

# Параметро-эффективная расшифровка линейных функций $k$ -значной логики

А. В. Быстрыгова

В работе рассматривается задача параметро-эффективной расшифровки линейных функций  $k$ -значной логики в рамках модели точной расшифровки. Получены точные значения сложности расшифровки для малого количества существенных переменных. Получены верхние и нижние оценки для общего случая для двух типов запросов: на значение и на сравнение. При стремлении числа переменных к бесконечности получен порядок сложности расшифровки для обоих типов запросов.

**Ключевые слова:** точная расшифровка функций, линейные функции  $k$ -значной логики, запросы на значение, запросы на сравнение.

## Введение

На практике часто может возникнуть ситуация, когда нам дано некоторое устройство — “черный ящик”, про которое мы знаем некоторую частичную информацию, например, что оно принадлежит некоторому классу устройств, и нам хочется понять, что за устройство перед нами. При этом мы можем подавать на вход устройства некоторые сигналы и снимать с выходов устройства результат. Если такое устройство является конечным автоматом [1–11], то раздел науки, занимающийся анализом таких устройств называется эксперименты с автоматами [12–16]. При этом если у нас есть только одно такое

устройство, то эксперимент называется простым, и мы, подавая на вход устройства входные воздействия, понимаем, что его состояние меняется каждый такт. Если у нас имеются несколько копий устройства, то эксперимент называется кратным. Если исследуемое устройство является устройством без памяти, то необходимость в нескольких копиях отпадает. Раздел науки, занимающийся анализом устройств без памяти называется расшифровкой функций, а в зарубежной литературе Machine Learning (машинное обучение).

Расшифровка функции — это игра между “учеником” (алгоритмом расшифровки) и “учителем”, в которой “учитель” загадывает функцию из класса, известного “ученику”, и тот должен за наименьшее количество запросов разрешенного вида разгадать функцию.

Наиболее известны две модели расшифровки функций: модель точной расшифровки (exact learning) [17–30] и модель вероятно примерно точной расшифровки (probably approximately correct learning, PAC) [30–35].

В модели точной расшифровки “ученик” должен точно расшифровать загаданную “учителем” функцию, то есть полностью восстановить таблицу значений функции, или если эта функция не дискретная, то восстанавливать ее с некоторой точностью (в этом случае этот процесс называется интерполяцией функции [36]). При вероятно примерно точной расшифровке “ученик” не может выбирать набор, значение функции на котором он хочет узнать. Он делает запрос к функции, а “учитель”, руководствуясь заранее определенным распределением вероятности на области определения функции, выбирает набор значений аргументов и выдает “ученику” пару: набор и значение функции на этом наборе. Цель “ученика” — восстановить вектор значений загаданной функции так, чтобы вероятность ошибки была небольшой.

Одной из первых рассматривалась задача расшифровки монотонных булевых функций. Эта же задача является наиболее полно решенной [18–21]. В работах [22–25] получено решение задачи расшифровки функции разбиения булевого куба на под-

кубы. В работе [26] предлагается общий подход к параллельной параметро-эффективной расшифровке интервально-постоянных функций. Этот класс функций включает в себя, в частности, и монотонные функции и функции разбиения булевого куба на подкубы.

В работе [27] рассматривалась задача расшифровки пороговых функций, а в работах [28, 29] исследовалась задача расшифровки арифметических сумм монотонных конъюнкций. И, наконец, в работах [30–32] изучалась задача расшифровки линейных булевых функций.

О параметро-эффективной расшифровке говорят, если функции из загадываемого класса зависят от большого числа переменных, но известно, что загаданная функция зависит существенно от малого числа переменных, и сложность расшифровки зависит от числа существенных переменных (даже если это число неизвестно “ученику”) и слабо зависит от общего числа переменных (это число всегда известно “ученику”).

В данной работе рассматривается задача параметро-эффективной расшифровки линейных функций  $k$ -значной логики в рамках модели точной расшифровки. Задача рассматривается в двух вариантах:

- 1) “ученик” может задавать “учителю” запросы на значение: подать “учителю” один набор, ответ на запрос: значение функции на этом наборе
- 2) “ученик” может задавать “учителю” запросы на сравнение: подать “учителю” два набора, ответ на запрос: вернуть на каком наборе значение больше или сказать, что значения на обоих наборах равны.

В работах [37, 38] рассматривалась расшифровка функций ранжирования интернет поисковиков запросами на сравнение.

В работе [30] было получено, что сложность расшифровки линейных булевых функции  $f$ , существенно зависящей от  $p$  переменных, равна сложности расшифровки функции  $f$ , существенно зависящей от  $n - p$  переменных, а также в рамках модели точной расшифровки запросами на значение получены верх-

ние оценки:  $\lceil \log n \rceil$  для  $p = 1$ ,  $\lceil 3 \log n \rceil - 2$  для  $p = 2$ ,  $\lceil 4 \log n \rceil - 3$  для  $p = 3$ .

В работах [30], [31], [32] была рассмотрена эта задача для произвольного  $p$  в рамках модели вероятно примерно точной расшифровки РАС. Для этой модели в работе [30] получена оценка  $O(p \log n/p)$ . В работе [31] получена оценка  $O(n^{1-\frac{1}{p}} \log n)$ , если известно, что загаданная функция зависит от не более чем  $p$  переменных.

В работе [40] была рассмотрена задача расшифровки линейных булевых функций в рамках модели точной расшифровки запросами на значение, получена верхняя оценка сложности  $p \log n + p$ .

В настоящей работе в рамках модели точной расшифровки получены оценки сложности расшифровки запросами на значение линейных булевых функций для случаев  $p = 2$  и  $p = 3$ , отличающиеся от точного значения не более чем на 2, получены верхние и нижние оценки сложности параметро-эффективной расшифровки запросами на значение и запросами на сравнение линейных функций  $k$ -значной логики с нулевым свободным членом. При стремлении числа переменных к бесконечности получен порядок сложности расшифровки для обоих типов запросов.

Автор выражает благодарность д.ф.-м.н. профессору Э.Э.Гасанову за научное руководство и помощь в работе.

## Основные понятия и формулировка результатов.

Пусть  $\Psi(k)$  — некоторый класс функций  $k$ -значной логики ( $k \geq 2$ ). Тогда  $\Psi^n(k) \subseteq \Psi(k)$  есть подкласс, состоящий из функций, зависящих от  $n$  переменных  $x_0, x_1, \dots, x_{n-1}$ ;  $\Psi^{p,n}(k) \subseteq \Psi^n(k)$  есть подкласс, состоящий из функций от  $n$  переменных  $x_0, \dots, x_{n-1}$ , существенно зависящих ровно от  $p$  переменных.

Под *запросом на значение* к функции  $f \in \Psi(k)$  будем понимать вектор (набор)  $a \in E^n$ ,  $E = \{0, 1, \dots, k-1\}$ . Под *ответом на запрос на значение* будем понимать значение  $f(a)$ .

