Запросы на сравнение в задаче точной расшифровки замкнутых классов Поста

А. В. Быстрыгова[1]

В данной работе рассматривается точная параметро-эффективная расшифровка замкнутых классов Поста запросами на сравнение. Для всех классов приведены оценки сложности расшифровки.
Ключевые слова: точная расшифровка, параметро-эффективная расшифровка, замкнутые классы Посты, запросы на сравнение.

1. Введение

Точная расшифровка булевых функций решетки Поста является объектом исследования данной работы. Рассматриваемая задача является по сути игрой между *учителем* и *учеником*. Учитель выбирает функцию одного из классов решетки, ученику известен сам класс и то, что у функции n переменных и не более k из них являются существенными. Ученик задает учителю запросы, учитель на них честно отвечает. Анализируя ответы учителя, ученик должен понять выбор учителя. В данной работе рассматривается версия задачи, в которой ученик может задавать учителю только запросы на сравнение, которые были введены в литературу сравнительно недавно [1]. Каждый запрос на сравнение — пара наборов, на которую учитель отвечает, чему равен знак разности выбранной им функции на этой паре.

При этом рассматривается задача параметро-эффективной расшифровки, поскольку для многих классов учитывается малость k относительно n. Например, при расшифровке классов функций логических сумм рассматриваются $k = o(n)$, а для классов линейных функций считается, что $\log_2 k = o(\log_2 n)$.

Стоит отметить, что данная задача уже исследовались с точки зрения сложности расшифровки запросами на значение [2, 3, 4, 5]. Поэтому интерес представляет разница результатов рассматриваемой задачи для запросов на значение, полученные в этих работах, и запросов на сравнение, рассматриваемые в этой.

В работе [6] уже рассматривалось применение запросов на сравнение в задаче параметро-эффективной расшифровки замкнутых классов

[1] *Быстрыгова Анастасия Викторовна* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: anastasiya.bistrigova@yandex.com.

Bistrigova Anastasiya Viktorovna — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

Поста. Было показано, что не все классы решетки Поста можно расшифровать запросами на сравнение, то есть нельзя привести алгоритм расшифровки запросами на сравнение, который определит, какая функция загадана. Оказалось, что класс нельзя расшифровать запросами на сравнение тогда и только тогда, когда в классе содержатся обе константы $0, 1$. Более того, для любого класса решетки Поста алгоритм расшифровки запросами на значение можно преобразовать, не меняя количество запросов, в алгоритм расшифровки запросами на сравнение. Поэтому сложность расшифровки любого класса решетки Поста, который можно расшифровать запросами на сравнение, не больше сложности расшифровки этого же класса запросами на значение. К тому же существует классы (классы группы O), для которых эта сложность строго меньше в случае запросов на сравнение.

Если в работе [6] в основном был сделан упор на верхние оценки сложности расшифровки замкнутых классов Поста, то в данной работе приводятся нижние оценки, позволяющие говорить в целом о характере поведения сложности расшифровки классов решетки Поста.

2. Основные определения и утверждения

2.1. Базовые определения

Если a — вещественное число, под $]a[$ будем понимать наименьшее целое, не меньшее a, под $[a]$ — наибольшее целое, не большее a.

Будем говорить, что $A(n), B(n)$ асимптотически равны и писать $A(n) \sim B(n)$, если $\lim_{n\to\infty} \frac{A(n)}{B(n)} = 1$. Будем писать $A(n) = o(B(n))$, если $\lim_{n\to\infty} \frac{A(n)}{B(n)} = 0$, и $A(n) \gtrsim B(n)$, если $\lim_{n\to\infty} \frac{A(n)}{B(n)} > 0$. Если $A(n) \gtrsim B(n), B(n) \gtrsim A(n)$, то будем говорить, что $A(n)$ и $B(n)$ равны по порядку при $n \to \infty$ и обозначать $A(n) \asymp B(n)$.

Бинарной покрывающей матрицей $P(n, k)$ будем называть любую бинарную матрицу с n столбцами, такую, что в любых ее k столбцах встретятся все 2^k двоичных наборов. Среди всех бинарных покрывающих матриц для чисел n, k выберем ту матрицу, у которой число строк наименьшее, это количество строк обозначим через $\alpha(n, k)$.

Обозначим через $S_{n,k}$ значение $\max_{p\in\mathbb{N}, 1\le p<k} (2^p - 1)\alpha(n-p, k-p)$.

2.2. Исследуемые классы функций

В данной работе исследуются все замкнутые классы решетки Поста, представленные на рисунке 1, словесное описание которых можно найти в работах [5, 6]. Для любого из классов Поста R под $R(n, k)$ будем понимать множество функций класса R, зависящих от n переменных, среди которых существенных переменных не более k.

2.3. Сложность расшифровки

Запросом на значение x для функции f является набор x, а ответом на него является значение функции f на наборе x. *Запросом на сравнение* (x, y) будем называть упорядоченную пару наборов x, y, а под ответом на этот запрос понимать знак разности $f(x) - f(y)$.

Будем говорить, что *последовательность запросов расшифровывает загаданную функцию f*, если последовательность конечна, состоит из запросов одного типа и функция f однозначно восстанавливается по ответам на запросы этой последовательности.

В определениях, которые введем далее, через $T \in \{MQ, CQ\}$ будем обозначать тип запроса, где MQ — запрос на значение, CQ — запрос на сравнение.

Алгоритмом расшифровки $A^T_{M,n}$ для запросов типа T будем называть процесс такого задания последовательности запросов типа T, что каждый элемент последовательности выбирается определенным образом в зависимости от ответов учителя на запросы — предыдущие члены последовательности, причем сформированная последовательность расшифровывает загаданную функцию $f \in M(n)$. Через $\mathcal{A}^T_{M,n}$ будем обозначать множество всех алгоритмов $A^T_{M,n}$ расшифровки для запросов типа T.

Обозначим через $q(A, f)$ минимальное количество первых запросов в последовательности запросов алгоритма A, которые расшифровывают функцию f. Тогда под *сложностью расшифровки запросами типа T* будем понимать число запросов типа T, которое придется задать наилучшему алгоритму для расшифровки самой плохой функции. Иными словами, сложность расшифровки запросами типа T задается следующим образом: $\varphi_T(M, n) = \min_{A \in \mathcal{A}^T_{M,n}} \max_{f \in M(n)} q(A, f)$.

