



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет вычислительной математики и кибернетики
Кафедра математической кибернетики

Гордеев Михаил Михайлович

**О сложности систем функций некоторых
многозначных логик в классе
поляризованных полиномов**

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Научный руководитель:

д.ф-м.н., доцент
С. Н. Селезнева

Москва, 2016

Оглавление

Введение	3
Основные определения	5
Постановка задачи	7
Результаты	8
Функции пятизначной логики	8
Функции одной переменной	8
Выражение через функции	10
Разложение функций	11
Функции одной переменной	12
Описание программы	14
Заключение	15
Приложение	16
Список литературы	20

Введение

В настоящее время активно исследуются представления дискретных функций над конечными полями. Одним из стандартных способов задания функций k -значной логики являются поляризованные полиномиальные формы (ППФ), которые также называются обобщенными формами Риды-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 [3–5, 7–10].

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 [13]

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

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 [7], 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 [8, 11] 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 [9]:

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

в [12] 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_α . 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 δ , $\delta = (d_1, \dots, d_n) \in E_k^n$, 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 δ . 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. По-

ложим, что $l(0) = 0$. При простых k длиной функции k -значной логики в классе ППФ называется величина $l^{\text{ППФ}}(f) = \min_{\delta \in E_k^n} l(P^\delta(f))$.

Сложностью системы ППФ, имеющих один и тот же вектор поляризации, называется число попарно различных слагаемых, встречающихся во всех этих ППФ. При простых k сложностью $L_k^{\text{ППФ}}(F)$ системы функций k -значной логики $F = \{f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)\}$ в классе ППФ называется минимальная сложность среди всех таких систем ППФ $\{p_1, \dots, p_m\}$, что все ППФ p_1, \dots, p_m имеют один и тот же вектор поляризации и ППФ p_j реализует функцию $f_j, j = 1, \dots, m$. Понятно, что для произвольной системы функций k -значной логики $F = \{f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)\}$ верна оценка $L_k^{\text{ППФ}}(F) \leq k^n$. Пусть k – простое число, и $A_k \subseteq P_k$, а $A_k^n = A_k \cap P_k^n$. Введем функцию Шеннона $L_{A_k}^{\text{ППФ}}(m, n)$ сложности систем функций k -значной логики, принадлежащих множеству A , в классе ППФ:

$$L_{A_k}^{\text{ППФ}}(m, n) = \max_{B \subseteq A_k^n, |B|=m} L_k^{\text{ППФ}}(B).$$

Если $A_k = P_k$, то функцию Шеннона будем обозначать через $L_k^{\text{ППФ}}(m, n)$.

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 π 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.

В данной работе будут рассмотрены функции пятизначной логики и многозначных логик для любого простого k .

Постановка задачи

1. Изучение систем функций в k -значных логиках при небольших (до 20) простых k с целью отыскание сложных систем.
2. Нахождение систем функций k -значных логик, являющихся сложными при любом простом k .
3. Написание программы (библиотеки) для работы с поляризованными полиномами.

Результаты

Функции пятизначной логики

Сначала были рассмотрены функции пяти? и семизначной логики. В ходе выполнения данной работы было написано несколько программ, одна из которых осуществляет поиск сложных систем функций методом ветвей и границ. Таким образом было найдено несколько сложных систем сначала пятизначной, а затем и семизначной логики.

После этого нужно было найти сложные системы функций одной переменной. Сложность функций k -значных логик в классе поляризованных полиномов была рассмотрена в работе [6]. В данной работе были рассмотрены аналогичные функции.

Функции одной переменной

Рассмотрим следующие функции пятизначной логики: $f = (1,4,4,4,3)$, $g = (3,4,4,4,1)$ и их линейные комбинации. Выпишем для этих функций поляризованные полиномы по всем поляризациям от 0 до 4.