Под *запросом на сравнение* к функции  $f \in \Psi(k)$  будем понимать пару  $(a, b)$ ,  $a, b \in E^n$ ,  $E = \{0, 1, \dots, k-1\}$ . Под *ответом на запрос на сравнение* будем понимать значение

$$\text{sign}(f(a) - f(b)) = \begin{cases} 1 & \text{если } f(a) > f(b) \\ 0 & \text{если } f(a) = f(b) \\ -1 & \text{если } f(a) < f(b) \end{cases}$$

Если в дальнейшем будем рассматривать задачу расшифровки запросами на значение, то будем в обозначениях дописывать индекс  $v$ , если запросами на сравнение, то  $c$ .

Под *алгоритмом расшифровки* будем понимать условный эксперимент, который последовательно генерирует запросы к функции в зависимости от ответов на предыдущие запросы. Будем говорить, что *алгоритм расшифровывает функцию  $f$  из  $\Psi^n(k)$* , если значения функции на наборах, сгенерированных условным экспериментом, однозначно определяют таблицу значений функции  $f$  при условии, что  $f \in \Psi^n(k)$ . Скажем, что *алгоритм расшифровывает класс функций  $\Psi(k)$* , если для любого  $n \in \mathbb{N}$  он расшифровывает любую функцию из  $\Psi^n(k)$  при условии, что он получает  $n$  в виде входного параметра. Обозначим множество алгоритмов расшифровки класса  $\Psi(k)$  через  $\mathcal{A}(\Psi(k))$ . Любой элемент множества  $\mathcal{A}(\Psi(k))$  можно понимать и как единичный алгоритм, получающий  $n$  на вход, и как последовательность алгоритмов, такую что  $n$ -й алгоритм последовательности расшифровывает функции из  $\Psi^n(k)$ .

Пусть  $A \in \mathcal{A}(\Psi(k))$ ,  $f \in \Psi(k)$ , тогда обозначим через  $\varphi(A, f)$  число запросов на значение функции, требуемое алгоритму  $A$  для расшифровки функции  $f$ . Будем называть  $\varphi(A, f)$  *сложностью алгоритма  $A$  на функции  $f$* .

Положим

$$\varphi(\Psi(k), n, p) = \min_{A \in \mathcal{A}(\Psi(k))} \max_{f \in \Psi^{p,n}(k)} \varphi(A, f).$$

Алгоритм расшифровки, на котором достигается минимум — это такой алгоритм, который работает лучше других алгоритмов на самой плохой функции из класса.

Далее под операцией  $+$  будем понимать обычное сложение чисел, под операцией  $\oplus$  - сложение по модулю  $k$ . Под операцией  $\cdot$  будем понимать умножение по модулю  $k$ . Если под этими знаками будет пониматься иная операция, то будет отдельно уточнено.

Пусть  $\Phi(k, n, p)$  — множество функций вида  $f(x_0, x_1, \dots, x_{n-1}) = c_0x_0 \oplus c_1x_1 \oplus \dots \oplus c_{n-1}x_{n-1}$ ,  $c_i, x_i \in \{0, 1, 2, \dots, k-1\}$ ,  $|\{i : c_i \neq 0\}| = p$ .

Поскольку в данной работе мы будем исследовать только такие функции, то обозначим

$$\varphi(k, n, p) = \varphi(\Phi(k), n, p).$$

Предложенные алгоритмы расшифровки заключаются в

- 1) нахождении существенных переменных  $x_i$ , то есть при которых  $c_i \neq 0$
- 2) нахождении значений ненулевых коэффициентов  $c_i$ .

Будем говорить, что *алгоритм находит существенную переменную и коэффициент перед ней функции  $f$  из  $\Phi(k, n, p)$* , если по значениям функции на наборах, сгенерированных алгоритмом, мы хотя бы для одной переменной функции  $f$  однозначно можем сказать, что она является существенной и знаем значение коэффициента перед ней. Обозначим множество алгоритмов, находящих существенную переменную и коэффициент перед ней для класса  $\Phi(k, n, p)$  через  $\mathcal{G}(\Phi(k, n, p))$ .

Пусть  $A \in \mathcal{G}(\Phi(k, n, p))$ ,  $f \in \Phi(k, n, p)$ , тогда обозначим через  $\mu(A, f)$  число запросов на значение функции, требуемое алгоритму  $A$  для нахождения какой-либо существенной переменной и коэффициента перед ней функции  $f$ .

Положим

$$\mu(k, n, p) = \min_{A \in \mathcal{G}(\Phi(k, n, p))} \max_{f \in \Phi(k, n, p)} \mu(A, f).$$

Если  $a$  — вещественное число, то через  $\lceil a \rceil$  или  $\text{ceil}(a)$  обозначим наименьшее целое не меньше  $a$ , через  $\lfloor a \rfloor$  обозначим

наибольшее целое не большее  $a$ . Под  $\log n$  будем понимать двоичный логарифм от  $n$ .

Пусть  $E_k = \{0, 1, \dots, k-1\}$ ,  $E_k^n = \{(x_1, \dots, x_n) : x_i \in E_k, 1 \leq i \leq n\}$ .

Под  $d(a, b)$  будем понимать  $|\{i : 1 \leq i \leq n, a_i \oplus b_i \neq 0\}|$ ,  $a, b \in E_k^n$ .

Будем называть подмножество  $G = \{g_1, \dots, g_s\}$  множества  $E_k^n$  кодом длины  $n$  с кодовым расстоянием  $D$ , если для любых двух его элементов  $g_i$  и  $g_j$   $d(g_i, g_j) \geq D$ .

Будем называть код  $G$   $k$ -линейным  $(n, p)$ -кодом, если он является  $p$ -мерным линейным подпространством пространства  $E_k^n$  с операциями сложения по модулю  $k$  в качестве внутреннего закона композиции, и умножения на число из  $E_k^n$  по модулю  $k$  в качестве внешнего закона композиции.

Матрицу  $H \in E_2^{m \times n}$  будем называть проверочной матрицей  $k$ -линейного  $(n, p)$  кода  $G$ , если  $Hg = 0 \forall g \in G$  и  $Hx \neq 0 \forall x \notin G$ .

Будем говорить, что вектор принадлежит нулевому пространству матрицы, если он ортогонален каждой строке матрицы.

Весом  $\|g\|$  элемента  $g \in E_k^n$  будем называть число  $|\{i : 1 \leq i \leq n, g_i \neq 0\}|$ .

**Теорема 1.** Для любого натурального  $n$ ,  $n > 2$ , справедливы неравенства  $2] \log n[-2 \leq \varphi_V(2, n, 2) \leq 2] \log n[-1$ , причем если  $\log n - [\log n] \in \{0\} \cup (1/2, 1)$ , то  $\varphi_V(2, n, 2) = 2] \log n[-1$ .

