МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2022 **№ 55**

Зарегистрирован в Федеральной службе по надзору
в сфере связи и массовых коммуникаций

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

Подписной индекс в объединённом каталоге «Пресса России» 38696

УЧРЕДИТЕЛЬ
Томский государственный университет

РЕДАКЦИОННАЯ КОЛЛЕГИЯ ЖУРНАЛА
«ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА»

Черемушкин А. В., д-р физ.-мат. наук, чл.-корр. Академии криптографии РФ (главный редактор); Девянин П. Н., д-р техн. наук, чл.-корр. Академии криптографии РФ (зам. гл. редактора); Панкратова И. А., канд. физ.-мат. наук, доц. (отв. секретарь); Агиевич С. В., канд. физ.-мат. наук; Алексеев В. Б., д-р физ.-мат. наук, проф.; Евдокимов А. А., канд. физ.-мат. наук, проф.; Колесникова С. И., д-р техн. наук; Крылов П. А., д-р физ.-мат. наук, проф.; Логачев О. А., д-р физ.-мат. наук, доц.; Мясников А. Г., д-р физ.-мат. наук, проф.; Романьков В.А., д-р физ.-мат. наук, проф.; Сафонов К. В., д-р физ.-мат. наук, проф.; Фомичев В. М., д-р физ.-мат. наук, проф.; Харин Ю. С., д-р физ.-мат. наук, чл.-корр. НАН Беларуси; Чеботарев А. Н., д-р техн. наук, проф.; Шоломов Л. А., д-р физ.-мат. наук, проф.

Адрес редакции и издателя: 634050, г. Томск, пр. Ленина, 36
E-mail: pank@mail.tsu.ru

В журнале публикуются результаты фундаментальных и прикладных научных исследований отечественных и зарубежных ученых, включая студентов и аспирантов, в области дискретной математики и её приложений в криптографии, компьютерной безопасности, кибернетике, информатике, программировании, теории надёжности, интеллектуальных системах.

Периодичность выхода журнала: 4 номера в год.

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

Подписано к печати 14.03.2022. Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 15. Тираж 300 экз.
Заказ № 4935. Цена свободная. Дата выхода в свет 18.03.2022.

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

СОДЕРЖАНИЕ

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

УДК 511 DOI 10.17223/20710410/55/1

ПРИМЕНЕНИЕ МУЛЬТИГАРМОНИЧЕСКИХ ЧИСЕЛ ДЛЯ СИНТЕЗА ЗАМКНУТЫХ ФОРМ ПАРАМЕТРИЧЕСКИ МОДИФИЦИРОВАННЫХ ФАКТОРИАЛ-ПРОИЗВОДЯЩИХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

И. В. Стаценко

Московский энергетический институт, г. Москва, Россия

E-mail: iwsta@yandex.ru

Представлены инструментарий и метод приведения к замкнутому виду последовательностей целых чисел, относящихся к классу факториал-производящих рекурсий. Определены признаки и свойства модифицированной факториал-производящей рекурсии одной и двух переменных. Наиболее известной факториал-производящей рекурсией двух переменных является последовательность чисел Стирлинга первого рода. Для синтеза аналитической модели рекурсии применяются модифицированные гипергармонические числа. Выявлены преимущества данных чисел для построения замкнутых форм факториал-производящих рекурсий. Синтезирована неполная замкнутая форма последовательности чисел Стирлинга первого рода.

Ключевые слова: *замкнутые формы рекуррентных уравнений с нелинейными коэффициентами, интерполяция рекуррентных последовательностей, производящие функции рекурсий, факториал-производящие последовательности, гипергармонические числа, мультигармонические числа, числа Стирлинга первого рода.*

APPLICATION OF MULTIHARMONIC NUMBERS FOR THE SYNTHESIS OF CLOSED FORMS OF PARAMETRICALLY MODIFIED FACTORIAL GENERATING SEQUENCES

I. V. Statsenko

Moscow Power Engineering Institute, Moscow, Russia

In this paper, using numbers of a special kind $H_n^{(r)} = \sum\limits_{m=r}^{n} \ldots \sum\limits_{l=3}^{s-1} \sum\limits_{j=2}^{l-1} \sum\limits_{i=1}^{j-1} \dfrac{1}{ijl\ldots m}$, $r, n \in \mathbb{N}$, called multiharmonic numbers, incomplete closed forms of two fundamental sequences of integers given as a recursion are synthesized. The first recursion $u_{k+1}^{(m)} = (k+m)[2u_k^{(m)} - (k-1)u_{k-1}^{(m)}]$, $u_k \in \mathbb{Z}$, $k \in \mathbb{N}$, $m \in \mathbb{Z}^+$, under the conditions $m = 0$, $u_0^{(0)} = u_1^{(0)} = 1$ is factorial-generating: $u_k^{(0)} = k!$. The second recursion defines a sequence of Stirling numbers of the first kind $s(n,k)$, $n, k \in \mathbb{Z}^+$, and by the property $|s(n,1)| = (n-1)!$ is also factorial-generating. The resulting closed form for the first recursion is $u_k^{(m)} = \sum\limits_{i=0}^{k-1} \mathrm{C}_{k-1}^{i} \mathrm{A}_{k+m-1}^{k-i} m^{i-1}$, $k, m \in \mathbb{N}$,

$\mathrm{A}_n^m = n!/(n-m)!$, $\mathrm{C}_n^m = n!/(n-m)!m!$. The closed form for the second recursion is $s(n,k) = H_{n-1}^{(k-1)}(n-1)!(-1)^{n+k}$, $k, n \in \mathbb{N}$. Closed forms are not complete, since they are not used for cases: $m = k = 0$, $n = k = 0$.

Keywords: *closed forms of recurrent equations with nonlinear coefficients, interpolation of recurrent sequences, generating recursion functions, factorial-generating sequences, hyperharmonic numbers, multiharmonic numbers, Stirling numbers of the first kind.*

Введение

Вопросам применения различного инструментария для анализа свойств рекуррентных последовательностей посвящено большое количество публикаций, в частности можно отметить [1–6]. Как правило, анализ возможности представления неизвестной рекуррентной последовательности в замкнутой форме (в виде формулы для общего члена последовательности) начинается с построения производящей функции, что по целому ряду причин не всегда приводит к положительному результату. Далее применяются различные инструменты анализа последовательностей: интерполяция, экстраполяция, полиномы Бернулли, Case-средства, специальные числа.

Для «замыкания» рекурсий с постоянными коэффициентами разработан и успешно применяется математический аппарат производящих функций. В то же время уникальность задач, связанных с анализом целочисленных рекуррентных последовательностей с нелинейными коэффициентами, в большинстве случаев требует разработки индивидуальных методик приведения данных рекурсий к замкнутым формам. Актуальность таких задач связана, в первую очередь, с необходимостью приведения к замкнутому виду фундаментальных рекуррентных целочисленных последовательностей. Необходимо отметить, что в настоящее время не известны приемлемые для практического использования замкнутые формы последовательностей чисел Бернулли и чисел Стирлинга первого рода.

В данной работе для анализа целочисленной последовательности применяется аппарат модифицированных гипергармонических чисел [7–10]. На основе обобщения взаимосвязей в факториал-производящих рекурсиях получен вариант замкнутой формы для последовательности чисел Стирлинга первого рода.

1. Постановка задачи

Рассмотрим рекуррентное уравнение вида

$$u_{k+1}^{(m)} = (k+m)\left[2u_k^{(m)} - (k-1)u_{k-1}^{(m)}\right], \quad u_k \in \mathbb{Z}, \tag{1}$$

где $k \in \mathbb{N}$; $m \in \mathbb{Z}^+$. Данное уравнение задаёт рекурсию квазифибоначчиевого типа, так как связывает очередной член последовательности с двумя предыдущими на множестве целых чисел, но при этом коэффициенты рекурсии — нелинейные функции, зависящие от k. Наличие нелинейных коэффициентов существенно усложняет поиск замкнутых форм. Величина m выступает в качестве параметра, и свойство данной рекурсии при $m = 0$ определяет основное её название.

Рекурсия (1) имеет следующие замечательные свойства:

1. При $m = 0$ и $u_0^{(0)} = u_1^{(0)} = 1$ имеем замкнутую форму в виде

$$u_k^{(0)} = k!. \tag{2}$$

По этому свойству при $m = 0$ рекурсию (1) будем называть факториал-производящей. При других значениях параметра m данную рекурсию будем называть параметрически модифицированной факториал-производящей.

Доказательство проведём непосредственной подстановкой замкнутой формы (2) в рекурсию: $(k+1)! = k[2k! - (k-1)(k-1)!]$. Отсюда получаем $(k+1)! \equiv (k+1)!$.

2. Для любых $k \in \mathbb{N}$, $m \in \mathbb{Z}^+$ значения формулы (1) не зависят от $u_0^{(m)}$.

3. Если начальные условия для рекурсии (1) ввести в виде $u_0^{(m)} \in \mathbb{Z}$, $u_1^{(m)} = 1$, а соответствующую замкнутую форму обозначить $u_k^{(m,1)}$, то для других вариантов выбора первого члена последовательности $u_1^{(m,2)} = 2$, $u_1^{(m,3)} = 3$, …, $u_1^{(m,s)} = s$ получим

$$u_k^{(m,s)} = s \cdot u_k^{(m,1)},$$

где $u_k^{(m,s)}$ — замкнутые формы рекурсии (1) с $u_1^{(m)} = s$, $s \in \mathbb{N}$.

4. Рекурсия (1) формирует нечетную функцию $u_k^{(m,s)} = -u_k^{(m,-s)}$ относительно величины s.

5. $\lim\limits_{k\to\infty} u_k^{(m,s)} = \infty$.

В таблице представлены некоторые варианты формирования рекурсии $u_k^{(m,1)}$.

k	1	2	3	4	5	6	7
$u_k^{(0,1)}$	1	2	6	24	120	720	5040
$u_k^{(1,1)}$	1	4	21	136	1045	9276	93289
$u_k^{(2,1)}$	1	6	44	380	3768	42112	523072
$u_k^{(3,1)}$	1	8	75	804	9681	129168	1889379

Поставим задачу найти замкнутую форму $u_k^{(m,1)}$ при $k \in \mathbb{N}$, $m \in \mathbb{Z}^+$ для рекурсии (1). Учитывая свойства 1–3, замкнутая форма для случая $m = 0$ известна, а замкнутые формы для $u_k^{(m,s)}$, $s = 2, 3, \ldots$, вычисляются с использованием $u_k^{(m,1)}$.

С целью упрощения обозначений все замкнутые формы будем искать далее для $u_1^{(m,1)} = 1$ и обозначать в виде $u_k^{(m)}$.

2. Анализ возможности применения производящей функции

На базе (1) составим уравнение для получения производящей функции в виде

$$\sum_{k=1}^{\infty} \frac{u_{k+1}^{(m)}}{k+m} z^k = \left[2 \sum_{k=1}^{\infty} u_k^{(m)} z^k - (k-1) \sum_{k=1}^{\infty} u_{k-1}^{(m)} z^k \right]. \tag{3}$$

Пусть

$$F_1(z) = \sum_{k=1}^{\infty} u_k^{(m)} z^k, \quad F_2(z) = \sum_{k=1}^{\infty} (k-1) u_{(k-1)}^{(m)} z^k, \quad F_3(z) = \sum_{k=1}^{\infty} \frac{u_{k+1}^{(m)} z^k}{k+m}, \quad z \in \mathbb{R}.$$

В новых обозначениях уравнение (3) приобретает вид

$$F_3(z) = 2F_1(z) - F_2(z). \tag{4}$$

Функцию $F_1(z)$ будем полагать производящей для последовательности $u_1^{(m)}$, $u_2^{(m)}$, … Выразим функции $F_2(z)$ и $F_3(z)$ через $F_1(z)$.

Можно получить следующие соотношения для $F_2(z)$ и $F_3(z)$:

$$F_2 = F_1^{'} z^2; \tag{5}$$

$$F_3 = \frac{\int F_1 z^{m-2} dz}{z^m} - \frac{u_1^{(m)}}{m}. \tag{6}$$

Подставив (5) и (6) в (4), получим уравнение

$$\frac{\int F_1 z^{m-2} dz}{z^m} - \frac{u_1^{(m)}}{m} = 2F_1 - F_1^{'} z^2.$$

Отсюда получаем

$$F_1^{''} z^4 + F_1^{'}[(m+2)z^3 - 2z^2] + F_1(1 - 2mz) = u_1^{(m)} z. \tag{7}$$

Решением данного уравнения является искомая производящая функция $F_1(z)$, $z \in \mathbb{R}$, коэффициенты разложения которой в ряд Тейлора представляют решение исходного рекуррентного уравнения (1).

Недостатком данного метода в применении производящей функции является отсутствие решения дифференциального уравнения (7) в элементарных функциях, начиная со случая $m = 0$, когда уравнение приобретает наиболее простой вид

$$F_1^{''} z^4 + 2F_1^{'}(z^3 - z^2) + F_1 = u_1^{(0)} z.$$

3. Применение мультигармонических чисел для синтеза замкнутых форм факториал-производящих рекурсий одной переменной

В работах [7–10] представлены свойства и асимптотика гипергармонических чисел и модифицированных гипергармонических (мультигармонических) чисел. На их основе могут быть сформированы целочисленные последовательности, имеющие свойства рекурсий (1).

Процедура синтеза замкнутых форм с использованием мультигармонических чисел включает в себя три этапа:

1) подбор подходящей модели;
2) настройка модели (интерполяция);
3) проверка модели на удовлетворение условиям рекурсии (1).

П е р в ы й э т а п — подбор подходящей модели — процедура творческая и трудно формализуемая. Основным критерием здесь является удовлетворение модели скорости роста исходной последовательности (1) для фиксированного значения параметра m. В результате такого анализа подобрана модель следующего вида:

$$G_{m,k} = k!(k+m-1) \sum_{j=1}^{k} \sum_{i=0}^{k-1} (-1)^{i+j+1} m^{i-m+1} P_{m-2}(i) k^{j-1} \frac{H_i^{(j-1)}}{(i+1)!}. \tag{8}$$

Здесь $k \in \mathbb{N}$; $m = 2, 3, \ldots$, $G_{m,k}$ — модель замкнутой формы рекурсии $u_k^{(m)}$; $P_{m-2}(n)$ — многочлен с целыми коэффициентами $(m-2)$-й степени от целого аргумента n;

$$H_k^{(r)} = \sum_{n=r}^{k} \frac{H_{n-1}^{(r-1)}}{n} \tag{9}$$

— k-е мультигармоническое число r-го порядка; $r \in \mathbb{N}$; $H_n^{(0)} = 1$ для всех n;

$$H_n^{(1)} = \sum_{i=1}^{n} \frac{1}{i}; \quad H_n^{(2)} = \sum_{j=2}^{n} \sum_{i=1}^{j-1} \frac{1}{ij}; \quad H_n^{(3)} = \sum_{l=3}^{n} \sum_{j=2}^{l-1} \sum_{i=1}^{j-1} \frac{1}{ijl};$$

$H_n^{(r)} = 0$ при $n < r$.

В т о р о й э т а п — подбор коэффициентов многочлена (интерполяция) на $(m-2)$ первых узлах (точках последовательности).

Т р е т и й э т а п — проверка настроенной модели на удовлетворение условиям рекурсии (1). В случае неудовлетворения необходимо вернуться к первому этапу синтеза — отказу от модели (8) либо её модернизации модели.

В предложенной схеме отсутствует случай $m = 1$. Рассмотрим его отдельно, используя модель (8) и многочлен $P_0(n) \equiv 1$:

$$G_{1,k} = k! \sum_{j=1}^{k} \sum_{i=0}^{k-1} (-1)^{i+j+1} k^{j-1} \frac{H_i^{(j-1)}}{i!}, \quad k \in \mathbb{N}. \tag{10}$$

Модель (10) можно представить в виде

$$G_{1,k} = \sum_{i=0}^{k-1} \frac{k!}{i!} \sum_{j=1}^{k} (-1)^{i+j+1} k^{j-1} H_i^{j-1}, \quad k \in \mathbb{N}.$$

Отсюда с использованием свойства мультигармонических чисел

$$\sum_{j=1}^{k} (-1)^{i+j+1} k^{j-1} H_i^{(j-1)} = \mathrm{C}_{k-1}^{i} = \frac{(k-1)!}{i!(k-i-1)!}$$

получим

$$G_{1,k} = \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^{i} \mathrm{A}_k^{k-i} = u_k^{(1)}, \quad k \in \mathbb{N}, \tag{11}$$

где $\mathrm{A}_n^m = n!/(n-m)!$.

Второй этап (настройку модели) в данном случае пропускаем, так как используется многочлен $P_0(n) \equiv 1$.

Третий этап предусматривает проверку на удовлетворение модели (11) свойствам рекурсии (1) для случая $m = 1$, т. е.

$$G_{1,k+1} = (k+1)[2G_{1,k} - (k-1)G_{1,k-1}]. \tag{12}$$

Проверим эти условия:

$$G_{1,k} = \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^{i} \mathrm{A}_k^{k-i} = u_k^{(1)}, \quad k \in \mathbb{N}; \tag{13}$$

$$\frac{G_{1,k+1}}{k+1} = \frac{1}{k+1} \sum_{i=0}^{k} \mathrm{C}_k^i \mathrm{A}_{k+1}^{k+1-i} = \sum_{i=0}^{k-1} \mathrm{C}_k^i \mathrm{A}_k^{k-i} + 1; \tag{14}$$

$$G_{1,k-1}(k-1) = (k-1) \sum_{i=0}^{k-2} \mathrm{C}_{k-2}^{i} \mathrm{A}_{k-2}^{k-i-2} = \sum_{i=0}^{k-2} \frac{((k-1)!)^2}{(i!)^2} \frac{1}{(k-i-2)!}; \tag{15}$$

$$G_{1,k-1}(k-1) = \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^{i} \mathrm{A}_{k-1}^{i} (k-i-1), \quad k \in \mathbb{N}. \tag{16}$$

После подстановки (13)–(16) в рекурсию (12) получим

$$\sum_{i=0}^{k-1} \mathrm{C}_k^i \mathrm{A}_k^{k-i} \frac{(i^2+i-k)}{k^2} \equiv -1. \tag{17}$$

Для доказательства (17) преобразуем его левую часть:

$$\sum_{i=0}^{k-1} (\mathrm{C}_k^i \mathrm{A}_k^{k-i} \frac{i^2}{k^2} - \mathrm{C}_k^i \mathrm{A}_k^{k-i} \frac{(k-i)}{k^2}) = \sum_{i=1}^{k-1} \mathrm{C}_{k-1}^{i-1} \mathrm{A}_{k-1}^{k-i} - \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^i \mathrm{A}_{k-1}^{k-i-1} =$$
$$= \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^{i-1} \mathrm{A}_{k-1}^{k-i} - \sum_{i=0}^{k-2} \mathrm{C}_{k-1}^i \mathrm{A}_{k-1}^{k-i-1} - 1 = \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^{i-1} \mathrm{A}_{k-1}^{k-i} - \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^{i-1} \mathrm{A}_{k-1}^{k-i} - 1 \equiv -1.$$

Таким образом, функция вида (13) является замкнутой формой для факториал-производящей рекурсии для случая $m = 1$.

Дальнейшие исследования на основе модели (8) показали следующие результаты:

$$u_k^{(2)} = (k+1)! \sum_{j=1}^{k} \sum_{i=0}^{k-1} (-1)^{i+j+1} 2^{i-1} k^{j-1} \frac{H_i^{(j-1)}}{(i+1)!} = \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^i \mathrm{A}_{k+1}^{k-i} 2^{i-1}, \quad k \in \mathbb{N};$$

$$u_k^{(3)} = k!(k+2) \sum_{j=1}^{k} \sum_{i=0}^{k-1} (-1)^{i+j+1} 3^{i-2} [i+3] k^{j-1} \frac{H_i^{(j-1)}}{(i+1)!} = \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^i \mathrm{A}_{k+2}^{k-i} 3^{i-1}, \quad k \in \mathbb{N};$$

$$u_k^{(4)} = k!(k+3) \sum_{j=1}^{k} \sum_{i=0}^{k-1} (-1)^{i+j+1} 4^{i-3} [i^2+7i+16] k^{j-1} \frac{H_i^{(j-1)}}{(i+1)!} = \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^i \mathrm{A}_{k+3}^{k-i} 4^{i-1}, \quad k \in \mathbb{N}.$$

Обобщение всех замкнутых форм представлено формулой

$$u_k^{(m)} = \sum_{i=0}^{k-1} \mathrm{C}_{k-1}^i \mathrm{A}_{k+m-1}^{k-i} m^{i-1}, \quad k, m \in \mathbb{N}. \tag{18}$$

По аналогии с методикой, показанной для случая $m = 1$, можно доказать, что модель (18) отвечает условиям рекурсии (1).

В замкнутых формах $u_k^{(3)}$ и $u_k^{(4)}$ результаты второго этапа синтеза (интерполяция) представлены многочленами в квадратных скобках.

4. Применение мультигармонических чисел для синтеза замкнутых форм факториал-производящих рекурсий двух переменных

Рассмотрим рекурсию следующего вида:

$$s(n,k) = s(n-1,k-1) - (n-1)s(n-1,k), \tag{19}$$

$n \in \mathbb{Z}^+$; $k = 0, 1, 2, 3, \ldots, n-1$; $s(0,0) = 1$; $s(n,0) = 0$ для всех $n > 0$; $s(0,k) = 0$ для всех $k > 0$.

Данная рекурсия моделирует числа Стирлинга первого рода и принадлежит к числу факториал-производящих благодаря свойству

$$|s(n,1)| = (n-1)!, \quad n \in \mathbb{Z}^+.$$

Если переменную k в данной последовательности объявить вспомогательной (параметром), то в целом рекурсия $s(n,k)$ принадлежит к классу параметрически модифицированных факториал-производящих целочисленных последовательностей.

Применяя методику п. 3 для анализа данной последовательности, получим следующее тождество:

$$s(n,k) \equiv H_{n-1}^{(k-1)}(n-1)!(-1)^{n+k}, \tag{20}$$

$n \in \mathbb{N}$; $k = 1, 2, 3, \ldots, n-1$.

Учитывая, что мультигармонические числа (МГЧ) можно представить в двух формах — в замкнутой форме и в виде рекурсии, первый вид представления позволяет получить замкнутую форму для чисел Стирлинга первого рода. При этом форма будет неполной, так как в тождестве (36) отсутствуют случаи $k = 0$ и $n = 0$.

Доказательство (20) проведём непосредственной подстановкой (20) в (19) с использованием следующего свойства МГЧ [10]:

$$H_{n-1}^{(k-1)} = H_{n-2}^{(k-1)} + \frac{1}{n-1} H_{n-2}^{(k-2)}, \tag{21}$$

$n = 2, 3, \ldots$; $k = 2, 3, \ldots$ Имеем

$$s(n,k) = H_{n-1}^{(k-1)}(n-1)!(-1)^{n+k}; \tag{22}$$

$$s(n-1,k-1) = H_{n-2}^{(k-2)}(n-2)!(-1)^{n+k-2}; \tag{23}$$

$$(n-1)s(n-1,k) = H_{n-2}^{(k-1)}(n-1)!(-1)^{n+k-1}. \tag{24}$$

После подстановки (22)–(24) в (19) получим

$$H_{n-1}^{(k-1)}(n-1) = H_{n-2}^{(k-2)} + H_{n-2}^{(k-1)}(n-1).$$

Далее используем свойство МГЧ (21):

$$H_{n-2}^{(k-1)}(n-1) + H_{n-2}^{(k-2)} \equiv H_{n-2}^{(k-2)} + H_{n-2}^{(k-1)}(n-1).$$

Тождество (20) доказано.

Перечислим некоторые недостатки замкнутой формы (20):

1) при использовании МГЧ в виде

$$H_n^{(0)} \equiv 1, \quad H_n^{(1)} = \sum_{i=1}^{n} \frac{1}{i}, \quad H_n^{(2)} = \sum_{j=2}^{n} \sum_{i=1}^{j-1} \frac{1}{ij}, \quad H_n^{(3)} = \sum_{l=3}^{n} \sum_{j=2}^{l-1} \sum_{i=1}^{j-1} \frac{1}{ijl},$$
$$H_n^{(r)} = \sum_{m=r}^{n} \ldots \sum_{l=3}^{s-1} \sum_{j=2}^{l-1} \sum_{i=1}^{j-1} \frac{1}{ijl \ldots m}$$

   формула имеет громоздкий вид за счёт использования r знаков суммирования, что предполагает её применение при небольших значениях r;

2) при использовании МГЧ в виде

$$H_k^{(r)} = \sum_{n=r}^{k} \frac{H_{n-1}^{(r-1)}}{n}$$

   формула (20) имеет вид рекурсии (незамкнутая форма);

3) формула не применяется при $n = 0$ и $k = 0$.

Заключение

Таким образом, предложен новый инструментарий синтеза замкнутых форм факториал-производящих рекурсий — мультигармонические числа. Представлен метод применения МГЧ в указанных целях.

Необходимо отметить существенный недостаток метода — трудно формализуемый этап подбора исходной модели для дальнейшей интерполяции.

Синтезированы замкнутые формы факториал-производящих рекурсий одной и двух переменных. Актуальность проведённых исследований определяется одновременно фундаментальным и прикладным характером представленных рекурсий.

Очевидным недостатком полученной замкнутой формы чисел Стирлинга первого рода является её неполнота и громоздкость при увеличении значений одной из двух переменных. В то же время достаточно простой для алгоритмической реализации и дальнейшего анализа вид позволяет рассматривать данную замкнутую форму в качестве базовой для описания чисел Стирлинга не только первого, но и второго рода, а также связанных с ними специальных чисел. Необходимо отметить, что алгоритмическая простота представленной замкнутой формы чисел Стирлинга первого рода, по всей видимости, не позволяет считать её абсолютно новой. С другой стороны, представление данной замкнутой формы в рамках предварительно введённых специальных чисел (МГЧ) позволяет считать её описательно новой.

Непосредственный практический выход данной работы связан с ускорением работы алгоритмов решения некоторых частных вычислительных задач, а также возможностью дополнения онлайн-энциклопедии целочисленных последовательностей OEIS — базы замкнутых форм известных фундаментальных и параметрически связанных с ними целочисленных рекурсий.

ЛИТЕРАТУРА

1. *Варин В. П.* Факториальное преобразование некоторых классических комбинаторных последовательностей // Ж. вычисл. матем. и матем. физ. 2018. Т. 58. № 11. С. 1747–1770.
2. *Варин В. П.* Об интерполяции некоторых рекуррентных последовательностей // Ж. вычисл. матем. и матем. физ. 2021. Т. 61. № 6. С. 913–925.
3. *Варин В. П.* Комбинаторные преобразования последовательностей как ускорители сходимости степенных рядов // Теоретические основы конструирования численных алгоритмов решения задач математической физики. Тез. докл. XXII Всерос. конф., посвящённой памяти К. И. Бабенко (Абрау-Дюрсо, 3–8 сентября, 2018). М.: ИПМ им. М. В. Келдыша, 2018. С. 29.
4. *Геут К. Л., Титов С. С.* О понижении порядка линейных рекуррентных уравнений с постоянными коэффициентами // Прикладная дискретная математика. Приложение. 2017. № 10. С. 12–13.
5. *Геут К. Л., Титов С. С.* О простых числах и рекуррентных соотношениях // Актуальные проблемы прикладной математики и механики. Тез. докл. VII Всерос. конф., посвящённой памяти академика А. Ф. Сидорова (Абрау-Дюрсо, 15–21 сентября, 2014). Екатеринбург: УрО РАН, 2014. С. 20–21.
6. *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. Основание информатики. М.: Мир, 1998.
7. *Benjamin A. T., Gaebler D., and Gaebler R.* A combinatorial approach to hyperharmonic numbers // Electr. J. Combinat. Number Theory. 2003. V. 3.
8. *Conway D. H. and Guy R. K.* The Book of Numbers. N.Y.: Springer Verlag, 1996.

9. *Mezö I.* Some inequalities for hyperharmonic series // Adv. in Inequalities for Special Functions. Nova Science Publ. House, 2006. P. 121–125.
10. *Стаценко И. В.* Расширение свойств мультигармонических чисел // Точная наука. 2021. № 107. С. 2–4.

REFERENCES

1. *Varin V. P.* Faktorial'noe preobrazovanie nekotorykh klassicheskikh kombinatornykh posledovatel'nostey [Factorial transformation of some classical combinatorial sequences]. Zhurnal Vychislitel'noy Matematiki i Matematicheskoy Fiziki, 2018, vol. 58, no. 11, pp. 1747–1770. (in Russian)
2. *Varin V. P.* Ob interpolyatsii nekotorykh rekurrentnykh posledovatel'nostey [On the interpolation of some recurrent sequences]. Zhurnal Vychislitel'noy Matematiki i Matematicheskoy Fiziki, 2021, vol. 61, no. 6, pp. 913–925. (in Russian)
3. *Varin V. P.* Kombinatornye preobrazovaniya posledovatel'nostey kak uskoriteli skhodimosti stepennykh ryadov [Combinatorial transformations of sequences as accelerators of convergence of power series]. Teoreticheskie Osnovy Konstruirovaniya Chislennykh Algoritmov Resheniya Zadach Matematicheskoy Fiziki. Proc. XXII All-Rus. Conf. dedicated to the memory of K. I. Babenko (Abrau-Durso, 3–8 September, 2018), Moscow, IPM Publ., 2018, p. 29. (in Russian)
4. *Geut K. L. and Titov S. S.* O ponizhenii poryadka lineynykh rekurrentnykh uravneniy s postoyannymi koeffitsientami [On reducing the order of linear recurrence equations with constant coefficients]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2017, no. 10, pp. 12–13. (in Russian)
5. *Geut K. L. and Titov S. S.* O prostykh chislakh i rekurrentnykh sootnosheniyakh [On prime numbers and recurrent relations]. Aktual'nye Problemy Prikladnoy Matematiki i Mekhaniki. Proc. VII All-Rus. Conf. dedicated to the memory of A. F. Sidorov (Abrau-Durso, 15–21 September, 2014). Yekaterinburg, UB RAS, 2014, pp. 20–21. (in Russian)
6. *Graham R., Knuth D., and Patashnik O.* Concrete Mathematics. Addison-Wesley Publ., 1989.
7. *Benjamin A., Gebler D., and Gebler R.* A combinatorial approach to hyperharmonic numbers. lectr. J. Combinat. Number Theory, 2013, vol. 3.
8. *Conway D. H. and Guy R. K.* The Book of Numbers. N.Y., Springer Verlag, 1996.
9. *Mezö I.* Some inequalities for hyperharmonic series. Adv. in Inequalities for Special Functions. Nova Science Publ. House, 2006, pp. 121–125.
10. *Statsenko I. V.* Rasshireniye svoystv mul'tigarmonicheskikh chisel [Extension of the properties of multiharmonic numbers]. Tochnaya Nauka, 2021, no. 107, pp. 2–4. (in Russian)

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7 DOI 10.17223/20710410/55/2

КЛЕПТОГРАФИЧЕСКИЕ (АЛГОРИТМИЧЕСКИЕ) ЗАКЛАДКИ В ГЕНЕРАТОРЕ КЛЮЧЕЙ RSA

А. В. Маркелова

ООО «НТЦ Альфа-Проект», г. Москва, Россия

E-mail: a@safe-crypto.ru

Рассмотрены основные виды алгоритмических закладок. Представлен способ построения асимметричных клептографических закладок в генераторе ключей RSA, позволяющий владельцу ключа закладки (разработчику или авторизованной спецслужбе) получать доступ к пользовательскому ключу, сгенерированному инфицированным алгоритмом. Сформулированы теоремы, иллюстрирующие работоспособность описанных алгоритмов, оценена вычислительная сложность этих алгоритмов. Продемонстрирована стойкость построенных закладок к некоторым классам атак даже при условии, что противник знает используемые методы и имеет доступ к исходному коду ключевого генератора.

Ключевые слова: *RSA, клептография, алгоритмическая закладка, лазейка, клептографическая закладка, бэкдор.*

KLEPTOGRAPHIC (ALGORITHMIC) BACKDOORS IN THE RSA KEY GENERATOR

A. V. Markelova

Science and Technology Center “AlphaProject”, Moscow, Russia

A cryptographic (algorithmic) backdoor is the ability of the backdoor key owner to gain an unauthorized access to user’s secret information embedded in the cryptoalgorithm. There are two independent classifications of backdoors: by the level of cryptographic strength (weak, symmetric, asymmetric backdoor) and by the method of implementing undeclared capabilities (based on covert channel or on implicit weakening of the cryptoalgorithm). We present examples of each type of backdoor and discuss a method for constructing an asymmetric backdoor based on an implicit weakening of the algorithm in the RSA key generator. Let it be required to generate a public module of the RSA key $n = pq$, $|n| = L$. We will generate such prime numbers that $|p| = |q| = L/2$. Let D be the backdoor parameter, $|D| = K$; ID is the identifier of the generator instance; i is the key generation counter; $\psi_s(a, ID, i)$ is a one-way (trapdoor) function with the trapdoor s on the first argument. Let $(a, D) = 1$ and

$$r'(a, D, r_0) = \begin{cases} \min\{r : r \geqslant r_0; (rD + a) \text{ is prime}\}, & \text{if } r{<}2^{L/2-K} \text{ and } rD + a{<}2^{L/2}; \\ 0, & \text{otherwise.} \end{cases}$$

Let's choose the function $R(x, y, z, i)$ and define $r_{ID}^{(i)}(a, D) = r'(a, D, R(a, D, ID, i))$. For any random $a_p \in \mathbb{Z}_D^*$ and $r'_0 \in \mathbb{Z}$, $(2^{L/2-1})/D < r'_0 < 2^{L/2-K}$, the following values are uniquely determined:

$$p = r_{ID}^{(i)}(a_p, D)D + a_p,$$
$$q = r'(\psi_s(a_p, ID, i)a_p^{-1} \bmod D, D, r'_0)D + \psi_s(a_p, ID, i)a_p^{-1} \bmod D.$$

At the same time, if $r_{ID}^{(i)}(\cdot) \neq 0$ and $r'(\cdot) \neq 0$, then the numbers p and q are prime, $|p| = |q| = L/2$, $|n| = |pq| \in \{L-1, L\}$. If the numbers p and q are generated in this way, then, provided that the secret s is known, the public module $n = pq$ can be factorized in polynomial time of the key length. Really, $p = r_{ID}^{(i)}(\psi_s^{-1}(n \bmod D, ID, i), D)D + \psi_s^{-1}(n \bmod D, ID, i)$. This approach allows to develop a cryptographically strong key generator, even if the adversary knows the methods used and has access to the source code of the key generator. This allows us to use a backdoor generator even in open source systems. Cryptographic strength depends on the choice of algorithm parameters: in particular, on the level of cryptographic strength of the function $\psi_s(a, ID, i)$.

Keywords: *RSA, kleptography, algorithmic backdoor, trapdoor, kleptographic backdoor, backdoor.*

Введение

Алгоритмическая закладка — это заложенная в криптоалгоритм возможность получения неавторизованного доступа владельца ключа закладки к секретной пользовательской информации.

Клептографическая закладка — это преднамеренная модификация криптографического алгоритма (разработанного изначально без алгоритмической закладки) с целью получения неавторизованного доступа владельца ключа закладки к секретной пользовательской информации при сохранении достаточного уровня стойкости к атакам противника и полиномиальной неотличимости работы модифицированного алгоритма от работы исходного алгоритма.

В данной теме, к сожалению, не выработано единой терминологии: например, ряд авторов называет закладки такого вида лазейками («trapdoor» [1, 2]), когда требуется подчеркнуть их отличие от «обычных» закладок (бэкдоров, «backdoor» [3]), вносящих изменения в программное обеспечение или аппаратную платформу. Реализацию алгоритма с клептографической закладкой можно также назвать инфицированным алгоритмом [4].

Важно отметить, что подобные механизмы могут быть частью основной структуры алгоритма (как в случае DUAL_EC [5]) — в этом случае будем называть их алгоритмическими закладками — или они могут быть привнесены в существующие криптоалгоритмы (например, методами [1, 2, 6–8]. Закладки второго вида будем называть клептографическими, чтобы подчеркнуть их отличие от алгоритмических закладок общего вида — в данном случае мы пользуемся понятийным аппаратом Адама Янга и Моти Юнга, которые в 1996 г. ввели термин «клептография», означающий раздел криптографии, посвящённый изучению закладок (бэкдоров, лазеек) в криптоалгоритмах [1]. Можно считать, что клептографические закладки являются частным случаем алгоритмических.

В настоящей работе под «противником» будем понимать того нарушителя/злоумышленника, который не является участником (разработчиком или пользователем)

информационной системы, то есть в общем случае не знает ни ключа закладки, ни ключа какого-либо пользователя, ни особенностей реализации.

Заметим, что алгоритмические/клептографические закладки можно рассматривать как один из способов доступа авторизованных спецслужб к пользовательской информации. В такой трактовке алгоритмическая закладка является одним из инструментов системы оперативно-розысных мероприятий. Таким образом, владелец ключа закладки может быть как нарушителем (если это недобросовестный разработчик прикладного программного обеспечения), так и честным участником системы — например, спецслужбой, чьей задачей как раз является использование данного ключа для проведения оперативно-розысных мероприятий.

Западные спецслужбы уже несколько десятилетий насаждают внедрение алгоритмических закладок в криптоалгоритмы и информационные системы, причём в последнее время это фактически является их официальной позицией. Изначально идеи базировались на депонировании ключа, как в случае с Clipper Chip [9], затем они трансформировались в концепцию «ответственного шифрования» («responsible encryption» [10]), а в последнее время спецслужбы всё чаще используют термин «исключительный доступ» («exceptional access» [11]).

Подобный ребрендинг представляет собой игру понятиями и не меняет сути явления. Во всех случаях речь идёт о том, что пользовательский секретный ключ или данные, защищённые им, становятся доступны спецслужбам.

Таким образом, данная тема давно вышла за рамки чисто теоретических исследований. Алгоритмические закладки проникают даже в международные криптографические стандарты. Самым ярким примером стал алгоритм генерации псевдослучайных чисел DUAL_EC, наличие закладки в котором на данный момент считается доказанным [5]. DUAL_EC вошёл в стандарт NIST [12] в 2006 г. и просуществовал до 2015 г., будучи внедрённым в ряд криптографических продуктов [13].

При невозможности стандартизировать инфицированную версию алгоритма её использование в программных реализациях может быть объяснено мерами по обфускации кода или оптимизацией вычислений, что позволяет автору закладки скрыть факт её внедрения.

В открытых источниках описано немало случаев, когда западные спецслужбы внедряли те или иные закладки в пользовательское программное обеспечение и криптоалгоритмы [3, 9, 14, 15], но это относится к вопросам недекларированных возможностей и выходит за рамки нашего исследования.

Алгоритмические (и в частности, клептографические) закладки отличаются от недекларированных возможностей программного обеспечения тем, что они используют математическую структуру заражаемых алгоритмов и протоколов: выходные данные криптоалгоритма видоизменяются таким образом, что для стороннего наблюдателя результат неотличим от «честного» алгоритма, а владелец ключа закладки может вычислить какую-либо секретную пользовательскую информацию.

В п. 1 представлена классификация алгоритмических закладок по двум ключевым признакам: уровню стойкости и способу реализации недекларированных возможностей. В п. 2 приведены примеры каждого класса для генератора ключей RSA.

В п. 3 рассмотрен авторский метод построения асимметричной закладки на основе неявного ослабления алгоритма. Такие закладки являются стойкими к атакам противника (в том числе в предположении доступа противника к исходному коду генератора). Пункты 4–6 иллюстрируют этод метод конкретными вариантами встраивания лазеек в генератор ключей RSA.

Общая идея такой асимметричной закладки предложена автором в [16]. В настоящей работе даётся более детальное описание инфицированного алгоритма, приводится математическое обоснование его работоспособности и надёжности (теоремы 1–7), а также демонстрируется его место в обобщённой классификации закладок.

1. Виды клептографических закладок

Клептографическая закладка является частным случаем алгоритмической закладки общего вида и характеризуется тем, что она модифицирует криптоалгоритм, который изначально был спроектирован без алгоритмической закладки.

Инфицированный криптоалгоритм должен обладать, как минимум, следующими свойствами:

— *идентичностью инициализации*: инфицированный и исходный алгоритмы работают на одних тех же входных данных (начальных условиях);
— *структурной идентичностью результата*: выходные данные инфицированного алгоритма имеют ту же структуру, что и у исходного алгоритма;
— *функциональной идентичностью результата*: выходные данные удовлетворяют тем же математическим соотношениям, которым должны удовлетворять выходные данные исходного алгоритма.

Дополнительные (опциональные) свойства инфицированного криптоалгоритма:

— *статистическая идентичность результата*: выходные данные инфицированного алгоритма статистически неотличимы от выходных данных исходного алгоритма;
— *неотличимость среднего времени работы* инфицированного криптоалгоритма от времени работы исходного криптоалгоритма.

В информационной системе с алгоритмической (клептографической) закладкой традиционно рассматривают три основные роли участников [4]: *разработчик*, *пользователь* и *противник* (*злоумышленник*).

В принятой терминологии [17, 18] пользователи могут быть внутренними нарушителями, а противник всегда является внешним нарушителем. То есть роль противника (злоумышленника) тождественна определяемой российской нормативно-правовой документацией роли внешнего нарушителя. Данное утверждение подтверждается и тем, что [18] определяет термины «внешний нарушитель» и «противник» как синонимы.

Далее, в соответствии с рекомендациями [18], будем использовать термин «противник», когда речь идёт о внешнем нарушителе. Под «внутренним нарушителем» будем подразумевать участников системы, выполняющих атаки на криптосистему со встроенной алгоритмической закладкой.

Принимая во внимание тенденции последних лет, ролевую модель можно видоизменить, добавив нового участника — *спецслужбу* — и ограничив уровень знания разработчика (производителя программного обеспечения или аппаратуры со встроенной реализацией криптоалгоритма) [19]: спецслужба, будучи автором закладки и владельцем её ключа, описывает производителю инфицированный алгоритм и, возможно, даёт какие-либо открытые данные, используемые этим алгоритмом. Для того чтобы не возникало смысловой путанницы между разработчиком (автором) алгоритма с закладкой и разработчиком соответствующего программного обеспечения, будем далее называть последнего *производителем*.

Разумеется, подобное расширение ролевой модели имеет смысл, если производитель не является автором закладки, а встраивает её в реализацию по указанию спецслужбы.

В [20] выделяют два вида противников:

— *различительный противник* (*distinguishing adversary*): его цель в том, чтобы отличить честную реализацию от инфицированной; различительная атака может либо выявлять отклонения реализации от эталонной (по времени работы, по статистическим характеристикам выхоных данных и т. п.), либо проверять наличие в реализации конкретного варианта клептографической закладки; в случаях, когда закладка официально встраивается в реализацию спецслужбой и документируется, различительная атака не имеет смысла;
— *противник-криптоаналитик* (*cryptanalyzing adversary*): его цель состоит в том, чтобы «сломать» безопасность данного устройства; это может включать нахождение секретного пользовательского ключа или данных, подделку подписи, вычисление ключа закладки и т. д.

По аналогии можем рассмотривать *различительного нарушителя* и *нарушителя-криптоаналитика*.

Нарушителем может быть практически любой участник системы, за исключением спецслужбы, если её действия по встраиванию закладок являются легальными в рамках информационной системы.

Производитель может рассматриваться как нарушитель, ставящий своей целью вычислить секретный ключ закладки и тем самым получить те же права, что и спецслужба. При этом различительная атака на реализацию со стороны производителя бессмысленна, поскольку он по определению знает о наличии закладки.

Пользователь может рассматриваться как нарушитель, причём его целью может быть как успешная различительная атака — то есть обнаружение факта инфицирования криптосистемы, так и получение ключа закладки для дальнейшего доступа к секретам других пользователей.

Резюмируя, выделим четыре основные роли в информационной системе с алгоритмической закладкой:

— *спецслужба* — владелец ключа закладки: обладает информацией о закладке, владеет ключом к закладке, не владеет секретным ключом пользователя, но может получить к нему полный или частичный доступ, используя ключ закладки;
— *производитель* — разработчик программного обеспечения и/или аппаратуры с реализацией инфицированного алгоритма: обладает информацией о закладке, не владеет ключом к закладке (если он не встроен в реализацию), не владеет секретным ключом пользователя;
— *пользователь*: не обладает (в общем случае) информацией о закладке, не владеет ключом закладки, владеет секретным ключом пользователя;
— *противник*: не обладает (в общем случае) информацией о закладке, не владеет ни ключом закладки, ни секретным ключом пользователя.

Алгоритмическая закладка (в частности, клептографическая), как и классическая программная закладка, реализует недекларируемую возможность: предоставляет доступ к секретным пользовательским данным при условии знания ключа закладки и особенностей её структуры. В зависимости от типа закладки доступ может быть предоставлен либо любому, кто знает о наличии закладки в реализации и о её структуре, либо только тому, кто дополнительно знает ключ закладки.

Таким образом, алгоритмические закладки можно классифицировать по уровню стойкости и по способу реализации недекларированных возможностей. По уровню стойкости закладки делятся на *слабые*, *симметричные* и *асимметричные*.

Слабые закладки являются бесключевыми. Единственный метод их защиты — сокрытие факта встраивания закладки и алгоритма её функционирования. Таким образом, если закладка обнаружена, то любой противник и/или внутренний нарушитель получает тот же доступ к пользовательским данным, что и спецслужба.

Симметричные и *асимметричные закладки* защищены с помощью соответственно симметричных и асимметричных ключей. В случае симметричной закладки ключ, необходимый для доступа к пользовательским данным, встроен в реализацию. Этот ключ участвует в защите скрытого канала или в механизме модификации криптоалгоритма. В случае асимметричной закладки ключ, встроенный в реализацию, не позволяет эффективно вычислить ключ доступа к закладке. Таким образом, асимметричные закладки могут быть использованы в реализациях с открытым исходным кодом — при этом доступ к закладке останется только у владельца ключа закладки (спецслужбы).

По способу реализации недекларированных возможностей можно выделить два вида закладок: *на основе скрытых каналов* и *на основе неявного ослабления криптоалгоритма*.

Скрытый канал — это непредусмотренный разработчиком коммуникационный канал, который может быть применён для нарушения политики безопасности [21]. Недекларированной возможностью в случае алгоритмической закладки на основе скрытого канала является передача автору закладки секретной пользовательской информации. При этом в качестве скрытого канала используется часть легально передаваемых по открытым каналам данных.

Для того чтобы скрытый канал реализовывал работу алгоритмической закладки, используется некоторая обратимая функция E. Пусть $D = E^{-1}$ — обратная функция. В качестве E может быть выбрано как бесключевое обратимое преобразование (в том числе тождественное), так и симметричный или асимметричный шифр. Через скрытый канал передаётся сообщение $m = E(x)$, где x — некоторая секретная информация пользователя.

Если E — бесключевое обратимое преобразование, то построенная алгоритмическая закладка является слабой. При этом $x = D(m)$ может вычислить любой противник или внутренний нарушитель, получивший информацию о структуре закладки (то есть о функциях E и D).

Если функция E является симметричным или асимметричным шифром, то построена соответственно симметричная или асимметричная закладка. В этом случае ключ расшифрования для E позволяет эффективно вычислять $D = E^{-1}$ и является ключом закладки. Владелец ключа закладки может вычислить $x = D(m)$ и получить доступ к секретной пользовательской информации.

В алгоритмических закладках на основе неявного ослабления криптоалгоритма недекларированной возможностью является такая модификация базового алгоритма, которая позволяет вычислить какие-либо пользовательские данные на основе открытых данных. То есть при формальной структурной и функциональной идентичности результата выходные данные инфицированного алгоритма удовлетворяют каким-либо дополнительным соотношениям, на основе которых можно восстановить секретную информацию.

Отметим, что две описанные классификации — по уровню стойкости и по способу реализации недекларированных возможностей — являются независимыми, то есть в результате мы определили шесть видов закладок (см. таблицу в заключении). Далее рассмотрим закладки разных видов на примере генератора ключей RSA.

2. Клептографические закладки в генераторе RSA-ключей

Алгоритм RSA [22] широко известен, и мы не будем останавливаться на его описании. Напомним только вид RSA-ключей. Пусть p и q — большие простые числа, $n = pq$. Выбираются такие числа d и e, называемые соответственно закрытой и открытой экспонентой, что

$$e \cdot d \equiv 1 \pmod{(p-1)(q-1)}.$$

Пара чисел (e, n) является открытым ключом (известна всем участникам системы), а d — закрытым (является секретом пользователя). Очевидно, что для нахождения закрытого ключа d достаточно разложить n на простые множители. В общем случае эта задача является вычислительно трудной.

Рассмотрим такие закладки в генераторе ключей RSA, зависящие в некоторых случаях от секретного значения, называемого «ключом закладки», которые позволяют владельцу этого ключа разложить n на множители p и q за полиномиальное от длины ключа время, тогда как для любого стороннего наблюдателя факторизация n по-прежнему является вычислительно сложной задачей. Для слабых (бесключевых) закладок подразумевается, что для факторизации n достаточно знания о структуре закладки.

Подробный обзор методов построения закладок на основе скрытых каналов в генераторе ключей RSA дан в [23]. В качестве скрытого канала может быть использована открытая экспонента e или часть бит открытого модуля n [6–8].

Самым тривиальным примером бесключевой закладки на основе ослабления криптоалгоритма является использование детерминированного числа p. Детерминированность в данном случае означает, что простой делитель n выбирается без использования генератора псевдослучайных чисел, например может быть заранее задан список простых чисел. Второе простое число q при этом вырабатывается случайным образом.

Число p может быть также функцией от идентификатора ключевого генератора или ещё каких-либо открытых данных. Заметим, что построение таким методом асимметричной закладки невозможно, поскольку для взлома (то есть для факторизации n) требуется ровно та же функция, что и для генерации p.

Более сложным примером бесключевой закладки является закладка на основе уязвимости ROCA, обнаруженной в 2016–2017 гг. в криптобиблиотеке RSALib компании «Infineon» [24, 25]. Исследования [25] показали, что все простые числа, вырабатываемые этой библиотекой, имеют вид

$$p = r_p \cdot M + (65537^{a_p} \bmod M). \tag{1}$$

Числа r_p и a_p выбираются, судя по всему, случайно (по крайней мере, авторы исследования не выявили каких-либо закономерностей), а M задаётся как заранее известное произведение нескольких простых чисел. Авторы [25] провели успешную атаку на ключи RSA, выработанные подобным образом, и смогли взломать RSA-512 и RSA-1024, а также оценили возможность взлома для RSA-2048 и RSA-4096.

Невозможно однозначно утверждать, была ли подобная реализация следствием ошибки разработчиков (как математиков, предложивших алгоритм, так и программистов, реализовавших его) или это было намеренным ослаблением алгоритма для упрощения работы спецслужб. Тем не менее данный метод может рассматриваться в качестве примера слабой закладки.

Примером симметричной закладки на основе неявного ослабления алгоритма является закладка Андерсона [2], в которой генератор вырабатывает простые числа вида

$$p = r_p \cdot D + a_p = r(A_p, D) \cdot D + A_p, \tag{2}$$

где D — секретное 200-битное число, называемое «ключом закладки»; $A_p < \sqrt{D}$ — 100-битные числа; $(A_p, D) = 1$; $r(A, D)$ — функция от двух переменных, возвращающая 56-битное значение.

Как можно видеть, структура закладки Андерсона похожа на ROCA (хотя исторически, конечно, правильнее сказать, что структура (1) похожа на (2)), но взлом для противника, на первый взгляд, затруднён из-за использования ключа закладки. Однако в 1994 г. закладка Андерсона всё-таки была взломана [26]: было продемонстрировано, что противник может вычислить ключ закладки, получив всего 14 различных открытых ключей, выработанных с общим значением D.

Наибольший интерес представляют асимметричные закладки. Методы построения асимметричных закладок без скрытых каналов, впервые предложенные в [16], развивают идею, положенную в основу закладки Андерсона и ROCA. Далее эти методы рассмотрены подробнее и сформулированы теоремы, обосновывающие корректность работы модифицированных генераторов, возможность доступа к закладке для владельца ключей закладки и стойкость к атакам противника и внутреннего нарушителя.

3. Асимметричная закладка в генераторе RSA-ключей: общая идея

Пусть требуется выработать ключ RSA с длиной открытого модуля L бит. Обозначим битовую длину произвольного числа X как $|X|$ при условии, что старший бит X равен 1, то есть $2^{|X|-1} \leqslant X < 2^{|X|}$.

Будем вырабатывать такие простые числа, что $|p| = |q| = L/2$. Пусть D — некоторый параметр лазейки, не обязательно секретный, $|D| = K$.

Для произвольных взаимно простых чисел a и D определим функцию

$$r'(a, D, r_0) = \begin{cases} \min\{r : r \geqslant r_0, (rD + a) \text{ — простое}\}, & \text{если } r < 2^{L/2-K} \\ & \text{и } rD + a < 2^{L/2}, \\ 0 & \text{иначе.} \end{cases} \tag{3}$$

То есть $r'(a, D, r_0)$ — это наименьший (но больший r_0) номер члена арифметической прогресии $rD+a$, являющегося простым числом. Для вычисления значения $r'(a, D, r_0)$ сначала задаётся $r = r_0$, а затем r увеличивается на 1, пока число $rD + a$ не окажется простым.

Так как $(a, D) = 1$, по теореме Дирихле [27] арифметическая прогрессия $rD + a$ содержит бесконечно много простых чисел. При этом теоретически возможна ситуация, когда минимальное простое $rD + a \geqslant 2^{L/2}$ или $r \geqslant 2^{L/2-K}$. В этом случае считаем, что $r'(a, D, r_0) = 0$.

Пусть ID — уникальный идентификатор, определённый для каждого экземпляра генератора, $|ID| = m$; i — счётчик генераций ключей. Выберем некоторую функцию

$$T(x, y, z, i) : \mathbb{Z}_2^K \times \mathbb{Z}_2^K \times \mathbb{Z}_2^m \times \mathbb{N} \to \mathbb{Z}_2^{L/2-K}$$

и положим

$$R(x, y, z, i) = \left\lceil \frac{2^{L/2-1}}{D} \right\rceil + T(x, y, z, i), \tag{4}$$

где $\lceil X \rceil$ — верхняя целая часть, то есть $X \leqslant \lceil X \rceil < X + 1$.

