусть $H = C^* \cap C'^*$ и $a \in H$ (рис. 2). Определим пути $w_q(i,j)$ из $i$ в $j$, проходящие через вершину $a$ контура $C'$, где $q = 0,1,\dots,\lambda-1$:

$$w_q(i,j) = q \cdot C(a) \bullet z(q) \cdot C'(a), \text{ если } i=j=a;$$
$$w_q(i,j) = q \cdot C(a) \bullet z(q) \cdot C'(a) \bullet [a,j], \text{ если } i=a, j \neq a;$$
$$w_q(i,j) = [i,a] \bullet q \cdot C(a) \bullet z(q) \cdot C'(a), \text{ если } i \neq a, j=a;$$
$$w_q(i,j) = [i,a] \bullet q \cdot C(a) \bullet z(q) \cdot C'(a) \bullet [a,j], \text{ если } i \neq a, j \neq a.$$

Длины путей $w_q(i,j)$ соответственно равны

$$\operatorname{len}(w_q(i,j)) = ql + z(q)\lambda, \text{ если } i = j = a; \tag{31}$$
$$\operatorname{len}(w_q(i,j)) = ql + z(q)\lambda + \operatorname{len}[a,j], \text{ если } i = a, j \neq a; \tag{32}$$
$$\operatorname{len}(w_q(i,j)) = ql + z(q)\lambda + \operatorname{len}[i,a], \text{ если } i \neq a, j = a; \tag{33}$$
$$\operatorname{len}(w_q(i,j)) = ql + z(q)\cdot\lambda + \operatorname{len}[i,a] + \operatorname{len}[a,j], \text{ если } i \neq a, j \neq a. \tag{34}$$

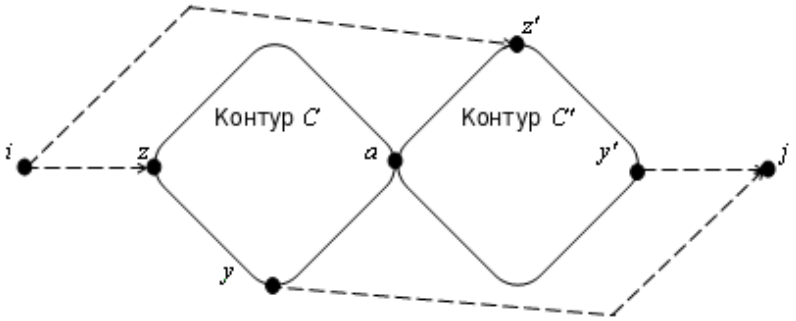


Рис. 2. Пересекающиеся контуры $C$ и $C'$ в орграфе

Из формул (31)–(34) следует, что в соответствии с (2) для любых $i,j \in \{1,\ldots,n\}$ множество путей $\{w_q(i,j) : q = 0,1,\ldots,\lambda-1\}$ есть $(\theta + t_2(i,j),\lambda)$-множество путей из $i$ в $j$, где

$$t_2(i,j) = 0, \text{ если } i = j = a; \tag{35}$$
$$t_2(i,j) = \operatorname{len}[a,j], \text{ если } i = a, j \neq a; \tag{36}$$
$$t_2(i,j) = \operatorname{len}[i,a], \text{ если } i \neq a, j = a; \tag{37}$$
$$t_2(i,j) = \operatorname{len}[i,a] + \operatorname{len}[a,j], \text{ если } i \neq a, j \neq a. \tag{38}$$

Следовательно, в соответствии с утверждениями 3,*б* и 4

$$\exp\Gamma \leqslant \theta + \max_{(i,j)} t_2(i,j). \tag{39}$$

В случае *б* путь $w_i$ при $i \notin C^*$ и путь $u_i$ при $i \notin C'^*$ определим так же, как в случае *а*, за исключением того, что конечные вершины путей могут, в частности, принадлежать $H$. В $W(\Gamma)$ для $i \notin C^* \cup C'^*$ содержится хотя бы один из путей $w_i$ и $u_i$.

В соответствии с (20)–(23) при $i \notin C^* \cup C'^*$

$$\operatorname{dist}(i,H) \leqslant \min\{\operatorname{len}(w_i), \operatorname{len}(u_i)\} + \max\{\rho(z,H), \rho'(z',H)\},$$

при $i \in C^* \cup C'^*$ оценка $\operatorname{dist}(i,H)$ ниже этой. При любом $i \notin C^* \cup C'^*$ по определению путей $w_i$ и $u_i$

$$\min\{\operatorname{len}(w_i), \operatorname{len}(u_i)\} \leqslant n - l - \lambda + h,$$

$\rho(z,H) \leqslant l - h$, $\rho'(z',b') \leqslant \lambda - h$. Тогда независимо от того, какие из путей $w_i$, $u_i$ содержатся в $W(\Gamma)$, для любого $i \in \{1,\ldots,n\}$ верна оценка

$$\operatorname{dist}(i,H) \leqslant n - \lambda. \tag{40}$$

Пусть $w_j^*$ — кратчайший путь от контура $C$ до $j$ ($y$ — начальная вершина) при $j \notin C^* \cup C'^*$, не проходящий через вершины из $C'^*$, за исключением, быть может, начальной вершины, и $u_j^*$ — кратчайший путь от контура $C'$ до $j$ ($y'$ — начальная вершина), не проходящий через вершины контура $C$, за исключением, быть может, начальной вершины. В $W(\Gamma)$ содержится хотя бы один из путей $w_j^*$ и $u_j^*$. Значит, для $j \notin C^* \cup C'^*$ и для любого $a \in H$ выполнено

$$\operatorname{len}[a, j] \leqslant \max\{\rho(a, y) + \operatorname{len}(w_j^*), \rho'(a, y') + \operatorname{len}(u_j^*)\}$$

(при $j \in C^* \cup C'^*$ и любом $a \in H$ оценка $\operatorname{len}[a, j]$ ниже этой), где по определению путей $w_j^*$ и $u_j^*$

$$\operatorname{len}(w_j^*) \leqslant n - l - \lambda + h, \operatorname{len}(u_j^*) \leqslant n - l - \lambda + h,$$

и $\rho(a, y) \leqslant l - 1$, $\rho'(a, y') \leqslant \lambda - 1$. Отсюда, независимо от того, какие из путей $w_j^*$, $u_j^*$ содержатся в $W(\Gamma)$, получаем для любого $j \in \{1, \ldots, n\}$ и любого $a \in H$

$$\operatorname{len}[a, j] \leqslant n - \lambda + h - 1. \tag{41}$$

Из формул (35)–(38) следует, что для любых $i, j \in \{1, \ldots, n\}$

$$t_2(i, j) \leqslant \operatorname{dist}(i, H) + \max_{a \in H}\{\operatorname{len}[a, j]\},$$

где наибольшее значение оценки достигается при $i, j \notin H$ (см. рис. 2). В остальных случаях оценка $t_2(i, j)$ ниже этой. Следовательно, используя оценки (40) и (41), получаем

$$\max_{(i,j)} t_2(i, j) \leqslant 2n - 2\lambda + h - 1.$$

Из этой оценки и из (39) получаем оценку теоремы 1,*б*. ■

**Следствие 1.** Для любого примитивного $n$-вершинного орграфа $\Gamma$ при $n > 2$ верно:

а) если циклы $C$ и $C'$ не имеют общих вершин, то

$$\exp\Gamma \leqslant \left\lfloor \frac{n+1}{2} \right\rfloor \left\lceil \frac{n+1}{2} \right\rceil \leqslant n^2/4 + n/2 + 1/4;$$

б) если циклы $C$ и $C'$ имеют $h$ общих вершин, где $1 \leqslant h \leqslant \lambda$, то

$$\exp\Gamma \leqslant \left\lfloor \frac{n+h+2}{2} \right\rfloor \left\lceil \frac{n+h+2}{2} \right\rceil - 2h - n \leqslant n^2 - 2n + 2.$$

***Доказательство.***

а) По теореме 1,*а* $\exp\Gamma \leqslant \psi(n, l, \lambda)$, где $\psi(n, l, \lambda) = l\lambda - 2l - 3\lambda + 3n$. Заметим, что $\psi(n, l, \lambda) - \psi(n, l, \lambda - 1) > 0$ при любых допустимых $n, l$. Так как $\lambda \leqslant n - l$, то

$$\exp\Gamma \leqslant \psi(n, l, n - l) = (n + 1 - l)l,$$

где произведение $(n - l + 1)l$ принимает наибольшее значение при $l = \left\lceil \dfrac{n+1}{2} \right\rceil$.

б) По теореме 2,*б* $\exp\Gamma \leqslant \varphi(n, l, \lambda, h)$, где $\varphi(n, l, \lambda, h) = l\lambda - l - 3\lambda + h + 2n$. Заметим, что функция $\varphi(n, l, \lambda, h)$ монотонно возрастает по переменной $\lambda$ при любых допустимых $n, l, h$. Так как $\lambda \leqslant n - l + h$, то

$$\exp\Gamma \leqslant \varphi(n, l, n-l+h, h) = (n+h+2-l)l - 2h - n,$$

где выражение $(n+h+2-l)l-2h-n$ принимает наибольшее значение при $l = \left\lceil \dfrac{n+h+2}{2} \right\rceil$.