**Теорема 2.** Для любого натурального  $n$ ,  $n > 5$ , справедливы соотношения

- 1)  $3] \log n[-4 \leq \varphi_V(2, n, 3) \leq 3] \log n[-2$ , если  $\log n - [\log n] \in (0, 1/3]$
- 2)  $3] \log n[-3 \leq \varphi_V(2, n, 3) \leq 3] \log n[-2$ , если  $\log n - [\log n] \in (1/3, 2/3]$
- 3)  $\varphi_V(2, n, 3) = 3] \log n[-2$ , если  $\log n - [\log n] \in \{0\} \cup (2/3, 1)$ .

**Теорема 3.** Для любых натуральных  $n, k, p$  ( $n > 2, 1 \leq p < n/2, k > 2$ ) имеет место следующее неравенство  $\varphi_V(k, n, p) \leq 1 + (p - 1) \cdot (\lceil \log_2(k - 1) \rceil + \lceil \log_2(n - 1) \rceil) + p \log_2 n$ .

**Теорема 4.** Для любых натуральных  $n, k, p$  ( $n > 2, 1 \leq p < n/2, k \geq 2$ ) имеет место следующее неравенство

- 1)  $\varphi_C(k, n, p) \leq 1 + (p - 1) \cdot (\lceil \log_2(k - 1) \rceil + \lceil \log_2(n - 1) \rceil) + p \log_2 n$ , если  $k > 2$ ;
- 2)  $\varphi_C(k, n, p) \leq 1 + (p - 1) \cdot (\lceil \log_2(k - 1) \rceil + \lceil \log_2(n - 1) \rceil) + p \log_2 n$ , если  $k = 2$ .

**Теорема 5.** Для любых натуральных  $n, k, p$  ( $n > 2, 1 \leq p < n/2, k \geq 2$ ) имеет место следующее неравенство  $\varphi_V(k, n, p) \geq p \log_2(k - 1) + p \cdot \log_2(n - p + 1) - \log_2 p$ .

**Теорема 6.** Для любых натуральных  $n, k, p$  ( $n > 2, 1 \leq p < n/2, k \geq 2$ ) имеет место следующее неравенство  $\varphi_C(k, n, p) \geq p \log_2(k - 1) + p \cdot \log_2(n - p + 1) - \log_2 p$ .

Из теорем 3, 4, 5, 6 получаем

**Следствие 1.**  $\varphi_V(k, n, p) = \varphi_C(k, n, p) = O(p \log n)$  при  $n \rightarrow \infty$

## Оценки сложности расшифровки запросами на значение линейных булевых функций от двух переменных

Приведем сначала классическую мощностную нижнюю оценку.

**Лемма 1** (мощностная нижняя оценка). Для любого класса функций  $\Psi(k)$ , любого алгоритма  $A$ , расшифровывающего класс  $\Psi(k)$ , для любого натурального  $n$  существует такая функция  $f$  из  $\Psi^n(k)$ , что  $\varphi_V(A, f) \geq \lceil \log_2 |\Psi^n(k)| \rceil$ .

*Доказательство.* Любой запрос  $a = (a_0, a_1, \dots, a_{n-1})$  на значение функции разбивает множество функций  $\Psi(k)^n$  на два подмножества: те функции, которые на наборе  $a$  принимают значение  $f(a)$ , и те — которые на этом наборе принимают значение не  $f(a)$ .

Чтобы найти функцию, которую произвольным фиксированным алгоритмом трудно расшифровать, нам достаточно “прятать” искомую функцию в множестве большей мощности. В этом случае каждый запрос будет сокращать множество потенциальных функций в лучшем случае в 2 раза. И, значит, если число запросов будет меньше чем  $\lceil \log |\Phi^n| \rceil$ , то расшифровываемая функция будет находиться в множестве мощности не менее 2, а, значит, еще не определена однозначно.

Лемма доказана. □

**Лемма 2.** Для любого натурального  $n$ ,  $n > 2$ , справедливо неравенство  $\varphi_V(2, n, 2) \geq \lceil 2 \log n \rceil - 2$ , причем если  $\log n - \lfloor \log n \rfloor \in \{0\} \cup (1/2, 1)$ , то  $\varphi_V(2, n, 2) \geq \lceil 2 \log n \rceil - 1$ .

*Доказательство.* Множество  $\Phi(2, n, 2)$  имеет мощность  $C_n^2 = n(n-1)/2 = n^2/2 - n/2$ . Так как при  $n > 2$  справедливо неравенство  $n^2/2 - n/2 > n^2/4$ , то с учетом леммы 1 имеем

$$\varphi_V(2, n, 2) \geq \lceil \log C_n^2 \rceil \geq \lceil \log n^2/4 \rceil = \lceil 2 \log n \rceil - 2,$$

т.е.

$$\varphi_V(2, n, 2) \geq \lceil 2 \log n \rceil - 1.$$

Обозначим  $\lfloor \log n \rfloor = m$ ,  $c = \log n - m$ ,  $c \in [0, 1)$ , т.е.  $\log n = m + c$ .

Если  $c = 0$ , то  $\lceil 2 \log n \rceil = \lceil 2m \rceil = 2 \log n$ .

Если  $c \in (0, 1/2]$ , то  $\lceil 2 \log n \rceil = \lceil 2m + 2c \rceil = 2m + 2$ , а  $\lceil 2 \log n \rceil - 2 = 2m$ , т.е.  $\lceil 2 \log n \rceil - 2 = \lceil 2 \log n \rceil - 2$ .

Если  $c \in (1/2, 1)$ , то  $\lceil 2 \log n \rceil = \lceil 2m + 2c \rceil = 2m + 3$ , а  $\lceil 2 \log n \rceil - 2 = 2m + 1$ , т.е.  $\lceil 2 \log n \rceil - 2 = \lceil 2 \log n \rceil - 2$ .

Лемма доказана. □

Теперь получим верхнюю оценку для  $\varphi_V(2, n, 2)$ .

Пусть  $B$  — некоторое подмножество множества переменных  $\{x_0, x_1, \dots, x_{n-1}\}$ . Под *запросом на множестве  $B$*  будем понимать запрос значения функции  $f$  на наборе, в котором все переменные из  $B$  установлены в 1, а остальные в 0. Будем обозначать его  $f(B)$ .