Определим для произвольных взаимно простых чисел a и D

$$r_{ID}^{(i)}(a, D) = r'(a, D, R(a, D, ID, i)). \tag{5}$$

Пусть $\psi_s(\cdot) : \mathbb{Z}_D^* \times \mathbb{Z}_2^m \times \mathbb{N} \to \mathbb{Z}_D^*$ — однонаправленная (односторонняя) по первому аргументу функция с секретом s. Генератор с лазейкой работает следующим образом.

На первом шаге вырабатывается случайное число a_p с условиями $(a_p, D) = 1$, $a_p < D$.

На втором шаге вычисляются значения

$$r_p = r_{ID}^{(i)}(a_p, D), \quad p = r_p D + a_p, \tag{6}$$

где i — монотонно увеличивающийся счётчик генераций ключа.

Если $r_p = 0$, то алгоритм возвращается на первый шаг. Если реализация предусматривает использование фиксированной открытой экспоненты e, то дополнительно надо проверить, что $(e, p - 1) = 1$. При невыполнении этого условия необходимо вернуться на первый шаг.

На третьем шаге вычисляются значения

$$c = \psi_s(a_p, ID, i), \quad a_q = c \cdot a_p^{-1} \pmod{D}. \tag{7}$$

Здесь и далее считаем, что в качестве значения по модулю выбирается наименьший положительный вычет.

На четвёртом шаге вырабатывается случайное число r'_0, $|r'_0| = L/2 - K$, и вычисляются значения

$$r_q = r'(a_q, D, r'_0), \quad q = r_q D + a_q. \tag{8}$$

Если $r_q = 0$ или $(e, q - 1) \neq 1$ (при использовании фиксированной открытой экспоненты), то четвёртый шаг повторяется.

Числа p и q — простые по определению функций (3) и (5). Значения d и e вырабатываются стандартными алгоритмами, $n = pq$ — открытый RSA-модуль.

На основе описанного алгоритма можно сформулировать следующую теорему.

Теорема 1. Пусть требуется выработать ключ RSA. Пусть $ID \in \mathbb{Z}_2^m$ — идентификатор экземпляра генератора; $i \in \mathbb{N}$ — счётчик генераций ключей; $D \in \mathbb{N}$ — некоторое натуральное число, $|D| = K$; функции $r'(a, D, r_0)$ и $r_{ID}^{(i)}(a, D)$ определены формулами (3) и (5) соответственно; $\psi_s(\cdot) : \mathbb{Z}_D^* \times \mathbb{Z}_2^m \times \mathbb{N} \to \mathbb{Z}_D^*$ — односторонняя по первому аргументу функция с секретом s. Тогда для любых случайных $a_p \in \mathbb{Z}_D^*$ и $r'_0 \in \mathbb{Z}$, $(2^{L/2-1})/D < r'_0 < 2^{L/2-K}$, однозначно определены значения

$$\begin{gathered} p = r_{ID}^{(i)}(a_p, D)D + a_p, \\ q = r'\left(\psi_s(a_p, ID, i)a_p^{-1} \pmod{D}, D, r'_0\right) D + \psi_s(a_p, ID, i)a_p^{-1} \pmod{D}. \end{gathered} \tag{9}$$

При этом если $r_{ID}^{(i)}(a_p, D) \neq 0$ и $r'\left(\psi_s(a_p, ID, i)a_p^{-1} \pmod{D}, D, r'_0\right) \neq 0$, то числа p и q являются простыми, $|p| = |q| = L/2$, $|n| = |pq| \in \{L - 1, L\}$ и сложность алгоритма генерации не превышает $O(K^3 + \Psi_{Dsm} + C_D)$ битовых операций, где Ψ_{Dsm} — сложность вычисления функции ψ_s; C_D — сложность вычисления значения функции $r_{ID}^{(i)}(a_p, D)$.

Доказательство. Если $r_{ID}^{(i)}(a_p, D) \neq 0$ и $r'\left(\psi_s(a_p, ID, i)a_p^{-1} \pmod{D}, D, r'_0\right) \neq 0$, то однозначность вычисления и простота чисел p и q в (9) следуют из определения функций $r'(a, D, r_0)$ и $r_{ID}^{(i)}(a, D)$.

По формулам (3)–(5) получаем, что

$$r_{ID}^{(i)}(a,D) = r'(a,D,R(a,D,ID,i)) \geqslant R(a,D,ID,i) \geqslant \frac{2^{L/2-1}}{D},$$

следовательно, $p = r_{ID}^{(i)}(a_p,D)D + a_p > 2^{L/2-1}$.

Поскольку $r_{ID}^{(i)}(a_p,D) \neq 0$, из определения (3) следует, что $p = rD + a < 2^{L/2}$, то есть $|p| = L/2$. Условие $|q| = L/2$ следует из выбора r'_0.

Так как $2^{L/2-1} < p,\ q < 2^{L/2}$, то $2^{L-2} < pq < 2^L$, то есть $|n| = |pq| \in \{L-1, L\}$.

Оценим сложность.

Для вычисления p требуется найти значение $r_{ID}^{(i)}(a_p,D)$, для чего понадобится C_D операций.

Обратный элемент $a_p^{-1} \pmod D$ вычисляется с помощью расширенного алгоритма Евклида. Как показано в [28, гл. 4.5.3, следствие L], необходимое количество шагов деления не превышает $O(K)$; битовая сложность каждого деления не превышает $O(K^2)$. Следовательно, получаем $O(K^3)$ операций.

Далее необходимо вычислить $\psi_s(a_p,ID,i)$ и произведение $\psi_s(a_p,ID,i)a_p^{-1} \pmod D$. Это в сумме потребует $O(K^3 + \Psi_{Dsm})$ операций.

Вычисление функции $r'(a,D,r_0)$ быстрее, чем $r_{ID}^{(i)}(a,D)$, поскольку не требуется находить значение $R(a,D,ID,i)$. Следовательно, на построение q уйдёт не более чем $O(K^3 + \Psi_{Dsm} + C_D)$ операций.

В итоге получаем общую оценку сложности $O(K^3 + \Psi_{Dsm} + C_D)$. ■

Отметим, что если $|n| = L-1$, то алгоритм возвращается на первый шаг и генерация простых чисел выполняется заново.

Рассмотрим, каким образом владелец ключа закладки может получить доступ к закрытому ключу пользователя. Для этого достаточно разложить n на простые множители. В общем случае факторизация является вычислительно сложной задачей, однако внедрённая закладка существенно упрощает вычисления.

В силу выбора p и q и условия (7) имеем $n = pq \equiv a_p a_q \equiv c \pmod D$. Поскольку владелец секрета s может обратить функцию ψ_s, то он может вычислить

$$a_p = \psi_s^{-1}(c,ID,i) = \psi_s^{-1}(n \mod D, ID, i).$$

После этого r_p вычисляется по формуле (5), $p = r_p D + a_p$, $q = n/p$.

Таким образом, доказана

Теорема 2. Пусть s — ключ лазейки, числа p и q выработаны по условиям теоремы 1 для некоторого $D \in \mathbb{N}$, $|D| = K$ и $n = pq$. Тогда

$$p = r_{ID}^{(i)}(\psi_s^{-1}(n \mod D, ID, i), D)D + \psi_s^{-1}(n \bmod D, ID, i).$$

Сложность алгоритма вычисления не превышает $O(K^3 + C_D + \Psi_{Ds^{-1}m})$, где $\Psi_{Ds^{-1}m}$ — сложность вычисления функции ψ_s^{-1} при условии знания s; C_D — сложность вычисления значения $r_{ID}^{(i)}(a_p,D)$.

Итак, знание ключа закладки действительно помогает восстановить закрытый ключ пользователя на основе знания открытого модуля n. Отметим, что если у владельца ключа закладки нет информации о текущем значении счётчика генераций, используемого в формуле (5), но при этом известно, что генератор вырабатывает небольшое количество ключей, то значения этого счётчика можно перебрать: необходимо

вычислять r_p для различных i до тех пор, когда полученное значение $p = r_p D + a_p$ окажется делителем n.

Если при взломе ключа нет возможности синхронизировать счётчик с ключевым генератором (т. е. владелец ключа закладки не может узнать, сколько ключей сгенерировано ранее данным генератором) и при этом генератор вырабатывает достаточно много значений, то можно использовать константное значение i (например, $i = 0$).

Стойкость закладки к атакам противника зависит от стойкости функции ψ_s (но не сводится только к ней).

Стойкость ψ_s (то есть возможность её обратить без знания секрета s) зависит, помимо непосредственно битовой длины s, от битовой длины первого аргумента и возвращаемого значения, то есть от битовой длины D. Выбор слишком маленького D может привести к взлому лазейки противником.

С другой стороны, слишком большие значения D сокращают возможное количество рассматриваемых кандидатов в простые. Например, в предельном случае, если битовый размер D равен размеру генерируемых простых чисел, то вычисление значения $rD + a_p$ может привести к превышению максимальной допустимой длины числа уже при $r = 1$.

Вероятно, для RSA с длиной ключа L бит наиболее оптимальным является размер D порядка $3L/8$, но данная оценка требует уточнения в каждом конкретном случае использования.

Мощность множества возможных пар простых чисел p и q при этом меньше, чем при «честной» генерации случайной пары простых чисел.

Во-первых, значение r_p определяется однозначно для каждого фиксированного a_p — то есть из каждой арифметической прогрессии $rD + a_p$ может быть выбрано ровно одно простое число, все остальные простые заведомо отсеиваются — ключевое множество сокращено. Для частичной компенсации этого эффекта в функцию R добавлена зависимость от идентификатора ключевого генератора (то есть общее множество возможных ключей для всех ключевых генераторов расширено) и от номера генерации (то есть даже при выработке одного и того же a_p на разных шагах итоговые открытые модули не будут иметь общих делителей).

Во-вторых, значение a_q тоже однозначно определяется из значения a_p — то есть не любая пара простых чисел может быть выбрана в качестве p и q. Эта зависимость также отчасти компенсируется параметрами ID и i в функции ψ_s. Более того, случайный выбор r'_0 расширяет ключевое множество.

Даже без учёта счётчика генераций у каждого генератора есть $\varphi(D)$ различных способов выбора a_p. Если число D достаточно большое, то теоретическое снижение криптографической стойкости не обязательно приводит к упрощению взлома ключа противником. Стойкость к взлому противником должна отдельно оцениваться в каждом конкретном случае в зависимости от выбранных параметров закладки.

Построим функцию ψ_s на основе задачи дискретного логарифмирования в произвольной циклической группе. Для простоты описания рассмотрим лазейку без учёта ID и счётика генераций.

Пусть $\mathbb{G} = \langle g \rangle$ — конечная циклическая группа. Задачей дискретного логарифмирования в группе $\mathbb{G}$ называется нахождение для произвольного элемента $a \in \mathbb{G}$ такого $x \in \mathbb{Z}$, что $g^x = a$. Решением задачи Диффи — Хеллмана в группе $\mathbb{G}$ называется нахождение элемента g^{xy} по известным элементам g^x и g^y без знания x и y. Будем рассматривать те группы $\mathbb{G}$, в которых задачи дискретного логарифмирования и Диффи — Хеллмана являются вычислительно сложными.

Пусть задано некоторое отображение $\gamma : \mathbb{G} \to \mathbb{Z}_D^*$ и $\theta_\gamma(a) = \{g_i : \gamma(g_i) = a\}$ — множество прообразов элемента $a \in \mathbb{Z}_D^*$ при отображении γ. Требуется, чтобы мощность $\theta_\gamma(a)$ была небольшой для любого a.

Зададим $S = g^s$ для некоторого секрета s и определим одностороннюю функцию с секретом следующим образом:

$$\psi_s(a) = \gamma(g^{c_0}), \text{ где } a = \gamma(S^{c_0}). \tag{10}$$

Очевидно, что в общем случае вычисление $\psi_s(a)$ является сложной задачей, поскольку определение c_0 требует решения задачи дискретного логарифмирования. Но если изначально задать число a в виде $\gamma(S^{c_0})$, то и $\psi_s(a)$ вычисляется легко. То есть сначала нужно выбрать случайное c_0, а потом уже вычислить a и $\psi_s(a)$.

Поскольку выполнено соотношение

$$a = \gamma(S^{c_0}) = \gamma(g^{sc_0}) = \gamma((g^{c_0})^s), \tag{11}$$

то $(g^{c_0})^s \in \theta_\gamma(a)$. Если γ является гомоморфизмом, то $\gamma((g^{c_0})^s) = (\gamma(g^{c_0}))^s$. Тогда для $c = \psi_s(a)$ выполнено

$$\psi_s^{-1}(c) = \psi_s^{-1}(\psi_s(a)) = a \overset{(11)}{=} (\gamma(g^{c_0}))^s = (\psi_s(a))^s = c^s.$$

В противном случае для $c = \psi_s(a) = \gamma(g^{c_0})$ имеет место $g_i = g^{c_0} \in \theta_\gamma(c)$, то есть

$$\psi_s^{-1}(c) = \psi_s^{-1}(\psi_s(a)) = a = \gamma((g^{c_0})^s) = \gamma(g_i^s), \text{ где } g_i \in \theta_\gamma(c).$$

Таким образом, при знании секрета s функция ψ_s обратима. Одновременно с этим верна

Теорема 3. Пусть $\mathbb{G} = \langle g \rangle$ — конечная циклическая группа, в которой задачи дискретного логарифмирования и Диффи — Хеллмана являются вычислительно сложными; $S = g^s$ для некоторого секрета s и $\psi_s(a) = \gamma(g^{c_0})$, где $a = \gamma(S^{c_0})$ для некоторого случайного c_0. Тогда:

1) для почти всех c (за исключением тривиальных) без знания s вычислительно трудно найти $\psi_s^{-1}(c)$;
2) при условии знания S, c_0, $\psi_s(a)$ вычислительно трудно найти s.

Доказательство. Пусть $c = \gamma(g^{c_0})$. Предположим, что мы можем вычислить $a = \psi_s^{-1}(c) = \gamma(S^{c_0})$ для произвольного c_0.

Рассмотрим задачу Диффи — Хэллмана в группе $\mathbb{G}$. Пусть заданы g^a и g^b, требуется найти g^{ab}. Пусть $s = a$, $c_0 = b$, тогда $S = g^a$, $c = \gamma(g^{c_0}) = \gamma(g^b)$. По предположению, мы можем вычислить

$$a = \psi_s^{-1}(c) = \gamma(S^{c_0}) = \gamma((g^a)^b) = \gamma(g^{ab}).$$

Поскольку по условиям выбора группы $\mathbb{G}$ и отображений γ и $\theta_\gamma(c) = \{g_i : \gamma(g_i) = c\}$ мощность $\theta_\gamma(c)$ является небольшой для любого c, перебор значений $\theta_\gamma(\gamma(g^{ab}))$ является быстрым — это даёт возможность найти g^{ab}.

Таким образом, мы свели решение задачи Диффи — Хеллмана к обращению функции ψ_s, а следовательно, вычисление $\psi_s^{-1}(c)$ без знания s является не менее сложной задачей, чем задача Диффи — Хеллмана в группе $\mathbb{G}$. Поскольку по условию выбора группы $\mathbb{G}$ решение задачи Диффи — Хеллмана является вычислительно трудным, доказано первое утверждение теоремы.

Предположим теперь, что для заданных S, c_0, $\psi_s(a)$ возможно вычислить s. Рассмотрим задачу дискретного логарифмирования по основанию g в группе $\mathbb{G}$. Пусть требуется вычислить x, для которого $X = g^x$.

Зададим $S = X$. Выберем случайное c_0 и вычислим $a = \gamma(S^{c_0})$, $\psi_s(a) = \gamma(g^{c_0})$. По предположению, из этих данных возможно вычислить s, то есть найти такое значение s, что $S = g^s$. Так как $\mathbb{G} = \langle g \rangle$, то $x = s$.

Таким образом, второе утверждение теоремы следует из того, что по условию выбора группы $\mathbb{G}$ задача дискретного логарифмирования в ней является вычислительно сложной. ■

Итак, заданная соотношением (10) функция ψ_s является однонаправленной для противника, то есть её обращение является вычислительно трудной задачей, что не позволяет противнику получить доступ к ключу закладки.

Далее рассмотрим конкретные примеры выбора группы $\mathbb{G}$: мультипликативную группу конечного простого поля и подгруппу группы точек эллиптической кривой.

4. Асимметричные закладки в генераторе RSA-ключей на основе функции дискретного логарифмирования по простому модулю

Пусть D — простое число, g — первообразный корень по модулю D. Выберем ключ лазейки — число s, взаимно простое с $(D-1)$. Открытый ключ $S \equiv g^s \bmod D$ встраивается в реализацию и не является секретом; S является элементом максимального порядка по модулю D.

Вариант 1. Определим функцию ψ_s следующим образом:

$$\psi_s(a) = g^{c_0} \bmod D, \text{ где } a = S^{c_0} \bmod D. \tag{12}$$

Фактически, это частный случай формулы (10) для $\mathbb{G} = \mathbb{Z}_D^*$ при $\gamma(x) = x$, $\theta_\gamma(x) = \{x\}$.

По теореме 3 функция ψ_s является стойкой от атак противника в том смысле, что противник не может за полиномиальное время получить доступ к ключу лазейки или обратить функцию ψ_s на основе данных, передаваемых по открытым каналам в процессе работы алгоритма.

Генератор работает так, как описано в п. 3. Теоремы 1 и 2 можно переформулировать следующим образом:

Теорема 4. Пусть требуется выработать ключ RSA. Пусть $ID \in \mathbb{Z}_2^m$ — идентификатор экземпляра генератора; $i \in \mathbb{N}$ — счётчик генераций ключей; $D \in \mathbb{N}$ — простое число; $|D| = K$; g — первообразный корень по модулю D; функции $r'(a, D, r_0)$ и $r_{ID}^{(i)}(a, D)$ определены формулами (3) и (5) соответственно; $s \in \mathbb{Z}_{D-1}^*$ — ключ закладки, $S = g^s \bmod D$. Тогда для любых случайных $c_0 \in \mathbb{Z}_{D-1}$ и $r_0' \in \mathbb{Z}$, $(2^{L/2-1})/D < r_0' < 2^{L/2-K}$, однозначно определены значения

$$p = r_{ID}^{(i)}(S^{c_0} \bmod D, D)D + (S^{c_0} \bmod D),$$
$$q = r'(g^{c_0} \cdot S^{-c_0} \bmod D, D, r_0')D + (g^{c_0} \cdot S^{-c_0} \bmod D).$$

При этом если $r_{ID}^{(i)}(S^{c_0} \bmod D, D) \neq 0$ и $r'(g^{c_0} \cdot S^{-c_0} \bmod D, D, r_0') \neq 0$, то числа p и q являются простыми, $|p| = |q| = L/2$, $|n| = |pq| \in \{L-1, L\}$ и сложность алгоритма генерации не превышает $O(K^3 + C_D)$ битовых операций, где C_D — сложность вычисления значения функции $r_{ID}^{(i)}(a_p, D)$.

Теорема 5. Пусть s — ключ лазейки, числа p и q выработаны по условиям теоремы 4 для некоторого $D \in \mathbb{N}$, $|D| = K$ и $n = pq$. Тогда

$$p = r_{ID}^{(i)}(n^s \bmod D, D)D + (n^s \bmod D).$$

Сложность алгоритма вычисления не превышает $O(K^3 + C_D)$, где C_D — сложность вычисления значения функции $r_{ID}^{(i)}(a_p, D)$.

Доказательство обеих теорем тривиально, если заметить, что

$$n^s \equiv (g^{c_0})^s \equiv S^{c_0} \equiv p \pmod{D}.$$

При реализации следует отсеивать тривиальные значения c_0: 1, -1, $(D-1)/2$.

Вариант 2. Определим функцию ψ_s следующим образом:

$$\psi_s(a) = g^{c_0} \bmod D, \text{ где } a_0 = S^{c_0} \bmod D,\ a = g^{a_0} \bmod D. \tag{13}$$

Как и в варианте 1, число a сразу вырабатывается в виде (13), т. е сначала выбирается c_0 — случайное целое число из интервала $\{2, \ldots, D-2\}$, $c_0 \neq (D-1)/2$, затем вычисляются a_0 и a, после чего находится $c = \psi_s(a) = g^{c_0} \bmod D$.

Так как выполнено соотношение $a_0 \equiv S^{c_0} \equiv g^{sc_0} \equiv (g^{c_0})^s \equiv c^s \pmod{D}$, то для $c = \psi_s(a)$ имеет место $\psi_s^{-1}(c) = \psi_s^{-1}(\psi_s(a)) = a \equiv g^{a_0} \equiv g^{(c^s \bmod D)} \pmod{D}$. Таким образом, владелец ключа лазейки может эффективно обращать функцию ψ_s.

Итак, после выбора c_0 и вычислений по формуле (13) генератор вычисляет

$$a_p \equiv g^{a_0} \pmod{D},\ b_0 = c_0 - a_0 \pmod{D-1},\ a_q \equiv c \cdot a_p^{-1} \equiv g^{c_0 - a_0} \equiv g^{b_0} \pmod{D}.$$

Далее простые числа p и q вычисляются по формулам (6) и (8).

Владелец ключа лазейки осуществляет доступ к пользовательским ключам по формуле

$$a_p \equiv g^{(n \bmod D)^s \bmod D} \pmod{D}.$$

5. Асимметричная закладка в генераторе RSA-ключей на основе функции дискретного логарифмирования в группе точек эллиптической кривой

Пусть D — большое простое число. Рассмотрим эллиптическую кривую в форме Вейерштрасса

$$\mathbb{E} : y^2 = x^3 + ax + b \bmod D). \tag{14}$$

Пусть P — точка этой эллиптической кривой, имеющая порядок t, где t — большое простое число. Будем использовать кривую, состоящую ровно из t точек. Большая часть рассуждений остаётся верной, если порядок кривой больше t или если t является составным числом, однако мы вводим дополнительные ограничения для повышения стойкости алгоритма.

Для произвольной точки A эллиптической кривой $\mathbb{E}$ будем обозначать x-координату этой точки как $x_G(A)$. Пусть $\mathbb{O}$ — нейтральная (нулевая) точка кривой $\mathbb{E}$, то есть $t \cdot A = \mathbb{O}$ для любой точки A.

Выберем ключ лазейки (число $s < t$) и вычислим точку $S = s \cdot P$.

Определим функцию ψ_s следующим образом:

$$\psi_s(a) = x_G(c_0 \cdot P), \text{ где } a = x_G(c_0 \cdot S). \tag{15}$$

Очевидно, что функция $\psi_s(a)$ определена не для всех a, но для построения лазейки это не имеет значения.

Данный вариант также является частным случаем лазейки, описанной в п. 3, с функцией ψ_s, задаваемой формулой (10), для $\mathbb{G} = \mathbb{E}$ при $\gamma(X) = x_G(X)$, $\theta_\gamma(x) = \{X, -X\}$.

Теоремы 1 и 2 можно переформулировать (конкретизировать для выбранных параметров) следующим образом:

Теорема 6. Пусть требуется выработать ключ RSA. Пусть $ID \in \mathbb{Z}_2^m$ — идентификатор экземпляра генератора; $i \in \mathbb{N}$ — счётчик генераций ключей; $D \in \mathbb{N}$ — простое число; $|D| = K$; P — точка эллиптической кривой (14), имеющая порядок t, где t — большое простое число; функции $r'(a, D, r_0)$ и $r_{ID}^{(i)}(a, D)$ определены формулами (3) и (5) соответственно; $s \in \mathbb{Z}_t^*$ — ключ закладки; $S = sP$.

Тогда для любых случайных $c_0 \in \mathbb{Z}_t^*$ и $r_0' \in \mathbb{Z}$, $(2^{L/2-1})/D < r_0' < 2^{L/2-K}$, однозначно определены следующие значения:

$$p = r_{ID}^{(i)}(x_G(c_0S), D)D + x_G(c_0S),$$
$$q = r'\left(x_G(c_0S)^{-1} \cdot x_G(c_0P) \pmod D, D, r_0'\right) D + (x_G(c_0S)^{-1} \cdot x_G(c_0P) \pmod D).$$

При этом если $r_{ID}^{(i)}(x_G(c_0S), D) \neq 0$ и $r'(x_G(c_0S)^{-1} \cdot x_G(c_0P) \pmod D, D, r_0') \neq 0$, то числа p и q являются простыми, $|p| = |q| = L/2$, $|n| = |pq| \in \{L-1, L\}$ и сложность алгоритма генерации не превышает $O(K^3 + C_D + CP_t)$ битовых операций, где C_D — сложность вычисления значения функции $r_{ID}^{(i)}(a_p, D)$; CP_t — сложность вычисления кратной точки.

Пусть функция $\Theta(x) : \mathbb{Z}_D \to \mathbb{E}$ возвращает точку на кривой $\mathbb{E}$ с заданной x-координатой. Заметим, что точек с заданной x-координатой x_0 либо не существует, либо ровно две (возможно, совпадающие): $Q_1 = (x_0, y_1)$ и $Q_2 = (x_0, y_2)$. При этом $Q_1 = -Q_2$, то есть $y_1 \equiv -y_2 \pmod D$.

Если рассматривать наименьшие положительные вычеты, то либо $y_1 = y_2 = 0$ (т. е. $Q_1 = Q_2$), либо ровно одно из чисел y_1, y_2 принадлежит диапазону $\{1, \ldots, (D-1)/2\}$. То есть можно однозначно определить $\Theta(x)$ следующим образом:

$$\Theta(x_0) = \begin{cases} Q = (x_0, y_0), & 0 \leqslant y_0 \leqslant (D-1)/2, \\ \mathbb{O}, & \text{если такой точки не существует.} \end{cases}$$

Теорема 7. Пусть s — ключ лазейки, числа p и q выработаны по условиям теоремы 6 для некоторого $D \in \mathbb{N}$, $|D| = K$ и $n = pq$. Тогда

$$p = r_{ID}^{(i)}(x_G(s \cdot \Theta(n \bmod D)), D)D + x_G(s \cdot \Theta(n \bmod D)).$$

Сложность алгоритма вычисления не превышает $O(K^3 + C_D + CP_t)$, где C_D — сложность вычисления значения функции $r_{ID}^{(i)}(a_p, D)$; CP_t — сложность вычисления кратной точки.

Доказательства теорем 6 и 7 тривиальны. Подробно вычисления объяснены в [16].

Отметим, что $\theta_\gamma(n \bmod D) = \{\Theta(n \bmod D), -\Theta(n \bmod D)\}$. В общем случае нужно было бы вычислить два кандидата в простые делители: p_0 и p_1, а затем в качестве p выбирать тот из них, который является делителем открытого модуля n. Однако нетрудно заметить, что $s \cdot \Theta(n \bmod D) = -\left(s(-\Theta(n \bmod D))\right)$, а следовательно,

$$x_G(s \cdot \Theta(n \bmod D)) = x_G\left(s(-\Theta(n \bmod D))\right),$$

то есть $p_0 = p_1$. Таким образом, кандидаты в простые делители совпадают.

Стойкость функции ψ_s к атакам противника следует из теоремы 3.

6. Асимметричная закладка, аналогичная ROCA

Рассмотрим ещё один вариант закладки. Его структура обобщает вид (1) простых чисел, генерируемых в библиотеке RSALib, подверженной уязвимости ROCA [25].

Пусть $D = \prod_{i=1}^{k} p_i^{\alpha_i}$, где p_i — небольшие простые. Отметим, что генерация чисел вида (6) и (8) для составного D позволяет исключить из рассмотрения составные числа, делящиеся на все p_i. Например, если D — чётное, то количество кандидатов в простые числа сокращается вдвое.

При этом необходимо, чтобы $(a_p, D) = 1$, иначе все числа вида (6) будут составными. Кроме того, поскольку a_q также должно быть взаимно просто с D, необходимо проверить, что $(c, D) = 1$, где c берётся из формулы (7). Для того чтобы избежать проверок $(a_p, D) = (c, D) = 1$, изменим общий вид простых чисел.

Пусть g — элемент максимального порядка по модулю D. Для поиска такого элемента достаточно выбрать первообразные корни $g_1, \ldots, g_m$ по модулям $p_1^{\alpha_1}, \ldots, p_m^{\alpha_m}$, а затем решить систему

$$\begin{cases} x \equiv g_1 \pmod{p_1^{\alpha_1}}, \\ \ldots \\ x \equiv g_m \pmod{p_m^{\alpha_m}}. \end{cases}$$

Поскольку модули взаимно просты, система имеет решение по китайской теореме об остатках:

$$x \equiv g \pmod{p_1^{\alpha_1} \cdot \ldots \cdot p_m^{\alpha_m}}.$$

Так как порядок каждого элемента g_i максимален по соответствующему модулю p_i, то и порядок g максимален по модулю D и равен $\lambda(D)$, где $\lambda(D) = \text{НОК}(\lambda(p_1^{\alpha_1}), \ldots, \lambda(p_m^{\alpha_m}))$ — функция Кармайкла.

Генератор вырабатывает простые числа вида

$$p = r_p D + g^{a_p}, \quad q = r_q D + g^{a_q}, \tag{16}$$

где $a_p, a_q < D$; r_p, r_q задаются по аналогии с формулами (6) и (8).

Мы рассматривали подобный вид простых в п. 4 во втором варианте лазейки, но для простого числа D.

Внедрение закладки похоже на описанный ранее вариант. Пусть есть односторонняя функция ψ_s с секретом s. Сначала выбирается число a_p, для которого затем вычисляются

$$c = \psi_s(a_p), \quad a_q \equiv c - a_p \pmod{\lambda(D)}, \quad r_p = r_{ID}^{(i)}(g^{a_p}, D), \quad r_q = r'(g^{a_q}, D, r'_0),$$

где функции $r_{ID}^{(i)}$ и r' определены, как и раньше, формулами (3) и (5), r'_0 выбирается случайно с условием $(2^{L/2-1})/D < r'_0 < 2^{L/2-K}$. Если какое-то из значений r_p и r_q оказалось равным 0, то необходимо вернуться на первый шаг алгоритма, к выбору a_p.

Числа p и q, вычисленные по формулам (16), являются простыми по определению функций $r_{ID}^{(i)}$ и r'. При этом

$$n \equiv g^c \pmod{D}.$$

Это условие равносильно системе

$$\begin{cases} n \equiv g_1^c \pmod{p_1^{\alpha_1}}, \\ \ldots \\ n \equiv g_m^c \pmod{p_m^{\alpha_m}}. \end{cases} \tag{17}$$

Поскольку все p_i небольшие, то можно решить каждое из уравнений системы (17), после чего найти c по китайской теореме об остатках (решение гарантированно существует по построению).

Далее владелец ключа s может обратить ψ_s, то есть вычислить $a_p = \psi_s^{-1}(c)$, а затем r_p, p и q. В качестве функции ψ_s можно использовать формулы (12) и (15).

Подобный генератор снижает мощность множества возможных пар простых чисел p и q, поскольку мультипликативная группа вычетов по модулю составного D не является циклической, а значит, не все простые числа могут быть представлены в виде (16) для фиксированных D и g. Тем не менее при больших D ключевое множество остаётся достаточно большим, при этом использование составного D позволяет увеличить скорость работы ключевого генератора.

Нетрудно заметить, что вид чисел (1) в библиотеке RSALib фактически повторяет вид (16). Однако использование числа 65537 вместо элемента g максимального порядка существенно снижает стойкость генератора к атакам противника. Именно благодаря этому генератор оказался подвержен атаке Копперсмита.

Заключение

Классифицированы алгоритмические (в частности, клептографические) закладки по способу реализации недекларированных возможностей и по уровню стойкости, выделены шесть основных классов закладок. Для каждого класса в п. 2 приведены примеры практического построения закладок (таблица); подробно рассмотрен самый значимый класс — асимметричные закладки на основе неявного ослабления алгоритма.

Классы алгоритмических закладок

Классы закладок	Слабые	Симметричные	Асимметричные
На основе скрытых каналов	— Закладка Крепо — Слакмона HSD [8] — Закладка Крепо — Слакмона HSPE [8] — Скрытый канал в старших/младших битах [6, 7] — PAP (Pretty-Awful-Privacy) [29] — и другие...		
	При условии использования бесключевой обратимой функции преобразования сообщения	При условии использования симметричного криптографического преобразования сообщения	При условии использования асимметричного криптографического преобразования сообщения
На основе неявного ослабления алгоритма	— Фиксированное p — ROCA [25]	Закладка Андерсона [2]	Закладки Маркеловой [16]

При рассмотрении закладок, базирующихся на идеях Р. Андерсона, возникает вопрос о количестве простых чисел вида (1), (2), (6), (16). По теореме Дирихле, в арифметической прогрессии содержится бесконечно много простых чисел, но строгие оценки наименьшего простого в арифметической прогрессии (теорема Линника [30]), а также количества простых чисел в арифметической прогрессии на заданном интервале (аналог теоремы о распределении простых чисел [27]) на данный момент слишком неточны, поэтому на их основе сложно получить аналитическую оценку времени работы ключевого генератора с закладкой.

Для экспериментальной оценки возможности практической реализации описанных закладок (в том числе на малоресурсных платформах) была сделана модификация генератора ключей RSA российской смарт-карточной ОС «Вигрид». Численные эксперименты проводились на аппаратном эмуляторе платформы P5CC081. В результате

получено, что генератор с закладкой Андерсона работает в среднем то же время, что и неоптимизированный генератор без закладки. Генератор с уязвимостью ROCA работает в среднем в 2 раза быстрее, чем генератор без закладки. Описанный в п. 6 генератор с асимметричной закладкой работает то же время, что и генератор с уязвимостью ROCA.

Подробности численных экспериментов, а также аналитические подходы к оценке быстродействия генераторов с закладками будут описаны в последующих работах.

Можно сделать вывод, что описанные варианты лазеек эффективны и могут использоваться в малоресурсных устройствах (таких, как смарт-карты, usb-токены, устройства интернета вещей). Кроме того, поскольку эти закладки являются асимметричными, то их можно применять в системах с открытым исходным кодом.

ЛИТЕРАТУРА

1. *Young A. and Yung M.* Kleptography: using cryptography against cryptography // EUROCRYPT'97. LNCS. 1998. V. 1233. P. 62–74.
2. *Anderson R. J.* Practical RSA trapdoor // Electronics Lett. 1993. V. 29. No. 11. P. 995.
3. FBI 'planted backdoor' in OpenBSD. 2010. `https://www.theregister.com/2010/12/15/openbsd\_backdoor\_claim`.
4. *Жуков А. Е.* Криптосистемы со встроенными лазейками // BYTE/Россия. 2007. № 2. С. 45–51.
5. *Bernstein D. J., Lange T., and Niederhagen R.* Dual EC: A standardized back door // The New Codebreakers. LNCS. 2016. V. 9100. P. 256–281.
6. *Desmedt Y.* Abuses in cryptography and how to fight them // LNCS. 1990. V. 403. P. 375–389.
7. *Lenstra A. K.* Generating RSA moduli with a predetermined portion // LNCS. 1998. V. 1514. P. 1–10.
8. *Crépeau C. and Slakmon A.* Simple backdoors for RSA key generation // LNCS. 2003. V. 2612. P. 403–416.
9. *Blaze M.* Protocol failure in the Escrowed Encryption Standard // Proc. CCS'94. 1994. `http://www.mattblaze.org/papers/eesproto.pdf`.
10. Deputy Attorney General Rod J. Rosenstein Delivers Remarks on Encryption at the United States Naval Academ. Annapolis, MD, October 10, 2017. `https://www.justice.gov/opa/speech/deputy-attorney-general-rod-j-rosenstein-delivers-remarks-encryption-united-states-naval`.
11. *Levy I. and Robinson C.* Principles for a More Informed Exceptional Access Debate. 2018. `https://www.lawfareblog.com/principles-more-informed-exceptional-access-debate`.
12. *Barker E. and Kelsey J.* NIST Special Publication 800-90. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. June 2006. `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-90.pdf`.
13. *Menn J.* Exclusive: Secret contract tied NSA and security industry pioneer. December 21, 2013. `https://www.reuters.com/article/us-usa-security-rsa/exclusive-secret-contract-tied-nsa-and-security-industry-pioneer-idUSBRE9BJ1C220131220`.
14. *Thomson K.* Reflection on trusting trust // Comm. ACM. 1984. V. 27. No. 8. P. 761–763.
15. *Schneier B.* Evaluating the GCHQ Exceptional Access Proposal. January 17, 2019. `https://www.lawfareblog.com/evaluating-gchq-exceptional-access-proposal`.
16. *Markelova A. V.* Embedding asymmetric backdoors into the RSA key generator // J. Computer Virology Hacking Techniques. 2021. No. 17. P. 37–46.

17. Методический документ. Методика определения угроз безопасности информации. Утвержден ФСТЭК России 5 февраля 2021 г.
18. Словарь криптографических терминов / под ред. Б. А. Погорелова, В. Н. Сачкова. М.: Изд-во МЦМНО, 2006.
19. *Жуков А. Е., Маркелова А. В.* Криптография и клептография: скрытые каналы и лазейки в криптоалгоритмах // Информационная безопасность. 2019. № 1. С. 36–41.
20. *Young A. and Yung M.* Malicious Cryptography. Exposing Cryptovirology. Wiley Publ., 2004. 392 p.
21. ГОСТ Р 53113.1-2008. Информационная технология. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Ч. 1. Общие положения. М.: Стандартинформ, 2018.
22. *Rivest R. L., Shamir A., and Adleman L. M.* A method for obtaining digital signatures and public-key cryptosystems // Comm. ACM. 1978. V. 21. P. 120–126.
23. *Жуков А. Е., Маркелова А. В.* Криптография и клептография. Скрытые каналы и клептографические закладки на их основе в криптосистеме RSA // Защита информации. Инсайд. 2020. № 2(92). С. 58–67.
24. *Švenda P., Nemec M., Sekan P., et al.* The million-key question — investigating the origins of RSA public keys // Proc. 25th USENIX Security'16. USENIX Association, 2016. P. 893–910.
25. *Nemec M., Sys M., Svenda P., et al.* The return of Coppersmith's attack: Practical factorization of widely used RSA moduli // Proc. CCS'17. ACM, 2017. P. 1631–1648.
26. *Kaliski B. S.* Anderson's RSA trapdoor can be broken // Electronics Lett. 1993. V. 29. No. 15. P. 1387.
27. *Дэвенпорт Г.* Мультипликативная теория чисел. М.: Наука, 1971. 200 с.
28. *Кнут Д.* Искусство программированя. Т. 2. Получисленные алгоритмы. 3-е изд. М.: Вильямс, 2001. 832 с.
29. *Young A. and Yung M.* The dark side of black-box cryptography // CRYPTO'96. LNCS. 1997. V. 1109. P. 89–103.
30. *Linnik Yu. V.* On the least prime in an arithmetic progression I. The basic theorem // Матем. сб. 1944. Т. 15(57). № 2. С. 139–178.

REFERENCES

1. *Young A. and Yung M.* Kleptography: using cryptography against cryptography. EUROCRYPT'97, LNCS, 1998, vol. 1233, pp. 62–74.
2. *Anderson R. J.* Practical RSA trapdoor. Electronics Lett., 1993, vol. 29, no. 11, pp. 995.
3. FBI 'planted backdoor' in OpenBSD. 2010. `https://www.theregister.com/2010/12/15/openbsd\_backdoor\_claim`.
4. *Zhukov A. E.* Kriptosistemy so vstroennymi lazeykami [Cryptosystems with built-in trapdoors]. BYTE/Russia, 2007, no. 2, pp. 45-51. (in Russian)
5. *Bernstein D. J., Lange T., and Niederhagen R.* Dual EC: A standardized back door. The New Codebreakers, LNCS, 2016, vol. 9100, pp. 256–281.
6. *Desmedt Y.* Abuses in cryptography and how to fight them. LNCS, 1990, vol. 403, pp. 375–389.
7. *Lenstra A. K.* Generating RSA moduli with a predetermined portion. LNCS, 1998, vol. 1514, pp. 1–10.
8. *Crépeau C. and Slakmon A.* Simple backdoors for RSA key generation. LNCS, 2003, vol. 2612, pp. 403–416.
9. *Blaze M.* Protocol failure in the Escrowed Encryption Standard. Proc. CCS'94, 1994, `http://www.mattblaze.org/papers/eesproto.pdf`.

10. Deputy Attorney General Rod J. Rosenstein Delivers Remarks on Encryption at the United States Naval Academ. Annapolis, MD, October 10, 2017, https://www.justice.gov/opa/speech/deputy-attorney-general-rod-j-rosenstein-delivers-remarks-encryption-united-states-naval.
11. *Levy I. and Robinson C.* Principles for a More Informed Exceptional Access Debate. 2018, https://www.lawfareblog.com/principles-more-informed-exceptional-access-debate.
12. *Barker E. and Kelsey J.* NIST Special Publication 800-90. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. June 2006, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-90.pdf.
13. *Menn J.* Exclusive: Secret contract tied NSA and security industry pioneer. December 21, 2013, https://www.reuters.com/article/us-usa-security-rsa/exclusive-secret-contract-tied-nsa-and-security-industry-pioneer-idUSBRE9BJ1C220131220.
14. *Thomson K.* Reflection on trusting trust. Comm. ACM, 1984, vol. 27, no. 8, pp. 761–763.
15. *Schneier B.* Evaluating the GCHQ Exceptional Access Proposal. January 17, 2019, https://www.lawfareblog.com/evaluating-gchq-exceptional-access-proposal.
16. *Markelova A. V.* Embedding asymmetric backdoors into the RSA key generator. J. Computer Virology Hacking Techniques, 2021, no. 17, pp. 37–46.
17. Metodicheskiy dokument. Metodika opredeleniya ugroz bezopasnosti informatsii [Methodical Document. Methodology for Determining Threats to Information Security]. Approved by the FSTEC of Russia on February 5, 2021. (in Russian)
18. Slovar' kriptograficheskikh terminov [Dictionary of Cryptographic Terms]. B. A. Pogorelov and V. N. Sachkov (eds.), Moscow, MCCME Publ., 2006. (in Russian)
19. *Zhukov A. E. and Markelova A. V.* Kriptografiya i kleptografiya: skrytye kanaly i lazeyki v kriptoalgoritmakh [Cryptography and kleptography: hidden channels and trapdoors in cryptoalgorithms]. Informatsionnaya Bezopasnost', 2019, no. 1, pp. 36–41. (in Russian)
20. *Young A. and Yung M.* Malicious Cryptography. Exposing Cryptovirology. Wiley Publ., 2004, 392 p.
21. GOST R 53113.1-2008. Informatsionnaya tekhnologiya. Zashchita informatsionnykh tekhnologiy i avtomatizirovannykh sistem ot ugroz informatsionnoy bezopasnosti, realizuemykh s ispol'zovaniem skrytykh kanalov. Ch. 1. Obshchie polozheniya [GOST R 53113.1-2008. Information Technology. Protection of Information Technologies and Automated Systems from Information Security Threats Implemented using Covert Channels. P. 1. General Principles]. Moscow, Standartinform Publ., 2018. (in Russian)
22. *Rivest R. L., Shamir A., and Adleman L. M.* A method for obtaining digital signatures and public-key cryptosystems. Comm. ACM, 1978, vol. 21, pp. 120–126.
23. *Zhukov A. E. and Markelova A. V.* Kriptografiya i kleptografiya. Skrytye kanaly i kleptograficheskie zakladki na ikh osnove v kriptosisteme RSA [Cryptography and kleptography. Covert channels and kleptographic backdoors based on them in the RSA cryptosystem]. Zaŝita Informacii. Inside, 2020, no. 2(92), pp. 58–67. (in Russian)
24. *Švenda P., Nemec M., Sekan P., et al.* The million-key question — investigating the origins of RSA public keys. Proc. 25th USENIX Security'16, USENIX Association, 2016, pp. 893–910.
25. *Nemec M., Sys M., Svenda P., et al.* The return of Coppersmith's attack: Practical factorization of widely used RSA moduli. Proc. CCS'17, ACM, 2017, pp. 1631–1648.
26. *Kaliski B. S.* Anderson's RSA trapdoor can be broken. Electronics Lett., 1993, vol. 29, no. 15, pp. 1387.
27. *Davenport H.* Mul'tiplikativnaya teoriya chisel [Multiplicative Number Theory]. Moscow, Nauka, 1971. 200 p. (in Russian)

28. *Knuth D.* The Art of Computer Programming, vol. 2: Seminumerical Algorithms. 3rd ed. Addison-Wesley, 1998.
29. *Young A. and Yung M.* The dark side of black-box cryptography. CRYPTO'96, LNCS, 1997, vol. 1109, pp. 89–103.
30. *Linnik Yu. V.* On the least prime in an arithmetic progression I. The basic theorem. Rec. Math. (Mat. Sbornik) N.S., 1944, vol. 15(57), no. 2, pp. 139–178.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ СТЕГАНОГРАФИИ

УДК 519.85 DOI 10.17223/20710410/55/3

СТЕГОАНАЛИЗ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ И ГЕТЕРОАССОЦИАТИВНЫХ ИНТЕГРАЛЬНЫХ ПРЕОБРАЗОВАНИЙ

М. А. Дрюченко, А. А. Сирота

Воронежский государственный университет, г. Воронеж, Россия

E-mail: m_dryuchenko@mail.ru, sir@cs.vsu.ru

Рассматривается задача стегоанализа цифровых изображений. Предложен подход, предусматривающий разбиение входного изображения большого размера на небольшие неперекрывающиеся блоки и проведение последовательного стегоанализа этих блоков с помощью относительно простых свёрточных сетей с модифицированной архитектурой, включающей слои специальной обработки. После этого в ходе вторичной постобработки проводится объединение получаемой совокупности результатов классификации блоков как последовательности бинарных ответов по схеме наивного байесовского классификатора, в том числе, при необходимости, с учётом потенцильной заполняемости блоков. В качестве средств дополнительной специальной обработки изображений в свёрточных сетях предлагается использовать так называемые интегральные гетероассоциативные преобразования, обеспечивающие выделение на обрабатываемом блоке изображения оценочной и стохастической (маскирующей) составляющих на основе модели прогноза одной части блока по отношению к другой и направленные на выявление нарушения структурных и статистических свойств изображений после внедрения стегосообщения. Такие преобразования включаются в архитектуру обучаемых нейронных сетей в качестве дополнительного слоя. Рассмотрены альтернативные варианты архитектуры используемых глубоких нейронных сетей как с использованием слоя интегрального гетероассоциативного преобразования, так и без него. Исследования проведены для базы цветных изображений PPG-LIRMM-COLOR base и нескольких алгоритмов стегоскрытия, включая классические блочные и блочно-спектральные алгоритмы Куттера, Коха — Жао, более современные алгоритмы EMD, MBEP и алгоритмы адаптивной пространственной стеганографии WOW и S-UNIWARD, обладающие высокой степенью скрытности для стегоанализа. Рассмотрены также разработанные авторами алгоритмы стегоскрытия данных, основанные на использовании гетероассоциативных сжимающих преобразований. Показано, что получаемая при реализации предложенных схем обработки информации точность стегоанализа для изображений большого размера при достаточно скромных вычислительных затратах сопоставима с результатами, полученными другими авторами, а в некоторых случаях и превосходит их.

Ключевые слова: *стеганография, стегоанализ, стегосообщение, цифровые изображения, машинное обучение, глубокие нейронные сети.*

IMAGE STEGOANALYSIS USING DEEP NEURAL NETWORKS AND HETEROASSOCIATIVE INTEGRAL TRANSFORMATIONS

M. A. Dryuchenko, A. A. Sirota

Voronezh State University, Voronezh, Russia

The problem of steganalysis of digital images is considered. The proposed approach is based on the use of deep convolutional neural networks with a relatively simple architecture, distinguished by the use of additional layers of special processing. These networks are trained and used for steganalysis of small fragments of the original large images. For the analysis of full sized images, it is proposed to carry out secondary post-processing, which involves combining the obtained classification results in blocks as a sequence of binary features according to the scheme of a naive Bayesian classifier. We propose to use integral heteroassociative transformations that provide the selection of the estimated and stochastic (masking) components on the processed image fragment based on the forecast model of one part of the fragment in relation to another to identify violations of the structural and statistical image properties after message embedding. Such transformations are included in the architecture of trained neural networks as an additional layer. Alternative versions of deep neural network architectures (with and without an integral layer of heteroassociative transformation) are considered. The PPG-LIRMM-COLOR images base was used to create data sets. Experiments have been carried out for several well-known stego algorithms (including the classic block and block-spectral algorithms of Kutter, Koha — Zhao, modern algorithms EMD, MBEP and algorithms for adaptive spatial steganography WOW and S-UNIWARD) and for the stego algorithms based on the use of heteroassociative compression transformations. It is shown that the accuracy of steganalysis obtained when implementing the proposed information processing schemes for large images with relatively low computational costs is comparable to the results obtained by other authors, and in some cases even exceeds them.

Keywords: *steganography, steganalysis, machine learning, deep neural networks.*

Введение

Задача стегоанализа (СА) [1] состоит в обнаружении факта внедрения визуально незаметного стегосообщения или цифрового водяного знака (ЦВЗ) в объект цифрового контента (изображение, видео, звуковой сигнал и т. п.) и оценке параметров внедрённого сообщения. Обычно в качестве модели стеганографически скрытой информации (ССИ) рассматривается псевдослучайная двоичная последовательность. Задача СА может решаться в прямой постановке как задача анализа контейнера с неизвестным содержимым, так и как обратная задача оценки скрытности алгоритмов компьютерной стеганографии.

В последние 10–15 лет развитие стегоанализа идёт по пути использования методов и алгоритмов машинного обучения как универсального и эффективного подхода к решению любых задач анализа данных. Разнообразие применяемых решений очень велико, что требует систематизированного анализа известных результатов по отношению к вновь предлагаемым решениям с целью определения перспективных направлений исследований и разработок. Одним из таких направлений является использование глубоких нейронных сетей. Следует отметить, что основные продвижения здесь связаны с отходом от традиционной архитектуры свёрточной сети и использованием слоёв

специальной обработки, нетрадиционных функций активации, более сложных архитектур.

Целью данной работы является исследование алгоритмов стегоанализа цифровых изображений, базирующихся на анализе совокупности относительно небольших фрагментов (блоков) на изображении большого размера с помощью глубоких нейронных сетей свёрточного типа, имеющих дополнительные слои для выполнения интегральных гетероассоциативных преобразований, и реализации вторичной постобработки результатов классификации в этих блоках для принятия окончательного решения по всему изображению.

1. Анализ результатов предшествующих работ

Суть подхода, основанного на применении метода машинного обучения, состоит в построении классификаторов объектов-контейнеров для обнаружения факта наличия ССИ на основе реализации процедур обучения по представительным наборам примеров, содержащих заполненные и незаполненные контейнеры. Отличительной особенностью подхода является, прежде всего, его универсальность. Методы и алгоритмы машинного обучения в приложении к задаче стегоанализа можно разделить на две большие группы: классические «неглубокие» методы и алгоритмы машинного обучения; методы и алгоритмы, основанные на применении глубоких нейронных сетей. В дальнейшем мы остановимся на представлении авторских результатов именно в этой области.

1.1. Применение неглубоких методов машинного обучения

К неглубоким алгоритмам машинного обучения относятся: наивный байесовский классификатор; метод опорных векторов; композиционные алгоритмы на основе бэггинга («случайный лес») и бустинга и ряд других. Характерной особенностью этих алгоритмов (в отличие от алгоритмов глубокого обучения) является необходимость предварительной обработки анализируемых объектов для извлечения информативных признаков, используемых при обучении классификаторов. Обзор публикаций, иллюстрирующий широту перечня применяемых методов и алгоритмов стегонализа, включая и указанные методы машинного обучения, приведён, например, в [2]. Одними из первых были работы [3, 4]. Предложенный авторами подход заключается в применении метода опорных векторов. В качестве набора признаков используется вектор размерности 72, вычисляемый из оценок статистических характеристик распределения групп пикселей изображения: математическое ожидание, дисперсия, среднеквадратичное отклонение и т. д. Обучение проводилось по выборке из 1800 пустых контейнеров и случайного подмножества из 1800 заполненных контейнеров, при этом полученная точность классификации составила более 95 % для относительно простых алгоритмов внедрения сообщений.

В развитие этого подхода для решения задачи стегоанализа в последующем были разработаны специальные многомерные системы (пространства) признаков: SPAM (Subtractive Pixel Adjacency Matrix) [5]; SRM (Spatial Rich Model) [6]; PSRM (Projection Spatial Rich Model) [7]. Эффективность различных алгоритмов обработки информации для обнаружения скрытых сообщений по таким признакам обычно демонстрировалась на чёрно-белых изображениях размера 512×512 из базы BOSSbase 1.01 [8]. Для цветных изображений обычно используется база 512×512 PPG-LIRMM-COLOR base [9].

Проверка и сравнение алгоритмов стегоанализа и систем признаков проводилась путём создания обучающих и тестовых примеров на основе наиболее скрытных алгоритмов встраивания ЦВЗ. Для контейнеров-изображений в этой постановке скрывае-

мая информация обычно внедряется при помощи методов адаптивной стеганографии HUGO, S-UNIWARD и WOW, которые считаются наиболее трудно обнаружимыми. Именно по ним приведены лучшие из известных результатов в области стегоанализа. Особенностью этих алгоритмов является то, что они осуществляют встраивание в те области контейнера, которые в минимальной степени искажают стастистические характеристики изображения [7, 10 – 12]. Идея адаптивного внедрения заключается в том, что позиции для внедрения выбираются исходя из свойств изображения; при этом с большей вероятностью внедрение осуществляется в те области, где обнаружить информацию труднее всего (обычно это наиболее «зашумлённые» области).

В ходе сравнительного анализа различных алгоритмов установлено, что наибольшей эффективностью обладают алгоритмы, основанные на использовании метода опорных векторов и ансамблевые (композиционные) алгоритмы. Результаты [7, 13, 14] показали, что в зависимости от объёма полезной нагрузки (payload, pl), измеряемой в среднем количестве бит внедряемого стегосообщения, приходящихся на пиксель контейнера (bit per pixel, bpp), точность стегонализа изменяется в пределах 62–94 %, что говорит о достаточно высоких показателях классификации пустых и заполненных контейнеров. Одним из эффективных приёмов для стегоанализа цифровых изображений является использование алгоритмов сжатия в различных постановках. В работе [14] предложен и использован интегральный классификатор на основе сжатия данных, состоящий из набора отдельных классификаторов, каждый из которых обрабатывает только те контейнеры, которые предварительно автоматически отфильтрованы для него. Принципиальное отличие этого подхода от других, использующих сжатие данных, заключается в том, что здесь сжатие используется на предварительном этапе выбора классификатора, но не для построения самого алгоритма стегоанализа.