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

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

$$f + g = 3x^4 + x^3 + 4x^2 + x + 4$$

$$f + 2g = 2x^4 + 3x^3 + 2x^2 + 3x + 2$$

$$f + 3g = x^4$$

$$f + 4g = 2x^3 + 3x^2 + 2x + 3$$

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

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

$$f + g = 3(x+1)^4 + 4(x+1)^3 + 4(x+1)^2 + 4(x+1) + 4$$

$$f + 2g = 2(x+1)^4$$

$$f + 3g = (x+1)^4 + (x+1)^3 + (x+1)^2 + (x+1) + 1$$

$$f + 4g = 2(x+1)^3 + 2(x+1)^2 + 2(x+1) + 2$$

$$\begin{aligned}
f &= 4(x+2)^4 + 2(x+2)^3 + 3(x+2)^2 + 4 \\
g &= 4(x+2)^4 + 2(x+2)^2 + (x+2) + 4 \\
f+g &= 3(x+2)^4 + 2(x+2)^3 + (x+2) + 3 \\
f+2g &= 2(x+2)^4 + 2(x+2)^3 + 2(x+2)^2 + 2(x+2) + 2 \\
f+3g &= (x+2)^4 + 2(x+2)^3 + 4(x+2)^2 + 3(x+2) + 1 \\
f+4g &= 2(x+2)^3 + (x+2)^2 + 4(x+2)
\end{aligned}$$

$$\begin{aligned}
f &= 4(x+3)^4 + (x+3)^3 + (x+3)^2 + 4(x+3) + 4 \\
g &= 4(x+3)^4 + 4(x+3)^3 + (x+3)^2 + (x+3) + 4 \\
f+g &= 3(x+3)^4 + 2(x+3)^2 + 3 \\
f+2g &= 2(x+3)^4 + 4(x+3)^3 + 3(x+3)^2 + (x+3) + 2 \\
f+3g &= (x+3)^4 + 3(x+3)^3 + 4(x+3)^2 + 2(x+3) + 1 \\
f+4g &= 2(x+3)^3 + 3(x+3)
\end{aligned}$$

$$\begin{aligned}
f &= 4(x+4)^4 + 2(x+4)^2 + 4(x+4) + 4 \\
g &= 4(x+4)^4 + 3(x+4)^3 + 3(x+4)^2 + 4 \\
f+g &= 3(x+4)^4 + 3(x+4)^3 + 4(x+4) + 3 \\
f+2g &= 2(x+4)^4 + (x+4)^3 + 3(x+4)^2 + 4(x+4) + 2 \\
f+3g &= (x+4)^4 + 4(x+4)^3 + (x+4)^2 + 4(x+4) + 1 \\
f+4g &= 2(x+4)^3 + 4(x+4)^2 + 4(x+4)
\end{aligned}$$

$$\begin{aligned}
f &= 4(x+5)^4 + 4(x+5)^3 + (x+5)^2 + 4(x+5) + 1 \\
g &= 4(x+5)^4 + 2(x+5)^3 + 3(x+5)^2 + 2(x+5) + 3 \\
f+g &= 3(x+5)^4 + (x+5)^3 + 4(x+5)^2 + (x+5) + 4 \\
f+2g &= 2(x+5)^4 + 3(x+5)^3 + 2(x+5)^2 + 3(x+5) + 2 \\
f+3g &= (x+5)^4 \\
f+4g &= 2(x+5)^3 + 3(x+5)^2 + 2(x+5) + 3
\end{aligned}$$

