

# Оглавление

Введение	3
Основные определения	5
Постановка задачи	9
Результаты	10
Теоремы . . . . .	10
Пример . . . . .	11
Заключение	16
Приложение	17
Список литературы	20

# Введение

Одним из стандартных способов задания функций  $k$ -значной логики являются поляризованные полиномиальные формы (ППФ), которые также называются обобщенными формами Рида-Мюллера, или каноническими поляризованными полиномами. В ППФ каждая переменная имеет определенную поляризацию. Длиной полиномиальной формы называется число попарно различных слагаемых в ней. Длиной функции  $f$  в классе ППФ называется наименьшая длина среди длин всех поляризованных полиномиальных форм, реализующих  $f$ . Функция Шеннона  $L_k^K(n)$  длины определяется как наибольшая длина среди всех функций  $k$ -значной логики в классе  $K$  от  $n$  переменных, если  $K$  опущено, то подразумевается класс ППФ. Практическое применение ППФ нашли при построении программируемых логических матриц (ПЛМ) [1, 2], сложность ПЛМ напрямую зависит от длины ППФ, по которой она построена. Поэтому в ряде работ исследуется сложность ППФ различных функций [3–9].

В 1993 В. П. Супрун [3] получил первые оценки функции Шеннона для функций алгебры логики :

$$L_2(n) \geq C_n^{\lfloor \frac{n}{2} \rfloor},$$
$$L_2(n) < 3 \cdot 2^{n-1},$$

где  $[a]$  обозначает целую часть  $a$ .

Точное значение функции Шеннона для функций алгебры логики в 1995 г. было найдено Н. А. Перязевым [4] :

$$L_2(n) = \left\lceil \frac{2^{n+1}}{3} \right\rceil.$$

Функции  $k$ -значных логик являются естественным обобщением функций алгебры логики. Для функций  $k$ -значной логики верхняя оценка функции Шеннона была получена в 2002 г. С. Н. Селезневой [5] :

$$L_k(n) < \frac{k(k-1)}{k(k-1)+1} k^n,$$

в 2015 году она была улучшена [12]

$$L_k(n) < \frac{k(k-1)-1}{k(k-1)} k^n.$$

При построении ПЛМ рассматривают и другие полиномиальные формы. Например класс обобщенных полиномиальных форм. В классе обобщенных полиномиальных форм, в отличие от класса поляризованных полиномиальных форм, переменные могут иметь различную поляризацию в разных слагаемых. В статье К. Д. Кириченко [6], опубликованной в 2005 г., был предложен метод построения обобщенных полиномиальных форм из которого следует  $L_2^{\text{О.П.}}(n) = O(\frac{2^n}{n})$ .

В работах [7, 10] получено, что  $L_k^{\text{О.П.}}(n) = O(\frac{k^n}{n})$ .

В 2012 г. Н. К. Маркеловым была получена нижняя оценка функции Шеннона для функции трехзначной логики в классе поляризованных полиномов [8]:

$$L_3(n) \geq \left\lceil \frac{3}{4} 3^n \right\rceil,$$

в [11] эта оценка была достигнута на симметрических функциях.

## Основные определения

Пусть  $k \geq 2$  – натуральное число,  $E_k = \{0, 1, \dots, k-1\}$ . Весом набора  $\alpha = (a_1, \dots, a_n) \in E_k^n$  назовем число  $|\alpha| = \sum_{i=1}^n a_i$ . Моном  $\prod_{a_i \neq 0} x_i^{a_i}$  назовем соответствующим набору  $\alpha = (a_1, \dots, a_n) \in E_k^n$  и обозначим через  $K_\alpha$ . По определению положим, что константа 1 соответствует набору из всех нулей. Функцией  $k$ -значной логики называется отображение  $f^{(n)} : E_k^n \rightarrow E_k$ ,  $n = 0, 1, \dots$ . Множество всех функций  $k$ -значной логики обозначим через  $P_k$ , множество всех функций  $k$ -значной логики, зависящих от переменных  $x_1, \dots, x_n$ , обозначим через  $P_k^n$ . Функция  $j_i(x) = \begin{cases} 1, & \text{если } x = i; \\ 0, & \text{если } x \neq i. \end{cases}$