1.2. Применение глубоких нейронных сетей

При использовании методов глубокого обучения бинарный классификатор для выявления факта скрытия данных (стегосообщения, ЦВЗ) задаётся в виде глубокой нейронной сети. В подавляющем большинстве работ рассматриваются свёрточные нейронные сети (CNN) в различных модификациях. Информативный англоязычный обзор на эту тему представлен в [15].

Одной из первых работ в этом направлении является [16]. В ней авторы предложили специализированную архитектуру свёрточной нейронной сети, которую они назвали CNN model called Gaussian-Neuron (GNCNN). Её особенностью является использование пространственного фильтра высоких частот с фиксированным ядром, специальных функций активации в виде гауссианы и слоёв субдискретизации с усреднением в пределах окна пуллинга. Использовано пять свёрточных слоёв для извлечения признаков и четыре полносвязных слоя для выполнения классификации. Главная идея такой обработки состоит в том, что пространственный высокочастотный фильтр локализует малые искажения в областях исходного контейнера, связанные с внедрением ССИ. Использование гауссовских активаций обеспечивает реакцию свёрточных слоёв сети на стегосигнал, значения которого локализованы в окрестности нуля, и подавление входных воздействий, вызванных прохождением через фильтр отдельных участков изображения. В итоге авторам по отношению к уже упоминавшимся алгоритмам адаптивной стеганографии HUGO, S-UNIWARD и WOW и чёрно-белых изображений из базы BOSSbase удалось получить меньшую ошибку обнаружения, чем при использовании SVM и признаковой системы SPAM. По сравнению с применением набора SRM и

ансамбля классификаторов, ошибка оказалось примерно на 2–5 % выше в зависимости от уровня полезной нагрузки.

В последних по времени публикациях, посвящённых использованию методологии Deep Learning в СА, используются самые разнообразные по архитектуре глубокие сети [15]. Из этих сетей, как достаточно эффективную, следует выделить Yedrouj-Net [17]. Её архитектура предполагает шесть свёрточных и три полносвязных слоя. Кроме того, в первых свёрточных слоях используются нелинейные активации в виде функций abs(...) и trunk(...) (линейная функции с ограничением снизу и сверху). Полученная с применением такой сети точность классификации при анализе данных из базы BOSSbase имеет значения порядка 72–86 % в зависимости от уровня полезной нагрузки и превосходит результаты, демонстрируемые другими архитектурами (сети Xu-Net, Ye-Net [15, 17]). Приведённые в обзоре [15] данные демонстрируют эффективность ещё одной архитектуры — сети ZhuNet [18], которая превосходит Yedrouj-Net и позволяет уменьшить ошибку на несколько процентов. Например, для алгоритма WOW с pl = 0,4 ошибка может уменьшиться с 0,14 до 0,12.

В работе отечественных авторов [19] проводится исследование собственной модели свёрточной сети и анализ известных результатов по использованию подобных алгоритмов в сравнении с алгоритмами неглубокого машинного обучения. В ходе экспериментов авторам на основе анализа сравнительно небольшого количества изображений удалось получить точность классификации порядка 85 %, что сопоставимо с ранее полученными результатами. Одновременно проводится количественный анализ различных классификаторов на основе систем признаков и нейросетевых алгоритмов. Отмечается, что существенным недостатком статистических классификаторов, отсутствующим в методах на основе нейронных сетей, является их узкая специализация на строго определённые методы формирования стегоконтейнеров.

2. Методы стегоанализа цифровых изображений с использованием глубоких нейронных сетей и гетероассоциативных преобразований

Общей идеей предлагаемого подхода является выполнение обработки информации, предусматривающей разбиение входного изображения большого размера на относительно небольшие неперекрывающиеся блоки и проведение последовательного стегоанализа этих блоков с помощью относительно простых свёрточных сетей с модифицированной архитектурой, опционально включающей дополнительный слой для специальных преобразований изображений. После этого проводится объединение результатов классификации блоков как последовательности бинарных ответов-признаков по схеме наивного байесовского классификатора в ходе вторичной постобработки. Можно ожидать, что такой подход позволит повысить точность стегоанализа, упростить процесс обучения сетей и обеспечить универсальный характер их применения, особенно при неравномерном заполнении контейнера стегосообщением, как, например, в случае, когда используются алгоритмы адаптивной пространственной стеганографии WOW, HUGO, S-UNIWARD.

В качестве средств дополнительной специальной обработки изображений в свёрточных сетях предлагается использовать так называемые интегральные гетероассоциативные преобразования (ИГП), как потенциально способные выявить нарушения структурных и статистических свойств изображений после внедрения стегосообщения (их описание дано далее). ИГП обеспечивают выделение на обрабатываемом фрагменте изображения оценочной и стохастической (маскирующей) составляющих на основе модели прогноза одной части фрагмента по отношению к другой. Такие преобразова-

ния могут включаться в архитектуру обучаемых нейронных сетей в качестве дополнительного слоя.

Как следствие блочного характера обработки в рамках реализуемого подхода, появляется возможность анализа потенциальной заполняемости блоков на этапе вторичной постобработки. При этом можно реализовать адаптивный характер стегоанализа блоков для принятия окончательного решения по всему изображению, т. е. при подсчёте «ответов» в блоках учитывать только те из них, которые могут потенциально иметь достаточно существенную заполняемость. Данный режим целесообразно использовать опционально для выявления стегосообщений, внедрённых с помощью алгоритмов адаптивной пространственной стеганографии.

Общая схема обработки информации показана на рис. 1.

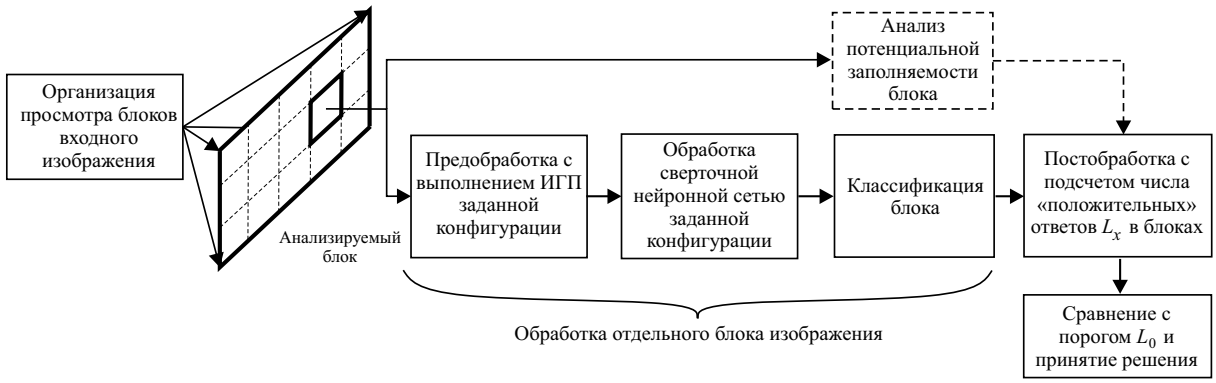


Рис. 1. Общая схема обработки информации в соответствии с предлагаемым подходом

Следует заметить, что в рамках данной схемы могут быть реализованы различные алгоритмы обработки информации, отличающиеся конкретным видом реализуемого ИГП, архитектурами нейронных сетей, способом постобработки.

В ходе проведенных исследований ставились и решались следующие задачи:

— исследование возможностей использования глубоких нейронных сетей свёрточного типа для стегоанализа цветных изображений по отношению к стандартным алгоритмам стегоскрытия в контексте общей идеи блочной обработки и их сравнение с результатами, полученными другими авторами;
— исследование возможностей повышения точности стегоанализа при использовании слоёв интегральных гетероассоциативных преобразований в глубоких нейронных сетях свёрточного типа;
— исследование возможностей повышения точности стегоанализа при использовании различных вариантов алгоритма вторичной постобработки.

2.1. Схема выполнения интегральных гетероассоциативных преобразований

В качестве исходной модели цветного изображения будем рассматривать его представление как реализацию случайного поля, заданного на дискретной сетке: $w(x,y) \in R^3$, $(x,y) \in \Psi = \{1,,\ldots,n\} \times \{1,\ldots,m\}$. Пусть $z \in R^N$ — случайный вектор, представляющий некоторую подобласть (блок) $\Omega \subset \Psi$ и формируемый путём развёртки $w(x,y)$, $(x,y) \in \Omega$, в заданном порядке $z = S_c(w)$, где $S_c(\ldots)$ — функция, определяющая порядок развёртки. Например, если Ω — прямоугольный блок размера $n_w \times m_w$, то $N = n_w \times m_w \times h$.

Вектор z может быть представлен как составной $z = (z_1^{\mathrm{T}}, z_2^{\mathrm{T}})^{\mathrm{T}}$, где $z_1 \in R^{N_1}$ представляет некоторую подобласть $\Omega_I \subset \Omega$ блока $\Omega \subset \Psi$, называемую входной частью, а

$z_2 \in R^{N_2}$ — подобласть $\Omega_O \subset \Omega$, называемую выходной частью. При этом $\Omega_I \cup \Omega_O = \Omega$, $\Omega_I \cap \Omega_O = \varnothing$. Требуется построить преобразование вида

$$F : Z_1 \to Z_{2/1} \quad z_{2/1} = F(z_1), \quad z_1 \in Z_1, \quad z_{2/1} \in Z_{2/1},$$

т. е. преобразование, выполняющее отображение данных входной части в данные выходной. Очевидно, что $z_{2/1}$ имеет определённую погрешность представления

$$z_2 = F(z_1) + V = z_{2/1} + V, \tag{1}$$

где V — стохастическая составляющая.

Указанные преобразования могут осуществляться как в полном объёме (1) с предоставлением всей информации о z_2 при известном z_1, так и со сжатием этой информации [20, 21]. Будем называть такие преобразования гетероассоциативными (ГП) или гетероассоциативными сжимающими преобразованиями (ГСП). При выполнении ГП общего вида области Ω_I, Ω_O могут быть произвольной конфигурации: прямоугольной формы, решётки случайной конфигурации внутри прямоугольных блоков и т. п. Для описания ГП случайных векторов z_1, z_2 могут использоваться как линейные, так и нелинейные модели преобразований.

Пусть некоторое количество сходных по топологии блоков Ω покрывает Ψ так, что $\bigcup\limits_{p=1}^{P} \Omega^{(p)} = \Psi$, $\Omega^{(p)} \bigcap \Omega^{(q)} = \varnothing$ для любых $p, q \in P$, таких, что $p \neq q$. Каждому блоку $\Omega^{(p)}$ соответствует реализация $z^{(p)}$ вектора z. В итоге на всём изображении может быть получена совокупность реализаций $\{z_1^{(p)}, z_2^{(p)} : p = 1, \ldots, P\}$ для входных и выходных частей $\Omega_I^{(p)}, \Omega_O^{(p)}$ блоков $\Omega^{(p)}$, $p = 1, \ldots, P$. Полученные таким образом данные служат обучающей выборкой для построения ГП с использованием прямых вычислений или нейронных сетей соответствующей архитектуры [20, 22, 23].

Если применить ГП к реализации случайной функции (к изображению) в целом, то можно выделить её полную оценочную составляющую $E(x, y) = \tilde{w}(x, y)$ на области определения, которую следует рассматривать как интерполированную функцию, полученную на основе известных входных частей всех блоков, покрывающих Ψ. Такая интерполяция является индивидуальной для каждого изображения, так как преобразование (1) формируется на основе обучающих данных. Точно так же можно выделить полную стохастическую составляющую $V(x, y) = w(x, y) - \tilde{w}(x, y)$, т. е. разложить изображение на две независимые части. Общей идеей такого разделения на составляющие, очевидно различающиеся по корреляционным и частотным свойствам, на основе блоков, покрывающих всё изображение, является использование прямого и обратного ИГП.

Под прямым ИГП понимается преобразование входной части в выходную, под обратным ИГП — выходной части во входную. Для обеспечения статистической однородности оценочной и стохастической маскирующей составляющих желательно, чтобы входная и выходная части были одинакового размера и конфигурации. С учётом использования прямого и обратного преобразований переобозначим введённое отображение входной части в выходную как $F_{\text{io}} : Z_1 \to Z_{2/1}$, $z_{2/1} = F_{\text{io}}(z_1)$. Аналогично введём обратное отображение $F_{\text{oi}} : Z_2 \to Z_{1/2}$, $z_{1/2} = F_{\text{oi}}(z_2)$. Тогда прямым ИГП будем называть отображение

$$G_{\text{io}} : W_1 \to W_{2/1}, \quad w_{2/1} = G_{\text{io}}(w_1), \quad w_1 \in W_1, \quad w_{2/1} \in W_{2/1},$$

где $w_1(x,y)$; $(x,y) \in \Psi_1 = \bigcup_{p=1}^{P} \Omega_1^{(p)}$ — случайное поле, заданное на объединённой области всех входных частей; если $(x,y) \in \Omega_1^{(p)}$, то $w_1^{(p)}(x,y) = S_c^{-1}(z_1^{(p)})$ формируется путём обратной развёртки из вектора $z_1^{(p)}$; $w_{2/1}(x,y)$, $(x,y) \in \Psi_2 = \bigcup_{p=1}^{P} \Omega_2^{(p)}$ — случайное поле, заданное на объединённой области всех выходных частей; если $(x,y) \in \Omega_2^{(p)}$, то $w_{2/1}^{(p)}(x,y) = S_c^{-1}(z_{2/1}^{(p)})$ формируется путём обратной развёртки вектора $z_{2/1}^{(p)} = F_{\text{io}}(z_1^{(p)})$.

Обратным ИГП будем называть отображение

$$G_{\text{oi}} : W_2 \to W_{1/2}, \quad w_{1/2} = G_{\text{oi}}(w_2), \quad w_{1/2} \in W_{1/2}, \quad w_2 \in W_2,$$

где $w_{1/2}(x,y)$; $(x,y) \in \Psi_1 = \bigcup_{p=1}^{P} \Omega_1^{(p)}$ — случайное поле, заданное на объединённой области всех входных частей; если $(x,y) \in \Omega_2^{(p)}$, то $w_{1/2}^{(p)}(x,y) = S_c^{-1}(z_{1/2}^{(p)})$ формируется путём обратной развёртки из вектора $z_{1/2}^{(p)} = F_{\text{oi}}(z_2^{(p)})$; $w_2(x,y)$, $(x,y) \in \Psi_2 = \bigcup_{p=1}^{P} \Omega_2^{(p)}$ — случайное поле, заданное на объединённой области всех выходных частей; если $(x,y) \in \Omega_2^{(p)}$, то $w_2^{(p)}(x,y) = S_c^{-1}(z_2^{(p)})$ формируется путём обратной развёртки из вектора $z_2^{(p)}$.

Для заданных $w_2 \in W_2$, $w_{2/1} \in W_{2/1}$ и $w_1 \in W_1$, $w_{1/2} \in W_{1/2}$ можно получить случайные поля стохастических составляющих как

$$v_2 = w_2 - G_{\text{io}}(w_1) = w_2 - w_{2/1}, \ v_2 \in V_2, \ v_1 = w_1 - G_{\text{oi}}(w_2) = w_1 - w_{1/2}, \ v_1 \in V_1,$$

где $v_2(x,y)$, $(x,y) \in \Psi_2$; если $(x,y) \in \Omega_2^{(p)}$, то $v_2(x,y) = S_c^{-1}(v_2^{(p)})$ формируется путём обратной развёртки из вектора $v_2^{(p)} = z_2^{(p)} - z_{2/1}^{(p)} = z_2^{(p)} - F_{\text{io}}(z_1^{(p)})$; $v_1(x,y)$, $(x,y) \in \Psi_1$; если $(x,y) \in \Omega_1^{(p)}$, то $v(x,y) = S_c^{-1}(v_1^{(p)})$ формируется путём обратной развёртки из вектора $v_1^{(p)} = z_1^{(p)} - z_{1/2}^{(p)} = z_1^{(p)} - F_{\text{oi}}(z_2^{(p)})$.

В итоге можно получить общее интегральное отображение для формирования оценочной и стохастической составляющих всего изображения в виде

$$G_e : W \to \tilde{W}, \ \tilde{w} = G_e(w), \ w = \{w_1, w_2\} \in W = \{W_1, W_2\},$$
$$\tilde{w} = \{w_{1/2}, w_{2/1}\} \in \tilde{W} = \{W_{1/2}, W_{2/1}\}, \ G_m : W \to V, \ v = G_m(w),$$
$$v = \{v_1, v_2\} \in V = \{V_1, V_2\}.$$

Общая схема получения ИГП с заданной конфигурацией входной и выходной частей представлена на рис. 2. Для управления уровнями оценочной и стохастической составляющих в схеме используется зависимость остаточной ошибки обучения нейросетевых преобразователей от числа циклов обучения.

В работах [20, 22 – 24] проведены теоретические исследования, которые позволили проанализировать свойства ГП и выявить наличие определённых преимуществ при их применении в различных задачах обработки изображений. В задачах стегоанализа подобные преобразования предлагается использовать как средства дополнительной обработки, способные выявить нарушения структурных и статистических свойств изображений после внедрения стегосообщения.

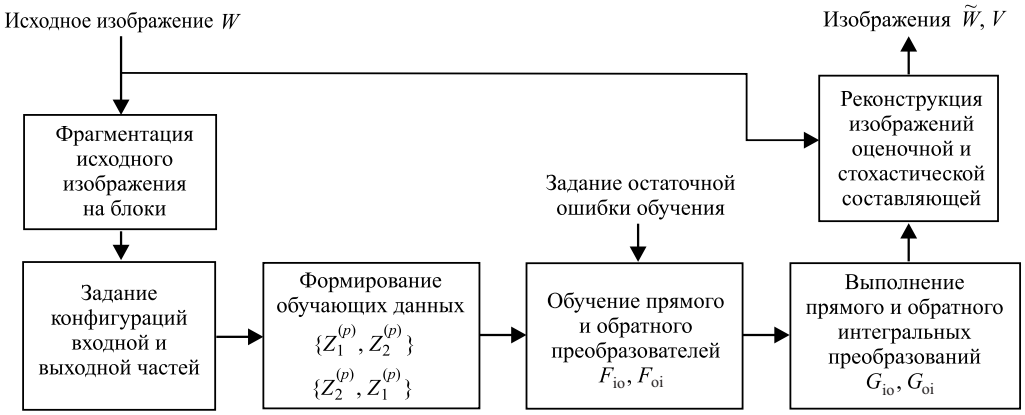


Рис. 2. Схема получения интегральных гетероассоциативных преобразований

2.2. Предлагаемые архитектуры глубоких нейронных сетей

В рамках предлагаемого подхода предложены и исследованы несколько архитектур глубоких нейронных сетей свёрточного типа. Эти сети обучались и тестировались на фрагментах изображений малого размера для последующего стегоанализа изображений большого размера на основе вторичной обработки результатов классификации, полученных на совокупности непересекающихся блоков, покрывающих изображение. При задании архитектуры сетей рассматривались возможности использования ИГП с выделением оценочной и маскирующей составляющих, которые подробно описаны в [20]. Далее представлены две глубокие нейронные сети, применение которых заслуживает особого внимания.

Первая из этих сетей имеет стандартную архитектуру, но содержит всего три обучаемых свёрточных слоя и три обучаемых полносвязных слоя. Её отличие от известных сетей, используемых для СА (например, GNCNN), состоит в том, что после входного слоя введён слой предобработки входного изображения для всех трёх каналов, в котором реализован пространственный высокочастотный фильтр с возможностью гибкой перестройки параметров [25]. Ядро фильтра является симметричным и описывается следующим выражением:

$$h(x,y) = \begin{cases} (-1)^{|x|+|y|+1} \exp\left[-0{,}5\alpha\sqrt{x^2+y^2}\right] \operatorname{sinc}(x/3)\operatorname{sinc}(y/3), & |x| \leqslant d/2, |y| \leqslant d/2, \\ 0, & |x| > d/2, |y| > d/2, \end{cases}$$

где x, y — целочисленные значения аргумента, принимающие положительные и отрицательные значения в пределах размера ядра d. Параметры функции ядра: $d = 5$, $\alpha = 0{,}1$.

Что касается используемых в свёрточных слоях функций активации, то в первом свёрточном слое после слоя высокочастотной фильтрации, свёртки и батч-нормализации реализована (по аналогии с GNCNN) гауссовская функция активации с настраиваемым в процессе обучения параметром влияния σ (среднеквадратичным отклонением). Начальная инициализация σ проводится датчиком случайных чисел в диапазоне 0,01–0,5. Во всех остальных свёрточных и полносвязных слоях применяются активации Relu, за исключением выходного полносвязного слоя, где используется стандартная активация Softmax. Архитектура сети представлена на рис. 3. Далее будем использовать для этой сети обозначение CNN-HF-GN (подчеркивая особенности первого свёрточного слоя).

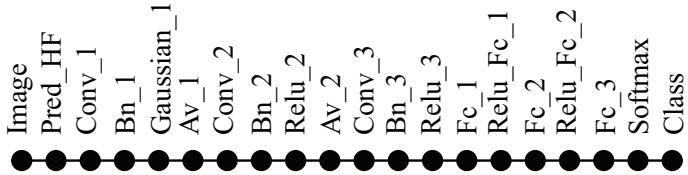


Рис. 3. Архитектура сверточной нейронной сети CNN-HF-GN

На рис. 3 использованы следующие обозначения: Image — входной слой $n_x \times m_y \times 3$, принимающий блок изображения в трёх каналах цветности; Pred_Hf — слой, отвечающий за выполнение пространственной высокочастотной фильтрации; Conv_1 — первый свёрточный слой с ядром свертки 5×5, шагом 1 и 32 каналами; Conv_2, Conv_3 — второй и третий слои с ядром свёртки 3×3, шагом 1, 64 и 128 каналами соответственно; Bn_1, Bn_2, Bn_3 — слои стандартной батч-нормализации; Gaussing_1 — гауссовская функция активации для первого слоя свёртки $f(x) = \exp(-x^2/2\sigma^2)$; Relu_2, Relu_3 — стандартные функции активации для второго и третьего слоёв типа «линейная с ограничением снизу» $f(x) = \max(0, x)$; Av-1, Av-2, Av-3 — слои субдискретизации (пуллинга) на основе вычисления среднего (average pooling) в окне 3×3 с шагом 2; Fc_1, Fc_2, Fc_3 — полносвязные слои, имеющие соответственно 128, 128 и 2 выхода; Relu_Fc_1, Relu_Fc_2 — стандартные функции активации для первого и второго полносвязных слоёв типа «линейная с ограничением снизу» $f(x) = \max(0, x)$; Softmax — стандартная функция активации для классификации на два класса с использованием в качестве функции потерь кросс-энтропии; Class — слой классификации, отвечающий за вычисление функции потерь кросс-энтропии при классификации на несколько взаимоисключающих классов образов.

Как показали многочисленные эксперименты, увеличение количества свёрточных слоев, а также использование слоев dropout не даёт ощутимого прироста точности классификации.

Вторая сеть имеет схожую с предыдущей архитектуру и содержит три обучаемых свёрточных слоя и три обучаемых полносвязных слоя. Её отличие от сети CNN-HF-GN состоит в том, что сразу после входного слоя введён слой обработки входного изображения, основанный на выполнении ИГП с выделением маскирующей составляющей (Masking Component, MC). Можно ожидать, что выделение MC, в которой в основном локализуются искажения, возникающие при внедрении стегосообщения, позволит повысить информативность обработки в целом.

Такие свёрточные сети будем обозначать CNN-MC-HF-GN. Может быть использована также архитектура CNN-MC-GN с исключением высокочастотного фильтра из схемы обработки. На рис. 4 приведена сеть CNN-MC-HF-GN, где обозначение Pred_MC определяет авторский слой обработки с выделением маскирующией составляющей. Все остальные обозначения соответствуют введённым для CNN-HF-GN.

Слой предобработки для выделения MC основан на выполнении ИГП, описанных в п. 2.1. Для построения операторов связи входной и выходной частей (как «вперёд», так и «назад») в качестве типовой использована конфигурация «шахматы», формируемая параллельно во всех трёх каналах цветности в блоках размера 8×8 с размером элементарного квадрата 2×2 (рис. 5).

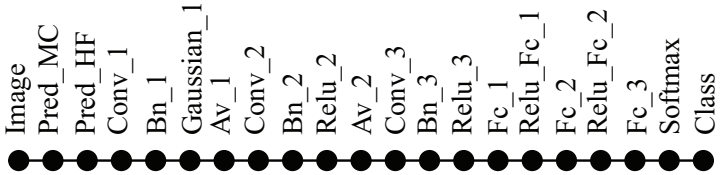
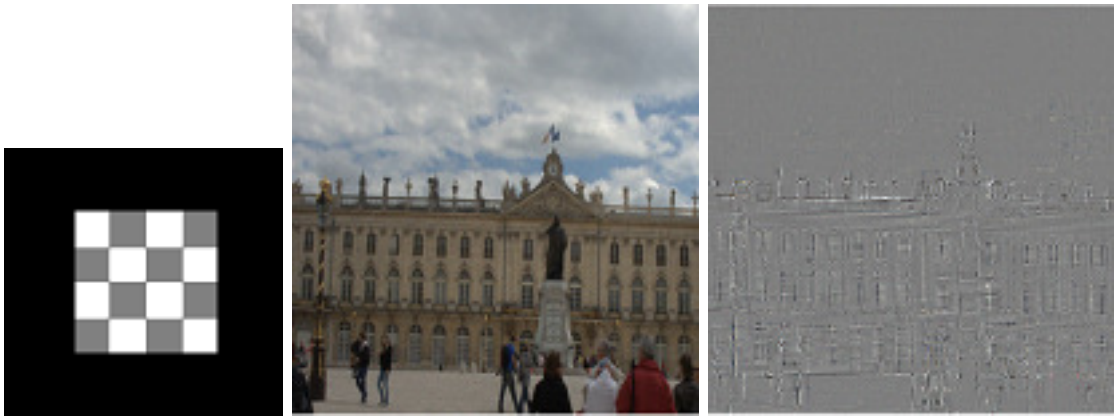


Рис. 4. Архитектура сверточной нейронной сети CNN-MC-HF-GN с использованием дополнительного слоя Pred_MC



а *б* *в*

Рис. 5. Типовая конфигурация входной и выходной частей в проекции на каждый цветовой канал (*а*) и пример исходного изображения (*б*) и его маскирующей составляющей (*в*)

Построение оператора связи осуществлялось на «пустых» изображениях той же выборки, что и при обучении глубокой нейронной сети. При построении оператора по обучающим примерам может использоваться как линейная, так и нелинейная оценка, в соответствии с процедурой, подробно описанной в [20].

2.3. Алгоритмы вторичной обработки результатов классификации

Как уже упоминалось, идея предлагаемого подхода состоит в обучении сетей относительно простой архитектуры на небольших фрагментах (блоках) размером 32×32, 64×64, 128×128 и последовательной вторичной обработке (постобработке) результатов классификации, выполненной с использованием ранее обученного нейросетевого классификатора на блоках целостного изображения большого размера, с целью принятия окончательного решения. При анализе цветных изображений размера $N_x \times M_y$ в ходе постобработки результатов классификации глубокой сетью совокупности блоков размера $n_x \times m_y$, размещаемых без перекрытия, получим $P = (N_x \cdot M_y)/(n_x \cdot m_y)$ бинарных ответов (считаем, что блоки по осям укладываются в исходном изображении кратное число раз). В статистическом смысле можно считать получаемые по каждому блоку решения независимыми. Обозначим эти решения x_k, $x_k \in \{1, 0\}$, $k = 1, \ldots, P$, где $x_k = 1$ обозначает решение в пользу наличия стегосообщения в анализируемом блоке, а $x_k = 0$ — в пользу его отсутствия.

Для синтеза окончательного решающего правила по результатам вторичной обработки введём обозначения для вероятностей значений бинарных признаков двух

классов ω_1 и ω_2, соответствующих пустому и заполненному контейнерам:

$$p_k = \mathsf{P}[x_k = 1 \,|\, \omega_1], \quad 1 - p_k = \mathsf{P}[x_k = 0 \,|\, \omega_1], \quad k = 1, \ldots, P,$$
$$q_k = \mathsf{P}[x_k = 1 \,|\, \omega_2], \quad 1 - q_k = \mathsf{P}[x_k = 0 \,|\, \omega_2], \quad k = 1, \ldots, P.$$

Тогда выражения для функций правдоподобия классов и логарифма отношения правдоподобия можно записать в следующем виде:

$$p(x \,|\, \omega_1) = \prod_{k=1}^{P} p_k^{x_k}(1 - p_k)^{1-x_k}, \quad p(x \,|\, \omega_2) = \prod_{k=1}^{P} q_k^{x_k}(1 - q_k)^{1-x_k},$$
$$g(x) = \sum_{k=1}^{P} \ln \frac{p(x_k \,|\, \omega_1)}{p(x_k \,|\, \omega_2)} = \sum_{k=1}^{P} x_k \ln \frac{p_k\,(1 - q_k)}{q_k\,(1 - p_k)} + P \ln \frac{(1 - p_k)}{(1 - q_k)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} l_0 = \ln \frac{p(\omega_2)}{p(\omega_1)}. \qquad (2)$$

Таким образом, структура классификатора предполагает взвешенное суммирование значений признаков и сравнение этой суммы с вычисляемым порогом. Такой классификатор может быть получен с помощью любого алгоритма машинного обучения по соответствующей выборке ответов первичного классификатора по всем используемым изображениям.

Однако представляет интерес получение подобного классификатора в простой форме. Предположим, что для всех ответов каждого класса вероятности единиц и нулей одинаковы: $p_k = p \neq 0$, $q_k = q \neq 0$, $k = 1, \ldots, P$, причём $p < q$. Тогда $\ln(p(1 - q)/q(1 - p)) < 0$, поэтому решающее правило преобразуется к виду

$$L_x = \sum_{k=1}^{P} x_k \underset{\omega_1}{\overset{\omega_2}{\gtrless}} L_0 = \left(l_0 - P \ln \frac{1 - p}{1 - q} \right) \Big/ \ln \frac{p(1 - q)}{q(1 - p)}, \qquad (3)$$

где L_x — количество единиц (ответов «да»), полученных в ходе наблюдения. Данная ситуация означает, что фактически проводится «опрос» P равноценных признаков и сравнение общего количества полученных единиц с пороговым значением, зависящим от априорных вероятностей классов и соотношения вероятностей ошибок первого и второго рода ответов первичной обработки $p \simeq er_{12}$, $1 - q \simeq er_{21}$, получаемых от нейросетевого классификатора в ходе анализа блоков. Определим порог принятия решения в (3) в виде целочисленной величины

$$L_0 = \operatorname{round} \left[\left(l_0 - P \ln \frac{1 - p}{1 - q} \right) \Big/ \ln \frac{p(1 - q)}{q(1 - p)} \right].$$

Модель принятия решений (3) является приближённой, особенно в ситуациях неравномерного заполнения контейнера стегосообщением, что характерно, например, для алгоритмов WOW, HUGO, S-UNIWARD. В подобных случаях вероятности ошибок принятия решений в блоках нельзя считать одинаковыми, что не позволяет точно рассчитать порог для L_0. Для алгоритма (3) можно спрогнозировать значения порога. Анализ выражения для L_0 при $l_0 = 0$ (стратегия максимального правдоподобия) показывает, что при одинаковых ошибках первого и второго рода ($p = 1 - q$, $q = 1 - p$) выполняется $L_0/P = 0{,}5$. При увеличении $er_{12} = p$, очевидно, пороговое отношение должно увеличиваться, а при увеличении $er_{21} = 1 - q$ — уменьшаться. Расчёты L_0/P показывают, что для значений соотношения $p/1 - q = er_{12}/er_{21}$, лежащих в интервале 0,5–2, значения порога находятся в пределах 0,4–0,6. В ходе тестирования нейросетевого классификатора на блоках из обучающей выборки можно провести оценку

оптимального значения порога, заменив тем самым обучение классификатора (2) обучением порога.

В качестве одной из возможных модификаций алгоритма с применением правила (3) в случае неравномерного заполнения контейнера стегосообщением можно предложить адаптивный алгоритм, предусматривающий подсчёт числа решений не по всем блокам анализируемого изображения, а только по тем из них, которые могут содержать определённое количество модифицированных пикселей. Для этого проводится анализ каждого блока с точки зрения его зашумлённости и выбираются только те блоки, в которых число потенциально модифицированных пикселей может быть больше заданного порога. Простейшим способом подобного отбора блоков является независимое встраивание в него произвольной псевдослучайной последовательности с помощью одного из алгоритмов WOW, HUGO, S-UNIWARD и подсчёта числа изменённых пикселей. Полученный результат может быть близок к реальному числу модифицируемых пикселей. Примерно аналогичные выводы даёт дисперсионный анализ блоков. Тогда правило (3) преобразуется к виду

$$L_x = \sum_{k=1}^{P_{\text{ch}}} x_{i_k} \underset{\omega_1}{\overset{\omega_2}{\gtrless}} L_{0,\text{ch}}, \quad R_{i_k} > \rho\, m_{\text{stego}}, \quad k = 1, \ldots, P, \quad m_{\text{stego}} = \frac{1}{P} \sum_{k=1}^{P} R_k, \qquad (4)$$

где P_{ch} — число блоков анализируемого изображения, которые могут содержать определённое количество модифицированных пикселей; $L_{0,\text{ch}}$ — порог принятия решения для таких блоков; R_{i_k} — количество потенциально модифицированных пикселей в блоке с индексом i_k, $k = 1, \ldots, P$; m_{stego} — среднее арифметическое потенциального заполнения блоков в данном изображении; ρ — коэффициент, определяющий порог, по которому блоки отбираются для использования в процессе принятия решения. Экспериментально установлено, что значения ρ должны устанавливаться в диапазоне 0,25–0,5 в зависимости от P. Подбор оптимального порога может проводиться в ходе эксперимента с учётом ошибок первого и второго рода, полученных на этапе обучения и тестирования нейронной сети, анализирующей блоки. Выявлено, что чем меньше величина pl, определяющая потенциально возможную полезную нагрузку изображения при встраивании в него ССИ, тем больше должен быть коэффициент ρ. Можно ожидать, что эффект от применения адаптивного алгоритма с правилом (4) будет проявляться при использовании блоков малых размеров и анализе изображений с минимальной полезной нагрузкой.

Обобщённая схема алгоритма вторичной обработки представлена на рис. 6, где показан адаптивный вариант алгоритма с правилом (4) с анализом заполняемости блоков. В случае правила (3) соответствующие блоки в этой схеме опускаются.

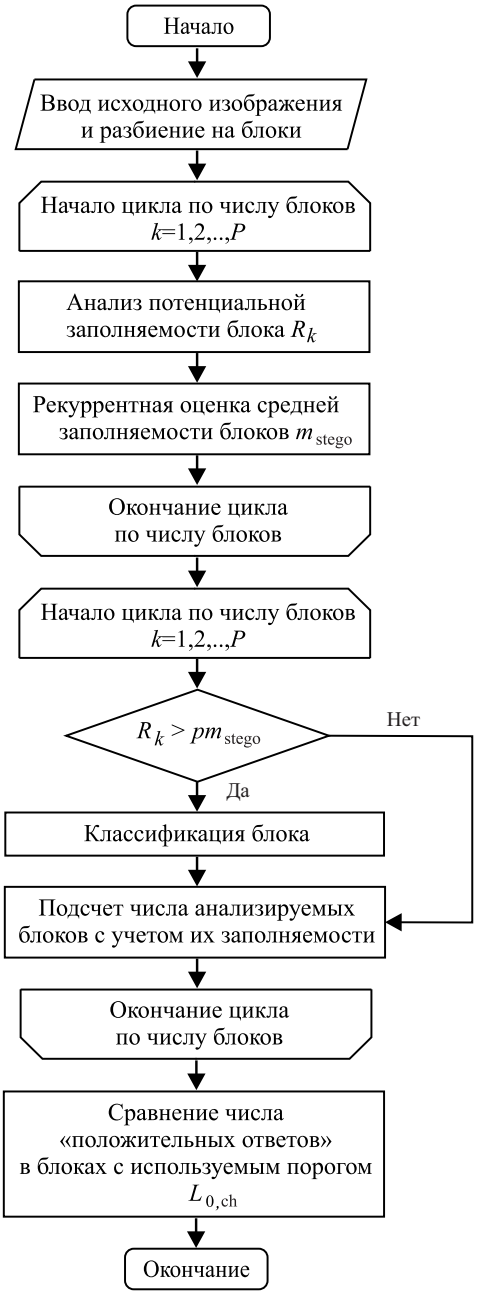


Рис. 6. Схема алгоритма вторичной обработки (постобработки)

2.4. Описание исследуемых алгоритмов ССИ и методики обучения глубоких сетей

В качестве объектов исследования выбрано четыре группы алгоритмов стегоскрытия, в каждой из которых рассматрено по два типовых представителя. Эти группы можно определить следующим образом:

— классические блочные алгоритмы стегоскрытия с внедрением элементов скрываемого сообщения в неперекрывающиеся блоки фиксированного размера — алгоритмы Куттера [26] и Коха — Жао [27];
— современные алгоритмы внедрения в непересекающиеся группы выбранных в псевдослучайном порядке элементов контейнера и адаптивные алгоритмы с использова-

нием различных по степени зашумлённости областей для встраивания — алгоритмы EMD [28] и MBEP [29];

— блочные алгоритмы стегоскрытия на основе ГСП с внедрением элементов сообщения путём модификации спектральных составляющих ГП при случайной конфигурации входной и выходной частей — алгоритмы ГСП+ и ГСП++ [20];

— современные алгоритмы адаптивной пространственной стеганографии, принципиально ориентированные на обеспечение высокой скрытности по отношению к стегоанализу — WOW [10] и S-UNIWARD [11].

Рассмотрим особенности применения этих алгоритмов в представленных далее экспериментах.

В алгоритме Куттера — Джордана — Боссена осуществлялось скрытие данных в пространственной области, разбитой на блоки размера 8×8, путём внесения относительных изменений яркости в синей цветовой компоненте. Это изменение пропорционально значению пикселя и может принимать как положительные, так и отрицательные значения в зависимости от значения встраиваемого бита ЦВЗ. В исследованиях параметр вносимых искажений λ задавался равным 0,05 (уровень модификации 5 %).

В алгоритме Коха — Жао осуществлялось скрытие элементов битовой последовательности в блоки 8×8, подвергнутые дискретному косинусному преобразованию, путём изменения соотношения уровней спектральных коэффициентов среднечастотных составляющих синей цветовой составляющей с позициями (5,4) и (4,5). Параметр изменения ε задавался равным 3, что соответствует среднему уровню вносимых искажений при минимальном значении $\varepsilon = 1$.

Алгоритм EMD (Exploiting Modification Direction) реализует встраивание элементов исходного сообщения, представленных значениями, заданными в позиционной системе счисления с основанием $(2n + 1)$, в выбранные в пседослучайном порядке группы из n элементов изображения. Вносимые в контейнер искажения не превышают одного уровня яркости/цветности. При проведении исследований параметры алгоритма (число элементов n в группах, процент используемых для наполнения групп пикселей контейнера) определялись таким образом, чтобы объём полезной нагрузки составлял примерно 0,4 bpp. Встраивание реализовывалось во все цветовые каналы (RGB) тестовых контейнеров.

Алгоритмы ГСП+ и ГСП++ относятся к блочным алгоритмам, осуществляющим скрытие данных в частотной области ГП [22, 23]. Для извлечения стегосообщения требуется иметь ключ в виде нейронной сети, обученной для классификации блоков изображений. Алгоритм ГСП+ основан на выполнении ГСП в прямом направлении с модификацией пикселей только выходной части блока и встраивании одного бита сообщения в каждый блок. Алгоритм ГСП++ основан на выполнении ГСП как в прямом, так и в обратном направлениях с модификацией пикселей как выходной, так и входной части блока и встраивании двух бит сообщения в каждый блок. В исследованиях использовалась реализация алгоритмов со случайной конфигурацией входной и выходной частей в «пироге трёх цветов» [24] для блоков размера 8×8. Амплитуда встраиваемой последовательности задавалась равной 3/255, что обеспечивает близкую к нулю вероятность ошибки при восстановлении стегосообщения.

В алгоритмах WOW и S-UNIWARD для восстановления стегосообщения требуется знание исходного немодифицированного изображения. Это позволяет обеспечить минимальный уровень искажения каждого пикселя, равного по модулю одному уровню квантованного представления (один бит), и высокую пропускную способность, сопоставимую с общим числом пикселей изображения. В экспериментах мы применяли

алгоритмы с двумя вариантами значений полезной нагрузки pl = 0,2 bpp и 0,4 bpp. Для каждого из этих значений использовалась процедура встраивания только в один цветовой канал (синий) или во все три цветовых канала (RGB).

Алгоритм MBEP (Keyless dynamic optimal multi-bit image steganography using energetic pixels), подобно алгоритмам WOW и S-UNIWARD, реализует адаптивное встраивание информации в пространственном представлении, выбирая для модификации лишь наиболее шумные участки. Однако, в отличие от WOW и S-UNIWARD, он не требует знания исходного контейнера при извлечении скрытых данных. Значения яркости/цветности пикселей контейнера на выбранных участках модифицируются путём перезаписи их младших разрядов битами встраиваемого сообщения (не более четырёх разрядов), что позволяет обеспечить высокую пропускную способность при относительно малых визуальных искажениях носителя. При проведении исследований объём полезной нагрузки для заполнения контейнера составлял примерно 0,4 bpp. Встраивание реализовывалось во все цветовые каналы (RGB) тестовых контейнеров.

В качестве исходного датасета при проведении исследований была взята база данных PPG-LIRMM-COLOR, содержащая 10000 цветных изображений размером 512×512 в формате *.bmp. В качестве стегосообщения генерировалась псевдослучайная последовательность с оригинальным для каждого изображения установочным ключом. Для создания и встраивания стегоообщений в изображения применялись перечисленные алгоритмы ССИ в оригинальной реализации, кроме алгоритмов адаптивной пространственной стеганографии WOW и S-UNIWARD, для которых была использована реализация симулятора [30]. Таким образом, всего проанализировано 20000 изображений: 10000 исходных и 10000 со стегосообщениями.

В табл. 1 приведены характеристики алгоритмов ССИ с точки зрения уровня вносимых искажений и возможности восстановления стегосообщения (MAE — максимальная абсолютная ошибка, MSE — средняя квадратичная ошибка, SSIM — индекс структурного сходства, P_{ex} — вероятность ошибки восстановления). Для алгоритмов WOW и S-UNIWARD данные приведены для случая использования одного канала цветности. При вычислении MAE бралось медианное значение по выборке изображений. Данные по показателям MSE и SSIM рассчитывались после приведения динамического диапазона яркостей пикселей в каналах цветности к диапазону $[0, 1]$.

Для обучения нейронных сетей использовалось 16000 изображений: 8000 исходных и 8000 со стегосообщениями. Валидация и тестирование проводились на 4000 оставшихся изображений: 2000 исходных и 2000 заполненных. При обучении и тестировании сети с различными размерами входного блока из исходных и заполненных изображений вырезались небольшие изображения размера $n_x \times m_y$ со случайным смещением и формировались подвыборки из такого же числа изображений малого размера.

Ввиду пространственной неравномерности заполнения контейнеров стегоообщениями, чтобы исключить использование при обучении детекторов алгоритмов WOW и S-UNIWARD пустых фрагментов заполненных изображений, проводился анализ каждого блока с точки зрения его зашумлённости с помощью алгоритма WOW и подсчёта потенциального числа изменённых пикселей (как для исследования по WOW, так и по S-UNIWARD). При формировании датасета в каждом изображении (как исходном, так и заполненном) выбирались только те блоки, в которых число потенциально модифицированных пикселей больше порога: $R_{i_k} > 0{,}5\, m_{\text{stego}}$, $k = 1, \ldots, P$ (см. (4)). Затем из полученного списка случайным образом выбирался один из блоков. Так получались обучающая и валидационная подвыборки из 16000 и 4000 изображений размера $n_x \times m_y$ для WOW и S-UNIWARD. Кроме того, из 4000 исходных полноразмерных

Таблица 1

Результаты сравнения алгоритмов ССИ по основным характеристикам стегоскрытия

Характеристика	Наименование алгоритма			
	Куттера	**Коха — Жао**	**EMD**	**MBEP**
MAE	12	7	1	15
MSE	9,3205e-06	9,8018e-06	1,6257e-07	6,9528e-06
SSIM	0,99973	0,99979	0,99997	0,99962
P_{ex}	∼0,05	∼0,00	∼0,00	∼0,00
	ГСП+	**ГСП++**	**WOW**	**S-UNIWARD**
MAE	2	2	1	1
MSE	6,9739e-07	1,4331e-06	1,2522e-06	1,0546e-06
SSIM	0,99998	0,99996	0,99996	0,99996
P_{ex}	∼0,01	∼0,01	Для восстановления требуется исходное изображение	Для восстановления требуется исходное изображение

изображений, не участвующих в обучающей подвыборке, формировалась тестирующая подвыборка со случайным выбором участвующих в ней блоков. Это обеспечивало практическую независимость тестирующей и валидационной подвыборок.

При обучении сети использовался оптимизатор adam на 30 эпохах с начальной скоростью 0,001, параметром L_2-регуляризации 0,001, размером минибатча 64. Затем проводилось дообучение сети на 10 эпохах с начальной скоростью 0,0001 и параметром L_2-регуляризации 0,0001.

При обучении нейронных сетей на изображениях с малым уровнем вносимых при ССИ искажений или с малой полезной нагрузкой может случиться, что сеть из-за слишком малых различий пустых и заполненных изображений не выходит в режим обучения. Для преодоления этой ситуации целесообразно проводить задание стартовых значений весовых коэффициентов с использованием сети, ранее обученной на изображениях для высоких уровней полезной нагрузки, и переобучать её на изображениях с малым уровнем полезной нагрузки. Подобная стратегия оказалась весьма эффективной и в ситуациях, когда сети удавалось обучаться самостоятельно при малой полезной нагрузке изображений.

Принципиальным моментом при проведении исследования также являлось использование, по возможности, одинаковых базовых архитектур нейронных сетей. Если есть отличия от базовой архитектуры, они обязательно оговариваются. Все алгоритмы обработки информации реализованы в среде Matlab с использованием пакета Deep Learning Toolbox.

3. Результаты и их обсуждение

Проведено обучение и тестирование предложенных глубоких сетей, а также алгоритма вторичной постобработки. Первоначально внимание уделялось исследованию наиболее скрытных для стегоанализа алгоритмов WOW и S-UNIWARD при различных объёмах полезной нагрузки и заполнении каналов цветности (подробно результаты представлены в [25]). В табл. 2 приведены результаты для базовой архитектуры глубокой сети CNN-HF-GN, полученные при тестировании на блоках изображений

небольших размеров и после постобработки совокупности блоков для изображений размера 512×512 на основе алгоритмов (3)/(4).

Таблица 2

Результаты оценки точности обнаружения в блоках и по всему изображению для нейронной сети CNN-HF-GN при различных объемах полезной нагрузки в %

Алгоритм ССИ	**WOW (grayscale)**		**WOW (RGB)**		**S-UNIWARD (grayscale)**		**S-UNIWARD (RGB)**	
Размеры и число блоков на изображении	pl=0,2	pl=0,4	pl=0,2	pl=0,4	pl=0,2	pl=0,4	pl=0,2	pl=0,4
$n_x = 32$,	54,83	60,55	61,60	72,45	53,37	60,10	61,35	73,12
$m_y = 32$,	55,97/	70,38/	81,30/	92,10/	56,40/	65,63/	77,90/	91,47/
$P = 256$	59,02	71,50	82,53	92,85	57,95	67,30	79,03	91,73
$n_x = 64$,	56,05	66,27	65,75	81,57	55,37	65,70	64,82	77,80
$m_y = 64$,	58,07/	76,57/	82,65/	93,92/	58,13/	74,28/	82,42/	92,47/
$P = 64$	62,50	77,28	82,17	93,83	59,02	75,35	82,82	92,03
$n_x = 128$,	59,78	74,05	76,18	87,90	60,62	75,55	73,62	85,62
$m_y = 128$,	65,38/	82,57/	87,30/	94,72/	66,30/	81,37/	84,42/	94,45/
$P = 16$	66,32	82,28	88,00	94,47	67,35	81,88	84,45	94,45

Анализ результатов показывает, что при заполнении всех трёх цветовых каналов точность обнаружения существенно повышается, как и при увеличении размера анализируемого фрагмента. При малой нагрузке pl=0,2 только в одном цветовом канале, что соответствует реальному заполнению контейнера ещё в 3 раза меньше, точность обработки не выше 60 %, при этом величина ошибки первого рода, когда пустой фрагмент принимается за заполненный, может превышать 50 % при существенно меньшей ошибке второго рода. Однако при достаточно большой нагрузке вероятности ошибок для оптимального значения порога отличаются не так существенно.

На рис. 7 показаны типичные гистограммы, описывающие распределение числа ошибок первого и второго рода после вторичной обработки и точности классификации в зависимости от отношения порога L_0/P при размере блока 32×32. Внедрение стегосообщения здесь осуществлялось алгоритмом WOW в трёх каналах цветности при объёме полезной нагрузки pl = 0,4. Максимальное значение точности стегоанализа 92,1 достигается при отношении $L_0/P = 0{,}65$. При использовании правила (4) максимальное значение точности 92,85 достигается при отношении $L_{0,\mathrm{ch}}/P_{\mathrm{ch}} = 0{,}5$.

Таким образом, предложенный подход и реализованные на его основе алгоритмы позволяют достичь точности обнаружения стегосообщений, сопоставимой с лучшими результатами, представленными в известных работах, а в некоторых случаях и превышающей их. Также следует отметить, что, как и ожидалось, применение адаптивного алгоритма вторичной обработки оправдано в большей степени при малых размерах анализируемых блоков и соответственно большом их количестве, а также при меньшей величине полезной нагрузки. Этот вариант алгоритма позволяет получить прирост точности от 0,5 до 4,5 %.

Результаты исследования возможностей стегоанализа цветных изображений по отношению к алгоритмам стегоскрытия на основе использования ИГП при построении глубоких нейронных сетей представлены в табл. 3. Применялись сети CNN-HF-GN и

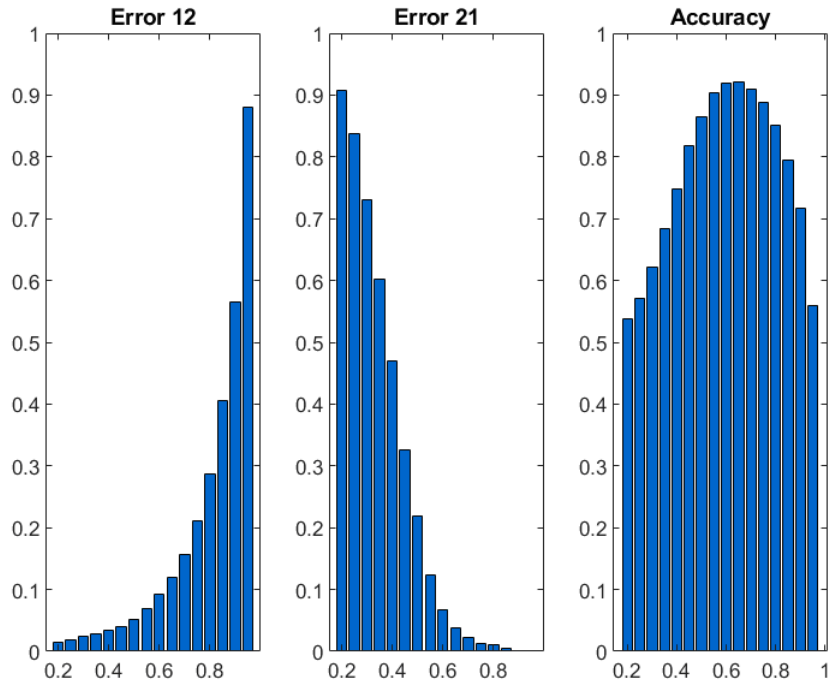


Рис. 7. Типичные гистограммы распределения числа ошибок первого и второго рода и точности классификации изображений от отношения L_0/P

CNN-MC-HF-GN, а также вариант CNN-MC-GN для блоков размера 64×64 и вторичной обработки ответов в блоках для изображений размером 512×512.

Таблица 3

Результаты оценки точности обнаружения в блоках и по всему изображению для нейронной сети CNN-HF-GN и CNN-MC-HF-GN (CNN-MC-GN)

Используемая сеть и размер изображения	CNN-HF-GN		CNN-MC-HF-GN / CNN-MC-GN	
	64×64	512×512	64×64	512×512
Алгоритм ССИ				
Куттера	86,78	96,50	85,70	96,77
Коха — Жао	73,10	84,75	79,15 / 91.27	89,15 / 98,05
EMD	85,88	96,25	86,60	96,57
MBEP	94,23	98,04	93,93 / 93,53	98,72 / 98,83
ГСП+	85,30	92,20	85,78	92,33
ГСП++	93,00	97,23	93,00 / 92,80	97,25 / 97,23
WOW(rgb), pl=0,2	67,90	83,57	69,45	85,35
WOW(rgb), pl=0,4	83,30	93,98	82,97	94,35
S-UNIWARD(rgb), pl=0,2	65,48	80,73	65,60	83,33
S-UNIWARD(rgb), pl=0,4	77,80	92,03	80,37	93,33

Анализ данных в табл. 3 показывает, что эффективность обработки на изображениях большого размера повышается при использовании дополнительного слоя, обеспечивающего выделение маскирующей составляющей на основе ИГП. Выигрыш составляет в среднем 0,5–2 %, а порой и более 5 % (для алгоритма Коха — Жао). Указанный эффект наблюдается даже в случаях, когда между CNN-HF-GN и CNN-MC-HF-GN нет существенных отличий в точности анализа блоков, что объясняется лучшей обобщаю-

щей способностью второй сети, фиксируемой также и на графиках процесса обучения. Для некоторых алгоритмов стегоскрытия хорошие результаты позволяет достичь использование CNN-MC-GN без высокочастотного фильтра, причём для алгоритма Коха — Жао этот вариант даёт весьма значительное преимущество — до 9 %.

Алгоритмы ГСП+ и ГСП++ обладают сопоставимой с другими устойчивостью по отношению к стегоанализу. При этом алгоритм ГСП+ в этом плане уступает только алгоритмам WOW и S-UNIWARD при малой полезной нагрузке pl=0,2. Поскольку последние требуют исходного изображения для восстановления стегосообщения, данный результат свидетельствует о высокой степени скрытности алгоритмов ССИ, основанных на использовании гетероассоциативных сжимающих преобразований.