**Лемма 3.** *Если  $B \subseteq \{x_0, x_1, \dots, x_{n-1}\}$  и в  $B$  содержится нечетное число существенных переменных функции  $f$  из  $\Phi(2, n, p)$ , то одну из существенных переменных множества  $B$  можно найти за не более  $\lceil \log |B| \rceil$  запросов.*

*Доказательство.* Разделим  $B$  произвольным образом на два непересекающихся множества  $B_1$  и  $B_2$ , отличающиеся по мощности не более чем на 1. Запросим значение на  $B_1$ . Если оно равно 0 (то есть в  $B_1$  четное число существенных переменных, а значит во множестве  $B_2$  нечетное), положим  $B = B_2$ , иначе, положим  $B = B_1$ . Затем опять разделим  $B$  на два множества и т.д. до тех пор пока  $B$  состоит из более чем одной переменной. Получившееся в итоге множество  $B$  состоит из одной переменной, которая и является существенной. Поскольку каждый раз множество  $B$  делится надвое, то для нахождения существенной переменной потребуется не более  $\lceil \log |B| \rceil$  запросов.

Лемма доказана. □

Как следствие получаем следующий результат, ранее опубликованный в [30].

**Следствие 2.** *Для любого натурального  $n$  имеет место равенство  $\varphi_V(2, n, 1) = \lceil \log n \rceil$ .*

*Доказательство.* Согласно мощностной нижней оценке  $\varphi_V(2, n, 1) \geq \lceil \log |\Phi(2, n, 1)| \rceil = \lceil \log n \rceil$ . С другой стороны, поскольку любая функция из  $\Phi(2, n, 1)$  имеет одну существенную переменную, то согласно лемме 3 ее можно найти не более чем за  $\lceil \log n \rceil$  запросов. □

Введем  $B^i$  — подмножество переменных множества  $B$ , у которых в двоичном представлении номера переменной  $i$ -й бит

равен 1 (нумерация идет от младших значимых битов, начиная с 0).

Например, пусть  $n = 8$ ,  $B = \{x_1, x_3, x_4\}$ . Тогда  $B^0 = \{x_1, x_3\}$ ,  $B^1 = \{x_3\}$ ,  $B^2 = \{x_4\}$ .

**Лемма 4.** Для любого натурального  $n$ ,  $n > 2$ , справедливо неравенство  $\varphi_V(2, n, 2) \leq 2 \lceil \log n \rceil - 1$ .

*Доказательство.* Пусть  $B = \{x_0, x_1, \dots, x_{n-1}\}$ . Запросим значение искомой функции  $f$  на множествах  $B^i$ ,  $i = 0, 1, \dots, \lceil \log n \rceil - 1$ , т.е. сделаем  $\lceil \log n \rceil$  запросов.

Если на  $B^i$  функция  $f$  равна 0, то в  $B^i$  содержится четное число существенных переменных, т.е. в нашем случае либо 0, либо 2, и, значит,  $i$ -й бит в номерах обоих существенных переменных одинаков.

Если  $f(B^i) = 1$ , то в  $B^i$  содержится нечетное число существенных переменных, что в нашем случае означает, что в  $B^i$  содержится ровно одна существенная переменная. Следовательно, номера существенных переменных по  $i$ -му биту отличаются.

Поскольку номера существенных переменных отличны, то обязательно существует хотя бы один бит, на котором они отличаются. Пусть эти номера отличаются в  $p$ -м бите. Тогда  $f(B^p) = 1$ , и в  $B^p$  содержится ровно одна существенная переменная. Согласно лемме 3 мы можем найти эту переменную за  $\lceil \log |B^p| \rceil$  запросов. Учитывая, что  $|B^p| \leq 2^{\lceil \log n \rceil - 1}$ , имеем, что на нахождение этой переменной мы потратим не более  $\lceil \log n \rceil - 1$  запросов.

Остается заметить, что поскольку мы опросили значение функции  $f$  на каждом из запросов  $B^i$ , то мы для каждого  $i \in \{0, 1, \dots, \lceil \log n \rceil - 1\}$  знаем совпадают ли  $i$ -е биты номеров существенных переменных. Тем самым, зная одну переменную, мы автоматически узнаем и вторую существенную переменную.

Таким образом, задав не более чем  $2 \lceil \log n \rceil - 1$  запросов, мы узнаем обе существенные переменные функции. Лемма доказана.  $\square$

Теорема 1 является простым следствием лемм 2 и 4.

## Оценки сложности расшифровки запросами на значение линейных булевых функций от трех переменных

**Лемма 5.** Для любого натурального  $n$ ,  $n > 5$ , справедливо неравенство  $\varphi_V(2, n, 3) \geq 3 \log n[-4$ , причем

1) если  $\log n - [\log n] \in (1/3, 2/3]$ , то  $\varphi_V(2, n, 3) \geq 3 \log n[-3$ ;

2) если  $\log n - [\log n] \in \{0\} \cup (2/3, 1)$ , то  $\varphi_V(2, n, 3) \geq 3 \log n[-2$ .

*Доказательство.* Множество  $\Phi(2, n, 3)$  имеет мощность  $C_n^3 = n(n-1)(n-2)/6 = n^3/6 - n^2/2 + n/3$ . Так как при  $n > 5$  справедливо неравенство  $n^3/6 - n^2/2 + n/3 > n^3/8$ , то с учетом леммы 1 имеем

$$\varphi_V(2, n, 3) \geq \lceil \log C_n^3 \rceil \geq \lceil \log(n^3/8) \rceil = 3 \log n[-3,$$

т.е.

$$\varphi_V(2, n, 3) \geq 3 \log n[-2.$$

Обозначим  $[\log n] = m$ ,  $c = \log n - m$ ,  $c \in [0, 1)$ , т.е.  $\log n = m + c$ .

Если  $c = 0$ , то  $\lceil 3 \log n \rceil = 3m = \lceil 3 \log n \rceil$ .

Если  $c \in (0, 1/3]$ , то  $\lceil 3 \log n \rceil = 3m + 3$ , а  $\lceil 3 \log n \rceil = \lceil 3m + 3c \rceil = 3m + 1$ , т.е.  $\lceil 3 \log n \rceil = \lceil 3 \log n \rceil[-2$ .

Если  $c \in (1/3, 2/3]$ , то  $\lceil 3 \log n \rceil = 3m + 3$ , а  $\lceil 3 \log n \rceil = \lceil 3m + 3c \rceil = 3m + 2$ , т.е.  $\lceil 3 \log n \rceil = \lceil 3 \log n \rceil[-1$ .

Если  $c \in (2/3, 1)$ , то  $\lceil 3 \log n \rceil = 3m + 3$ , а  $\lceil 3 \log n \rceil = \lceil 3m + 3c \rceil = 3m + 3$ , т.е.  $\lceil 3 \log n \rceil = \lceil 3 \log n \rceil$ .