Если  $k$  – простое число, то каждая функция  $k$ -значной логики  $f(x_1, \dots, x_n)$  может быть однозначно задана формулой вида

$$f(x_1, \dots, x_n) = \sum_{\alpha \in E_k^n : c_f(\alpha) \neq 0} c_f(\alpha) K_\alpha,$$

где  $c_f(\alpha) \in E_k$  – коэффициенты,  $\alpha \in E_k^n$ , и операции сложения и умножения рассматриваются по модулю  $k$ . Это представление функций  $k$ -значной логики называется ее полиномом по модулю  $k$ . При простых  $k$  однозначно определенный полином по модулю  $k$  для функции  $k$ -значной логики  $f$  будем обозначать через  $P(f)$ .

Определим поляризованные полиномиальные формы по модулю  $k$ . Поляризованной переменной  $x_i$  с поляризацией  $d$ ,  $d \in E_k$ , назовем выражение вида  $(x_i + d)$ . Поляризованным мономом по вектору поляризации  $\delta$ ,  $\delta = (d_1, \dots, d_n) \in E_k^n$ , назовем произведение вида  $(x_{i_1} + d_{i_1})^{m_1} \cdots (x_{i_r} + d_{i_r})^{m_r}$ , где  $1 \leq i_1 < \dots < i_r \leq n$ , и  $1 \leq m_1, \dots, m_r \leq k-1$ . Обычный моном является мономом, поляризованным по вектору  $\tilde{0} = (0, \dots, 0) \in E_k^n$ .

Выражение вида  $\sum_{i=1}^l c_i \cdot K_i$ , где  $c_i \in E_k \setminus \{0\}$  – коэффициенты,  $K_i$  – попарно различные мономы, поляризованные по вектору  $\delta = (d_1, \dots, d_n) \in E_k^n$ ,  $i = 1, \dots, l$ , назовем поляризованной полиномиальной нормальной формой (ППФ) по вектору поляризации  $\delta$ . Мы будем считать, что константа 0 является ППФ по произвольному вектору поляризации. Заметим, что при простых  $k$  для каждого вектора поляризации каждую функцию  $k$ -значной логики можно однозначно представить ППФ по этому вектору поляризации [5]. При простых  $k$  однозначно определенную ППФ по вектору поляризации  $\delta \in E_k^n$  для функции  $f \in P_k^n$  будем обозначать через  $P^\delta(f)$ .

Длиной  $l(p)$  ППФ  $p$  назовем число попарно различных слагаемых в этой ППФ. Положим, что  $l(0) = 0$ . При простых  $k$  длиной функции  $k$ -значной логики в классе ППФ называется величина  $l^{\text{ППФ}}(f) = \min_{\delta \in E_k^n} l(P^\delta(f))$ .

Функция  $k$ -значной логики  $f(x_1, \dots, x_n)$  называется симметрической, если

$$f(\pi(x_1), \dots, \pi(x_n)) = f(x_1, \dots, x_n)$$

для произвольной перестановки  $\pi$  на множестве переменных  $\{x_1, \dots, x_n\}$ . Множество всех симметрических функций  $k$ -значной логики обозначим через  $S_k$ . Симметрическая функция  $f(x_1, \dots, x_n)$  называется периодической с периодом  $\tau = (\tau_0 \tau_1 \dots \tau_{T-1}) \in E_k^T$ , если  $f(\alpha) = \tau_j$  при  $|\alpha| = j \pmod{T}$  для каждого набора  $\alpha \in E_k^n$ . При этом число  $T$  называется длиной периода. Периодическую функцию  $k$ -значной логики  $f(x_1, \dots, x_n)$  с периодом  $\tau = (\tau_0 \tau_1 \dots \tau_{T-1}) \in E_k^T$  будем обозначать через  $f_{(\tau_0 \tau_1 \dots \tau_{T-1})}^{(n)}$ . Понятно, что такое обозначение полностью определяет эту функцию.