Для сравнения с результатами других авторов приведены данные в табл. 4. Здесь указаны характеристики архитектур (количество и параметры обучаемых слоёв) наиболее эффективных сетей Yedroudj-Net и Zhu-Net [17, 18] и достигнутая ими точность стегоанализа для алгоритмов WOW и S-UNIWARD на изображениях размера 256×256. Приведены аналогичные данные по архитектуре и точности сети CNN-MC-HF-GN, показавшей лучшие результаты в наших экспериментах, для блоков размера 64×64 и после вторичной обработки блоков на изображениях размера 512×512.

Таблица 4

Результаты оценки точности обнаружения факта стегоскрытия для сетей Yedroudj-Net, Zhu-Net и CNN-MC-HF-GN

Используемая сеть и характеристики её архитектуры	Yedroudj-Net (6 слоев свертки — ядра 5×5, 3×3; 3 полносвязных слоя)	Zhu-Net (6 слоев свертки — ядра 3×3; 2 полносвязных слоя)	CNN-MC-HF-GN (3 слоя свертки — ядра 5×5, 3×3; 3 полносвязных слоя)	
	256×256	256×256	64×64	512×512
Алгоритм ССИ				
WOW(rgb), pl=0,2	72,20	76,70	69,45	85,35
WOW(rgb), pl=0,4	85,60	88,20	82,97	94,35
S-UNIWARD(rgb), pl=0,2	63,40	71,50	65,60	83,33
S-UNIWARD(rgb), pl=0,4	77,20	84,70	80,37	93,33

Анализ представленных данных показывает, что после обучения даже в блоках размера 64×64 сеть CNN-MC-HF-GN с использованием слоя ИГП показывает сопоставимые результаты с Yedroudj-Net и несколько хуже Zhu-Net. В ходе обработки изображений большого размера её эффективность резко возрастает и даёт возможность ожидать, как минимум, не худшую точность, чем рассматриваемые аналоги. При этом архитектура сети CNN-MC-HF-GN с точки зрения количества обучаемых слоёв проще, что, с учётом существенно меньшей размерности входного изображения в блоке, позволяет снизить затраты на обучение сетей с предлагаемой архитектурой.

Заключение

В развитие известных результатов в области машинного обучения и применения глубоких нейронных сетей для стегоанализа цифровых изображений предложен подход, основанный на использовании свёрточных сетей в качестве классификаторов для последовательного анализа небольших блоков изображений с вторичной постобработкой для вынесения решения по всему изображению в целом. Рассмотрены и исследованы альтернативные варианты архитектур глубоких сетей. Предложена простая архитектура свёрточной сети, состоящей из трех свёрточных и трёх полносвязных

обучаемых слоёв. В первом свёрточном слое дополнительно реализован слой пространственной высокочастотной фильтрации с возможностью гибкой перестройки параметров фильтра и гауссовская функция активация с настраиваемым в процессе обучения параметром влияния. Альтернативный вариант архитектуры сети основан на использовании слоя, выполняющего ИГП. Последнее обеспечивает выделение в обрабатываемом блоке изображения оценочной и стохастической (маскирующей) составляющих путём построения модели прогноза одной части блока по отношению к другой. В качестве алгоритма вторичной обработки результатов классификации совокупности блоков в пределах одного изображения использован простой алгоритм сравнения общего числа «положительных» и «отрицательных» решений с экспериментально подбираемым (обучаемым) порогом.

При проведении исследований рассматривались возможности стегоанализа цветных изображений по отношению к широкому перечню алгоритмов стегоскрытия. Показано, что предложенные алгоритмы стегоанализа позволяют выявлять факт наличия стеганографически скрытой информации при использовании наиболее скрытных алгоритмов внедрения с точностью, не уступающей результатам, полученным в работах других авторов. При этом вычислительные затраты на процесс обучения снижаются за счёт более простой архитектуры и уменьшения размерности входных данных.

Установлено, что результирующая эффективность обработки на изображениях большого размера повышается при использовании дополнительного слоя, обеспечивающего выделение маскирующей составляющей на основе ИГП. Как одно из преимуществ предлагаемого подхода следует отметить независимость реализуемой схемы обработки от размера анализируемого изображения в той её части, где проводится обучение нейронных сетей, а также возможность быстрого переобучения нейронных сетей, ранее обученных на изображениях с высокой полезной нагрузкой, для обнаружения стегосообщений на изображениях с малой полезной нагрузкой.

ЛИТЕРАТУРА

1. *Шелухин О. И.* Стеганография. Алгоритмы и программная реализация. М.: Горячая линия — Телеком, 2017. 592 с.
2. *Czaplewski B.* Current trends in the field of steganalysis and guidelines for constructions of new steganalysis schemes // Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne. 2017. No. 10. P. 1121–1125.
3. *Lyu S. and Farid H.* Detecting hidden messages using higher-order statistics and support vector machines // Intern. Workshop Inform. Hiding. Berlin; Heidelberg: Springer, 2002. P. 340–354. `https://farid.berkeley.edu/downloads/publications/ih02.pdf`.
4. *Lyu S. and Farid H.* Steganalysis using color wavelet statistics and one-class support vector machines // Proc. SPIE. California, USA, 2004. P. 35–45. `https://www.researchgate.net/publication/221011180\_Steganalysis\_using\_color\_wavelet\_statistics\_and\_one-class\_support\_vector\_machines`.
5. *Pevny T., Bas P., and Fridrich J.* Steganalysis by subtractive pixel adjacency matrix // IEEE Trans. Inform. Forensics Security. 2010. V. 5. No. 2. P. 215–224.
6. *Fridrich J.* Rich models for steganalysis of digital images // IEEE Trans. Inform. Forensics Security. 2012. V. 7. No. 3. P. 868–882.
7. *Holub V. and Fridrich J.* Random projections of residuals for digital image steganalysis // IEEE Trans. Inform. Forensics Security. 2013. V. 8. No. 12. P. 1996–2006.
8. *Bas P., Filler T., and Pevny T.* Break our steganographic system the ins and outs of organizing BOSS // LNCS. 2011. V. 6958. P. 59–70.

9. http://www.lirmm.fr/chaumont/PPG-LIRMM-COLOR.html — PPG-LIRMM-COLOR base.
10. *Pevny T., Filler T., and Bas P.* Using high-dimensional image models to perform highly undetectable steganography // LNCS. 2010. V. 6387. P. 161–177.
11. *Holub V. and Fridrich J.* Digital image steganography using universal distortion // Proc. 1st ACM Workshop IHMMSec. ACM, 2013. P. 59–68.
12. *Holub V. and Fridrich J.* Designing steganographic distortion using directional filters // Proc. 4th IEEE Intern. Workshop WIFS. 2012. P. 234–239.
13. *Kodovsky J., Fridrich J., and Holub V.* Ensemble classifiers for steganalysis of digital media // IEEE Trans. Inform. Forensics Security. 2010. V. 7. No. 2. P. 434–444.
14. *Монарев В. А., Пестунов А. И.* Эффективное обнаружение стеганографически скрытой информации посредством интегрального классификатора на основе сжатия данных // Прикладная дискретная математика. 2018. № 40. С. 59–71.
15. *Tabares-Soto R., Ramos-Pollan R., and Isaza G* Deep learning applied to steganalysis of digital images: A systematic review // IEEE Access. 2019. V. 7. P. 68970–68990.
16. *Qian Y., Dong J., Wang W., and Tan T.* Deep learning for steganalysis via convolutional neural networks // Proc. Int. Symp. Electron. Imag. 2015. V. 9409. Art. no. 94090J.
17. *Yedroudj M., Comby F., and Chaumont M* Yedrouj-Net: An efficient CNN for spatial steganalysis // Proc. IEEE Intern. Conf. Acoustics, Speech Signal Processing. 2018. P. 2092–2096.
18. *Zhang R., Zhu F., Liu J., and Liu G* Efficient Feature Learning and Multisizeimage Steganalysis Based on CNN. https://arxiv.org/pdf/1807.11428.pdf. 2018.
19. *Полунин А. А., Яндашевская Э. А.* Использование аппарата сверточных нейронных сетей для стегоанализа цифровых изображений // Труды ИСП РАН. 2020. Т. 32. № 4. С. 155–164.
20. *Сирота А. А., Дрюченко М. А.* Обобщенные алгоритмы сжатия изображений на фрагментах произвольной формы и их реализация с использованием искусственных нейронных сетей // Компьютерная оптика. 2015. № 5. С. 751–761.
21. *Dryuchenko M. A. and Sirota A. A.* Interpolation and masking effects of heteroassociative compressive transformations // J. Phys.: Conf. Ser. 2020. V. 1902. P. 1–10. https://iopscience.iop.org/article/10.1088/1742-6596/1902/1/012058/pdf.
22. *Дрюченко М. А., Сирота А. А.* Гетероассоциативные сжимающие преобразования цифровых изображений и их интерполирующие и маскирующие свойства // Сб. трудов Междунар. науч.-техн. конф. «Актуальные проблемы прикладной математики, информатики и механики». Воронеж, 07–09 декабря 2020. С. 312–322.
23. *Сирота А. А., Дрюченко М. А., Митрофанова Е. Ю.* Метод создания цифровых водяных знаков на основе гетероассоциативных сжимающих преобразований изображений и его реализация с использованием искусственных нейронных сетей // Компьютерная оптика. 2018. № 3. С. 483–494.
24. *Сирота А. А., Дрюченко М. А., Митрофанова Е. Ю.* Нейросетевые функциональные модели и алгоритмы преобразования информации для создания цифровых водяных знаков // Изв. вузов. Радиоэлектроника. 2015. № 1. С. 3–16.
25. *Сирота А. А., Дрюченко М. А., Иванков А. Ю.* Стегоанализ цифровых изображений с использованием методов поверхностного и глубокого машинного обучения: известные подходы и новые решения // Вестник Воронежского гос. ун-та. Сер. Системный анализ и информационные технологии. 2021. № 1. С. 33–53.
26. *Kutter M., Jordan F., and Bossen F* Digital signature of color images using amplitude modulation // Proc. SPIE. 1997. P. 518–526.

27. *Zhao J. and Koch E* Embedding robust labels into images for copyright protection // Proc. Intern. Congress Intellectual Property Rights for Specialized Information, Knowledge and New Technologies. Vienna, August 1995. P. 242–251.
28. *Zhang X. P. and Wang S. Z* Efficient steganographic embedding by exploiting modification direction // IEEE Commun. Let. 2006. V. 10. No. 11. P. 781–783.
29. *Paul G., Davidson I., Mukherjee I., and Ravi S. S.* Keyless dynamic optimal multi-bit image steganography using energetic pixels // Multimedia Tools Appl. 2017. V. 76. P. 7445–7471.
30. `http://dde.binghamton.edu/download/stego\_algorithms/` — Digital Data Embedding Laboratory Department of Electrical and Computer Engineering SUNY Binghamton.

REFERENCES

1. *Sheluhin O. I.* Steganografiya. Algoritmy i programmnaya realizatsiya. [Steganography. Algorithms and Their Implementation]. Moscow, Goryachaya liniya — Telekom, 2017. 592 p. (in Russian)
2. *Czaplewski B.* Current trends in the field of steganalysis and guidelines for constructions of new steganalysis schemes. Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne, 2017, no. 10, pp. 1121–1125.
3. *Lyu S. and Farid H.* Detecting hidden messages using higher-order statistics and support vector machines. Intern. Workshop Inform. Hiding, Berlin; Heidelberg, Springer, 2002, pp. 340–354. `https://farid.berkeley.edu/downloads/publications/ih02.pdf`.
4. *Lyu S. and Farid H.* Steganalysis using color wavelet statistics and one-class support vector machines. // Proc. SPIE, California, USA, 2004, pp. 35–45. `https://www.researchgate.net/publication/221011180\_Steganalysis\_using\_color\_wavelet\_statistics\_and\_one-class\_support\_vector\_machines`.
5. *Pevny T., Bas P., and Fridrich J.* Steganalysis by subtractive pixel adjacency matrix. IEEE Trans. Inform. Forensics Security, 2010, vol. 5, no. 2, pp. 215–224.
6. *Fridrich J.* Rich models for steganalysis of digital images. IEEE Trans. Inform. Forensics Security, 2012, vol. 7, no. 3, pp. 868–882.
7. *Holub V. and Fridrich J.* Random projections of residuals for digital image steganalysis. IEEE Trans. Inform. Forensics Security, 2013, vol. 8, no. 12, pp. 1996–2006.
8. *Bas P., Filler T., and Pevny T.* Break our steganographic system the ins and outs of organizing BOSS. LNCS, 2011, vol. 6958, pp. 59–70.
9. `http://www.lirmm.fr/chaumont/PPG-LIRMM-COLOR.html` — PPG-LIRMM-COLOR base.
10. *Pevny T., Filler T., and Bas P.* Using high-dimensional image models to perform highly undetectable steganography. LNCS, 2010, vol. 6387, pp. 161–177.
11. *Holub V. and Fridrich J.* Digital image steganography using universal distortion. Proc. 1st ACM Workshop IHMMSec, ACM, 2013, pp. 59–68.
12. *Holub V. and Fridrich J.* Designing steganographic distortion using directional filters. Proc. 4th IEEE Intern. Workshop WIFS, 2012, pp. 234–239.
13. *Kodovsky J., Fridrich J., and Holub V.* Ensemble classifiers for steganalysis of digital media. IEEE Trans. Inform. Forensics Security, 2010, vol. 7, no. 2, pp. 434–444.
14. *Monarev V. A. and Pestunov A. I.* Effektivnoye obnaruzheniye steganograficheski skrytoy informatsii posredstvom integral'nogo klassifikatora na osnove szhatiya dannykh [Efficient steganography detection by means of compression-based integral classifier]. Prikladnaya Diskretnaya Matematika, 2018, no. 40, pp. 59–71. (in Russian)
15. *Tabares-Soto R., Ramos-Pollan R., and Isaza G* Deep learning applied to steganalysis of digital images: A systematic review. IEEE Access, 2019, vol. 7, pp. 68970–68990.

16. *Qian Y., Dong J., Wang W., and Tan T.* Deep learning for steganalysis via convolutional neural networks. Proc. Int. Symp. Electron. Imag., 2015, vol. 9409, Art. no. 94090J.
17. *Yedroudj M., Comby F., and Chaumont M* Yedrouj-Net: An efficient CNN for spatial steganalysis. Proc. IEEE Intern. Conf. Acoustics, Speech Signal Processing, 2018, pp. 2092–2096.
18. *Zhang R., Zhu F., Liu J., and Liu G* Efficient Feature Learning and Multisizeimage Steganalysis Based on CNN. `https://arxiv.org/pdf/1807.11428.pdf`. 2018.
19. *Polunin A. A. and Yandashevskaya E. A.* Ispol'zovaniye apparata svertochnykh neyronnykh setey dlya stegoanaliza tsifrovykh izobrazheniy [Using of convolutional neural networks for steganalysis of digital images]. Proc. ISP RAS, 2020, vol. 32, no. 4, pp. 155–164. (in Russian)
20. *Sirota A. A. and Dryuchenko M. A.* Obobshchennyye algoritmy szhatiya izobrazheniy na fragmentakh proizvol'noy formy i ikh realizatsiya s ispol'zovaniyem iskusstvennykh neyronnykh setey [Generalized image compression algorithms for arbitrarily-shaped fragments and their implementation using artificial neural networks]. Computer Optics, 2015, no. 5, pp. 751–761. (in Russian)
21. *Dryuchenko M. A. and Sirota A. A.* Interpolation and masking effects of heteroassociative compressive transformations. J. Phys.: Conf. Ser., 2020, vol. 1902, pp. 1–10. `https://iopscience.iop.org/article/10.1088/1742-6596/1902/1/012058/pdf`.
22. *Dryuchenko M. A. and Sirota A. A.* Geteroassotsiativnye szhimayushchie preobrazovaniya tsifrovykh izobrazheniy i ikh interpoliruyushchie i maskiruyushchie svoystva [Interpolation and masking effects of heteroassociative compressive transformations]. Proc. Intern. Conf. AMCSM, Voronezh, 07–09 December 2020, pp. 312–322. (in Russian)
23. *Sirota A. A., Dryuchenko M. A., Mitrofanova E. Yu.* Metod sozdaniya tsifrovykh vodyanykh znakov na osnove geteroassotsiativnykh szhimayushchikh preobrazovaniy izobrazheniy i ego realizatsiya s ispol'zovaniem iskusstvennykh neyronnykh setey [Digital watermarking method based on heteroassociative image compression and its realization with artificial neural networks]. Computer Optics, 2018, no. 3, pp. 483–494. (in Russian)
24. *Sirota A. A., Dryuchenko M. A., and Mitrofanova E. Yu.* Neyrosetevye funktsional'nye modeli i algoritmy preobrazovaniya informatsii dlya sozdaniya tsifrovykh vodyanykh znakov [Neural network functional models and information transformation algorithms for creating digital watermarks]. Radioelectronics and Communications Systems, 2015, no. 1, pp. 3–16. (in Russian)
25. *Sirota A. A., Dryuchenko M. A., Ivankov A. Yu.* Stegoanaliz tsifrovykh izobrazheniy s ispol'zovaniem metodov poverkhnostnogo i glubokogo mashinnogo obucheniya: izvestnye podkhody i novye resheniya [Steganalysis of digital images by means of shallow and deep machine learning: existing approaches and new solutions]. Proc. Voronezh State University, Ser. Systems Analysis and Inform. Technol., 2021, no. 1, pp. 33–53. (in Russian)
26. *Kutter M., Jordan F., and Bossen F* Digital signature of color images using amplitude modulation. Proc. SPIE, 1997, pp. 518–526.
27. *Zhao J. and Koch E* Embedding robust labels into images for copyright protection. Proc. Intern. Congress Intellectual Property Rights for Specialized Information, Knowledge and New Technologies, Vienna, August 1995, pp. 242–251.
28. *Zhang X. P. and Wang S. Z* Efficient steganographic embedding by exploiting modification direction. IEEE Commun. Let., 2006, vol. 10, no. 11, pp. 781–783.
29. *Paul G., Davidson I., Mukherjee I., and Ravi S. S.* Keyless dynamic optimal multi-bit image steganography using energetic pixels. Multimedia Tools Appl., 2017, vol. 76, pp. 7445–7471.
30. `http://dde.binghamton.edu/download/stego\_algorithms/` — Digital Data Embedding Laboratory Department of Electrical and Computer Engineering SUNY Binghamton.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ НАДЁЖНОСТИ ВЫЧИСЛИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

УДК 519.718.7 DOI 10.17223/20710410/55/4

КОРОТКИЕ ЕДИНИЧНЫЕ ПРОВЕРЯЮЩИЕ ТЕСТЫ ДЛЯ СХЕМ ПРИ ПРОИЗВОЛЬНЫХ НЕИСПРАВНОСТЯХ ФУНКЦИОНАЛЬНЫХ ЭЛЕМЕНТОВ[1]

К. А. Попков

Институт прикладной математики им. М. В. Келдыша РАН, г. Москва, Россия

E-mail: kirill-formulist@mail.ru

Доказано, что любую неконстантную булеву функцию от n переменных можно реализовать неизбыточной схемой из функциональных элементов в базисе $\{\&, \oplus, \neg\}$, допускающей при $n \geqslant 3$ единичный проверяющий тест длины не более $6n - 10$ относительно произвольных неисправностей элементов.

Ключевые слова: *схема из функциональных элементов, булева функция, неисправность, единичный проверяющий тест.*

SHORT SINGLE FAULT DETECTION TESTS FOR LOGIC NETWORKS UNDER ARBITRARY FAULTS OF GATES

K. A. Popkov

Keldysh Institute of Applied Mathematics, Moscow, Russia

It was proved that one can implement any non-constant Boolean function in n variables by an irredundant logic network in the basis $\{\&, \oplus, \neg\}$, allowing, when $n \geqslant 3$, a single fault detection test with length not more than $6n - 10$ relative to arbitrary faults of gates.

Keywords: *logic network, Boolean function, fault, single fault detection test.*

Введение

Рассматривается задача синтеза легкотестируемых схем, реализующих заданные булевы функции. Логический подход к тестированию электрических схем предложен С. В. Яблонским и И. А. Чегис в [1]; этот подход также применим к тестированию схем из функциональных элементов (СФЭ; [2 – 4]). Пусть имеется СФЭ S с одним выходом, реализующая булеву функцию $f(\tilde{x}^n)$, где $\tilde{x}^n = (x_1, \ldots, x_n)$. Представим, что под воздействием некоторого источника неисправностей один из элементов схемы S может перейти в неисправное состояние. В результате данная схема вместо исходной функции $f(\tilde{x}^n)$ будет реализовывать некоторую булеву функцию $g(\tilde{x}^n)$, вообще

[1]Работа выполнена при поддержке гранта РНФ, проект № 19-71-30004.

говоря, отличную от f. Все такие функции $g(\tilde{x}^n)$ называются *функциями неисправности* схемы S. *Единичным проверяющим тестом* (ЕПТ) для схемы S называется такое множество T наборов значений переменных $x_1, \ldots, x_n$, что для любой отличной от $f(\tilde{x}^n)$ функции неисправности $g(\tilde{x}^n)$ данной схемы в T найдётся набор $\tilde{\sigma}$, на котором $f(\tilde{\sigma}) \neq g(\tilde{\sigma})$. Число наборов в T называется *длиной* теста. В качестве тривиального ЕПТ длины 2^n для схемы S всегда можно взять множество, состоящее из всех двоичных наборов длины n. Единичные проверяющие тесты обычно рассматривают для *неизбыточных схем* [4, с. 110–111], т. е. для таких схем, в которых любая допустимая неисправность любого одного элемента приводит к функции неисправности, отличной от исходной функции, реализуемой данной схемой.

Любое множество булевых функций будем называть *базисом*.

Ранее в качестве неисправностей функциональных элементов традиционно рассматривались константные либо инверсные неисправности на входах и/или выходах элементов. Константная неисправность на входе (выходе) функционального элемента означает, что значение на этом его входе (выходе) становится равно некоторой булевой константе. Неисправности на входах и/или выходах элементов называются однотипными константными типа p, если эта константа одна и та же для каждого неисправного входа/выхода элемента и равна p, и произвольными константными, если эта константа может быть равна как 0, так и 1 для каждого неисправного входа/выхода элемента независимо от неисправностей других входов/выходов элементов. Инверсная неисправность на входе (выходе) функционального элемента означает, что значение на этом входе (выходе) меняется на противоположное по сравнению со случаем, когда данный элемент исправен.

Основные результаты, касающиеся ЕПТ для схем из функциональных элементов при указанных неисправностях, получены в работах [5–20]; более подробный обзор см. в работе автора [21]. Отметим, что в [5–7, 11, 13–20] установлены константные верхние оценки минимально возможных длин ЕПТ при реализации произвольных булевых функций от n переменных неизбыточными СФЭ в различных базисах при различных неисправностях элементов.

В настоящей работе, так же как и в [21], будем рассматривать произвольные неисправности функциональных элементов: допустим, что каждый неисправный элемент E вместо исходной приписанной ему булевой функции $\varphi_E(\tilde{x}^m)$ реализует произвольную другую булеву функцию $\varphi'_E(\tilde{x}^m)$ (от своих входов). Тогда элемент E может находиться в любом (неизменном в ходе тестирования) из $2^{2^m} - 1$ неисправных состояний, характеризующихся функцией $\varphi'_E(\tilde{x}^m)$. Например, в случае константной неисправности типа p (в случае инверсной неисправности) на выходе этого элемента имеем $\varphi'_E \equiv p$ (соответственно $\varphi'_E = \overline{\varphi}_E$), а в случае константной неисправности типа p (инверсной неисправности) на входе элемента E, отвечающем переменной x_1, имеем $\varphi'_E(\tilde{x}^m) = \varphi_E(p, x_2, \ldots, x_m)$ (соответственно $\varphi'_E(\tilde{x}^m) = \varphi_E(\overline{x}_1, x_2, \ldots, x_m)$).

В соответствии с [22, с. 105] будем говорить, что СФЭ *содержит k фиктивных входных переменных и реализует функцию $f(\tilde{x}^n)$*, если данная схема содержит k входных переменных, отличных от переменных $x_1, \ldots, x_n$, и реализует булеву функцию, не зависящую существенно от этих k переменных и равную функции $f(\tilde{x}^n)$. Будем также предполагать, что все наборы из любого ЕПТ для такой схемы имеют длину $n + k$ (по общему числу её входных переменных).

В [21] установлено, что любую неконстантную булеву функцию $f(\tilde{x}^n)$ можно реализовать неизбыточной СФЭ в базисе $\{\&, \oplus, \neg\}$, содержащей одну фиктивную входную переменную и допускающей ЕПТ длины не более $2n + 3$.

Введём обозначения $\tilde{0}^l = \underbrace{0,\ldots,0}_{l}$, $\tilde{1}^l = \underbrace{1,\ldots,1}_{l}$, где $l \in \mathbb{N} \cup \{0\}$ (в случае $l = 0$ они обозначают пустую строку: например, $(\tilde{0}^n, \tilde{1}^0) = (\tilde{0}^n)$), а также

$$x^\sigma = \begin{cases} x, & \text{если } \sigma = 1, \\ \overline{x}, & \text{если } \sigma = 0, \end{cases}$$

где $x \in \{0, 1\}$.

Две различные неисправности в произвольной СФЭ назовём *равносильными*, если соответствующие им функции неисправности равны друг другу.

Вместо «вход схемы S, отвечающий переменной x_i» для краткости будем писать «вход «x_i» схемы S».

1. Формулировка основной теоремы

Рассмотрим базис $B = \{\&, \oplus, \neg\}$. Любой функциональный элемент, реализующий функцию вида $x\&y$ (вида $x \oplus y$, $\overline{x}$) от своих входов, будем называть *конъюнктором* (соответственно *сумматором*, *инвертором*). Вход любого конъюнктора или сумматора, отвечающий переменной x, будем считать *левым*, а другой вход — *правым*.

Теорема 1. Любую неконстантную булеву функцию $f(\tilde{x}^n)$ можно реализовать неизбыточной СФЭ в базисе B, допускающей при $n \geqslant 3$ ЕПТ длины не более $6n - 10$.

Замечание 1. В отличие от основной теоремы работы [21], фигурирующая в формулировке теоремы 1 СФЭ не содержит фиктивных входных переменных.

Замечание 2. Константные булевы функции нельзя реализовать неизбыточными СФЭ в базисе B. Действительно, константная неисправность типа α на выходе выходного элемента любой схемы в базисе B, реализующей функцию $f(\tilde{x}^n) \equiv \alpha$, где $\alpha \in \{0, 1\}$, приводит к функции неисправности $g(\tilde{x}^n) \equiv \alpha$.

Для доказательства теоремы 1 требуется ввести следующее понятие.

1.1. Регулярные булевы функции

Рассмотрим произвольную булеву функцию $f_n(\tilde{x}^n)$, где $n \geqslant 3$. Её можно представить, причём единственным образом, в виде

$$f_n(\tilde{x}^n) = x_n f_{n-1}(\tilde{x}^{n-1}) \oplus f'_{n-1}(\tilde{x}^{n-1}),$$

где f_{n-1} и f'_{n-1} — булевы функции от переменных $x_1, \ldots, x_{n-1}$. Действительно, подставив в последнее равенство вместо x_n поочерёдно 0 и 1, получим, что

$$f'_{n-1}(\tilde{x}^{n-1}) = f_n(x_1, \ldots, x_{n-1}, 0) \tag{1}$$

и $f_n(x_1, \ldots, x_{n-1}, 1) = f_{n-1}(\tilde{x}^{n-1}) \oplus f'_{n-1}(\tilde{x}^{n-1}) = f_{n-1}(\tilde{x}^{n-1}) \oplus f_n(x_1, \ldots, x_{n-1}, 0)$, откуда

$$f_{n-1}(\tilde{x}^{n-1}) = f_n(x_1, \ldots, x_{n-1}, 1) \oplus f_n(x_1, \ldots, x_{n-1}, 0). \tag{2}$$

Аналогично функцию $f_{n-1}(\tilde{x}^{n-1})$ в случае $n - 1 \geqslant 3$ можно единственным образом представить в виде

$$f_{n-1}(\tilde{x}^{n-1}) = x_{n-1} f_{n-2}(\tilde{x}^{n-2}) \oplus f'_{n-2}(\tilde{x}^{n-2}),$$

и вообще, для любого $i \in \{2, \ldots, n-1\}$, если уже определена функция $f_{i+1}(\tilde{x}^{i+1})$, то её можно единственным образом представить в виде

$$f_{i+1}(\tilde{x}^{i+1}) = x_{i+1} f_i(\tilde{x}^i) \oplus f'_i(\tilde{x}^i),$$

где $f_i(\tilde{x}^i)$ и $f'_i(\tilde{x}^i)$ — булевы функции. Указанным образом по функции $f_n(\tilde{x}^n)$ однозначно определяются булевы функции $f_i(\tilde{x}^i)$ и $f'_i(\tilde{x}^i)$ для каждого $i = 2, \ldots, n-1$.

Назовём булеву функцию $f_n(\tilde{x}^n)$, $n \geqslant 3$, *регулярной*, если для любого $i \in \{2, \ldots, n-1\}$ выполнено хотя бы одно из трёх условий: а) функция $f_i(\tilde{x}^i)$ является константной; б) функция $f'_i(\tilde{x}^i)$ является константной; в) функция $f_i(\tilde{x}^i)$ отлична от каждой из функций $f'_i(\tilde{x}^i), \overline{f'_i}(\tilde{x}^i)$.

Пример 1. Никакая функция $f_n(\tilde{x}^n)$, $n \geqslant 3$, представимая в виде

$$f_n(\tilde{x}^n) = \overline{x}_n g(\tilde{x}^{n-1}) \oplus c,$$

где g — произвольная неконстантная булева функция от переменных $x_1, \ldots, x_{n-1}$ и $c \in \{0, 1\}$, не является регулярной. Действительно,

$$f_n(\tilde{x}^n) = (x_n \oplus 1) g(\tilde{x}^{n-1}) \oplus c = x_n g(\tilde{x}^{n-1}) \oplus (g(\tilde{x}^{n-1}) \oplus c),$$

поэтому $f_{n-1}(\tilde{x}^{n-1}) = g(\tilde{x}^{n-1})$ и $f'_{n-1}(\tilde{x}^{n-1}) = g(\tilde{x}^{n-1}) \oplus c$, а тогда

$$f'_{n-1}(\tilde{x}^{n-1}) = \begin{cases} f_{n-1}(\tilde{x}^{n-1}), & \text{если } c = 0, \\ \overline{f_{n-1}}(\tilde{x}^{n-1}), & \text{если } c = 1. \end{cases}$$

Пример 2. Функция $f_4(\tilde{x}^4) = x_1\overline{x}_3 \oplus \overline{x}_1\overline{x}_2\overline{x}_4$ является регулярной. Действительно, пользуясь (1) и (2), получаем

$$\begin{gathered} f'_3(\tilde{x}^3) = x_1\overline{x}_3 \oplus \overline{x}_1\overline{x}_2, \\ f_3(\tilde{x}^3) = x_1\overline{x}_3 \oplus (x_1\overline{x}_3 \oplus \overline{x}_1\overline{x}_2) = \overline{x}_1\overline{x}_2, \\ f'_2(\tilde{x}^2) = \overline{x}_1\overline{x}_2, \\ f_2(\tilde{x}^2) = \overline{x}_1\overline{x}_2 \oplus \overline{x}_1\overline{x}_2 \equiv 0, \end{gathered}$$

откуда следует, что функция f_3 отлична от функций f'_3 и $\overline{f'_3}$, а функция f_2 является константной.

Для удобства будем считать, что все булевы функции, формально зависящие от одной или двух переменных, являются регулярными. Легко видеть, что если булева функция $f_n(\tilde{x}^n)$, $n \geqslant 3$, регулярна, то и все функции $f_i(\tilde{x}^i)$, где $i = 2, \ldots, n-1$, регулярны.

1.2. Доказательство основной теоремы

Справедливость теоремы 1 очевидным образом вытекает из следующих трёх лемм.

Лемма 1. Любую неконстантную регулярную булеву функцию $f_n(\tilde{x}^n)$ можно реализовать неизбыточной СФЭ в базисе B, допускающей при $n \geqslant 3$ ЕПТ длины не более $6n - 10$.

Лемма 2. Для любой неконстантной булевой функции $f(\tilde{x}^n)$ существуют такие $\sigma_1, \ldots, \sigma_n \in \{0, 1\}$, что функция $f(x_1^{\sigma_1}, \ldots, x_n^{\sigma_n})$ регулярна и отлична от констант.

Лемма 3. Если неконстантная булева функция $f(\tilde{x}^n)$ и числа $\sigma_1, \ldots, \sigma_n \in \{0, 1\}$ таковы, что функцию $f(x_1^{\sigma_1}, \ldots, x_n^{\sigma_n})$ можно реализовать неизбыточной СФЭ в базисе B, допускающей при $n \geqslant 3$ ЕПТ длины не более $6n - 10$, то функцию $f(\tilde{x}^n)$ можно реализовать СФЭ с теми же свойствами.

Сформулируем одно вспомогательное утверждение.

Лемма 4. Любую неконстантную булеву функцию $f(\tilde{x}^n)$ можно реализовать неизбыточной СФЭ в базисе B, содержащей одну фиктивную входную переменную и допускающей ЕПТ T длины не более $2n+4$, подмножеством которого является множество $T^0_{n+1} \cup \{\tilde{\sigma}\}$, где $T^0_{n+1} = \{(\tilde{0}^r, \tilde{1}^{n+1-r}), (\tilde{1}^s, 0, \tilde{1}^{n-s}) : r \in \{0, \ldots, n+1\}, s \in \{1, \ldots, n\}\}$, а $\tilde{\sigma}$ — некоторый такой двоичный $(n+1)$-разрядный набор, что его $(n+1)$-я компонента равна 0 и n-разрядный набор $\tilde{\sigma}'$, получающийся из набора $\tilde{\sigma}$ удалением $(n+1)$-й компоненты, удовлетворяет условию $f(\tilde{\sigma}') \neq f(\tilde{0}^n)$.

Доказательство основано на доказательстве теоремы 1 работы [21], которое разбивается на два случая. В случае 1 доказано [22, с. 94–95], что функцию $f(\tilde{x}^n)$ можно реализовать неизбыточной СФЭ в базисе B, содержащей одну фиктивную входную переменную и допускающей ЕПТ T^0_{n+1} (длины $2n+2$). Функция f отлична от константы $f(\tilde{0}^n)$, поэтому существует такой двоичный n-разрядный набор $\tilde{\sigma}'$, что $f(\tilde{\sigma}') \neq f(\tilde{0}^n)$. Пусть $\tilde{\sigma}$ — набор, получающийся из набора $\tilde{\sigma}'$ добавлением $(n+1)$-й компоненты, равной 0. Тогда утверждение леммы 4 справедливо при $T = T^0_{n+1} \cup \{\tilde{\sigma}\}$.

В случае 2 идёт ссылка (см. [22, с. 95]) на лемму 2 работы [21]. В формулировке этой леммы (см. [22, с. 89]) фигурирует двоичный $(n+1)$-разрядный набор $\tilde{\sigma}_{K_1}$, определяемый в [22, с. 88]. Из его определения и соотношения [21, (3)] следует, что $(n+1)$-я компонента набора $\tilde{\sigma}_{K_1}$ равна 0. Обозначим через $\tilde{\sigma}'_{K_1}$ набор, получающийся из набора $\tilde{\sigma}_{K_1}$ удалением $(n+1)$-й компоненты. Тогда из определения данных наборов и соотношения [21, (2)] вытекает, что

$$f(\tilde{0}^n) = K_1(\tilde{0}^n) \oplus \ldots \oplus K_m(\tilde{0}^n) \oplus c = \underbrace{0 \oplus 0}_{m} \oplus c = c,$$

$$f(\tilde{\sigma}'_{K_1}) = K_1(\tilde{\sigma}'_{K_1}) \oplus K_2(\tilde{\sigma}'_{K_1}) \oplus \ldots \oplus K_m(\tilde{\sigma}'_{K_1}) \oplus c = 1 \oplus \underbrace{0 \oplus 0}_{m-1} \oplus c = \overline{c},$$

поэтому $f(\tilde{\sigma}'_{K_1}) \neq f(\tilde{0}^n)$. В таком случае утверждение леммы 4 следует из [21, лемма 2] при $\tilde{\sigma} = \tilde{\sigma}_{K_1}$. ■

Доказательство леммы 1. Пусть сначала $n = 1$. Тогда $f_1(x_1) = x_1$ или $f_1(x_1) = \overline{x}_1$. В первом случае функцию f_1 можно реализовать СФЭ, не содержащей функциональных элементов, а во втором — содержащей один инвертор. Множество функций неисправности первой схемы пусто, а список возможных неисправностей второй схемы ограничивается константной неисправностью типа 0, константной неисправностью типа 1 и инверсной неисправностью на выходе инвертора, при которых на выходе схемы вместо «правильной» функции $\overline{x}_1$ реализуются функции соответственно 0, 1, x_1. Поэтому обе рассматриваемые схемы неизбыточны, что и требовалось доказать.

Далее будем считать, что $n \geqslant 2$. Докажем индукцией по n более сильное утверждение: любую неконстантную регулярную булеву функцию $f_n(\tilde{x}^n)$ можно реализовать неизбыточной СФЭ S_n в базисе B, допускающей ЕПТ T_n, подмножеством которого является множество $T^0_n = \{(\tilde{0}^r, \tilde{1}^{n-r}), (\tilde{1}^s, 0, \tilde{1}^{n-s-1}) : r \in \{0, \ldots, n\};\ s \in \{1, \ldots, n-1\}\}$, причём $|T_n| \leqslant 6n - 10$ при $n \geqslant 3$.

Б а з а и н д у к ц и и: $n = 2$. Надо доказать, что любую неконстантную булеву функцию $f_2(x_1, x_2)$ можно реализовать неизбыточной СФЭ в базисе B, допускающей ЕПТ $T_2 = \{(0,0), (0,1), (1,0), (1,1)\}$. Любая СФЭ, реализующая функцию $f_2(x_1, x_2)$, допускает тривиальный ЕПТ T_2, поэтому достаточно установить, что эту функцию можно реализовать неизбыточной СФЭ S_2 в базисе B. Заметим, что $f_2(x_1, x_2) \in \in \{x_1, x_2, \overline{x}_1, \overline{x}_2, (x_1^{\sigma_1} \& x_2^{\sigma_2})^c, (x_1 \oplus x_2)^c : \sigma_1, \sigma_2, c \in \{0,1\}\}$. В случае $f_2(x_1, x_2) \in \{x_1, x_2\}$

в качестве S_2 можно взять схему, не содержащую функциональных элементов, а в случае $f_2(x_1, x_2) \in \{x_1, x_2\}$ — схему, состоящую из одного инвертора.

Пусть $f_2(x_1, x_2) = (x_1 \oplus x_2)^c$, где $c \in \{0, 1\}$. Реализуем функцию f_2 схемой S_2, состоящей из одного сумматора E, на входы которого подаются переменные x_1 и x_2, и — в случае $c = 0$ — одного инвертора, вход которого соединён с выходом этого сумматора. При произвольной неисправности элемента E, при которой он реализует некоторую булеву функцию $\varphi'_E(x_1, x_2)$, отличную от $\varphi_E(x_1, x_2) = x_1 \oplus x_2$, на выходе схемы S_2, очевидно, возникнет функция $(\varphi'_E(x_1, x_2))^c$, отличная от $(x_1 \oplus x_2)^c$; в случае же $c = 0$ и неисправности инвертора на выходе схемы S_2 возникнет одна из функций $0, 1, x_1 \oplus x_2$, отличная от функции $f_2(x_1, x_2) = \overline{x_1 \oplus x_2}$. Таким образом, схема S_2 неизбыточна.

Пусть, наконец, $f_2(x_1, x_2) = (x_1^{\sigma_1} \& x_2^{\sigma_2})^c$, где $\sigma_1, \sigma_2, c \in \{0, 1\}$. Сначала для каждого такого $j \in \{1, 2\}$, что $\sigma_j = 0$, реализуем функцию $x_j^{\sigma_j} = \overline{x}_j$ с использованием одного инвертора I_j, на вход которого подаётся переменная x_j. Далее реализуем функцию f_2 схемой S_2, содержащей один конъюнктор E, на входы которого подаются функции $x_1^{\sigma_1}$ и $x_2^{\sigma_2}$, и — в случае $c = 0$ — инвертор I, вход которого соединён с выходом этого конъюнктора; каждая функция $x_j^{\sigma_j}$, $j = 1, 2$, берётся со входа x_j схемы в случае $\sigma_j = 1$ и с выхода инвертора I_j в случае $\sigma_j = 0$. Легко видеть, что любая константная либо инверсная неисправность на выходе инвертора I_j (при наличии этого инвертора) равносильна такой же неисправности на входе конъюнктора E, отвечающего переменной x_j, а других неисправностей у инвертора быть не может. В случае произвольной неисправности элемента E, при которой он реализует некоторую булеву функцию $\varphi'_E(x_1, x_2)$ (от своих входов), отличную от $\varphi_E(x_1, x_2) = x_1 \& x_2$, на выходе схемы S_2 возникнет функция $(\varphi'_E(x_1^{\sigma_1}, x_2^{\sigma_2}))^c$, отличная от $(\varphi_E(x_1^{\sigma_1}, x_2^{\sigma_2}))^c = (x_1^{\sigma_1} \& x_2^{\sigma_2})^c$; в случае же $c = 0$ и неисправности инвертора на выходе схемы S_2 возникнет одна из функций $0, 1, x_1^{\sigma_1} \& x_2^{\sigma_2}$, отличная от функции $f_2(x_1, x_2) = \overline{x_1^{\sigma_1} \& x_2^{\sigma_2}}$. Таким образом, схема S_2 неизбыточна. База индукции доказана.

П р е д п о л о ж е н и е и ш а г и н д у к ц и и: пусть требуемое утверждение доказано для $n = t$, где $t \geqslant 2$; докажем его для $n = t + 1$. Надо доказать, что любую неконстантную регулярную булеву функцию $f_{t+1}(\tilde{x}^{t+1})$ можно реализовать неизбыточной СФЭ S_{t+1} в базисе B, допускающей ЕПТ T_{t+1} длины не более $6t - 4$, подмножеством которого является множество $T_{t+1}^0 = \big\{(\tilde{0}^r, \tilde{1}^{t+1-r}), (\tilde{1}^s, 0, \tilde{1}^{t-s}) : r \in \{0, \ldots, t+1\};\ s \in \in \{1, \ldots, t\}\big\}$. Функцию f_{t+1} можно единственным образом представить в виде

$$f_{t+1}(\tilde{x}^{t+1}) = x_{t+1} f_t(\tilde{x}^t) \oplus f'_t(\tilde{x}^t), \tag{3}$$

где f_t и f'_t — булевы функции. Введём обозначение

$$a = \begin{cases} 1, & \text{если обе функции } f_t, f'_t \text{ неконстантные и либо } f_t \leqslant f'_t, \text{ либо } f_t \leqslant \overline{f'_t}, \\ 0 & \text{иначе} \end{cases}$$

(неравенство $h_1 \leqslant h_2$, где $h_1(\tilde{x}^t)$, $h_2(\tilde{x}^t)$ — булевы функции, означает, что $h_1(\tilde{\sigma}) \leqslant h_2(\tilde{\sigma})$ для любого двоичного t-разрядного набора $\tilde{\sigma}$). Перепишем равенство (3) в виде

$$f_{t+1}(\tilde{x}^{t+1}) = x_{t+1}(f_t(\tilde{x}^t) \oplus a) \oplus f'_t(\tilde{x}^t) \oplus a x_{t+1}. \tag{4}$$

Построим СФЭ S_{t+1} в базисе B, реализующую при отсутствии в ней неисправностей функцию $f_{t+1}(\tilde{x}^{t+1})$, в соответствии с представлением (4) (рис. 1). Из регулярности функции $f_{t+1}(\tilde{x}^{t+1})$ следует регулярность функции $f_t(\tilde{x}^t)$. Если последняя функция

отлична от констант, то её по предположению индукции можно реализовать неизбыточной СФЭ S_t в базисе B, допускающей ЕПТ T_t, подмножеством которого является множество $T_t^0 = \{(\tilde{0}^r, \tilde{1}^{t-r}), (\tilde{1}^s, 0, \tilde{1}^{t-s-1}) : r \in \{0, \ldots, t\};\ s \in \{1, \ldots, t-1\}\}$, причём

$$|T_t| \leqslant 6t - 10 \tag{5}$$

при $t \geqslant 3$. В случае $a = 0$ соединим вход «x_{t+1}» схемы S_{t+1} и выход схемы S_t с левым и правым входами конъюнктора $E^{\&}_{t+1}$ соответственно; в случае $a = 1$ соединим выход схемы S_t с входом инвертора I_{t+1} и далее соединим вход «x_{t+1}» схемы S_{t+1} и выход инвертора I_{t+1} с левым и правым входами конъюнктора $E^{\&}_{t+1}$ соответственно. Обозначим построенную к настоящему моменту схему, выход которой совпадает с выходом элемента $E^{\&}_{t+1}$, через S^1_{t+1}. Будем считать, что в случае $f_t \equiv 1$ подсхема S_t схемы S_{t+1} пуста, а подсхема S^1_{t+1} не содержит функциональных элементов и её выход совпадает со входом «x_{t+1}» схемы S_{t+1}; в случае $f_t \equiv 0$ подсхемы S_t и S^1_{t+1} пусты (отметим, что в каждом из этих случаев $a = 0$). Легко видеть, что во всех перечисленных в данном абзаце случаях на выходе подсхемы S^1_{t+1}, если она непуста, реализуется функция $x_{t+1}(f_t(\tilde{x}^t) \oplus a)$.

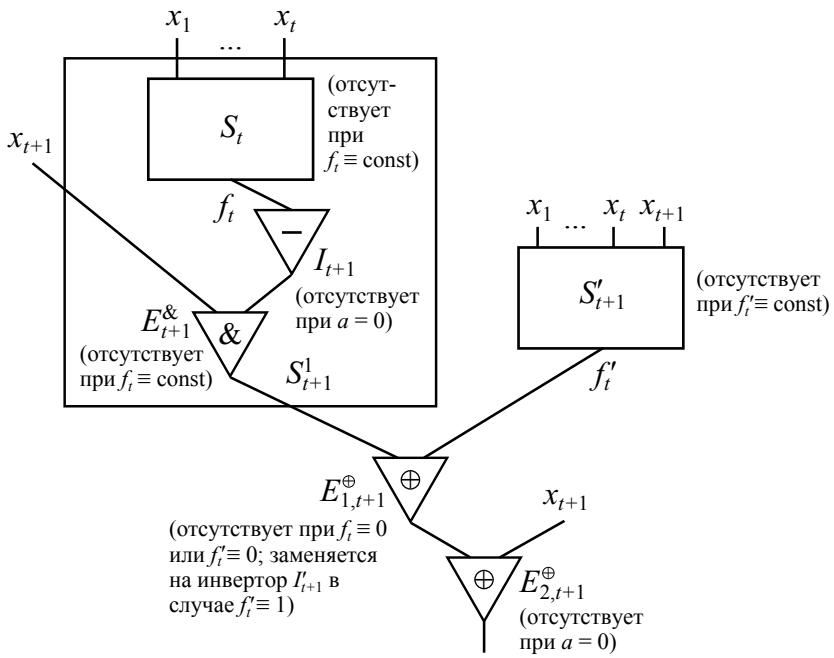


Рис. 1. Схема S_{t+1}

Далее, если функция $f'_t(\tilde{x}^t)$ отлична от констант, то её по лемме 4 можно реализовать неизбыточной СФЭ S'_{t+1} в базисе B, содержащей одну фиктивную входную переменную x_{t+1} и допускающей ЕПТ T'_{t+1} длины не более $2t+4$, подмножеством которого является множество $T^0_{t+1} \cup \tilde{\rho}'_{t,0}$, где $\tilde{\rho}'_{t,0}$ — некоторый такой двоичный $(t+1)$-разрядный набор, что его $(t+1)$-я компонента равна 0 и t-разрядный набор $\tilde{\rho}'_t$, получающийся из $\tilde{\rho}'_{t,0}$ удалением $(t+1)$-й компоненты, удовлетворяет условию $f'_t(\tilde{\rho}'_t) \neq f'_t(\tilde{0}^t)$. В случае $f_t \not\equiv 0$ соединим выходы подсхем S^1_{t+1} и S'_{t+1} схемы S с левым и правым входами сумматора $E^{\oplus}_{1,t+1}$ соответственно. При $a = 0$ выход элемента $E^{\oplus}_{1,t+1}$ объявим выходом схемы S_{t+1}; при $a = 1$ соединим выход этого элемента и вход «x_{t+1}» схемы S_{t+1} с левым и правым входами сумматора $E^{\oplus}_{2,t+1}$ соответственно, выход которого объявим выходом данной схемы. В случае $f_t \equiv 0$ будем считать, что схема S_{t+1} совпадает со схемой S'_{t+1}.

Если $f'_t \equiv 0$ или $f'_t \equiv 1$, то $f_t \not\equiv 0$ (иначе в силу (3) функция $f_{t+1}(\tilde{x}^{t+1})$ была бы константной), $a = 0$ и подсхема S'_{t+1} предполагается пустой. В случае $f'_t \equiv 0$ будем считать, что схема S_{t+1} совпадает с подсхемой S^1_{t+1}, а в случае $f'_t \equiv 1$ соединим выход подсхемы S^1_{t+1} со входом инвертора I'_{t+1}, выход которого объявим выходом схемы S_{t+1}. Нетрудно проверить, что во всех случаях на выходе этой схемы реализуется функция $x_{t+1}(f_t(\tilde{x}^t) \oplus a) \oplus f'_t(\tilde{x}^t) \oplus a x_{t+1}$, равная $f_{t+1}(\tilde{x}^{t+1})$ в силу (4).

Для доказательства шага индукции требуется показать, что построенная схема S_{t+1} неизбыточна и допускает ЕПТ T_{t+1} длины не более $6t - 4$, подмножеством которого является множество T^0_{t+1}.

Докажем следующее утверждение $(*)$: в случае $f_t \not\equiv 0$ схема S^1_{t+1} неизбыточна и допускает ЕПТ T^1_{t+1}, подмножеством которого является множество T^0_{t+1}, причём $|T^1_{t+1}| \leqslant 6t - 7$ при $t \geqslant 3$.

Если $f_t \equiv 1$, то по построению схема S^1_{t+1} не содержит функциональных элементов, поэтому у неё нет ни одной функции неисправности, она неизбыточна и любое множество двоичных наборов длины $t + 1$, в том числе $T^1_{t+1} = T^0_{t+1}$, является для неё ЕПТ, причём

$$|T^1_{t+1}| = |T^0_{t+1}| = 2t + 2 < 6t - 7$$

при $t \geqslant 3$. Пусть функция $f_t(\tilde{x}^t)$ отлична от констант. Выше для этого случая определено множество T_t. Пусть $\hat{T}_{t+1}$ — множество, состоящее из наборов $(\tilde{0}^{t+1})$ и $(\tilde{1}^t, 0)$, а также наборов, получающихся добавлением к каждому набору из множества T_t компоненты с номером $t + 1$, равной единице. Тогда

$$|\hat{T}_{t+1}| = |T_t| + 2, \tag{6}$$

а из соотношения $T^0_t \subseteq T_t$ очевидным образом вытекает, что

$$T^0_{t+1} \subseteq \hat{T}_{t+1} \tag{7}$$

(см. определения множеств T^0_t и T^0_{t+1}).

Пусть $\tilde{\sigma}_t$ — произвольный двоичный набор длины t, для которого $f_t(\tilde{\sigma}_t) \neq f_t(\tilde{0}^t)$; такой набор найдётся в силу отличия функции f_t от констант. Обозначим через $\tilde{\sigma}_{t,0}$ набор, получающийся из $\tilde{\sigma}_t$ добавлением $(t+1)$-й компоненты, равной нулю, и положим $T^1_{t+1} = \hat{T}_{t+1} \cup \{\tilde{\sigma}_{t,0}\}$. При $t \geqslant 3$ с учётом (5) и (6) имеем

$$|T^1_{t+1}| \leqslant |\hat{T}_{t+1}| + 1 = |T_t| + 3 \leqslant 6t - 7.$$

Отсюда, а также из соотношений (7) и $\hat{T}_{t+1} \subseteq T^1_{t+1}$ вытекает, что для доказательства утверждения $(*)$ достаточно доказать следующее утверждение: схема S^1_{t+1} неизбыточна, а множество T^1_{t+1} является для неё ЕПТ.

При переходе в произвольное неисправное состояние произвольного одного элемента E_t из подсхемы S_t схемы S^1_{t+1} значение, выдаваемое этой подсхемой хотя бы на одном наборе $\tilde{\pi}_t$ из множества T_t, изменится с 0 на 1 или наоборот, поскольку T_t — ЕПТ для неизбыточной схемы S_t. Пусть $\tilde{\pi}_{t,1}$ — набор, получающийся из набора $\tilde{\pi}_t$ добавлением $(t + 1)$-й компоненты, равной единице. Тогда

$$\tilde{\pi}_{t,1} \in \hat{T}_{t+1} \subseteq T^1_{t+1}$$

в силу определений множеств $\hat{T}_{t+1}, T^1_{t+1}$. При подаче на входы схемы S^1_{t+1} набора $\tilde{\pi}_{t,1}$ на её вход «x_{t+1}» поступает значение 1, а на вход подсхемы S_t — набор $\tilde{\pi}_t$. Поэтому

при переходе элемента E_t в указанное неисправное состояние набор, поступающий в схеме S^1_{t+1} на входы конъюнктора $E^{\&}_{t+1}$, изменится с $(1,0)$ на $(1,1)$ или наоборот вне зависимости от наличия в ней инвертора I_{t+1}, а тогда изменится значение на выходе этого конъюнктора, т. е. на выходе схемы S^1_{t+1}. Тем самым показано, что любая неисправность любого одного элемента из подсхемы S_t схемы S^1_{t+1} обнаруживается хотя бы на одном наборе из множества T^1_{t+1}.