Заметим, что функция $\varphi(n, l, n-l+h, h)$ монотонно возрастает по переменной $h$ при любых допустимых $n, l$ и возрастает по переменной $l$ при любых допустимых $n, h$. Отсюда следует абсолютная оценка Виландта для примитивных $n$-вершинных орграфов $\exp\Gamma \leqslant \varphi(n, n, n-1, n-1) = n^2 - 2n + 2$. ■

При $n > 2$ (при $n = 2$) *$n$-вершинным графом Виландта* назовем орграф, состоящий из гамильтонова контура $C$, к которому добавлена дуга $(i, j)$, где $\rho(i, j) = 2$ (петля $(i, i)$), $i \in \{1, \ldots, n\}$. Множество $n$-вершинных графов Виландта обозначим $\Gamma_{\mathrm{W}}(n)$.

**Теорема 2.** При любом $n > 1$ множество $\Gamma_{\mathrm{W}}(n)$ состоит из $n!$ изоморфных графов; абсолютная оценка Виландта достигается на графах Виландта, и только на них. Для остальных примитивных $n$-вершинных орграфов $\Gamma$ верна оценка, достижимая при $l = n$, $\lambda = n-2$, где $n > 3$ и нечетное:

$$\exp\Gamma \leqslant n^2 - 3n + 4.$$

***Доказательство.*** В силу определения множество $\Gamma_{\mathrm{W}}(n)$ инвариантно относительно любой перенумерации вершин, следовательно, $\Gamma_{\mathrm{W}}(n)$ состоит из $n!$ изоморфных графов, экспоненты которых одинаковы.

Пусть граф $\Gamma$ состоит из гамильтонова контура $C = (1, 2, \ldots, n)$ и к дугам контура $C$ добавлена дуга $(n-1, 1)$. Тогда граф $\Gamma$ есть объединение контура $C$ длины $n$ и простого контура $C' = (1, 2, \ldots, n-1)$ длины $n-1$. Следовательно, по теореме $1,б$ граф $\Gamma$ примитивен и $\exp\Gamma$ оценивается сверху величиной $\varphi(n, n, n-1, n-1) = n^2 - 2n + 2$. Покажем, что $\exp\Gamma = n^2 - 2n + 2$.

В $\Gamma$ всякий путь $w$ из $n$ в $n$ состоит из $p$-кратно пройденного контура $C$ и $q$-кратно пройденного контура $C'$, где $p$ — натуральное, $q$ — целое неотрицательное. Тогда в $\Gamma$ не имеется пути из $n$ в $n$ длины $(n-1)^2$ в силу неразрешимости относительно $p$ и $q$ уравнения $pn + q(n-1) = (n-1)^2$. Следовательно, по утверждению $3,в$ $\exp\Gamma > (n-1)^2$, т. е. $\exp\Gamma = (n-1)^2 + 1 = n^2 - 2n + 2$.

Покажем, что $\exp\Gamma' < n^2 - 2n + 2$ для любого $n$-вершинного примитивного орграфа $\Gamma' \notin \Gamma_{\mathrm{W}}(n)$. В самом деле, в $\Gamma'$ длина $l$ кратчайшего простого контура не превышает $n-2$. Поэтому в соответствии с (1) получаем при $l \leqslant n-2$ оценку теоремы 2. ■

**Замечание 1.** Известно, что имеются натуральные числа, меньшие $n^2 - 2n + 2$ и не являющиеся значениями экспонента какого-либо $n$-вершинного орграфа. Эти числа образуют так называемые «лакуны», т. е. пропуски в натуральном ряду чисел. Например, первыми были обнаружены (авторами A. L. Dulmage, N. S. Mendelsohn) «лакуны» вида $[n^2 - 3n + 5, (n-1)^2]$ и $[n^2 - 4n + 7, n^2 - 3n + 2]$. В дальнейшем эти результаты были обобщены.

### 3. Оценки экспонентов неориентированных графов

Для неориентированного цикла $C$ длины $l > 2$ и для $i, j \in C^*$ обозначим через $c(i, j)$ кратчайший путь от $i$ до $j$, составленный только из ребер цикла $C$, где $c(i, i) = w_{\varnothing}$, и через $y(i, j)$ — путь от $i$ до $j$, дополняющий путь $c(i, j)$ до полного цикла $C$. При нечетном $l$ длины путей $c(i, j)$ и $y(i, j)$, равные соответственно $\rho(i, j)$ и $l - \rho(i, j)$, имеют различную четность, при этом $\rho(i, j) < l - \rho(i, j)$.

Для цикла $C'$ длины 2 с начальной вершиной $a$ обозначим: $q \cdot \mathbf{2}(a) = q \cdot C'(a)$, $q$ — целое неотрицательное.

**Утверждение 5.** Пусть $l$ нечетное, $q = (l-1-2\rho(i,j))/2$ и $z(i,j) = c(i,j) \bullet q \cdot \mathbf{2}(j)$, где $i, j \in C^*$, тогда

$$\operatorname{len}(y(i,j)) - \operatorname{len}(z(i,j)) = 1.$$

***Доказательство.*** По определению $q \cdot \mathbf{2}(j)$ — цикл длины $2q$ с начальной вершиной $j$, где по условию $q$ — целое неотрицательное. Тогда $c(i,j) \bullet q \cdot \mathbf{2}(j)$ есть путь от $i$ до $j$ длины $\rho(i,j) + 2q$, т. е.

$$\operatorname{len}(z(i,j)) = \rho(i,j) + 2q = l - 1 - \rho(i,j).$$

Вместе с тем по определению $\operatorname{len}(y(i,j)) = l - \rho(i,j) = \operatorname{len}(z(i,j)) + 1$. ■

Обозначим через $e(C)$ эксцентриситет цикла $C$ в неориентированном графе $\Gamma'$, т. е.

$$e(C) = \max_{i \notin C} \operatorname{dist}[i, C] = \max_{i \notin C} \{\min_{j \in C} \operatorname{len}[i,j]\}.$$

**Теорема 3.**

а) Пусть $n > 1$, $l$ — длина длиннейшего простого цикла $C$ нечетной длины в примитивном $n$-вершинном графе $\Gamma'$, $1 \leqslant l \leqslant n$. Тогда

$$\exp \Gamma' \leqslant 2e(C) + l - 1 \leqslant 2n - l - 1.$$

б) Если простые циклы нечетных длин покрывают множество вершин графа $\Gamma'$, то

$$\exp \Gamma' \leqslant n - 1.$$

***Доказательство.***

а) Каждая вершина графа $\Gamma'$ принадлежит хотя бы одному циклу длины 2, поэтому в соответствии с утверждением 4 для получения оценки $\exp \Gamma \leqslant t$ достаточно для любых $i, j \in \{1, \ldots, n\}$ построить $(t, 2)$-множество путей из $i$ в $j$.

При $n > 2$ обозначим: $C$ — простой цикл нечетной длины $l' > 2$ в графе $\Gamma'$ (рис. 3); $a$ и $b$ — конечные вершины кратчайших путей от цикла $C$ соответственно до $i$ и до $j$, где $i, j \notin C^*$, $a, b \in C^*$. Определим пути $u(i,j)$ и $v(i,j)$ из $i$ в $j$:

$$u(i,j) = z(i,j), v(i,j) = y(i,j), \text{ если } i, j \in C^*;$$
$$u(i,j) = [i,a] \bullet z(a,j), v(i,j) = [i,a] \bullet y(a,j), \text{ если } i \notin C^*, j \in C^*;$$
$$u(i,j) = z(i,b) \bullet [b,j], v(i,j) = y(i,b) \bullet [b,j], \text{ если } i \in C^*, j \notin C^*;$$
$$u(i,j) = [i,a] \bullet z(a,b) \bullet [b,j], v(i,j) = [i,a] \bullet y(a,b) \bullet [b,j], \text{ если } i, j \notin C^*.$$

Тогда пара путей $(u(i,j), v(i,j))$ в соответствии с утверждением 5 образует $(t(i,j), 2)$-множество путей из $i$ в $j$ при любых $i, j \in \{1, \ldots, n\}$, где

$$t(i,j) = l' - \rho(i,j) - 1, \text{ если } i, j \in C^*;$$
$$t(i,j) = \operatorname{len}[i,a] + l' - \rho(a,j) - 1, \text{ если } i \notin C^*, j \in C^*;$$
$$t(i,j) = l' - \rho(i,b) - 1 + \operatorname{len}[b,j], \text{ если } i \in C^*, j \notin C^*;$$
$$t(i,j) = \operatorname{len}[i,a] + l' - \rho(a,b) - 1 + \operatorname{len}[b,j], \text{ если } i, j \notin C^*.$$

В соответствии с определением $\max\{\operatorname{len}[i,a], \operatorname{len}[b,j]\} \leqslant e(C) \leqslant n - l'$ при всех $i, j \notin C^*$ и $\rho(a,b) \geqslant 0$ при всех $a, b \in C^*$. Значит, $t(i,j) \leqslant 2e(C) + l' - 1 \leqslant 2n - l' - 1$ при

любых $i, j \in \{1, \ldots, n\}$, т. е. $\exp \Gamma' \leqslant 2e(C) + l' - 1 \leqslant 2n - l' - 1$. Так как рассмотрен произвольный простой цикл нечетной длины $l' > 2$, то получаем оценку теоремы 3,*а*.

