Московский Государственный Университет имени М. В. Ломоносова

На правах рукописи

Быстрыгова Анастасия Викторовна

Параметро-эффективная расшифровка булевых функций

Специальность 1.1.5 —
«Математическая логика, алгебра, теория чисел и дискретная математика»

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
д.ф.-м.н., профессор
Гасанов Эльяр Эльдарович

Москва — 2022

Оглавление

Стр.

Стр.

Введение

Практически полвека назад активно стало развиваться направление математики под названием теория расшифровки (computational learning theory), основополагающими работами в которой стали работы Коробкова [11], Валианта [40], Англуин [19], Дамашке [22], Литлстоуна [32]. Эта область остается актуальной и по сей день, поскольку связана с восстановлением оптимальным образом информации об исследуемом объекте на основе частичных сведений о нем. Более формально, под расшифровкой функции из заданного класса F понимают игру между *учеником* и *учителем*, в которой учитель загадывает одну функцию из класса F, а ученик, зная этот класс F полностью, но не зная выбор учителя, задает учителю запросы разрешенного типа, получает ответы от учителя и на основе этих ответов восстанавливает какую-то информацию про загаданную функцию.

При этом, существует несколько моделей расшифровки, и выбор модели определяет то, как происходит процесс расшифровки и что является его результатом. Наиболее популярны две модели:

- модель точной расшифровки (exact learning)

  В этой модели ученик сам выбирает запросы и его цель — полностью восстановить вектор значений загаданной функции.
- модель вероятно примерно точной расшифровки (probably approximately correct learning, PAC)

  В этой модели не ученик выбирает запрос, запрос выбирается в соответствии с заданным на множестве всех запросов вероятностным распределением. Цель — восстановить вектор значений загаданной функции так, чтобы вероятность ошибки была небольшой.

Чтобы оценить насколько быстро можно расшифровать функции из того или иного класса, вводят понятие *сложности расшифровки* как максимальное число запросов, которое надо задать учителю для расшифровки самой “плохой” функции. Иными словами, ученик выбирает “лучшую” стратегию восстановления функции, затем проверяется, а сколько запросов потребуется задать ученику, использующему эту стратегию, чтобы восстановить каждую функцию из класса, за сложность расшифровки принимают максимум среди этих значений.

Довольно часто возникающие на практике функции зависят от большого числа переменных, но лишь небольшая их часть существенно влияет на поведение функции. Поэтому становится актуальной *параметро-эффективная расшифровка* данных классов, в которой при разработке алгоритмов восстановления векторов столбцов значений функций весомым образом используется факт, что существенных переменных очень мало.

Помимо того, что при расшифровке важен выбор модели расшифровки, учитывание факта, что число существенных параметров сильно мало по сравнению с общим числом параметров, также значителен и выбор типа(ов) используемых при расшифровке запросов. Чаще встречаются:

- запросы на значение, или membership query (Коробков [11], Золотых[5], Хофмейстер[29], Дамашке[22], Осокин[13; 14], Уэхара и др. [39]): ученик выбирает набор, учитель говорит значение функции на выбранном наборе;
- запросы на сравнение (Гасанов[4], Хегай[16]): ученик выбирает пару наборов, а учитель возвращает в ответ знак разности значений функций на этих наборах;
- запросы на ограниченную эквивалентность, или equivalence query (Англуин[19; 20], Бертони[21], Бшоути и др.[33]): ученик выбирает функцию h из рассматриваемого класса, учитель отвечает “ДА”, если $f \equiv h$, иначе выдает набор b, на котором значения функций h, f отличаются;
- запросы на расширенную эквивалентность, или extended equivalence query (Англуин[20]): ученик выбирает булеву функцию h, которая необязательно принадлежит рассматриваемому классу, учитель отвечает “ДА”, если $f \equiv h$, иначе выдает набор b, на котором значения функций h, f отличаются;
- superset (subset) query (Англуин[19], Бшоути и др.[33]): ученик выбирает функцию h, учитель отвечает “ДА”, если $f \Rightarrow h$ ($h \Rightarrow f$), иначе выдает набор b такой, что $h(b) \neq f(b) = 1$ ($h(b) \neq f(b) = 0$).

В теории расшифровки функций привычной практикой стало рассмотрение задачи расшифровки одного и того же класса функций разными типами запросов или даже разными комбинациями нескольких типов запросов (Англуин[19; 20], Вороненко и Чистиков[3], Бертони и др.[21]). В 2004 году Англуин [20] ввела обозначения для запросов на значение MQ, на ограниченную EQ и

расширенную эквивалентность XEQ, которые и используются в диссертационной работе.

Данная диссертационная работа посвящена некоторым вопросам параметро-эффективной точной расшифровки булевых функций разными типами запросов. В первой главе работы описываются классы функций и типы запросов, для которых исследуется вопрос сложности расшифровки. В следующих трех ее главах приводятся результаты, полученные в результате исследования.

Вторая глава диссертационной работы основана на работах [43; 44] и посвящена исследованию параметро-эффективной расшифровки класса функций фиксированного веса для каждого из следующих четырех типов запросов в отдельности: на значение, на сравнение, на расширенную эквивалентность, на ограниченную эквивалентность. Для этого класса уже проведено исследование [8] функции Шеннона мощности плоских схем, реализующих такие функции. Но с точки зрении вопросов сложности расшифровки данный класс ранее никем не исследовался, если не брать в расчет результат Англуин [19] для функций веса 1 для трех из упомянутых типов запросов.

Третья и четвертая главы диссертационной работы, результаты которых опубликованы в работах [42] и [41; 45] соответственно, посвящены вопросам сложности расшифровки замкнутых классов Поста. Эти классы стали предметом изучения в разных задачах с 1930–1950 годов после того как Пост описал (рис. 1) все замкнутые классы двузначной логики в своих работах [35; 36]. К настоящему времени уже известны результаты о сложности автоматной реализации этих классов [9], получены все спектры (множества длин базисов) [2], приведены оценки слоистости [17]. Помимо этого, исследован порядок функции Шеннона средней и максимальной мощности плоских схем для всех замкнутых классов [7] и рассмотрен вопрос о мощности генерирующих множеств по операциям из классов решетки Поста [10].

Отдельное направление исследований было связано с расшифровкой функций из этих классов запросами на значение. Одним из первых классов, расшифровкой функций из которого занялись исследователи, стал класс монотонных функций. Задачу расшифровки этого класса в разных версиях рассматривали в своих работах Дамашке [22; 23], Коробков [11], Ансель [1], Осокин [14] и Селезнева с Лю [15]. Ансель предложил алгоритм, благодаря которому он получил верхнюю оценку сложности расшифровки монотонных функций от n переменных, совпадающую с нижней оценкой, доказанной ранее Короб-

ковым, и равную

$$C_n^{[n/2]} + C_n^{[n/2]+1}.$$

Селезенева с Лю показали, что эта оценка сохраняется даже в случае возможного одного неверного ответа учителя.

Осокина интересовал вопрос сложности расшифровки монотонных функций от n переменных, где не более k переменных существенные. Им в 2010 году был получен порядок сложности расшифровки монотонных функций

$$\frac{2^k}{\sqrt{k}} + k \log n.$$

Дамашке помимо расшифровки монотонных функций рассматривал и расшифровку всех функций алгебры логики в целом (класс C_1 решетки замкнутых классов Поста на рис. 1). Его работы показали, что задача расшифровки функций из замкнутых классов Поста нередко сводится к задаче построения binary covering array для заданных чисел: арности n, верхней оценки на число существенных переменных k.

Binary covering array являются частным случаем ортогональных массивов. Сам термин "covering array" был введен в 1993 году Слоаном [37] и с тех пор прижился. В русскоязычной литературе этот объект освещен мало. В своей работе [12] 2011 года авторы предлагают называть его *покрывающим набором*, но кажется более естественным называть его *покрывающей матрицей*, поэтому это название и используется в тексте диссертационной работы.

Бинарная покрывающая матрица (binary covering array) — это бинарная матрица, у которой n столбцов, обладающая свойством, что если зафиксировать любые k столбцов, то в этих зафиксированных столбцах будут содержаться все 2^k двоичных наборов-строк.

Изучение этих матриц представляет интерес в силу их разных применений. Причем в приложениях можно не ограничиваться алфавитом $\{0, 1\}$, а брать более мощный алфавит. Наиболее известное применение покрывающих матриц — это софтверное и хардверное тестирование, предложенное [38] еще в 1997 году. Предположим, у программы n входных параметров, где каждый аргумент может принимать v значений. Тогда рассмотрим покрывающую матрицу с n столбцами, где каждый элемент принимает значение от 0 до $v - 1$ включительно и при фиксации любых k столбцов в строках встречаются все v^k комбинаций. Каждую строку такой матрицы можно рассматривать как тест.

Благодаря такой матрице, можно проверить корректность работы программы на разных комбинациях любых k параметров. Ясно, что чем меньше строк в матрице, тем меньше тестов придется сделать.

Другое необычное приложение оценок на число строк покрывающих матриц заметил Хартман [28], изучая задачу о слепых роботах на прямой (blind dyslectic synchronized robots on a line), которую ранее рассматривала в своей работе [31] Лим. Благодаря своему замечанию он сумел понизить имеющуюся тогда оценку [25] для этой задачи на $]\log_2 n[$.

Как выяснилось, и в расшифровке функций находят свое применение покрывающие матрицы. Дамашке показал [22], что булеву функцию, зависящую от n переменных, не более k из которых существенные, можно расшифровать за не более $\alpha(n,k) + k\log n$ запросов на значение, также им была приведена тривиальная нижняя оценка $\alpha(n,k)$, где $\alpha(n,k)$ — минимальное число строк в бинарной покрывающей матрице для чисел n,k.

Хотелось бы оценить число $\alpha(n,k)$, но, к сожалению, на данный момент неизвестна асимптотика, или даже порядок этой функции, хотя вопрос построения бинарных покрывающих матриц для n,k с наименьшим числом строк $\alpha(n,k)$ изучается давно (Клейтман[30], Гаргано[26], Слоун[37], Годбол[27], Лоуренс[18], Саркар[34], Дас[24]), но пока лишь получены некие неравенства и соотношения.