Будем говорить, что *алгоритм расшифровывает функцию* $f \in F(n, k)$, если ответы на запросы, заданные алгоритмом, однозначно определяют вектор значений функции $f \in F(n, k)$.

Будем говорить, что *класс $F(n, k)$ можно расшифровать запросами на сравнение*, если существует алгоритм расшифровки запросами на сравнение, который для любой функции $f \in F(n, k)$ расшифровывает f. Если про класс $F(n, k)$ нельзя сказать, что его можно расшифровать запросами на сравнение, тогда будем, говорить, что *класс $F(n, k)$ нельзя расшифровать запросами на сравнение*.

Будем говорить, что *класс F можно расшифровать запросами на сравнение*, если для любых целых n, k, где $n \geq k \geq 1$, можно расшифровать запросами на сравнение класс $F(n,k)$. Если про класс F нельзя сказать, что его можно расшифровать запросами на сравнение, тогда будем, говорить, что *класс F нельзя расшифровать запросами на сравнение.*

2.4. Результаты

Под условной оценкой по порядку будем понимать то, что оценка по порядку равна величине, связанной с $\alpha(n,k)$, порядок, а тем более асимптотику, которой мы не знаем, хотя как показано в [5] исследование $\alpha(n,k)$ ведется уже несколько десятелетий.

Напомним, что классы из "правой" половины решетки Поста заведомо упускаются из дальнейшего рассмотрения, так как они являются двойственными к классам из "левой" половины, следовательно задача расшифровки классов из "правой" половины сводится к задаче расшифровки классов из "левой" половины.

В [6] было показано, что классы $C_1, A_1, L_1, S_6, O_9, O_6$ нельзя расшифровать запросами на сравнение, потому что они содержат обе константы. Чтобы не терять полностью из рассмотрения эти замкнутые классы лишь по тому, что они содержат обе константы, далее все же будем рассматривать эти классы, но без константы 0, и помечать символом *. Причем заметим, что замыкание класса C_1^* совпадает с C_1, L_1^* — с L_1, O_9^* — с O_9, а $A_1^* = A_2, S_6^* = S_3, O_6^* = O_5$.

Стоит отметить, что верхние оценки для классов получаются использованием теоремы 2 работы [6], в которой утверждается, что сложность расшифровки фиксированного класса запросами на сравнение не больше сложности расшифровки этого класса запросами на значение. Нижние оценки получаются адаптацией доказательств нижних оценок этих классов для запросов на значение из [5]. Если в случае запросов на значение для класса нижняя оценка мощностная, то в случае запросов на сравнение она по порядку такая же. Если нижняя оценка получалась конструктивным образом, то и для запросов на сравнение она конструктивная, причем в качестве загадаваемой функции учитель выбирает для запросов на сравнение такие же, как и в случае запросов на значение.

Теорема 1. *Замкнутые классы решетки Поста по характеру известной на данный момент сложности точной расшифровки запросами на сравнение разделены на четыре группы в случае $n, k \to \infty$:*

1) точная оценка

- $\varphi_{CQ}(O_1, n, 1) = \,]\log_3 n[$;

2) асимптотика

- $\varphi_{CQ}(O_4, n, 1) \sim \log_3 n$ *при* $n \to \infty$;
- $\varphi_{CQ}(O_5, n, 1) \sim \log_3 n$ *при* $n \to \infty$;
- $\varphi_{CQ}(O_8, n, 1) \sim \log_3 n$ *при* $n \to \infty$;
- $\varphi_{CQ}(O_9^*, n, 1) \sim \log_3 n$ *при* $n \to \infty$;

3) условный порядок

- $\varphi_{CQ}(C_1^*, n, k) \asymp \alpha(n, k)$ *при* $n, k \to \infty, k = o(n)$;
- $\varphi_{CQ}(C_2, n, k) \asymp \alpha(n, k)$ *при* $n, k \to \infty, k = o(n)$;
- $\varphi_{CQ}(C_4, n, k) \asymp \alpha(n-1, k-1)$ *при* $n, k \to \infty, k = o(n)$;
- $\varphi_{CQ}(D_1, n, k) \asymp \alpha(n-1, k-1)$ *при* $n, k \to \infty, k = o(n)$;
- $\varphi_{CQ}(D_3, n, k) \asymp \alpha(n-1, k-1)$ *при* $n, k \to \infty, k = o(n)$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_1^i, n, k) \asymp S_{n,k}$ *при* $n, k \to \infty$, $k = o(n)$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_4^i, n, k) \asymp \alpha(n, k)$ *при* $n, k \to \infty$, $k = o(n)$.

4) порядок

- $\varphi_{CQ}(A_2, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $n, k \to \infty$;
- $\varphi_{CQ}(A_4, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $n, k \to \infty$;
- $\varphi_{CQ}(D_2, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $n, k \to \infty$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_2^i, n, k) \asymp k \log n + \frac{2^k}{\sqrt{k}}$ *при* $n, k \to \infty$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_3^i, n, k) \asymp k \log n + \frac{2^k}{\sqrt{k}}$ *при* $n, k \to \infty$;
- $\varphi_{CQ}(S_1, n, k) \asymp k \log(n/k)$ *при* $n, k \to \infty, k = o(n)$;
- $\varphi_{CQ}(S_3, n, k) \asymp k \log(n/k)$ *при* $n, k \to \infty, k = o(n)$;
- $\varphi_{CQ}(S_5, n, k) \asymp k \log(n/k)$ *при* $n, k \to \infty, k = o(n)$;
- $\varphi_{CQ}(L_1^*, n, k) \asymp k \log n$ *при* $n, k \to \infty, \log k = o(\log n)$;
- $\varphi_{CQ}(L_2, n, k) \asymp k \log n$ *при* $n, k \to \infty, \log k = o(\log n)$;
- $\varphi_{CQ}(L_4, n, k) \asymp k \log n$ *при* $n, k \to \infty, \log k = o(\log n)$;
- $\varphi_{CQ}(L_5, n, k) \asymp k \log n$ *при* $n, k \to \infty, \log k = o(\log n)$.