Выражение через функции

$$hx^4 + tx^3 + tx^2 + tx + t$$

$$tx^4 + 2hx^3 + 2hx^2 + 2hx + 2h$$

$$(h+t)x^4 + (2h+t)x^3 + (2h+t)x^2 + (2h+t)x + (2h+t)$$

$$(h+2t)x^4 + (4h+t)x^3 + (4h+t)x^2 + (4h+t)x + (4h+t)$$

$$(h+3t)x^4 + (h+t)x^3 + (h+t)x^2 + (h+t)x + (h+t)$$

$$(h+4t)x^4 + (3h+t)x^3 + (3h+t)x^2 + (3h+t)x + (3h+t)$$

$$h(x+1)^4 + (h+t)(x+1)^3 + (h+3t)(x+1)^2 + (h+2t)(x+1) + h$$

$$t(x+1)^4 + (2h+t)(x+1)^3 + (h+t)(x+1)^2 + (4h+t)(x+1) + t$$

$$(h+t)(x+1)^4 + (3h+2t)(x+1)^3 + (2h+4t)(x+1)^2 + 3t(x+1) + (h+t)$$

$$(h+2t)(x+1)^4 + 3t(x+1)^3 + 3h(x+1)^2 + (4h+4t)(x+1) + (h+2t)$$

$$(h+3t)(x+1)^4 + (2h+4t)(x+1)^3 + (4h+t)(x+1)^2 + 3h(x+1) + (h+3t)$$

$$(h+4t)(x+1)^4 + 4h(x+1)^3 + 2t(x+1)^2 + (2h+t)(x+1) + (h+4t)$$

$$h(x+2)^4 + (2h+t)(x+2)^3 + 4h(x+2)^2 + (3h+4t)(x+2) + h$$

$$t(x+2)^4 + (2h+2t)(x+2)^3 + 4t(x+2)^2 + (3h+3t)(x+2) + t$$

$$(h+t)(x+2)^4 + (4h+3t)(x+2)^3 + (4h+4t)(x+2)^2 + (h+2t)(x+2) + (h+t)$$

$$(h+2t)(x+2)^4 + h(x+2)^3 + (4h+3t)(x+2)^2 + 4h(x+2) + (h+2t)$$

$$(h+3t)(x+2)^4 + (3h+2t)(x+2)^3 + (4h+2t)(x+2)^2 + (2h+3t)(x+2) + (h+3t)$$

$$(h+4t)(x+2)^4 + 4t(x+2)^3 + (4h+t)(x+2)^2 + t(x+2) + (h+4t)$$

$$h(x+3)^4 + (3h+t)(x+3)^3 + (4h+2t)(x+3)^2 + (2h+2t)(x+3) + h$$

$$t(x+3)^4 + (2h+3t)(x+3)^3 + (4h+4t)(x+3)^2 + (4h+2t)(x+3) + t$$

$$(h+t)(x+3)^4 + 4t(x+3)^3 + (3h+t)(x+3)^2 + (h+4t)(x+3) + (h+t)$$

$$(h+2t)(x+3)^4 + (2h+2t)(x+3)^3 + 2h(x+3)^2 + t(x+3) + (h+2t)$$

$$(h+3t)(x+3)^4 + 4h(x+3)^3 + (h+4t)(x+3)^2 + (4h+3t)(x+3) + (h+3t)$$

$$(h+4t)(x+3)^4 + (h+3t)(x+3)^3 + 3t(x+3)^2 + 3h(x+3) + (h+4t)$$

$$\begin{aligned}
& h(x+4)^4 + (4h+t)(x+4)^3 + (h+4t)(x+4)^2 + (4h+t)(x+4) + (h+4t) \\
& t(x+4)^4 + (2h+4t)(x+4)^3 + (3h+t)(x+4)^2 + (2h+4t)(x+4) + (3h+t) \\
& (h+t)(x+4)^4 + h(x+4)^3 + 4h(x+4)^2 + h(x+4) + 4h \\
& (h+2t)(x+4)^4 + (3h+4t)(x+4)^3 + (2h+t)(x+4)^2 + (3h+4t)(x+4) + (2h+t) \\
& (h+3t)(x+4)^4 + 3t(x+4)^3 + 2t(x+4)^2 + 3t(x+4) + 2t \\
& (h+4t)(x+4)^4 + (2h+2t)(x+4)^3 + (3h+3t)(x+4)^2 + (2h+2t)(x+4) + (3h+3t)
\end{aligned}$$