Лемма доказана.  $\square$

**Лемма 6.** Для любого натурального  $n$ ,  $n > 3$ , справедливо неравенство  $\varphi_V(2, n, 3) \leq 3 \log n[-2$ .

*Доказательство.* Пусть  $B = \{x_0, x_1, \dots, x_{n-1}\}$ . Запросим значение  $f$  на  $B^0$ . Если  $f(B^0) = 1$ , положим  $C = B^0$ , иначе  $C = B \setminus B^0$ . В  $C$  содержится нечетное количество существенных переменных. Согласно лемме 3 одну из существенных переменных можно найти за  $\lceil \log |C| \rceil$  запросов. Поскольку

Параметро-эффективная расшифровка линейных функций  $k$ -значной логики

$|C| \leq 2^{\lceil \log n \rceil - 1}$ , имеем, что для нахождения одной существенной переменной мы потратили не более  $\lceil \log n \rceil - 1$  запросов.

Теперь применяя лемму 4 к  $B$ , найдем за не более  $2 \lceil \log n \rceil - 2$  запросов остальные существенные переменные (мы уже знаем значение  $f$  на  $B^0$ , поэтому не нужно заново запрашивать это значение, следовательно потребуется за один запрос меньше).

Таким образом, задав не более  $1 + (\lceil \log n \rceil - 1) + (2 \lceil \log n \rceil - 2) = 3 \lceil \log n \rceil - 2$  запросов, мы узнаем все существенные переменные функции  $f$ .

Лемма доказана.  $\square$

Теорема 2 является следствием лемм 5 и 6.

## Верхние оценки для общего случая.

### Расшифровка запросами на значение.

**Лемма 7.** Если  $f \in \Phi(k, n, p)$ ,  $B \subseteq \{x_1, \dots, x_n\}$ ,  $f(B) \neq 0$ , то  $\mu(k, n, p) \leq \text{ceil}(\log_2 |B|)$  запросов на значение.

*Доказательство.* Разделим произвольным образом  $B$  на два множества  $B_0, B_1$ , отличающиеся по мощности не более чем на 1. Запросим значение  $f(B_0)$ . Если  $f(B_0) \neq 0$ , положим  $B = B_0$ , иначе  $B = B_1$ . Будем продолжать делить, пока  $B$  состоит из более чем одной переменной.

Переменная, оставшаяся в  $B$  в итоге, и есть существенная, а коэффициент при ней в представлении  $f$  равен  $f(B)$ .

Каждый раз множество  $B$  сокращается по мощности в минимум два раза, поэтому потребуется  $\lceil \log_2 |B| \rceil$  запросов на значение.  $\square$

**Лемма 8.** В каждом  $k$ -линейном коде  $G$  кодовое расстояние  $D$  равно весу его минимального ненулевого элемента:  $D = \min_{g \neq 0, g \in G} \|g\|$ .

*Доказательство.* Кодовое расстояние не больше веса минимального ненулевого элемента, так как  $(0, \dots, 0) \in G$ . Покажем, что  $D$  не может быть строго меньше этого числа.

Доказывать будем от противного. Пусть  $D < \min_{g \neq 0, g \in G} \|g\|$ . Значит, существуют  $g_1, g_2 \in G$  такие, что  $d(g_1, g_2) < \min_{g \neq 0, g \in G} \|g\|$ . Имеем,  $d(g_1, g_2) = \|g_1 \oplus g_2\|$ ,  $g_1 \oplus g_2 \in G$ . Получается,  $\|g_1 \oplus g_2\| < \min_{g \neq 0, g \in G} \|g\|$ . Противоречие.

Отсюда следует, что  $D = \min_{g \neq 0, g \in G} \|g\|$ . □

**Лемма 9.** *Для того, чтобы матрица  $H$  была проверочной матрицей  $k$  – линейного кода с кодовым расстоянием  $D$  необходимо и достаточно, чтобы любые  $D-1$  столбцов матрицы  $H$  были линейно независимы.*

*Доказательство.* Докажем необходимость. Пусть  $H$  – проверочная матрица  $k$  – линейного кода  $G$  с кодовым расстоянием  $D$ .

Если в матрице  $H$  линейная комбинация столбцов  $h_{i_1}, h_{i_2}, \dots, h_{i_s}$   $\alpha_1 h_{i_1} \oplus \alpha_2 h_{i_2} \oplus \dots \oplus \alpha_s h_{i_s} = 0$ , где  $\alpha_i \neq 0$ , то  $Hv = 0$  для  $v$ , у которого компоненты с номерами  $i_1, i_2, \dots, i_s$  соответственно равны  $\alpha_1, \alpha_2, \dots, \alpha_s$ , а остальные компоненты равны 0. Получается,  $v \in G$ , по лемме 8 следует  $s \geq D$ . Значит, любая нетривиальная комбинация из меньше чем  $D$  столбцов будет линейно независима. Необходимость доказана.

Докажем достаточность. Пусть любые  $D-1$  столбцов матрицы  $H$  линейно независимы. Тогда произведение матрицы  $H$  и любого ненулевого вектора  $v$  с не более чем  $D-1$  ненулевыми компонентами не равно нулевому вектору. Заметим, что нулевое пространство матрицы  $H$  и будет  $k$  – линейным кодом с кодовым расстоянием не меньшим  $D$ . Достаточность доказана. □

**Лемма 10.** *Если числа  $n, m \geq 1, D \geq 1, k \geq 2$  удовлетворяют неравенству  $2^m > \binom{n-1}{D-1} \cdot (k-1)^{D-1}$ , то существует проверочная матрица  $H \in E_2^{m \times n}$   $k$  – линейного кода с расстоянием  $D+1$ .*

*Доказательство.* В качестве первого столбца выберем любой вектор из  $E_2^m$ . Предположим, что уже выбраны первые  $j < n$  столбцов матрицы так, что любые  $D$  из них линейно независимы. Есть  $\binom{j}{D-1} \cdot (k-1)^{D-1}$  способов выбрать линейную комбинацию из  $D-1$  столбцов из них. Поэтому в качестве  $j+1$

Параметро-эффективная расшифровка линейных функций  $k$ -значной логики

столбца  $\binom{j}{D-1} \cdot (k-1)^{D-1}$  векторов нельзя выбирать. Но так как  $2^m > \binom{n-1}{D-1} \cdot (k-1)^{D-1} \geq \binom{j}{D-1} \cdot (k-1)^{D-1}$ , существует хотя бы один вектор, который можно выбрать в качестве  $j+1$  столбца. По лемме 9 полученная в итоге матрица будет проверочной матрицей кода с расстоянием  $D+1$ .  $\square$

**Лемма 11.** Пусть  $A_1, A_2, \dots, A_m$  - строки проверочной матрицы  $H \in E_2^{m \times n}$  с кодовым расстоянием  $p+1$ ,  $f \in \Phi(k, n, p)$ . Тогда  $f$  можно расшифровать за не более  $m+p \log_2 n$  запросов на значение.

*Доказательство.* Пусть  $c = (c_1, c_2, \dots, c_n)$ , где  $c_i$  - коэффициенты в представлении  $f$ . Заметим, что  $Hc = (f(A_1), \dots, f(A_m))^T$ .