1. В 1993 году получена [26] асимптотическая оценка при $n \to \infty$
$$\alpha(n,2) = \log_2 n(1+o(1)).$$
2. В 1996 году Годболу, Скипперу и Санли [27] удалось получить следующую верхнюю оценку для случая $k \geqslant 2, n \to \infty$
$$\alpha(n,k) \leqslant (1+o(1)))\frac{k-1}{\log_2 \frac{2^k}{2^k-1}} \cdot \log_2 n.$$
В 2017 году было показано [24], что при $k \to \infty$ правая часть асимптотически равна $2^k(k-1)\ln 2 \cdot \log_2 n$.
3. $\alpha(n,k) \geqslant 2\alpha(n-1,k-1)$.
4. В 2016 году коллектив авторов [34] доказал, что для $k \geqslant 2$ при $n \to \infty$ верна нижняя оценка
$$\alpha(n,k) \geqslant 2^{k-2} \cdot \alpha(n-k+2,2) = 2^{k-2}\log_2(n-k+2)(1+o(1)).$$
5. $\alpha(n,1) = 2$ для любого натурального n.

Помимо классов монотонных функций и класса всех булевых функций внимание исследователей с точки зрения расшифровки привлекали также линейные функции, логические суммы и селекторы. В 1997 году существенно в вопросе изучения сложности расшифровки этих классов продвинулся коллектив авторов Уэхара, Цутида, Вегенер [39]. Они показали, что сложность точной расшифровки запросами на значение класса функций арности n, равные логической сумме k своих переменных (класс S_1 на рис. 1), не меньше $\rceil \log_2 C_n^k \lceil$ и не больше $k\rceil \log(n/k)\lceil + 2k - 2$. Более того, они разработали алгоритмы расшифровки линейных функций с 1, 2, 3 существенными переменными и нулевым свободным членом, требующие не более $\rceil \log_2 n[$, $3\rceil \log_2 n[-2$ и $4\rceil \log_2 n[-3$ запросов соответственно для точной расшифровки функции. Заметим, что линейные функции с ровно одной существенной переменной и нулевым свободным членом и есть селекторы, поэтому авторами была получена точная оценка сложности расшифровки класса O_1 решетки Поста (рис. 1).

Случай линейных функций с произвольным числом существенных переменных был рассмотрен авторами в модели PAC. Ими было показано, что для класса линейных функций арности n, у которых ровно k существенных переменных, существует рандомизированный алгоритм расшифровки с нулевой ошибкой, сложность которого равна $k \log \frac{n}{k} + O(k)$ запросов. Вопрос точной расшифровки этого класс смог закрыть Хофмейстер [29] в 1999 году. Для получения верхней оценки $k \log_2 n + k$ сложности точной расшифровки он применил теорию построения линейных кодов и получил асимптотику сложности расшифровки функций класса L_3 решетки Поста, у которых из n переменных не более k являются существенными.

Таким образом, по задаче расшифровки запросами на значение функций из классов решетки Поста было много работ, поэтому вполне обоснована необходимость собрать все имеющиеся результаты и привести оценки для ранее не освещавшихся классов (классов самодвойственных функций и классов “счетной этажерки”), чтобы понять, насколько сложна расшифровка функций из разных классов, если в качестве запросов используются запросы на значение. Этому посвящена третья глава диссертационной работы. А четвертая же освещает данную задачу для запросов на сравнение. Запросы на сравнение были введены в литературу Гасановым сравнительно недавно [4], поэтому неудивительно, что еще не исследованы ни классы решетки Поста, ни класс функций ограниченного веса с точки зрения сложности точной расшифровки их запросами

на сравнение. Четвертая глава диссертационной работы посвящена закрытию этого пробела.

Целью данной работы является изучение в рамках модели точной расшифровки сложности параметро-эффективной расшифровки замкнутых классов Поста и класса функций ограниченного веса. Для класса функций ограниченного веса необходимо получить оценки сложности расшифровки для четырех типов запросов в отдельности: на значение, на сравнение, на расширенную или ограниченную эквивалентность. Для всех замкнутых классов Поста необходимо получить эти оценки для двух типов запросов в отдельности: на значение и сравнение. При этом в случае класса функций ограниченного веса ученику известна арность функции и верхняя оценка на количество единиц в векторе значений функции, которых значительно меньше, чем длина вектора значений. В случае замкнутых классов Поста ученику известна арность функции и верхняя оценка на количество существенных переменных, причем существенных переменных сильно меньше общего числа переменных.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Получить точные значения сложности расшифровки класса функций ограниченного веса для трех типов запросов в отдельности: на значение, на расширенную и ограниченную эквивалентность.
2. Оценить порядок сложности расшифровки класса функций ограниченного веса запросами на сравнение.
3. Оценить характер сложности расшифровки замкнутых классов Поста запросами на значение.
4. Оценить характер сложности расшифровки замкнутых классов Поста запросами на сравнение.

Научная новизна:

1. Впервые получены точные значения сложности расшифровки класса функций ограниченного веса для трех типов запросов в отдельности: на значение, на расширенную и ограниченную эквивалентность.
2. Впервые получены точные значения сложности расшифровки запросами на сравнение класса функций малого веса: 1, 2, 3.
3. Впервые получена практически точная оценка сложности расшифровки запросами на сравнение классов функций веса: ограниченного снизу нулем и ограниченного снизу единицей.

4. Впервые получен порядок сложности расшифровки запросами на сравнение класса функций ограниченного веса в случае, когда растет арность функции, а ее вес не меняется.
5. Впервые получены оценки сложности расшифровки запросами на значение замкнутых классов самодвойственных функций и классов “счетной этажерки” решетки Поста.
6. Впервые получены оценки сложности расшифровки запросами на сравнение всех замкнутых классов решетки Поста.

Теоретическая и практическая значимость. Диссертационная работа в основном носит теоретический характер. Результаты работы могут быть использованы в дальнейшем теоретическом исследовании оценок сложности расшифровки других классов булевых функций. Тем не менее приведенные в работе утверждения могут быть также применены на практике в задачах восстановления информации об объекте из частичных сведений о нем, если известно, какими свойствами обладает этот объект.

Методология и методы исследования. В работе используются методы дискретного анализа, комбинаторики, теории графов, а также математического анализа.

Основные положения, выносимые на защиту. На защиту выносятся обоснование актуальности проведенного исследования и его научной новизны, цели и поставленные задачи, методы исследования, примененные для получения результатов, а также следующие положения, которые подтверждаются результатами исследований, представленными в Заключении диссертации.

1. Значения сложности расшифровки класса функций ограниченного веса для трех типов запросов в отдельности: на значение, на расширенную и ограниченную эквивалентность.
2. Значения сложности расшифровки запросами на сравнение класса функций малого веса: 1, 2, 3.
3. Оценки сложности расшифровки запросами на сравнение класса функций веса, ограниченного сверху произвольным числом, а снизу либо единицей, либо нулем.
4. Порядок сложности расшифровки запросами на сравнение класса функций ограниченного веса в случае, когда арность функции растет, но вес не меняется.

5. Оценки сложности расшифровки запросами на значение замкнутых классов самодвойственных функций и классов “счетной этажерки” решетки Поста.
6. Оценки сложности расшифровки запросами на сравнение всех замкнутых классов решетки Поста.

Достоверность полученных результатов обеспечивается строгими математическими доказательствами. Результаты работы прошли апробацию на всероссийских и международных научных конференциях, научных семинарах и опубликованы в рецензируемых научных журналах. Результаты других авторов, используемые в тексте данной диссертационной работы, приводятся с указанием выходных данных публикаций.

Апробация работы. Основные результаты работы докладывались на научном семинаре “Математические вопросы кибернетики” кафедр дискретной математики и математической теории интеллектуальных систем механико-математического факультета и кафедры математической кибернетики факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова (2022), а также следующих семинарах механико-математического факультета МГУ им. М. В. Ломоносова: “Теория автоматов” под руководством академика, проф., д.ф.-м.н. В. Б. Кудрявцева (2020), “Вопросы сложности алгоритмов поиска” под руководством проф., д.ф.-м.н. Э. Э. Гасанова (2016–2022), “Кибернетика и информатика” под руководством академика, проф., д.ф.-м.н. В. Б. Кудрявцева и к.ф-м.н, с.н.с. А. В. Галатенко (2018).

Помимо этого, результаты работы были представлены на международной научной конференции студентов и аспирантов “Ломоносов” (2020) и конференции “Ломоносовские чтения”, секция “Математика” (2016, 2018–2021), а также на X международной конференции “Дискретные модели в теории управляющих систем” (Красновидово, 2018), семинаре компании Huawei Moscow Research Center “Intelligent Systems Workshop” (г. Москва, 2020), XIX международной конференции “Проблемы теоретической кибернетики” (г. Казань, 2021).

Личный вклад. Все приводимые в работе результаты, за исключением специально выделенных, сформулированы и доказаны автором лично.

Публикации. Соискатель имеет 5 опубликованных работ [41—45], 5 из которых по теме диссертации, из них 2 опубликованы в периодических научных журналах, индексируемых Web of Science, Scopus и RSCI [42; 44], 3 опубликованы в рецензируемом научном издании из дополнительного списка,

утвержденного ученым советом МГУ, в котором могут быть опубликованы научные результаты диссертаций по направлению физико-математические науки [41; 43; 45]. Работ, написанных в соавторстве, нет.

Диссертационная работа была выполнена в рамках работы Междисциплинарной научно-образовательной школы Московского университета "Мозг, когнитивные системы, искусственный интеллект".

Краткое содержание работы.

Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, приводится обзор научной литературы по изучаемой проблеме, формулируется цель, ставятся задачи работы, излагается научная новизна и научная значимость представляемой работы.

Первая глава состоит из трех разделов, в которых вводятся обозначения и определения, используемые на протяжении всей работы. В первом ее разделе перечисляются значения как распространенных обозначений (например, символы $|, \&$) для того, чтобы избежать неоднозначного их трактования в результатах, так и вводимых в работе функций $G(k,m) = k \cdot [m/(k+1)] + (m \bmod (k+1))$ и $S_{n,k} = \max_{p\in\mathbb{N}, 1\leqslant p<k} (2^p - 1)\alpha(n-p, k-p)$, где $\alpha(n,k)$ — число строк в бинарных покрывающих матрицах, про которые говорилось выше.

Во втором разделе первой главы дается определение используемых в работе типов запросов: *запросы на значение (MQ), запросы на сравнение (CQ), запросы на расширенную эквивалентность (XEQ), запросы на ограниченную эквивалентность (EQ)*, а также приводится определение понятия *сложность расшифровки $\varphi_T(M,n)$ запросами типа T*, где $T \in \{MQ, CQ, XEQ, EQ\}$, а M — множество булевых функций арности n.