$$\begin{aligned}
& h(x+5)^4 + t(x+5)^3 + t(x+5)^2 + t(x+5) + t \\
& t(x+5)^4 + 2h(x+5)^3 + 2h(x+5)^2 + 2h(x+5) + 2h \\
& (h+t)(x+5)^4 + (2h+t)(x+5)^3 + (2h+t)(x+5)^2 + (2h+t)(x+5) + (2h+t) \\
& (h+2t)(x+5)^4 + (4h+t)(x+5)^3 + (4h+t)(x+5)^2 + (4h+t)(x+5) + (4h+t) \\
& (h+3t)(x+5)^4 + (h+t)(x+5)^3 + (h+t)(x+5)^2 + (h+t)(x+5) + (h+t) \\
& (h+4t)(x+5)^4 + (3h+t)(x+5)^3 + (3h+t)(x+5)^2 + (3h+t)(x+5) + (3h+t)
\end{aligned}$$

Разложение функций

Рассмотрим функции

$$\begin{aligned}
h &= hx^{k-1} + tx^{k-2} + \dots + tx + t \\
t &= tx^{k-1} + ehx^{k-2} + \dots + ehx + eh, \text{ где}
\end{aligned}$$

e – некоторый коэффициент.

Через C_{hi}^{hd} обозначим коэффициент у функции h при x^i , при функции h , поляризация d .

Через C_{ti}^{hd} обозначим коэффициент у функции h при x^i , при функции t , поляризация d .

Через C_{hi}^{td} обозначим коэффициент у функции t при x^i , при функции h , поляризация d .

Через C_{ti}^{td} обозначим коэффициент у функции t при x^i , при функции t , поляризация d .

$$\begin{aligned}
C_{hi}^{hd} &= \binom{k-1}{i} (-d)^{k-1-i} \\
C_{ti}^{hd} &= \sum_{j=i}^{k-2} \binom{j}{i} (-d)^{j-i}
\end{aligned}$$

$$C_{hi}^{td} = \sum_{j=i}^{k-2} e \binom{j}{i} (-d)^{j-i}$$

$$C_{ti}^{td} = \binom{k-1}{i} (-d)^{k-1-i}$$

Заметим, что $\binom{k-1}{i} \not\equiv 0 \pmod{k}$, тогда у функции h при любой поляризации присутствует слагаемое с h , а у функции t при любой поляризации присутствует слагаемое с t .

Пусть $f_a = h + at$, где $a \in [1..k-1]$.

Теорема 1. Для любых d и a у полинома функции $f_a^{(d)}$ k слагаемых, если e – квадратичный невычет по модулю k .

Доказательство. Пусть существуют a, d, i такие, что $f_a^{(d)}[i] = 0$, тогда C_{hi}^{hd} должно быть равно $-aC_{hi}^{td}$, а C_{ti}^{hd} должно быть равно $-aC_{ti}^{td}$.

$$\begin{cases} \binom{k-1}{i} (-d)^{k-1-i} &= -a \sum_{j=i}^{k-2} e \binom{j}{i} (-d)^{j-i} \\ \sum_{j=i}^{k-2} \binom{j}{i} (-d)^{j-i} &= -a \binom{k-1}{i} (-d)^{k-1-i} \end{cases}$$

$\sum_{j=i}^{k-2} \binom{j}{i} (-d)^{j-i} \neq 0$ так как $\binom{k-1}{i} (-d)^{k-1-i} \neq 0$. Следовательно

$$a^{-1} \sum_{j=i}^{k-2} \binom{j}{i} (-d)^{j-i} = a \sum_{j=i}^{k-2} e \binom{j}{i} (-d)^{j-i}$$