Рассмотрим теперь произвольную неисправность конъюнктора $E^{\&}_{t+1}$. Пусть данный элемент при ней реализует вместо «правильной» функции $\varphi_{E^{\&}_{t+1}}(x,y) = x\&y$ некоторую другую булеву функцию $\varphi'_{E^{\&}_{t+1}}(x,y)$ (от своих входов). Обозначим через $(\alpha_{1,t+1}, \alpha_{2,t+1})$ произвольный двоичный набор, на котором значения этих функций различаются. Докажем, что при последовательной подаче на входы схемы S^1_{t+1} некоторых четырёх наборов из множества T^1_{t+1} на входы конъюнктора $E^{\&}_{t+1}$ поступают наборы $(0,0)$, $(0,1)$, $(1,0)$ и $(1,1)$. Если $f_t(\tilde{x}^t) = x_{i_t}$ для некоторого $i_t \in \{1,\ldots,t\}$, то в качестве указанных наборов можно взять $(\tilde{0}^{t+1})$, $(\tilde{1}^t, 0)$, $(\tilde{0}^t, 1)$ и $(\tilde{1}^{t+1})$ соответственно. В противном случае в подсхеме S_t обязательно содержится выходной элемент. Для каждого $p \in \{0,1\}$ константная неисправность типа $\overline{p}$ на его выходе должна обнаруживаться на каком-то наборе $\tilde{\tau}^{(p)}_t \in T_t$, откуда следует, что $f_t(\tilde{\tau}^{(p)}_t) = p$. Набор $\tilde{\tau}^{(p)}_{t,1}$, получающийся из $\tilde{\tau}^{(p)}_t$ добавлением $(t+1)$-й компоненты, равной единице, по построению принадлежит T^1_{t+1}. Тогда в качестве указанных четырёх наборов можно взять $(\tilde{0}^{t+1})$, $\tilde{\sigma}_{t,0}$, $\tilde{\tau}^{(0)}_{t,1}$ и $\tilde{\tau}^{(1)}_{t,1}$ (возможно, в другом порядке). Действительно, переменная x_{t+1}, подающаяся на левый вход конъюнктора $E^{\&}_{t+1}$, на этих наборах принимает значения $0,0,1,1$ соответственно, а функция $f_t(\tilde{x}^t) \oplus a$, подающаяся на правый вход данного конъюнктора, — значения $f_t(\tilde{0}^t) \oplus a$, $f_t(\tilde{\sigma}_t) \oplus a$, $f_t(\tilde{\tau}^{(0)}_t) \oplus a = a$, $f_t(\tilde{\tau}^{(1)}_t) \oplus a = \overline{a}$ соответственно, первые два из которых не равны друг другу в силу выбора набора $\tilde{\sigma}_t$.

При подаче на входы схемы S^1_{t+1} какого-то из четырёх выбранных наборов на входы конъюнктора $E^{\&}_{t+1}$ поступит набор $(\alpha_{1,t+1}, \alpha_{2,t+1})$. Тогда при переходе элемента $E^{\&}_{t+1}$ в рассматриваемое неисправное состояние значение на его выходе, т. е. на выходе схемы S^1_{t+1}, изменится с $\varphi_{E^{\&}_{t+1}}(\alpha_{1,t+1}, \alpha_{2,t+1})$ на $\varphi'_{E^{\&}_{t+1}}(\alpha_{1,t+1}, \alpha_{2,t+1})$ и неисправность будет обнаружена на каком-то наборе из множества T^1_{t+1}.

Наконец, любая константная либо инверсная неисправность на выходе инвертора I_{t+1} (при наличии этого инвертора) равносильна такой же неисправности на правом входе конъюнктора $E^{\&}_{t+1}$. Таким образом, любая неисправность любого одного элемента схемы S^1_{t+1} обнаруживается хотя бы на одном наборе из множества T^1_{t+1}. Это означает, что схема S^1_{t+1} неизбыточна, а множество T^1_{t+1} является для неё ЕПТ. Утверждение $(*)$ доказано.

Докажем теперь основное утверждение: схема S_{t+1} неизбыточна и допускает ЕПТ T_{t+1} длины не более $6t-4$, подмножеством которого является множество T^0_{t+1}. Заметим, что в случае $t=2$ достаточно доказать только неизбыточность схемы S_{t+1}, так как в качестве T_{t+1} можно взять тривиальный ЕПТ длины $2^{t+1} = 8 = 6t-4$ для этой схемы. Рассмотрим пять случаев:

1. Пусть $f'_t \equiv 0$. Тогда $f_t \not\equiv 0$ и схема S_{t+1} по построению совпадает с подсхемой S^1_{t+1}. По утверждению $(*)$ схема S^1_{t+1}, т. е. S_{t+1}, неизбыточна и допускает ЕПТ T^1_{t+1}, подмножеством которого является множество T^0_{t+1}, причём $|T^1_{t+1}| \leqslant 6t-7$ при $t \geqslant 3$. Тогда в случае $t \geqslant 3$ можно взять $T_{t+1} = T^1_{t+1}$.

2. Пусть $f'_t \equiv 1$ и $f_t \equiv 1$. Тогда по построению схема S_{t+1} состоит из одного инвертора I'_{t+1}, вход которого соединён со входом «x_{t+1}» схемы, а её выходом является выход элемента I'_{t+1}. Очевидно, что в случае, когда инвертор I'_{t+1} исправен, на его вы-

ходе реализуется функция $f_{t+1}(\tilde{x}_{t+1}) = \overline{x}_{t+1}$, а при произвольной неисправности этого инвертора — одна из функций 0, 1 и x_{t+1}, каждую из которых можно отличить от функции $\overline{x}_{t+1}$ хотя бы на одном из наборов $(\tilde{0}^{t+1}), (\tilde{0}^t, 1) \in T^0_{t+1}$. Отсюда следует, что схема S_{t+1} неизбыточна, а множество T^0_{t+1} является для неё ЕПТ длины $2t+2 < 6t-4$, и можно взять $T_{t+1} = T^0_{t+1}$.

3. Пусть $f'_t \equiv 1$ и $f_t \not\equiv 1$. Из первого соотношения вытекает, что $f_t \not\equiv 0$, значит, функция $f_t(\tilde{x}^t)$ отлична от констант. В этом случае по построению выход схемы S^1_{t+1} совпадает с выходом конъюнктора $E^{\&}_{t+1}$, а схема S_{t+1} получается из схемы S^1_{t+1} добавлением инвертора I'_{t+1}, вход которого соединяется с выходом подсхемы S^1_{t+1}, и переносом выхода схемы на выход данного инвертора. При переходе в произвольное неисправное состояние произвольного одного элемента из подсхемы S^1_{t+1} схемы S_{t+1} значение, выдаваемое этой подсхемой хотя бы на одном наборе из множества T^1_{t+1}, изменится, поскольку T^1_{t+1} — ЕПТ для неизбыточной схемы S_t в силу утверждения $(*)$. Тогда изменится значение, поступающее на вход инвертора I'_{t+1}, а следовательно, и значение, возникающее на его выходе, т. е. на выходе схемы S_{t+1}. Заметим также, что константная неисправность типа p (инверсная неисправность) на выходе инвертора I_{t+1} равносильна константной неисправности типа $\overline{p}$ (инверсной неисправности) на выходе конъюнктора $E^{\&}_{t+1}$, где p — произвольное число из множества $\{0, 1\}$. Тем самым показано, что любая неисправность любого одного элемента схемы S^1_{t+1} обнаруживается хотя бы на одном наборе из множества T^1_{t+1}. Отсюда следует, что схема S_{t+1} неизбыточна и допускает ЕПТ T^1_{t+1}, подмножеством которого является множество T^0_{t+1}, причём $|T^1_{t+1}| \leqslant 6t-7$ при $t \geqslant 3$ (см. утверждение $(*)$), и в случае $t \geqslant 3$ можно взять $T_{t+1} = T^1_{t+1}$.

4. Пусть $f_t \equiv 0$. Тогда по построению схема S_{t+1} совпадает с неизбыточной схемой S'_{t+1}, допускающей ЕПТ T'_{t+1} длины не более $2t+4$, подмножеством которого является множество T^0_{t+1}; в случае $t \geqslant 3$ с учётом неравенства $2t+4 < 6t-7$ можно взять $T_{t+1} = T'_{t+1}$.

5. Пусть $f_t \not\equiv 0$, $f'_t \not\equiv 0$ и $f'_t \not\equiv 1$. Тогда по построению схема S_{t+1} содержит подсхемы S^1_{t+1} и S'_{t+1}, выходы которых соединены с левым и правым входами сумматора $E^{\oplus}_{1,t+1}$ соответственно; в случае $a = 0$ выход схемы S_{t+1} совпадает с выходом элемента $E^{\oplus}_{1,t+1}$, а в случае $a = 1$ выход этого элемента и вход «x_{t+1}» схемы S_{t+1} соединены с левым и правым входами сумматора $E^{\oplus}_{2,t+1}$ соответственно, выход которого совпадает с выходом схемы. Схема S^1_{t+1} реализует при отсутствии в ней неисправностей функцию $x_{t+1}(f_t(\tilde{x}^t) \oplus a)$, по утверждению $(*)$ неизбыточна и допускает ЕПТ T^1_{t+1}, подмножеством которого является множество T^0_{t+1}, причём $|T^1_{t+1}| \leqslant 6t-7$ при $t \geqslant 3$. Выход схемы S^1_{t+1} совпадает с выходом конъюнктора $E^{\&}_{t+1}$ в случае $f_t \not\equiv 1$ и со входом «x_{t+1}» схемы S_{t+1} в случае $f_t \equiv 1$. В первом из этих случаев константная неисправность типа 0 на выходе элемента $E^{\&}_{t+1}$ должна обнаруживаться на каком-то наборе $\tilde{\rho}_{t,1} \in T^1_{t+1}$, откуда следует, что значение функции $x_{t+1}(f_t(\tilde{x}^t) \oplus a)$ на наборе $\tilde{\rho}_{t,1}$ равно единице, т. е. $(t+1)$-я компонента данного набора равна единице и

$$f_t(\tilde{\rho}_t) \oplus a = 1, \tag{8}$$

где $\tilde{\rho}_t$ — набор, получающийся из набора $\tilde{\rho}_{t,1}$ удалением $(t+1)$-й компоненты. В случае $f_t \equiv 1$ положим $\tilde{\rho}_t = (\tilde{0}^t)$ и $\tilde{\rho}_{t,1} = (\tilde{0}^t, 1)$; тогда равенство (8) также выполнено (с учётом того, что в данном случае $a = 0$), $(t+1)$-я компонента набора $\tilde{\rho}_{t,1}$ также равна единице и $\tilde{\rho}_{t,1} \in T^0_{t+1} \subseteq T^1_{t+1}$.

Схема S'_{t+1} содержит одну фиктивную входную переменную x_{t+1} и реализует функцию $f'_t(\tilde{x}^t)$, т. е. на выходе этой схемы реализуется булева функция $f'^{(+1)}_t(\tilde{x}^{t+1})$, не за-

висящая существенно от переменной x_{t+1} и равная $f'_t(\tilde{x}^t)$. Кроме того, по построению схема S'_{t+1} неизбыточна и допускает ЕПТ T'_{t+1} длины не более $2t+4$, подмножеством которого является множество $T^0_{t+1} \cup \tilde{\rho}'_{t,0}$, где $\tilde{\rho}'_{t,0}$ — некоторый такой двоичный $(t+1)$-разрядный набор, что его $(t+1)$-я компонента равна 0 и t-разрядный набор $\tilde{\rho}'_t$, получающийся из набора $\tilde{\rho}'_{t,0}$ удалением $(t+1)$-й компоненты, удовлетворяет условию

$$f'_t(\tilde{\rho}'_t) \neq f'_t(\tilde{0}^t).$$

Докажем существование такого двоичного t-разрядного набора $\tilde{\rho}''_t$, что

$$f_t(\tilde{\rho}''_t) \oplus a = 1; \tag{9}$$
$$f'_t(\tilde{\rho}''_t) \neq f'_t(\tilde{\rho}_t). \tag{10}$$

Пусть сначала $a = 0$. Из определения числа a и предположения случая 5 вытекает, что либо $f_t \equiv 1$, либо не выполнено ни одно из функциональных неравенств $f_t \leqslant f'_t$, $f_t \leqslant \overline{f'_t}$. В обоих случаях существуют такие двоичные t-разрядные наборы $\tilde{\lambda}_t$ и $\tilde{\lambda}'_t$, что

$$f_t(\tilde{\lambda}_t) = f_t(\tilde{\lambda}'_t) = f'_t(\tilde{\lambda}_t) = 1$$

и $f'_t(\tilde{\lambda}'_t) = 0$. Среди наборов $\tilde{\lambda}_t$ и $\tilde{\lambda}'_t$ можно выбрать такой набор $\tilde{\rho}''_t$, что выполнены соотношения (9) (с учётом равенства $a = 0$) и (10), что и требовалось доказать.

Пусть теперь $a = 1$. Из определения числа a следует, что обе функции f_t, f'_t неконстантные и либо $f_t \leqslant f'_t$, либо $f_t \leqslant \overline{f'_t}$. Тогда $f_t \leqslant f'_t \oplus c_t$ для некоторого $c_t \in \{0,1\}$. Из регулярности функции $f_{t+1}(\tilde{x}^{t+1})$ и отличия обеих функций f_t, f'_t от констант вытекает, что функция $f_t(\tilde{x}^t)$ отлична от $f'_t(\tilde{x}^t)$ и от $\overline{f'_t}(\tilde{x}^t)$, в частности, от функции $f'_t(\tilde{x}^t) \oplus c_t$. Значит, существует такой двоичный t-разрядный набор $\tilde{\lambda}_t$, что $f_t(\tilde{\lambda}_t) = 0$ и $f'_t(\tilde{\lambda}_t) \oplus c_t = 1$, т. е. $f'_t(\tilde{\lambda}_t) = \bar{c}_t$. Функция $f'_t(\tilde{x}^t)$ отлична от константы $\bar{c}_t$, поэтому существует такой двоичный t-разрядный набор $\tilde{\lambda}'_t$, что $f'_t(\tilde{\lambda}'_t) = c_t$. Тогда

$$f_t(\tilde{\lambda}'_t) \leqslant f'_t(\tilde{\lambda}'_t) \oplus c_t = c_t \oplus c_t = 0,$$

откуда $f_t(\tilde{\lambda}'_t) = 0$. Получаем, что $f_t(\tilde{\lambda}_t) = f_t(\tilde{\lambda}'_t) = 0$, $f'_t(\tilde{\lambda}_t) = \bar{c}_t$ и $f'_t(\tilde{\lambda}'_t) = c_t$. Среди наборов $\tilde{\lambda}_t$ и $\tilde{\lambda}'_t$ можно выбрать такой набор $\tilde{\rho}''_t$, что выполнены соотношения (9) (с учётом равенства $a = 1$) и (10), что и требовалось доказать.

Обозначим через $\tilde{\rho}''_{t,1}$ набор, получающийся из набора $\tilde{\rho}''_t$ добавлением $(t+1)$-й компоненты, равной единице, и положим $T_{t+1} = T^1_{t+1} \cup (T'_{t+1} \setminus T^0_{t+1}) \cup \{\tilde{\rho}''_{t,1}\}$. Тогда

$$T^0_{t+1} \subseteq T^1_{t+1} \subseteq T_{t+1},$$

а при $t \geqslant 3$ имеем

$$|T_{t+1}| \leqslant |T^1_{t+1}| + |T'_{t+1} \setminus T^0_{t+1}| + 1 = |T^1_{t+1}| + |T'_{t+1}| - |T^0_{t+1}| + 1 \leqslant 6t - 7 + 2t + 4 - (2t+2) + 1 = 6t - 4.$$

Поэтому достаточно доказать, что схема S_{t+1} неизбыточна, а множество T_{t+1} является для неё ЕПТ. Заметим, что $T^1_{t+1} \cup (T'_{t+1} \setminus T^0_{t+1}) = T^1_{t+1} \cup T'_{t+1}$, так как $T^0_{t+1} \subseteq T^1_{t+1}$. Значит, $T_{t+1} = T^1_{t+1} \cup T'_{t+1} \cup \{\tilde{\rho}''_{t,1}\}$. Также нам понадобятся соотношения

$$(\tilde{0}^{t+1}) \in T^0_{t+1} \subseteq T^1_{t+1} \subseteq T_{t+1},$$
$$\tilde{\rho}'_{t,0} \in T'_{t+1} \subseteq T_{t+1},$$
$$\tilde{\rho}_{t,1} \in T^1_{t+1} \subseteq T_{t+1},$$
$$\tilde{\rho}''_{t,1} \in T_{t+1}.$$

Несмотря на различие строения схемы S_{t+1} в случаях $a = 0$ и $a = 1$, ряд рассуждений можно провести для обоих этих случаев. При переходе в произвольное неисправное состояние произвольного одного элемента из подсхемы S^1_{t+1} (подсхемы S'_{t+1}) схемы S_{t+1} значение, выдаваемое этой подсхемой хотя бы на одном наборе из множества T^1_{t+1} (соответственно T'_{t+1}), изменится, поскольку данное множество является ЕПТ для указанной подсхемы. Тогда изменится значение, поступающее на левый (соответственно правый) вход сумматора $E^{\oplus}_{1,t+1}$, а значение на другом его входе останется неизменным, поэтому значение, возникающее на его выходе, изменится хотя бы на одном наборе из множества $T^1_{t+1} \cup T'_{t+1} \subseteq T_{t+1}$.

Рассмотрим теперь произвольную неисправность сумматора $E^{\oplus}_{1,t+1}$. Пусть данный элемент при ней реализует вместо «правильной» функции $\varphi_{E^{\oplus}_{1,t+1}}(x, y) = x \oplus y$ некоторую другую булеву функцию $\varphi'_{E^{\oplus}_{1,t+1}}(x, y)$ (от своих входов). Обозначим через $(\beta_{1,t+1}, \beta_{2,t+1})$ произвольный двоичный набор, на котором значения этих функций различаются. В схеме S_{t+1} на левый и правый входы сумматора $E^{\oplus}_{1,t+1}$ подаются функции $x_{t+1}(f_t(\tilde{x}^t) \oplus a)$ и $f_t'^{(+1)}(\tilde{x}^{t+1})$ с выходов подсхем S^1_{t+1} и S'_{t+1} соответственно. На наборах $(\tilde{0}^{t+1})$, $\tilde{\rho}'_{t,0}$, $\tilde{\rho}_{t,1}$ и $\tilde{\rho}''_{t,1}$, каждый из которых принадлежит T_{t+1}, пара функций $(x_{t+1}(f_t(\tilde{x}^t) \oplus a), f_t'^{(+1)}(\tilde{x}^{t+1}))$ принимает пары значений $(0, f'_t(\tilde{0}^t))$, $(0, f'_t(\tilde{\rho}'_t))$, $(f_t(\tilde{\rho}_t) \oplus a, f'_t(\tilde{\rho}_t))$ и $(f_t(\tilde{\rho}''_t) \oplus a, f'_t(\tilde{\rho}''_t))$ соответственно, т. е. пары значений $(0, f'_t(\tilde{0}^t))$, $(0, \overline{f'_t}(\tilde{0}^t))$, $(1, f'_t(\tilde{\rho}_t))$ и $(1, \overline{f'_t}(\tilde{\rho}_t))$ (см. (8)–(10)). Очевидно, что среди них присутствуют все пары $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$.

Таким образом, установлено, что при последовательной подаче на входы схемы S_{t+1} некоторых четырёх наборов из множества T_{t+1} на входы сумматора $E^{\oplus}_{1,t+1}$ поступают наборы $(0, 0)$, $(0, 1)$, $(1, 0)$ и $(1, 1)$. При подаче на входы схемы какого-то из четырёх выбранных наборов на входы сумматора поступит набор $(\beta_{1,t+1}, \beta_{2,t+1})$. При переходе элемента $E^{\oplus}_{1,t+1}$ в рассматриваемое неисправное состояние значение на его выходе изменится с $\varphi_{E^{\oplus}_{1,t+1}}(\beta_{1,t+1}, \beta_{2,t+1})$ на $\varphi'_{E^{\oplus}_{1,t+1}}(\beta_{1,t+1}, \beta_{2,t+1})$. Рассмотрим два подслучая:

5.1. Пусть $a = 0$. Тогда по построению выход схемы S_{t+1} совпадает с выходом сумматора $E^{\oplus}_{1,t+1}$. В силу рассуждений из предыдущих трёх абзацев любая неисправность любого одного элемента из подсхемы S^1_{t+1} или S'_{t+1}, а также любая неисправность элемента $E^{\oplus}_{1,t+1}$ обнаруживается в схеме S_{t+1} хотя бы на одном наборе из множества T_{t+1}. Это означает, что схема S_{t+1} неизбыточна, а множество T_{t+1} является для неё ЕПТ, что и требовалось доказать.

5.2. Пусть $a = 1$. В этом случае по построению выход элемента $E^{\oplus}_{1,t+1}$ и вход «x_{t+1}» схемы S_{t+1} соединены в ней с левым и правым входами сумматора $E^{\oplus}_{2,t+1}$ соответственно, выход которого совпадает с выходом схемы. Как показано выше, при переходе в произвольное неисправное состояние сумматора $E^{\oplus}_{1,t+1}$ либо произвольного одного элемента из подсхемы S^1_{t+1} или S'_{t+1} значение, возникающее в схеме S_{t+1} на выходе элемента $E^{\oplus}_{1,t+1}$ хотя бы на одном наборе из множества T_{t+1}, изменится. Это значение поступит на левый вход сумматора $E^{\oplus}_{2,t+1}$, а значение на его правом входе останется неизменным, поэтому значение на его выходе, т. е. на выходе схемы S_{t+1}, изменится. Тем самым показано, что любая неисправность любого одного элемента из подсхемы S^1_{t+1} или S'_{t+1}, а также любая неисправность элемента $E^{\oplus}_{1,t+1}$ обнаруживается в схеме S_{t+1} хотя бы на одном наборе из множества T_{t+1}.

Рассмотрим теперь произвольную неисправность сумматора $E^{\oplus}_{2,t+1}$. Пусть данный элемент при ней реализует вместо «правильной» функции $\varphi_{E^{\oplus}_{2,t+1}}(x, y) = x \oplus y$ некоторую другую булеву функцию $\varphi'_{E^{\oplus}_{2,t+1}}(x, y)$ (от своих входов). Обозначим через

$(\gamma_{1,t+1}, \gamma_{2,t+1})$ произвольный двоичный набор, на котором значения этих функций различаются. В схеме S_{t+1} на левый и правый входы сумматора $E^{\oplus}_{2,t+1}$ подаются функции $x_{t+1}(f_t(\tilde{x}^t)\oplus a)\oplus f_t'^{(+1)}(\tilde{x}^{t+1})$ и x_{t+1} с выхода сумматора $E^{\oplus}_{1,t+1}$ и входа «x_{t+1}» схемы соответственно. На наборах $(\tilde{0}^{t+1})$, $\tilde{\rho}'_{t,0}$, $\tilde{\rho}_{t,1}$ и $\tilde{\rho}''_{t,1}$, каждый из которых принадлежит T_{t+1}, пара функций $(x_{t+1}(f_t(\tilde{x}^t)\oplus a)\oplus f_t'^{(+1)}(\tilde{x}^{t+1}), x_{t+1})$ принимает пары значений $(0\oplus f'_t(\tilde{0}^t), 0)$, $(0\oplus f'_t(\tilde{\rho}'_t), 0)$, $((f_t(\tilde{\rho}_t)\oplus a)\oplus f'_t(\tilde{\rho}_t), 1)$ и $((f_t(\tilde{\rho}''_t)\oplus a)\oplus f'_t(\tilde{\rho}''_t), 1)$ соответственно, т. е. пары значений $(f'_t(\tilde{0}^t), 0)$, $(\overline{f'_t}(\tilde{0}^t), 0)$, $(1\oplus f'_t(\tilde{\rho}_t), 1)$ и $(1\oplus \overline{f'_t}(\tilde{\rho}_t), 1)$ (см. (8)–(10)). Очевидно, что среди них присутствуют все пары $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$.

Таким образом, установлено, что при последовательной подаче на входы схемы S_{t+1} некоторых четырёх наборов из множества T_{t+1} на входы сумматора $E^{\oplus}_{2,t+1}$ поступают наборы $(0,0)$, $(0,1)$, $(1,0)$ и $(1,1)$. При подаче на входы схемы какого-то из четырёх выбранных наборов на входы сумматора поступит набор $(\gamma_{1,t+1}, \gamma_{2,t+1})$. При переходе элемента $E^{\oplus}_{2,t+1}$ в рассматриваемое неисправное состояние значение на его выходе, т. е. на выходе схемы S_{t+1}, изменится с $\varphi_{E^{\oplus}_{1,t+1}}(\gamma_{1,t+1}, \gamma_{2,t+1})$ на $\varphi'_{E^{\oplus}_{1,t+1}}(\gamma_{1,t+1}, \gamma_{2,t+1})$ и неисправность будет обнаружена на каком-то наборе из множества T_{t+1}.

В итоге получаем, что любая неисправность любого одного элемента схемы S_{t+1} обнаруживается хотя бы на одном наборе из данного множества. Это означает, что схема S_{t+1} неизбыточна, а множество T_{t+1} является для неё ЕПТ, что и требовалось доказать. Случай 5 разобран.

Случаями 1–5 исчерпываются все возможные случаи. Основное утверждение, а вместе с ним шаг индукции и лемма 1 доказаны. ■

Доказательство леммы 2. Докажем индукцией по n, что для любой неконстантной булевой функции $f_n(\tilde{x}^n)$ существуют такие $\sigma_{1,n}, \ldots, \sigma_{n,n} \in \{0,1\}$, что функция $f_n(x_1^{\sigma_{1,n}}, \ldots, x_n^{\sigma_{n,n}})$ регулярна и отлична от констант; отсюда следует утверждение леммы.

Б а з а и н д у к ц и и: $n = 1$ или 2. По определению любая булева функция от одной или двух переменных является регулярной, поэтому в случаях $n = 1$, 2 можно взять соответственно $\sigma_{1,1} = 1$, $\sigma_{1,2} = \sigma_{2,2} = 1$.

П р е д п о л о ж е н и е и ш а г и н д у к ц и и: пусть утверждение доказано для $n = t$, где $t \geqslant 2$; докажем его для $n = t + 1$. Функцию f_{t+1} можно единственным образом представить в виде (3), где f_t и f'_t — булевы функции. Рассмотрим два случая:

1. Пусть $f_t \equiv c$ для некоторого $c \in \{0,1\}$. Если $t \geqslant 3$, то функцию f_t можно единственным образом представить в виде

$$f_t(\tilde{x}^t) = x_t f_{t-1}(\tilde{x}^{t-1}) \oplus f'_{t-1}(\tilde{x}^{t-1}),$$

где f_{t-1} и f'_{t-1} — булевы функции, причём этот вид достигается при $f_{t-1} \equiv 0$ и $f'_{t-1} \equiv c$. В случае $t \geqslant 4$ функцию $f_{t-1} \equiv 0$ можно единственным образом представить в виде

$$f_{t-1}(\tilde{x}^{t-1}) = x_{t-1} f_{t-2}(\tilde{x}^{t-2}) \oplus f'_{t-2}(\tilde{x}^{t-2}),$$

где f_{t-2} и f'_{t-2} — булевы функции, причём этот вид достигается при $f_{t-2} \equiv 0$ и $f'_{t-2} \equiv 0$, и т. д. В итоге получаем, что для любого $i \in \{2, \ldots, t\}$ функция $f_i(\tilde{x}^i)$ является константной, а тогда функция $f_{t+1}(\tilde{x}^{t+1})$ по определению регулярна и можно взять $\sigma_{1,t+1} = \ldots = \sigma_{t+1,t+1} = 1$.

2. Пусть $f_t \not\equiv 0$ и $f_t \not\equiv 1$. Достаточно доказать существование таких $\sigma_{1,t+1}, \ldots, \sigma_{t+1,t+1} \in \{0,1\}$, что функция $f_{t+1}(x_1^{\sigma_{1,t+1}}, \ldots, x_{t+1}^{\sigma_{t+1,t+1}})$ регулярна; отличие её от кон-

стант будет следовать из отличия функции $f_{t+1}(\tilde{x}^{t+1})$ от констант. По предположению индукции существуют такие $\sigma_{1,t},\dots,\sigma_{t,t} \in \{0,1\}$, что функция $h_t(\tilde{x}^t) = f_t(x_1^{\sigma_{1,t}},\dots,x_t^{\sigma_{t,t}})$ регулярна и отлична от констант. В силу (3) имеем

$$f_{t+1}(x_1^{\sigma_{1,t}},\dots,x_t^{\sigma_{t,t}},x_{t+1}) = x_{t+1}h_t(\tilde{x}^t) \oplus h'_t(\tilde{x}^t), \tag{11}$$

где $h'_t(\tilde{x}^t) = f'_t(x_1^{\sigma_{1,t}},\dots,x_t^{\sigma_{t,t}})$. Если функция $h'_t(\tilde{x}^t)$ является константной или функция $h_t(\tilde{x}^t)$ отлична от $h'_t(\tilde{x}^t)$ и от $\overline{h'_t}(\tilde{x}^t)$, то из регулярности функции $h_t(\tilde{x}^t)$ следует регулярность функции $f_{t+1}(x_1^{\sigma_{1,t}},\dots,x_t^{\sigma_{t,t}},x_{t+1})$ и можно взять $\sigma_{1,t+1}=\sigma_{1,t}$, …, $\sigma_{t,t+1}=\sigma_{t,t}$, $\sigma_{t+1,t+1}=1$. В противном случае $h'_t(\tilde{x}^t) = h_t(\tilde{x}^t) \oplus c_t$ для некоторого $c_t \in \{0,1\}$. Преобразуем равенство (11):

$$f_{t+1}(x_1^{\sigma_{1,t}},\dots,x_t^{\sigma_{t,t}},x_{t+1}) = x_{t+1}h_t(\tilde{x}^t) \oplus h_t(\tilde{x}^t) \oplus c_t = (x_{t+1}\oplus 1)h_t(\tilde{x}^t) \oplus c_t = \overline{x}_{t+1}h_t(\tilde{x}^t) \oplus c_t,$$
$$f_{t+1}(x_1^{\sigma_{1,t}},\dots,x_t^{\sigma_{t,t}},\overline{x}_{t+1}) = x_{t+1}h_t(\tilde{x}^t) \oplus c_t.$$

С учётом того, что последнее слагаемое в правой части последнего равенства — константа, из регулярности функции $h_t(\tilde{x}^t)$ вытекает регулярность функции $f_{t+1}(x_1^{\sigma_{1,t}},\dots,x_t^{\sigma_{t,t}},\overline{x}_{t+1})$ и можно взять $\sigma_{1,t+1}=\sigma_{1,t}$, …, $\sigma_{t,t+1}=\sigma_{t,t}$, $\sigma_{t+1,t+1}=0$. Шаг индукции, а вместе с ним лемма 2 доказаны. ■

Доказательство леммы 3. Пусть S' — неизбыточная СФЭ в базисе B, реализующая функцию $f(x_1^{\sigma_1},\dots,x_n^{\sigma_n})$ и допускающая при $n \geqslant 3$ ЕПТ T' длины не более $6n-10$. Если выход схемы S' совпадает с одним из её входов, то на этом выходе реализуется функция $f(x_1^{\sigma_1},\dots,x_n^{\sigma_n}) = x_i$ для некоторого $i \in \{1,\dots,n\}$. Тогда $f(\tilde{x}^n) = x_i^{\sigma_i}$, т. е. $f(\tilde{x}^n) = x_i$ или $f(\tilde{x}^n) = \overline{x}_i$. В первом (во втором) из этих случаев функцию $f(\tilde{x}^n)$ можно реализовать СФЭ, не содержащей функциональных элементов (соответственно содержащей один инвертор); данная схема, очевидно, является неизбыточной и допускает ЕПТ $\{(\tilde{0}^n),(\tilde{1}^n)\}$, длина которого равна $2 < 6n-10$ при $n \geqslant 3$.

Пусть теперь в схеме S' содержится выходной элемент. Для удобства будем считать, что в случаях $n=1,2$ множество T' представляет собой тривиальный ЕПТ для схемы S', т. е. состоит из всех 2^n двоичных n-разрядных наборов. Для каждого такого $i \in \{1,\dots,n\}$, что $\sigma_i = 0$, и каждого входа каждого элемента схемы, соединённого со входом «x_i» схемы, отсоединим указанный вход элемента от входа «x_i» схемы и соединим его с выходом своего инвертора, на вход которого подадим переменную x_i со входа схемы. Все указанные преобразования произведём одновременно. Полученную в результате схему обозначим через S; легко видеть, что она является СФЭ в базисе B и реализует булеву функцию, получающуюся из функции $f(x_1^{\sigma_1},\dots,x_n^{\sigma_n})$ подстановкой для каждого такого $i \in \{1,\dots,n\}$, что $\sigma_i = 0$, вместо переменной x_i её отрицания, т. е. функцию $f(\tilde{x}^n)$. Обозначим через T множество, получающееся заменой для каждого такого $i \in \{1,\dots,n\}$, что $\sigma_i = 0$, i-й компоненты каждого набора из множества T' на противоположную. Очевидно, что $|T| = |T'| \leqslant 6n-10$ при $n \geqslant 3$, поэтому достаточно доказать, что схема S неизбыточна, а множество T является для неё ЕПТ.

Пусть M — множество всех инверторов схемы S, добавленных в ходе преобразований в неё схемы S'. Рассмотрим произвольную неисправность произвольного элемента E схемы S, не принадлежащего M. Такая же неисправность этого же элемента в схеме S' обязана обнаруживаться хотя бы на одном наборе $\tilde{\pi}'$ из множества T', поскольку T' — ЕПТ для неизбыточной схемы S'. Обозначим через $\tilde{\pi}$ набор, получающийся из $\tilde{\pi}'$ заменой для каждого такого $i \in \{1,\dots,n\}$, что $\sigma_i = 0$, i-й компоненты на противоположную. Тогда $\tilde{\pi} \in T$. Легко заметить, что при подаче на входы схемы S набора $\tilde{\pi}$ на

входах и выходе каждого элемента $\hat{E}$, не принадлежащего множеству M, возникают те же значения, что на входах и выходе того же элемента в схеме S' при подаче на её входы набора $\tilde{\pi}'$ как в случае отсутствия неисправностей в схемах S и S', так и в случае одинаковой неисправности элемента E в каждой из данных схем и исправной работы всех остальных элементов (формально это можно доказать, двигаясь по каждой из схем S, S' «сверху вниз» от входов к выходу). Взяв в качестве $\hat{E}$ выходной элемент каждой из схем S, S', получим, что рассматриваемая неисправность элемента E в схеме S обнаруживается на наборе $\tilde{\pi}$. Тем самым показано, что любая неисправность любого одного элемента схемы S, не принадлежащего M, обнаруживается хотя бы на одном наборе из множества T.

Осталось заметить, что в схеме S любая константная либо инверсная неисправность на выходе произвольного инвертора из множества M равносильна такой же неисправности на входе следующего за ним элемента, уже не принадлежащего M, соединённом с выходом этого инвертора. Приведённые рассуждения показывают, что схема S неизбыточна, а множество T является для неё ЕПТ. Лемма 3 доказана. ■

Заключение

В работе предложен метод реализации любой неконстантной булевой функции $f(\tilde{x}^n)$ схемой из функциональных элементов в базисе $\{\&, \oplus, \neg\}$, неизбыточной и допускающей при $n \geqslant 3$ единичный проверяющий тест длины не более $6n - 10$ относительно произвольных неисправностей элементов, существенно более короткий, чем тривиальный тест из 2^n наборов. Данный метод может быть использован на практике для построения легкотестируемых интегральных схем в случае, когда тип допустимых неисправностей содержащихся в них элементов никак не ограничивается (в частности, не ограничивается хорошо изученными константными либо инверсными неисправностями на входах и/или выходах элементов).

ЛИТЕРАТУРА

1. *Чегис И. А., Яблонский С. В.* Логические способы контроля работы электрических схем // Труды МИАН. 1958. Т. 51. С. 270–360.
2. *Яблонский С. В.* Надёжность и контроль управляющих систем // Материалы Всесоюз. семинара по дискретной математике и её приложениям (Москва, 31 января – 2 февраля 1984 г.). М.: Изд-во МГУ, 1986. С. 7–12.
3. *Яблонский С. В.* Некоторые вопросы надёжности и контроля управляющих систем // Математические вопросы кибернетики. Вып. 1. М.: Наука, 1988. С. 5–25.
4. *Редькин Н. П.* Надёжность и диагностика схем. М.: Изд-во МГУ, 1992. 192 с.
5. *Бородина Ю. В.* О схемах, допускающих единичные тесты длины 1 при константных неисправностях на выходах элементов // Вестник Московского университета. Сер. 1. Математика. Механика. 2008. № 5. С. 49–52.
6. *Попков К. А.* Нижние оценки длин единичных тестов для схем из функциональных элементов // Дискретная математика. 2017. Т. 29. Вып. 2. С. 53–69.
7. *Попков К. А.* Единичные проверяющие тесты для схем из функциональных элементов в базисе «конъюнкция-отрицание» // Прикладная дискретная математика. 2017. № 38. С. 66–88.
8. *Reddy S. M.* Easily testable realizations for logic functions // IEEE Trans. Comput. 1972. V. C-21. Iss. 11. P. 1183–1188.
9. *Коляда С. С.* Единичные проверяющие тесты для схем из функциональных элементов // Вестник Московского университета. Сер. 1. Математика. Механика. 2013. № 4. С. 32–34.

10. *Коляда С. С.* Верхние оценки длины проверяющих тестов для схем из функциональных элементов: дис. ... канд. физ.-мат. наук. М., 2013. 77 с.
11. *Романов Д. С.* Метод синтеза легкотестируемых схем, допускающих единичные проверяющие тесты константной длины // Дискретная математика. 2014. Т. 26. Вып. 2. С. 100–130.
12. *Попков К. А.* Короткие единичные тесты для схем при произвольных константных неисправностях на выходах элементов // Дискретная математика. 2018. Т. 30. Вып. 3. С. 99–116.
13. *Коваценко С. В.* Синтез легкотестируемых схем в базисе Жегалкина для инверсных неисправностей // Вестник Московского университета. Сер. 15. Вычислительная математика и кибернетика. 2000. № 2. С. 45–47.
14. *Редькин Н. П.* О единичных проверяющих тестах схем при инверсных неисправностях элементов // XII Междунар. конф. по проблемам теоретической кибернетики (Н. Новгород, 1999). Тез. докл. М.: Изд-во механико-математического факультета МГУ, 1999. С. 196.
15. *Редькин Н. П.* Единичные проверяющие тесты для схем при инверсных неисправностях элементов // Математические вопросы кибернетики. Вып. 12. М.: Физматлит, 2003. С. 217–230.
16. *Попков К. А.* Синтез легкотестируемых схем при однотипных константных неисправностях на входах и выходах элементов // Интеллектуальные системы. Теория и приложения. 2018. Т. 23. Вып. 3. С. 131–147.
17. *Романов Д. С., Романова Е. Ю.* Короткие тесты для схем в базисе Жегалкина // Интеллектуальные системы. Теория и приложения. 2016. Т. 20. Вып. 3. С. 73–78.
18. *Романов Д. С., Романова Е. Ю.* Метод синтеза неизбыточных схем, допускающих единичные проверяющие тесты константной длины // Дискретная математика. 2017. Т. 29. Вып. 4. С. 87–105.
19. *Попков К. А.* Синтез легкотестируемых схем при произвольных константных неисправностях на входах и выходах элементов // Прикладная дискретная математика. 2019. № 43. С. 78–100.
20. *Попков К. А.* Метод построения легко диагностируемых схем из функциональных элементов относительно единичных неисправностей // Прикладная дискретная математика. 2019. № 46. С. 38–57.
21. *Попков К. А.* О схемах, допускающих короткие единичные проверяющие тесты при произвольных неисправностях функциональных элементов // Прикладная дискретная математика. 2021. № 51. С. 85–100.
22. *Попков К. А.* Короткие полные проверяющие тесты для схем из двухвходовых функциональных элементов // Дискретный анализ и исследование операций. 2019. Т. 26. № 1. С. 89–113.

REFERENCES

1. *Chegis I. A. and Yablonskiy S. V.* Logicheskie sposoby kontrolya raboty elektricheskikh skhem [Logical methods of control of work of electric circuits]. Trudy Mat. Inst. Steklov, 1958, vol. 51, pp. 270–360. (in Russian)
2. *Yablonskiy S. V.* Nadezhnost' i kontrol' upravlyayushchikh sistem [Reliability and verification of control systems]. Materialy Vsesoyuznogo seminara po diskretnoy matematike i ee prilozheniyam (Moscow, 31 Jan.–2 Feb. 1984). Moscow, MSU Publ., 1986, pp. 7–12. (in Russian)

3. *Yablonskiy S. V.* Nekotorye voprosy nadezhnosti i kontrolya upravlyayushchikh sistem [Some questions of reliability and verification of control systems]. Matematicheskie Voprosy Kibernetiki, iss. 1. Moscow, Nauka Publ., 1988, pp. 5–25. (in Russian)
4. *Red'kin N. P.* Nadezhnost' i diagnostika skhem [Circuits Reliability and Diagnostics]. Moscow, MSU Publ., 1992. 192 p. (in Russian)
5. *Borodina Yu. V.* Circuits admitting single-fault tests of length 1 under constant faults at outputs of elements. Mosc. Univ. Math. Bull., 2008, vol. 63, iss. 5, pp. 202–204.
6. *Popkov K. A.* Lower bounds for lengths of single tests for Boolean circuits. Discrete Math. Appl., 2019, vol. 29, iss. 1, pp. 23–33.
7. *Popkov K. A.* Edinichnye proveryayushchie testy dlya skhem iz funktsional'nykh elementov v bazise "kon'yunktsiya-otritsanie" [Single fault detection tests for logic networks in the basis "conjunction-negation"]. Prikladnaya Diskretnaya Matematika, 2017, no. 38, pp. 66–88. (in Russian)
8. *Reddy S. M.* Easily testable realizations for logic functions. IEEE Trans. Comput., 1972, vol. C-21, iss. 11, pp. 1183–1188.
9. *Kolyada S. S.* Single fault detection tests for circuits of functional elements. Mosc. Univ. Math. Bull., 2013, vol. 68, iss. 4, pp. 192–193.
10. *Kolyada S. S.* Verkhnie otsenki dliny proveryayushchikh testov dlya skhem iz funktsional'nykh elementov [Upper bounds on length of fault detection tests for logic networks]. Cand. Diss. Moscow, 2013. 77 p. (in Russian)
11. *Romanov D. S.* Method of synthesis of easily testable circuits admitting single fault detection tests of constant length. Discrete Math. Appl., 2014, vol. 24, iss. 4, pp. 227–251.
12. *Popkov K. A.* Short single tests for circuits with arbitrary stuck-at faults at outputs of gates. Discrete Math. Appl., 2019, vol. 29, iss. 5, pp. 321–333.
13. *Kovatsenko S. V.* Sintez legkotestiruemykh skhem v bazise Zhegalkina dlya inversnykh neispravnostey [Synthesis of easily testable logic networks in the Zhegalkin basis for inverse faults]. Vestnik MSU, Ser. 15, 2000, no. 2, pp. 45–47. (in Russian)
14. *Red'kin N. P.* O edinichnykh proveryayushchikh testakh skhem pri inversnykh neispravnostyakh elementov [On single fault detection tests of logic networks under inverse faults of gates]. XII Mezhd. konf. po problemam teoreticheskoy kibernetiki (Nizhny Novgorod, 1999). Tezisy dokladov. Moscow, Mech.-Math. Faculty of MSU Publ., 1999, p. 196. (in Russian)
15. *Red'kin N. P.* Edinichnye proveryayushchie testy dlya skhem pri inversnykh neispravnostyakh elementov [Single fault detection tests for logic networks under inverse faults of gates]. Matematicheskie Voprosy Kibernetiki, iss. 12. Moscow, Fizmatlit Publ., 2003, pp. 217–230. (in Russian)
16. *Popkov K. A.* Sintez legkotestiruemykh skhem pri odnotipnykh konstantnykh neispravnostyakh na vkhodakh i vykhodakh elementov [Synthesis of easily testable logic networks under one-type stuck-at faults at inputs and outputs of gates]. Intellektual'nye Sistemy. Teoriya i Prilozheniya, 2018, vol. 23, iss. 3, pp. 131–147. (in Russian)
17. *Romanov D. S. and Romanova E. Yu.* Korotkie testy dlya skhem v bazise Zhegalkina [Short tests for circuits in the Zhegalkin basis]. Intellektual'nye Sistemy. Teoriya i Prilozheniya, 2016, vol. 20, iss. 3, pp. 73–78. (in Russian)
18. *Romanov D. S. and Romanova E. Yu.* A method of synthesis of irredundant circuits admitting single fault detection tests of constant length. Discrete Math. Appl., 2019, vol. 29, iss. 1, pp. 35–48.
19. *Popkov K. A.* Sintez legkotestiruemykh skhem pri proizvol'nykh konstantnykh neispravnostyakh na vkhodakh i vykhodakh elementov [Synthesis of easily testable

logic networks under arbitrary stuck-at faults at inputs and outputs of gates]. Prikladnaya Diskretnaya Matematika, 2019, no. 43, pp. 78–100. (in Russian)

20. *Popkov K. A.* Metod postroeniya legko diagnostiruemykh skhem iz funktsional'nykh elementov otnositel'no edinichnykh neispravnostey [A method of constructing of easily diagnosable logic networks regarding single faults]. Prikladnaya Diskretnaya Matematika, 2019, no. 46, pp. 38–57. (in Russian)
21. *Popkov K. A.* O skhemakh, dopuskayushchikh korotkie edinichnye proveryayushchie testy pri proizvol'nykh neispravnostyakh funktsional'nykh elementov [On logic networks allowing short single fault detection tests under arbitrary faults of gates]. Prikladnaya Diskretnaya Matematika, 2021, no. 51, pp. 85–100. (in Russian)
22. *Popkov K. A.* Short complete fault detection tests for logic networks with fan-in two. J. Appl. Industr. Math., 2019, vol. 13, iss. 1, pp. 118–131.

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

УДК 519.17 DOI 10.17223/20710410/55/5

О $\langle 2\rangle$-ЭКСПОНЕНТАХ ОРГРАФОВ НЕЛИНЕЙНОСТИ РЕГИСТРОВЫХ ПРЕОБРАЗОВАНИЙ

В. М. Фомичёв[*,**], В. М. Бобров[***]

[*] *Финансовый университет при Правительстве Российской Федерации, г. Москва, Россия*
[**] *Федеральный исследовательский центр «Информатика и управление» Российской академии наук (ФИЦ ИУ РАН), г. Москва, Россия*
[***] *Национальный исследовательский ядерный университет «МИФИ», г. Москва, Россия*

E-mail: fomichev.2016@yandex.ru, bvm_15@mail.ru

Матрично-графовый подход применяется для оценки множества существенных и нелинейных переменных координатных функций произведения преобразований векторных пространств. Для существенных переменных оценки получаются с помощью умножения двоичных перемешивающих матриц (или орграфов) умножаемых преобразований, для нелинейных переменных — с помощью умножения троичных матриц нелинейности умножаемых преобразований или соответствующих им орграфов нелинейности, дуги которых помечены числами множества $\{0, 1, 2\}$. Для степеней заданного преобразования область нетривиальных оценок ограничена: для множества существенных переменных — экспонентом перемешивающей матрицы (орграфа); для множества нелинейных переменных — $\langle 2\rangle$-экспонентом матрицы (орграфа) нелинейности. Для класса преобразований двоичных регистров сдвига получена достижимая оценка $\langle 2\rangle$-экспонентов, выраженная через длину регистра сдвига и множества номеров существенных и нелинейных переменных функции обратной связи. Для регистровых преобразований, орграф нелинейности которых имеет петлю, получена точная формула $\langle 2\rangle$-экспонента. Результаты могут быть использованы для оценки характеристик нелинейности криптографических функций, построенных на основе итераций регистровых преобразований.

Ключевые слова: *преобразование регистра сдвига, орграф нелинейности, $\langle 2\rangle$-примитивность, локальная $\langle 2\rangle$-примитивность, $\langle 2\rangle$-экспонент орграфа, локальный $\langle 2\rangle$-экспонент орграфа.*

$\langle 2\rangle$-EXPONENTS OF SHIFT REGISTER TRANSFORMATIONS NONLINEARITY DIPGRAPHS

V. M. Fomichev[*,**], V. M. Bobrov[***]

[*] *Financial University under the Government of the Russian Federation, Moscow, Russia*
[**] *Federal Research Center "Computer Science and Control", RAS, Moscow, Russia*
[***] *National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, Russia*

The matrix-graph approach is used to estimate sets of essential and non-linear variables of coordinate functions of the vector space transformations product. Estimates are obtained by multiplying binary mixing matrices or digraphs of multiplied transformations in the case of the set of essential variables, or by multiplying ternary nonlinearity matrices or corresponding nonlinearity digraphs with the arcs marked by numbers from the set $\{0, 1, 2\}$. For powers of a given transformation the non-trivial estimates domain is limited by the mixing matrix (digraph) exponent in the case of the essential variables and by the nonlinearity matrix (digraph) $\langle 2\rangle$-exponent in the case of the nonlinear variables. Let $f(x_0, \ldots, x_{n-1})$ be the feedback function of shift register, $D = \{d_0, \ldots, d_m\}$ be the set of its essential variables (registers extraction points), $1 < m \leqslant n$, $0 = d_0 < d_1 < \ldots < d_m < n$, $E = \{e_0, \ldots, e_l\}$ be the set of its nonlinear variables (registers nonlinear extraction points), $1 < l \leqslant n$, $e_0 < e_1 < \ldots < e_m < n$, and shift register transformation nonlinearity digraph G be $\langle 2\rangle$-primitive. Then $\langle 2\rangle$-exponent of G is not greater than $F(L)+1+n+\Delta_f$, where $F(L)$ is a Frobenius number of the set L of the lengths of all digraph simple circuits, $\Delta_f = \max\limits_{i\in D^0\cup D^1} \mu(i-1)$,

$$\mu(u) = \begin{cases} \min\{u - e(u), u - d(u) + n - e_l\}, & e_0 \leqslant u < n; \\ u - d(u) + n - e_l, & 0 \leqslant u < e_0, \end{cases}$$

$$D^0 = \{d_s \in D, 1 \leqslant s \leqslant m : d_{s-1} - e(d_{s-1}) \leqslant \lambda, e_0 \leqslant d_{s-1} < n\} \cup S_0(n),$$

$$D^1 = \{d_s \in D, 1 \leqslant s \leqslant m : 0 \leqslant d_{s-1} < e_0 \text{ or } d_{s-1} - e(d_{s-1}) > \lambda\} \cup S_1(n),$$

where $d(u)$ is the greatest number of D such that $d(u) \leqslant u$, $0 \leqslant u < n$; $e(u)$ is the greatest number of E such that $e(u) \leqslant u$, $e_0 \leqslant u < n$; $S_0(n) = S_1(n) = \varnothing$, if $d_m = n - 1$; $S_0(n) = \{n\}$, if $d_m < n - 1$ and $d_m \in E$; and $S_1(n) = \{n\}$, if $d_m < n-1$ and $d_m \in E$. In the case of the variable x_{n-1} being essential for the function $f(x_0, \ldots, x_{n-1})$, the exact formula for the $\langle 2\rangle$-exponent of the nonlinearity digraph has been derived. Calculation examples are presented. The results can be used to estimate the nonlinearity characteristics of cryptographic functions constructed from iterated register transformations.

Keywords: *shift register transformation, nonlinearity digraph, $\langle 2\rangle$-primitivity, local $\langle 2\rangle$-primitivity, $\langle 2\rangle$-exponent of digraph, local $\langle 2\rangle$-exponent of digraph.*

Введение

Матрично-графовый подход (МГП) [1] позволяет оценить множества существенных переменных координатных функций произведения преобразований векторных пространств. Основой МГП является исследование двоичной перемешивающей матрицы $M = (m_{i,j})$ преобразования g (или его перемешивающего орграфа, что равносильно в силу биекции между множеством орграфов и множеством их матриц смежности), где $m_{i,j} = 1$, если j-я координатная функция преобразования g зависит от i-й переменной существенно, и $m_{i,j} = 0$ в противном случае. Известно, что перемешивающая матрица произведения преобразований ограничена сверху (поэлементно) произведением перемешивающих матриц умножаемых преобразований. Это позволяет оценивать характеристики произведения преобразований с помощью произведения перемешивающих матриц (орграфов) сомножителей. Область нетривиальности таких оценок для степеней преобразования g ограничена экспонентом его перемешивающей матрицы (орграфа). Получению оценок экспонентов неотрицательных матриц посвящено много российских и зарубежных работ, результаты достаточно полно отражены в обзоре [1].

Исследование экспонентов неотрицательных матриц началось с поставленной Фробениусом [2] задачи по распознаванию положительной матрицы среди элементов цик-

лической полугруппы $\langle M\rangle$, порождённой квадратной матрицей M с неотрицательными элементами. При наличии положительной матрицы порождающая матрица M называется примитивной, а наименьшая степень t, при которой M^t положительная, называется экспонентом матрицы M [3]. Критерий примитивности получен в [4]: сильносвязный орграф примитивен, если длины его контуров взаимно просты. Много работ посвящено получению как универсальных, так и частных оценок экспонентов неотрицательных матриц и орграфов.

В [5, 6] представлено расширение МГП, позволяющее оценивать характеристики нелинейности произведения преобразований. Исследуется троичная матрица нелинейности M_θ, в которой зависимость j-й координатной функции от x_i кодируется двумя значениями: $m_{ij} = 2$, если указанная зависимость нелинейная, и $m_{ij} = 1$, если линейная. Таким образом, по сравнению с перемешивающей матрицей матрица M_θ более глубоко оценивает свойства преобразований.

Множеству троичных матриц биективно соответствует множество помеченных орграфов, для которых эти матрицы являются матрицами смежности вершин, где дуга (i, j) орграфа помечена элементом $m_{i,j}$ матрицы. Графовая модель вместо произведения троичных матриц позволяет изучать пути в помеченных орграфах, что нередко технически более удобно.