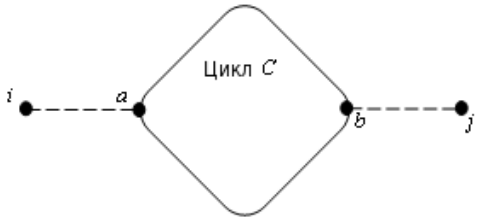


Рис. 3. Цикл $C$ в неориентированном графе

Если $n > 1$ и $l = 1$, т. е. в $\Gamma'$ имеется петля в вершине $a$ и не имеется циклов нечетной длины $l' > 2$, то пути $u(i, j)$ и $v(i, j)$ из $i$ в $j$ определим так:

$$u(a, a) = 2 \cdot (a, a), v(i, j) = (a, a);$$
$$u(i, a) = [i, a] \bullet (a, a), v(i, a) = [i, a], \text{ если } i \neq a;$$
$$u(a, j) = (a, a) \bullet [a, j], v(a, j) = [a, j], \text{ если } j \neq a;$$
$$u(i, j) = [i, a] \bullet (a, a) \bullet [a, j], v(i, j) = [i, a] \bullet [a, j], \text{ если } i \neq a, j \neq a.$$

Пара путей $(u(i, j), v(i, j))$ в соответствии с утверждением 5 образует $(t(i, j), 2)$-множество путей из $i$ в $j$ при любых $i, j \in \{1, \ldots, n\}$, где $t(a, a) = 1$, $\max\{t(i, a), t(a, j)\} \leqslant n - 1$ при $i \neq a$, $j \neq a$ и $t(i, j) \leqslant 2n - 2$ при $i \neq a$, $j \neq a$. Следовательно, $t(i, j) \leqslant 2n - 2$ при любых $i, j \in \{1, \ldots, n\}$, т. е. $\exp \Gamma' \leqslant 2n - 2$ при $n > 1$. Эта оценка совпадает с оценкой $2n - l - 1$ теоремы 3,*а* при $l = 1$.

б) При условиях теоремы 3,*б* $i \in C_i^*$, где $C_i$ — простой цикл нечетной длины $l_i$ и $C_i^*$ — блок покрытия множества вершин, $i \in \{1, \ldots, n\}$. Пусть $b$ — концевая вершина кратчайшего пути от цикла $C_i$ до $j$, где $b \in C_i^*$. Тогда $(t(i, j), 2)$-множество путей из $i$ в $j$ при любых $i, j \in \{1, \ldots, n\}$ образовано путями $u(i, j)$ и $v(i, j)$ из $i$ в $j$, где при $n > 2$

$$u(i, j) = z(i, j), v(i, j) = y(i, j), \text{ если } j \in C_i^*;$$
$$u(i, j) = z(i, b) \bullet [b, j], v(i, j) = y(i, b) \bullet [b, j], \text{если } j \notin C_i^*.$$

Отсюда, обозначая через $\rho_i$ расстояния на цикле $C_i, i \in \{1, \ldots, n\}$, получаем

$$t(i, j) = l_i - \rho_i(i, j) - 1, \text{ если } j \in C_i^*;$$
$$t(i, j) = l_i - \rho_i(i, b) - 1 + \operatorname{len}[b, j], \text{ если } j \notin C_i^*.$$

В соответствии с определением $\operatorname{len}[b, j] \leqslant n - l_i$ при всех $j \notin C^*$ и $\rho_i(i, b) \geqslant 0$ при всех $b \in C^*$. Отсюда $t(i, j) \leqslant n - 1$ при любых $i, j \in \{1, \ldots, n\}$, т. е. $\exp \Gamma' \leqslant n - 1$ при $n > 2$. Теорема доказана. ■

Если $n = 2$, то при условиях теоремы 3,*б* граф $\Gamma'$ полный, следовательно, $\exp \Gamma' = 1$.

При $l = 1$ из теоремы 3,*а* получаем известную оценку для связных графов с петлей.

**Следствие 2.** Для любого примитивного $n$-вершинного графа $\Gamma'$ имеет место $\exp \Gamma' \leqslant 2n - 2$.

Обозначим через $\Gamma_P(n)$ множество примитивных $n$-вершинных графов, состоящих из гамильтонова пути $w$ и петли, инцидентной одной из концевых вершин.

**Теорема 4.** При любом $n > 1$ множество $\Gamma_P(n)$ состоит из $n!$ изоморфных графов; абсолютная оценка $2n - 2$ достигается на графах из $\Gamma_P(n)$, и только на них.

***Доказательство.*** Из определения множества $\Gamma_P(n)$ следует, что оно инвариантно относительно любой перенумерации вершин, следовательно, $\Gamma_P(n)$ состоит из $n!$ изоморфных графов. По следствию теоремы 3 $\exp\Gamma' \leqslant 2n-2$ для любого графа $\Gamma' \in \Gamma_P(n)$.

Пусть граф $\Gamma'$ состоит из гамильтонова пути $P=(1,2,\ldots,n)$ и петли $(n,n)$. Тогда пути из 1 в 1 нечетной длины $2n-3$ не существует. Значит, в соответствии с утверждением $3,в$ $\exp\Gamma' = 2n-2$.

Покажем, что на связном графе $\Gamma$, не принадлежащем $\Gamma_P(n)$, оценка $2n-2$ не достигается. В силу теоремы $3,a$ абсолютная оценка $2n-2$ достижима только на связном графе $\Gamma$ с петлей, не имеющем циклов нечетной длины $l>1$.

Пусть $\Gamma$ — гамильтонов цикл $(1,2,\ldots,n)$ с петлей в вершине $i$, где $i\in\{1,\ldots,n\}$, или гамильтонов путь с петлей в вершине $i$, где $1<i<n$, тогда $e(i)\leqslant n-2$.

Пусть $\Gamma$ не есть гамильтонов путь или цикл, т. е. содержит, кроме петли в вершине $i$, где $i\in\{1,\ldots,n\}$, вершину $r$ степени выше 2. Кратчайший путь $[i,j]$ при $i\neq j$ либо проходит, либо не проходит через вершину $r$. Тогда в первом случае $\mathrm{len}[i,j]\leqslant \mathrm{len}[i,r]+$ $+\mathrm{len}[r,j]\leqslant n-2$, так как кратчайшие пути $[i,r]$ и $[r,j]$ не содержат общих вершин, кроме $r$ (иначе кратчайший путь $[i,j]$ не проходит через $r$), и содержат не более двух вершин, смежных с $r$. Во втором случае $\mathrm{len}[i,j]\leqslant n-2$, так как путь $[i,j]$ не содержит вершину $r$. Следовательно, $e(i)\leqslant n-2$, если $\Gamma\notin\Gamma_P(n)$, и по теореме $3,a$ $\exp\Gamma\leqslant 2n-4$. Теорема доказана. ■

## ЛИТЕРАТУРА

1. *Биркгоф Г.* Теория решёток. М.: Наука, 1984.
2. *Сачков В. Н., Тараканов В. Е.* Комбинаторика неотрицательных матриц. М.: ТВП, 2000.
3. *Wielandt H.* Unzerlegbare nicht negative Matrizen // Math. Zeitschr. 1950. No. 52. P. 642–648.
4. *Берж К.* Теория графов и её применение. М.: ИЛ, 1962.

# ДИСКРЕТНЫЕ МОДЕЛИ РЕАЛЬНЫХ ПРОЦЕССОВ

DOI 10.17223/20710410/12/10

УДК 621.391.1:004.7

## КУМУЛЯТИВНЫЙ СИНТЕЗ: КЛЕТОЧНО-АВТОМАТНАЯ МОДЕЛЬ ФИЗИКО-ХИМИЧЕСКИХ ПРОЦЕССОВ НА СТАДИИ СХЛОПЫВАНИЯ ПОРОШКОВОЙ ОБЛИЦОВКИ[1]

О. Л. Бандман*, С. А. Кинеловский**

*Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск, Россия

** Институт гидродинамики им. М. А. Лаврентьева СО РАН, г. Новосибирск, Россия

**E-mail:** bandman@ssd.sscc.ru, skin@hydro.nsc.ru

Предложена дискретная математическая модель, предназначенная для компьютерного моделирования физико-химических процессов в порошковой смеси под воздействием ударного давления, вызванного взрывом. Математическая модель представляет собой двумерный клеточный автомат типа «решеточный газ» с гексагональной структурой дискретного пространства в плоскости, перпендикулярной к оси формирующейся струи. Гранулы порошка имитируются «частицами» решеточного газа, имеющими разную массу для разных веществ и при столкновениях вступающих в химическую реакцию. Программная реализация эволюции клеточного автомата выполнена для пяти случаев, различающихся углами между воздействием ударной волны и осью образующейся струи. Моделирование проводилось для смеси порошков вольфрама и углерода. Результатами моделирования являются количества вольфрама, углерода и карбидов типа WC и $W_2C$, получающихся на начальной стадии образования порошковой струи при воздействии взрыва. Сравнение результатов с полученными при натурных испытаниях позволило определить основные параметры клеточно-автоматной модели.

**Ключевые слова:** *компьютерное моделирование, клеточный автомат, решеточный газ, карбиды вольфрама.*

### Введение

Клеточно-автоматные (КА) модели пространственной динамики привлекают большое внимание, поскольку они способны отображать нелинейные и разрывные процессы и могут быть полезными там, где дифференциальные уравнения с частными производными оказываются бессильными [1, 2]. В основном это относится к процессам на микро- и наноуровнях, поскольку КА может оперировать дискретными событиями, такими, как перемещения, изменения состояний, химические превращения и другие взаимодействия между реальными или абстрактными частицами, иногда наделяя их скоростью движения в дискретном пространстве и дискретном времени. В частности, одной из наиболее развитых областей применения КА является так называемая КА-гидродинамика, получившая название «решеточный газ» (Lattice-Gas models) [3].