Значит $e = (a^{-1})^2$, что противоречит с тем, что e – квадратичный невычет по модулю k . □

Функции одной переменной

Рассмотрим теперь следующие функции h и t одной переменной:

$$h = ex^{k-1} + (x-1)^{k-1}$$

$$t = x^{k-1} + e(x-1)^{k-1}$$

Они выглядят так:

$$\begin{aligned} h &= (e+1)x^{k-1} + x^{k-2} + \dots + x + 1 \\ t &= (e+1)x^{k-1} + ex^{k-2} + \dots + ex + e \end{aligned}$$

Через C_i^{hd} обозначим коэффициент при x^i у функции h при поляризации d .

Через C_i^{td} обозначим коэффициент при x^i у функции t при поляризации d .

$$\begin{aligned} C_i^{hd} &= \binom{k-1}{i}(-d)^{k-1-i} + \sum_{j=i}^{k-2} e \binom{j}{i}(-d)^{j-i} \\ C_i^{hd} &= e \binom{k-1}{i}(-d)^{k-1-i} + \sum_{j=i}^{k-2} \binom{j}{i}(-d)^{j-i} \end{aligned}$$

Теорема 2. Для любой поляризации d , для любого i у любой пары f и g функций из $\{h, t, h+at\}$ ($\forall a \in [1..k-1]$) коэффициенты при x^i не могут быть равны 0 одновременно.

Доказательство. 1) Рассмотрим случай, когда f и $g \in \{h, t\}$. Предположим, что $C_i^{hd} = 0$ и $C_i^{td} = 0$. Тогда, так как $\binom{k-1}{i}(-d)^{k-1-i} \neq 0$, то и $\sum_{j=i}^{k-2} \binom{j}{i}(-d)^{j-i} \neq 0$. А также

$$e \sum_{j=i}^{k-2} \binom{j}{i}(-d)^{j-i} = e^{-1} \sum_{j=i}^{k-2} \binom{j}{i}(-d)^{j-i}$$

Следовательно $e^2 = 1$, но $e \in [2..k-2]$ чего не может быть так как у 1 только два корня 1 и -1.

2) Пусть $f = h + at$, а $g \in \{h, t\}$.

$$C_i^{fd} = C_i^{hd} + aC_i^{td}.$$

Если $C_i^{gd} = 0$, то в C_i^{fd} всего 1 слагаемое и, в силу предыдущего пункта, оно отлично от 0.

3) Последний случай: $f = h + at$, $g = h + bt$.

$$\begin{aligned} C_i^{fd} &= C_i^{hd} + aC_i^{td}, \\ C_i^{gd} &= C_i^{hd} + bC_i^{td}. \end{aligned}$$

Если $C_i^{td} \neq 0$, то $C_i^{fd} \neq C_i^{gd}$, а если $C_i^{td} = 0$, то из первого пункта следует, что $C_i^{hd} \neq 0$.

□

Описание программы

Библиотека для работы с полиномами написана на языке программирования perl. Основные возможности это:

- Для полиномов — класс `Polynomial`
 - Создание поляризованных полиномов над полем по простому модулю.
 - Изменение поляризации полинома.
 - Сложение полиномов и умножение на число.
 - Вывод полиномов в удобном виде (для вставки в tex или csv файл).
- Для систем полиномов — класс `AOP`
 - Создание систем поляризованных полиномов в поле по простому модулю.
 - Проверка системы на сложность.
 - Вывод систем полиномов в виде tex или csv файла.

Заклучение

В работе получены следующие результаты:

1. Найдены сложные системы для пятизначной и семизначной логик.
2. Доказано существование сложных систем в k -значных логиках для любого простого k .
3. Написана программа для работы с символьными полиномами.

Приложение

В приложении приводится код программ, использованных для получения некоторых результатов. Также программы сами по себе представляют интерес, так как могут быть использованы для более широкого ряда задач, чем требовалось при выполнении этой работы.