Введем функцию  $\text{rol}_i(\alpha) \in E_k^n \times E_k \rightarrow E_k^n$ , производящую циклический сдвиг вектора  $\alpha$  влево. Пусть  $\alpha = (a_1, \dots, a_n)$ , тогда  $\text{rol}_i(\alpha) = (a_{(1+i) \bmod T}, \dots, a_{(n+i) \bmod T})$ , где  $T$  – длина вектора  $\alpha$ .

Функция  $\text{rol}_i(\alpha) \in E_k^n \times E_k \rightarrow E_k^n$  производит циклический сдвиг вектора  $\alpha$  вправо. Пусть  $T$  – длина вектора  $\alpha$ , тогда  $\text{rol}_i(\alpha) = \text{rol}_{(-i) \bmod T}(\alpha)$ .

В [9] дается описание быстрого алгоритма построения поляризованного полинома по заданному вектору функции и поляризации  $d$ . Данный алгоритм можно использовать для получения периодов, участвующих в разложении периодической симметрической функции  $f^{(n+1)}$  с периодом  $\tau$ .

$$A_d \cdot F(\tau) = \begin{pmatrix} \tau_0 \\ \tau_1 \\ \vdots \\ \tau_{k-1} \end{pmatrix}, \text{ где}$$

$$f_{\tau}^{(n+1)} = f_{\tau_{k-1}}^{(n)} (x_{n+1} + d)^{k-1} + f_{\tau_{k-2}}^{(n)} (x_{n+1} + d)^{k-2} + \dots + f_{\tau_0}^{(n)} \quad (1)$$

$$A = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & k-1 & -(2^{k-2}) \bmod k & \dots & -((k-1)^{k-2}) \bmod k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k-1 & k-1 & k-1 & \dots & k-1 \end{pmatrix}$$

$A_d$  получается из матрицы  $A$  циклическим сдвигом столбцов  $A$  влево на  $d$  столбцов или построчным применением функции  $\text{rol}_d$  к матрице  $A$ .

$$F(\tau) = \begin{pmatrix} \text{rol}_0(\tau) \\ \text{rol}_1(\tau) \\ \vdots \\ \text{rol}_{k-1}(\tau) \end{pmatrix}$$

$F(\tau)$  – матрица, в которой в  $i$ -той строке стоит сдвиг влево периода  $\tau$  на  $i$  элементов.

Например в трехзначной логике для функции с периодом  $[1,1,2,2]$  получим

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 1 \\ 2 & 2 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 & 2 \\ 1 & 0 & 2 & 0 \\ 2 & 1 & 1 & 2 \end{pmatrix}$$

Пусть матрица  $A$  состоит из строк  $a_i, a'_i$  – строка, полученная из  $a_i$ , добавлением нулей справа так, что длина строки  $a'_i$  равна  $T$ . Тогда матрица  $A^{(i)}$  состоит из  $T$  циклических сдвигов вправо строки  $a'_i$ :

$$A^{(i)} = \begin{pmatrix} \text{ror}_0(a'_i) \\ \text{ror}_1(a'_i) \\ \vdots \\ \text{ror}_{T-1}(a'_i) \end{pmatrix}$$

Матрица  $A^{(0)}$  – единичная матрица размера  $T \times T$ . Для всех  $\tau_i$  из формулы (1) верно  $\tau_i = A^{(i)} \cdot \tau$ . Период  $\tau$  называется вырожденным, если  $l(f_\tau^{(n)}) = \bar{o}(k^n)$  при  $n \rightarrow \infty$ . Период  $\tau$  называется сложным, если  $\exists n_0, c > 0 : \forall n \geq n_0 \Rightarrow l(f_\tau^{(n)}) \geq c \cdot k^n$ . Обозначим через  $\tilde{0}$  – нулевой период.