В последнем разделе первой главы приводятся описания классов функций, расшифровка которых является целью диссертационной работы. В этом разделе вводится обозначение $F(n,k,i)$ для класса функций ограниченного веса, то есть множества булевых функций арности n, вес которых лежит в диапазоне $[i,k]$, $k \in (0, 2^n], i \in [0, 2^n)$, $i \leqslant k$. Помимо этого, в разделе предоставляется словесное описание всех замкнутых классов Поста (рис. 1) и уточняется, что классы из "правой" половины решетки Поста заведомо опускаются из 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. Также в этом разделе для удобства

вводятся обозначение $\varphi_T(F, n, k, i)$ сложности расшифровки запросами типа T класса функций ограниченного веса и обозначение $\varphi_T(R, n, k)$ сложности расшифровки запросами типа T класса R, где R — один из замкнутых классов решетки Поста, а $R(n, k)$ — все функции из R, у которых арность n и не более k существенных переменных. Под $\varphi_T(F, n, k, i)$ понимается $\varphi_T(F(n, k, i), n)$, под $\varphi_T(R, n, k)$ — $\varphi_T(R(n, k), n)$.

В следующих трех главах излагаются результаты решения задач, поставленных в рамках диссертационной работы.

Во **второй главе** рассматривается точная расшифровка класса функций ограниченного веса для четырех типов запросов: на значение, на расширенную и ограниченную эквивалентность, а также на сравнение.

В первых трех разделах приводится доказательство теорем с точными значениями сложности расшифровки упомянутого класса для первых трех типов запросов.

Теорема 1. *Сложность расшифровки класса $F(n, k, i)$ запросами на значение равна*

$$\varphi_{MQ}(F, n, k, i) = \begin{cases} 2^n - 1 & при\ i = k, \\ 2^n & при\ i < k. \end{cases}$$

Теорема 2. *Сложность расшифровки класса $F(n, k, i)$ запросами на расширенную эквивалентность равна*

$$\varphi_{XEQ}(F, n, k, i) = \min(k, 2^n - i).$$

Теорема 3. *Сложность расшифровки класса $F(n, k, i)$ запросами на ограниченную эквивалентность равна*

$$\varphi_{EQ}(F, n, k, i) = \begin{cases} k & при\ i = 0, \\ 2^n - 1 & при\ 0 < i = k, \\ 2^n & при\ 0 < i < k < 2^n, \\ 2^n - i & при\ 0 < i < k = 2^n. \end{cases}$$

Из этих оценок видно, что для класса функций ограниченного веса лучшими с точки зрения наименьшей сложности расшифровки являются запросы на расширенную эквивалентность. Между тем использовать запросы на значение и запросы на ограниченную эквивалентность для этого класса нецелесообразно

в силу того, что сложность расшифровки этими типами запросов почти всегда схожа с восстановлением всего вектора значений функций.

Вторую главу завершает раздел с оценками сложности расшифровки класса функций ограниченного веса запросами на сравнение.

В этом разделе вводятся понятия *класс можно (нельзя) расшифровать запросами на сравнение* и доказывается критерий того, когда класс булевых функций расшифровать можно, то есть в каком случае существует алгоритм расшифровки запросами на сравнение, который сможет восстановить функцию, загаданную учителем, независимо от того, какая функция им выбрана.

Теорема 4. *Класс булевых функций расшифровать запросами на сравнение нельзя тогда и только тогда, когда ему принадлежат обе константные функции* $0, 1$.

Далее в разделе приводится следующая верхняя оценка для функций фиксированного веса.

Теорема 5. *Пусть* $k \leqslant 2^{n-1}$ *и для целых положительных* $x_m, x_{m+1}, \ldots, x_{k-1}, x_k$, *где* $m = \left](k+1)/2\right[$, *верно равенство*

$$2^n = m \cdot x_m + (m+1) \cdot x_{m+1} + \ldots + (k-1) \cdot x_{k-1} + k \cdot x_k.$$

Тогда справедлива следующая верхняя оценка

$$\varphi_{CQ}(F, n, k, k) \leqslant 2^n - (x_m + x_{m+1} + \ldots + x_{k-1} + x_k) + [\max(x_m, x_{m+1}, \ldots, x_{k-1}, x_k)/2].$$

Следующее следствие получается после подстановки определенных значений $x_m, x_{m+1}, \ldots, x_k$ в последнюю теорему.

Следствие 1. *Пусть* $3 \leqslant k \leqslant 2^{n-1}, m =](k+1)/2[, s = m + (m+1) + \ldots + (k-1) + k$, *верно равенство* $2^n = s \cdot q + r, r \in [0, s), q \geqslant m$, q, r — *целые положительные числа. Тогда справедлива следующая верхняя оценка*

$$\varphi_{CQ}(F, n, k, k) \leqslant 2^n - (k-m+1)q - c + \left[0.5 \cdot \max(q - r + (m+1)c, q + r - mc, q)\right] \leqslant 2^n - k/2 \cdot [2^n/s] + \left[0.5 \cdot \left([2^n/s] + \right](k+1)/2\left[+ k^2\right)\right],$$

где c *вычисляется следующим образом*

– $c = [2r/(2m+1)]$ *при* $2r \bmod (2m+1) \leqslant m$,
– $c = [2r/(2m+1)] + 1$ *при* $2r \bmod (2m+1) > m$.

Для того, чтобы понять насколько лучше полученная в этом следствии оценка по сравнению с тривиальной — $(2^n - 1)$ запросов, приводится следующее следствие.

Следствие 2. *При* $2^n > k$ *справедлива следующая верхняя оценка*

$$\varphi_{CQ}(F, n, k, k) \leqslant 2^n\big(1 - \frac{2k}{3(k+1)^2} + \frac{4}{3k^2}\big) + \frac{k^2}{2} + \frac{3k}{4} + 1.$$

Причем, при $k \geqslant 5$ *эта величина строго меньше* $2^n - 1$.

Далее доказываются точные оценки сложности класса функций малого веса: веса 1, 2, 3. Причем, для функций веса 1 точная оценка получается довольно просто, но для функций веса 2 и 3 уже требуется более сложный разбор возможных случаев.

Теорема 6. *Сложность расшифровки класса* $F(n, 1, 1)$ *запросами на сравнение равна* $\varphi_{CQ}(F, n, 1, 1) = 2^{n-1}$.

Теорема 7. *При* $n \geqslant 2$ *сложность расшифровки класса* $F(n, 2, 2)$ *запросами на сравнение равна* $\varphi_{CQ}(F, n, 2, 2) = [2^{n+1}/3]$.

Теорема 8. *При* $n \geqslant 6$ *сложность расшифровки класса* $F(n, 3, 3)$ *запросами на сравнение равна*

$$\varphi_{CQ}(F, n, 3, 3) = 2^n -]3/2 \cdot [2^n/5][-[(2^n \bmod 5)/2].$$

После этого приводится оценки для самых больших представителей класса функций ограниченного веса: веса неограниченного снизу и ограниченного снизу единицей.

Теорема 9. *При* $n \geqslant 2, 2^{n-1} \geqslant k$ *сложность расшифровки класса* $F(n, k, 0)$ *запросами на сравнение равна*

$$\varphi_{CQ}(F, n, k, 0) = G(k, 2^n).$$

Теорема 10. *При* $n \geqslant 2, 2^{n-1} \geqslant k \geqslant 1, 2^n \bmod (k+1) = k$ *сложность расшифровки класса* $F(n, k, 1)$ *запросами на сравнение равна*

$$\varphi_{CQ}(F, n, k, 1) = G(k, 2^n) - 1.$$

При $n \geqslant 2, 2^{n-1} \geqslant k \geqslant 1, 2^n \bmod (k+1) = 0$ *сложность расшифровки класса* $F(n,k,1)$ *запросами на сравнение равна*

$$\varphi_{CQ}(F,n,k,1) = G(k,2^n).$$

При $n \geqslant 2, 2^{n-1} \geqslant k \geqslant 1, 2^n \bmod (k+1) \in (0,k)$ *сложность расшифровки класса* $F(n,k,1)$ *запросами на сравнение удовлетворяет следующим ограничениям:*

$$G(k,2^n) - 1 \leqslant \varphi_{CQ}(F,n,k,1) \leqslant G(k,2^n).$$

Раздел, а вместе с ним и вторая глава, завершается теоремой про порядок сложности расшифровки функций ограниченного веса запросами на сравнение.

Теорема 11. *Для любого* $k = k(n)$*, такого, что* $k \geqslant 2, k = o(2^n)$*, сложность расшифровки класса* $F(n,k,i)$ *запросами на сравнение при* $n \to \infty$ *удовлетворяет следующим соотношениям:*

$$\begin{cases} 7/10 \cdot 2^n \lesssim \varphi_{CQ}(F,n,k,i) \lesssim k/(k+1) \cdot 2^n & \text{при } i \leqslant 3 \leqslant k, \\ 2/3 \cdot 2^n \lesssim \varphi_{CQ}(F,n,k,i) \lesssim k/(k+1) \cdot 2^n & \text{при } i > 3 \text{ или } k = 2. \end{cases}$$

В **третьей главе** данной работы рассматривается параметро-эффективная расшифровка запросами на значение всех замкнутых классов Поста. Глава начинается с раздела со вспомогательными определениями и утверждениями. Далее следуют семь разделов с описанием результатов для групп классов, объединенных по букве в их обозначении: $C_i, A_i, D_i, F_j^i, S_i, L_i, O_i$. Главу завершает раздел с двумя теоремами, описывающие результаты рассматриваемой задачи для всех замкнутых классов решетки Поста для двух случаев:

1. и арность n, и верхняя оценка на число существенных переменных k стремятся к бесконечности,
2. только n стремится к бесконечности, а k зафиксирован.

Под условной асимптотической оценкой в формулировке приводимой теоремы будем понимать то, что оценка асимптотически равна величине, связанной с $\alpha(n,k)$, асимптотика которой неизвестна.