[1]Работа поддержана Программой фундаментальных исследований Президиума РАН 14-6, 2009, и Междисциплинарным интеграционным проектом ИП-32 СО РАН, 2009.

Поскольку кумулятивная струя может быть описана на основе гидродинамической модели [4], а химические превращения при взаимодействиях частиц являются дискретными событиями, представляется естественным использовать модель «решеточного газа», встроив в нее дискретные события химических взаимодействий между отталкивающимися частицами. На этих соображениях основывается предлагаемая КА-модель физико-химических процессов в порошковой струе, получающейся в устройстве, используемом для исследования кумулятивного синтеза новых наноструктурных соединений [5].

В работе формально и содержательно описана КА-модель (п. 1), дан алгоритм моделирования (п. 2) и, наконец, представлены результаты компьютерного моделирования процесса схлопывания облицовки из порошковой смеси вольфрама и углерода в соответствии с испытаниями, проводимыми в рамках создания методов кумулятивного синтеза новых материалов и структур в Институте гидродинамики СО РАН (п. 3).

## 1. КА-модель процесса схлопывания порошковой облицовки

Гидродинамическая КА-модель, называемая в российской научной литературе «решеточным газом», а в западной — FHP-моделью (по фамилиям авторов работы [3]), имитирует движение абстрактных частиц в дискретном пространстве гексагональной структуры, плотно заполненном клетками. В каждой клетке одновременно может находиться несколько частиц, каждая из которых снабжена вектором скорости, направленным в одном из шести направлений гексагональной решетки. Такое состояние клетки однозначно представляется булевым вектором, число компонент в котором $n = 6$. Массив состояний всех клеток называется глобальной конфигурацией. Смена глобальных конфигураций происходит синхронно во всех клетках одновременно и называется итерацией. Итерация разделена на две фазы: фазу сдвига и фазу столкновения. На фазе сдвига имитируется перемещение каждой частицы в соседнюю клетку по направлению скоростей движения частиц. На фазе столкновения происходит изменение состояний клетки, иногда с заданной вероятностью (рис. 1). Изменения состояний клеточного массива подчиняются законам сохранения массы и импульса.

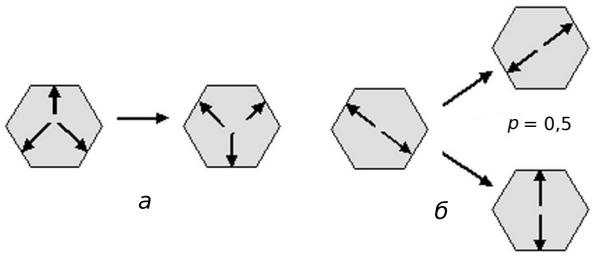


Рис. 1. Графическое изображение правил столкновения частиц в FHP-модели: *а* — детерминированные правила; *б* — вероятностные

FHP-модель является математическим описанием потока вязкой жидкости. Этот факт имеет строгое аналитическое доказательство [3]. Правомерность применения ее для моделирования процесса образования струи из порошковой смеси под действием взрыва с химическими взаимодействиями между частицами основывается на соединении известного опыта применения КА типа решеточного газа для потоков жидкостей и вероятностных кинетических моделей для химических реакций. Поскольку в этом случае аналитические доказательства соответствия модели и явления невозможны, то

параметры предлагаемой модели (правила и вероятности смены состояний КА) должны выбираться, исходя из данных натурных испытаний. Последние на данном этапе исследований относятся к категории научного поиска и постоянного совершенствования, и, следовательно, то же самое должно происходить с моделью. С одной стороны, модель постоянно уточняет свои параметры на основе натурных результатов. С другой стороны, результаты компьютерного моделирования помогают анализировать результаты испытаний, которые заключаются в следующем. Порошковая струя образуется при схлопывании конструкции, состоящей их двух конусных оболочек, между которыми засыпана смесь из измельченных вольфрама и углерода в пропорции 1:1 по объему [6]. Взрыв вещества за пределами внешнего конуса создает ударное давление на него. В результате порошок устремляется к оси конуса и образует струю, вытекающую в камеру, где она затем улавливается специальной ловушкой. На рис. 2 схематично даны продольный и поперечный разрезы конусных оболочек.

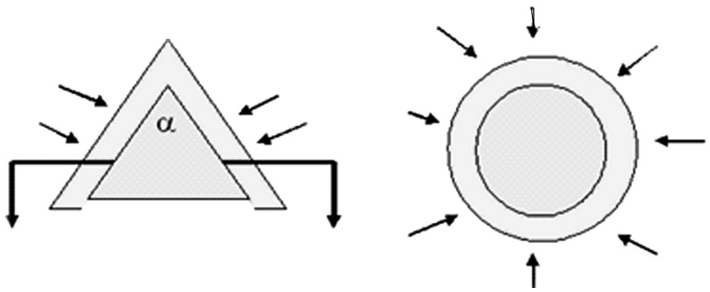


Рис. 2. Схематическое изображение схлопывающейся порошковой облицовки

Поскольку главная составляющая скорости движения частиц направлена от стенок конуса к его оси и происходящий процесс симметричен относительно оси конуса, то в первом приближении допустима двумерная аппроксимация процесса в виде проекции на плоскость, перпендикулярную оси конуса. Конкретные физические данные взяты из экспериментального исследования процессов кумулятивного синтеза в Институте гидродинамики им. М. А. Лаврентьева СО РАН [6].

В основу построения КА-модели положены следующие принципы.

1) Каждой грануле ставится в соответствие модельная частица. В процессе участвуют три типа модельных частиц: частица углерода C с модельной массой $\mathrm{m(C)} = 1$, частицы вольфрама W, карбида WC и карбида $\mathrm{W_2C}$ с модельными массами $\mathrm{m(W)} = 10$, $\mathrm{m(WC)} = 8$ и $\mathrm{m(W_2C)} = 9$ соответственно. Модельные массы выбраны так, чтобы их отношения были равны отношениям их физических масс, например $\mathrm{m(C)/m(W)} \approx \approx \mathrm{M(C)/M(W)}$.

2) Структура дискретного пространства, в котором перемещаются частицы, — гексагональная. Каждая клетка имеет шесть соседей, расстояния между центрами клеток равны единице.

3) Состояние каждой клетки представлено вектором, в котором содержится информация о частицах, находящихся в клетке, с указанием направления скорости их движения.

4) Функционирование модели синхронное, т. е. на каждом итерационном шаге все клетки одновременно меняют свои состояния в соответствии с функциями переходов КА.

5) Итерационный шаг имеет две фазы: на фазе сдвига каждая частица перемещается в соседнюю клетку, на которую указывает вектор ее скорости. На фазе столкновения в клетке происходит изменение направлений скоростей и типов частиц.

6) Правила перехода в новые состояния строятся таким образом, чтобы при моделировании не нарушались законы сохранения массы и импульса.

7) При столкновениях частиц вольфрама с частицами углерода происходит химическая реакция, приводящая к образованию карбидов WC и $W_2C$. Вероятность образования того или иного типа карбида зависит от кинетической энергии столкновения, а также от энергии активации происходящих реакций и факторов, учитывающих расход энергии на другие преобразования.

Для формального описания КА-модели необходимо математически представить три понятия $\langle A, X, \Theta \rangle$, где $A$ — алфавит состояний клеток, $X$ — дискретное пространство и $\Theta$ — набор функций переходов.

*Алфавит состояний* $A$ — это множество векторов вида $s = (s_{11}, s_{10}, \ldots, s_g, \ldots, s_0)$. Каждая $k$-я пара $(ss)_k = (s_{g+1}, s_g)$, где $k = g/2$, $g \bmod 2 = 0$, соответствует частице одного типа:

$(ss)_k = (0,0)$ — частица отсутствует,

$(ss)_k = (0,1)$ — присутствует частица C,

$(ss)_k = (1,0)$ — присутствует частица W

при направлении их движения к соседней клетке с номером $n_k$. Так, например, состояние клетки $s = (0,1,0,0,1,0,0,0,0,0,0,1)$ означает, что в клетке находятся три частицы $(\mathrm{C}, -, \mathrm{W}, -, -, \mathrm{C})$ (рис. 3).

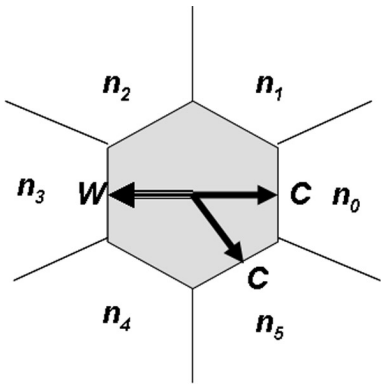


Рис. 3. Клетка, состояние которой равно $s = (0,1,0,0,1,0,0,0,0,0,0,1)$, содержит одну частицу вольфрама и две частицы углерода

Каждое состояние клетки характеризуется суммарными модельными массой $\mathrm{m}(s)$ и импульсом $\mathbf{p}(s)$:

$$\mathrm{m}(s) = \sum_{k=0}^{5} \mathrm{m}((ss)_k), \quad \mathbf{p}(s) = \sum_{k=0}^{5} \mathbf{e}_k \mathrm{m}((ss)_k),$$

где $\mathbf{e}_k$, $k = 0, \ldots, 5$, — единичные векторы скорости, направленные к центрам соседних клеток.