На рисунке 1 схематично приведены основные результаты этой теоремы. Если класс выделен квадратом с обеими диагоналями, тогда этот

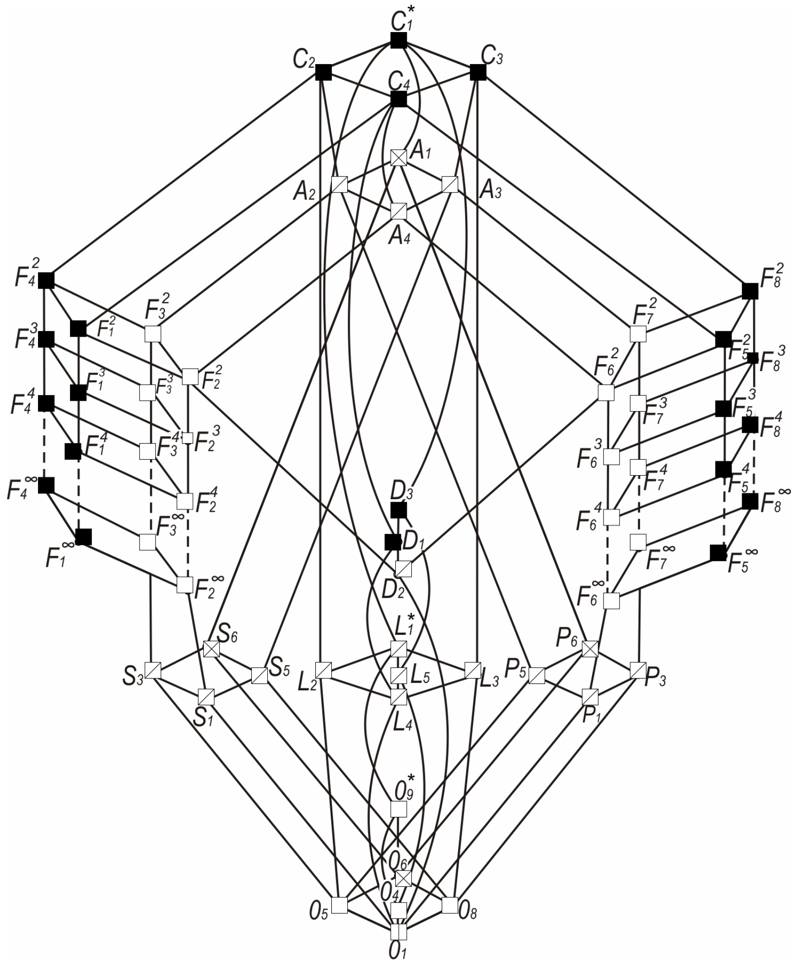

Рис. 1. Результаты сложности расшифровки замкнутых классов Поста запросами на сравнение при $n, k \to \infty$.

класс, согласно [6], нельзя расшифровать запросами на сравнение. Если класс выделен вертикальной чертой в квадрате, то имеется точная оценка. Если класс выделен белым, то для него получена асимптотическая оценка, если обозначен косой линией внутри квадрата, то оценка по порядку, если черным, то условная оценка по порядку.

Теорема 2. *Замкнутые классы решетки Поста по характеру известной на данный момент сложности точной расшифровки запросами на сравнение в случае, если k не меняется и n растет, разделены на три группы:*

1) точная оценка

- $\varphi_{CQ}(O_1, n, 1) = \,]\log_3 n[$;

2) асимптотика

- $\varphi_{CQ}(O_4, n, 1) \sim \log_3 n$ *при* $n \to \infty$;
- $\varphi_{CQ}(O_5, n, 1) \sim \log_3 n$ *при* $n \to \infty$;
- $\varphi_{CQ}(O_8, n, 1) \sim \log_3 n$ *при* $n \to \infty$;
- $\varphi_{CQ}(O_9^*, n, 1) \sim \log_3 n$ *при* $n \to \infty$;

3) порядок

- $\varphi_{CQ}(C_1^*, n, k) \asymp \log n$ *при* $n \to \infty, k \geq 2$;
- $\varphi_{CQ}(C_2, n, k) \asymp \log n$ *при* $n \to \infty, k \geq 2$;
- $\varphi_{CQ}(C_4, n, k) \asymp \log n$ *при* $n \to \infty, k \geq 2$;
- $\varphi_{CQ}(A_2, n, k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{CQ}(A_4, n, k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{CQ}(D_1, n, k) \asymp \log n$ *при* $n \to \infty, k \geq 2$;
- $\varphi_{CQ}(D_2, n, k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{CQ}(D_3, n, k) \asymp \log n$ *при* $n \to \infty, k \geq 2$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_1^i, n, k) \asymp \log n$ *при* $n \to \infty$, $k \geq 2$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_2^i, n, k) \asymp \log n$ *при* $n \to \infty$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_3^i, n, k) \asymp \log n$ *при* $n \to \infty$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_4^i, n, k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{CQ}(S_1, n, k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{CQ}(S_3, n, k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{CQ}(S_5, n, k) \asymp \log n, n \to \infty$;

- $\varphi_{CQ}(L_1^*, n, k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{CQ}(L_2, n, k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{CQ}(L_4, n, k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{CQ}(L_5, n, k) \asymp \log n$ *при* $n \to \infty$.

3. Вспомогательные определения и утверждения

Через $M_i(n, k), i \in \{0, 1\}$, будем обозначать любую бинарную матрицу с наименьшим числом строк и n столбцами, такую, что в любых k ее столбцах встретятся все 2^k наборов кроме быть может набора, все компоненты которого равны i. Назовем ее почти покрывающей типа i, число строк в ней обозначим за $\beta_i(n, k)$.

Выберем произвольные различные k переменных из n и присвоим выбранным переменным произвольным образом значения из $\{0, 1\}$. Полученные k переменных с присвоенными им значениями назовем *фиксацией*. Будем говорить, что *запрос содержит фиксацию*, если переменные в фиксации имеют такие же значения, как и эти же переменные в запросе.

Для удобства приведем следующие две леммы, которые объединили в себе формулировки лемм 2–5 работы [5], на которые мы будем часто ссылаться в следующих разделах.

Лемма 1. *Справедливо неравенство* $\beta_i(n, k) \geq \alpha(n, k) - 1$, $i \in \{0, 1\}$.