Введем индуктивно множество периодов, порожденных периодом  $\tau$  за  $i$  шагов –  $\Pi_i(\tau)$ . В данном случае матрицы рассматриваются, как множества строк.

$$\Pi_0(\tau) = \{\tau\}$$

$$\Pi_1(\tau) = \bigcup_{d \in E_k} A_d \cdot F(\tau)$$

$$\Pi_{i+1}(\tau) = \bigcup_{\theta \in \Pi_i(\tau)} \Pi_1(\theta)$$

Обозначим через  $\Pi(\tau)$  такое  $\Pi_i(\tau)$ , что  $\Pi_i(\tau) = \Pi_{i+1}(\tau)$ .

## Постановка задачи

1. Нахождение периодов, являющихся сложными, в  $k$ -значных логиках.
2. Отыскание критериев сложности периодов.
3. Получение нижних оценок длин некоторых симметрических функций  $k$ -значных логик в классе ППФ.



# Результаты

## Теоремы

**Теорема 1.** Если  $\tilde{0} \notin \Pi(\tau)$ , то  $\tau$  сложный период.

*Доказательство.* Теорема 1 является следствием теоремы 2. □

**Теорема 2.** Если  $\tilde{0} \notin \Pi(\tau)$ , то  $\forall \sigma \in \Pi(\tau) \forall n \geq n_0 \Rightarrow l(f_\sigma^{(n)}) \geq c \cdot k^n$ , где

$T$  – длина периода  $\tau$

$k^{n_0} > T > k^{n_0-1}$ , нетрудно заметить, что  $n_0 = \lceil \log_k T \rceil$

$$c = \frac{1}{k^{n_0}}$$

*Доказательство.* Доказательство проведем индукцией по  $n$ .

Базис индукции:

$$l(f_\sigma^{(n_0)}) \geq 1 = \frac{1}{k^{n_0}} \cdot k^{n_0}$$

Индуктивный переход:

Предположим, что теорема верна для  $n \geq n_0$ , докажем ее для  $n + 1$ . Для любой поляризации  $d \in E_k$  функция  $f_\sigma^{(n+1)}$  выражается следующим полиномом:

$$f_\sigma^{(n+1)} = f_{\sigma_{k-1}}^{(n)} \cdot (x_{n+1} + d)^{k-1} + \dots + f_{\sigma_1}^{(n)} \cdot (x_{n+1} + d) + f_{\sigma_0}^{(n)}, \text{ где}$$

функции  $f_{\sigma_i}^{(n)}$  могут быть различными при разных поляризациях  $d$ .

По предположению индукции  $\forall i \in E_k \Rightarrow l(f_{\sigma_i}^{(n)}) \geq \frac{1}{k^{n_0}} \cdot k^n$ , следовательно  $l(f_{\sigma_i}^{(n+1)}) \geq k \cdot \frac{1}{k^{n_0}} \cdot k^n = \frac{1}{k^{n_0}} \cdot k^{n+1}$  □

**Утверждение.** Все периоды пятизначной логики длины 3, отличные от  $[0,0,0]$ ,  $[1,1,1]$ ,  $[2,2,2]$ ,  $[3,3,3]$ ,  $[4,4,4]$  являются сложными.

## Пример

Рассмотрим функцию  $f = f_{(0,1,4)}$ , эта функция порождает семейство функций  $F$ :

$$f = f_{(0,1,4)}, g = f_{(1,0,4)}, h = f_{(1,1,3)}, p = f_{(1,2,2)}, s = f_{(1,3,1)}, t = f_{(1,4,0)}$$

Для этих функций верны следующие разложения.

$$f = f_{(0,1,4)}$$

$$f_{n+1} = x_{n+1}^4 3f_n + x_{n+1}^3 t_n + x_{n+1}^2 3p_n + x_{n+1} 2g_n + f_n$$

$$f_{n+1} = (x_{n+1} + 1)^4 3f_n + (x_{n+1} + 1)^3 p_n + (x_{n+1} + 1)^2 2f_n + (x_{n+1} + 1) 4p_n + f_n$$

$$f_{n+1} = (x_{n+1} + 2)^4 3f_n + (x_{n+1} + 2)^3 g_n + (x_{n+1} + 2)^2 2p_n + (x_{n+1} + 2) 2t_n + f_n$$