Теорема 12. *Замкнутые классы решетки Поста по характеру известной на данный момент сложности точной расшифровки запросами на значение разделены на четыре группы в случае* $n, k \to \infty$*:*

1. точная оценка

- $\varphi_{MQ}(O_4, n, 1) = \rceil \log_2 n \lfloor$ *при* $n > 1$;

2. *асимптотика*
   - $\varphi_{MQ}(S_1, n, k) \sim k \log_2 (n/k), n, k \to \infty, k = o(n)$;
   - $\varphi_{MQ}(S_3, n, k) \sim k \log_2 (n/k), n, k \to \infty, k = o(n)$;
   - $\varphi_{MQ}(S_6, n, k) \sim k \log_2 (n/k), n, k \to \infty, k = o(n)$;
   - $\varphi_{MQ}(L_1, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
   - $\varphi_{MQ}(L_2, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
   - $\varphi_{MQ}(L_3, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
   - $\varphi_{MQ}(L_4, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
   - $\varphi_{MQ}(L_5, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
   - $\varphi_{MQ}(O_5, n, 1) \sim \log_2 n, n \to \infty$;
   - $\varphi_{MQ}(O_6, n, 1) \sim \log_2 n, n \to \infty$;
   - $\varphi_{MQ}(O_8, n, 1) \sim \log_2 n, n \to \infty$;
   - $\varphi_{MQ}(O_9, n, 1) \sim \log_2 n, n \to \infty$;

3. *условная асимптотика*
   - $\varphi_{MQ}(C_1, n, k) = \alpha(n, k) \cdot (1 + o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - $\varphi_{MQ}(C_2, n, k) = \alpha(n, k) \cdot (1 + o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - $\varphi_{MQ}(C_4, n, k) = 2\alpha(n-1, k-1) \cdot (1 + o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - $\varphi_{MQ}(D_3, n, k) = \alpha(n-1, k-1) \cdot (1+o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - $\varphi_{MQ}(D_1, n, k) = \alpha(n-1, k-1) \cdot (1+o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_4^i, n, k) = \alpha(n, k) \cdot (1 + o(1))$ *при* $k, n \to \infty, k = o(n)$, $k \geqslant 2$;
   - *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_1^i, n, k) = S_{n,k} \cdot (1 + o(1))$ *при* $k, n \to \infty, k = o(n)$;

4. *порядок*
   - $\varphi_{MQ}(A_1, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $k, n \to \infty$;
   - $\varphi_{MQ}(A_2, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $k, n \to \infty$;
   - $\varphi_{MQ}(A_4, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $k, n \to \infty$;
   - $\varphi_{MQ}(D_2, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $k, n \to \infty$;
   - *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_2^i, n, k) \asymp k \log n + \frac{2^k}{\sqrt{k}}$ *при* $k, n \to \infty$;
   - *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_3^i, n, k) \asymp k \log n + \frac{2^k}{\sqrt{k}}$ *при* $k, n \to \infty$.

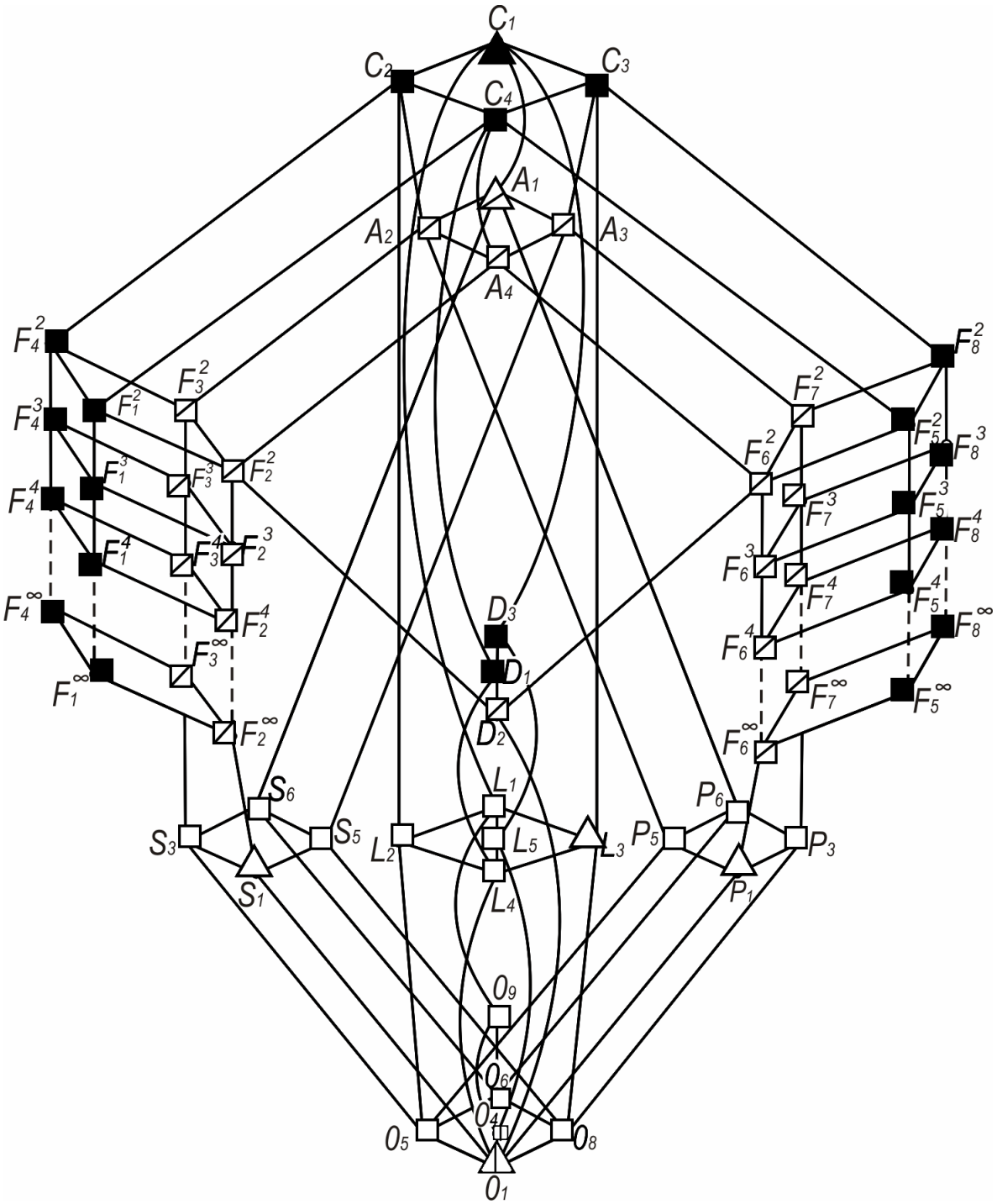

Рисунок 1 — Результаты сложности расшифровки замкнутых классов Поста запросами на значение при $n, k \to \infty$.

На рисунке 1 схематично приведены результаты этой теоремы. Если класс выделен вертикальной чертой в квадрате или треугольнике, то имеется точная оценка. Если класс выделен белым, то для него получена асимптотическая оценка, если обозначен косой линией внутри квадрата или треугольника, то оценка по порядку, если черным, то условная асимптотическая оценка. Если класс выделен треугольником, то это результат, полученный ранее в других работах, все такие результаты в третьей главе будут называться утверждениями с указанием источника и автора. Если класс выделен квадратом, то результат ранее в литературе не встречался и впервые получен автором диссертационной работы, такие результаты в работе будут называться теоремами.

Теорема 13. *Замкнутые классы решетки Поста по характеру известной на данный момент сложности точной расшифровки запросами на значение в случае, когда n растет, а k не меняется, разделены на три группы:*

1. *точная оценка*
   - $\varphi_{MQ}(O_4, n, 1) = \rceil \log_2 n \lfloor$ *при* $n > 1$;
2. *асимптотика*
   - $\varphi_{MQ}(O_5, n, 1) \sim \log_2 n, n \to \infty$;
   - $\varphi_{MQ}(O_6, n, 1) \sim \log_2 n, n \to \infty$;
   - $\varphi_{MQ}(O_8, n, 1) \sim \log_2 n, n \to \infty$;
   - $\varphi_{MQ}(O_9, n, 1) \sim \log_2 n, n \to \infty$;
3. *порядок*
   - $\varphi_{MQ}(C_1, n, k) \asymp \log n, n \to \infty, k \geqslant 2$;
   - $\varphi_{MQ}(C_2, n, k) \asymp \log n, n \to \infty, k \geqslant 2$;
   - $\varphi_{MQ}(C_4, n, k) \asymp \log n, n \to \infty, k \geqslant 2$;
   - $\varphi_{MQ}(A_1, n, k) \asymp \log n$ *при* $n \to \infty$;
   - $\varphi_{MQ}(A_2, n, k) \asymp \log n$ *при* $n \to \infty$;
   - $\varphi_{MQ}(A_4, n, k) \asymp \log n$ *при* $n \to \infty$;
   - $\varphi_{MQ}(D_1, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
   - $\varphi_{MQ}(D_2, n, k) \asymp \log n$ *при* $n \to \infty$;
   - $\varphi_{MQ}(D_3, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
   - *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_1^i, n, k) \asymp \log n$ *при* $n \to \infty$, $k \geqslant 2$;
   - *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_2^i, n, k) \asymp \log n$ *при* $n \to \infty$;
   - *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_3^i, n, k) \asymp \log n$ *при* $n \to \infty$;

- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_4^i, n, k) \asymp \log n$ *при* $n \to \infty$, $k \geqslant 2$;
- $\varphi_{MQ}(S_1, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(S_3, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(S_6, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_1, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_2, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_3, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_4, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_5, n, k) \asymp \log n, n \to \infty$.

Содержательную часть работы завершает **четвертая глава**, в которой рассматривается параметро-эффективная расшифровка запросами на сравнение всех замкнутых классов Поста. Глава начинается с раздела во вспомогательными определениями и утверждениями. В этом разделе приводится следствие, которое разделяет классы решетки Поста на две группы: которые можно и которые нельзя расшифровать запросами на сравнение.

Следствие 3. *Если* $F \in \{C_1, A_1, L_1, S_6, O_9, O_6\}$, *то расшифровать запросами на сравнение класс* F *невозможно. Если* $F \in \{C_2, C_3, C_4,\ A_2, A_3, A_4,\ F_1^i, F_2^i, F_3^i, F_4^i,\ D_1, D_2, D_3,\ S_1, S_3, S_5,\ L_2, L_3, L_4, L_5,\ O_1, O_4, O_5, O_8\}$, *то класс* F *можно расшифровать.*

Но чтобы не терять полностью из рассмотрения замкнутые классы $C_1, A_1, L_1, S_6, O_9, O_6$ лишь по тому, что они содержат обе константы, предлагается все же их рассматривать, но без константы 0, и обозначать символом *. Причем обращается внимание, что замыкание класса C_1^* совпадает с C_1, L_1^* — с L_1, O_9^* — с O_9, а $A_1^* = A_2, S_6^* = S_3, O_6^* = O_5$, поэтому для классов C_1^*, L_1^*, O_9^* будет приводится отдельно оценка сложности расшифровки, а для остальных — нет.