Лемма 2. *Справедливы следующие соотношения.*

1) При $k = o(n)$, $k, n \to \infty$

$$k \log_2 n = o(\alpha(n, k)).$$

2) При $1 \leq p < k - 1, k = o(n)$, $k, n \to \infty$

$$k \log_2 n = o((2^p - 1)\alpha(n - p, k - p)).$$

3) При $k = o(n), k, n \to \infty$

$$S_{n,k} = \max_{p \in \mathbb{N}, 1 \leq p < k} (2^p - 1)\alpha(n - p, k - p) > (2^{k-1} - 1)\alpha(n - (k - 1), k - (k - 1)).$$

Лемма 3. *При* $k \geq 2$, $n \to \infty$ *справедливо соотношение* $\alpha(n, k) \asymp \log n$.

Доказательство. Доказательство следует из нижней оценки, полученной в [7], и верхней оценки из работы [8], которые имеют место для $k \geq 2$ и $n \to \infty$:

$$\alpha(n,k) \geq 2^{k-2} \cdot \alpha(n-k+2,2) = 2^{k-2} \log_2 (n-k+2)(1+o(1)),$$

$$\alpha(n,k) \leq (1+o(1))) \frac{k-1}{\log_2 \frac{2^k}{2^k-1}} \cdot \log_2 n.$$

□

Лемма 4. *При $k \geq 2$ справедливо соотношение $S_{n,k} \asymp \log n$ при $n \to \infty$.*

Доказательство. Доказательство следует из определения $S_{n,k}$ и лемм 2, 3. □

4. Классы C_i

Лемма 5. *Имеет место неравенство*

$$\varphi_{CQ}(C_2, n, k) \geq [0.5(\alpha(n,k) - 1)].$$

Доказательство. Для получения нижней оценки на первые $[0.5(\alpha(n,k) - 1)] - 1$ запросов ученика будем отвечать 0. Пока ученик не опросит все фиксации k переменных из n, он не поймет ему загадали константу 1 или функцию с ровно одним нулем в векторе значений, полученном при удалении всех фиктивных переменных. Так как $f \in C_2(n,k)$, то $f(1,\ldots,1) = 1$, следовательно ученику не интересны фиксации k переменных из n из одних единиц. Следовательно, ученик должен опросить все строки почти покрывающей матрицы типа 1. Чтобы покрыть запросами на сравнение все наборы почти покрывающей матрицы типа 1, необходимо задать как минимум $[0.5(\alpha(n,k) - 1)]$, что следует из леммы 1 и факта, что в один запрос на сравнение можно включить две строки почти покрывающей матрицы типа 1. □

Лемма 6. *При $n > 1, k > 1$ имеет место неравенство*

$$\varphi_{CQ}(C_4, n, k) \geq \alpha(n-1, k-1) - 1.$$

Доказательство. Аналогично доказательству для запросов на значение в [5] необходимо заставить ученика запросами на сравнение покрыть n-местные бинарные наборы двух типов:

1) строки покрывающей матрицы $M_0(n-1,k-1)$ типа 0, где столбцам матрицы соответствуют переменные $x_2,\dots,x_n$, а $x_1=0$,

2) строки покрывающей матрицы $M_1(n-1,k-1)$ типа 1, где столбцам матрицы соответствуют переменные $x_2,\dots,x_n$, а $x_1=1$.

Играем за учителя. Ученику сразу расскажем, что x_1 — существенная переменная. На первые $\alpha(n-1,k-1)-2$ запросов ученика будем отвечать следующим образом. Если запрос имеет вид $(a_1a_2a_3\dots a_n, b_1b_2b_3\dots b_n)$, тогда ответ на него $sign(a_1-b_1)$.

Учитывая лемму 1, заметим, что своими $\alpha(n-1,k-1)-2$ запросами на сравнение ученик покрывает максимум $2\alpha(n-1,k-1)-4$ различных строк упомянутого типа, а для того, чтобы покрыть все строки, ему необходимо задать $\beta_0(n,k)+\beta_1(n,k)\geq 2\alpha(n,k)-2$. Соответственно, какая-то из фиксаций не покрыта запросами на сравнение и значение функции на этом наборе k переменных можно задать любым способом, поэтому обе функции, используемые при доказательстве нижней оценки в случае запросов на значение в [5], будут удовлетворять ответам учителя на $\alpha(n-1,k-1)-2$ запросов на сравнение. □

Теорема 3. *Справедливы соотношения*

- $\varphi_{CQ}(C_1^*,n,k)\asymp\alpha(n,k), n,k\to\infty, k=o(n)$;
- $\varphi_{CQ}(C_2,n,k)\asymp\alpha(n,k), n,k\to\infty, k=o(n)$;
- $\varphi_{CQ}(C_4,n,k)\asymp\alpha(n-1,k-1), n,k\to\infty, k=o(n)$.

Доказательство. Доказательство первого и второго пункта теоремы следует из верхней оценки $\alpha(n,k)+k\log_2 n$, полученной в [5], и лемм 5, 2.

Доказательство третьего пункта теоремы следует из верхней оценки $2\alpha(n-1,k-1)+k\log_2 n$, полученной в [5], леммы 6 и второго пункта леммы 2 при $p=1$. □

Теорема 4. *При $k\geq 2$ справедливы следующие соотношения.*

- $\varphi_{CQ}(C_1^*,n,k)\asymp\log n, n\to\infty$;
- $\varphi_{CQ}(C_2,n,k)\asymp\log n, n\to\infty$;
- $\varphi_{CQ}(C_4,n,k)\asymp\log n, n\to\infty$.

Доказательство. Доказательство следует из доказательство теоремы 3, а также леммы 3. □

5. Классы A_i

Теорема 5. *Справедливы соотношения*

- $\varphi_{CQ}(A_2, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $k, n \to \infty$*;*
- $\varphi_{CQ}(A_4, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $k, n \to \infty$*.*

Доказательство. Верхняя оценка для классов A_2, A_4 следует из теоремы 3 из [5]. Нижняя оценка, полученная в [4], для класса $A_1(n, k)$, как и для классов A_2, A_4 в [5], мощностная. Мощностная нижняя оценка в случае запросов на значение получается логарифмированием количества функций из класса по основанию 2. В случае запросов на сравнение это основание заменяется на 3, поэтому оценка сложности расшифровки при $k, n \to \infty$ по порядку остается той же $\frac{2^k}{\sqrt{k}} + k \log n$. □

Как следствие из этой теоремы получаем следующую теорему.