*Дискретное клеточное пространство* — это множество $X$ координат клеток, которые фактически являются их именами. Пара $(s_i, x_i)$, в которой $s_i \in A$, $x_i \in X$, является *клеткой*, а множество клеток $\Omega$, в котором все клетки имеют разные имена и $|\Omega| = |X|$, называется *клеточным массивом*. Далее имя произвольной клетки обозначается либо парой координат $x = (i, j)$, либо одним символом $x$, а множество соседей любой

клетки — множеством $n(x) = \{n_0(x), \ldots, n_5(x)\}$. Координаты центров клеток в гексагональной структуре можно задавать по-разному. Здесь принят самый простой способ отображения гексагональной решетки на квадратную с растяжением оси $j$. Он состоит в том, что одному гексагону ставится в соответствие пара пикселей $(i, j)$ и $(i, j + 1)$. Эти пары в четных и нечетных строках сдвинуты по отношению друг к другу (рис. 4). Такой способ отображения упрощает визуализацию процесса в клеточном массиве, так как обратное сжатие массива позволяет каждой клетке поставить в соответствие один пиксель на мониторе, интенсивность окраски которого соответствует исследуемой величине.

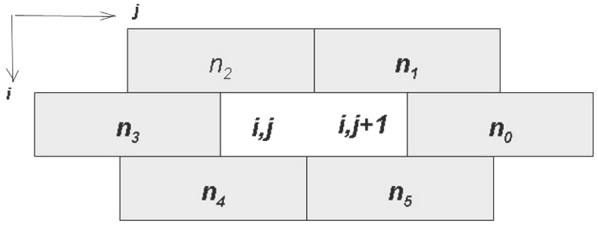


Рис. 4. Отображение гексагональной структуры на прямоугольную декартову решетку с растяжением по оси $j$. Клетка, содержащая два пикселя с координатами $(i, j)$ и $(i, j+1)$, и ее шесть соседей

*Множество функций переходов* $\Theta = \Theta_{\text{сдв}} \cup \Theta_{\text{ст}}$ задаёт функционирование КА, изменяя состояние каждой клетки в зависимости от состояний ее соседей. Переход клетки в новое состояние происходит за две фазы: фазу сдвига и фазу столкновения, соответственно $\Delta t = \Delta t_{\text{сдв}} + \Delta t_{\text{ст}}$. Сдвиг означает перемещение частицы из клетки с именем $x$ в соседнюю клетку $n_k(x)$, расположенную по направлению вектора ее скорости. Таким образом, каждая пара компонент вектора состояний после фазы сдвига, т. е. в момент $t + \Delta t_{\text{сдв}}$, равна

$$(ss)_k(x, t + \Delta t_{\text{сдв}}) = (ss)_k(n_{(k+3) \bmod 6}(x), t), \quad k = 0, \ldots, 5.$$

Столкновение означает изменение состояния клетки с определенной вероятностью. В функции столкновения аргументами являются только компоненты состояния самой клетки, т. е.

$$s(x, t + \Delta t_{\text{ст}}) = \Theta_{\text{ст}}(s(x, t)).$$

Функции столкновения задаются соотношениями, имеющими вид подстановок, в которых каждому текущему состоянию ставится в соответствие несколько возможных новых, каждое из которых имеет свою вероятность. Если при столкновении химической реакции не происходит, то функция столкновения сохраняет массу и импульс в клетке, т. е.

$$\mathrm{m}(s(x, t + \Delta t_{\text{ст}})) = \mathrm{m}(s(x, t)), \quad \mathbf{p}(s(x, t + \Delta t_{\text{ст}})) = \mathbf{p}(s(x, t)). \tag{1}$$

Когда происходит химическая реакция, то законы сохранения соблюдаются для системы «в целом», так как полученные частицы карбидов из клеток удаляются, переходя в формирующуюся кумулятивную струю.

Применение функций сдвига и столкновения ко всем клеткам клеточного массива называется *итерацией*. Итерация переводит $\Omega(t)$ в новое глобальное состояние $\Omega(t + \Delta t)$. Последовательность $\Gamma = \Omega(0), \ldots, \Omega(t), \ldots, \Omega(\hat{T})$ называется *эволюцией* КА.

Если существует отображение $\zeta : \Gamma \to F(\mathbf{y}, t')$, то говорят, что КА моделирует процесс, который может быть представлен пространственно-временной функцией $F(\mathbf{y}, t')$, где $\mathbf{y}$ — вектор координат в непрерывном пространстве, а $t'$ — непрерывное время. Поскольку состояния клеток являются булевыми векторами, получение отображения $\zeta$ требует осреднения модельных величин плотности (общей массы частиц в клетке) и скорости (средней величины и направления вектора скорости частиц). Осреднение выполняется по выбранной окрестности осреднения $\mathrm{Av}(x)$, характеризуемой радиусом осреднения $r \ll N$:

$$\langle \rho(x) \rangle = \frac{1}{|\mathrm{Av}(x)|} \sum_{x_i \in \mathrm{Av}(x)} \mathrm{m}(s(x_i)), \quad \langle \mathbf{u}(x) \rangle = \frac{1}{|\mathrm{Av}(x)|} \sum_{x_i \in \mathrm{Av}(x)} \mathbf{u}(s(x_i)), \quad x \in X.$$

## 2. Алгоритм моделирования

Компьютерное моделирование процессов в порошковой смеси для конкретного случая схлопывания конусной облицовки при воздействии взрыва (см. рис. 1) начинается с построения алгоритма функционирования КА, который затем легко преобразуется в программу. Процедура построения алгоритма состоит из следующих этапов:

1) определение параметров КА и его исходного состояния;
2) составление энергетического баланса событий и, на его основе, вычисление вероятностей выполнения функций переходов;
3) написание функций переходов КА.

*Параметры КА и его исходное состояние* определяются физическими данными объекта моделирования. Область моделирования вписывается в квадрат $40 \times 40$ мм$^2$; диаметр внешнего конуса в плоскости разреза — 28 мм, внутреннего — 22 мм. Размер частицы в порошке равен $10^3$ мкм$^3$, пористость порошка — 0,5. Масса гранулы вольфрама $\mathrm{M(W)} = 19,3 \cdot 10^{-12}$ г, углерода $\mathrm{M(C)} = 2{,}2 \cdot 10^{-12}$ г. С учетом этих данных, а также возможностей персонального компьютера размеры клеточного пространства выбраны следующим образом: $|X| = 400 \times 400$ гексагональных клеток, что потребовало массива размером $400 \times 800$ квадратов и соответственно столько же пикселей для визуализации процесса в реальном вычислительном времени.

Клеточный массив разделен концентрическими окружностями на три области: $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$. Область $\Omega_1$ находится между внутренней стенкой камеры и внешней стенкой конуса с радиусом $R_2 = 140$. Она соответствует пространству, в котором находится взрывчатое вещество. Это пространство не входит в область моделирования, т. е. в него частицы типа C и W не проникают. Область $\Omega_2$ между окружностями с радиусами $R_2 = 140$ и $R_3 = 110$ соответствует порошковой облицовке, где находится смесь частиц вольфрама и углерода в соотношении 1:1 по объёму. Исходное состояние этой области получается путем заполнения клеток случайными состояниями таким образом, чтобы осредненная плотность при одинаковом количестве частиц W и C была равна $\langle \rho_0 \rangle = 0{,}5$. Область $\Omega_3$, которая находится во внутреннем круге, заполнена условным «газом». Газ имитируется пространством с частицами углерода, плотность которых примерно в 100 раз меньше, чем плотность порошковой смеси: $\langle \rho_{\mathrm{gas}} \rangle = 0{,}17$. Такая имитация газа позволяет не вводить ещё один тип частиц, принимая допущение, что наличие дополнительного малого числа частиц углерода не испортит картины образования карбида, а только дополнит плотность углерода, который в любом случае избыточен и не является измеряемым количеством.

*Энергетический баланс* предполагает вычисление соотношения производимой и расходуемой энергий при столкновениях частиц. Поскольку нас интересует количество

полученных карбидов, то баланс рассчитывается для сталкивающихся частиц C и W при выполнения одной из реакций $r_i \in \mathrm{Reac}$, где $\mathrm{Reac} = \{r_1, r_2\}$; $r_1 = \mathrm{W} + \mathrm{C} \to \mathrm{WC}$; $r_2 = \mathrm{W} + \mathrm{W} + \mathrm{C} \to \mathrm{W_2C}$.

Реакция $r_i \in \mathrm{Reac}$ может произойти в клетке с состоянием $s$, если выполняется следующее условие:

$$Q(r_i)E(s) > E_a(r_i), \quad r_i \in \mathrm{Reac}, \quad s \in A,$$

где $E(s)$ — кинетическая энергия столкновения частиц реактантов в клетке с состоянием $s$; $E_a(r_i)$ — энергия активации реакции $r_i$; $Q(r_i)$ — так называемый *стерический коэффициент* реакции $r_i$, который в теории столкновений [7] определяется как отношение кинетической энергии сталкивающихся частиц к той энергии, которая на самом деле тратится на реакцию. Этот коэффициент всегда меньше единицы и зависит от формы и размеров частиц и свойств реактантов. Его величина известна (экспериментально измерена) для некоторых реакций в газах и жидкостях. Для нашего случая она не известна и является одним из тех параметров модели, которые должны быть установлены в процессе ее отладки. Вывод, который может быть сделан на основе литературных данных [7], позволяет предположить, что $Q(r) = 0{,}5 \div 0{,}6$.