Далее следуют семь разделов с описанием результатов для групп классов, объединенных по букве в их обозначении: $C_i, A_i, D_i, F_j^i, S_i, L_i, O_i$. Глава завершается разделом с двумя теоремами, в которые объединены результаты всех предыдущих разделов главы для двух случаев: оба параметра n, k стремятся к бесконечности, только n стремится к бесконечности, а k зафиксирован.

Под условной оценкой по порядку, встречающейся в формулировке теоремы, будем понимать то, что оценка по порядку равна величине, связанной с $\alpha(n, k)$, порядок и асимптотика которой неизвестны.

Теорема 14. *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*
   - $\varphi_{CQ}(O_1, n, 1) = \left]\, \log_3 n \,\right[$;
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* $n, k \to \infty, \log k = o(\log n)$.

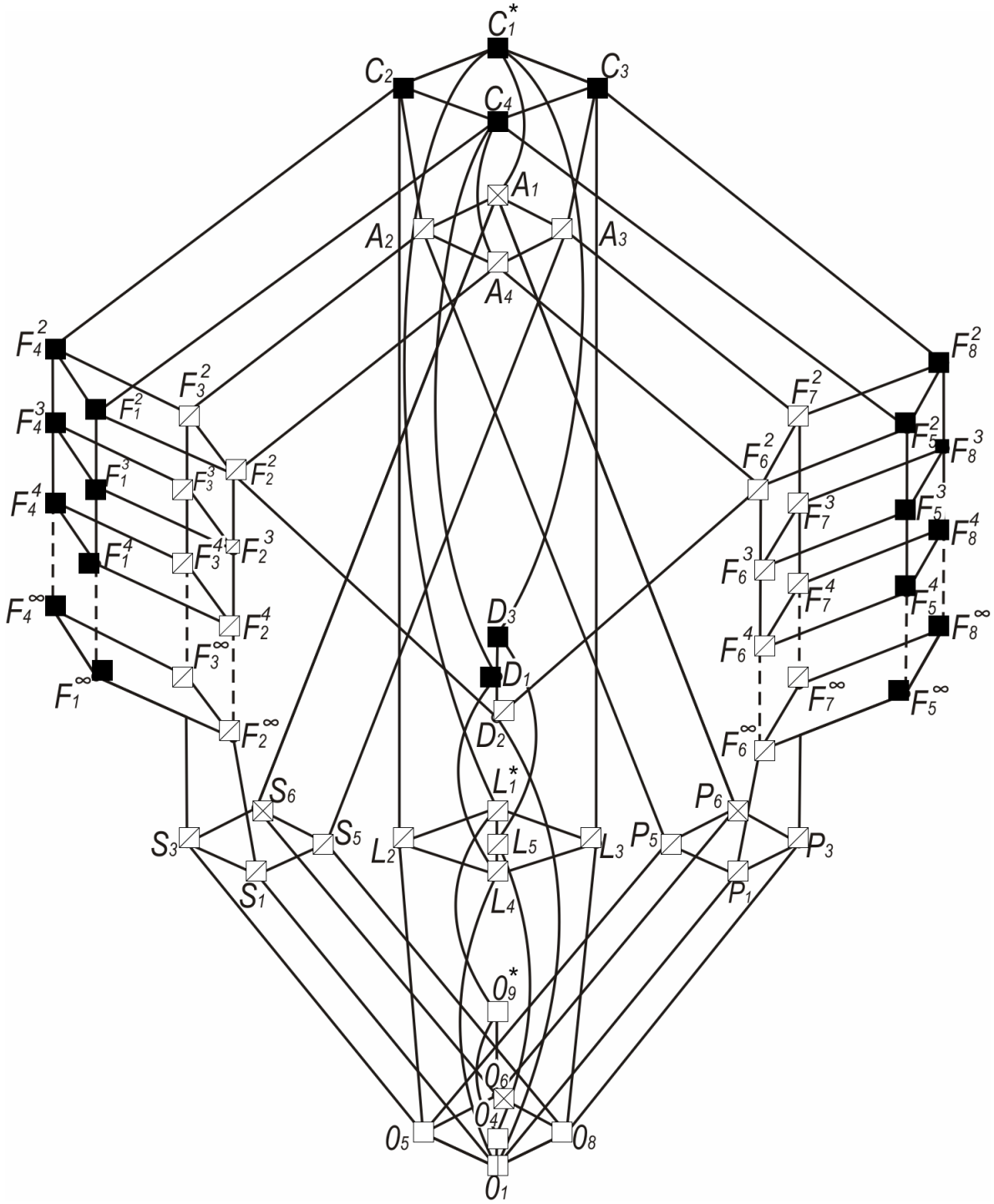

Рисунок 2 — Результаты сложности расшифровки замкнутых классов Поста запросами на сравнение при $n, k \to \infty$.

На рисунке 2 схематично приведены основные результаты четвертой главы. Если класс выделен квадратом с обеими диагоналями, тогда этот класс 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 15. *Замкнутые классы решетки Поста по характеру известной на данный момент сложности точной расшифровки запросами на сравнение в случае, когда n растет, а k не меняется, разделены на три группы:*

1. *точная оценка*
   - $\varphi_{CQ}(O_1, n, 1) = \left]\, \log_3 n \,\right[$;
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* $n \to \infty$;
   - $\varphi_{CQ}(C_2, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
   - $\varphi_{CQ}(C_4, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
   - $\varphi_{CQ}(A_2, n, k) \asymp \log n$ *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* $n \to \infty, k \geqslant 2$;
   - $\varphi_{CQ}(D_2, n, k) \asymp \log n$ *при* $n \to \infty$;
   - $\varphi_{CQ}(D_3, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
   - *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_1^i, n, k) \asymp \log n$ *при* $n \to \infty$, $k \geqslant 2$;
   - *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* $n \to \infty$, $k \geqslant 2$;
   - $\varphi_{CQ}(S_1, n, k) \asymp \log n$ *при* $n \to \infty$;
   - $\varphi_{CQ}(S_3, n, k) \asymp \log n$ *при* $n \to \infty$;
   - $\varphi_{CQ}(S_5, n, k) \asymp \log n$ *при* $n \to \infty$;
   - $\varphi_{CQ}(L_1^*, n, k) \asymp \log n$ *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* $n \to \infty$.

Работу завершает **заключение**, в котором излагаются основные результаты проведенного исследования.

Благодарности. Автор приносит благодарность профессорско-преподавательскому составу Ташкентского филиала МГУ имени М. В. Ломоносова и механико-математического факультета МГУ имени М. В. Ломоносова, а особенно коллективу кафедры математической теории интеллектуальных систем, за опыт и знания, полученные от них на протяжении всего периода обучения.

Особую признательность автор выражает доктору физико-математических наук, профессору Эльяру Эльдаровичу Гасанову за научное руководство и неугасаемое внимание к работе, а также Тимуру Рашидовичу Сытдыкову и кандидату физико-математических наук Александру Павловичу Пивоварову, чьи ценные замечания и предложения способствовали существенному улучшению текстов статей, которые стали основой данной работы.

Автор благодарит своих родителей: Быстрыгову Светлану Викторовну и Быстрыгова Виктора Николаевича — за поддержку и веру в успех на академическом и научном поприще, а также своего школьного учителя математики Киреву Тамару Михайловну и преподавателя математики в колледже Ильину Наталию Ивановну за занятия, которые привили интерес к математике, а также их внимание и раннее приобщение к олимпиадной деятельности.

Объем и структура работы. Диссертация состоит из введения, 4 глав, заключения. Полный объём диссертации составляет 134 страницы, включая 4 рисунка. Список литературы содержит 45 наименований.

Глава 1. Основные понятия и обозначения

Первой глава состоит из трех разделов и предназначена для определения обозначений и понятий, используемых в работе.

В первом разделе перечисляются значения как распространенных обозначений (например, символы $|, \&$) для того, чтобы избежать неоднозначного их трактования в результатах, так и вводимых впервые в работе функций.

Во втором разделе вводится определение используемых в работе типов запросов, понятия *сложности расшифровки запросами определенного типа множества функций*.

В третьем разделе приводятся описания классов функций, являющихся объектом исследования данной работы: класса функций ограниченного веса и всех замкнутых классов решетки Поста.

1.1 Базовые определения

Если a — вещественное число, под $]a[$ будем понимать наименьшее целое, не меньшее a, под $[a]$ — наибольшее целое, не большее a, под $a \bmod b$ — остаток от деления a на b, под $|a|$ — модуль числа a. Если A — компонента связности в графе, то под $|A|$ будем понимать ее размер, т.е. количество вершин в 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 $A(n) \asymp B(n)$.

Под операцией $\oplus$ будем понимать сложение по модулю 2.

Под операцией $|$ будем понимать дизъюнкцию (логическое ИЛИ).

Под операцией $\&$ будем понимать конъюнкцию (логическое И).

Обозначим через $G(k, m)$ функцию $k \cdot [m/(k+1)] + (m \bmod (k+1))$.

Обозначим через $S_{n,k}$ значение $\max_{p\in\mathbb{N}, 1\leqslant p<k} (2^p - 1)\alpha(n-p, k-p)$, где $\mathbb{N}$ — множество натуральных чисел.

1.2 Сложность расшифровки и типы запросов

Будем обозначать через $P(n)$ множество булевых функций арности n. Пусть учителем выбрана функция $f \in M(n)$, где $M(n)$ — подмножество $P(n)$, при этом ученику известно $M(n)$ и неизвестен выбор f. Тогда определим рассматриваемые типы запросов ученика и ответы на эти запросы учителем следующим образом.

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 $f(x) - f(y)$.