Теорема 6. *Справедливы соотношения*

- $\varphi_{CQ}(A_2, n, k) \asymp \log n$ *при* $n \to \infty$*;*
- $\varphi_{CQ}(A_4, n, k) \asymp \log n$ *при* $n \to \infty$*.*

6. Классы D_i

Лемма 7. *Для* $k > 1$ *имеет место неравенство*

$$\varphi_{CQ}(D_3, n, k) \geq \big\lceil 0.5\alpha(n-1, k-1) \big\rceil.$$

Доказательство. На все $\big\lceil 0.5\alpha(n-1, k-1)\big\rceil - 1$ запросов ученика будем отвечать следующим образом. Если запрос на сравнение имеет вид $(a_1 a_2 \ldots a_n, b_1 b_2 \ldots b_n)$, тогда ответим $sign(a_1 - b_1)$. Тогда аналогично доказательству подобной леммы для запросов на значение в [5] утверждается, что ученик обязан своими запросами покрыть все строки какой-то покрывающей матрицы $\alpha(n-1, k-1)$, где столбцы матрицы соответствуют переменным $x_2, x_3, \ldots, x_n$. По сути, в начале игры ученику раскрывается информация о том, что переменная x_1 точно существенная, а ему уже остается своими запросами понять, имеются ли еще какие-то существенные переменные помимо x_1.

Рассмотрим запросы, которые задает ученик. Каждый набор из запроса преобразуем следующим образом:

- если $x_1 = 0$ в наборе, то оставим набор без изменений;
- если $x_1 = 1$ в наборе, то инвертируем все переменные в нем.

Теперь рассмотрим все преобразованные запросы. Всего наборов ровно $2\lceil 0.5\alpha(n-1,k-1)\rceil - 2$. Значит ученик не опросил какую-то фиксацию $k-1$ переменной из переменных $x_2, x_3, \dots, x_n$. Фиксируем одну такую фиксацию, а затем зафиксируем любой n-местный набор, содержащий эту фиксацию. Заметим, что значение функции на этом наборе можно зафиксировать любым способом, то есть либо 0, либо 1.

Если значение на этом наборе установить равным значению переменной x_1, тогда всем $\lceil 0.5\alpha(n-1,k-1)\rceil - 1$ запросам ученика будет удовлетворять функция $x_1 \in D_3(n,k)$.

Если значение на этом наборе установить равным значению $\overline{x_1}$, тогда всем $\lceil 0.5\alpha(n-1,k-1)\rceil - 1$ запросам ученика будет удовлетворять функция, у которой существенными являются только все переменные зафиксированной фиксации и переменная x_1. Иными словами, во втором случае получилась бы самодвойственная функция, у которой при выбрасывании всех фиктивных переменных ($n-k$ штук) вектор значений в первой его половине содержит ровно одну единицу, то есть функция принадлежит классу $D_3(n,k)$.

Таким образом, не задав последнего $\lceil 0.5\alpha(n-1,k-1)\rceil$-го запроса ученик не знает, какая функция загадана: x_1 или функция с ровно k существенными переменными. $\square$

Лемма 8. *Для любого $k > 1$ имеет место неравенство*

$$\varphi_{CQ}(D_1,n,k) \geq \lceil 0.5(\alpha(n-1,k-1)-1)\rceil.$$

Доказательство. Ученик знает, что значение функции на наборе $(0,\dots,0)$ равно 0, а на наборе $(1,\dots,1)$ в силу самодвойственности функции — 1.

На все $\lceil 0.5(\alpha(n-1,k-1)-1)\rceil - 1$ запросов ученика будем отвечать следующим образом. Если запрос на сравнение имеет вид $(a_1a_2\dots a_n, b_1b_2\dots b_n)$, тогда ответим $sign(a_1-b_1)$.

Далее доказательство повторяет доказательство леммы 7 с той лишь разницей, что теперь утверждается, что ученик обязан своими запросами покрыть все $\beta_0(n-1,k-1)$ строки какой-то почти покрывающей матрицы типа 0, где столбцы матрицы соответствуют переменным $x_2, x_3, \dots, x_n$.

Рассмотрим запросы, которые задает ученик. Каждый набор из запроса преобразуем следующим образом:

- если $x_1 = 0$ в наборе, то оставим набор без изменений;
- если $x_1 = 1$ в наборе, то инвертируем все переменные в нем.

Теперь рассмотрим все преобразованные запросы. Всего наборов ровно $2\lceil 0.5(\alpha(n-1,k-1)-1)\rceil - 2$. Согласно лемме 1 верно соотношение

$\beta_0(n-1,k-1) \geq \alpha(n-1,k-1)-1$, поэтому своими $\left[0.5(\alpha(n-1,k-1)-1)\right]-1$ запросами на сравнение ученик покрыл точно не все строки ни одной из почти покрывающих матриц типа 0. Значит ученик не опросил какую-то фиксацию $k-1$ переменной из переменных $x_2, x_3, \ldots, x_n$. Среди неопрошенных фиксаций точно имеется фиксация отличная от полностью нулевой. В противном случае, неопрошенную строку почти покрывающей матрицы типа 0 можно было бы выкинуть из матрицы, а значит исходная матрица была не минимальной. Фиксируем любую неопрошенную ненулевую фиксацию, а затем зафиксируем любой n-местный набор, содержащий эту фиксацию. А далее полностью повторяем доказательство леммы 7. □

Лемма 9. *Имеет место неравенство*

$$\varphi_{CQ}(D_2,n,k) \gtrsim k\log(n/k) + \sqrt{\frac{2}{\pi}} \cdot \frac{2^{k-1}}{\sqrt{k}}$$

при $k, n \to \infty$, $k < n$.

Доказательство. Результат леммы получается заменой основания логарифма с 2 на 3 в доказательстве мощностной нижней оценке сложности расшифровки этого же класса для случая запросов на значение (лемма 11 в [5]). □

Теорема 7. *Имеют место следующие соотношения.*

1) $\varphi_{MQ}(D_3,n,k) \asymp \alpha(n-1,k-1)$ *при* $k, n \to \infty, k = o(n)$.

2) $\varphi_{MQ}(D_1,n,k) \asymp \alpha(n-1,k-1)$ *при* $k, n \to \infty, k = o(n)$.