Кинетическая энергия столкновения $E(s)$ модельных частиц C и W, вступающих в реакцию, рассчитывается по известным формулам столкновения упругих шаров с разными массами:

$$E(s) = \left(\mathrm{M(W)}v_{\mathrm{W}}^2 + \mathrm{M(C)}v_{\mathrm{C}}^2\right)/2,$$

где

$$v_{\mathrm{W}} = v\frac{\mathrm{M(C)}}{\mathrm{M(C)} + \mathrm{M(W)}}, \quad v_{\mathrm{C}} = v\frac{\mathrm{M(W)}}{\mathrm{M(C)} + \mathrm{M(W)}} \quad —$$

относительные скорости частиц, движущихся в системе, соответствующей состоянию клетки; $v$ — скорость движения частиц до столкновения. Максимальное значение $v_0$ скорость имеет в момент взрыва при $t = 0$. Значение $v_0$ вычисляется, согласно [8], по следующей эмпирической формуле:

$$v_0 = D\frac{3r}{4r + 9} = 2{,}2 \cdot 10^3\,\text{м/с}, \tag{2}$$

где $D = 7{,}5 \cdot 10^3\,\text{м/с}$ — скорость детонации; $r = (\rho_{\text{ВВ}}\sigma_{\text{ВВ}})/(\rho_{\text{об}}\sigma_{\text{об}})$; $\rho_{\text{ВВ}} = 1{,}65\,\text{г/см}^3$ и $\rho_{\text{об}} = 9{,}15\,\text{г/см}^3$ — плотности взрывчатых веществ и порошковой смеси с учетом пористости; $\sigma_{\text{ВВ}} = 2{,}6\,\text{см}$, $\sigma_{\text{об}} = 0{,}3\,\text{см}$ — физические размеры ширины колец с взрывчатым веществом и порошком соответственно.

Значение кинетической энергии столкновения двух частиц «в лоб» при скоростях их движения, равных $v_0$, является максимальным:

$$E_{\max} = \frac{1}{2}v_0^2\frac{\mathrm{M(W)\ M(C)}}{\mathrm{M(W)} + \mathrm{M(C)}} = 3{,}96 \cdot 10^{-9}\,\text{Дж}. \tag{3}$$

Поскольку скорость направлена перпендикулярно к образующей конуса (т. е. под углом $\alpha$ к плоскости моделирования) и модельные скорости частиц в клетке с состоянием $s$ находятся под углом $\beta(s)$, то при расчете скорости следует ввести коэффициент

$$S(\alpha, \beta) = \cos\alpha\cos\beta(s).$$

Более того, при каждом столкновении происходит потеря скорости, связанная с тем, что частицы не являются абсолютно упругими. Этот фактор учитывается в механике

*коэффициентом реституции* COR (Coefficient Of Restitution [9]), который для исследуемого случая не известен и тоже является настраиваемым параметром. Поскольку COR следует учитывать при каждом столкновении, то влияние его накапливается со временем, т. е. на $t$-й итерации кинетическая энергия столкновения равна

$$E(s,t) = E_{\max}(s) \cdot S^2 \cdot \mathrm{COR}^t.$$

Энергия активации реакций $E_a(r_i)$ при столкновениях частиц определяется исходя из известных значений $E_a(\mathrm{WC}) = 35{,}2$ кДж/моль и $E_a(\mathrm{W_2C}) = 47{,}2$ кДж/моль и составляет для получения одной частицы WC или $\mathrm{W_2C}$ энергию, равную

$$E_a(\mathrm{WC}) = 1{,}4 \cdot 10^{-9}\,\text{Дж/частица}, \quad E_a(\mathrm{W_2C}) = 1{,}135 \cdot 10^{-9}\,\text{Дж/частица}. \tag{4}$$

Вероятность выполнения реакции $r_i \in \mathrm{Reac}$ определяется величиной

$$\mathrm{P}_{r_i}(s,t) = \frac{Q(r_i)E(s,t) - E_a(r_i)}{Q(r_i)E(s,t)}. \tag{5}$$

*Функции переходов* задаются выражениями, имеющими вид подстановок, в которых текущему состоянию ставится в соответствие несколько возможных новых, каждое из которых имеет свою вероятность. Поскольку число возможных состояний равно $3^6$, прямым перебором их составить чрезвычайно сложно. Приходится прибегать к некоторым приемам сокращения этой работы. Во-первых, учитываются только те переходы в новые состояния, которые удовлетворяют законам сохранения массы и импульса (1). Во-вторых, симметричная структура клеточного пространства позволяет написать функции только для одной конфигурации распределения частиц по направлениям, а остальные вычислять, используя симметрию гексагонов по формулам поворота, или, что то же самое, циклического сдвига компонентов вектора состояний.

Множество функций столкновения делится на две группы $\Theta_{\text{ст}} = \Theta_{\text{дв}} \cup \Theta_{\text{реак}}$. Первая группа содержит правила, ответственные за движение. В них участвуют частицы только одного типа. Очевидно, что в этих случаях никакая химическая реакция не возможна. Частицы только меняют направления своего движения, функции переходов из этой группы соответствуют классической FHP-модели [3] и показаны на рис. 1.

Вторую, наибольшую по составу группу функций столкновений $\Theta_{\text{реак}}$ составляют те, в которых возможны столкновения частиц типа C с частицами типа W и, следовательно, могут происходить химические реакции. К этой группе относятся переходы из состояний $s \in A_{\text{реак}}$, $A_{\text{реак}} \subseteq A$, содержащих две инверсные друг другу пары $(s,s)_k$ и $(s,s)_{(k+h) \bmod 6}$:

$$(ss)_k = \overline{ss}_{(k+h) \bmod 6}, \quad h \in \{2,3,4\}, \quad k = 0,1,\ldots,5,$$

что соответствует столкновению частиц W и C при углах $\beta = 120$ и $180°$. Применение функции столкновения приводит к образованию одной частицы WC или половины частицы $\mathrm{W_2C}$ и замене в векторе состояния $s$ компонент $(s,s)_k$ и $(s,s)_{(k+h) \bmod 6}$ на $(0,0)$. При этом счетчики образующихся карбидов увеличиваются на соответствующую величину.

Предполагается, что импульс, действующий на внешнюю оболочку, происходит мгновенно при $t = 0$. Это событие выражается только в том, что частицы в клеточном пространстве $\Omega_2 \cup \Omega_3$ становятся активными. Далее все клетки действуют в соответствии со следующим итерационным алгоритмом.

Счетчики частиц WC, W и $W_2C$ устанавливаются в 0. На каждом итерационном шаге $t$ над клеточным массивом $\Omega(t)$ выполняются следующие действия.

1. Для всех клеток $(s,x) \in \Omega_2(t) \cup \Omega_3(t)$ вычисляются новые значения $(ss)_k$, $k = 0,\ldots,5$, компонент вектора $s$, и все клетки меняют состояния на полученные значения, переводя $\Omega_2(t) \cup \Omega_3(t)$ в новое состояние $\Omega_2(t + \Delta t_{\text{сдв}}) \cup \Omega_3(t + \Delta t_{\text{сдв}})$.

2. Для каждой клетки $(s,x) \in \Omega_2(t + \Delta t_{\text{сдв}}) \cup \Omega_3(t + \Delta t_{\text{сдв}})$ с состоянием $s(x,t) \in$ $\in A_{\text{реак}}$ вычисляется функция столкновения $\Theta_{\text{реак}}$ с вероятностью (5), и счетчик соответствующих частиц увеличивается.

3. Для каждой клетки $(s,x) \in \Omega_2(t + \Delta t_{\text{сдв}}) \cup \Omega_3(t + \Delta t_{\text{сдв}})$ с состоянием $s(x,t) \notin$ $\notin A_{\text{реак}}$ вычисляется значение функции столкновения $\Theta_{\text{дв}}$.

4. Во всех клетках из $\Omega_2(t + \Delta t_{\text{сдв}}) \cup \Omega_3(t + \Delta t_{\text{сдв}})$ производится замена состояний на вычисленные в пп. 2 и 3.

5. Значение $t$ увеличивается на 1, и переход к п. 1.

Работа алгоритма заканчивается, когда значения счетчиков перестают изменяться. По значениям счетчиков легко определить количества полученных карбидов.

## 3. Результаты моделирования

Компьютерное моделирование производилось для пяти значений конусных углов: $\alpha_1 = 15°$, $\alpha_2 = 22{,}5°$, $\alpha_3 = 30°$, $\alpha_4 = 37{,}5°$, $\alpha_5 = 45°$. Для всех случаев исходные состояния различались только углом $\alpha$. Значения начальной скорости (2), начальной кинетической энергии столкновения (3), энергии активации (4) для всех случаев приняты одинаковыми.

Реализация программы проводилась многократно для каждого случая с различными значениями стерического коэффициента $Q(r_i)$ и коэффициента реституции COR: $Q(r_i)$ изменялся от 0,45 до 0,55, COR — от 0,88 до 0,97 для обеих реакций. Сопоставление результатов моделирования с результатами натурных испытаний [10] позволило определить значения этих коэффициентов (табл. 1).