Через $f \equiv g$ при $f, g \in P(n)$ будем обозначать ситуацию, в которой для любого $x \in \{0,1\}^n$ верно равенство $f(x) = g(x)$. Под *запросом на ограниченную эквивалентность g для функции f* принято считать функцию $g \in M(n)$, а под ответом на указанный запрос — слово YES, если $f \equiv g$, и любой набор y, такой, что $f(y) \neq g(y)$, в противном случае. Под *запросом на расширенную эквивалентность g* понимают функцию $g \in P(n)$, под ответом на этот запрос считают слово YES, если $f \equiv g$, и какой-то набор y, такой, что $f(y) \neq g(y)$, в противном случае.

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 $T \in \{MQ, CQ, EQ, XEQ\}$ будем обозначать тип запроса, где MQ — запрос на значение, CQ — запрос на сравнение, EQ — запрос на ограниченную эквивалентность, XEQ — запрос на расширенную эквивалентность.

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 *класс можно расшифровать запросами на сравнение*, если существует алгоритм расшифровки запросами на сравнение, который для любой функции класса расшифровывает ее. Если про класс нельзя сказать, что его можно расшифровать запросами на сравнение, тогда будем, говорить, что *класс нельзя расшифровать запросами на сравнение.*

1.3 Исследуемые классы функций

Введем обозначения для рассматриваемых в работе классов функций: класса функций ограниченного веса и замкнутых классов Поста.

Для каждой функции $f \in P(n)$ под $|f|$ будем понимать вес f, т.е. количество единиц в векторе значений функции f. Под $F(n, k, i)$, где $k \in (0, 2^n], i \in [0, 2^n)$, $0 \leqslant i \leqslant k$, будем понимать множество $\{f \in P(n) : i \leqslant |f| \leqslant k\}$, иными словами, *класс функций ограниченного веса* $F(n, k, i)$ — множество булевых функций арности n, вес которых лежит в диапазоне $[i, k]$.

Под $\varphi_T(F, n, k, i)$ будем понимать $\varphi_T(F(n, k, i), n)$.

Напомним обозначения замкнутых классов Поста (рис. 1). Классы из “правой” половины решетки Поста заведомо опустим, 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.

Под словами *классы счетной этажерки* будем понимать все классы $F^i_4, F^\infty_4, F^i_3, F^\infty_3, F^i_2, F^\infty_2, F^i_1, F^\infty_1$ в силу визуального сходства части решетки Поста по этим классам с этажеркой.

C_1 — все функции двузначной логики.

C_2 — все функции, сохраняющие 1.

C_3 — все функции, сохраняющие 0.

C_4 — пересечение C_2, C_3.

A_1 — все монотонные функции.

A_2 — все монотонные функции, сохраняющие 1.

A_3 — все монотонные функции, сохраняющие 0.

A_4 — пересечение A_2, A_3.

F_4^i — класс функций, таких что для любого j $(2 \leqslant j \leqslant i)$ верно, что любые j наборов, на которых f обращается в 0, имеют общую нулевую компоненту.

F_4^∞ — класс функций, таких что для любого натурального j $(2 \leqslant j)$ верно, что любые j наборов, на которых f обращается в 0, имеют общую нулевую компоненту.

F_3^i — пересечение класса F_4^i и A_2.

F_1^i — пересечение класса F_4^i и C_4.

F_2^i — пересечение класса F_4^i и A_4.

D_3 — все самодвойственные функции.

D_1 — все самодвойственные функции, сохраняющие 0.

D_2 — пересечение классов D_3, F_2^2.

S_1 — все логические суммы вида $x_{i_1} \vee x_{i_2} \vee \ldots \vee x_{i_k}$.

S_3 — S_1 и константа 1.

S_5 — S_1 и константа 0.

S_6 — объединение классов S_3, S_5.

L_1 — все линейные функции.

L_5 — пересечение L_1, D_3.

L_2 — пересечение L_1, C_2.

L_3 — пересечение L_1, C_3.

L_4 — пересечение L_2, L_3, L_5.

$O_9 = \{1, 0, x, \overline{x}\}$.

$O_6 = \{0, 1, x\}$.

$O_4 = \{x, \overline{x}\}$.

$O_5 = \{1, x\}$.

$O_8 = \{0, x\}$.

$O_1 = \{x\}$.

Если класс K решетки Поста помечен символом *, то значит рассматривается этот класс без константы 0. Например, $C_1^* = C \setminus \{f \equiv 0\}$.

Для любого из перечисленных классов Поста R под $R(n,k)$ будем понимать множество функций класса R, зависящих от n переменных, из которых существенных переменных не более k.

В силу этих обозначений для любого из перечисленных классов Поста R под $\varphi_T(R,n,k)$ будем понимать $\varphi_T(R(n,k),n)$.

Будем говорить, что *класс R решетки Поста можно расшифровать запросами на сравнение*, если для любых целых n,k, где $n \geqslant k \geqslant 1$, n — арность функций, k — верхняя оценка на число существенных переменных, можно расшифровать запросами на сравнение класс $R(n,k)$. Если про класс R нельзя сказать, что его можно расшифровать запросами на сравнение, тогда будем, говорить, что *класс R нельзя расшифровать запросами на сравнение.*

Глава 2. Расшифровка функций ограниченного веса

В данной главе рассматривается параметро-эффективная расшифровка класса функций ограниченного веса, где учителем выбирается функция арности n, у которой в векторе значений число единиц из отрезка $[i, k]$, при этом ученику неизвестен выбор учителя, но известны числа n, k, i. При этом числа n, k, i связаны следующими соотношениями: $0 \leqslant i \leqslant k \leqslant 2^n, 0 < k, i < 2^n$, поскольку в случае функций веса 0 или веса 2^n задача становится тривиальной. Задача расшифровки рассматривается для четырех типов запросов в отдельности: запросы на значение, на сравнение, на расширенную и ограниченную эквивалетность. В первых трех разделах главы демонстрируется значение сложности расшифровки для всевозможных допустимых значений параметров n, k, i для запросов на значение и запросов на эквивалентность. В четвертом разделе главы исследуется сложность расшифровки запросами на сравнение. Относительно других типов запросов определение сложности расшифровки для запросов на сравнение для произвольных допустимых n, k, i не столь тривиальна. В указанном разделе приводятся точные оценки сложности расшифровки запросами на сравнение функций веса 1, 2 и 3. Помимо этого, демонстрируются алгоритмы расшифровки функций ограниченного веса, основанный на разбиении всех 2^n наборов в группы определенного размера. В конце раздела приводится доказательство теоремы о порядке сложности расшифровки запросами на сравнение.

В этой главе, как и в последующих, для доказательстве того, что сложность расшифровки φ_T какого-то класса функций запросами типа T равна определенной функции ψ, необходимо показать, что, во-первых, удовлетворяется неравенство $\varphi_T \leqslant \psi$, называемое верхней оценкой и говорящее, что сложность расшифровки не больше чем фиксированная функция, и, во-вторых, верно неравенство $\varphi_T \geqslant \psi$, называемое нижней оценкой и говорящее, что сложность расшифровки не меньше той же фиксированной функции. При этом для доказательства верхней оценки достаточно привести алгоритм расшифровки запросами типа T, такой, что любую функцию из рассматриваемого класса можно восстановить при помощи этого алгоритма за не более ψ запросов типа T. А для доказательства нижней оценки достаточно показать, что какой бы алгоритм расшифровки запросами типа T ученик не использовал в игре с учителем, учитель для этого алгоритма загадает такую функцию из класса, что

ученик сумеет восстановить ее лишь после того, как задаст как минимум ψ запросов. Иными словами, далее при доказательстве верхних оценок сложности расшифровки мы будем “играть” за ученика, помогая ему выбрать наилучший алгоритм, а при доказательстве нижних оценок — за учителя, помогая так отвечать на запросы ученика, чтобы заставить последнего потратить как можно больше запросов независимо от его выбора алгоритма расшифровки.

2.1 Расшифровка запросами на значение

Для доказательства теоремы о сложности расшифровки функций ограниченного веса запросами на значение докажем сначала лемму о сложности расшифровки функций фиксированного веса, то есть для функций веса k, а затем лемму о сложности расшифровки функций веса из полуинтервала $[i, k)$.

Лемма 1. *Сложность расшифровки класса $F(n, k, k)$ запросами на значение равна $2^n - 1$.*

Доказательство. Верхняя оценка. Заметим, что в работе [19], было доказано неравенство $\varphi_{MQ}(F, n, 1, 1) \leqslant 2^n - 1$. Докажем данную верхнюю оценку для произвольного k. Запросим значение на любых $2^n - 1$ наборах. Если среди ответов на эти запросы встретилось ровно k единиц, значит значение функции на неопрошенном наборе равно 0, иначе 1. Соответственно, функция f восстановлена.

Нижняя оценка. Если ученик повторит запрос, просто ответим на него также, как отвечали прежне. Поэтому можно считать, что ученик не повторяет запросы. На первые $2^n - 2 - (k - 1)$ запросов ученика будем отвечать 0, на следующие $k - 1$ запросов ответим 1. Для ученика останутся неопрошенными 2 набора и не найдена одна единица загаданной функции. Поэтому он будет вынужден задать еще один запрос. □

Лемма 2. *Сложность расшифровки класса $F(n, k, i)$ запросами на значение равна 2^n при $i < k$.*

Доказательство. Нижняя оценка. Если ученик повторит запрос, просто ответим на него так же, как отвечали прежде. Поэтому можно считать, что ученик

не повторяет запросы. На первые $2^n - k$ запросов учитель отвечает 0, на следующие $k-1$ запросов отвечает 1. В силу того что для любого $i < k$ в $F(n,k,i)$ лежат все функции веса k и $k-1$, ученик вынужден узнавать значение и на оставшемся неопрошенном наборе.

Верхняя оценка. Оценка в 2^n запросов очевидна в силу того, что за 2^n запросов полностью восстанавливается вектор значений функции. □

Теорема 1. *Сложность расшифровки класса $F(n,k,i)$ запросами на значение равна*

$$\varphi_{MQ}(F,n,k,i) = \begin{cases} 2^n - 1 & \text{при } i = k, \\ 2^n & \text{при } i < k. \end{cases}$$

Доказательство. Доказательство следует из лемм 1 и 2. □

2.2 Расшифровка запросами на эквивалентность

Перейдем к доказательству сложности расшифровки функций ограниченного веса для двух типов запросов на эквивалентность: расширенную (XEQ) и ограниченную (EQ).

Отметим, что в постановке задачи, а соответственно и в формулировке теорем 2 и 3 работы [44] не рассматривался случай $i < k = 2^n$, а также имеются неточности в доказательстве этих теорем, поэтому приведем далее их дополненные формулировки и доказательства.