$$f_{n+1} = (x_{n+1} + 3)^4 3f_n + (x_{n+1} + 3)^3 s_n + (x_{n+1} + 3)^2 4s_n + (x_{n+1} + 3) s_n + 4g_n$$

$$f_{n+1} = (x_{n+1} + 4)^4 3f_n + (x_{n+1} + 4)^3 h_n + (x_{n+1} + 4)^2 h_n + (x_{n+1} + 4) h_n + t_n$$

$$g = f_{(1,0,4)}$$

$$g_{n+1} = x_{n+1}^4 3g_n + x_{n+1}^3 4f_n + x_{n+1}^2 4s_n + x_{n+1} 2t_n + g_n$$

$$g_{n+1} = (x_{n+1} + 1)^4 3g_n + (x_{n+1} + 1)^3 3s_n + (x_{n+1} + 1)^2 2g_n + (x_{n+1} + 1) 2s_n + g_n$$

$$g_{n+1} = (x_{n+1} + 2)^4 3g_n + (x_{n+1} + 2)^3 t_n + (x_{n+1} + 2)^2 s_n + (x_{n+1} + 2) 3f_n + g_n$$

$$g_{n+1} = (x_{n+1} + 3)^4 3g_n + (x_{n+1} + 3)^3 4h_n + (x_{n+1} + 3)^2 h_n + (x_{n+1} + 3) 4h_n + 4t_n$$

$$g_{n+1} = (x_{n+1} + 4)^4 3g_n + (x_{n+1} + 4)^3 2p_n + (x_{n+1} + 4)^2 2p_n + (x_{n+1} + 4) 2p_n + 4f_n$$

$$h = f_{(1,1,3)}$$

$$h_{n+1} = x_{n+1}^4 3h_n + x_{n+1}^3 s_n + x_{n+1}^2 2t_n + x_{n+1} 4p_n + h_n$$

$$h_{n+1} = (x_{n+1} + 1)^4 3h_n + (x_{n+1} + 1)^3 4t_n + (x_{n+1} + 1)^2 2h_n + (x_{n+1} + 1)t_n + h_n$$

$$h_{n+1} = (x_{n+1} + 2)^4 3h_n + (x_{n+1} + 2)^3 2p_n + (x_{n+1} + 2)^2 3t_n + (x_{n+1} + 2)2s_n + h_n$$

$$h_{n+1} = (x_{n+1} + 3)^4 3h_n + (x_{n+1} + 3)^3 2f_n + (x_{n+1} + 3)^2 3f_n + (x_{n+1} + 3)2f_n + 3p_n$$

$$h_{n+1} = (x_{n+1} + 4)^4 3h_n + (x_{n+1} + 4)^3 3g_n + (x_{n+1} + 4)^2 3g_n + (x_{n+1} + 4)3g_n + s_n$$

$$p = f_{(1,2,2)}$$

$$p_{n+1} = x_{n+1}^4 3p_n + x_{n+1}^3 2h_n + x_{n+1}^2 4f_n + x_{n+1} s_n + p_n$$

$$p_{n+1} = (x_{n+1} + 1)^4 3p_n + (x_{n+1} + 1)^3 3f_n + (x_{n+1} + 1)^2 2p_n + (x_{n+1} + 1)2f_n + p_n$$

$$p_{n+1} = (x_{n+1} + 2)^4 3p_n + (x_{n+1} + 2)^3 3s_n + (x_{n+1} + 2)^2 f_n + (x_{n+1} + 2)4h_n + p_n$$

$$p_{n+1} = (x_{n+1} + 3)^4 3p_n + (x_{n+1} + 3)^3 g_n + (x_{n+1} + 3)^2 4g_n + (x_{n+1} + 3)g_n + 2s_n$$

$$p_{n+1} = (x_{n+1} + 4)^4 3p_n + (x_{n+1} + 4)^3 4t_n + (x_{n+1} + 4)^2 4t_n + (x_{n+1} + 4)4t_n + 2h_n$$

$$s = f_{(1,3,1)}$$

$$s_{n+1} = x_{n+1}^4 3s_n + x_{n+1}^3 3p_n + x_{n+1}^2 3g_n + x_{n+1} 3h_n + s_n$$