Орграф нелинейности преобразования двоичного векторного пространства размерности n имеет множество вершин $\{0, \ldots, n-1\}$, $n > 1$, и множество дуг, кодирующих характер зависимости каждой координатной функции преобразования от каждой переменной [5, 6]. Дуга (i, j) орграфа помечена символом «1» или «2», если j-я координатная функция зависит от x_i соответственно линейно или нелинейно; если j-я координатная функция несущественно зависит от x_i, то в орграфе дуги (i, j) нет. В произведении помеченных орграфов Γ_1 и Γ_2 дуга (i, j) помечена символом $\max\{a, b\}$, если в Γ_1 и Γ_2 имеются дуги (i, k) и (k, j) соответственно, $0 \leqslant k < n$, одна из которых помечена символом a и другая — символом b, $a, b \in \{1, 2\}$. Помеченный орграф называется $\langle 2\rangle$-примитивным, если его некоторая степень есть полный орграф с петлями и каждая дуга имеет метку «2». Указанная степень называется $\langle 2\rangle$-экспонентом орграфа.

В криптографических системах сложное преобразование часто построено с помощью итерации более простого, но удобно реализуемого преобразования. В частности, в симметричных блочных шифрах количество раундов, требуемое для обеспечения перемешивающих и нелинейных свойств, оценивается снизу $\langle 2\rangle$-экспонентом орграфа нелинейности раундовой подстановки [6].

В работе изучена зависимость $\langle 2\rangle$-экспонента орграфа нелинейности регистрового преобразования векторного пространства от длины регистра сдвига и множеств номеров существенных и нелинейных переменных функции обратной связи. Эта задача решена с помощью развития метода получения точной формулы экспонента перемешивающих орграфов регистровых преобразований [7]. Начальные результаты в этом направлении представлены в [8].

Орграф нелинейности преобразования двоичного регистра левого сдвига длины n с нелинейной обратной связью (ячейки регистра нумеруются слева направо числами от 0 до $n-1$) представляет собой объединение нескольких контуров с общей вершиной $n-1$. Для класса помеченных $\langle 2\rangle$-примитивных орграфов нелинейности регистровых преобразований получены оценки $\langle 2\rangle$-экспонентов и локальных $\langle 2\rangle$-экспонентов, выраженные через длину регистра сдвига и множества номеров существенных и нелинейных переменных функции обратной связи. Для регистровых преобразований, орграф нелинейности которых имеет петлю, получены точные формулы $\langle 2\rangle$-экспонентов и ло-

кальных $\langle 2\rangle$-экспонентов. Приведены примеры вычислений. Случаи регистров правого и левого сдвигов рассматриваются двойственно.

1. Основные определения

Исследуем двоичный регистр левого сдвига длины $n > 2$ с нелинейной булевой функцией обратной связи $f(x_0, \ldots, x_{n-1})$. Ячейки регистра занумеруем слева направо числами $0, \ldots, n-1$.

Обозначим:

- V_n — множество двоичных векторов длины n (состояний регистра);
- $F(A)$ — число Фробениуса для множества натуральных аргументов A, где НОД$(A) = 1$, определяемое как наибольшее целое число, не содержащееся в аддитивной полугруппе, порождённой множеством A;
- $Y_n = \{0, \ldots, n-1\}$ — множество номеров ячеек регистра сдвига;
- ϕ_f — преобразование множества V_n, реализуемое нелинейным регистром левого сдвига с обратной связью f;
- $D = \{d_0, \ldots, d_m\} \subseteq Y_n$, где $0 < m \leqslant n-1$, — непустое множество номеров существенных переменных функции $f(x_0, \ldots, x_{n-1})$ (точек съёма регистра); далее считаем $0 = d_0 < d_1 < \ldots < d_m < n$ (условие $d_0 = 0$ не ограничивает общность рассуждений, так как в случае $d_0 > 0$ реальная длина регистра меньше n);
- $E = \{e_0, \ldots, e_l\}$ — множество номеров переменных, от которых функция обратной связи f зависит нелинейно (нелинейных точек съёма), отсюда $E \subseteq D$, $0 < l \leqslant m$, $0 \leqslant e_0 < e_l \leqslant d_m$;
- $d(u)$ — наибольшее число из D, такое, что $d(u) \leqslant u$, $0 \leqslant u < n$, число $d(u)$ существует для любого u в силу равенства $0 = d_0$;
- $e(u)$ — наибольшее число из E, такое, что $e(u) \leqslant u$, где $e_0 \leqslant u < n$; при $0 \leqslant u < e_0$ число $e(u)$ не существует; по определению $e(u) \leqslant d(u)$ для любого $u \geqslant e_0$;
- (i, j) — дуга в орграфе, инцидентная вершинам i и j, (i, j) — петля при $i = j$;
- $w(i, j)$ — путь в орграфе из i в j; $w(i, j)$ — путь нулевой длины при $i = j$;
- $\circ$ — операция конкатенации (присоединения) путей, где конечная вершина первого пути совпадает с начальной вершиной второго пути;
- $\Gamma(\phi_f)$ — перемешивающий орграф преобразования ϕ_f регистра сдвига, имеющий множество вершин Y_n, дуга (i, j) имеется в графе $\Gamma(\phi_f)$ тогда и только тогда, когда j-я координатная функция преобразования ϕ_f зависит от переменной x_i существенно, $0 \leqslant i, j < n$; заметим, что при $d_m = n-1$ граф $\Gamma(\phi_f)$ имеет петлю в вершине $n-1$;
- $\Gamma_\theta(\phi_f)$ — орграф нелинейности преобразования ϕ_f регистра сдвига, имеющий множество вершин Y_n, дуга (i, j) которого помечена символом $a_{i,j}$, равным 0, 1 или 2 тогда и только тогда, когда j-я координатная функция преобразования ϕ_f зависит от переменной x_i несущественно, линейно или нелинейно соответственно, $0 \leqslant i, j < n$.

2. Определяющие свойства помеченных орграфов

Напомним определения и свойства [5, 6, 8], связанные с орграфами.

Определим полугрупповую коммутативную операцию умножения на множестве $G = \{0, 1, 2\}$. Для любых $a, b \in G$ положим: $0a = 0$, $b = \max\{a, b\}$, если $a, b \in \{1, 2\}$.

На множестве помеченных n-вершинных орграфов определена операция умножения орграфов: если в Γ_0 имеется дуга (i, j) с меткой $m_0 \in G$ и в Γ_1 имеется дуга (j, k) с меткой $m_1 \in G$, то в орграфе $\Gamma_0\Gamma_1$ имеется дуга (i, k) с меткой $m_0 m_1 \in G$, где умножение меток выполняется в полугруппе G.

Меткой пути (контура) назовём наибольшую ненулевую метку всех дуг, составляющих данный путь (контур). Путь с меткой «2» назовём 2-путём.

Сильносвязный орграф Γ называется примитивным, если существует $\gamma \in \mathbb{N}$, такое, что орграф Γ^γ (с петлями) является полным. Наименьшее такое число γ обозначается $\exp\Gamma$ и называется экспонентом орграфа Γ [1].

Сильносвязный помеченный орграф Γ называется $\langle 2\rangle$-примитивным, если существует $\gamma \in \mathbb{N}$, такое, что орграф Γ^γ (с петлями) есть полный 2-граф, то есть полный граф, в котором каждая дуга имеет метку «2». Наименьшее такое число γ обозначается $\langle 2\rangle$-$\exp\Gamma$ и называется $\langle 2\rangle$-экспонентом орграфа Γ.

Орграф Γ $\langle 2\rangle$-примитивный тогда и только тогда, когда он примитивный и имеет дугу с меткой «2», при этом

$$\langle 2\rangle\text{-}\exp\Gamma \leqslant \max\{\omega_2, \delta_2\} + \exp\Gamma, \tag{1}$$

где в Γ обозначено: ω_2 — наибольшая из длин кратчайших 2-путей, исходящих из всех вершин; δ_2 — наибольшая из длин кратчайших 2-путей, заходящих во все вершины.

Орграф Γ называется $i\times j$-$\langle 2\rangle$-примитивным, если существует $\gamma \in \mathbb{N}$, такое, что для любого $t \geqslant \gamma$ в орграфе Γ^t метка дуги (i, j) есть «2». Наименьшее такое γ обозначается $\gamma_{i,j}^{[2]} = i \times j$-$\langle 2\rangle$-$\exp\Gamma$ и называется $i \times j$-$\langle 2\rangle$-экспонентом (локальным экспонентом) орграфа Γ [8]. Из данных определений следует, что

$$\langle 2\rangle\text{-}\exp\Gamma = \max_{0\leqslant i,j<n}\{i \times j\text{-}\langle 2\rangle\text{-}\exp\Gamma\}. \tag{2}$$

В связи с операцией в G определена операция умножения на множестве троичных матриц с элементами из полугруппы G. Если $A = (a_{i,j})$, $B = (b_{i,j})$, $AB = C = (c_{i,j})$, то

$$c_{i,j} = \max_{0\leqslant k<n}\{a_{i,k}b_{k,j}\},\ 0 \leqslant i < n,\ 0 \leqslant j < n,$$

где умножение элементов матриц выполняется в полугруппе G.

Матрица смежности вершин орграфа, дуги которого помечены символами «1» и «2», — это троичная матрица, где нулевой элемент в i-й строке и j-м столбце означает отсутствие дуги (i, j) в орграфе.

Матрица смежности произведения помеченных орграфов равна произведению троичных матриц смежности умножаемых орграфов. Если дуги (i, k) и (k, j) умножаемых орграфов помечены числами $a, b \in \{1, 2\}$, то в произведении орграфов дуга (i, j) имеет метку $\max\{a, b\}$. Орграф, в котором все дуги имеют метку «2», называется 2-графом.

В данной работе оценка (1) уточнена в терминах характеристик регистрового преобразования пространства V_n.

3. Свойства орграфа нелинейности регистра сдвига

Преобразование g множества V_n называется преобразованием регистра левого сдвига с обратной связью $f(x_0, \ldots, x_{n-1})$, если $g(x_0, \ldots, x_{n-1}) = (x_1, \ldots, x_{n-1}, f(x_0, \ldots, x_{n-1}))$.

Отметим свойства орграфа нелинейности $\Gamma_\theta(\phi_f)$ и функций, связанных с регистром сдвига. Для краткости используем обозначение $\Gamma_\theta(\phi_f) = \Gamma$.

Свойство 1. В соответствии с определением преобразования регистра сдвига множество дуг орграфа нелинейности Γ с меткой «1» равно

$$\{(1, 0), (2, 1), \ldots, (n-1, n-2)\} \cup \bigcup_{s\in D\setminus E}\{(s, n-1)\},$$

и множество дуг орграфа нелинейности Γ с меткой «2» — это

$$\bigcup_{s\in E}\{(s, n-1)\}.$$

Следовательно, орграф нелинейности Γ есть объединение $m+1$ контуров $C_0, \ldots, C_m$, где $C_s = (n-1, n-2, \ldots, d_s)$, $s = 0, \ldots, m-1$; $C_m = (n-1, \ldots, d_m)$, если $d_m < n-1$, и $C_m = (n-1)$ — петля в вершине $n-1$ при $d_m = n-1$. Отсюда следует:

— орграф Γ сильносвязный;

— контур C_s есть 2-контур тогда и только тогда, когда $d_s \in E$.

Свойство 2. Длина контура C_s равна $n - d_s$, $s = 0, \ldots, m$. Наименьшая из длин 2-контуров равна $(n - e_l)$, наименьшая — $(n - d_m)$.

Обозначим через L множество длин всех контуров орграфа Γ:

$$L = \{n, n-d_1, \ldots, n-d_m\}.$$

Отсюда следует, что орграф Γ примитивный тогда и только тогда, когда НОД$(L) = 1$. В частности, наличие петли в графе Γ достаточно для его примитивности.

Свойство 3. В силу нелинейности функции обратной связи примитивный орграф Γ является $\langle 2\rangle$-примитивным.

Свойство 4. Функции $d(u)$ и $e(u)$ монотонны по аргументу u на множествах $\{0, \ldots, n-1\}$ и $\{e_0, \ldots, n-1\}$ соответственно.

4. Оценка $\langle 2\rangle$-экспонента орграфа нелинейности

Получим сначала оценку $i \times j$-$\langle 2\rangle$-экспонента орграфа Γ.

Обозначим: $\gamma^{[2]} = \langle 2\rangle\text{-}\exp\Gamma$; $w(n-1, j)$ — путь длины $n-1-j$, являющийся частью контура C_0; C — кратчайший 2-контур длины $n - e_l$, пройденный из вершины $n-1$; $C'(t)$ — контур длины t, пройденный из вершины $n-1$ через некоторые контуры (возможно, неоднократно) орграфа Γ;

$$\mu(u) = \begin{cases} \min\{u - e(u), u - d(u) + n - e_l\}, & e_0 \leqslant u < n; \\ u - d(u) + n - e_l, & 0 \leqslant u < e_0. \end{cases}$$

Таким образом, $\mu(u)+1$ есть длина проходящего через дугу с меткой «2» кратчайшего пути из вершины u в вершину $n-1$, $u \in Y_n$.

Теорема 1. Если орграф Γ примитивный, то

$$\gamma^{[2]}_{i,j} \leqslant F(L) + 1 + n - j + \mu(i).$$

Доказательство. Оценим наименьшее τ, при котором из вершины i в вершину j имеется 2-путь любой длины, не меньшей τ.

Построим 2-пути w_0 и w_1 из i в j с помощью конкатенации:

$$w_0(i,j) = w(i, e(i)) \circ (e(i), n-1) \circ C'(t) \circ w(n-1, j), \quad e_0 \leqslant i < n;$$
$$w_1(i,j) = w(i, d(i)) \circ (d(i), n-1) \circ C \circ C'(t) \circ w(n-1, j), \quad 0 \leqslant i < n.$$

Каждый из них есть 2-путь, так как проходит либо через дугу $(e(i), n-1)$ с меткой «2», либо через 2-контур C.

В силу взаимной простоты чисел множества L, вытекающей из примитивности орграфа Γ, при подходящем построении последовательности контуров орграфа Γ

длина контура $C'(t)$ может быть равна любому значению $t > F(L)$. Следовательно, если $e_0 \leqslant i < n$, то длина 2-пути $w_0(i,j)$ может быть любой не меньшей $i - e(i) + F(L) + 1 + n - j$, если $e_0 \leqslant i < n$, а длина 2-пути $w_1(i,j)$ может быть любой не меньшей $i - d(i) + F(L) + 1 + n - j + n - e_l$, если $0 \leqslant i < n$. Следовательно, в орграфе Γ имеется 2-путь из i в j любой длины $\tau \geqslant F(L) + 1 + n - j + \mu(i)$. ■

Обозначим: z_m — метка контура C_m; λ — длина кратчайшего 2-контура.

Следствие 1. Если переменная x_{n-1} существенная для $f(x_0, \ldots, x_{n-1})$, то

$$\gamma_{i,j}^{[2]} = \begin{cases} i - e(i) + n - j, & e_0 \leqslant i < n,\ d(i) - e(i) \leqslant \lambda, \\ i - d(i) + 1 + n - j, & 0 \leqslant i < e_0 \text{ или } d(i) - e(i) > \lambda,\ z_m = 2, \\ i - d(i) + \lambda + n - j, & 0 \leqslant i < e_0 \text{ или } d(i) - e(i) > \lambda,\ z_m = 1. \end{cases}$$

Доказательство. В этих условиях контур C_m — это петля в вершине $n-1$, значит, $1 \in L$ и по определению $F(L) = -1$. Тогда из теоремы 1 следует, что

$$\gamma_{i,j}^{[2]} \leqslant n - j + \mu(i).$$

Заметим, что в Γ кратчайший 2-путь из i в j при $e_0 \leqslant i < n$ — это

$w(i, e(i)) \circ (e(i), n-1) \circ w(n-1, j)$, если $d(i) - e(i) \leqslant \lambda$;
$w(i, d(i)) \circ (d(i), n-1) \circ C_m \circ w(n-1, j)$, если $d(i) - e(i) > \lambda$ и $z_m = 2$;
$w(i, d(i)) \circ (d(i), n-1) \circ C \circ w(n-1, j)$, если $d(i) - e(i) > \lambda$, $z_m = 1$ и C есть кратчайший 2-контур.

При $0 \leqslant i < e_0$ кратчайший 2-путь из i в j — это

$w(i, d(i)) \circ (d(i), n-1) \circ C_m \circ w(n-1, j)$, если $z_m = 2$;
$w(i, d(i)) \circ (d(i), n-1) \circ C \circ w(n-1, j)$, если $z_m = 1$.

Во всех случаях не существует 2-пути меньшей длины в силу размещения меток «2» в орграфе Γ (свойство 1). Пути любой длины больше указанной имеются, так как каждый путь проходит через вершину $n-1$ с петлей. ■

Следствие 2. Если функция обратной связи $f(x_0, \ldots, x_{n-1})$ нелинейна по всем своим существенным переменным, то

$$\gamma_{i,j}^{[2]} \leqslant F(L) + 1 + n - j + i - e(i);$$

если при этом переменная x_{n-1} существенная для $f(x_0, \ldots, x_{n-1})$, то

$$\gamma_{i,j}^{[2]} = n - j + i - e(i).$$

Доказательство. В данных условиях $e(i) = d(i)$ для всех i и $e_0 = 0$, значит, $\mu(u) = u - e(u)$, где $0 \leqslant u < n$. Отсюда из теоремы 1 следует нужная оценка.

Если переменная x_{n-1} существенная, то из следствия 1 получаем значение локального экспонента. ■

Для оценки $\gamma_{i,j}^{[2]}$ обозначим:

$$D^0 = \{d_s \in D, 1 \leqslant s \leqslant m : d_{s-1} - e(d_{s-1}) \leqslant \lambda, e_0 \leqslant d_{s-1} < n\} \cup S_0(n),$$
$$D^1 = \{d_s \in D, 1 \leqslant s \leqslant m : 0 \leqslant d_{s-1} < e_0 \text{ или } d_{s-1} - e(d_{s-1}) > \lambda\} \cup S_1(n),$$

где $S_0(n) = S_1(n) = \varnothing$ при $d_m = n-1$; $S_0(n) = \{n\}$ при $d_m < n-1$ и $d_m \in E$; $S_1(n) = \{n\}$ при $d_m < n-1$ и $d_m \notin E$;

$$\Delta_f = \max_{i \in D^0 \cup D^1} \mu(i-1).$$

По определению $D^1 = \varnothing$, если при $d_m < n-1$ нелинейны все существенные переменные функции $f(x_0, \ldots, x_{n-1})$ или при $d_m = n-1$ нелинейны все существенные переменные, кроме, быть может, x_{d_m}.

Теорема 2. Если орграф Γ примитивный, то

$$\gamma^{[2]} \leqslant F(L) + 1 + n + \Delta_f.$$

Доказательство. В силу теоремы 1 при любом фиксированном i значение $\gamma^{[2]}_{i,j}$ наибольшее при $j = 0$. Тогда из равенства (2) и теоремы 1 имеем

$$\gamma^{[2]} \leqslant F(L) + 1 + n + \max_{0 \leqslant i < n} \mu(i). \tag{3}$$

Функция $\mu(u)$ монотонна по переменной u при $d_{s-1} \leqslant u < d_s$, $s = 1, \ldots, m$, и при $d_m \leqslant u < n$, если $d_m < n-1$. При этом если $d_{s-1} \leqslant i \leqslant u < d_s$, то

$$i - e(i) + n - e_l \leqslant u - e(u) + n - e_l.$$

Поэтому в правой части неравенства (3) можно сузить множество, по которому берется максимум, то есть выполнено неравенство

$$\gamma^{[2]} \leqslant F(L) + 1 + n + \max_{i \in D^0 \cup D^1} \{\mu(i-1)\}.$$

Теорема 2 доказана. ■

Следствие 3. Если переменная x_{n-1} существенная для $f(x_0, \ldots, x_{n-1})$, то

$$\gamma^{[2]} = n + \max\{\max_{i \in D^0}(i - e(i-1)), \xi(z_m) + \max_{i \in D^1}(i - d(i-1))\} - 1,$$

где $\xi(z_m) = \begin{cases} 1, & z_m = 2, \\ \lambda, & z_m = 1. \end{cases}$

Доказательство. В данных условиях из следствия 1 при $j = 0$ с учётом теоремы 2 получаем

$$\max_{i \in D^0} \gamma^{[2]}_{i,j} = n + \max_{i \in D^0}(i - e(i-1)) - 1,$$

$$\max_{i \in D^1} \gamma^{[2]}_{i,j} = \begin{cases} n + 1 + \max\limits_{i \in D^1}(i - d(i-1)) - 1, & z_m = 2, \\ n + \lambda + \max\limits_{i \in D^1}(i - d(i-1)) - 1, & z_m = 1. \end{cases}$$

Отсюда в соответствии с равенством (2) следует нужная формула. ■

Пример 1. Определим точные значения $\langle 2 \rangle$-экспонентов и локальных $\langle 2 \rangle$-экспонентов орграфа нелинейности регистрового преобразования с обратной связью $f(x_0, \ldots, x_{11}) = x_0 \oplus x_3x_5 \oplus x_7x_8x_{11}$ (рис. 1).

Так как переменная x_{11} существенная для функции обратной связи $f(x_0, \ldots, x_{11})$, точное значение $\langle 2 \rangle$-экспонента орграфа нелинейности регистрового преобразования определяется следствием 3, а локальные $\langle 2 \rangle$-экспоненты орграфа — следствием 1.

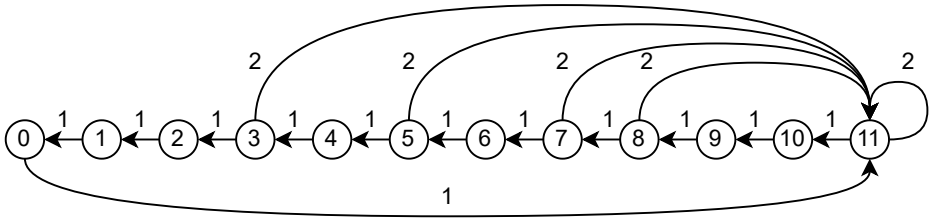


Рис. 1. Орграф нелинейности преобразования из примера 1

Параметры данного регистра сдвига: $n = 12$, $m = 5$, $l = 4$, $D = \{0, 3, 5, 7, 8, 11\}$, $E = \{3, 5, 7, 8, 11\}$, отсюда $e_0 = 3$, $D^0 = \{5, 7, 8, 11\}$, $D^1 = \{3\}$, так как все точки съёма являются нелинейными, за исключением нулевой. Метка пятого контура $z_5 = 2$, отсюда $\xi(z_5) = 1$, длина кратчайшего 2-контура $\lambda = 1$. Так как $d(2) = 0$, то

$$\xi(z_5) + \max_{i \in D^1}\{i - d(i-1)\} = \xi(z_5) + 3 - d(2) = 4.$$

Так как $e(4) = 3$, $e(6) = 5$, $e(7) = 7$, $e(10) = 8$, то

$$\max_{i \in D^0}\{i - e(i-1)\} = \max\{5-3, 7-5, 8-7, 11-8\} = 3.$$

Тогда, согласно следствию 3, $\gamma^{[2]} = 12 + \max\{4, 3\} - 1 = 15$.

Определим значения локальных $\langle 2 \rangle$-экспонентов орграфа.

Если $3 \leqslant i < 12$, то $d(i) - e(i) \leqslant \lambda$ и $\gamma^{[2]}_{i,j} = i - e(i) + 12 - j$.

При $0 \leqslant i < 3$ имеем $\gamma^{[2]}_{i,j} = i - d(i) + 1 + 12 - j$.

Наибольшие значения получаются при $j = 0$, максимальное из них равно 15.

Пример 2. Оценим значения $\langle 2 \rangle$-экспонентов и локальных $\langle 2 \rangle$-экспонентов орграфа нелинейности регистрового преобразования с обратной связью $f(x_0, \ldots, x_{11}) = x_0 \oplus x_3x_5 \oplus x_4x_6x_7$ (рис. 2).

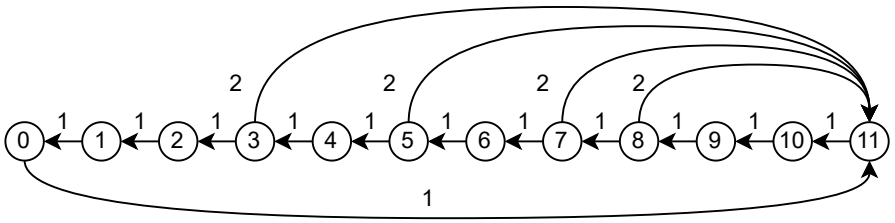


Рис. 2. Орграф нелинейности преобразования из примера 2

Параметры данного регистра сдвига: $n = 12$, $m = 5$, $l = 4$, множество длин простых контуров $L = \{12, 9, 8, 7, 6, 5\}$, $D = \{0, 3, 4, 5, 6, 7\}$, $E = \{3, 4, 5, 6, 7\}$, отсюда $e_0 = 3$, $D^0 = \{4, 5, 6, 7\}$, $D^1 = \{3\}$, так как все точки съёма являются нелинейными, за исключением нулевой. Длина кратчайшего 2-контура $\lambda = 5$.

Оценим сверху с помощью теоремы 1 локальные $\langle 2 \rangle$-экспоненты орграфа. Так как НОД$(L) = 1$, то по свойству 2 орграф нелинейности $\langle 2 \rangle$-примитивный и верны оценки теоремы 1. Вычисляем: $F(L) = 4$, так как любое число больше 4 представимо линейной комбинацией чисел из L с неотрицательными целыми коэффициентами.

Так как $\lambda = 5$, то $\mu(i) = i - e(i)$, если $3 \leqslant i < 12$, и $\mu(i) = i - d(i) + 5$, если $0 \leqslant i < 3$. В табл. 1 приведены значения функции $\mu(i)$ при $0 \leqslant i < 12$, в табл. 2 — значения оценок локальных $\langle 2 \rangle$-экспонентов.

Таблица 1

Значения функции $\mu(i)$ из примера 2

i	0	1	2	3	4	5	6	7	8	9	10	11
$\mu(i)$	5	6	7	0	0	0	0	0	1	2	3	4

Таблица 2

Верхние оценки значений $i \times j$-$\langle 2 \rangle$-экспонентов из примера 2

i	j											
	0	1	2	3	4	5	6	7	8	9	10	11
0	22	21	20	19	18	17	16	15	14	13	12	11
1	23	22	21	20	19	18	17	16	15	14	13	12
2	24	23	22	21	20	19	18	17	16	15	14	13
3	17	16	15	14	13	12	11	10	9	8	7	6
4	17	16	15	14	13	12	11	10	9	8	7	6
5	17	16	15	14	13	12	11	10	9	8	7	6
6	17	16	15	14	13	12	11	10	9	8	7	6
7	17	16	15	14	13	12	11	10	9	8	7	6
8	18	17	16	15	14	13	12	11	10	9	8	7
9	19	18	17	16	15	14	13	12	11	10	9	8
10	20	19	18	17	16	15	14	13	12	11	10	9
11	21	20	19	18	17	16	15	14	13	12	11	10

Из табл. 2 в соответствии с (2) получаем оценку $\langle 2 \rangle$-экспонента орграфа нелинейности: $\gamma^{[2]} \leqslant 24$.

ЛИТЕРАТУРА

1. *Фомичёв В. М., Авезова Я Э., Коренева А. М., Кяжин С. Н.* Примитивность и локальная примитивность орграфов и неотрицательных матриц // Дискретный анализ и исследование операций. 2018. Т. 25. № 3. С. 95–125.
2. *Frobenius G.* Über Matrizen aus nicht negativen Elementen // Sitzungsber K. Preuss. Akad. Wiss. Berlin, 1912. P. 456–477.
3. *Dulmage A. L. and Mendelsohn N. S.* The exponent of a primitive matrix // Canadian Math. Bull. 1962. No. 5. P. 241–244.
4. *Perkins P.* A theorem on regular graphs // Pacific J. Math. 1961. V. 2. P. 1529–1533.
5. *Фомичёв В. М.* Оценка характеристик нелинейности итеративных преобразований векторного пространства // Дискретный анализ и исследование операций. 2020. Т. 27. № 4. С. 131–151.
6. *Fomichev V. M. and Koreneva A. M.* Encryption performance and security of certain wide block ciphers // J. Computer Virology Hacking Tech. 2020. V. 16. No. 1. P. 197–216.
7. *Фомичёв В. М., Авезова Я. Э.* Точная формула экспонентов перемешивающих орграфов регистровых преобразований // Дискретный анализ и исследование операций. 2020. Т. 27. № 2. С. 117–135.
8. *Фомичёв В. М., Бобров В. М.* Оценка с помощью матрично-графового подхода характеристик локальной нелинейности итераций преобразований векторных пространств // Прикладная дикретная математика. Приложение. 2019. № 12. С. 32–35.

REFERENCES

1. *Fomichev V. M., Avezova Ya. E., Koreneva A. M., and Kyazhin S. N.* Primitivity and local primitivity of digraphs and nonnegative matrices. J. Appl. Industr. Math., 2018, vol. 12, no. 3, pp. 453–469.
2. *Frobenius G.* Über Matrizen aus nicht negativen Elementen. Sitzungsber K. Preuss. Akad. Wiss., Berlin, 1912, pp. 456–477.
3. *Dulmage A. L. and Mendelsohn N. S.* The exponent of a primitive matrix. Canadian Math. Bull., 1962, no. 5, pp. 241–244.
4. *Perkins P.* A theorem on regular graphs. Pacific J. Math., 1961, vol. 2, pp. 1529–1533.
5. *Fomichev V. M.* Estimating nonlinearity characteristics for iterative transformations of a vector space. Appl. Industr. Math., 2020, vol. 14, no. 4, pp. 610–622.
6. *Fomichev V. M. and Koreneva A. M.* Encryption performance and security of certain wide block ciphers. J. Computer Virology Hacking Tech., 2020, vol. 16, no. 1, pp. 197–216.
7. *Fomichev V. M. and Avezova Ya. E.* Exact formula for exponents of mixing digraphs for register transformations. J. Appl. Industr. Math., 2020, vol. 14, no. 2, pp. 308–319.
8. *Fomichev V. M. and Bobrov V. M.* Otsenka s pomoshch'yu matrichno-grafovogo podkhoda kharakteristik lokal'noy nelineynosti iteratsiy preobrazovaniy vektornykh prostranstv [Estimation of local nonlinearity characteristics of vector space transformation iteration using matrix-graph approach]. Prikladnaya Diksretnaya Matematika. Prilozheniye, 2019, no. 12, pp. 32–35. (in Russian)

УДК 519.17 DOI 10.17223/20710410/55/6

UNIQUE LIST COLORABILITY OF THE GRAPH $K_2^n + K_r$

L. X. Hung

Hanoi University for Natural Resources and Environment, Hanoi, Vietnam

E-mail: lxhung@hunre.edu.vn

Given a list $L(v)$ for each vertex v, we say that the graph G is L-colorable if there is a proper vertex coloring of G, where each vertex v takes its color from $L(v)$. The graph is uniquely k-list colorable if there is a list assignment L such that $|L(v)| = k$ for every vertex v and the graph has exactly one L-coloring with these lists. If a graph G is not uniquely k-list colorable, we also say that G has property $M(k)$. The least integer k such that G has the property $M(k)$ is called the m-number of G, denoted by $m(G)$. In this paper, we characterize the unique list colorability of the graph $G = K_2^n + K_r$. In particular, we determine the number $m(G)$ of the graph $G = K_2^n + K_r$.

Keywords: *vertex coloring, list coloring, uniquely list colorable graph, complete r-partite graph.*

ОДНОЗНАЧНАЯ СПИСОЧНАЯ РАСКРАШИВАЕМОСТЬ ГРАФА $K_2^n + K_r$

Л. Хунг

Ханойский университет природных ресурсов и окружающей среды, г. Ханой, Вьетнам

Имея список $L(v)$ для каждой вершины v, мы говорим, что граф L-раскрашиваем, если существует правильная раскраска его вершин, в которой каждая вершина v окрашена цветом из $L(v)$. Граф является однозначно k-раскрашиваемым, если существует список L, такой, что $|L(v)| = k$ для каждой вершины v и граф имеют ровно одну L-раскраску. Если граф G не является однозначно k-раскрашиваемым, то G обладает свойством $M(k)$. Наименьшее целое число k, такое, что G обладает свойством $M(k)$, называется m-числом графа G и обозначается $m(G)$. В работе охарактеризована однозначность списочной раскрашиваемости графа $G = K_2^n + K_r$, в частности определено значение $m(G)$ этого графа.

Ключевые слова: *раскраска вершин графа, раскраска списком, однозначно раскрашиваемый граф, полный r-долевой граф.*

1. Introduction

All graphs considered in this paper are finite undirected graphs without loops or multiple edges. If G is a graph, then $V(G)$ and $E(G)$ (or V and E in short) will denote its vertex-set and its edge-set, respectively. The set of all neighbours of a subset $S \subseteq V(G)$ is denoted by $N_G(S)$ (or $N(S)$ in short). The subgraph of G induced by $W \subseteq V(G)$ is denoted by $G[W]$. The empty graphs (independent sets) and complete graphs of order n are denoted by O_n and K_n, respectively. Unless otherwise indicated, our graph-theoretic terminology will follow [1].

A graph $G = (V, E)$ is called *r-partite graph* if V admits a partition $V = V_1 \cup V_2 \cup \ldots \cup V_r$ such that the subgraphs of G induced by V_i, $i = 1, \ldots, r$, are empty. An r-partite graph

in which every two vertices from different partition classes are adjacent is called complete r-partite graph and is denoted by $K_{|V_1|,|V_2|,\ldots,|V_r|}$. The complete r-partite graph $K_{|V_1|,|V_2|,\ldots,|V_r|}$ with $|V_1| = |V_2| = \ldots = |V_r| = s$ is denoted by K_s^r.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \varnothing$. Their *union* $G = G_1 \cup G_2$ has, as expected, $V(G) = V_1 \cup V_2$ and $E(G) = E_1 \cup E_2$. Their *join* is denoted by $G_1 + G_2$ and consists of $G_1 \cup G_2$ and all edges joining V_1 with V_2.

The *complement* $\overline{G} = (\overline{V}, \overline{E})$ of $G = (V, E)$ is the graph with $\overline{V} = V$ and for every $u, v \in V$, $uv \in \overline{E}$ if and only if $uv \notin E$.

Let $G = (V, E)$ be a graph and λ is a positive integer.

A *λ-coloring* of G is a mapping $f : V(G) \to \{1, 2, \ldots, \lambda\}$ such that $f(u) \neq f(v)$ for any adjacent vertices $u, v \in V(G)$. The smallest positive integer λ such that G has a λ-coloring is called the *chromatic number* of G and is denoted by $\chi(G)$. We say that a graph G is *n-chromatic* if $n = \chi(G)$.

Let $(L(v))_{v \in V}$ be a family of sets. We call a coloring f of G with $f(v) \in L(v)$ for all $v \in V$ a *list coloring from the lists* $L(v)$. We will refer to such a coloring as an L-coloring. The graph G is called *λ-list-colorable*, or *λ-choosable*, if for every family $(L(v))_{v \in V}$ with $|L(v)| = \lambda$ for all v, there is a coloring of G from the lists $L(v)$. The smallest positive integer λ such that G is λ-choosable is called the *list-chromatic number*, or *choice number* of G and is denoted by $ch(G)$. The idea of list colorings of graphs was given independently by V. G. Vizing [2], P. Erdös, A. L. Rubin, and H. Taylor [3].

Let G be a graph with n vertices and suppose that for each vertex v in G, there exists a list of k colors $L(v)$, such that there exists a unique L-coloring for G, then G is called a *uniquely k-list colorable graph* or a UkLC graph in short. If a graph G is not uniquely k-list colorable, we also say that G has property $M(k)$. So G has the property $M(k)$ if and only if for any collection of lists assigned to its vertices, each of size k, either there is no list coloring for G or there exist at least two list colorings. The smallest positive integer k such that G has the property $M(k)$ is called the *m-number* of G, denoted by $m(G)$. The idea of uniquely colorable graph was introduced in [4, 5].

For example, one can easily see that the graph $G = K_{1,1,2}$ is U2LC and it has the property M(3), i.e., $m(G) = 3$. Indeed, let $V(G) = \{u_1, u_2, v_1, v_2\}$, $E(G) = \{u_1v_1, u_1v_2, u_2v_1, u_2v_2, v_1v_2\}$. We assign the following lists for the vertices: $L(u_1) = \{1, 3\}$, $L(u_2) = = \{2, 3\}$, $L(v_1) = \{1, 3\}$, and $L(v_2) = \{2, 3\}$. Then, a unique coloring f of G exists from the assigned lists: $f(u_1) = f(u_2) = 3$, $f(v_1) = 1$, $f(v_2) = 2$. Thus, G is U2LC. If $G = K_{1,1,2}$ is U3LC, then there exists lists for the vertices $L(u_1) = \{a_{11}, a_{12}, a_{13}\}$, $L(u_2) = \{a_{21}, a_{22}, a_{23}\}$, $L(v_1) = \{b_{11}, b_{12}, b_{13}\}$, and $L(v_2) = \{b_{21}, b_{22}, b_{23}\}$ such that there exists a unique coloring f of G, we may assume that $f(u_1) = a_{11}$, $f(u_2) = a_{21}$, $f(v_1) = b_{11}$, $f(v_2) = b_{21}$. If there exists $x \in \{a_{12}, a_{13}\}$ such that $x \notin \{b_{11}, b_{21}\}$, then there is a coloring g of G with $g(u_1) = x$, $g(u_2) = a_{21}$, $g(v_1) = b_{11}$, and $g(v_2) = b_{21}$, it follows that $g \neq f$, a contradiction. So $\{a_{12}, a_{13}\} = \{b_{11}, b_{21}\}$. Similarly, we can show that $\{a_{22}, a_{23}\} = \{b_{11}, b_{21}\}$. If $a_{11} \in \{b_{12}, b_{13}\}$, then there is a coloring g of G with $g(u_1) = b_{11}$, $g(u_2) = b_{11}$, $g(v_1) = a_{11}$, and $g(v_2) = b_{21}$, it follows that $g \neq f$, a contradiction. So $a_{11} \notin \{b_{12}, b_{13}\}$. Similarly, we can show that $a_{21} \notin \{b_{12}, b_{13}\}$. Let $y \in \{b_{12}, b_{13}\} \setminus \{b_{21}\}$. Then there is a coloring g of G with $g(u_1) = a_{11}$, $g(u_2) = a_{21}$, $g(v_1) = y$, and $g(v_2) = b_{21}$, it follows that $g \neq f$, a contradiction. Thus, G is not U3LC.

The list coloring model can be used in the channel assignment. The fixed channel allocation scheme leads to low channel utilization across the whole channel. It requires a more effective channel assignment and management policy, which allows unused parts of channel to become available temporarily for other usages so that the scarcity of the channel

can be largely mitigated [6]. It is a discrete optimization problem. A model for channel availability, observed by the secondary users, is introduced in [6]. The research of list coloring consists of two parts: choosability and unique list colorability. In [7], we characterized list-chromatic number of the graph $G = K_2^n + O_r$, we have proved that $ch(G) = n + 1$ if $1 \leqslant r \leqslant 2$. In [8], we characterized list-chromatic number and characterized unique list colorability of the graph $G = K_2^n + K_r$, we have proved that $ch(G) = n + r$, G is U3LC if and only if $2n + r \geqslant 7$ and $n \geqslant 2$.

In this paper, we continue to characterize the unique list colorability of the graph $G = K_2^n + K_r$. In particular, we determine the number $m(G)$ of the graph $G = K_2^n + K_r$.

2. Preliminaries

We need the following Lemmas 1–10 to prove our results.

Lemma 1 [5]. Each UkLC graph is also a U$(k-1)$LC graph.

Lemma 2 [5]. The graph G is UkLC if and only if $k < m(G)$.

Lemma 3 [5]. A connected graph G has the property $M(2)$ if and only if every block of G is either a cycle, a complete graph, or a complete bipartite graph.

Lemma 4 [9]. For every graph G we have $m(G) \leqslant |E(\overline{G})| + 2$.

Lemma 5 [9]. Every UkLC graph has at least $3k - 2$ vertices.

Lemma 6. With $G = K_2^n + K_r$, we have $m(G) \leqslant n + 2$.

Proof. It is clear that $|E(\overline{G})| = n$. By Lemma 4, $m(G) \leqslant n + 2$. ■

Lemma 7.

(i) If $n = 1$ and $r = 1$, then $G = K_2^n + K_r$ has the property $M(2)$.

(ii) If $n = 1$ and $r \geqslant 2$, then $m(K_2^n + K_r) = 3$.

Proof.

(i) If $n = 1$ and $r = 1$, then $G = K_2^n + K_r$ is a complete bipartite graph, then by Lemma 3, G has the property $M(2)$.

(ii) By Lemma 3, $G = K_2^n + K_r$ is U2LC. It is not difficult to see that $|E(\overline{G})| = 1$. By Lemma 4, $m(K_2^n + K_r) \leqslant 3$. Thus, $m(K_2^n + K_r) = 3$ if $n = 1$ and $r \geqslant 2$. ■

Lemma 8. $m(K_2^2 + K_r) = 3$ for every $1 \leqslant r \leqslant 2$.

Proof. By Lemma 3, $G = K_2^2 + K_r$ is U2LC. Suppose that G is U3LC. By Lemma 5, $|V(G)| \geqslant 7$, a contradiction. So $m(G) = 3$. ■

Lemma 9 [8]. $G = K_2^2 + K_r$ is U3LC for every $r \geqslant 3$.

Lemma 10. $m(K_2^2 + K_r) = 4$ if and only if $r \geqslant 3$.

Proof. Suppose that $m(K_2^2 + K_r) = 4$. If $1 \leqslant r \leqslant 2$, then by Lemma 8, $m(K_2^2 + K_r) = 3$, a contradiction.

Suppose that $r \geqslant 3$. By Lemma 9, $G = K_2^2 + K_r$ is U3LC. So $m(K_2^2 + K_r) \geqslant 4$. By Lemma 6, $m(K_2^2 + K_r) \leqslant 4$. Thus, $m(K_2^2 + K_r) = 4$. ■

3. Main Results

Theorem 1. Let $K_2^{n-1} + K_{r-1}$ is UkLC for every $n, r \geqslant 2$. Then

(i) $K_2^n + K_{r-1}$ is UkLC and

$$m(K_2^{n-1} + K_{r-1}) \leqslant m(K_2^n + K_{r-1}) \leqslant m(K_2^{n-1} + K_{r-1}) + 2;$$

(ii) $K_2^{n-1} + K_r$ is UkLC and

$$m(K_2^{n-1} + K_{r-1}) \leqslant m(K_2^{n-1} + K_r) \leqslant m(K_2^{n-1} + K_{r-1}) + 1.$$

Proof.

(i) We prove $G = K_2^n + K_{r-1}$ is UkLC by induction on n. If $n = 2$, then by Lemma 7 to Lemma 10, we deduce what to prove. So let $n > 2$ and assume the assertion for smaller values of n.

Let $V(G) = V_1 \cup V_2 \cup V_3 \cup \ldots \cup V_{n+r-1}$ is a partition of $V(G)$ such that $|V_1| = |V_2| = \ldots = |V_n| = 2$, $|V_{n+1}| = |V_{n+2}| = \ldots = |V_{n+r-1}| = 1$ and for every $i = 1, 2, \ldots, n$ the subgraph of G induced by V_i is an independent set. Set $V_i = \{u_{i1}, u_{i2}\}$ for every $i = 1, \ldots, n$ and $G' = G - V_n$. By the induction hypothesis, for each vertex v in G', there exists a list of k colors $L'(v)$, such that there exists a unique f' for G'. We assign the following lists for the vertices of G:

$$L(u_{n1}) = L(u_{n2}) = \{f'(u_{11}), f'(u_{21}), \ldots, f'(u_{(k-1)1}), l\},$$

with $l \notin f'(G')$, $L(v) = L'(v)$ if $v \in V(G')$. A unique coloring f of G exists from the assigned lists: $f(u_{n1}) = f(u_{n2}) = l$, $f(v) = f'(v)$ if $v \in V(G')$.

Thus, $G = K_2^n + K_{r-1}$ is UkLC. It follows that $m(K_2^{n-1} + K_{r-1}) \leqslant m(K_2^n + K_{r-1})$.

Put $m(K_2^{n-1} + K_{r-1}) = t$. For suppose on the contrary that graph $G = K_2^n + K_{r-1}$ satisfies $m(G) = h > t + 2$. So there exists a list of $h - 1$ colors $L(v)$ for each vertex $v \in V(G)$, such that there exists a unique L-coloring f for G. We consider separately two cases.

C a s e 1: $|f(V_n)| = 1$.

In this case, $f(u_{n1}) = f(u_{n2}) = a$. We assign the following lists $L'(v)$ for the vertices v of G':

(a) If $a \in L(v)$, then $L'(v) = L(v) \setminus \{a\}$;

(b) If $a \notin L(v)$, then $L'(v) = L(v) \setminus \{b\}$, where $b \in L(v)$ and $b \neq f(v)$.

It is clear that $|L'(v)| = h - 2 \geqslant t + 1$ for every $v \in V(G')$. Since G' has the property $M(t)$, by Lemma 1, G' has the property $M(t+1)$, so G' has the property $M(h-2)$. It follows that with lists $L'(v)$, there exist at least two list colorings for the vertices v of G'. So it is not difficult to see that with lists $L(v)$, there exist at least two list colorings for the vertices v of G, a contradiction.

C a s e 2: $|f(V_n)| = 2$.

In this case, $f(u_{11}) = a, f(u_{12}) = b, a \neq b$. We assign the following lists $L'(v)$ for the vertices v of G':

(a) If $a, b \in L(v)$, then $L'(v) = L(v) \setminus \{a, b\}$;

(b) If $a \in L(v), b \notin L(v)$, then $L'(v) = L(v) \setminus \{a, c\}$, where $c \in L(v)$ and $c \neq f(v)$;

(c) If $a \notin L(v), b \in L(v)$, then $L'(v) = L(v) \setminus \{b, c\}$, where $c \in L(v)$ and $c \neq f(v)$;

(d) If $a, b \notin L(v)$, then $L'(v) = L(v) \setminus \{c, d\}$, where $c, d \in L(v), c \neq d$ and $c, d \neq f(v)$.

It is clear that $|L'(v)| = h - 3 \geqslant t$ for every $v \in V(G')$. Since G' has the property $M(t)$, by Lemma 1, G' has the property $M(h-3)$. It follows that with lists $L'(v)$, there exist at least two list colorings for the vertices v of G'. So it is not difficult to see that with lists $L(v)$, there exist at least two list colorings for the vertices v of G, a contradiction.

Thus, $m(K_2^{n-1} + K_{r-1}) \leqslant m(K_2^n + K_{r-1}) \leqslant m(K_2^{n-1} + K_{r-1}) + 2$.

(ii) We prove $G = K_2^{n-1} + K_r$ is UkLC by induction on r. For $r = 2$, it is not difficult we deduce what to prove. So let $r > 2$ and assume the assertion for smaller values of r. Let $V(G) = V_1 \cup V_2 \cup V_3 \cup \ldots \cup V_{n+r-1}$ is a partition of $V(G)$ such that $|V_1| = |V_2| = \ldots = |V_{n-1}| = 2$, $|V_n| = |V_{n+1}| = \ldots = |V_{n+r-1}| = 1$ and for every $i = 1, 2, \ldots, n-1$ the subgraph of G induced by V_i is an independent set. Set $V_i = \{v_i\}$ for every $i = n, n+1, \ldots, n+r-1$ and $G' = G - V_n$. By the induction hypothesis, for each vertex v in G', there exists a list of k colors $L'(v)$, such that there exists a unique f' for G'.

We assign the following lists for the vertices of G:

$$L(v_n) = \{t_1, t_2, \ldots, t_{k-1}, t_k\}$$

with $t_1, t_2, \ldots, t_{k-1} \in f'(G')$, $t_k \notin f'(G')$, $L(v) = L'(v)$ if $v \in V(G')$. A unique coloring f of G exists from the assigned lists: $f(v_n) = t_k$, $f(v) = f'(v)$ if $v \in V(G')$.

Thus, $K_2^{n-1} + K_r$ is UkLC. It follows that $m(K_2^{n-1} + K_{r-1}) \leqslant m(K_2^{n-1} + K_r)$.

Put $m(K_2^{n-1} + K_{r-1}) = t$. For suppose on the contrary that graph $G = K_2^{n-1} + K_r$ satisfies $m(G) = h > t + 1$. So there exists a list of $h - 1$ colors $L(v)$ for each vertex $v \in V(G)$, such that there exists a unique L-coloring f for G. Let $f(v_n) = a$. We assign the following lists $L'(v)$ for the vertices v of G':

(a) If $a \in L(v)$, then $L'(v) = L(v) \setminus \{a\}$;

(b) If $a \notin L(v)$, then $L'(v) = L(v) \setminus \{b\}$, where $b \in L(v)$ and $b \neq f(v)$.

It is clear that $|L'(v)| = h - 2 \geqslant t$ for every $v \in V(G')$. Since G' has the property $M(t)$, so G' has the property $M(h-2)$. It follows that with lists $L'(v)$, there exist at least two list colorings for the vertices v of G'. So it is not difficult to see that with lists $L(v)$, there exist at least two list colorings for the vertices v of G, a contradiction.

Thus, $m(K_2^{n-1} + K_{r-1}) \leqslant m(K_2^{n-1} + K_r) \leqslant m(K_2^{n-1} + K_{r-1}) + 1$. ■

Lemma 11 [8]. *$G = K_2^3 + K_r$ is U3LC for every $r \geqslant 1$.*

Theorem 2.

(i) $m(K_2^3 + K_r) = 5$ if and only if $r \geqslant 4$;

(ii) $m(K_2^3 + K_r) = 4$ if and only if $1 \leqslant r \leqslant 3$.

Proof.

(i) Suppose that $m(K_2^3 + K_r) = 5$. If $1 \leqslant r \leqslant 3$, then by Lemma 11 and Lemma 5, $m(K_2^3 + K_r) = 4$, a contradiction.

Now we suppose that $r \geqslant 4$. First, we prove $G = K_2^3 + K_r$ is U4LC. Let $V(G) = V_1 \cup V_2 \cup V_3 \cup \ldots \cup V_{3+r}$ is a partition of $V(G)$ such that $|V_1| = |V_2| = |V_3| = 2$, $|V_4| = |V_5| = \ldots = |V_{3+r}| = 1$ and for every $i = 1, 2, 3$ the subgraph of G induced by V_i is an independent set. Set $V_i = \{u_{i1}, u_{i2}\}$ for every $i = 1, 2, 3$ and $V_{3+i} = \{v_i\}$ for every $i = 1, 2, \ldots, r$. We assign the following lists for the vertices of $G = K_2^3 + K_r$:

$L(u_{i1}) = L(v_1) = \{1, 2, 3, 4\}$ for every $i = 1, 2, 3$;

$L(u_{i2}) = \{5, 6, 7, i + 1\}$ for every $i = 1, 2, 3$;

$L(v_j) = \{2, 3, 4, 3 + j\}$ for every $j = 2, 3, \ldots, r$.

A unique coloring f of G exists from the assigned lists:

$f(u_{i1}) = i + 1$ for every $i = 1, 2, 3$;

$f(u_{i2}) = i + 1$ for every $i = 1, 2, 3$;

$f(v_1) = 1$, $f(v_j) = 3 + j$ for every $j = 2, 3, \ldots, r$.

It follows that G is U4LC. So $m(G) \geqslant 5$. By Lemma 6, $m(G) \leqslant 5$. Thus, $m(G) = 5$.

(ii) Suppose that $m(K_2^3 + K_r) = 4$. If $r \geqslant 4$, then by (i), $m(K_2^3 + K_r) = 5$, a contradiction.

Suppose that $1 \leqslant r \leqslant 3$. By Lemma 11 and Lemma 5, $m(K_2^3 + K_r) = 4$. ■

Theorem 3. Let $G = K_2^n + K_r$ be a graph with $n \geqslant 4$ and $r \geqslant 1$. Then

(i) G is UkLC with $k = \lfloor n/2 \rfloor + 1$;

(ii) If $1 \leqslant r < n - 2$, then G has the property $M(n)$;

(iii) If $r \geqslant n - 1$, then G is UnLC;

(iv) If $n - 1 \leqslant r \leqslant n$, then $m(G) = n + 1$;

(v) If $r \geqslant n + 1$, then $m(G) = n + 2$.

Proof. Let $V(G) = V_1 \cup V_2 \cup V_3 \cup \ldots \cup V_{n+r}$ is a partition of $V(G)$ such that $|V_1| = |V_2| = \ldots = |V_n| = 2$, $|V_{n+1}| = |V_{n+2}| = \ldots = |V_{n+r}| = 1$ and for every $i = 1, 2, \ldots, n$ the subgraph of G induced by V_i is an independent set. Set $V_i = \{u_{i1}, u_{i2}\}$ for every $i = 1, \ldots, n$ and $V_{n+i} = \{v_i\}$ for every $i = 1, 2, \ldots, r$.

(i) Put $t = \lfloor n/2 \rfloor$. We assign the following lists for the vertices of G:

$L(u_{i1}) = \{1, 2, \ldots, t+1\}$ for every $i = 1, 2, \ldots, t+1$;

$L(u_{i2}) = \{t+2, t+3, \ldots, 2t+1, i\}$ for every $i = 1, 2, \ldots, t+1$;

$L(u_{(t+1+i)j}) = \{2, 3, \ldots, t+1, t+1+i\}$ for every $i = 1, 2, \ldots, n-t-1$, $j = 1, 2$;

$L(v_i) = \{2, 3, \ldots, t+1, n+i\}$ for every $i = 1, 2, \ldots, r$.

A unique coloring f of G exists from the assigned lists:

$f(u_{i1}) = i$ for every $i = 1, 2, \ldots, t+1$;

$f(u_{i2}) = i$ for every $i = 1, 2, \ldots, t+1$;

$f(u_{(t+1+i)j}) = t+1+i$ for every $i = 1, 2, \ldots, n-t-1$, $j = 1, 2$;

$f(v_i) = n+i$ for every $i = 1, 2, \ldots, r$.

(ii) If $G = K_2^n + K_r$ is UnLC, then by Lemma 5, $|V(G)| \geqslant 3n-2$, a contradiction.

(iii) We assign the following lists for the vertices of G:

$L(u_{i1}) = \{1, 2, \ldots, n\}$ for every $i = 1, 2, \ldots, n$;

$L(u_{i2}) = \{n+1, n+2, \ldots, 2n-1, i\}$ for every $i = 1, 2, \ldots, n$;

$L(v_j) = \{1, 2, \ldots, n-1, n+j\}$ for every $j = 1, 2, \ldots, r$.