Теорема 2. *Сложность расшифровки класса $F(n,k,i)$ запросами на расширенную эквивалентность равна*

$$\varphi_{XEQ}(F,n,k,i) = \min(k, 2^n - i).$$

Доказательство. Верхняя оценка. Пусть $k \leqslant 2^n - i$. В роли ученика в качестве первого запроса мы отправим учителю константу 0. В силу того что $k > 0$, в ответ мы получим YES или набор, на котором значение загаданной функции равно 1, тем самым мы раскроем информацию об одной единице. Далее отправим функцию, которая равна нулю всюду, за исключением раскрытой единицы, и в ответ получим YES или информацию о второй единице. Действуя дальше

аналогично, не более чем за k запросов мы восстановим загаданную функцию. Если на запрос с номером $q < k$ придет в ответ YES, функция будет расшифрована, иначе если, задав ровно k запросов, мы ни разу в ответ не получим YES, то загадана функция веса k и все k единиц мы нашли.

Пусть $k > 2^n - i$. Нулей у загаданной функции не меньше $2^n - k$ и не больше $2^n - i$. Применим приведенный выше алгоритм, заменив повсюду 0 на 1, то есть теперь будем раскрывать за каждый запрос нуль функции. В роли ученика в качестве первого запроса мы отправим учителю константу 1. Если в ответ придет YES, то функция расшифрована. Иначе, в ответ мы получим набор, на котором значение загаданной функции равно 0, тем самым мы раскроем информацию об одном нуле. Далее отправим функцию, которая равна единице всюду, за исключением раскрытого нуля, и в ответ получим информацию YES или информацию о втором нуле. Действуя дальше аналогично, не более чем за $2^n - i$ запросов мы найдем все нули функции и значит восстановим загаданную функцию. Если на запрос с номером $q < 2^n - i$ придет в ответ YES, функция будет расшифрована, иначе если, задав ровно $2^n - i$ запросов, мы ни разу в ответ не получим YES, то загадана функция с максимально возможным числом $2^n - i$ нулей и все $2^n - i$ нулей мы нашли.

Нижняя оценка. В силу того что $F(n, k, k) \subseteq F(n, k, i)$ и $F(n, i, i) \subseteq F(n, k, i)$, учитель имеет право загадать любую функцию веса ровно k или веса ровно i.

Докажем нижнюю оценку. Положим A — множество всех 2^n двоичных n-местных наборов, значение загаданной функции на которой пока неизвестно.

На каждый очередной запрос g ученика отвечаем следующим образом:

1. если в A лежит набор x, на котором g обращается в 1, тогда возвращаем ученику x и удаляем x из A,
2. иначе, если в A лежит набор x, на котором g обращается в 0, тогда возвращаем ученику x и удаляем x из A.

Если не сработал ни один из пунктов, значит множество A пустое, а значит вся функция f восстановлена. Если учитель воспользовался первым пунктом своей стратегии, то он раскрыл ученику информацию об одном нуле. Если учитель воспользовался вторым пунктом своей стратегии, то он раскрыл ученику информацию об одной единице.

За один запрос ученик узнает информацию о значении загаданной функции ровно на одном наборе. Если в какой-то момент уже найдено k единиц, то

ученик расшифровал функцию. Если в какой-то момент он найдет $2^n - i$ нулей, то есть загадана функция веса i, то ученик вновь расшифровал функцию. Если ученик вынуждает учителя постоянно пользоваться пунктом 1, тогда он восстановит функцию за $2^n - i$ запросов. Если он вынуждает учителя постоянно пользоваться пунктом 2, тогда он восстановит функцию за k запросов.

Заметим, что ученику невыгодно опрашивать учителя так, чтобы тот использовал оба пункта своей стратегии ответов. Докажем от противного. Пусть существует последовательность запросов ученика, которая состоит меньше, чем из $\min(k, 2^n - i)$ запросов, и расшифровывает функцию. Задав строго меньше $\min(k, 2^n - i)$ запросов, ученик не найдет все k единиц или все $2^n - i$ нулей, а значит любой из оставшихся неопрошенных наборов, а такие существуют, поскольку $\min(k, 2^n - i) - 1 < 2^n$, может быть как нулем, так и единицей функции, а значит функция не восстановлена. Противоречие.

□

Теорема 3. *Сложность расшифровки класса $F(n, k, i)$ запросами на ограниченную эквивалентность равна*

$$\varphi_{EQ}(F, n, k, i) = \begin{cases} k & при\ i = 0, \\ 2^n - 1 & при\ 0 < i = k, \\ 2^n & при\ 0 < i < k < 2^n, \\ 2^n - i & при\ 0 < i < k = 2^n. \end{cases}$$

Доказательство. Верхняя оценка. **Случаи $0 < i = k$ и $0 < i < k < 2^n$.** Заметим, что в работе [19], было доказано неравенство $\varphi_{EQ}(F, n, 1, 1) \leqslant 2^n - 1$. Докажем данную верхнюю оценку для произвольного k. Положим A — множество всех 2^n двоичных n-местных наборов, про которые мы пока не знаем, чему равно значение загаданной функции на них. A_0, A_1 — множество наборов, на которых значение загаданной функции равно 0 и 1 соответственно. Изначально, оба множества A_0, A_1 пусты. Каждый запрос формируем следующим образом. Выбираем любое подмножество B мощности $(k - |A_1|)$ из A и учителю передаем функцию g, которая равна единице на наборах из $B \cup A_1$ и равна нулю на остальных наборах (то есть наборах из $A_0 \cup (A \setminus B)$). Если учитель возвращает в ответ запрос x, то:

1. если $x \in B$, тогда удаляем x из A и добавляем в A_0,
2. если $x \notin B$, тогда удаляем x из A и добавляем в A_1.

Если учитель возвращает в ответ запрос YES, то функция восстановлена.

В **случае** $0 < i = k$, как только $|A| + |A_1| = k$, тогда делаем вывод, что загаданная функция равна единице на всех наборах в A и значит f восстановлена. В **обоих случаях** $0 < i = k$ **и** $0 < i < k < 2^n$, если $|A_1| = k$, тогда загаданная функция также восстановлена.

Оценим количество запросов в каждом из рассматриваемых случаев.

В **случае** $0 < i = k$ достаточно задать $2^n - 1$ запросов, так как за один запрос раскрывается значение ровно на одном наборе. Причем, если $|A_1| < k$ после опроса $2^n - 1$ запросов, то значение функции на оставшемся наборе равно единице, иначе 0. В **случае** $0 < i < k < 2^n$, если $|A_1| < k$ после опроса $2^n - 1$ запросов, то придется узнать значение на еще одном наборе.

Случай $i = 0$. Заметим, что в классе $F(n, k, i)$ лежат все функции, используемые в алгоритме расшифровки, приведенном для случая $k \leqslant 2^n - i$ при доказательстве верхней оценки $\varphi_{XEQ}(F, n, i, k)$ теоремы 2, — функции веса $0,\ 1,\ 2,\ \ldots,\ k-1$. Вследствие этого можем применить его и для получения верхней оценки $\varphi_{EQ}(F, n, i, k)$ независимо от того, справедливо неравенство $k \leqslant 2^n - i$ или нет.

Случай $0 < i < k = 2^n$. Заметим, что в классе $F(n, k, i)$ лежат все функции, используемые в алгоритме расшифровки, приведенном для случая $k > 2^n - i$ при доказательстве верхней оценки $\varphi_{XEQ}(F, n, i, k)$ теоремы 2, — функции веса $2^n, 2^n - 1, 2^n - 2, \ldots, i + 1$. Вследствие этого можем применить его и для получения верхней оценки $\varphi_{EQ}(F, n, i, k)$ независимо от того, справедливо неравенство $k > 2^n - i$ или нет.

Нижняя оценка. Если в ответ на свой запрос ученик получил значение функции на каком-то наборе, а позже отправил запрос-функцию, которая отличается в соответствующем наборе от верного значения, мы, выполняя роль учителя, в качестве ответа вновь отправляем ученику этот набор. Поэтому можно считать, что ученик не посылает такие “бесполезные” запросы.

Рассмотрим два случая $k < 2^n$ и $k = 2^n$.

1. Доказательство нижней оценки для **случая** $k < 2^n$**.**

   Случай $i = 0$. В качестве загаданной функции учитель мог выбрать любую функцию веса k. На каждый из первых k запросов ученика будем возвращать в ответ любой набор, на котором функция, присланная учеником, равна 0. Такой набор точно существует, поскольку $k < 2^n$. Тем самым за один запрос мы раскроем ученику одну единицу.

Случай $0 < i = k$. На каждый из первых $2^n - 2 - (k-1)$ запросов будем возвращать набор, на котором функция ученика равна 1. Такой набор точно найдется, так как $i > 0$. Следовательно, за каждый такой запрос, мы раскроем информацию об одном нуле. На каждый из $k-1$ последующих запросов будем возвращать набор, на котором функция ученика равна 0. Такой набор точно найдется, так как $k < 2^n$. Таким образом, за каждый такой запрос мы раскроем информацию об одной единице. Ученику неизвестно значение на ровно двух наборах и неизвестна одна единица. Поэтому он вынужден сделать еще один запрос.

Случай $0 < i < k$. На каждый из первых $2^n - k$ запросов будем возвращать набор, на котором функция ученика равна 1. Такой набор точно найдется, поскольку $i > 0$. Следовательно, за каждый такой запрос мы раскроем информацию об одном нуле. В ответ на каждый из следующих $k-1$ запросов вернем набор, на котором функция ученика равна 0. Такой набор точно найдется, поскольку $k < 2^n$, а значит, раскроем ровно одну единицу. Оставшийся неопрошенный набор может быть как нулем, так и единицей, т.е. может быть загадана функция веса $k-1$ или k, поэтому ученику следует задать еще один запрос.

2. Доказательство нижней оценки для **случая** $k = 2^n$**.**

   Случай $i = 0$. В качестве загаданной функции учитель мог выбрать абсолютно любую булеву функцию арности n. Пронумеруем все 2^n наборов как $x_1, x_2, x_3, \ldots, x_{2^n}$. На i-й запрос ученика будем отвечать x_i. Ясно, что ученик в этом случае вынужден опросить все $2^n = k$ наборов.

   Случай $0 < i = k = 2^n$. Отсутствует в формулировке задачи, поскольку является тривиальным.