$$s_{n+1} = (x_{n+1} + 1)^4 3s_n + (x_{n+1} + 1)^3 g_n + (x_{n+1} + 1)^2 2s_n + (x_{n+1} + 1)4g_n + s_n$$

$$s_{n+1} = (x_{n+1} + 2)^4 3s_n + (x_{n+1} + 2)^3 4h_n + (x_{n+1} + 2)^2 2g_n + (x_{n+1} + 2)p_n + s_n$$

$$s_{n+1} = (x_{n+1} + 3)^4 3s_n + (x_{n+1} + 3)^3 2t_n + (x_{n+1} + 3)^2 3t_n + (x_{n+1} + 3)2t_n + h_n$$

$$s_{n+1} = (x_{n+1} + 4)^4 3s_n + (x_{n+1} + 4)^3 2f_n + (x_{n+1} + 4)^2 2f_n + (x_{n+1} + 4)2f_n + 3p_n$$

$$t = f_{(1,4,0)}$$

$$t_{n+1} = x_{n+1}^4 3t_n + x_{n+1}^3 4g_n + x_{n+1}^2 h_n + x_{n+1} 3f_n + t_n$$

$$t_{n+1} = (x_{n+1} + 1)^4 3t_n + (x_{n+1} + 1)^3 2h_n + (x_{n+1} + 1)^2 2t_n + (x_{n+1} + 1) 3h_n + t_n$$

$$t_{n+1} = (x_{n+1} + 2)^4 3t_n + (x_{n+1} + 2)^3 4f_n + (x_{n+1} + 2)^2 4h_n + (x_{n+1} + 2) 3g_n + t_n$$

$$t_{n+1} = (x_{n+1} + 3)^4 3t_n + (x_{n+1} + 3)^3 3p_n + (x_{n+1} + 3)^2 2p_n + (x_{n+1} + 3) 3p_n + f_n$$

$$t_{n+1} = (x_{n+1} + 4)^4 3t_n + (x_{n+1} + 4)^3 s_n + (x_{n+1} + 4)^2 s_n + (x_{n+1} + 4) s_n + 4g_n$$

Функции одной переменной.