3) $\varphi_{MQ}(D_2,n,k) \asymp \frac{2^k}{\sqrt{k}} + k\log n$ *при* $k, n \to \infty$.

Доказательство. Верхние оценки для всех классов совпадают с таковыми в случае запросов на значение из [5]. Соотношение для класса D_3 следует из лемм 7, 2 при $p = 1$. Соотношение для класса D_1 следует из лемм 8, 2 при $p = 1$. Нижняя оценка для класса D_2 следует из леммы 9. □

Теорема 8. *Верны следующие соотношения.*

1) $\varphi_{CQ}(D_1,n,k) \asymp \log n$ *при* $n \to \infty, k \geq 2$.

2) $\varphi_{CQ}(D_2,n,k) \asymp \log n$ *при* $n \to \infty$.

3) $\varphi_{CQ}(D_3,n,k) \asymp \log n$ *при* $n \to \infty, k \geq 2$.

Доказательство. Доказательство следует из доказательства теоремы 7 и леммы 3. □

7. Классы $F_i^j, 1 \le j \le 4$

Лемма 10. *Имеет место неравенство*

$$\varphi_{CQ}(F_4^i, n, k) \ge \left[0.5(\alpha(n,k) - 1)\right].$$

Доказательство. Аналогично доказательству леммы 12 из [5], в роли учителя выберем константу 1 и отвечать на все запросы ученика будем в соответствии с выбранной функцией. Тогда своими запросами на сравнение ученик должен покрыть все строки почти покрывающей матрицы типа 1. Если ученик не опросит хотя бы одну строку, будет не опрошена хотя бы одна фиксация. На наборе с этой фиксацией значение может быть как 0, так и 1. Если значение на нем равно 0, то ученик бы понял, что одна из ранее невстречавшихся фиксаций, отличных от фиксации из одних единиц, и содержит все номера существенных переменных. Учитывая то, что один запрос на сравнение покрывает не более двух строк почти покрывающей матрицы типа 1, а также вспоминая лемму 1, получаем неравенство доказываемой леммы. □

Лемма 11. *Имеет место неравенство*

$$\varphi_{CQ}(F_2^\infty, n, k) \gtrsim k \log{(n/k)} + \sqrt{\frac{2}{\pi}} \cdot \frac{2^{k-1}}{\sqrt{k-1}}$$

при $k, n \to \infty$, $k < n$.

Доказательство. Нижняя оценка сложности расшифровки запросами на сравнение класса $F_2^\infty(n,k)$ также как и в случае запросов на значение — мощностная. Поэтому результат леммы получается заменой основания логарифма с 2 на 3 в доказательстве мощностной нижней оценке сложности расшифровки этого же класса для случая запросов на значение. □

Лемма 12. *Пусть* $i > 1, k > 1, 1 \le p < k$, p *— целое. Тогда справедливо неравенство*

$$\varphi_{CQ}(F_1^i, n, k) \ge \left[0.5(2^p - 1)\alpha(n-p, k-p)\right].$$

Доказательство. Доказательство опирается на доказательство леммы 14 в [5]. Учитель отвечает на запросы ученика так, чтобы ответам на все запросы кроме последнего заданного удовлетворяли две функции, которые использовались в доказательстве упомянутой леммы: $x_1|x_2|\dots|x_p$ и функция, отличающаяся от $x_1|x_2|\dots|x_p$ ровно в одном наборе и равная 1 на всех наборах с $x_1 = x_2 = \ldots = x_p = 1$.

Напомним, что в [5] различались следуюшие три категории n-местных наборов $x_1 x_2 \dots x_n$:

1) $x_1 = x_2 = \ldots = x_p = 0$,

2) $x_1 = x_2 = \ldots = x_p = 1$,

3) остальные наборы.

Также напомним, что считается, что группа наборов с фиксированными значениями переменных $x_1, x_2, \ldots, x_p$ опрошена, если были опрошены все фиксации из k переменных, в которые вошли переменные $x_1, x_2, \ldots, x_p$ с этими фиксированными значениями.

Заметим, что мы могли ученику подсказать, что значение функции на наборах группы второй категории равно 1, и он мог не опрашивать их. Тем не менее, все наборы групп первой и третьей категории он обязан задать, потому что не задав хотя бы один из этих запросов, у него нет уверенности, p или больше существенных переменных у загаданной функции. Для того, чтобы опросить все группы наборов за исключение группы наборов второй категории, в случае запросов на значение ученику необходимо было задать $(2^p - 1)\alpha(n-p, k-p)$ запросов. Вспоминая, что один запрос на сравнение включает в себя два набора, получаем неравенство доказываемой леммы. □

Теорема 9. *Справедливы соотношения*

- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_4^i, n, k) \asymp \alpha(n, k)$ *при* $k, n \to \infty, k = o(n)$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_1^i, n, k) \asymp S_{n,k}$ *при* $k, n \to \infty, k = o(n)$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_2^i, n, k) \asymp k \log n + \frac{2^k}{\sqrt{k}}$ *при* $k, n \to \infty$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_3^i, n, k) \asymp k \log n + \frac{2^k}{\sqrt{k}}$ *при* $k, n \to \infty$.

Доказательство. В качестве верхних оценок всех классов берутся верхние оценки сложности расшифровки запросами на значение из работы [5]. Соотношение, касающееся класса $F_4^i(n, k)$, следует из лемм 10, 2. Соотношение, касающееся класса $F_1^i(n, k)$, следует из лемм 12, 2. Соотношения, касающееся классов $F_2^i(n, k), F_3^i(\mathrm{n, k})$, следует из леммы 11, потому что класс $F_2^\infty(n, k)$ вкладывается в классы $F_2^i(n, k), F_3^i(n, k)$. □

Теорема 10. *Справедливы соотношения*

- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_1^i, n, k) \asymp \log n$ *при* $n \to \infty, k \geq 2$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_2^i, n, k) \asymp \log n$ *при* $n \to \infty$;

- *если $i > 1$ или $i = \infty$, то $\varphi_{CQ}(F_3^i, n, k) \asymp \log n$ при $n \to \infty$;*

- *если $i > 1$ или $i = \infty$, то $\varphi_{CQ}(F_4^i, n, k) \asymp \log n$ при $n \to \infty$.*

Доказательство. Доказательство всех пунктов следует из доказательства теоремы 9 и лемм 3, 4. □

8. Классы S_i

В работе [5] для классов группы S_i получены асимптотические оценки сложности расшифровки запросами на значение, причем нижние оценки — мощностные. Поэтому заменив в нижних оценках основание логарифма с 2, используемого для запросов на значение, на 3, применяемого для запросов на сравнение, несложно получить порядок сложности расшифровки этих классов запросами на сравнение.