Таблица 1

**Полученные в результате моделирования значения коэффициентов $Q$ и COR**

| $\alpha$ | $Q(\text{W}+\text{C})$ | $Q(\text{W}+\text{W}+\text{C})$ | COR |
|---|---|---|---|
| 15°, 22,5° | 0,64 | 0,704 | 0,87 |
| 30°, 37,5° | 0,51 | 0,56 | 0,96 |

Зависимости количества получаемых частиц карбидов от времени в процессе моделирования показывают (рис. 5), что химические реакции заканчиваются за 30 итераций, когда частицы еще находятся в начале пути к центру. Зависимости количества частиц полученных карбидов от конусного угла даны на рис. 6. В табл. 2 приведены результаты моделирования при значениях коэффициентов из табл. 1.

Сравнение полученных путем моделирования и путем физических испытаний значений $\text{WC}/\text{W}(T)$ и $\text{W}_2\text{C}/\text{W}(T)$ в табл. 2 и 3 показывает, что максимальная разница равна $\approx 15\,\%$, а средняя ошибка не превышает $2\,\%$.

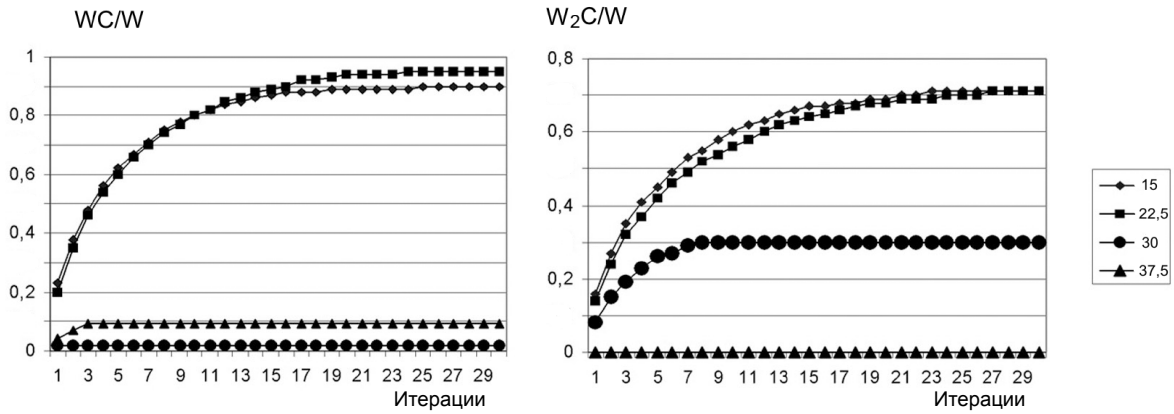


Рис. 5. Количества образующихся веществ по отношению к количеству остающегося вольфрама в зависимости от времени

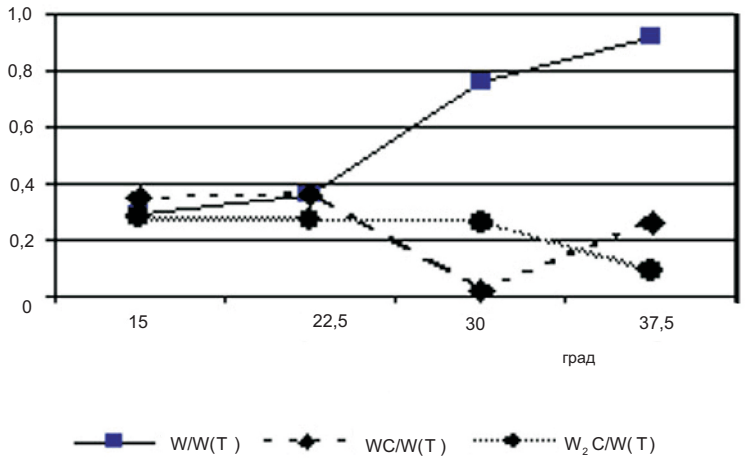


Рис. 6. Зависимости количества образовавшихся карбидов от конусного угла

Таблица 2

**Полученные в результате моделирования значения веществ в отношении к исходному (W(0)) и оставшемуся (W($T$)) количеству вольфрама**

| $\alpha$ | W($T$)/W(0) | WC($T$)/W(0) | $W_2C$/W(0) | WC/W($T$) | $W_2C$/W($T$) |
|---|---|---|---|---|---|
| 15° | 0,484 | 0,344 | 0,273 | 0,896 | 0,71 |
| 22,5° | 0,362 | 0,358 | 0,267 | 0,95 | 0,713 |
| 30° | 0,757 | 0,018 | 0,259 | 0,025 | 0,298 |
| 37,5° | 0,917 | 0,0 | 0,086 | 0,025 | 0,092 |
| 45° | 0,999 | 0,0 | 0,0 | 0,0 | 0,0 |

Таблица 3

**Полученные в результате физического испытания значения веществ в отношении к полученному количеству вольфрама $\mathbf{W}(T)$**

| $\alpha$ | $\mathrm{WC}/\mathrm{W}(T)$ | $\mathrm{W_2C}/\mathrm{W}(T)$ |
|---|---|---|
| 15° | $\approx 0{,}9$ | $\approx 0{,}6$ |
| 22,5° | $\approx 1{,}1$ | $\approx 0{,}75$ |
| 30° | $\approx 0{,}03$ | $\approx 0{,}225$ |
| 37,5° | $< 0{,}025$ | $< 0{,}1$ |
| 45° | 0,0 | 0,0 |

## Заключение

Представленная КА-модель сложного физико-химического процесса в порошковой среде является новой модификацией КА-моделей решеточного газа, в которой впервые использованы правила переходов, моделирующие не только движение потока порошковых частиц, но и химические взаимодействия. Применение КА-модели для исследования процесса, происходящего в облицовке из порошковой смеси при воздействии взрыва, является частью работ, проводимых в рамках Междисциплинарного интеграционного проекта СО РАН «Динамика структурно-фазовых состояний и фундаментальные основы синтеза нанокомпозитов в кумулятивных потоках». Несмотря на существенные допущения (двумерность, приближенный расчет вероятностей), результаты компьютерного моделирования и сравнение их с результатами натурных испытаний позволяют сделать вывод о возможности и полезности КА-моделирования для определения параметров планируемых физических испытаний, а также о необходимости совершенствования модели и уточнения ее параметров.

Авторы благодарны Ирине Анатольевне Панкратовой за полезные замечания и предложения при подготовке статьи к печати.

## ЛИТЕРАТУРА

1. Simulating Complex Systems by Cellular Automata / eds. A. G. Hoekstra, J. Kroc, P. M. A. Sloot. Berlin: Springer, 2010. 384 p.
2. *Бандман О. Л.* Клеточно-автоматные модели пространственной динамики // Системная информатика. Новосибирск: СО РАН, 2006. Вып. 10. С. 59–113.
3. *Frish U., d'Humiéres D., Hasslacher B., et al.* Lattice-Gas hydrogynamics in two and three dimensions // Complex Systems. 1987. V. 1. P. 49–707.
4. *Лаврентьев М. А.* Кумулятивный заряд и принципы его работы // Успехи математических наук. 1957. Т. XII. Вып. 4(76). С. 41–56.
5. *Громилов С. А., Кинеловский С. А., Попов Ю. Н., Тришин Ю. А.* О возможности физико-химических превращений веществ при кумулятивном нанесении покрытий // ФГВ. 1997. Т. 33. № 6. С. 127–130.
6. *Кинеловский С. А., Громилов С. А.* Особенности образования кристаллических фаз системы W–C–N в кумулятивном процессе // ФГВ. 2001. Т. 37. № 2. С. 135–139.
7. *Berdnikov V. M. and Doktorov A. B.* Steric factor diffusion-controlled chemical reactions // Chem. Phys. 1982. V. 69. No. 1–2. P. 205–212.
8. *Кинеловский С. А.* О неодномерном метании и соударении пластин // Сб. трудов VI Междунар. симпозиума «Использование энергии взрыва для производства металлических материалов с новыми свойствами». Готвальдов, ЧССР, 1985. С. 101–108.

9. Coefficients of Restitution The Physics Factbook. 2006. `http://hypertextbook.com/facts/2006/restitution.shtml`
10. *Кедринский В. К., Кинеловский С. А., Кузавов В. Т., Корчагин М. А.* Экспериментальное исследование синтеза соединений и фаз в кумулятивных течениях // Отчет по Интеграционному проекту 29 «Разработка научных основ кумулятивного синтеза новых наноструктурных соединений и покрытий методом встречных пучков и мишений». Гл. 1. Новосибирск: ИГ СО РАН, 2009.