A unique coloring f of G exists from the assigned lists:

$f(u_{i1}) = i$ for every $i = 1, 2, \ldots, n$;

$f(u_{i2}) = i$ for every $i = 1, 2, \ldots, n$;

$f(v_j) = n+j$ for every $j = 1, 2, \ldots, r$.

Thus, G is UnLC.

(iv) By (iii), G is UnLC. If G is U$(n+1)$LC, then by Lemma 5, $|V(G)| \geqslant 3n+1$, a contradiction. So $m(G) = n+1$.

(v) We assign the following lists for the vertices of G:

$L(u_{i1}) = L(v_1) = \{1, 2, \ldots, n+1\}$ for every $i = 1, 2, \ldots, n$;

$L(u_{i2}) = \{n+2, n+3, \ldots, 2n+1, i+1\}$ for every $i = 1, 2, \ldots, n$;

$L(v_j) = \{2, 3, \ldots, n+1, n+j\}$ for every $j = 2, 3, \ldots, r$.

A unique coloring f of G exists from the assigned lists:

$f(u_{i1}) = i+1$ for every $i = 1, 2, \ldots, n$;

$f(u_{i2}) = i+1$ for every $i = 1, 2, \ldots, n$;

$f(v_1) = 1$, $f(v_j) = n+j$ for every $j = 2, 3, \ldots, r$.

It follows that G is U$(n+1)$LC. So $m(G) \geqslant n+2$.

By Lemma 6, $m(G) \leqslant n+2$. Thus, $m(G) = n+2$. ■

REFERENCES

1. *Behzad M. and Chartrand G.* Introduction to the Theory of Graphs. Boston, Allyn and Bacon, 1971.
2. *Vizing V. G.* Raskraska vershin grafa v zadannye tsveta [Coloring the vertices of a graph in prescribed colors]. Diskret. Analiz, 1976, no. 29, pp. 3–10. (in Russian)
3. *Erdös P., Rubin A. L., and Taylor H.* Choosability in graphs. Proc. West Coast Conf. Combinatorics, Graph Theory, Computing, Arcata, CA, September 1979, Congr. Numer., no. 26, pp.125–157.
4. *Dinitz J. H. and Martin W. J.* The stipulation polynomial of a uniquely list colorable graph. Australasian J. Combin., 1995, no, 11, pp. 105–115.

5. *Mahdian M. and Mahmoodian E. S.* A characterization of uniquely 2-list colorable graphs. Ars Combin., 1999, vol. 51, pp. 295–305.
6. *Wang W. and Liu X.* List-coloring based channel allocation for open-spectrum wireless networks. Proc. VTC '05, 2005, pp. 690–694.
7. *Hung L. X.* List-chromatic number and chromatically unique of the graph $K_2^r + O_k$. Selecciones Matemáticas, Universidad Nacional de Trujillo, 2019, vol. 06(01), pp. 26–30.
8. *Hung L. X.* Colorings of the graph $K_2^m + K_n$. J. Sib. Fed. Univ. Math. Phys., 2020, vol. 13, iss. 3, pp. 297–305.
9. *Ghebleh M. and Mahmoodian E. S.* On uniquely list colorable graphs. Ars Combin., 2001, vol. 59, pp. 307–318.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

УДК 510.52 DOI 10.17223/20710410/55/7

О ГЕНЕРИЧЕСКОЙ СЛОЖНОСТИ ПРОБЛЕМЫ ВХОЖДЕНИЯ ДЛЯ ПОЛУГРУПП ЦЕЛОЧИСЛЕННЫХ МАТРИЦ[1]

А. Н. Рыбалов

Институт математики им. С. Л. Соболева СО РАН, г. Новосибирск, Россия

E-mail: alexander.rybalov@gmail.com

Проблема вхождения в конечно порождённую подгруппу (подполугруппу) для групп (полугрупп) является классической алгоритмической проблемой в алгебре, активно изучаемой многие десятилетия. Уже для достаточно простых групп и полугрупп эта проблема становится неразрешимой. Например, К. А. Михайлова в 1966 г. доказала неразрешимость проблемы вхождения в конечно порождённые подгруппы и, следовательно, подполугруппы для прямого произведения $F_2 \times F_2$ двух свободных групп ранга 2. Так как по известной теореме Санова группа F_2 имеет точное представление целочисленными матрицами порядка 2, группа $F_2 \times F_2$ является подгруппой группы $\mathrm{GL}_4(\mathbb{Z})$ целочисленных матриц порядка 4. Отсюда легко следует неразрешимость рассматриваемой проблемы для группы $\mathrm{GL}_k(\mathbb{Z})$ при $k \geqslant 4$. Неразрешимость проблемы вхождения в подполугруппы полугрупп целочисленных матриц порядка $\geqslant 3$ следует из результата М. Патерсона 1970 г. В данной работе предлагается сильно генерический алгоритм, решающий проблему вхождения в подполугруппы полугрупп целочисленных матриц произвольного порядка для подмножества входов, последовательность относительных плотностей которого при увеличении размера экспоненциально быстро сходится к 1.

Ключевые слова: *генерическая сложность, проблема вхождения, полугруппа целочисленных матриц.*

GENERIC COMPLEXITY OF THE MEMBERSHIP PROBLEM FOR SEMIGROUPS OF INTEGER MATRICES

A. N. Rybalov

Sobolev Institute of Mathematics, Novosibirsk, Russia

The membership problem for finitely generated subgroups (subsemigroups) of groups (semigroups) is a classical algorithmic problem, actively studied for many decades. Already for sufficiently simple groups and semigroups, this problem becomes undecidable. For example, K. A. Mikhailova in 1966 proved the undecidability of the membership problem for finitely generated subgroups (hence and for subsemigroups) of a direct product $F_2 \times F_2$ of two free groups of rank 2. Since, by the well-known Sanov theorem, the group F_2 has an exact representation by integer matrices of order 2,

[1]Работа выполнена в рамках государственного задания ИМ СО РАН, проект FWNF-2022-0003.

the group $F_2 \times F_2$ is a subgroup of the group $\mathrm{GL}_4(\mathbb{Z})$ of integer matrices of order 4. It easily implies the undecidability of this problem for the group $\mathrm{GL}_k(\mathbb{Z})$ for $k \geqslant 4$. Undecidability of the membership problem for finitely generated subsemigroups of semigroups of integer matrices of order $\geqslant 3$ follows from Paterson's result proved in 1970. In this paper, we propose a strongly generic algorithm deciding the membership problem for semigroups of integer matrices of arbitrary order for inputs from a subset whose sequence of frequencies exponentially fast converges to 1 with increasing size.

Keywords: *generic complexity, membership problem, semigroups of integer matrices.*

Введение

Проблема вхождения в конечно порождённую подгруппу (подполугруппу) для групп (полугрупп) является классической алгоритмической проблемой в алгебре, особенно активно изучаемой с момента появления формализации понятия алгоритма. Как оказалось, уже для достаточно простых групп и полугрупп эта проблема становится неразрешимой. Например, К. А. Михайлова [1] доказала неразрешимость проблемы вхождения в конечно порождённые подгруппы и, следовательно, подполугруппы для прямого произведения $F_2 \times F_2$ двух свободных групп ранга 2. Так как по известной теореме Санова группа F_2 имеет точное представление целочисленными матрицами порядка 2, группа $F_2 \times F_2$ является подгруппой группы $\mathrm{GL}_4(\mathbb{Z})$ целочисленных матриц порядка 4. Отсюда легко следует неразрешимость рассматриваемой проблемы для группы $\mathrm{GL}_k(\mathbb{Z})$ при $k \geqslant 4$. Неразрешимость проблемы вхождения в подполугруппы полугрупп целочисленных матриц порядка $\geqslant 3$ следует из результата М. Патерсона [2] о неразрешимости более узкой проблемы проверки принадлежности нулевой матрицы заданной подполугруппе. В литературе эта проблема называется проблемой умирающих матриц [3]. Отметим, что для полугруппы целочисленных матриц порядка 2 разрешимость проблемы умирающих матриц и проблемы вхождения в подполугруппы до сих пор не установлена [3]. Также открыт вопрос о разрешимости проблемы вхождения в конечно порождённые подгруппы группы целочисленных матриц $\mathrm{GL}_3(\mathbb{Z})$.

Генерический подход [4] — это один из подходов к изучению алгоритмических проблем для «почти всех» входов. Исследования вычислительной сложности для «почти всех» входов началось в 1970–1980-х гг., после того как был выделен огромный пласт трудноразрешимых алгоритмических проблем — NP-полных проблем, для которых не удалось найти эффективных алгоритмов, работающих за полиномиальное время для всех входов. Оказалось, что если немного ослабить требование эффективности — рассматривать не все входы, а «почти все» или случайные входы, то иногда можно быстро решать задачу для таких типичных входов. Этот подход имеет практический смысл, когда алгоритм должен решать быстро задачу для случайных входных данных: если вероятность «наткнуться» на «плохой» вход пренебрежимо мала, то алгоритм будет быстро работать практически всегда. Ярким примером такого алгоритма является симплекс-метод: этот алгоритм имеет экспоненциальную сложность в худшем случае, но за полиномиальное время решает задачу линейного программирования для почти всех входных данных. Ещё можно упомянуть алгоритм Бабаи, Эрдеша и Селкова [5], решающий за полиномиальное время знаменитую проблему изоморфизма графов для почти всех пар конечных графов. Отметим также алгоритмы Гимади, Глебова, Перепелицы [6] и Селиверстова [7] для некоторых проблем дискретной оптимизации. В теории сложности вычислений поведение алгоритмов на множестве «почти всех» входов тра-

диционно изучается в рамках подхода к сложности в среднем [8, 9], при этом время работы алгоритма усредняется по всему множеству входных данных. В отличие от сложности в среднем, генерический подход является более универсальным, так как может оказаться, что на множестве «плохих» входов даже усреднённое время работы алгоритма неполиномиально. Генерический же алгоритм просто игнорирует эти входы. Более того, генерический подход применим и к алгоритмически неразрешимым проблемам. Таким образом, может оказаться, что проблема трудноразрешима или вообще неразрешима в классическом смысле, но легкоразрешима в генерическом смысле.

В работе [10] доказано, что проблема вхождения в конечно порождённые подполугруппы полугрупп целочисленных матриц произвольного порядка генерически разрешима. В данной работе этот результат усиливается: предлагается сильно генерический алгоритм, решающий проблему для входов из подмножества, последовательность относительных плотностей которого при увеличении размера экспоненциально быстро сходится к 1.

1. Генерические алгоритмы

Пусть I — некоторое множество входов. Для подмножества $S \subseteq I$ определим последовательность

$$\rho_n(S) = \frac{|S_n|}{|I_n|},\ n = 1, 2, 3, \ldots,$$

где I_n — множество входов размера n; $S_n = S \cap I_n$ — множество входов из S размера n. Заметим, что $\rho_n(S)$ — это вероятность попасть в S при случайной и равновероятной генерации входов из I_n. *Асимптотической плотностью* S назовём предел (если он существует)

$$\rho(S) = \lim_{n\to\infty} \rho_n(S).$$

Множество S называется *генерическим*, если $\rho(S) = 1$, и *пренебрежимым*, если $\rho(S) = 0$. Очевидно, что S генерическое тогда и только тогда, когда его дополнение $I \setminus S$ пренебрежимо. Назовём множество S *сильно пренебрежимым*, если последовательность $\rho_n(S)$ экспоненциально быстро сходится к 0, т. е. существуют константы σ, $0 < \sigma < 1$, и $C > 0$, такие, что для любого n

$$\rho_n(S) < C\sigma^n.$$

Множество S называется *сильно генерическим*, если его дополнение $\overline{S}$ сильно пренебрежимо.

Алгоритм $\mathcal{A}$ с множеством входов I называется (*сильно*) *генерическим*, если множество $\{x \in I : \mathcal{A}(x)\downarrow\}$ (сильно) генерическое. Здесь через $\mathcal{A}(x)\downarrow$ обозначается тот факт, что алгоритм $\mathcal{A}$ останавливается на входе x. (Сильно) генерический алгоритм $\mathcal{A}$ вычисляет функцию $f : I \to J$, если

$$\forall x \in I\ \mathcal{A}(x)\downarrow \Rightarrow\ f(x) = \mathcal{A}(x).$$

(Сильно) генерический алгоритм $\mathcal{A}$ работает за полиномиальное время, если существует полином $p(n)$, такой, что

$$\forall x \in I\ \mathcal{A}(x)\downarrow \Rightarrow t_{\mathcal{A}}(x) < p(\mathrm{size}(x)).$$

Такие алгоритмы будем называть полиномиальными генерическими.

С практической точки зрения, когда требуется построить алгоритм, решающий конкретную алгоритмическую проблему для почти всех входов, удобнее рассматривать алгоритмы следующего типа: алгоритм останавливается на всех входах, на входах из некоторого генерического множества выдаёт правильный ответ, а на пренебрежимом множестве остальных входов выдает специальный ответ «?» — «Не знаю». Определение такой эффективной генерической вычислимости можно найти в обзоре [11] и в гораздо более ранней работе [12].

Алгоритм $\mathcal{A}$ с множеством входов I и множеством выходов $J \cup \{?\}$ $(? \notin J)$ называется *эффективно* (*сильно*) *генерическим*, если

1) $\mathcal{A}$ останавливается на всех входах из I;
2) множество $\{x \in I : \mathcal{A}(x) = ?\}$ (сильно) пренебрежимо.

Эффективно (сильно) генерический алгоритм $\mathcal{A}$ вычисляет функцию $f : I \to J$, если

$$\forall x \in I \ \mathcal{A}(x) \neq ? \ \Rightarrow \ f(x) = \mathcal{A}(x).$$

Множество $S \subseteq I$ и соответствующая проблема распознавания (S, I) (*эффективно*) (*сильно*) *генерически разрешимы*, если существует (эффективно) (сильно) генерический алгоритм, вычисляющий характеристическую функцию S.

Легко видеть, что из эффективной генерической разрешимости следует генерическая разрешимость. Действительно, любой эффективно генерический алгоритм можно переделать в генерический, заменив выдачу ответа «?» на бесконечное зацикливание. В обратную сторону это неверно — см., например, теорему 2.22 и следствие 2.24 в [13]. Однако для полиномиальной (экспоненциальной) сложности верно и обратное: из полиномиальной (экспоненциальной) генерической разрешимости следует полиномиальная (экспоненциальная) эффективная генерическая разрешимость. Действительно, если имеется полиномиальная оценка $p(n)$ на время работы генерического алгоритма, когда он останавливается, то можно завести счётчик T числа шагов и в случае, если $T > p(n)$, можно обрывать вычисление и выдавать ответ «?» — алгоритм уже не остановится. Таким образом получается эффективно генерический полиномиальный алгоритм, решающий ту же проблему. То же верно и для сильно генерических алгоритмов.

С учётом сказанного в дальнейшем при доказательстве существования (сильно) генерического алгоритма будем строить эффективно (сильно) генерические алгоритмы. Из существования эффективного (сильно) генерического алгоритма будет следовать существование (сильно) генерического алгоритма.

2. Вспомогательный результат

Пусть $\mathrm{M}_k(\mathbb{Z})$ — полугруппа целочисленных матриц порядка k по стандартному умножению матриц. Элементы $\mathrm{M}_k(\mathbb{Z})$ будем представлять матрицами из целых чисел. Размер целого числа a, обозначаемый $\mathrm{size}(a)$, — это длина двоичной записи его модуля. Таким образом, $\mathrm{size}(a) = n$, если $2^{n-1} \leqslant |a| < 2^n$. Отдельно положим $\mathrm{size}(0) = 0$. Под размером матрицы $M = ||a_{ij}||$ будем понимать

$$\mathrm{size}(M) = \max\{\mathrm{size}(a_{ij}) : i, j = 1, \ldots, k\}.$$

Таким образом, если матрица M имеет размер n, то каждый её элемент лежит в отрезке $[-2^n + 1, 2^n - 1]$ и для него возможны $2^{n+1} - 1$ вариантов выбора.

Обозначим через $\mathrm{M}_k^r(\mathbb{Z})$ подмножество матриц в $\mathrm{M}_k(\mathbb{Z})$ с определителем, равным r.

Лемма 1. Пусть r равно 0, 1 или -1. Тогда для любого достаточно большого n имеет место оценка

$$\frac{|\mathrm{M}_k^r(\mathbb{Z})_{\leqslant n}|}{|\mathrm{M}_k(\mathbb{Z})_{\leqslant n}|} \leqslant \frac{1}{2^n}.$$

Доказательство. Напомним лемму Шварца — Зиппеля [14–16]. Она утверждает, что если $P(x_1, x_2, \ldots, x_n)$ — ненулевой многочлен степени d над полем $\mathbb{R}$, S — конечное подмножество $\mathbb{R}$ и элементы $r_1, r_2, \ldots, r_n$ выбраны из S равномерно и независимо друг от друга, то

$$\mathsf{P}[P(r_1, r_2, \ldots, r_n) = 0] \leqslant \frac{d}{|S|}.$$

Заметим, что если положить

$$P(x_1, \ldots, x_{n^2}) = \det \begin{pmatrix} x_1 & x_2 & \ldots & x_n \\ x_{n+1} & x_{n+2} & \ldots & x_{2n} \\ \ldots & & & \\ x_{n^2-n+1} & x_{n^2-n+2} & \ldots & x_{n^2} \end{pmatrix} - r,$$

то будет иметь место

$$\frac{|\mathrm{M}_k^r(\mathbb{Z})_{\leqslant n}|}{|\mathrm{M}_k(\mathbb{Z})_{\leqslant n}|} = \mathsf{P}[P(r_1, r_2, \ldots, r_{n^2}) = 0],$$

где элементы $r_1, \ldots, r_{n^2}$ равновероятно и независимо друг от друга выбраны из множества $S = \{-2^n + 1, \ldots, 2^n - 1\}$. Учитывая, что степень многочлена $P(x_1, \ldots, x_{n^2})$ по любой переменной равна 1, по лемме Шварца — Зиппеля получаем

$$\frac{|\mathrm{M}_k^r(\mathbb{Z})_{\leqslant n}|}{|\mathrm{M}_k(\mathbb{Z})_{\leqslant n}|} \leqslant \frac{1}{|S|} = \frac{1}{2^{n+1} - 1} \leqslant \frac{1}{2^n}.$$

Лемма 1 доказана. ■

3. Основной результат

Проблема вхождения для полугруппы $\mathrm{M}_k(\mathbb{Z})$ формулируется следующим образом. По произвольным заданным матрицам $M_1, \ldots, M_n, M$ из $\mathrm{M}_k(\mathbb{Z})$, таким, что $\mathrm{size}(M_1), \ldots, \mathrm{size}(M_n), \mathrm{size}(M) \leqslant n$, определить, принадлежит ли матрица M подполугруппе, порождённой матрицами $M_1, \ldots, M_n$. Другими словами, представима ли матрица M в виде некоторого произведения матриц из набора $M_1, \ldots, M_n$, возможно, с повторами? Размером входа здесь является число n.

Теорема 1. Проблема вхождения для $\mathrm{M}_k(\mathbb{Z})$ сильно генерически разрешима.

Доказательство. Эффективный сильно генерический алгоритм для проблемы вхождения работает на входе $(M_1, \ldots, M_n, M)$ размера n следующим образом:

1. Вычисляет определители $\det(M_1), \ldots, \det(M_n), \det(M)$.
2. Если среди $\det(M_1), \ldots, \det(M_n), \det(M)$ найдётся определитель, равный 0, 1 или -1, выдаёт ответ «?».
3. Иначе перебирает всевозможные наборы матриц $M_{i_1}, \ldots, M_{i_k}$, такие, что $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ и $\det(M_{i_1}) \ldots \det(M_{i_k}) = \det(M)$, и проверяет, выполнено ли равенство $M_{i_1} \ldots M_{i_k} = M$. Если равенство получилось, выдаёт ответ «ДА», иначе — ответ «НЕТ».

Заметим, что этот алгоритм всегда останавливается, так как число матриц в тестируемых наборах ограничено числом целых делителей $\det(M)$, отличных от 1 и -1.

Для доказательства эффективной сильной генеричности алгоритма нужно показать, что множество входов S, на котором алгоритм выдаёт ответ, отличный от «?», является сильно генерическим. Заметим, что

$$S_n = \{(M_1, \ldots, M_n, M) : \operatorname{size}(M_i) \leqslant n, \operatorname{size}(M) \leqslant n, \det(M_i) \neq 0, 1, -1,\ i = 1, \ldots, n\} = \\ = \Big(\mathrm{M}_k(\mathbb{Z})_{\leqslant n} \setminus \big(\mathrm{M}_k^0(\mathbb{Z})_{\leqslant n} \cup \mathrm{M}_k^1(\mathbb{Z})_{\leqslant n} \cup \mathrm{M}_k^{-1}(\mathbb{Z})_{\leqslant n}\big)\Big)^n \times \mathrm{M}_k(\mathbb{Z})_{\leqslant n}.$$

Пусть I — множество всех входов. Тогда $I_n = (\mathrm{M}_k(\mathbb{Z})_{\leqslant n})^{n+1}$. Поэтому

$$\rho_n(S) = \frac{|S_n|}{|I_n|} = \Big(\frac{|\mathrm{M}_k(\mathbb{Z})_{\leqslant n} \setminus \big(\mathrm{M}_k^0(\mathbb{Z})_{\leqslant n} \cup \mathrm{M}_k^1(\mathbb{Z})_{\leqslant n} \cup \mathrm{M}_k^{-1}(\mathbb{Z})_{\leqslant n}|}{|\mathrm{M}_k(\mathbb{Z})_{\leqslant n}|}\Big)^n = \\ = \Big(1 - \frac{|\mathrm{M}_k^0(\mathbb{Z})_{\leqslant n}|}{|\mathrm{M}_k(\mathbb{Z})_{\leqslant n}|} - \frac{|\mathrm{M}_k^1(\mathbb{Z})_{\leqslant n}|}{|\mathrm{M}_k(\mathbb{Z})_{\leqslant n}|} - \frac{|\mathrm{M}_k^{-1}(\mathbb{Z})_{\leqslant n}|}{|\mathrm{M}_k(\mathbb{Z})_{\leqslant n}|}\Big)^n.$$

По лемме 1 можно оценить

$$\rho_n(S) \geqslant \Big(1 - \frac{3}{2^n}\Big)^n = \left(\Big(1 - \frac{3}{2^n}\Big)^{2^n}\right)^{n/2^n} > e^{-3n/2^n} > 1 - \frac{3n}{2^n}$$

для достаточно больших n. Последнее выражение экспоненциально быстро стремится к 1 при стремлении n к бесконечности, следовательно, множество S сильно генерическое. ■

Автор выражает благодарность рецензенту за полезные замечания и предложения по улучшению текста статьи.

ЛИТЕРАТУРА

1. *Михайлова К. А.* Проблема вхождения для прямых произведений групп // Математический сборник. 1966. Т. 112. № 2. С. 241–251.
2. *Paterson M. S.* Unsolvability in 3×3 matrices // Studies Appl. Math. 1970. V. 49. No. 1. P. 105–107.
3. *Halava V. and Harju T.* Mortality in matrix semigroups // Amer. Math. Monthly. 2001. V. 108. No. 7. P. 649–653.
4. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks // J. Algebra. 2003. V. 264. No. 2. P. 665–694.
5. *Babai L., Erdos P, and Selkow S.* Random graph isomorphism // SIAM J. Computing. 1980. V. 9. No. 3. P. 628–635.
6. *Гимади Э. Х., Глебов Н. И., Перепелица В. А.* Алгоритмы с оценками для задач дискретной оптимизации // Проблемы кибернетики. 1975. Т. 31. С. 35–42.
7. *Селиверстов А.В.* Двоичные решения для больших систем линейных уравнений // Прикладная дискретная математика. 2021. № 52. С. 5–15.
8. *Gurevich Y.* Average case completeness // J. Computer System Sci. 1991. V. 42. P. 346–398.
9. *Levin L.* Average case complete problems // SIAM J. Computing. 1987. V. 15. P. 285–286.
10. *Рыбалов А.* Генерический алгоритм для проблемы вхождения в полугруппах целочисленных матриц // Вестник Омского университета. 2020. Т. 25. № 3. С. 8–12.
11. *Hirschfeldt D.* Some questions in computable mathematics // Computability and Complexity. 2017. P. 22–55.

12. *Meyer A.* An open problem on creative sets // Recursive Function Theory Newsletter. 1973. V. 4. P. 15–16.
13. *Jockusch C. and Schupp P.* Generic computability, Turing degrees, and asymptotic density // J. London Math. Soc. 2012. V. 85. No. 2. P. 472–490.
14. *Schwartz J.* Fast probabilistic algorithms for verification of polynomial identities // J. ACM. 1980. V. 27. No. 4. P. 701–717.
15. *Zippel R.* Probabilistic algorithms for sparse polynomials // Symbolic Algebraic Computation. 1979. V. 72. P. 216–226.
16. *DeMillo R. and Lipton R.* A probabilistic remark on algebraic program testing // Inform. Processing Lett. 1978. V. 7. P. 193–195.

REFERENCES

1. *Mikhaylova K. A.* Problema vkhozhdeniya dlya pryamykh proizvedeniy grupp [Membership problem for direct products of groups]. Matematicheskiy Sbornik, 1966, vol. 112, no. 2, pp. 241–251. (in Russian)
2. *Paterson M.S.* Unsolvability in 3×3 matrices. Studies Appl. Math., 1970, vol. 49, no. 1, pp. 105–107.
3. *Halava V. and Harju T.* Mortality in matrix semigroups. Amer. Math. Monthly, 2001, vol. 108, no. 7, pp. 649–653.
4. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks. J. Algebra, 2003, vol. 264, no. 2, pp. 665–694.
5. *Babai L., Erdos P, and Selkow S.* Random graph isomorphism. SIAM J. Computing, 1980, vol. 9, no. 3, pp. 628–635.
6. *Gimadi E. X., Glebov N. I., and Perepelitsa V. A.* Algoritmy s otsenkami dlya zadach diskretnoy optimizatsii [Algorithms with bounds for problems of discrete optimization]. Problemy Kibernetiki, 1975, vol. 31, pp. 35–42. (in Russian)
7. *Seliverstov A. V.* Dvoichnye resheniya dlya bol'shikh sistem lineynykh uravneniy [Binary solutions to large systems of linear equations]. Prikladnaya Diskretnaya Matematika, 2021, no. 52, pp. 5–15. (in Russian)
8. *Gurevich Y.* Average case completeness. J. Computer System Sci., 1991, vol. 42, pp. 346–398.
9. *Levin L.* Average case complete problems. SIAM J. Computing, 1987, vol. 15, pp. 285–286.
10. *Rybalov A.* Generichеskii algoritm dlya problemy vhozhdeniya v polugruppah celochislennyh matrits [A generic algorithm for the membership problem in semigroups of integer matrices] // Vestnik Omskogo universiteta. 2020. V. 25. № 3. P. 8–12.
11. *Hirschfeldt D.* Some questions in computable mathematics. Computability and Complexity, 2017, pp. 22–55.
12. *Meyer A.* An open problem on creative sets. Recursive Function Theory Newsletter, 1973, vol. 4, pp. 15–16.
13. *Jockusch C. and Schupp P.* Generic computability, Turing degrees, and asymptotic density. J. London Math. Soc., 2012, vol. 85, no. 2, pp. 472–490.
14. *Schwartz J.* Fast probabilistic algorithms for verification of polynomial identities. J. ACM, 1980, vol. 27, no. 4, pp. 701–717.
15. *Zippel R.* Probabilistic algorithms for sparse polynomials. Symbolic Algebraic Computation, 1979, vol. 72, pp. 216–226.
16. *DeMillo R. and Lipton R.* A probabilistic remark on algebraic program testing. Inform. Processing Lett., 1978, vol. 7, pp. 193–195.

УДК 004.4'413 DOI 10.17223/20710410/55/8

ПОСТРОЕНИЕ СИНТАКСИЧЕСКИХ АНАЛИЗАТОРОВ НА ОСНОВЕ СИНТАКСИЧЕСКИХ ДИАГРАММ С МНОГОВХОДОВЫМИ КОМПОНЕНТАМИ

Ю. Д. Рязанов, С. В. Назина

Белгородский государственный технологический университет им. В. Г. Шухова, г. Белгород, Россия

E-mail: razanov.yd@bstu.ru, lanalana9808@gmail.com

Рассматривается задача построения синтаксических анализаторов по синтаксическим диаграммам с многовходовыми компонентами (СД). Предлагается основанный на алгоритме GLL алгоритм построения синтаксического анализатора, результатом работы которого является компактное преставление леса разбора входной цепочки. Предложенный алгоритм позволяет строить синтаксические анализаторы по СД произвольной структуры и не требует предварительных преобразований СД. Построенные синтаксические анализаторы могут применяться для анализа любых контекстно-свободных языков, включая недетерминированные и неоднозначные. Вводятся понятия «дерево вывода» и «лес разбора» для СД, описываются структуры данных, используемые анализатором, такие, как стек с графовой структурой, дескриптор синтаксического анализатора, компактное представление леса разбора. Описывается алгоритм построения синтаксических анализаторов по СД и приводится пример построения такого анализатора.

Ключевые слова: *синтаксический анализ, синтаксические диаграммы с многовходовыми компонентами, лес разбора.*

BUILDING PARSERS BASED ON SYNTAX DIAGRAMS WITH MULTIPORT COMPONENTS

Yu. D. Ryazanov, S. V. Nazina

Belgorod State Technological University named after V. G. Shukhov, Belgorod, Russia

The problem of constructing parsers from syntax diagrams with multiport components (SD) is solved. An algorithm for constructing a parser based on the GLL algorithm is proposed, which results in the compact representation of the input chain parse forest. The proposed algorithm makes it possible to build parsers based on the SD of an arbitrary structure and does not require preliminary SD transformations. We introduce the concepts of "inference tree" and "parsing forest" for SD and describe the data structures used by the parser, such as a graph-structured stack, a parser descriptor, and a compact representation of the parsing forest. The algorithm for constructing parsers based on SD is described and an example of parser constructing is given.

Keywords: *parsing, syntax diagrams with multiport components, parse forest.*

Введение

Среди алгоритмов синтаксического анализа наиболее простым и наглядным является метод рекурсивного спуска [1], позволяющий строить анализаторы, структура которых соответствует структуре грамматики. Недостатком таких анализаторов является уязвимость к левой рекурсии, что приводит к необходимости предварительной обработки грамматики. Кроме того, использование возвратов в случае недетерминированного разбора может привести к экспоненциальному росту времени работы анализатора. Эти проблемы решает Generalised LL (GLL-анализ) [2, 3] — алгоритм, расширяющий метод рекурсивного спуска до построения синтаксических анализаторов без ограничений на класс контекстно-свободных грамматик. Результатом работы анализатора является лес разбора, который может представлять бесконечное число различных деревьев вывода заданной цепочки. Построенные анализаторы имеют в худшем случае кубическую сложность (временную и по памяти) и линейную сложность для LL-грамматик [3].

Описанный в [2, 3] алгоритм применим для построения синтаксических анализаторов по формальным грамматикам. Синтаксические диаграммы с многовходовыми компонентами (СД) [4] являются наглядным и более компактным представлением языка, чем формальные грамматики или диаграммы Вирта. GLL-анализ может быть расширен до построения синтаксических анализаторов по синтаксическим диаграммам с многовходовыми компонентами, что позволит строить компактные и эффективные по памяти и времени работы анализаторы, структура которых соответствует структуре исходной диаграммы.

В работе приводится основанный на GLL-анализе алгоритм построения синтаксических анализаторов по СД. Сначала поясняются понятия, связанные с синтаксическими диаграммами с многовходовыми компонентами, необходимые для дальнейшего изложения. Далее приводится описание структур данных, используемых анализатором в процессе разбора: леса разбора, стека с графовой структурой и дескриптора анализатора. После этого описывается построение синтаксических анализаторов по СД и приводится пример построения и работы такого анализатора.

1. Синтаксические диаграммы с многовходовыми компонентами

Синтаксическая диаграмма с многовходовыми компонентами представлена на рис. 1.

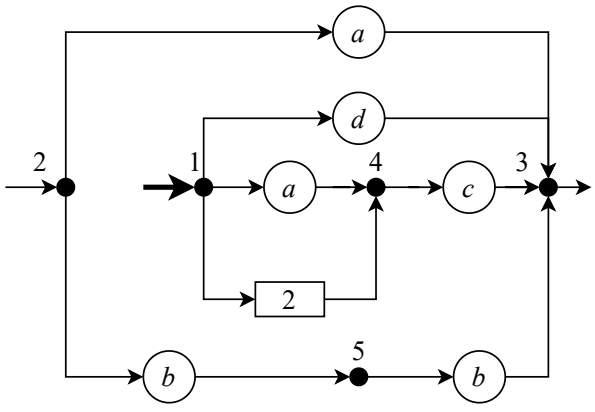


Рис. 1. Синтаксическая диаграмма с многовходовыми компонентами

Синтаксическая диаграмма с многовходовыми компонентами [4] задается восьмёркой $R = (T, u, u', u'', u_0, G, F_T, F_U)$, где

- T — конечное множество терминалов;
- u — конечное множество узлов;
- u' — конечное множество начальных узлов, $u' \subseteq u$;
- u'' — конечное множество заключительных узлов, $u'' \subseteq u$;
- u_0 – стартовый узел, $u_0 \in u'$;
- $G = (V, E)$ — ориентированный граф, $V = V_T \cup V_N \cup u$ — множество вершин, V_T — множество терминальных вершин, V_N — множество нетерминальных вершин, внутри которых записан начальный узел; $E = E_1 \cup E_2$, E_1 — множество дуг, выходящих из узлов и входящих в терминальные и нетерминальные вершины, E_2 — множество дуг, выходящих из терминальных и нетерминальных вершин и входящих в узлы;
- $F_T : V_T \to T$ — отображение множества терминальных вершин в множество терминалов;
- $F_U : V_N \to u'$ — отображение множества нетерминальных вершин в множество начальных узлов.

Терминальная вершина изображается на диаграмме кружком, в который вписан терминал. Нетерминальная вершина изображается прямоугольником, в который вписан начальный узел одной из компонент. Узлы пронумерованы и обозначаются жирными точками. Терминальные и нетерминальные вершины назовём символьными. Дуги соединяют между собой узел и символьную вершину. Начальные узлы обозначаются входящей стрелкой, заключительные — выходящей, стартовый — жирной входящей стрелкой. В символьную вершину может входить только одна дуга, и из символьной вершины может выходить только одна дуга. Ограничений на количество дуг, входящих и выходящих из узлов, нет.

С помощью СД можно получить любую цепочку языка, заданного СД. Процесс получения цепочки языка назовём выводом. Для определения вывода в работе [4] вводятся следующие понятия:

- цепочка, связывающая узел u с заключительными узлами — это цепочка, состоящая из терминалов и/или начальных узлов, которую можно получить, «двигаясь» в СД от узла u к заключительному узлу и выписывая из вершин по пути символы (терминалы или начальные узлы) в изначально пустую цепочку;
- $L(u)$ — множество всех цепочек, связывающих узел u с заключительными узлами.

Вывод цепочки языка, заданного СД со стартовым узлом u_0, заключается в следующем. Возьмём цепочку, принадлежащую $L(u_0)$. Если она содержит начальный узел u_i, заменим его цепочкой из множества $L(u_i)$. Если новая цепочка содержит начальный узел, то аналогичные действия повторяем. Вывод заканчивается, когда будет получена цепочка, не содержащая начальных узлов. Такая цепочка, дополненная концевым маркером, принадлежит языку, заданному СД.

Любая цепочка языка, заданного СД, имеет хотя бы один вывод. Если любая цепочка языка имеет только один вывод в заданной СД, то такую СД назовём однозначной; если хотя бы одна цепочка языка имеет более одного вывода, то — неоднозначной. Например, цепочка ac принадлежит языку, заданному СД (рис. 1), и имеет два вывода, которые можно представить следующим образом:

1) $1 \Rightarrow ac$;
2) $1 \Rightarrow 2c \Rightarrow ac$.

Следовательно, СД на рис. 1 является неоднозначной.

Вывод в СД можно представить и по-другому: как движение по дугам от стартового узла начальной компоненты к точке выхода. При этом если дуга идёт в терминальную вершину, то вписанный в неё терминальный символ добавляется в терминальную цепочку; если в нетерминальную, то переходим во вписанный в неё начальный узел и движемся далее аналогичным образом до точки выхода, после чего возвращаемся в предыдущую компоненту и продолжаем движение. После прохождения выходной дуги начальной компоненты в терминальную цепочку добавляем концевой маркер и вывод заканчивается.

Символ x, который может быть добавлен в терминальную цепочку непосредственно после прохождения дуги e, принадлежит множеству выбора дуги e ($x \in \text{ВЫБОР}(e)$). Алгоритм вычисления множества выбора дуги описан в [5–7]. Объединение множеств выбора всех выходящих из узла дуг образует множество выбора узла. Узел u называется детерминированным, если множества выбора любых двух дуг, выходящих из узла u, не пересекаются. Назовём вывод цепочки в СД детерминированным, если на каждом шаге вывода текущий символ цепочки принадлежит множеству выбора только одной выходящей из узла дуги. СД является детерминированной, если в ней все узлы детерминированные, иначе — недетерминированной [8, 9].

2. Дерево вывода

Любой вывод цепочки, принадлежащей языку, заданному СД, можно представить деревом вывода. Ориентированное упорядоченное дерево D назовём деревом вывода в синтаксической диаграмме с многовходовыми компонентами $R = (T, u, u', u'', u_0, G, F_T, F_U)$, если выполняются следующие условия:

- корень дерева отмечен узлом u_0;
- листья дерева отмечены терминалами из множества T или символом ϵ, обозначающим пустую цепочку;
- внутренние узлы дерева отмечены узлами из множества u';
- если узел w_0 дерева вывода отмечен начальным узлом z_0 и имеет сыновей $w_1, \ldots, w_n$, $n \geqslant 1$, отмеченных соответственно терминалами или начальными узлами $z_1, \ldots, z_n$, то в R существует путь из узла z_0 до заключительного узла компоненты, последовательно проходящий только через вершины, отмеченные $z_1, \ldots, z_n$;
- если узел w, отмеченный начальным узлом z_0, имеет единственного сына, отмеченного ϵ, то $z_0 \in u''$.

Два вывода цепочки ac в СД на рис. 1, приведённые в п. 1, представлены в виде двух деревьев на рис. 2.

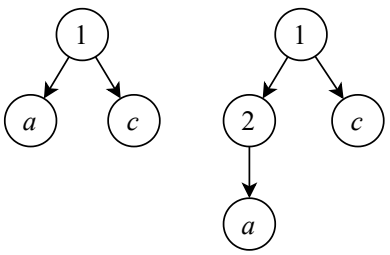


Рис. 2. Деревья вывода цепочки ac

3. Лес разбора

Множество всех деревьев вывода заданной цепочки в неоднозначной СД образует лес разбора. Если СД содержит путь ненулевой длины из узла u в узел u, при движении по которому выводится пустая терминальная цепочка, то возможно бесконечное множество различных выводов одной цепочки, проходящих через узел u. Лес разбора в этом случае содержит бесконечное множество деревьев. В этой работе будем предполагать, что любая выводимая в СД цепочка имеет конечное множество деревьев вывода.

Для компактного представления леса разбора (КПЛР) используется структура, основанная на SPPF (Shared Packed Parse Forest) [10], которая имеет три типа узлов: символьные, промежуточные и упаковывающие. Деревья на рис. 2 образуют КПЛР цепочки ac на рис. 3.

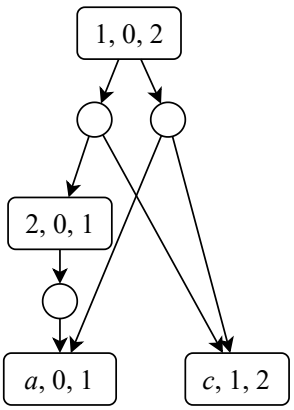


Рис. 3. КПЛР цепочки ac

Символьные узлы имеют вид (x, i, j), где x — терминал или начальный узел компоненты; i и j — левая и правая границы подцепочки, выводимой из x. Если x — терминал, то назовём такой узел терминальным, иначе — нетерминальным. Символьные узлы изображаются в виде прямоугольников с закругленными углами.

Промежуточные узлы имеют метку $(s \to v, i, j)$, где s — начальный узел компоненты; v — узел компоненты; i и j — левая и правая границы подцепочки, выводимой на промежутке движения от узла s к узлу v. Промежуточные узлы изображаются в виде прямоугольников.

Дочерними узлами символьных и промежуточных узлов могут быть только упаковывающие узлы, ограничение на количество дочерних узлов не накладывается. Упаковывающие узлы являются уникальными в множестве детей своего родителя и имеют не более двух дочерних узлов. Правый дочерний узел упаковывающего узла является символьным узлом, а левый может быть символьным или промежуточным. Метка упаковывающего узла позволяет идентифицировать конкретный вывод подцепочки. Необходимо определить такую метку упаковывающего узла, которая бы однозначно определяла ветвь вывода. Положим, левый дочерний узел имеет метку (t, i, k), а правый — (x, k, j). При этом если левый дочерний узел — символьный, то t — терминал или начальный узел, иначе t имеет вид $s \to w$. Тогда определим метку упаковывающего узла как (t, x, k). Если к родительскому узлу необходимо присоединить дочерние узлы через упаковывающий узел (t, x, k), то сначала проверяется, не существует ли узла (t, x, k) в множестве детей родительского узла. Если такой узел найден, то соответ-

ствующий ему вывод уже добавлен в лес разбора. Чтобы не загромождать рисунок, упаковывающие узлы будем изображать окружностью без меток.

Если лес разбора представляет собой конечное множество деревьев вывода, то КПЛР не содержит циклов.

Рассмотрим получение всех деревьев вывода по КПЛР, не содержащему циклов (алгоритм 1). Будем использовать следующие обозначения: $[a]$ — список, состоящий из элемента a; \$ — неопределённое значение переменной.

Алгоритм 1. Получение деревьев вывода по КПЛР

Вход: v — узел леса разбора.
Выход: деревья вывода.

ПРОЦЕДУРА $getTrees(v)$
$H \leftarrow \varnothing$.
Если v — терминальный узел с меткой (x, i, j), **то**
  $T \leftarrow$ создать дерево с корнем, отмеченным x; добавить $[T]$ в H;
иначе если v — нетерминальный узел с меткой (x, i, j), **то**
  Для всех дочерних узлов p узла v
    пусть z и r — левый и правый дочерние узлы p соответственно;
    $H_1 \leftarrow getTrees(z)$.
    Если $r \neq$ \$, **то**
      $H_2 \leftarrow getTrees(r)$.
      Для всех $(h_1, h_2) \in H_1 \times H_2$
        $T \leftarrow$ создать дерево с корнем w, отмеченным x;
        последовательно сделать все корневые узлы деревьев из списков h_1 и h_2 дочерними узлами w;
        добавить $[T]$ в H;
    иначе
      Для всех $h \in H_1$
        $T \leftarrow$ создать дерево с корнем w, отмеченным x;
        последовательно сделать все корневые узлы деревьев из списка h_1 дочерними узлами w;
        добавить $[T]$ в H;
иначе
  Для всех дочерних узлов p узла v
    пусть z и r — левый и правый дочерние узлы p соответственно;
    $H_1 \leftarrow getTrees(z)$.
    Если $r \neq$ \$, **то**
      $H_2 \leftarrow getTrees(r)$.
      Для всех $(h_1, h_2) \in H_1 \times H_2$
        $L \leftarrow$ конкатенация списков h_1 и h_2;
        добавить L в H;
    иначе
      добавить все элементы H_1 в H.
Вернуть H.

Будем обрабатывать узлы леса разбора, начиная с корневого узла, отмеченного $(s, 0, m)$, где s — стартовый узел СД; m — длина входной цепочки. В результате при-

менения процедуры $getTrees$ к корневому узлу леса разбора получим множество H, состоящее из одноэлементных списков. Элементы списков будут представлять собой деревья вывода исходной цепочки.

4. Процедуры для работы с лесом разбора

Опишем используемые синтаксическим анализатором процедуры для работы с лесом разбора.

Обозначим $getNodeI(s, v, w, z)$ процедуру, которая находит или создаёт промежуточный узел со значением $s \to v$, при этом левым дочерним узлом упаковывающего узла становится узел w, а правым — z (алгоритм 2).

Алгоритм 2. Получение промежуточного узла леса разбора

Вход: s — начальный узел СД; v — текущий узел СД; w — левый дочерний узел; z — правый дочерний узел.
Выход: промежуточный узел леса разбора.

ПРОЦЕДУРА $getNodeI(s, v, w, z)$
Если $(w = \$)$, **то**
  Вернуть z,
иначе
  пусть w отмечен (t, i, k), z отмечен (q, k, j);
  $y \leftarrow$ найти или создать узел с меткой $(s \to v, i, j)$.
  Если y не имеет упаковывающего ребёнка с меткой (t, q, k), **то**
    создать такого ребёнка с левым потомком w и правым z.
Вернуть y.

Обозначим $getNodeN(s, w, z)$ процедуру, которая находит или создаёт нетерминальный узел со значением s, при этом левым дочерним узлом упаковывающего узла становится узел w, а правым — z (алгоритм 3).

Алгоритм 3. Получение нетерминального узла леса разбора

Вход: s — начальный узел СД; w — левый дочерний узел; z — правый дочерний узел.
Выход: нетерминальный узел леса разбора.

ПРОЦЕДУРА $getNodeN(s, w, z)$
Пусть z отмечен (q, k, j).
Если $w \neq \$$, **то**
  пусть w отмечен (t, i, k),
иначе
  $i \leftarrow k$, $t \leftarrow \$$;
$y \leftarrow$ найти или создать узел с меткой (s, i, j).
Если y не имеет дочернего упаковывающего узла с меткой (t, q, k), **то**
  Если $w \neq \$$, **то**
    создать такой дочерний узел с левым потомком w и правым z,
  иначе
    создать такой дочерний узел с потомком z.
Вернуть y.

Обозначим:

— $convert(z)$ — процедура, которая преобразует промежуточный узел z к символьному узлу (алгоритм 4);
— $getNodeT(x, i)$ — процедура, которая находит или создаёт терминальный узел с меткой $(x, i, i + 1)$, где x — терминал, соответствующий i-й позиции входного буфера;
— $getNodeE(i)$ — процедура, которая находит или создаёт ϵ-узел с меткой (ϵ, i), где i — позиция входного буфера.

Эти процедуры возвращают найденный или созданный узел леса разбора.

Алгоритм 4. Преобразование промежуточного узла леса разбора в символьный узел

Вход: z — промежуточный узел леса разбора.
Выход: символьный узел леса разбора.

```
1: ПРОЦЕДУРА convert(z)
2: пусть (n → v, i, j) — метка z;
3: y ← найти или создать узел леса разбора с меткой (n, i, j).
4: Для всех дочерних узлов x узла z:
5:    Если не существует дуги из y в x, то
6:       добавить дугу.
7: Вернуть y.
```

5. Дескриптор и стек синтаксического анализатора

Определим конфигурацию синтаксического анализатора пятёркой (s, L, u, i, y), где s — начальный узел компоненты; L — метка программы; u — узел стека; i — позиция входного буфера; y — узел леса разбора. Назовём пятёрку (s, L, u, i, y) дескриптором синтаксического анализатора. Дескрипторы помещаются в множество дескрипторов D. Во внешнем цикле алгоритма работы синтаксического анализатора из D извлекается дескриптор и анализатор возобновляет свою работу, применяя сохранённую конфигурацию. Чтобы избежать зацикливания, дополнительно вводится множество U, в котором сохраняются все созданные дескрипторы.

Перед созданием дескриптора проверяется, существует ли уже такой дескриптор в множестве U. В этом случае повторного добавления дескриптора в множество D не происходит. Обозначим $add(s, L, u, i, y)$ процедуру, включающую дескриптор (s, L, u, i, y) в множества U и D, если множество U ещё не содержит (s, L, u, i, y).

Представим разбор как движение по СД, начинающееся со стартового узла. Если при обработке узла возможно несколько путей разбора (текущий символ принадлежит множеству выбора нескольких выходящих из узла дуг), то для каждого из них создаётся соответствующий дескриптор и помещается в множество D, а затем происходит переход к внешнему циклу алгоритма. При этом метка, сохраняемая в дескрипторе, — это метка программы, соответствующая фрагменту, описывающему разбор по одному из возможных путей.

При движении по компоненте будем строить часть леса разбора, соответствующую этой компоненте. Текущий узел на каждом шаге будем рассматривать как корень леса разбора. При входе в компоненту текущий узел леса разбора не определён. При выходе из компоненты корень должен представлять собой символьный узел (s, i, j), где s — начальный узел, через который был произведён вход в компоненту; i и j — границы подцепочки, выводимой в ходе движения по компоненте.

При переходе через терминальную вершину будем создавать в лесу разбора соответствующий терминалу узел, а затем объединять его с текущим корнем, создавая для них родительский промежуточный узел. Текущий узел леса разбора при этом станет левым ребёнком, а терминальный узел — правым. Созданный родительский узел станет новым корнем леса разбора.

При переходе через нетерминальную вершину будем запоминать позицию возврата и текущий корень леса разбора в стеке (см. ниже), а затем переходить к обработке компоненты, начальный узел которой записан в нетерминальной вершине.

При выходе из компоненты необходимо «извлечь» текущую вершину стека. При извлечении узла стека в множество D добавляются дескрипторы для каждой выходящей из извлекаемого узла дуги. Текущий корень леса разбора при этом нужно преобразовать в символьный узел, соответствующий начальному узлу обработанной компоненты, а затем объединить с узлом, которым отмечена выходящая дуга. Результат объединения добавляется в дескриптор в качестве узла леса разбора. После обработки компоненты происходит переход к внешнему циклу алгоритма работы синтаксического анализатора. В конце успешного разбора в лесу разбора должен быть создан корневой узел $(s, 0, m)$, где s — стартовый узел; m — длина входной цепочки.

Синтаксический анализатор использует стек с графовой структурой (Graph-Structured Stack, GSS) [11] для сохранения позиции возврата после обработки компоненты. Стек с графовой структурой представляет собой направленный граф, где каждый путь является стеком. Например, стек на рис. 4, *а* задаёт множество стеков на рис. 4, *б*. Если в процессе анализа происходит переход через нетерминальную вершину, то необходимо сохранить информацию об узле, следующем за нетерминальной вершиной, и текущем входе компоненты. Дуги стека отмечаются узлами леса разбора. В алгоритме GLL в стеке с графовой структурой разрешаются циклы, необходимые для обработки левой рекурсии.

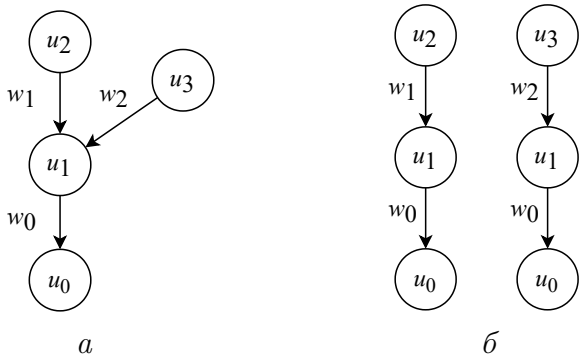


Рис. 4. Стек с графовой структурой (*а*) и задаваемые им стеки (*б*)

Определим метку узла стека как (s, v, i), где s — начальный узел СД, соответствующий текущему входу компоненты; v — узел СД, следующий за нетерминальной вершиной. Корневой узел стека обозначим u_0. Введём множество P, в которое будем сохранять узел стека u и узел леса разбора, которым отмечена выходящая из u дуга, при извлечении u из стека.

Определим процедуру $create(s, v, u, i, w)$, создающую в стеке родительский для узла u узел с меткой (s, v, i) (алгоритм 5). Дуга при этом помечается узлом w леса разбора.

Алгоритм 5. Создание узла стека

Вход: s — начальный узел компоненты; v — узел компоненты; u — узел стека; i — позиция входного буфера; w — узел леса разбора.

Выход: узел стека.

1: **ПРОЦЕДУРА** $create(s, v, u, i, w)$
2: $y \leftarrow$ найти или создать узел стека с меткой (s, v, i).
3: **Если** не существует дуги из y в u с меткой w, **то**
4: создать дугу из y в u с меткой w.
5: **Для всех** $(y, z) \in P$
6: $x \leftarrow getNodeI(s, v, w, z)$;
7: пусть h — правая граница z;
8: $add(s, L_v, u, h, x)$.
9: **Вернуть** y.

Определим процедуру $pop(u, i, z)$, извлекающую узел u из стека (алгоритм 6). Для каждой выходной дуги создаётся соответствующий ей дескриптор. В качестве узла леса разбора в дескриптор помещается родительский узел для z и узла, которым отмечена дуга.

Алгоритм 6. Извлечение узла стека

Вход: u — узел стека; u_0 — корневой узел стека; i — позиция входного буфера; z — узел леса разбора.

1: **ПРОЦЕДУРА** $pop(u, i, z)$
2: **Если** $u \neq u_0$, **то**
3: пусть (s, v, k) — метка u;
4: добавить (u, z) в P.
5: **Для всех** дуг (u, w, y)
6: $x \leftarrow getNodeI(s, v, w, z)$;
7: $add(s, L_v, y, i, x)$.

6. Построение синтаксического анализатора

Синтаксический анализатор состоит из фрагментов, соответствующих узлам СД, фрагментов, соответствующих выходящим из узлов СД дугам, и фрагмента, соответствующего внешнему циклу алгоритма. Каждому узлу v СД поставим в соответствие фрагмент, отмеченный меткой L_v; i-й дуге, выходящей из узла v, — фрагмент, отмеченный меткой $L_{(v,i)}$. Фрагмент, соответствующий внешнему циклу алгоритма, отметим L_0.

Введём следующие обозначения:

— I — входной поток токенов;
— c_I — текущая позиция во входном буфере;
— c_U — текущий узел стека;
— c_N — текущий узел леса разбора;
— c_S — текущий начальный узел компоненты СД.

Вместо вызова функций и возврата из функций, осуществляемых анализатором рекурсивного спуска, синтаксический анализатор будет выполнять переход на соответствующую метку.

6.1. Определение фрагментов, соответствующих узлам

Пусть из узла v выходят n дуг $e_1, \ldots, e_n$. Множество выбора узла v равно объединению множеств выбора выходящих из него дуг: ВЫБОР(v) = ВЫБОР$(e_1) \cup \ldots \cup$ $\cup$ВЫБОР(e_n). Для каждого терминала $t \in$ ВЫБОР(v) определим множество дуг $E(t)$, таких, что t принадлежит множеству выбора этих дуг: $E(t) = \{e : t \in \text{ВЫБОР}(e)\}$.