Теорема 11. *Справедливы соотношения*

- $\varphi_{CQ}(S_1, n, k) \asymp k \log(n/k), n, k \to \infty, k = o(n)$;

- $\varphi_{CQ}(S_3, n, k) \asymp k \log(n/k), n, k \to \infty, k = o(n)$;

- $\varphi_{CQ}(S_5, n, k) \asymp k \log(n/k), n, k \to \infty, k = o(n)$.

Теорема 12. *Справедливы соотношения*

- $\varphi_{CQ}(S_1, n, k) \asymp \log(n), n \to \infty$;

- $\varphi_{CQ}(S_3, n, k) \asymp \log(n), n \to \infty$;

- $\varphi_{CQ}(S_5, n, k) \asymp \log(n), n \to \infty$.

9. Классы L_i

Для классов линейных функций в случае запросов на значение в [5] была получена асимптотическая оценка сложности расшифровки и причем нижние оценки — мощностные. Поэтому аналогично предыдущему разделу, легко получить порядок сложности расшифровки классов L_i запросами на сравнение.

Теорема 13. *Справедливы соотношения*

- $\varphi_{CQ}(L_1^*, n, k) \asymp k \log n, n, k \to \infty, \log k = o(\log n)$;

- $\varphi_{CQ}(L_2, n, k) \asymp k \log n, n, k \to \infty, \log k = o(\log n)$;

- $\varphi_{CQ}(L_4, n, k) \asymp k \log n, n, k \to \infty, \log k = o(\log n)$;

- $\varphi_{CQ}(L_5, n, k) \asymp k \log n, n, k \to \infty, \log k = o(\log n)$.

Теорема 14. *Справедливы соотношения*

- $\varphi_{CQ}(L_1^*, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{CQ}(L_2, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{CQ}(L_4, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{CQ}(L_5, n, k) \asymp \log n, n \to \infty$.

10. Классы O_i

Лемма 13. *Для $n > 1$ справедливы следующее соотношение:*

1) *соотношение* $\left]\log_3 n + \log_3 2\right[\; = \; \left]\log_3 n\right[$ *справедливо в одном из двух случаев:*

   - $\log_3 n$ — *нецелое и* $(\log_3 n + \log_3 2)$ — *целое,*
   - *оба* $\log_3 n$ *и* $(\log_3 n + \log_3 2)$ — *нецелые и верно равенство*

   $$\Big[\log_3 n\Big] = \Big[\log_3 n + \log_3 2\Big],$$

2) *соотношение* $\left]\log_3 n + \log_3 2\right[= \left]\log_3 n\right[+ 1$ *справедливо в одном из двух случаев:*

   - $\log_3 n$ — *целое и* $(\log_3 n + \log_3 2)$ — *нецелое,*
   - *оба* $\log_3 n$ *и* $(\log_3 n + \log_3 2)$ — *нецелые, и выполнено равенство*

   $$\Big[\log_3 n\Big] = \Big[\log_3 n + \log_3 2\Big] - 1.$$

Доказательство. Случай, когда $\log_3 n$ и $(\log_3 n + \log_3 2)$ — целые, невозможен, в силу того, что $\log_3 2$ нецелое, а значит сумма $\log_3 n + \log_3 2$ тоже нецелая при целом $\log_3 n$.

Докажем первый пункт леммы.

Случай, когда оба $\log_3 n$ и $(\log_3 n + \log_3 2)$ — нецелые, и выполнено равенство
$$\Big[\log_3 n\Big] = \Big[\log_3 n + \log_3 2\Big].$$
Равенство первого пункта леммы верно.

Случай, когда $\log_3 n$ — нецелое и $(\log_3 n + \log_3 2)$ — целое, причем в этом случае очевидно, что выполнено равенство

$$\Big[\log_3 n\Big] = \Big[\log_3 n + \log_3 2\Big] - 1 = \log_3 n + \log_3 2 - 1.$$

Равенство первого пункта леммы верно.

Докажем второй пункт леммы.

Случай, когда $\log_3 n$ — целое и $(\log_3 n + \log_3 2)$ — нецелое, причем очевидно выполнение равенства $\log_3 n = \left[\log_3 n\right] = \left[\log_3 n + \log_3 2\right]$. Равенство второго пункта леммы верно.

Случай, когда оба $\log_3 n$ и $(\log_3 n + \log_3 2)$ — нецелые, и выполнено равенство

$$\left[\log_3 n\right] = \left[\log_3 n + \log_3 2\right] - 1.$$

Равенство второго пункта леммы верно.

$\square$

Теорема 15. *При $n > 1$ справедливы соотношения*

- $\varphi_{CQ}(O_1, n, 1) = \left]\log_3 n\right[$*;*
- $\varphi_{CQ}(O_4, n, 1) \sim \log_3 n, n \to \infty$*;*
- $\varphi_{CQ}(O_5, n, 1) \sim \log_3 n, n \to \infty$*;*
- $\varphi_{CQ}(O_8, n, 1) \sim \log_3 n, n \to \infty$*;*
- $\varphi_{CQ}(O_9^*, n, 1) \sim \log_3 n, n \to \infty$.

Доказательство. Точная оценка $\left]\log_3 n\right[$ для класса O_1 и верхняя оценка $\left]\log_3 n\right[+1$ для класса O_4 следует из теоремы 3 в [6].

Нижняя мощностная оценка для класса O_4 следующая:

$$\varphi_{CQ}(O_4, 1, n) \geq \left]\log_3(2n)\right[= \left]\log_3 n + \log_3 2\right[.$$

Согласно лемме 13, при определенных значениях n оценка $\left]\log_3 n+\log_3 2\right[$ равна значению $\left]\log_3 n\right[+1$, то есть совпадает с верхней оценкой для класса O_4, но при каких-то значениях нижняя оценка строго на 1 меньше верхней оценки. Несмотря на такое отличие в некоторых случаях верхней и нижней оценок, второй пункт теоремы доказан.