Программа поиска сложных систем.

```
use strict;
use warnings;
use Carp;
use Getopt::Long;
use autodie;
use feature qw(say);
use Polynomial;

our $k;
my $treshold;
my $infile;
my $outfile;

Getopt::Long::Configure ("bundling");
GetOptions (
    'k=i'          => \$k,
    't|treshold=i' => \$treshold,
    'in:0'         => \$infile, # output file STDIN by default
    'out:0'        => \$outfile # output file STDOUT by default
);

my $in;
unless (defined $infile) {
    open($in, '<', "complex_${k}_${treshold}.txt");
} elsif ($infile) {
```



```

        open($in, '<', $infile);
    } else {
        open($in, '<-');
    }

my $out;
unless (defined $outfile) {
    open($out, '>', "pairs_${k}_$treshhold.txt");
} elsif ($outfile) {
    open($out, '>', $outfile);
} else {
    open($out, '>-');
}

my %functions;

while (my $fun = <$in>) {
    chomp $fun;
    $functions{$fun} = 1;
}

for my $f (keys %functions) {
    for my $g (keys %functions) {
        if ($f lt $g) {
            my $complex = 1;
            for my $c (1..$k-1) {
                my $h = add_mul($f, $g, $c);
                #say $h;
                unless ($functions{$h}) {
                    $complex = 0;
                    last;
                }
            }
        }
    }
}

```

```

        say $out "$f,$g" if $complex;
    }
}
}

```

Программа, которая проверяет сложность систем и выводит их в удобном виде.

```

use strict;
use warnings;
use feature qw(say);

use Polynomial;
use AOP;
use Math::Prime::Util qw(binomial znprimroot is_primitive_root);
use Getopt::Long;

my $k = 5;
my $c;
my $e;
my $v;
my $s;
my $csv;

Getopt::Long::Configure ("bundling");
GetOptions (
    'k=i' => \$k,
    'c=i' => \$c,
    'e=i' => \$e,
    'csv' => \$csv,
    'v=s' => \$v,
    's=s' => \$s,
);

```

```

$c //= znprimroot($k);
$e //= -1*znprimroot($k) % $k;
my $k_1 = $k - 1;

my $hs = "h*x^$k_1 + $k_1*t*x^$k_1 + t*(x+$k_1)^$k_1";
my $ts = "t*x^$k_1 + -$c*h*x^$k_1 + $c*h*(x+$k_1)^$k_1";

my $h = Polynomial->new(k => $k,
    str => "h*x^$k_1 + $k_1*t*x^$k_1 + t*(x+$k_1)^$k_1");
my $t = Polynomial->new(k => $k,
    str => "t*x^$k_1 + -$c*h*x^$k_1 + $c*h*(x+$k_1)^$k_1");

#my $av = AOP->new(k => $k,
    #gens => ["$e*x^$k_1 + (x+1)^$k_1", "$e*(x+1)^$k_1 + (x+2)^$k_1"]);
my $av = AOP->new(
    k => $k,
    gens => ["$e*x^$k_1 + (x+1)^$k_1", "x^$k_1 + $e*(x+1)^$k_1"]
);

my $as = AOP->new(k => $k, gens => [$hs, $ts]);

if ($s) {
    open(my $fh, '>>', $s);
    say $fh $as->fprint(del=>','');
}

if ($v) {
    open(my $fh, '>>', $v);
    say $fh $av->to_csv;
}

```

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. Селезнева С. Н. О сложности поляризованных полиномов функций многозначных логик, зависящих от одной переменной. Дискретная математика. 16. №2. 2004. С. 117–120.
7. Кириченко К. Д. Верхняя оценка сложности полиномиальных нормальных форм булевых функций // Дискретная математика. 17. №3. 2005. С. 80–88.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. 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.
14. Библиотека для работы с поляризованными полиномами: <https://github.com/Obirvalger/SymbolicPolynomial>.