Узнаем значения  $f(A_1), f(A_2), \dots, f(A_m)$ . Так как у вектора  $c$  ровно  $p$  ненулевых компонент, то по определению проверочной матрицы  $Hc \neq 0$ . То есть существует  $r$  такой, что  $f(A_r) \neq 0$ . Воспользуемся леммой 7 и за не более  $\log_2 n$  запросов на значение найдем существенную переменную  $x_i$  и коэффициент  $c_i$  перед ней в представлении  $f$ . Далее положим  $f = f - c_i \cdot x_i$  и получим задачу с меньшим числом существенных переменных и аналогично будем находить одну за одной существенные переменные, пока не найдем все. Только запрашивать заново  $f(A_1), f(A_2), \dots, f(A_m)$  ненужно, можно восстановить ответы на новые запросы на основе номеров и коэффициентов уже разгаданных существенных переменных. Итого, будет сделано не более  $m+p \log_2 n$  запросов на значение для расшифровки  $f$ .  $\square$

Докажем теорему 3

*Доказательство.* Построим проверочную матрицу с  $m$  рядами и кодовым расстоянием  $p+1$ .

Положим  $S = \binom{n-1}{p-1} \cdot (k-1)^{p-1}$ .

Заметим, что

$$\log_2 S \leq (p-1) \log_2 (k-1) + \log_2 \binom{n-1}{p-1} \leq (p-1) \log_2 (k-1) \cdot$$

$$\begin{aligned} & \cdot [ + \log_2 ((n-1)(n-2) \dots (n-p+1)) / ((p-1)(p-2) \dots 2 \cdot 1) ] \leq \\ & \leq (p-1) \log_2 (k-1) [ + (p-1) \log_2 (n-1) ] \leq (p-1) \cdot \\ & \cdot ( \log_2 (k-1) [ + \log_2 (n-1) ] ). \end{aligned} \quad (1)$$

Положим  $m = 1 + \log_2 S$ . По лемме 10 существует проверочная матрица с  $m$  строками и  $n$  столбцами с кодовым расстоянием  $p+1$ . По лемме 11 по построенной в лемме 10 проверочной матрице можно расшифровать  $f$  за не более  $m + p \log_2 n \leq 1 + (p-1) \cdot ( \log_2 (k-1) [ + \log_2 (n-1) ] + p \log_2 n$  запросов на значение.  $\square$

### Расшифровка запросами на сравнение.

Известно, что  $f \in \Phi(k, 1, 1)$ ,  $f = q \cdot x$ . Узнаем значение на следующих запросах на сравнение  $(2^i, 2^{i+1})$ ,  $i = 0, \dots, \text{ceil}(\log_2 k) - 1$ .

Под операцией  $*$  будем понимать обычное умножение чисел.

**Лемма 12.** *Если  $k > 2$ , то по ответам на указанные выше  $\text{ceil}(\log_2 k)$  запросов коэффициент  $q$  восстанавливается однозначно.*

*Доказательство.* Докажем от противного. Пусть существуют  $a$  и  $b$  ( $0 < a < b < k$ ), для которых все ответы на указанные выше запросов совпали.

Докажем индукцией по номеру запроса, что  $0 < 2^i * (b-a) < k$  и  $a \cdot 2^i < b \cdot 2^i$ .

**База индукции.**  $i = 0$ .

$$0 < 2^0 * (b-a) < k \text{ и } a < b.$$

$$i = 1$$

Пусть ответ на запрос  $(2^0, 2^1)$  равен 1. Тогда  $a < k \leq 2 * a < 2 * k$ ,  $b < k \leq 2 * b < 2 * k$ , то есть  $a < b < k \leq 2 * a < 2 * b < 2 * k$ .  
 $\implies a \cdot 2 < b \cdot 2$ ,  $0 < 2 * (b-a) < k$ .

Пусть ответ на запрос  $(2^0, 2^1)$  равен  $-1$ . Тогда  $a < 2 * a < k$ ,  
 $b < 2 * b < k \implies a \cdot 2 < b \cdot 2$ ,  $0 < 2 * (b - a) < k$ .

Пусть ответ на запрос  $(2^0, 2^1)$  равен  $0$ . Тогда  $a = 2 \cdot a$ ,  $b = 2 \cdot b$   
 $\implies a = b = 0 \implies$  Противоречит тому, что  $a < b$ , значит такого  
 ответа на запрос не может быть.

**Предположение индукции.** Для  $i$  верно, что  $0 < 2^i * (b - a) < k$ ,  
 $a \cdot 2^i < b \cdot 2^i$ .

**Индуктивный переход.** Докажем, что  $0 < 2^{i+1} * (b - a) < k$ ,  
 $a \cdot 2^{i+1} < b \cdot 2^{i+1}$ .

Возможны 3 случая:

- 1) Ответ на запрос  $(2^i, 2^{i+1})$  равен  $-1$ , то есть  $0 < (a \cdot 2^i) < (a \cdot 2^i) * 2 < k$ ,  $0 < (b \cdot 2^i) < (b \cdot 2^i) * 2 < k \implies$  Из предположения индукции следует, что  $a \cdot 2^{i+1} < b \cdot 2^{i+1}$ .

Теперь осталось показать, что  $0 < 2^{i+1} * (b - a) < k$ .

$$a \cdot 2^i = (a * 2^i) \text{ mod } k.$$

$$a * 2^i = k * p + (a \cdot 2^i)$$

$$b * 2^i = k * q + (b \cdot 2^i)$$

Из предположения индукции следует, что  $q = p$ . Действительно, если бы

- $q > p$ , то  $2^i * (b - a) = k * (q - p) + (b \cdot 2^i - a \cdot 2^i) > k$
- $q < p$ , то  $2^i * (b - a) = k * (q - p) + (b \cdot 2^i - a \cdot 2^i) < 0$

То есть

$$a * 2^i = k * p + (a \cdot 2^i)$$

$$b * 2^i = k * p + (b \cdot 2^i)$$

$$a * 2^{i+1} = 2 * k * p + 2 * (a \cdot 2^i)$$

$$b * 2^{i+1} = 2 * k * p + 2 * (b \cdot 2^i)$$

$$2^{i+1} * (b - a) = 2 * (b \cdot 2^i - a \cdot 2^i) > 0.$$

Так как  $0 < (a \cdot 2^i) < (a \cdot 2^i) * 2 < k$ ,  $0 < (b \cdot 2^i) < (b \cdot 2^i) * 2 < k$   
 и  $a \cdot 2^i < b \cdot 2^i$ , получаем, что  $2^{i+1} * (b - a) < k$ .