$$f = f_{(0,1,4)}$$

$$f_1 = 3x + 2x^2 + 2x^3 + 4x^4$$

$$f_1 = 1 + 4(x + 1) + (x + 1)^3 + 4(x + 1)^4$$

$$f_1 = (x + 2) + (x + 2)^2 + 4(x + 2)^4$$

$$f_1 = 4 + 3(x + 3) + 4(x + 3)^3 + 4(x + 3)^4$$

$$f_1 = 1 + 4(x + 4) + 2(x + 4)^2 + 3(x + 4)^3 + 4(x + 4)^4$$

$$g = f_{(1,0,4)}$$

$$g_1 = 1 + x + 4x^3 + 4x^4$$

$$g_1 = 2(x + 1) + 2(x + 1)^2 + 3(x + 1)^3 + 4(x + 1)^4$$

$$g_1 = 1 + (x + 2) + 2(x + 2)^2 + 2(x + 2)^3 + 4(x + 2)^4$$

$$g_1 = 4 + 2(x + 3) + (x + 3)^3 + 4(x + 3)^4$$

$$g_1 = 4(x + 4) + (x + 4)^2 + 4(x + 4)^4$$

$$h = f_{(1,1,3)}$$

$$h_1 = 1 + 4x + 2x^2 + x^3 + 3x^4$$

$$h_1 = 1 + (x + 1) + 2(x + 1)^2 + 4(x + 1)^3 + 3(x + 1)^4$$

$$h_1 = 1 + 2(x + 2) + 3(x + 2)^2 + 2(x + 2)^3 + 3(x + 2)^4$$

$$h_1 = 3 + 3(x + 3)^4$$

$$h_1 = 1 + 3(x + 4) + 3(x + 4)^2 + 3(x + 4)^3 + 3(x + 4)^4$$

$$p = f_{(1,2,2)}$$

$$p_1 = 1 + 2x + 4x^2 + 3x^3 + 2x^4$$

$$p_1 = 2 + 2(x + 1)^2 + 2(x + 1)^4$$

$$p_1 = 1 + 3(x + 2) + 4(x + 2)^2 + 2(x + 2)^3 + 2(x + 2)^4$$

$$p_1 = 2 + 3(x + 3) + 4(x + 3)^3 + 2(x + 3)^4$$

$$p_1 = 2 + 2(x + 4) + (x + 4)^3 + 2(x + 4)^4$$

$$s = f_{(1,3,1)}$$

$$s_1 = 1 + x^2 + x^4$$

$$s_1 = 3 + 4(x + 1) + 2(x + 1)^2 + (x + 1)^3 + (x + 1)^4$$

$$s_1 = 1 + 4(x + 2) + 2(x + 2)^3 + (x + 2)^4$$

$$s_1 = 1 + (x + 3) + 3(x + 3)^3 + (x + 3)^4$$

$$s_1 = 3 + (x + 4) + 2(x + 4)^2 + 4(x + 4)^3 + (x + 4)^4$$

$$t = f_{(1,4,0)}$$

$$t_1 = 1 + 3x + 3x^2 + 2x^3$$

$$t_1 = 4 + 3(x + 1) + 2(x + 1)^2 + 2(x + 1)^3$$

$$t_1 = 1 + (x + 2)^2 + 2(x + 2)^3$$

$$t_1 = 4(x + 3) + 2(x + 3)^3$$

$$t_1 = 4 + 4(x + 4)^2 + 2(x + 4)^3$$

В следующей таблице содержатся длины функций одной переменной.

	0	1	2	3	4
$f_1$	4	4	3	4	5
$g_1$	4	4	5	4	3
$h_1$	5	5	5	2	5
$p_1$	5	3	5	4	4
$s_1$	3	5	4	4	5
$t_1$	4	4	3	2	3

**Теорема 3.** Для любой функции  $f_\tau^n \in F$  верно:  $l(f_\tau^n) \geq \frac{2}{5} \cdot 5^n$ .

*Доказательство.* Доказательство аналогично доказательству теоремы 2, за исключением того, что, длина полиномов всех функций одной переменной из  $F$  при любой поляризации не меньше 2.  $\square$

## Заключение

В работе получены следующие результаты:

1. Доказана теорема о нижней оценки длины функций, задаваемых сложными периодами.
2. Получено достаточное условие сложности периодов.
3. Написана программа для работы с периодами в которой реализованы функции:
  - Проверка периода на сложность
  - Построение периодов, порожденных данным
  - Построение специальных матриц, описанных в основных определениях, и функций для работы с ними.
  - Вывод полиномов функций по векторам периодов.
4. С помощью этой программы были получены все сложные периоды пятизначной логики длины три.
5. Был рассмотрен класс функций пятизначной логики и для него доказана нижняя оценка.