Определим отношение R на множестве ВЫБОР(v): $R = \{(t_i, t_j) : E(t_i) = E(t_j)\}$. Оно является отношением эквивалентности и разбивает множество ВЫБОР(v) на классы эквивалентности $Q_1, \ldots, Q_n$. Если $(t_i, t_j) \in R$, то терминалы t_i и t_j назовём эквивалентными. Класс эквивалентности Q_i представляет собой множество попарно эквивалентных терминалов. Множество дуг $E(Q_i)$ равно множеству $E(t)$, если $t \in Q_i$.

Фрагмент для узла v опишем следующим образом: L_v : $code(v)$. Определение $code(v)$ приведено в алгоритме 7.

Алгоритм 7. Фрагмент $code(v)$, соответствующий узлу v

1: **Если** $I[c_I] \in Q_1$, **то**
2: $\quad code(Q_1)$.
3: ...
4: **Если** $I[c_I] \in Q_n$, **то**
5: $\quad code(Q_n)$.
6: Переход L_0.

Положим $E(Q_i) = \{e_1, \ldots, e_n\}$. Определение $code(Q_i)$ приведено в алгоритме 8.

Алгоритм 8. Фрагмент $code(Q_i)$, соответствующий множеству Q_i

1: $add(c_S, L_{(v,1)}, c_U, c_I, c_N)$;
2: ...
3: $add(c_S, L_{(v,n)}, c_U, c_I, c_N)$;
4: переход L_0.

Если $E(Q_i) = \{e_i\}$ (одноэлементное множество), то создание дескриптора можно опустить. В этом случае $code(Q_i)$ = переход $L_{(v,i)}$.

6.2. Определение фрагментов, соответствующих дугам

Определим фрагмент, описывающий i-ю дугу e, выходящую из узла v, следующим образом: $L_{(v,i)} : code(e)$.

Для дуги $e = (v, x, w)$, с которой начинается переход из узла v в узел w через вершину x, возможны следующие ситуации:

1) Если x — терминал, то в лесу разбора необходимо создать терминальный узел, а затем объединить его с текущим корнем леса разбора (алгоритм 9).
2) Если x — начальный узел, то в стек помещается позиция возврата и происходит переход на метку, соответствующую этому узлу (алгоритм 10).

При переходе по выходной дуге $e = (v, \epsilon)$ необходимо преобразовать текущий корень леса разбора в символьный узел, соответствующий входу обрабатываемой компоненты, а затем извлечь узел из стека. Текущий корень c_N может быть как промежу-

Алгоритм 9. Фрагмент $code(v, x, w)$, соответствующий дуге $e = (v, x, w)$

1: $c_R \leftarrow getNodeT(x, c_I)$;
2: $c_N \leftarrow getNodeI(c_S, w, c_N, c_R)$;
3: $c_I \leftarrow c_I + 1$;
4: переход L_w.

Алгоритм 10. Фрагмент $code(v, X, w)$, соответствующий дуге $e = (v, X, w)$

1: $c_U \leftarrow create(c_S, w, c_U, c_I, c_N)$;
2: $c_N \leftarrow \$$;
3: $c_S \leftarrow X$;
4: переход L_x.

точным узлом, если в ходе движения по компоненте было пройдено несколько терминальных или нетерминальных вершин, так и символьным узлом, если в ходе движения по компоненте была пройдена только одна вершина. Если $c_N = \$$, то такая ситуация соответствует переходу по ϵ-пути в компоненте и требует создания ϵ-узла в дереве разбора. Определение $code(v, \epsilon)$ приведено в алгоритме 11. При этом если в компоненте нет ϵ-пути, то ветвь по условию $c_N = \$$ можно опустить.

Алгоритм 11. Фрагмент $code(v, \epsilon)$, соответствующий выходной дуге

1: **Если** $c_N = \$$, **то**
2: $c_R \leftarrow getNodeE(c_I)$;
3: $c_N \leftarrow getNodeN(c_S, c_N, c_R)$,
4: **иначе**
5: **Если** c_N — символьный узел, **то**
6: $c_N \leftarrow getNodeN(c_S, \$, c_N)$,
7: **иначе**
8: $c_N \leftarrow convert(c_N)$.
9: $pop(c_U, c_I, c_N)$;
10: переход L_0.

6.3. Уменьшение количества операторов перехода

В предложенном подходе переход на метку будет выполняться после обработки каждой выходящей из узла дуги. Для уменьшения количества операторов перехода определим цепочку как последовательность узлов и дуг, которые могут быть обработаны без передачи управления другому фрагменту. Вместо фрагментов, описывающих выходящие из узлов дуги, будем использовать фрагменты, описывающие цепочки.

Цепочка начинается дугой, выходящей из узла, и заканчивается дугой, входящей в узел. Промежуточные узлы и дуги принадлежат цепочке. На рис. 5 из узла 1 выходят четыре цепочки: первая цепочка начинается дугой e_1 и заканчивается дугой e_2, вторая — начинается дугой e_3 и заканчивается дугой e_4, третья — начинается дугой e_5 и заканчивается дугой e_6, четвертая — начинается дугой e_7 и заканчивается дугой e_8. Анализатор будет содержать фрагмент, описывающий узел, только в том случае, если узел не принадлежит ни одной цепочке. Такими узлами являются начальные узлы компонент и узлы, в которые входят последние дуги цепочек (на рис. 5 это узлы 1, 4, 6 и 8). Определим множество выбора цепочки как множество выбора дуги, с которой

начинается цепочка. Во фрагменте, описывающем узел, будем проверять принадлежность текущего символа множествам выбора выходящих из узла цепочек.

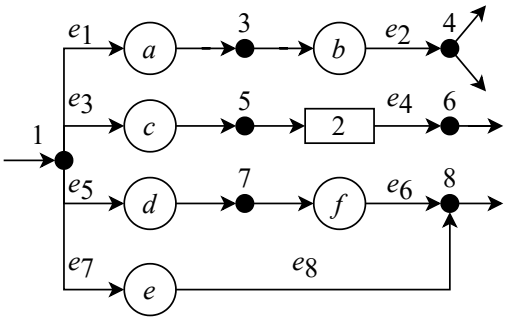


Рис. 5. Выделение цепочек

Пусть $e_1, \ldots, e_n$ — принадлежащие одной цепочке дуги, выходящие из узлов, и первая дуга цепочки — это i-я дуга, выходящая из узла v. Определим фрагмент, описывающий эту цепочку, следующим образом: $L_{(v,i)}$: $code(v, i)$. Определение $code(v, i)$ приведено в алгоритме 12.

Алгоритм 12. Фрагмент $code(v, i)$, соответствующий цепочке

```
code(e_1)
Если I[c_I] ∈ ВЫБОР(e_2), то
    code(e_2),
иначе
    переход L_0.
...
Если I[c_I] ∈ ВЫБОР(e_n), то
    code(e_n),
иначе
    переход L_0.
```

При этом из всех фрагментов $code(e_i)$ (см. п. 6.2), кроме последнего, необходимо исключить операторы перехода.

Определим ситуации, в которых цепочка прерывается:

1) дуга входит в узел, из которого выходит несколько дуг (цепочка $e_1 - e_2$);
2) дуга выходит из нетерминальной вершины (цепочка $e_3 - e_4$);
3) дуга входит в узел, в который входит несколько дуг (цепочки $e_5 - e_6$, $e_7 - e_8$);
4) дуга входит в начальный узел компоненты.

6.4. Общий вид синтаксического анализатора

Общий вид синтаксического анализатора, построенного по синтаксической диаграмме с многовходовыми компонентами, приведён в алгоритме 13. Здесь $A, \ldots, X$ — узлы СД, из которых выходят цепочки; S — стартовый узел; n — количество цепочек, выходящих из узла A; k — количество цепочек, выходящих из узла X.

Алгоритм 13. Общий вид синтаксического анализатора

Вход: m — длина входного буфера без учёта символа конца цепочки; I — входной поток токенов.

Выход: корневой узел КПЛР входной цепочки.

ПРОЦЕДУРА $parse(I, m)$
  $c_U \leftarrow$ создать в стеке узел u_0 с меткой $(s, 0, 0)$;
  $c_N \leftarrow \$$, $c_I \leftarrow 0$, $c_S \leftarrow S$;
  $U \leftarrow \varnothing$, $R \leftarrow \varnothing$, $P \leftarrow \varnothing$;
  переход L_S.
L_0:
  Если $D \neq \varnothing$, **то**
    извлечь (v, L_k, u, i, w) из D;
    $c_U \leftarrow u$, $c_N \leftarrow w$, $c_I \leftarrow I$, $c_S \leftarrow v$;
    переход L_k,
  иначе
    Если существует узел леса разбора с меткой $(S, 0, m)$, **то**
      удалить из леса разбора все узлы, недостижимые из узла $(S, 0, m)$.
      Вернуть узел $(S, 0, m)$,
    иначе
      Вернуть ошибку.

L_A:
  $code(A)$
...
L_X:
  $code(X)$
$L_{(A,1)}$:
  $code(A, 1)$
...
$L_{(A,n)}$:
  $code(A, n)$
...
$L_{(X,1)}$:
  $code(X, 1)$
...
$L_{(X,k)}$:
  $code(X, k)$

7. Пример построения синтаксического анализатора на основе синтаксической диаграммы с многовходовыми компонентами

Построим синтаксический анализатор по синтаксической диаграмме с многовходовыми компонентами на рис. 1. Общий вид анализатора приведён в алгоритме 14.

Обозначим i-ю дугу, выходящую из узла v, как v_i (нумерация сверху вниз). Определим множества выбора всех дуг СД ($\perp$ — символ конца цепочки):

ВЫБОР$(1_1) = \{d\}$;
ВЫБОР$(1_2) = \{a\}$;
ВЫБОР$(1_3) = \{a, b\}$;
ВЫБОР$(2_1) = \{a\}$;
ВЫБОР$(2_2) = \{b\}$;
ВЫБОР$(3_1) = \{c, \perp\}$;
ВЫБОР$(4_1) = \{c\}$;
ВЫБОР$(5_1) = \{b\}$.

Определим узлы, из которых выходят цепочки: 1, 2, 3, 4. Узлы 1 и 2 являются начальными, в узлы 3 и 4 входит несколько дуг, кроме того, в узел 4 входит дуга,

Алгоритм 14. Общий вид анализатора, построенного по СД на рис. 1

```
1: ПРОЦЕДУРА parse(I, m)
2:   c_U ← создать в стеке узел u_0 с меткой (1, 0, 0);
3:   c_N ← $, c_I ← 0, c_S ← 1, U ← ∅, D ← l, P ← ∅;
4:   переход L_1.
5: L_0:
6:   Если D ≠ ∅, то
7:     извлечь (v, L_k, u, i, w) из D;
8:     c_U ← u, c_N ← w, c_I ← I, c_S ← v;
9:     переход L_k,
10:  иначе
11:    Если существует узел леса разбора с меткой (1, 0, m), то
12:      удалить из леса разбора все недостижимые из узла (1, 0, m) узлы.
13:      Вернуть узел (1, 0, m),
14:    иначе
15:      Вернуть ошибку.
16: <фрагменты для узлов СД>
17: <фрагменты для цепочек СД>
```

выходящая из нетерминальной вершины. Фрагменты для цепочек приведены в алгоритме 15.

Разобьём терминалы из множества выбора каждого узла на непересекающиеся подмножества $Q_1, \ldots, Q_n$ и определим для каждого подмножества множество $E(Q_i)$. Множество выбора узла 1 содержит терминалы a, b, d. Терминал d принадлежит множеству выбора первой дуги, терминал a — множествам выбора второй и третьей дуги, терминал b — множеству выбора третьей дуги. Получаем $Q_1 = \{d\}$ и $E(Q_1) = \{1\}$, $Q_2 = \{a\}$ и $E(Q_2) = \{2, 3\}$, $Q_3 = \{b\}$ и $E(Q_3) = \{3\}$. Фрагменты для узлов приведены в алгоритме 16.

Алгоритм 15. Фрагменты для цепочек СД на рис. 1

$L_{(1,1)}$:
  $c_R \leftarrow getNodeT(d, c_I)$;
  $c_N \leftarrow getNodeI(c_S, 3, c_N, c_R)$;
  $c_I \leftarrow c_I + 1$;
  переход L_3.
$L_{(1,2)}$:
  $c_R \leftarrow getNodeT(a, c_I)$;
  $c_N \leftarrow getNodeI(c_S, 4, c_N, c_R)$;
  $c_I \leftarrow c_I + 1$;
  переход L_4.
$L_{(1,3)}$:
  $c_U \leftarrow create(c_S, 4, c_U, c_I, c_N)$;
  $c_N \leftarrow \$$;
  $c_S \leftarrow 2$;
  переход L_2.
$L_{(2,1)}$:
  $c_R \leftarrow getNodeT(a, c_I)$;
  $c_N \leftarrow getNodeI(c_S, 3, c_N, c_R)$;
  $c_I \leftarrow c_I + 1$;
  переход L_3.
$L_{(2,2)}$:
  $c_R \leftarrow getNodeT(b, c_I)$;
  $c_N \leftarrow getNodeI(c_S, 5, c_N, c_R)$;
  $c_I \leftarrow c_I + 1$;
  Если $I[c_I] \in \{b\}$, **то**
    $c_R \leftarrow getNodeT(b, c_I)$;
    $_N \leftarrow getNodeI(c_S, 3, c_N, c_R)$;
    $c_I \leftarrow c_I + 1$;
    переход L_3,
  иначе
    переход L_0.
$L_{(3,1)}$:
  Если c_N — символьный узел, **то**
    $c_N \leftarrow getNodeN(c_S, \$, c_N)$,
  иначе
    $c_N \leftarrow convert(c_N)$.
  $pop(c_U, c_I, c_N)$;
  переход L_0.
$L_{(4,1)}$:
  $c_R \leftarrow getNodeT(c, c_I)$;
  $c_N \leftarrow getNodeI(c_S, 3, c_N, c_R)$;
  $c_I \leftarrow c_I + 1$;
  переход L_3.

Алгоритм 16. Фрагменты для узлов СД на рис. 1

L_1:
  Если $I[c_I] \in \{d\}$, **то**
    переход $L_{(1,1)}$.
  Если $I[c_I] \in \{a\}$, **то**
    $add(c_S, L_{(1,2)}, c_U, c_I, c_N)$;
    $add(c_S, L_{(1,3)}, c_U, c_I, c_N)$;
    переход L_0.
  Если $I[c_I] \in \{b\}$, **то**
    переход $L_{(1,3)}$;
  переход L_0.
L_2:
  Если $I[c_I] \in \{a\}$, **то**
    переход $L_{(2,1)}$.
  Если $I[c_I] \in \{b\}$, **то**
    переход $L_{(2,2)}$;
  переход L_0.
L_3:
  Если $I[c_I] \in \{c, \perp\}$, **то**
    переход $L_{(3,1)}$;
  переход L_0.
L_4:
  Если $I[c_I] \in \{c\}$, **то**
    переход $L_{(4,1)}$;
  переход L_0.

Заключение

Предложенный алгоритм позволяет строить компактные и эффективные по времени работы синтаксические анализаторы по синтаксическим диаграммам с многовходовыми компонентами. Для детерминированной СД временная сложность полученного анализатора — $O(m)$, где m — длина входной цепочки, при условии, что поиск узлов в лесу разбора и стеке организован за независимое от m время. Если вывод заданной цепочки в произвольной СД детерминирован, то время работы анализатора также линейно относительно длины цепочки. Для организации быстрого поиска узлы могут храниться в многомерных массивах, размер которых зависит от длины входной цепочки и количества узлов диаграммы. Недостатком этого подхода является большой объём памяти, занимаемой таким массивом.

Предложенный алгоритм может быть применён к СД произвольной структуры, что исключает необходимость её предварительной обработки.

Алгоритм построения синтаксических анализаторов может быть использован как в системах автоматизированного построения трансляторов, так и при «ручном» проектировании. Построенные по предложенному алгоритму синтаксические анализаторы могут применяться для анализа любых контекстно-свободных языков, включая недетерминированные и неоднозначные.

ЛИТЕРАТУРА

1. *Grau E. A.* Recursive processes and ALGOL translation // Comm. ACM. 1961. V. 4. P. 10–15.
2. *Scott E. and Johnstone A.* GLL parsing // Electr. Notes Theor. Comput. Sci. 2010. V. 253. P. 177–189.
3. *Scott E. and Johnstone A.* GLL parse-tree generation // Sci. Comput. Programming. 2013. V. 78. P. 1828–1844.
4. *Рязанов Ю. Д.* Минимизация синтаксических диаграмм с многовходовыми компонентами // Прикладная дискретная математика. 2018. № 41. С. 85–97.
5. *Рязанов Ю. Д., Севальнева М. Н.* Анализ синтаксических диаграмм и синтез программ-распознавателей линейной сложности // Научные ведомости БелГУ. Сер. История. Политология. Экономика. Информатика. 2013. № 8. С. 128–136.
6. *Рязанов Ю. Д., Крамаренко П. В.* Графовый способ анализа синтаксических диаграмм // Научный электронный архив. `http://econf.rae.ru/article/8214`. 2014.
7. *Поляков В. М., Рязанов Ю. Д.* Алгоритм построения нерекурсивных программ-распознавателей линейной сложности по детерминированным синтаксическим диаграммам // Вестник БГТУ им. В. Г. Шухова. 2013. № 6. С. 194–199.
8. *Рязанов Ю. Д.* Преобразование недетерминированных синтаксических диаграмм в детерминированные // Вестник Воронежского госуниверситета. Сер. Системный анализ и информационные технологии. 2015. № 1. С. 139–147.
9. *Рязанов Ю. Д.* Способ устранения конфликтов типа «переход — выход» в синтаксических диаграммах // Вестник Воронежского госуниверситета. Сер. Системный анализ и информационные технологии. 2015. № 4. С. 130–137.
10. *Tomita M.* Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems. Kluwer Academic Publ., 1985.
11. *Tomita M.* Graph-structured stack and natural language parsing // 26th Ann. Meeting of the Association of Computational Linguistics, 7–10 June 1988, Buffalo, New York, USA. P. 249–257.

REFERENCES

1. *Grau E. A.* Recursive processes and ALGOL translation. Comm. ACM, 1961, vol. 4, pp. 10–15.
2. *Scott E. and Johnstone A.* GLL parsing. Electr. Notes Theor. Comput. Sci., 2010, vol. 253, pp. 177–189.
3. *Scott E. and Johnstone A.* GLL parse-tree generation. Sci. Comput. Programming, 2013, vol. 78, pp. 1828–1844.
4. *Ryazanov Yu. D.* Minimizaciya sintaksicheskih diagramm s mnogovhodovymi komponentami [Minimization syntax diagrams with multiport components]. Prikladnaya Diskretnaya Matematika, 2018, no. 41. pp. 85–97. (in Russian)
5. *Ryazanov Yu. D. and Seval'neva M. N.* Analiz sintaksicheskih diagramm i sintez programm-raspoznavatelej linejnoj slozhnosti [The analysis of syntax diagrams and automatic generation of linear-time programs-recognizer]. Belgorod State University Scientific Bull. Ser. History. Political Science. Economics. Inform. Technologies, 2013, no. 8, pp. 128–136. (in Russian)
6. *Ryazanov Yu. D. and Kramarenko P. V.* Grafovyj sposob analiza sintaksicheskih diagramm [Graph method for parsing syntax diagrams]. `http://econf.rae.ru/article/8214`, 2014. (in Russian)
7. *Polyakov V. M. and Ryazanov Yu. D.* Algoritm postroeniya nerekursivnyh programm-raspoznavatelej linejnoj slozhnosti po determinirovannym sintaksicheskim diagrammam [Algorithm for not recursive linear-time programs-recognizer design from deterministic syntax diagrams]. Bull. BSTU named after V. G. Shukhov, 2013, no. 6, pp. 194–199. (in Russian)
8. *Ryazanov Yu. D.* Preobrazovanie nedeterminirovannyh sintaksicheskih diagramm v determinirovannye [Converting nondeterministic syntax diagrams to deterministic ones]. Bull. Voronezh State University. Ser. Systems Analysis and Inform. Technology, 2015, no. 1, pp. 139–147. (in Russian)
9. *Ryazanov Yu. D.* Sposob ustraneniya konfliktov tipa "perekhod — vyhod" v sintaksicheskih diagrammah [Method for resolving conflicts of type "shift — reduce" in syntax diagrams]. Bull. Voronezh State University. Ser. Systems Analysis and Inform. Technology, 2015, no. 4, pp. 130–137. (in Russian)
10. *Tomita M.* Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems. Kluwer Academic Publ., 1985.
11. *Tomita M.* Graph-structured stack and natural language parsing. 26th Ann. Meeting of the Association of Computational Linguistics, 7–10 June 1988, Buffalo, New York, USA, pp. 249–257.

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 512.772 DOI 10.17223/20710410/55/9

IMPLEMENTATION OF POINT-COUNTING ALGORITHMS ON GENUS 2 HYPERELLIPTIC CURVES BASED ON THE BIRTHDAY PARADOX[1]

N. Kolesnikov

Immanuel Kant Baltic Federal University, Kaliningrad, Russia

E-mail: NiKolesnikov1@kantiana.ru

Our main contribution is an efficient implementation of the Gaudry — Schost and Galbraith — Ruprai point-counting algorithms on Jacobians of hyperelliptic curves. Both of them are low memory variants of Matsuo — Chao — Tsujii (MCT) Baby-Step Giant-Step-like algorithm. We present an optimal memory restriction (a time-memory tradeoff) that minimizes the runtime of the algorithms. This tradeoff allows us to get closer in practical computations to theoretical bounds of expected runtime at $2.45\sqrt{N}$ and $2.38\sqrt{N}$ for the Gaudry — Schost and Galbraith — Ruprai algorithms, respectively. Here N is the size of the 2-dimensional searching space, which is as large as the Jacobian group order, divided by small modulus m, precomputed by using other techniques. Our implementation profits from the multithreaded regime and we provide some performance statistics of operation on different size inputs. This is the first open-source parallel implementation of 2-dimensional Galbraith — Ruprai algorithm.

Keywords: *hyperelliptic curve, Jacobian, point-counting, birthday paradox.*

РЕАЛИЗАЦИЯ АЛГОРИТМОВ ПОДСЧЁТА ТОЧЕК В ЯКОБИАНАХ ГИПЕРЭЛЛИПТИЧЕСКИХ КРИВЫХ РОДА 2, ОСНОВАННЫХ НА ПАРАДОКСЕ ДНЕЙ РОЖДЕНИЯ

Н. С. Колесников

Балтийский федеральный университет им. Иммануила Канта, г. Калининград, Россия

Представлена эффективная программная реализация алгоритма Годри — Шоста и его модификации Гэлбрайта — Рупрая для подсчёта точек в якобианах гиперэллиптических кривых. Эти алгоритмы представляют собой версии алгоритма Мацуо — Чао — Цуджия с малым использованием памяти и реализуют стратегию Гельфонда — Шенкса больших и малых шагов. Выводится оптимальный размер памяти, позволяющий минимизировать время работы указанных алгоритмов и получить на практике ожидаемое время их работы, близкое к теоретическим оценкам $2{,}45\sqrt{N}$ и $2{,}38\sqrt{N}$ для алгоритмов Годри — Шоста и Гэлбрайта — Рупрая соответственно. Здесь N — размер двумерной области поиска, равный порядку

[1]The publication was supported by the Russian Academic Excellence Project '5-100' 2016–2020.

якобиана кривой, уменьшенному в m раз с помощью других методов. Предлагаемая реализация алгоритмов имеет многопоточный режим работы. Представлена статистическая зависимость времени работы от размера входных данных. Данная реализация алгоритма Гэлбрайта — Рупрая для размерности 2 является первой опубликованной многопоточной реализацией этого алгоритма с открытым исходным кодом.

Ключевые слова: *гиперэллиптическая кривая, якобиан, подсчёт точек, парадокс дней рождения.*

1. Introduction

Jacobians of hyperelliptic curves can be regarded as groups of large order that is difficult to compute. They have many applications in cryptography, such as DLP-based cryptosystems, where the efficiency of order computing algorithms is crucial. The MCT point-counting algorithm [1] is a natural attempt to adapt the general purpose BSGS algorithm to Jacobians of genus 2 hyperelliptic curves. This algorithm turned out to be unusable for cryptographic size instances due to its extremely high memory consumption. P. Gaudry and E. Schost have proposed [2] a low-memory version of the MCT algorithm and provided its experimental implementation. However, the memory limit in their implementation must be set by the user, and the authors do not give its optimal values.

Our main contribution is an efficient multithreaded implementation of the Gaudry — Schost point-counting algorithm with optimal memory settings that can be viewed as a time-memory tradeoff. We also implement an improvement of the Gaudry — Schost algorithm, proposed by S. Galbraith and R. Ruprai [3], and test the performance of both algorithms. This is the first open source implementation of the 2-dimensional Galbraith — Ruprai algorithm.

The paper is organized as follows. In Section 3, we give a review of the most efficient BSGS-like techniques proposed in [2, 3]. Both of them have heuristic complexity analyses based on the birthday paradox. These algorithms use the strategy of pseudo-random walks and store in RAM only a portion of points computed on "baby" and "giant" steps. In Section 4, we describe an efficient data structure to store distinguished points and an optimal storage size to reach the tradeoff between time and memory. Section 5 describes our implementation of the Gaudry — Schost and Galbraith — Ruprai algorithms and shows some statistics on their performance.

2. Preliminaries

We consider genus $g = 2$ hyperelliptic curves defined over a finite field $\mathbb{F}_q$ with prime $q > 2$. The curve is defined by the equation

$$\mathcal{C} : y^2 = x^{2g+1} + \sum_{i=0}^{2g} f_i x^i.$$

A set of all reduced divisors of a curve form a group $Jac(\mathcal{C})$, which is called a Jacobian of a curve. Its elements sometimes are regarded as "points". The group law (point addition) on Jacobian is defined by Cantor's addition formulas. We normally write reduced divisors in Mumford coordinates $D = (u(x), v(x))$, where $u(x)$ and $v(x)$ are polynomials, such that $\deg v(x) < \deg u(x) \leqslant g$. For further details on these algebraic structures, we refer to [4].

We denote by $\chi(T)$ the characteristic polynomial of the Frobenius endomorphism $\pi_{\mathcal{C}}$ on $Jac(\mathcal{C})$:

$$\chi(T) = T^4 - s_1 T^3 + s_2 T^2 - q s_1 T + q^2.$$

3. Review of Birthday-paradox algorithms

3.1. Gaudry — Schost

Following P. Gaudry and E. Schost [5], before launching an exponential MCT point-counting algorithm, one can calculate $\#Jac(\mathcal{C}) \bmod m$, where $m = \ell_1^{k_1} \cdot \ldots \cdot \ell_s^{k_s}$, and ℓ_i are small primes. The cost of computing $\#Jac(\mathcal{C}) \bmod \ell$ takes $O(\log^7(\ell))$ operations [5, p. 2] in $\mathbb{F}_q$. In practice, we computed $\#Jac(\mathcal{C})$ modulo $\ell \leqslant 43$, which took ~ 40 Gb of RAM.

After gathering some modular information about $\#Jac(\mathcal{C})$, we proceed to the exponential Gaudry — Schost point-counting algorithm. The desired order of the Jacobian can be computed by substituting $T = 1$ into Frobenius characteristic polynomial

$$\#Jac(\mathcal{C}) = \chi(1) = q^2 + 1 - s_1(q+1) + s_2. \tag{1}$$

Thus, we have to find two values s_1 and s_2. We write them in the form $s_i = \overline{s_i} + m\widetilde{s_i}$, $i = 1, 2$, assuming that we have $\overline{s_1}, \overline{s_2}$ precomputed. This reduces our search space for a tuple of unknowns $(\widetilde{s_1}, \widetilde{s_2})$ to the following bounds [2, p. 3]:

$$-\frac{\sqrt{q}}{m} \leqslant \widetilde{s_1} \leqslant \frac{\sqrt{q}}{m}, \quad -\frac{2q}{m} \leqslant \widetilde{s_2} \leqslant \frac{6q}{m}. \tag{2}$$

Then we proceed as follows. We choose a random divisor $D \in Jac(\mathcal{C})$ and try to compute its order. As the order $\mathrm{ord}(D)$ divides the group order $\#Jac(\mathcal{C})$, we have $\chi(1)D = 0$. Combining this equation with (1) gives

$$\left(q^2 + 1 - \overline{s_1}(q+1) + \overline{s_2}\right) D + \left(-\widetilde{s_1}(q+1) + \widetilde{s_2}\right) m \cdot D = 0. \tag{3}$$

We form the two sets of divisors: W ("Wild") and T ("Tame"), and enumerate elements in these sets until we find a point belonging to both W and T (a collision). A constant K' contains known terms of (3) and a correction term to make the bounds (2) on $\widetilde{s_1}, \widetilde{s_2}$ symmetric. We later subtract this correction term once we find a collision:

$$W = \{K' \cdot D + (-\sigma_1(q+1) + \sigma_2)\, m \cdot D : (\sigma_1, \sigma_2) \in R\},$$
$$T = \{(-\sigma_1(q+1) + \sigma_2)\, m \cdot D : (\sigma_1, \sigma_2) \in R\},$$
$$R = [B_{1,\min}, B_{1,\max}] \times [B_{2,\min}, B_{2,\max}],$$
$$B_{1,\min} = -\frac{\sqrt{q}}{m}, \quad B_{1,\max} = \frac{\sqrt{q}}{m}, \quad B_{2,\min} = -\frac{2q}{m}, \quad B_{2,\max} = \frac{6q}{m},$$
$$K' = q^2 + 1 - \overline{s_1}(q+1) + \overline{s_2} + m\left(-\left\lfloor \frac{B_{1,\min} + B_{1,\max}}{2} \right\rfloor (q+1) + \left\lfloor \frac{B_{2,\min} + B_{2,\max}}{2} \right\rfloor\right).$$

Two colliding points $D_W = (\sigma_{1W}, \sigma_{2W})$ and $D_T = (\sigma_{1T}, \sigma_{2T})$ give us the unknown values $\widetilde{s_1}$, $\widetilde{s_2}$ by the following equations derived from (3):

$$\widetilde{s_i} = \sigma_{i,W} - \sigma_{i,T} + \left\lfloor \frac{B_{i,\min} + B_{i,\max}}{2} \right\rfloor, \quad i = 1, 2.$$

The time complexity of the algorithm depends on the cardinality of the intersection $|W \cap T|$ which lies in the interval $[0.25|R|..|R|]$ depending on the curve given. The expected number of points to be constructed until we get a collision is $\sqrt{\pi \cdot |W \cap T|}$, that follows from the Theorem 1 below. Thus, the expected number of points in the best, worst and average [6] cases are $1.77|R|$, $3.54|R|$ and $2.43|R|$ respectively.

Theorem 1 (Tame-Wild birthday paradox). Suppose that we have two urns both containing M balls numbered from 1 to M. The first urn contains only white balls, the second urn contains only red balls. We are choosing the ball uniformly at random in course from the first and the second urn, save its number and color, and return the ball to the urn. Then the expected number of selections until we get two colliding numbers of different colors is

$$\sqrt{\pi M} + O(1).$$

We denote by $\mathbb{P}(M, m_1, m_2)$ the probability that after m_1 steps in urn 1 and m_2 steps in urn 2 no matches were found. K. Nishimura and M. Sibuya prove in [7] that if we are restricted to $m_1 = m_2 = m = O(\sqrt{M})$, $M \to \infty$, this probability tends to

$$\mathbb{P}(M, m, m) \to \exp\left(-\frac{m^2}{M}\left[1 + O\left(\frac{1}{\sqrt{M}}\right)\right]\right) \approx \exp\left(-\frac{m^2}{M}\right).$$

Let X be a random variable that represents the number of selections of any urn before we get a collision. Then the cumulative distribution function is

$$F_X(m) = 1 - \mathsf{P}[X > m] = 1 - \mathbb{P}(M, m, m).$$

This fact is used to calculate the expectation of X that gives

$$\mathbb{E}(X) = \sqrt{\pi M}/2.$$

.

Remark 1. One should note that if we fix the input parameters $(\mathbb{F}_q, \mathcal{C}, m)$ and run the Gaudry — Schost point-counting algorithm several times, the best and the worst running time estimates could not be compared with the values $1.77|R|$ and $3.54|R|$. To estimate the deviation on running time with the fixed input, we need to compute the variance $\mathrm{Var}(X)$ of the random variable above. This deviation we will use in practical experiments:

$$\mathbb{E}(X^2) = \sum_{i=1}^{\infty} i^2 \cdot \mathsf{P}[X = i] = \sum_{i=1}^{\infty} i^2 \cdot \mathsf{P}[X > i - 1] - \sum_{i=1}^{\infty} i^2 \cdot \mathsf{P}[X > i] =$$

$$= \sum_{i=0}^{\infty} (i+1)^2 \cdot \mathsf{P}[X > i] - \sum_{i=1}^{\infty} i^2 \cdot \mathsf{P}[X > i] =$$

$$= \sum_{i=0}^{\infty} (2i+1)\,\mathsf{P}[X > i] = \int_{i=0}^{\infty} (2i+1)\exp\left(-i^2/M\right) di =$$

$$= \int_{x=0}^{\infty} 2x \cdot \exp\left(-x^2/M\right) dx + \int_{x=0}^{\infty} \exp\left(-x^2/M\right) dx = M + \frac{\sqrt{\pi M}}{2},$$

$$\mathrm{Var}(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2 = M \cdot \left(1 - \frac{\pi}{4}\right) + \frac{\sqrt{\pi M}}{2}.$$

Thus, the standard deviation for X is given by

$$\sqrt{\mathrm{Var}(X)} \underset{M\to\infty}{=} \sqrt{1 - \pi/4} \cdot \sqrt{M} \approx 0.46\sqrt{M}.$$

Gaudry and Schost in [5] also present an approach of random walks and distinguished points that significantly reduces the memory requirement of the algorithm. The idea is to name a portion of points in the search space R as "distinguished" points. This is done by selecting an appropriate hash function and looking at some bits of each hash value.

As before, we choose a random divisor $D \in Jac(\mathcal{C})$, calculate its hash value $h(D)$. This value determines the direction of a random walk. The next divisor we choose is $D + \mathcal{O}_{h(D)}$, where "$\mathcal{O}$" is a short list of precomputed shifts that defines the behavior of a random walk. We continue the walk unless a distinguished point is hit. As soon as this happens, the distinguished point is saved to an appropriate W or T list. Although the points in the constructed chain are not taken from the search space uniformly at random, the complexity analysis, based on the Tame-Wild birthday paradox, heuristically remains valid. However, an average runtime and memory requirements vary depending on the random walk parameters. We discuss the optimal random walk set up in Section 4.

3.2. Galbraith — Ruprai

S. Galbraith and R. Ruprai proposed in [3] an improvement for the Gaudry — Schost algorithm, that reduces the search space R in a tricky way, and makes the cardinality of intersection $|W \cap T|$ constant for all curve instances. As a result, the expected number of points to be constructed in any of the cases: best, worst, or average is invariant and equals to $2.38|R|$. The notions of random walks, distinguished points, and its complexity analysis remain the same. The search space and Tame-Wild sets are defined as follows: the Tame searching rectangle is reduced in length and width by 2/3 times, while the Wild searching rectangle is constructed as a union of 4 disjointed "corners" of R:

$$R_T = [2/3B_{1,\min}, 2/3B_{1,\max}] \times [2/3B_{2,\min}, 2/3B_{2,\max}],$$
$$R_W = [B_{1,\min}, B_{1,\min} + (B_{1,\max} - B_{1,\min})/3] \times [B_{2,\max}, B_{2,\max} - (B_{2,\max} - B_{2,\min})/3] \cup$$
$$\cup [B_{1,\max}, B_{1,\max} - (B_{1,\max} - B_{1,\min})/3] \times [B_{2,\max}, B_{2,\max} - (B_{2,\max} - B_{2,\min})/3] \cup$$
$$\cup [B_{1,\min}, B_{1,\min} + (B_{1,\max} - B_{1,\min})/3] \times [B_{2,\min}, B_{2,\min} + (B_{2,\max} - B_{2,\min})/3] \cup$$
$$\cup [B_{1,\max}, B_{1,\max} - (B_{1,\max} - B_{1,\min})/3] \times [B_{2,\min}, B_{2,\min} + (B_{2,\max} - B_{2,\min})/3],$$
$$W' = \{K' \cdot D + (-\sigma_1(q+1) + \sigma_2)\, m \cdot D : (\sigma_1, \sigma_2) \in R_W\},$$
$$T' = \{(-\sigma_1(q+1) + \sigma_2)\, m \cdot D : (\sigma_1, \sigma_2) \in R_T\}.$$

Random walks on T' and W' operate similarly to the Gaudry — Shost algorithm. The only difference is that we choose a random divisor that initiates a chain in one of the corners of R_W and do not jump to another corner. The number of distinguished points belonging to any corner of R_W is 1/4 of distinguished points belonging to T. We also change the step size of precomputed shifts "O" to prohibit overjumping the searching area, this problem is the point of discussion in Section 4.

4. Time-memory tradeoff for Gaudry — Schost algorithm

First, we describe the aspects of our implementation. We will use the following notations:

— E is the expected number of distinguished points to be stored, i.e., the expected memory requirement of the algorithm. Note that the actual memory size to store one distinguished point is about $7 \log q + O(1)$ bits. The associated data structure contains 5 long integers of size q, representing a divisor in Mumford coordinates $(u_0, u_1, u_2, v_0, v_1)$, 2 long integers $\sigma_1, \sigma_2 \in R$, encoding a position of the point in a searching rectangle, a hash value of a divisor, that is a 32-bit integer, and a Boolean value indicating Wild or Tame walk;

— θ is the probability for a random point $D \in \{T, W\}$ to be a distinguished point. It is easy to see that the expected length of a chain of a random walk is $U = 1/\theta$. Following Gaudry, we use a 32-bit hash function, that has a weak correlation with the arithmetic properties of a point. We consider a point to be distinguished iff some bits in a hash

value are equal to zero. Thus, the probability of being distinguished can be customized stepwise starting from 1/32, 1/16, 3/32, and so on.

We store all distinguished points in a single array. We do not sort this array directly because the points themselves are rather "heavy" as noted above, and their relocation will lead to suboptimal time. Instead of this, we store an additional vector of pointers. This vector addresses the elements sorting them by hash value. That is why the hash value is stored together with a point itself. As soon as a distinguished point is hit, we save it to the end of the main array, that has $O(1)$ time complexity. Then we find an appropriate position for this point in a sorted list of pointers, which is done in $O(\log E)$ by binary search. Then we insert a new pointer to the list, which also has time complexity $O(1)$.

Proposition 1. The time complexity of our Gaudry — Schost implementation (in average case for a random curve) is given by

$$T = \alpha + \frac{1}{\theta} + E \cdot \log_2 E \text{ (operations in } \mathbb{F}_q), \tag{4}$$

where $\alpha = 2.43(1+\epsilon)\sqrt{|R|}$ for Gaudry — Schost algorithm; $\alpha = 2.38(1+\epsilon)\sqrt{|R_T|}$ for its Galbraith — Ruprai improvement.

Proof. The value α in (4) is an expected number of points in the search space to be enumerated until we find a collision. Evaluation of α relies on the Tame-Wild birthday paradox and can be found in [2, 6]. To start a new random walk, we choose a point uniformly at random, that takes constant time. Each step in a random walk requires one group operation in $Jac(\mathcal{C})$, which can be done in time $O(1)$ of operations in $\mathbb{F}_q$ by applying explicit formulas [8]. There are two kinds of "bad" points that could not be accounted when applying the Tame-Wild birthday paradox:

1) points that give a cycle in a random walk. Once we get a loop, a random walk will never hit a distinguished point and must be aborted. P. C. van Oorschot and M. J. Wiener have shown [9] that if we restrict the maximum length of a chain to $20/\theta$, then the number of such "bad" points is at most $5 \cdot 10^{-8}$;
2) points that lie outside the search space R, R_T or R_W. To reduce the number of such overjumps, we follow Gaudry — Schost [2] and make the precomputed shifts "O" not greater than
$$\ell_2 = \frac{(B_{2,\max} - B_{2,\min})\,\theta}{10}, \quad \ell_1 = \frac{(B_{1,\max} - B_{1,\min})\sqrt{\theta}}{9}$$
for both directions. Thus, the expected length of a chain is one tenth of the search space R in both σ_1 and σ_2 directions. "Bad" points of this type give a correction factor $(1+\epsilon)$ to the Tame-Wild birthday paradox theorem, where ϵ, following [3], is a small factor between 0.02 and 0.04.

The term $1/\theta$ in (4) takes into account the length of the last chain, because a collision may occur at any intermediate point of a walk. However, the walk continues up to the distinguished point. The last term $E \cdot \log_2 E$ is the time wasted on binary search in W or T lists to find a collision. ■

Proposition 2. The memory restriction that gives us a time-memory tradeoff for our implementation is

$$E = \sqrt{\frac{2\alpha \ln 2}{\log(2\alpha \ln 2)}},$$

where α is defined in Proposition 1.

Proof. Rewrite equation (4), assuming $1/\theta$ is the average length of a chain and E is the number of chains constructed:

$$T = T(E) = \alpha + \frac{\alpha}{E} + E \cdot \log_2 E.$$

Find the minimum value of time function $T(E)$:

$$T'(E) = -\frac{\alpha}{E^2} + \log_2 E + \frac{1}{\ln 2} = 0.$$

The only critical point that is a point of local minimum for a function $T(E)$ is

$$E = \frac{\sqrt{D}}{\sqrt{W(D)}} = \sqrt{e^{W(D)}} \approx \sqrt{e^{\log D - \log\log D}} = \sqrt{\frac{D}{\log D}},$$

where $D = 2\alpha \ln 2$ and W is a Lambert W function. ■

5. Implementation and tests

5.1. General description

We present an optimized implementation of Gaudry — Schost and Galbraith — Ruprai point-counting algorithms. This is a fork from Gaudry's NTLJac2 [10] package. This package is implemented in C++ and extends the Number Theory Library (NTL) with special tools for Jacobians of genus 2 hyperelliptic curves. It contains data structures to represent divisors on a curve and its arithmetic. On top of Gaudry's package, we added the efficient data storage for the distinguished points as described in Section 4 and implemented an improvement proposed by Galbraith and Ruprai. Moreover, we made our implementation multithreaded.

5.2. Runs on different curves

First, we collected some statistics to test Gaudry — Schost time complexity and compare it with those of Galbraith — Ruprai improvement. We fixed a field $\mathbb{F}_q$ with $q = 2^{45} + 59$, randomly generated $N = 300$ curves and for each of them computed s_1, s_2 modulo $m = 3 \cdot 5 \cdot 7 \cdot 11 = 1155$. As noted in Remark 1, it is not correct to run an algorithm once to evaluate its runtime. So for each curve, we did $n = 30$ runs and calculated an average runtime and number of stored distinguished points. Remark 1 shows that for $n = 30$ experiments the standard deviation on the number of distinguished points reduces from $0.92\sqrt{M}$ to $\frac{0.92}{\sqrt{n}}\sqrt{M} \approx 0.17\sqrt{M}$. Thus, the number of distinguished points constructed for each curve deviates on average $0.17/2.54 \approx 6.7\,\%$ of the theoretical value. This deviation is still quite significant but allows to select "the best" and "the worst" curve instances.

We did the same experiment for the Gaudry — Schost algorithm and its improvement. All of the $N = 300$ curves are sorted by the quantity of distinguished points constructed on average. The number of a curve tested is placed on the X-axis, whereas the quantity of stored points (Fig. 1, a) or overall elapsed time (Fig. 1, b) is on Y-axis. The expected quantity of distinguished points to be stored is $E = 2282$ for the above input, which agrees with our statistics.

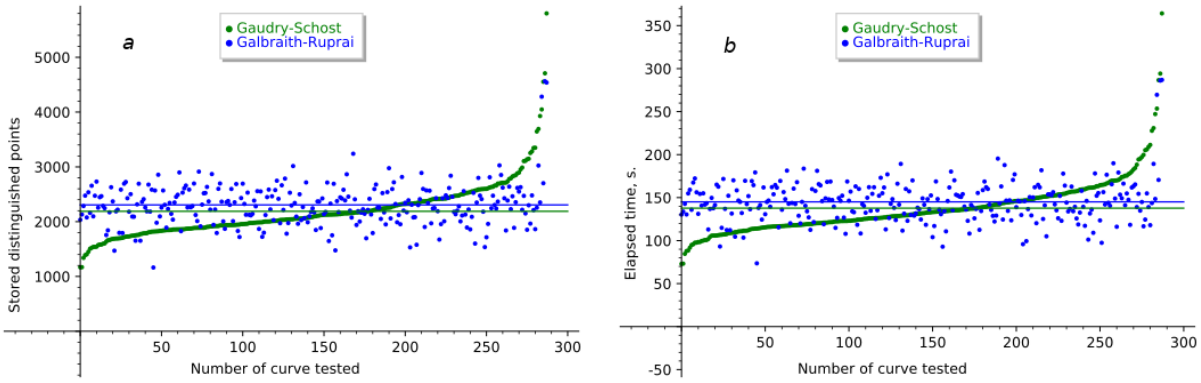


Fig. 1. Gaudry — Schost and Galbraith — Ruprai algorithms. Performance on different curves: a — memory; b — time

5.3. Several runs on the same curve

The aim of running our software on a fixed curve is to test the time-memory tradeoff bound, stated in Proposition 1. We selected $N = 3$ curves from the test above, that are close to the best, worst, and average cases of the Gaudry — Schost algorithm. We modified the parameter E to $E/2$, $E/4$, $2E$, $4E$ and $E = 1000$ as proposed by Gaudry. For each value, we run the software $n = 100$ times. This guarantees the deviation of the experimental number of selected points from the theoretical value by 3.6 % on average. According to proposition 2, the time-memory tradeoff for our input data requires to store $E = 2282$ points for Gaudry — Schost algorithm and $E = 1890$ for its Galbriath — Ruprai improvement (Tables 1 and 2).

All performance tests described above have been executed on Xeon E-2146G 6C 3.50GHz, system RAM available is 16GB. The software is compiled with gcc 9.3.0 compilers under Ubuntu 20.04 operating system.

Table 1

Gaudry — Schost performance on different memory restrictions

		$E = 456$	$E = 1141$	$E = 2282$ (tradeoff)	$E = 4564$	$E = 11410$	$E = 1000$
Curve 1	Dist. points	494	1037	1892	3828	7553	1086
	Time, s.	104.8	109.6	98.9	100.3	98.8	112.8
Curve 120	Dist. points	493	1195	1907	4183	8006	1207
	Time, s.	103.5	124.8	99.1	109.4	106.3	126.7
Curve 300	Dist. points	623	1180	2248	4646	10986	1326
	Time, s.	131.9	123.5	117.2	123.4	144.6	138.6

Table 2

Galbraith — Ruprai improvement on different memory restrictions

		$E = 378$	$E = 945$	$E = 1890$ (tradeoff)	$E = 3780$	$E = 9450$
Curve 1	Dist. points	491	1121	1910	4572	20354
	Time, s.	102.8	117.3	100.3	120.7	136.9
Curve 120	Dist. points	660	1239	2082	4237	19855
	Time, s.	138.0	129.7	109.6	108.8	131.5
Curve 300	Dist. points	635	1140	2217	4113	19233
	Time, s.	133.9	119.9	116.2	112.4	127.6

6. Conclusion

We presented efficient implementations of two BSGS-like point-counting algorithms based on the birthday paradox. A time-memory tradeoff has been provided for both algorithms. It allows us to minimize the runtime by allocating enough memory. We did not test our implementation on cryptographic size input, as we were unable to precompute $\overline{s_1}, \overline{s_2}$ for sufficiently large moduli m. However, we believe our implementation might be useful for computations with any size of Jacobians, in combination with other techniques. The source code of our implementation can be found here: `https://github.com/kn02262/Jac2pc`.

REFERENCES

1. *Matsuo K., Chao J., and Tsujii S.* An improved baby step giant step algorithm for point counting of hyperelliptic curves over finite fields. LNCS, 2002, vol. 2369, pp. 461–474.
2. *Gaudry P. and Schost E.* A low-memory parallel version of Matsuo, Chao and Tsujii's algorithm. LNCS, 2004, vol. 3076, pp. 208–222.
3. *Galbraith S. and Ruprai R.* An improvement to the Gaudry — Schost algorithm for multidimensional discrete logarithm problems. LNCS, 2009, vol. 5921, pp. 368–382.
4. *Cohen H., Frey G., Avanzi R., et al.* Handbook of Elliptic and Hyperelliptic Curve Cryptography. CRC Press, 2005.
5. *Gaudry P. and Schost E.* Genus 2 point counting over prime fields. J. Symbolic Comput., 2012, vol. 47, iss. 4, pp. 368–400.
6. *Ruprai R. S.* Improvements to the Gaudry — Schost Algorithm for Multidimensional Discrete Logarithm Problems and Applications. PhD Thesis, Department of Mathematics, Royal Holloway University of London, 2010. `https://www.math.auckland.ac.nz/~sgal018/Ruprai-thesis.pdf`.
7. *Nishimura K. and Sibuya M.* Probability to meet in the middle. J. Cryptology, 1990, no. 2, pp. 13–22.
8. *Hisil H. and Costello C.* Jacobian coordinates on genus 2 curves. J. Cryptology, 2017, vol. 30, iss. 2, pp. 572–600. `https://doi.org/10.1007/s00145-016-9227-7`.
9. *Van Oorschot P. C. and Wiener M. J.* Parallel collision search with cryptanalytic applications. J. Cryptology, 2013, vol. 12, pp. 1–28.
10. *Gaudry P.* C++ NTLJac2 Library, 2003, `http://www.lix.polytechnique.fr/Labo/Pierrick.Gaudry/NTLJac2`.

СВЕДЕНИЯ ОБ АВТОРАХ

БОБРОВ Владимир Михайлович — аспирант НИЯУ МИФИ, г. Москва.
E-mail: **bvm_15@mail.ru**

ДРЮЧЕНКО Михаил Анатольевич — кандидат технических наук, доцент, доцент кафедры технологий обработки и защиты информации Воронежского государственного университета, г. Воронеж. E-mail: **m_dryuchenko@mail.ru**

КОЛЕСНИКОВ Никита Сергеевич — младший научный сотрудник лаборатории математических методов защиты и обработки информации Балтийского федерального университета им. Иммануила Канта, г. Калининград.
E-mail: **NiKolesnikov1@kantiana.ru**

МАРКЕЛОВА Александра Викторовна — кандидат физико-математических наук, технический директор ООО «НТЦ Альфа-Проект», г. Москва.
E-mail: **a@safe-crypto.ru**

НАЗИНА Светлана Витальевна — студентка кафедры программного обеспечения вычислительной техники и автоматизированных систем Белгородского государственного технологического университета им. В. Г. Шухова, г. Белгород.
E-mail: **lanalana9808@gmail.com**

ПОПКОВ Кирилл Андреевич — кандидат физико-математических наук, научный сотрудник Института прикладной математики им. М. В. Келдыша РАН, г. Москва.
E-mail: **kirill-formulist@mail.ru**

РЫБАЛОВ Александр Николаевич — кандидат физико-математических наук, старший научный сотрудник лаборатории комбинаторных и вычислительных методов алгебры и логики Института математики им. С. Л. Соболева СО РАН, г. Новосибирск.
E-mail: **alexander.rybalov@gmail.com**

РЯЗАНОВ Юрий Дмитриевич — доцент, доцент Белгородского государственного технологического университета им. В. Г. Шухова, г. Белгород.
E-mail: **razanov.yd@bstu.ru**

СИРОТА Александр Анатольевич — доктор технических наук, профессор, заведующий кафедрой технологий обработки и защиты информации Воронежского государственного университета, г. Воронеж. E-mail: **sir@cs.vsu.ru**

СТАЦЕНКО Игорь Викторович — кандидат технических наук, доцент кафедры высшей математики Московского энергетического института, г. Москва.
E-mail: **iwsta@yandex.ru**

ФОМИЧЕВ Владимир Михайлович — доктор физико-математических наук, профессор, профессор Финансового университета при Правительстве Российской Федерации, ведущий научный сотрудник ФИЦ ИУ РАН, г. Москва.
E-mail: **fomichev.2016@yandex.ru**

HUNG Le Xuan — Doctor, Lecturer, HaNoi University for Natural Resources and Environment, Hanoi, Vietnam. E-mail: **lxhung@hunre.edu.vn**

Журнал «Прикладная дискретная математика» входит в перечень ВАК рецензируемых научных изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание учёной степени кандидата и доктора наук по специальностям 01.01.06 — «Математическая логика, алгебра и теория чисел», 01.01.09 — «Дискретная математика и математическая кибернетика», 05.13.11 — «Математическое и программное обеспечение вычислительных машин,комплексов и компьютерных сетей», 05.13.17 — «Теоретические основы информатики», 05.13.19 — «Методы и системы защиты информации, информационная безопасность», а также в перечень журналов, рекомендованных ФУМО ВО ИБ в качестве учебной литературы по специальности «Компьютерная безопасность».

Журнал индексируется в базах данных Web of Science (Emerging Sources Citation Index (ESCI) и Russian Science Citation Index (RSCI)), Scopus, MathSciNet и Zentralblatt MATH.

Журнал «Прикладная дискретная математика» распространяется по подписке; его подписной индекс 38696 в объединённом каталоге «Пресса России». Полнотекстовые электронные версии вышедших номеров журнала доступны на его сайте journals.tsu.ru/pdm и на Общероссийском математическом портале www.mathnet.ru. На сайте журнала можно найти также правила подготовки рукописей статей для публикации в журнале.

Тематика публикаций журнала:

- *Теоретические основы прикладной дискретной математики*
- *Математические методы криптографии*
- *Математические методы стеганографии*
- *Математические основы компьютерной безопасности*
- *Математические основы надёжности вычислительных и управляющих систем*
- *Прикладная теория кодирования*
- *Прикладная теория автоматов*
- *Прикладная теория графов*
- *Логическое проектирование дискретных автоматов*
- *Математические основы информатики и программирования*
- *Вычислительные методы в дискретной математике*
- *Дискретные модели реальных процессов*
- *Математические основы интеллектуальных систем*
- *Исторические очерки по дискретной математике и её приложениям*