- 2) Ответ на запрос  $(2^i, 2^{i+1})$  равен 1, значит  $0 < (a \cdot 2^i) < k \leq 2 * (a \cdot 2^i) < 2 * k$ ,  $0 < (b \cdot 2^i) < k \leq 2 * (b \cdot 2^i) < 2 * k$ , то есть  $0 < (a \cdot 2^i) < (b \cdot 2^i) < k \leq 2 * (a \cdot 2^i) < 2 * (b \cdot 2^i) < 2 * k \implies$   
Из  $a \cdot 2^{i+1} < b \cdot 2^{i+1}$ .

Теперь осталось показать, что  $0 < 2^{i+1} * (b - a) < k$ .

$$a * 2^i = k * p + (a \cdot 2^i)$$

$$b * 2^i = k * q + (b \cdot 2^i)$$

Аналогично пункту 1 показываем, что  $p = q$ . Значит

$$a * 2^i = k * p + (a \cdot 2^i)$$

$$b * 2^i = k * p + (b \cdot 2^i)$$

$$a * 2^{i+1} = 2 * k * p + 2 * (a \cdot 2^i) = (2 * p + 1) * k + x, 0 \leq x < k$$

$$b * 2^{i+1} = 2 * k * p + 2 * (b \cdot 2^i) = (2 * p + 1) * k + y, x < y < k$$

$$0 < 2^{i+1} * (b - a) = y - x < k.$$

- 3) Ответ на запрос  $(2^i, 2^{i+1})$  равен 0, значит  $(a \cdot 2^i) = 2 \cdot (a \cdot 2^i)$ ,  $(b \cdot 2^i) = 2 \cdot (b \cdot 2^i) \implies a \cdot 2^i = b \cdot 2^i = 0$ , что противоречит предположению индукции  $a \cdot 2^i < b \cdot 2^i$ . Следовательно, такого ответа на запрос не может быть.

Значит для любого  $i = 0, \dots, \text{ceil}(\log_2 k)$  верно  $0 < 2^i * (b - a) < k$ . Но с другой стороны, так как  $a < b \implies 2^{\text{ceil}(\log_2 k)} * (b - a) \geq k$ . Противоречие. Следовательно, не найдутся такие  $a, b$  ( $a < b$ ), что ответы для них полностью совпали.

Поэтому по ответам на указанные выше  $\text{ceil}(\log_2 k)$  запросов коэффициент  $q$  восстанавливается однозначно.  $\square$

Докажем теорему 4

*Доказательство.* Для нахождения номеров существенных переменных воспользуемся теоремой 3 со следующим изменением: вместо запроса на значение  $f(a)$  будем делать запрос на сравнение  $(a, (0 \dots 0))$ . Заметим, что везде в процессе расшифровки запросами на значение было важно только то, что ответ на запрос отличен от нуля или нет, поэтому замена запроса на

Параметро-эффективная расшифровка линейных функций  $k$ -значной логики

значение таким запросом на сравнение это свойство ответов на запросы не изменит.

Но теорема 3 найдет нам только номера существенных переменных. Для  $k = 2$  этого достаточно, чтобы считать, что алгоритм расшифровал функцию. Но для  $k > 2$  недостаточно. Поэтому для  $k > 2$  применяя лемму 12 к каждой существенной переменной, найдем коэффициенты перед ними.

Итого, потратим  $1 + (p - 1) \cdot (\lceil \log_2(k - 1) \rceil + \lceil \log_2(n - 1) \rceil) + p \lceil \log_2 n \rceil$  запросов на сравнение на нахождение номеров существенных переменных и для  $k > 2$  еще  $p \cdot \text{ceil}(\log_2 k)$  запросов на сравнение на нахождение значений коэффициентов перед существенными переменными.  $\square$

## Нижние оценки.

### Расшифровка запросами на значение.

Докажем теорему 5

*Доказательство.* Множество  $\Phi(k, n, p)$  имеет мощность  $\binom{n}{p} \cdot (k - 1)^p$ . По лемме 1 имеем, что

$$\begin{aligned} \varphi_V(k, n, p) &\geq \log_2 \left( \binom{n}{p} \cdot (k - 1)^p \right) \geq p \log_2(k - 1) + \\ &+ p \cdot \log_2(n - p + 1) - \log_2 p \end{aligned}$$

Лемма доказана.  $\square$

### Расшифровка запросами на сравнение.

Приведем классическую мощностную нижнюю оценку.

**Лемма 13** (мощностная нижняя оценка). *Для любого класса функций  $\Psi(k)$ , любого алгоритма  $A$ , расшифровывающего класс  $\Psi(k)$ , для любого натурального  $n$  существует такая функция  $f$  из  $\Psi^n(k)$ , что  $\varphi_C(A, f) \geq \lceil \log_2 |\Psi^n(k)| \rceil$ .*

*Доказательство.* Любой запрос  $(a, b)$  на сравнение функции разбивает множество функций  $\Psi^n(k)$  на два подмножества: те функции, для которых  $\text{sign}(f(a) - f(b))$  совпадает с ответом на запрос, и те, для которых это значение не совпадает с ответом на запрос.

Чтобы найти функцию, которую произвольным фиксированным алгоритмом трудно расшифровать, нам достаточно “прятать” искомую функцию в множестве большей мощности. В этом случае каждый запрос будет сокращать множество потенциальных функций в лучшем случае в 2 раза. И, значит, если число запросов будет меньше чем  $\lceil \log_2 |\Psi^n(k)| \rceil$ , то расшифровываемая функция будет находиться в множестве мощности не менее 2, а, значит, еще не определена однозначно.

Лемма доказана. □

Докажем теорему 6

*Доказательство.* Множество  $\Phi(k, n, p)$  имеет мощность  $\binom{n}{p} \cdot (k-1)^p$ . По лемме 5 имеем, что

$$\begin{aligned} \varphi_C(k, n, p) \geq \log_2 \left( \binom{n}{p} \cdot (k-1)^p \right) &\geq p \log_2 (k-1) + \\ &+ p \cdot \log_2 (n-p+1) - \log_2 p \end{aligned}$$

Лемма доказана. □

## Список литературы

- [1] Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов. Издательство «Наука», Москва, 1985.
- [2] Алешин С.В. Полугруппы и группы автоматов // Интеллектуальные системы. — 2013. — Т. 17, вып. 1–4. — С. 129–141.
- [3] Иванов И.Е. О некоторых свойствах автоматов с магазинной памятью // Интеллектуальные системы. — 2014. — Т. 18, вып. 1. — С. 243–252.