# Приложение

Сдесь приведен код основных функций, используемых в программе.

```
k :: Int
k = 5

rol :: [a] -> Int -> [a]
rol xs i = let (l,r) = splitAt i xs in r ++ l

ror :: [a] -> Int -> [a]
ror xs i = let (l,r) = splitAt (length xs - i) xs in r ++ l

step n v = iterate iter [v] !! n where
  iter vs = nub $
    concatMap (\v->concatMap (\d->toLists $ modkM $ aPolar d * fromPeriod v) [0..k-1]) vs

invert' k 1 = 1
invert' k p = (n * k + 1) 'div' p
  where n = p - invert' p (k 'mod' p)

invert = invert' k

allVectors n = sequence $ take (fromIntegral n) $ repeat [0..k-1]

allNormalVectors n = map (1:) $ allVectors (n-1)

normalize :: [Int] -> [Int]
normalize [] = error "Empty list"
normalize xs = normalize' xs xs where
  normalize' [] ys      = error "Zero list"
  normalize' (x:xs) ys | x == 0    = normalize' xs ys
                      | otherwise = map (\n -> n * invert x 'mod' k) ys

reverseMatrix = fromLists . reverse . toLists

modkM = fmap ('mod' k)

a_ij = uncurry a_ij' where
```



```

a_ij' i j | i == 1 && j == 1 = 1
          | i == k           = (-1) 'mod' k
          | i == 1 || j == 1 = 0
          | otherwise = (-(j-1)^((k-1-(i-1)) 'mod' k)) 'mod' k

a = matrix k k a_ij

aPolar d = fromLists $ Data.List.transpose $
  rol (Data.List.transpose $ toLists a) d

fromPeriod xs = fromLists $ map (rol xs) [0..k-1]

polarCoeffs d xs = reverse $ toLists $ modkM $ aPolar d * fromPeriod xs

allCoeffs xs = concatMap (\d -> polarCoeffs d xs) [0..k-1]

setSort = compress . sort where
  compress [] = []
  compress (x:xs) = x : (compress $ dropWhile (== x) xs)

makeAllFamily ps = makeAllFamily' [normalize ps] $ setSort $
  map normalize $ allCoeffs ps where
    makeAllFamily' xss yss | xss == yss = yss
                          | otherwise = makeAllFamily' (insert zs xss) zss where
                                zs = normalize $ firstNotIn yss xss
                                zss = setSort $ yss ++ (map normalize $ allCoeffs zs)
  firstNotIn (y:ys) (x:xs) | y < x = y
                          | y > x = firstNotIn (y:ys) xs
                          | y == x = firstNotIn ys xs
  firstNotIn (y:ys) [] = y

singular [] = putStr ""
singular (x:xs) = do
  result <- try (evaluate (makeAllFamily x)) :: IO (Either SomeException [[Int]])
  case result of
    Left ex -> putStrLn $ "Singular " ++ show x
    Right val -> putStrLn $ "Nonsingular " ++ show x ++ " " ++ (show $ length val)
  singular xs

```

## Список литературы

1. Угрюмов Е. П. Цифровая схемотехника. СПб.: БХВ-Петербург, 2004.
2. Sasao T., Besslich P. On the complexity of mod-2 sum PLA's // IEEE Trans.on Comput. 39. N 2. 1990. P. 262–266.
3. Супрун В. П. Сложность булевых функций в классе канонических поляризованных полиномов // Дискретная математика. 5. №2. 1993. С. 111–115.
4. Перязев Н. А. Сложность булевых функций в классе полиномиальных поляризованных форм // Алгебра и логика. 34. №3. 1995. С. 323–326.
5. Селезнева С. Н. О сложности представления функций многозначных логик поляризованными полиномами. Дискретная математика. 14. №2. 2002. С. 48–53.
6. Кириченко К. Д. Верхняя оценка сложности полиномиальных нормальных форм булевых функций // Дискретная математика. 17. №3. 2005. С. 80–88.
7. Селезнева С. Н. Дайняк А. Б. О сложности обобщенных полиномов  $k$ -значных функций // Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. №3. 2008. С. 34–39.
8. Маркелов Н. К. Нижняя оценка сложности функций трехзначной логики в классе поляризованных полиномов // Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. №3. 2012. С. 40–45.
9. Селезнева С. Н. Маркелов Н. К. Быстрый алгоритм построения векторов коэффициентов поляризованных полиномов  $k$ -значных функций // Ученые записки Казанского университета. Серия Физико-математические науки. 2009. 151. №2 С. 147–151.
10. Башов М. А., Селезнев С. Н. О длине функций  $k$ -значной логики в классе полиномиальных нормальных форм по модулю  $k$  // Дискретная математика. – 2014. – Т. 26, вып. 3. – С. 3–9.

11. Селезнева С.Н. Сложность систем функций алгебры логики и систем функций трехзначной логики в классах поляризованных полиномиальных форм // Дискретная математика. – 2015. – Т. 27, вып. 1. – С. 111 – 122.
12. Балюк А.С., Янушковский Г.В. Верхние оценки функций над конечными полями в некоторых классах кронекеровых форм // Известия Иркутского государственного университета. Серия: Математика. – 2015. – Т.14. – С. 3-17.