# СВЕДЕНИЯ ОБ АВТОРАХ

**АБРОСИМОВ Михаил Борисович** — доцент, кандидат физико-математических наук, доцент Саратовского государственного университета им. Н. Г. Чернышевского, г. Саратов. E-mail: **mic@rambler.ru**

**АЛЕКСЕЕВ Евгений Константинович** — аспирант Московского государственного университета им. М. В. Ломоносова, г. Москва. E-mail: **geni-cmc@mail.ru**

**БАНДМАН Ольга Леонидовна** — профессор, доктор технических наук, главный научный сотрудник Института вычислительной математики и математической геофизики СО РАН, г. Новосибирск.
E-mail: **bandman@ssd.sscc.ru, bandman@academ.org**

**БЕРЛИНКОВ Михаил Владимирович** — аспирант кафедры алгебры и дискретной математики Уральского государственного университета, г. Екатеринбург. E-mail: **berlm@mail.ru**

**БЫКОВА Валентина Владимировна** — доцент, кандидат технических наук, профессор Института математики Сибирского федерального университета, г. Красноярск. E-mail: **bykvalen@mail.ru**

**ВЛАСОВА Анастасия Владимировна** — аспирантка Саратовского государственного университета им. Н. Г. Чернышевского, г. Саратов.
E-mail: **VAnastasiyaV@gmail.com**

**ДЕВЯНИН Петр Николаевич** — доктор технических наук, доцент, заместитель заведующего кафедрой Института криптографии, связи и информатики, г. Москва. E-mail: **peter_devyanin@hotmail.com**

**КАРМАНОВА Евгения Олеговна** — аспирантка Саратовского государственного университета им. Н. Г. Чернышевского, г. Саратов. E-mail: **lkb@info.sgu.ru**

**КИНЕЛОВСКИЙ Сергей Анатольевич** — профессор, доктор физико-математических наук, ведущий научный сотрудник Института гидродинамики им. М. А. Лаврентьева, г. Новосибирск. E-mail: **skin@hydro.nsc.ru**

**САЛИЙ Вячеслав Николаевич** — профессор, кандидат физико-математических наук, заведующий кафедрой Саратовского государственного университета им. Н. Г. Чернышевского, г. Саратов. E-mail: **SaliiVN@info.sgu.ru**

**ФОМИЧЁВ Владимир Михайлович** — старший научный сотрудник, доцент, доктор физико-математических наук, ведущий научный сотрудник Учреждения Российской академии наук «Институт проблем информатики РАН», г. Москва. E-mail: **fomichev@nm.ru**

# АННОТАЦИИ СТАТЕЙ НА АНГЛИЙСКОМ ЯЗЫКЕ

*Alekseev E. K.* **ON SOME MEASURES OF NONLINEARITY FOR BOOLEAN FUNCTIONS.** A nonlinearity measure is defined for a Boolean function $f$ as a distance from $f$ to the set of algebraic degenerated functions. Relations between this measure and some early offered measures of the nonlinearity are considered. Also, we investigate the order of algebraic degeneration of those functions which are mostly close to $f$.
**Keywords:** *nonlinearity of Boolean functions, algebraic degenerated functions, linear structures space, cryptography.*

*Devyanin P. N.* **FORMATION OF A DICTIONARY OF TERMS IN THE THEORY OF MODELING THE SECURE ACCESS AND INFORMATION FLOWS CONTROL IN COMPUTER SYSTEMS.** The work presents a dictionary project for the theory of modeling the secure access and information flows control in computer systems. The purpose of the dictionary formation is to define the main notions and terms in the computer security for using in science and education. The dictionary is created on the basis of the terminology existing in standards, in the classical models and in the DP-models of the computer security.
**Keywords:** *computer security, access control, information flows, DP-model, dictionary.*

*Bykova V. V.* **FPT-ALGORITHMS AND THEIR CLASSIFICATION ON THE BASIS OF ELASTICITY.** We give a brief overview of the results and problems of parameterized algorithmics as the new direction of computational complexity theory. For a parameterized algorithm, we offer a new indicator of computational complexity which can be used to measure the growth rate of its complexity function depending on many variables. This indicator is a partial elasticity of the complexity function. We offer a two-dimensional classification of parameterized algorithms with the complexity function having a multiplicative form of presentation.
**Keywords:** *computation complexity, parameterized algorithms, elasticity of algorithms.*

*Berlinkov M. V.* **A NOTE ON POLYNOMIAL APPROXIMATION OF SYNCHRONIZING OPTIMAL COLORING.** A strongly connected aperiodic directed graph with constant out-degree is called admissible. An automaton $A$ is a coloring of admissible graph $G$ if the underlying graph of $A$ equals $G$. A word is called synchronizing for an automaton $A$ if it takes $A$ to a one particular state no matter of the starting state. Optimal coloring of admissible graph $G$ is a synchronizing coloring with shortest reset word among all synchronizing colorings of $G$. The length of the corresponding reset word is called optimal coloring value. We prove that unless $\mathcal{P} = \mathcal{NP}$, no polynomial-time algorithm approximates optimal coloring or optimal coloring value within a factor less than 2 in the 3-letter alphabet case. We also extend this result to the 2-letter alphabet case.
**Keywords:** *road coloring problem, optimal coloring, synchronizing automata.*

*Salii V. N.* **SKELETON AUTOMATA.** A skeleton automaton is an automaton in which the relation of mutual accessibility of states is the identity relation. We prove that automata that admit a regular enumeration of states are exactly skeleton automata. It is shown how for a given automaton one can construct an automaton with minimal number of states that

has the same subautomata lattice, and is necessarily a skeleton automaton. A procedure is proposed to obtain a skeleton automaton from a given automaton by removal of minimal number of arcs in its transition diagram.
**Keywords:** *automaton, strongly connected automaton, skeleton automaton, regular enumeration of states, subautomata lattice.*

*Abrosimov M. B.* **MINIMAL EDGE EXTENSIONS OF ORIENTED AND DIRECTED STARS.** A complete description of all minimal edge $k$-extensions of oriented and directed stars is given.
**Keywords:** *star graph, minimal edge extension, fault tolerance.*

*Vlasova A. V.* **ATTRACTORS OF DYNAMICAL SYSTEMS ASSOCIATED WITH CYCLES.** We prove a theorem that describes attractors of dynamical systems associated with cycles. States of such a system are binary vectors of a given dimension, and evolutional function transforms vectors according to the following rules: if both the initial component is 0 and the final one is 1 they are replaced by 1 and 0 respectively and all digrams 10 are replaced simultaneously by 01.
**Keywords:** *attractor, dynamical system, evolutional function.*

*Karmanova E. O.* **ON CONGRUENCES OF PATHS.** A congruence of a path is an equivalence relation on the set of path's vertices all of whose classes are independent subsets. It is shown that each connected graph is a quotient-graph of a suitable path. Valuations are established for the minimal length of a chain whose quotient-graph is a given graph.
**Keywords:** *path, congruence, quotient-graph, tree, star.*

*Fomichev V. M.* **THE ESTIMATES OF EXPONENTS FOR PRIMITIVE GRAPHS.** The estimates of exponents of $n$-vertex primitive digraphs and undirected graphs are improved. The digraphs considered contain two prime contours with coprime lengths $l$ and $\lambda$. For them, accessible estimates of the order $\mathrm{O}(\max\{l\lambda, f(l,\lambda,n)\}$ are obtained, where $f(l,\lambda,n)$ is a linear polynomial. The exponent of undirected graph is no more $2n-l-1$, where $l$ is the length of the longest cycle with odd length in graph. Primitive digraphs with maximal exponent ($n^2-2n+2$, H. Wielandt, 1950) and undirected graphs with maximal exponent $(2n-2)$ are completely described.
**Keywords:** *graph exponent, primitive graphs.*

*Bandman O. L., Kinelovsky S. A.* **CUMULATIVE SYNTHESIS: A CELLULAR-AUTOMATA MODEL OF PHYSICAL-CHEMICAL PROCESSES ON THE STAGE OF POWDER REVETMENT COLLAPSING.** A discrete mathematical model is proposed which i s intended for computer simulation of physical-chemical processes in a powder mixture under the impact pressure produced by the explosion. The model is a two-dimensional "Lattice-Gas" cellular automaton with hexagonal structure of the discrete space in the plane perpendicular to the direction of emerging jet. The powder grains are simulated by "particles" of a Lattice-Gas model, having different mass for different powder substances and capable to enter in chemical interactions. Program implementation of cellular automaton evolution is done for five cases, differing by the angles between the front of impact pressure wave and the direction of emerging powder jet. Simulation aimed to investigate the production of tungsten carbides when the mixture of powdered tungsten and carbon are used. Relative amount of carbides WC and $W_2C$ obtained by simulation are compared to those produced in natural tests, which allowed to specify previously unknown cellular automata model coefficients.
**Keywords:** *cumulative synthesis, cellular automaton, Lattice-gas model, tungsten carbides.*

Журнал «Прикладная дискретная математика» включен в перечень ВАК рецензируемых российских журналов, в которых должны быть опубликованы основные результаты диссертаций, представляемых на соискание учёной степени кандидата и доктора наук, а также в перечень журналов, рекомендованных УМО в области информационной безопасности РФ в качестве учебной литературы по специальности «Компьютерная безопасность».

Журнал «Прикладная дискретная математика» распространяется по подписке; его подписной индекс 38696 в объединённом каталоге «Пресса России». Полнотекстовые электронные версии вышедших номеров журнала доступны на его сайте vestnik.tsu.ru/pdm и на Общероссийском математическом портале www.mathnet.ru. На сайте журнала можно найти также и правила подготовки рукописей статей в журнал.

**Тематика публикаций журнала:**

- *Теоретические основы прикладной дискретной математики*
- *Математические методы криптографии*
- *Математические методы стеганографии*
- *Математические основы компьютерной безопасности*
- *Математические основы надежности вычислительных и управляющих систем*
- *Прикладная теория кодирования*
- *Прикладная теория автоматов*
- *Прикладная теория графов*
- *Логическое проектирование дискретных автоматов*
- *Математические основы информатики и программирования*
- *Вычислительные методы в дискретной математике*
- *Дискретные модели реальных процессов*
- *Математические основы интеллектуальных систем*
- *Исторические очерки по дискретной математике и ее приложениям*