- [4] Часовских А.А. Условия полноты линейно- $p$ -автоматных функций // Интеллектуальные системы. — 2014. — Т. 18, вып. 3. — С. 203–252.
- [5] Александров Д.Е. Об оценках автоматной сложности распознавания классов регулярных языков // Интеллектуальные системы. — 2014. — Т. 18, вып. 4. — С. 161–190.
- [6] Гасанов Э.Э. Прогнозирование периодических сверхсобытий автоматами // Интеллектуальные системы. — 2015. — Т. 19, вып. 1. — С. 23–34.
- [7] Иванов И.Е. О сохранении периодических последовательностей автоматами с магазинной памятью с однобуквенным магазином // Интеллектуальные системы. — 2015. — Т. 19, вып. 1. — С. 145–160.
- [8] Летуновский А.А. Выразимость линейных автоматов относительно расширенной суперпозиции // Интеллектуальные системы. — 2015. — Т. 19, вып. 1. — С. 161–170.
- [9] Гербус В.Г. О связи функций автомата и автоматной функции // Интеллектуальные системы. — 2015. — Т. 19, вып. 2. — С. 109–116.
- [10] Миронов А.М. Критерий реализуемости функций на строках вероятностными автоматами Мура с числовым выходом // Интеллектуальные системы. — 2015. — Т. 19, вып. 2. — С. 149–160.
- [11] Терехина И.Ю. Модель невлияния для квантовых автоматов // Интеллектуальные системы. — 2015. — Т. 19, вып. 2. — С. 183–190.
- [12] Пантелеев П.А. Об отличимости состояний решетчатых автоматов // Интеллектуальные системы. — 2004. — Т. 8, вып. 1–4. — С. 529–542.
- [13] Кирнасов А.Е. Об отношении сложностей условного и безусловного установочного экспериментов // Интеллектуальные системы. — 2005. — Т. 9, вып. 1–4. — С. 433–444.

- [14] Уваров Д.В. О сложности кратных диагностических экспериментов для подмножеств состояний автоматов // Интеллектуальные системы. — 2005. — Т. 9, вып. 1–4. — С. 485–504.
- [15] Кудрявцев В.Б., Грунский И.С., Козловский В.А. Анализ и синтез автоматов по их поведению // Интеллектуальные системы. — 2006. — Т. 10, вып. 1–4. — С. 345–448.
- [16] Пантелеев П.А. Об отличимости состояний автомата при искажениях на входе // Интеллектуальные системы. — 2007. — Т. 11, вып. 1–4. — С. 653–678.
- [17] Angluin D. Queries and Concept Learning // Machine Learning. Vol. 2. 1988. P. 319–342.
- [18] Ансель Ж. О числе монотонных булевых функций  $n$  переменных. — Кибернетический сборник, новая серия, вып. 5, 1968, С. 53–57.
- [19] Sokolov N.A. (1982). On the optimal evaluation of monotonic Boolean functions // USSR Computational Mathematics and Mathematical Physics, Volume 22, Issue 2, 1982, Pages 207–220.
- [20] Damaschke, P. (2003). On Parallel Attribute-Efficient, Learning // Journal of Computer and System Sciences, Volume 67, Issue 1, August 2003, 46–62.
- [21] Осокин В. В. О расшифровке монотонных булевых функций с несущественными переменными // Дискретная математика, 22:3 (2010), 134–145.
- [22] Осокин В.В. Асимптотически оптимальный алгоритм расшифровки разбиения булевого куба на подкубы // Интеллектуальные системы. — 2007. — Т. 11, вып. 1–4. — С. 635–652.
- [23] Осокин В. В. О сложности расшифровки разбиения булевого куба на подкубы // Дискретная математика, 20:2 (2008), 46–62.

- [24] Воронин Б.В., Осокин В.В. О сложности расшифровки существенных переменных функции, задающей разбиение булевого куба // Интеллектуальные системы. — 2008. — Т. 12, вып. 1–4. — С. 159–178.
- [25] Осокин В.В. О параллельной расшифровке разбиений булевого куба // Интеллектуальные системы. — 2009. — Т. 13, вып. 1–4. — С. 427–454.
- [26] Осокин В.В. О параллельной параметро-эффективной расшифровке псевдо-булевских функций // Интеллектуальные системы. — 2010. — Т. 14, вып. 1–4. — С. 429–458.
- [27] Zolotykh N.Yu., Shevchenko V.N. (1997). Lower Bounds for the Complexity of Learning Half-Spaces with Membership Queries // ALT'98, Otzenhausen, Germany, October 8-10, 1998.
- [28] Nakamura A., Abe N. (1995) Exact learning of linear combinations of monotone terms from function value queries // Theoretical Computer Science, Volume 137, Issue 1, 159-176, 1995.
- [29] Гасанов Э.Э., Ниязова З.А. Расшифровка арифметических сумм малого числа монотонных конъюнкций // Материалы XI Международного семинара Дискретная математика и ее приложения (Москва, 18-23 июня 2012 г.). — Изд-во механико-математического ф-та МГУ Москва, 2012. — С. 335–337.
- [30] Ryuhei Uehara, Kensei Tsuchida, and Ingo Wegener. Optimal Attribute-Efficient Learning Of Disjunction, Parity, And Threshold Functions // EuroCOLT '97 Proceedings of the Third European Conference on Computational Learning Theory, 1997.
- [31] Adam R. Klivans, Rocco A. Servedio. Toward Attribute Efficient Learning of Decision Lists and Parities // The Journal of Machine Learning Research Volume 7, 2006.

- [32] Vitaly Feldman. On Attribute Efficient and Non-adaptive Learning of Parities and DNF Expressions // The Journal of Machine Learning Research Volume 8, 2007
- [33] Valiant L. G. A theory of the learnable // ACM Press New York, NY, USA, 1984, Volume 27, Issue II, 1134-1142.
- [34] Blum A. Learning a Function of  $r$  Relevant Variables // COLT 2003, Open problems.
- [35] Arpe J., Reischuk B. Learning Juntas in the Presence of Noise // Theoret. Comput. Sci. 384(1): 2-21, 2007.
- [36] Костюченко О.В. Сплайновая интерполяция с плавающими узлами // Интеллектуальные системы. — 2007. — Т. 11, вып. 1–4. — С. 715-720.
- [37] Гасанов Э.Э. Расшифровка линейных функций ранжирования // Материалы XI Международного семинара "Дискретная математика и ее приложения посвященного 80-летию со дня рождения академика О.Б.Лупанова (Москва, 18-23 июня 2012 г.). Изд-во мех-мат фак-та МГУ. 2012. С. 332-334.
- [38] Хегай С.И. Расшифровка полиномиальных функций ранжирования // Интеллектуальные системы. 2015. 19:1. 213 – 230.
- [39] Быстрыгова А.В. Сложность расшифровки линейных булевых функций // Интеллектуальные системы, 2015, Т. 19, Вып. 3, С. 101-126
- [40] Thomas Hofmeister An Application of Codes to Attribute-Efficient Learning // EuroCOLT'99 Proceedings of the 4th European Conference on Computational Learning Theory, 1999.
- [41] Чашкин А.В. Лекции по дискретной математике. Учебное пособие // МГУ им.М.В.Ломоносова, Мех.-мат. факультет, Москва, 2007.