Для получения верхней оценки $\varphi_{CQ}(O_5, n, 1)$ или $\varphi_{CQ}(O_8, n, 1)$ предлагается следующий алгоритм расшифровки: узнаем значение на запросе $(0\ldots0, 1\ldots1)$, если ответ равен 0, то загадана единственная в классе константа, иначе загадана функция $x_i (1 \leq i \leq n)$. В случае функции x_i применяем алгоритм расшифровки, описанный в доказательстве утверждения 4 в [6], для нахождения за $\left]\log_3 n\right[$ запросов на сравнение номера существенной переменной. Поэтому сложность расшифровки классов O_5, O_8 не превосходит $\left]\log_3 n\right[+1$. Нижняя оценка для обоих классов мощностная и не меньше $\left]\log_3(n+1)\right[$, что совпадает с $\left]\log_3 n\right[+1$ при целом $\log_3 n$ и на 1 меньше $\left]\log_3 n\right[+1$ в остальных случаях. Третий и четвертый пункты теоремы доказаны.

Нижняя мощностная оценка для класса O_9^* следующая:

$$\varphi_{CQ}(O_9^*, 1, n) \geq \left]\log_3(2n+1)\right[\geq \left]\log_3 n + \log_3 2\right[.$$

Для получения верхней оценки $\varphi_{CQ}(O_9^*, n, 1)$ предлагается следующий алгоритм расшифровки: узнаем значение на запросе $(0\dots 0, 1\dots 1)$. Если ответ равен 0, то загадана единственная в классе константа 1. Если ответ равен 1, то загадан селектор $x_i (1 \leq i \leq n)$. Если ответ равен -1, то загадано отрицание селектора $\overline{x_i} (1 \leq i \leq n)$. Аналогично предыдущим рассматриваемым пунктам получаем верхнюю оценку

$$\varphi_{CQ}(O_9^*, 1, n) \leq 1 + \left]\log_3 n\right[.$$

$\square$

11. Теорема о сложности расшифровки для всех классов Поста

Теорема 1 получается объединением результатов теорем 3, 5, 7, 9, 11, 13, 15. Теорема 2 является объединением результатов теорем 4, 6, 8, 10, 12, 14, 15.

12. Благодарности

Автор приносит благодарность научному руководителю профессору Э. Э. Гасанову за постановку задачи и помощь в работе.

Исследование выполнено при поддержке Междисциплинарной научно-образовательной школы Московского университета “Мозг, когнитивные системы, искусственный интеллект”.

Список литературы

[1] Гасанов Э. Э., “Расшифровка линейных функций ранжирования”, *Материалы XI Международного семинара «Дискретная математика и ее приложения» (Москва, 18-23 июня 2012 г.)*, 2012, 332–334.

[2] R. Uehara, K. Tsuchida, I. Wegener, “Optimal Attribute-Efficient Learning Of Disjunction, Parity, And Threshold Functions”, *EuroCOLT '97 Proceedings of the Third European Conference on Computational Learning Theory*, 1997.

[3] T. Hofmeister, "An Application of Codes to Attribute-Efficient Learning", *EuroCOLT'99 Proceedings of the 4th European Conference on Computational Learning Theory*, 1999.

[4] В. В. Осокин, "О расшифровке монотонных булевых функций с несущественными переменными", *Дискрет. матем.*, **22**:3 (2010), 134–145.

[5] А. В. Быстрыгова, "Параметро-эффективная расшифровка булевых функций из замкнутых классов Поста", *Дискрет. матем.*, **31**:2 (2019), 34–58.

[6] А. В. Быстрыгова, "Запросы на сравнение в задаче параметро-эффективной расшифровки булевых функций", *Интеллектуальные системы. Теория и приложения*, **23**:4 (2019), 115–124.

[7] K. Sarkar, C.J.Colbourn, A.D. Bonis, U. Vaccaro, "Partial Covering Arrays: Algorithms and Asymptotics", International Workshop on Combinatorial Algorithms IWOCA 2016: Combinatorial Algorithms, 437–448.

[8] A. Godbole, D. Skipper, R. Sunley, "t-Covering Arrays: Upper Bounds and Poisson Approximations", *Combinatorics, Probability Computing*, **5** (1996), 105–117.

Using comparation queries in exact learning of Post closed classes

Bistrigova A.V.

We consider exact attribute-efficient learning of functions from Post closed classes using comparation queries and obtain bounds on learning complexity.

Keywords: exact learning, attribute-efficient learning, Post lattice of closed classes, comparation queries.

References

[1] E. E. Gasanov, "Learning of linear ranking functions", *Proceedings of the XI International Seminar ≪Discrete Mathematics and its Applications≫ (Moscow, June 18-23)*, 2012, 332–334.

[2] R. Uehara, K. Tsuchida, I. Wegener, "Optimal Attribute-Efficient Learning Of Disjunction, Parity, And Threshold Functions", *EuroCOLT '97 Proceedings of the Third European Conference on Computational Learning Theory*, 1997.

[3] T. Hofmeister, “An Application of Codes to Attribute-Efficient Learning”, *EuroCOLT’99 Proceedings of the 4th European Conference on Computational Learning Theory*, 1999.

[4] V.V. Osokin, “On learning monotone Boolean functions with irrelevant variables”, *Discrete Mathematics and Applications*, **20**:3 (2010), 307–320.

[5] A. V. Bystrygova, “Attribute-efficient learning of Boolean functions from Post closed classes”, *Discrete Mathematics and Applications*, **30**:5 (2020), 285–301.

[6] A. V. Bistrigova, “Using comparation queries in attribute-efficient learning of Boolean functions”, *Intelligent Systems. Theory and Applications*, **23**:4 (2019), 115–124.

[7] K. Sarkar, C.J.Colbourn, A.D. Bonis, U. Vaccaro, “Partial Covering Arrays: Algorithms and Asymptotics”, International Workshop on Combinatorial Algorithms IWOCA 2016: Combinatorial Algorithms, 437–448.

[8] A. Godbole, D. Skipper, R. Sunley, “t-Covering Arrays: Upper Bounds and Poisson Approximations”, *Combinatorics, Probability Computing*, **5** (1996), 105–117.