   Случай $0 < i < k = 2^n$. В качестве загаданной функции учитель мог выбрать любую функцию веса i, иными словами функцию с $2^n - i$ нулями. На каждый из первых $2^n - i$ запросов ученика будем возвращать в ответ любой набор, на котором функция, присланная учеником, равна 1. Такой набор точно существует, поскольку $i > 0$. Тем самым за один запрос мы раскроем ученику один нуль функции.

□

2.3 Расшифровка запросами на сравнение

Прежде чем переходить в этом разделе к доказательствам основных результатов, связанных со сложностью расшифровки функций ограниченного веса запросами на сравнение, обозначим некоторые особенности рассматриваемой задачи в случае запросов на сравнение.

Если каждый из 2^n наборов представлять вершиной графа, то будем считать, что один запрос на сравнение (x, y) объединяет в одну компоненту связности компоненты связности, в которых содержатся вершины x и y. Компонентами связности запроса (x, y) будем называть две компоненты связности, в которых лежат наборы x и y.

Будем говорить, что запрос (x, y) покрывает наборы x и y или наборы x, y покрыты запросом (x, y). Аналогично будем говорить, что множество запросов на сравнение W покрывает наборы $x_1, x_2, \ldots, x_t$, если каждый набор $x_i, i \in \{1, t\}$, покрыт каким-то запросом из W.

Замечание 1. *Если на все запросы на сравнение, которые объединили вершины $v_1, v_2, \ldots, v_s$ в одну компоненту связности, в ответ был получен 0, то значение загаданной функции $f \in F(n, k, i)$ на соответствующих наборах одинаковое, т.е. все наборы лежат в одном классе эквивалентности. При этом в случае $s > k$ значение функции равно 0, поскольку ее вес строго меньше s.*

Если ответ на запрос (x, y) не равен нулю, то однозначно восстанавливается значение на обоих наборах x, y. Аналогично если при формировании компоненты связности из вершин-наборов хотя бы раз был получен ответ 1 или -1, то однозначно восстанавливается значение на всех наборах этой компоненты.

2.3.1 Вспомогательные утверждения

В начале этого раздела доказывается критерий возможности использования запросов на сравнение в задаче точной параметро-эффективной расшифровки произвольного множества булевых функций. Далее доказываются леммы,

результаты которых используются при доказательстве теорем из следующих разделов. В частности, лемма 3 понадобится при доказательстве сложности расшифровки функций веса 1, 2 и функций веса, ограниченного снизу 0 (теоремы 6, 7), леммы 6, 7, 8 при доказательстве нижней оценки сложности расшифровки функций веса k (лемма 23).

Напомним, что через $G(k,m)$ мы обозначили функцию $k \cdot [m/(k+1)] + (m \bmod (k+1))$.

Теорема 4. *Класс булевых функций расшифровать запросами на сравнение нельзя тогда и только тогда, когда ему принадлежат обе константные функции* $0, 1$.

Доказательство. Докажем достаточность. Обе константные функции на любом запросе на сравнение возвращают значение 0, поэтому распознать, какая из них загадана невозможно.

Докажем необходимость. Обозначим рассматриваемое множество булевых функций через M. От противного. Пусть $\{0,1\} \not\subseteq M$, но класс M расшифровать нельзя.

Поскольку класс M расшифровать нельзя, то какой бы алгоритм расшифровки $A \in \mathcal{A}_M^{CQ}$ не взять, то найдется функция $f \in M$, которую этот алгоритм не сможет расшифровать, то есть нельзя отличить ее от какой-то другой функции из класса M. Рассмотрим следующий алгоритм Q. В качестве запросов на сравнение возьмем множество всевозможных пар (a,b). Зафиксируем любую функцию $f \in M$.

1. Если $f \in \{0,1\}$, то ответы на все запросы будут равны 0, что отличает ее от остальных неконстантных функций, для которых среди ответов на запросы обязательно встретится хотя бы одна 1.
2. Если $f \notin \{0,1\}$, то среди ответов на запросы встречается хотя бы одна 1, что отличает эту функцию от константной. Рассмотрим любую отличную от f функцию $g \in M, g \notin \{0,1\}$. Функции f, g отличаются, значит, существует набор a такой, что $f(a) \neq g(a)$. Возможны два случая:
   - существует набор b такой, что $f(b) = g(b)$, тогда на запросе (a,b) функции f, g вернут разные ответы;

– для любого набора b верно неравенство $f(b) \neq g(b)$, тогда на запросе (c, d), где $f(c) = 0, f(d) = 1$, функция f вернет ответ -1, а функция g вернет ответ 1.

Следовательно, алгоритм Q расшифровывает класс M, получили противоречие.

□

Непосредственно из этой теоремы получаем, что для случая $0 = i < k = 2^n$ класс $F(n, k, i)$ расшифровать запросами на сравнение нельзя. Поэтому далее в формулировках результатов будем напоминать о недопустимости комбинации $0 = i < k = 2^n$.

Лемма 3. *Для того чтобы Q наборов объединить в компоненты связности размера не менее p, где $p \leqslant Q$, необходимо и достаточно задать в точности $G(p-1, Q) = (p-1) \cdot [Q/p] + (Q \bmod p)$ запросов на сравнение.*

Доказательство. Пусть $Q = pq + r$, где $r \in [0, p-1], q > 0$, q, p — целые числа. Тогда возможны два случая.

Случай $r = 0$. *Достаточность.* Если объединять наборы в компоненты связности, так чтобы получились только компоненты связности размера p, тогда необходимо ровно $Q \cdot (p-1)/p$ запросов. Поскольку все Q наборов распадутся на Q/p множеств мощности равной p. А чтобы объединить p элементов в одну компоненту связности необходимо в точности $p-1$ запросов на сравнение, соответствующее количеству ребер в дереве из p вершин. Общее число использованных ребер в точности соответствует определению $G(p-1, Q)$.

Необходимость. Рассмотрим общий случай итоговой системы множеств после применения операций объединения. Пусть Q_1 наборов распадутся на компоненты связности размера p, Q_2 — размера p_2, …, Q_r — размера p_r, при этом $p < p_2 < \ldots < p_r, Q_1 + Q_2 + \ldots + Q_r = Q$. Тогда для этого потребуется в точности $Q_1(p-1)/p + Q_2(p_2-1)/p_2 + \ldots + Q_r(p_r-1)/p_r$ запросов на сравнение.

Но учитывая, неравенство $\frac{t_1-1}{t_1} < \frac{t_2-1}{t_2}$ при $0 < t_1 < t_2$, получаем цепочку неравенств

$$Q\frac{p-1}{p} - (Q_1\frac{p-1}{p} + Q_2\frac{p_2-1}{p_2} + \ldots + Q_r\frac{p_r-1}{p_r}) <$$
$$< Q\frac{p-1}{p} - \frac{p-1}{p}(Q_1 + Q_2 + \ldots + Q_r) = \frac{p-1}{p} \cdot (Q - Q) = 0.$$

Следовательно, $Q\frac{p-1}{p} < (Q_1\frac{p-1}{p} + Q_2\frac{p_2-1}{p_2} + \ldots + Q_r\frac{p_r-1}{p_r})$.

Иными словами, для объединения изначальных Q наборов в компоненты связности размера не менее p оптимальнее всего объединить их в компоненты связности в точности равной p.

Случай $r > 0$. *Достаточность.* Аналогично доказательству достаточности для случая $r = 0$. Если объединять $Q - r$ наборов в компоненты связности размера ровно p и r наборов в одну компоненту связности размера r, то потребуется ровно $[Q/p] \cdot (p-1) + (r-1)$ запросов. Поскольку $Q \geqslant p$, то как минимум одна компонента связности размера p создана. Затем еще одним ребром присоединим компоненту связности размера r к любой компоненте размера p. Использованное число ребер $[Q/p] \cdot (p-1) + r$ соответствует определению $G(p-1, Q)$.

Необходимость. От противного. Пусть существуют Q, p, x, r — целые положительные, такие что $Q = px + r, 0 < r < p$, и существует порядок объединения исходных Q наборов в компоненты связности размера не менее p, в котором используется $x \cdot (p-1) + (r-1)$ запросов на сравнение.

Учитывая известный факт, что в графе с V вершинами, E ребрами и K компонентами связности справедливо неравенство $E + K \geqslant V$, исходные Q вершин после добавления $x \cdot (p-1) + (r-1)$ ребер распадутся на не менее $Q - (x \cdot (p-1) + r - 1) = x + 1$ компонент связности. При этом, в текущем рассматриваемом случае выше утверждается, что все получившиеся компоненты связности имеют мощность не менее p. Соответственно, суммарно во всех получившихся компонентах связности не менее $p(x+1) = px + p$ вершин. Получили противоречие с исходным количеством вершин $Q = px + r < px + p$. □

Лемма 4. *Пусть* n, k, x *— целые числа, такие, что* $0 < k < 2^n, x \in [1, k]$, *тогда функция* $u(x) = k \cdot [(2^n - x)/(k+1)] + ((2^n - x) \bmod (k+1)) + (x-1)$ *неубывающая.*

Доказательство. Пусть $2^n = q \cdot (k+1) + r$, где q, r — целые неотрицательные числа, такие, что $r \in [0, k]$. Заметим, что $q > 0$, поскольку по условию $2^n > k$.

Если $x < r$, то $u(x+1) = kq + (r - (x+1)) + x, u(x) = kq + (r-x) + (x-1), u(x+1) = u(x)$.

Если $x = r$, то $u(x+1) = k(q-1) + k + x, u(x) = kq + 0 + (x-1), u(x+1) = u(x) + 1$.

Если $x > r$, то $u(x+1) = k(q-1) + ((k+1) - (x+1-r)) + x, u(x) = k(q-1) + ((k+1) - (x-r)) + (x-1), u(x+1) = u(x)$. □

Лемма 5. *Пусть n, k — целые числа, такие, что $0 < k < 2^n$, тогда при $2^n \bmod (k+1) = 0$ справедливо равенство $G(k, 2^n - 1) = G(k, 2^n)$, а при $2^n \bmod (k+1) > 0$ — равенство $G(k, 2^n - 1) = G(k, 2^n) - 1$.*

Доказательство. Пусть $2^n = q \cdot (k+1) + r$, где q, r — целые неотрицательные числа, такие, что $r \in [0, k]$.

Если $r = 0$, то $G(k, 2^n - 1) = k \cdot [(2^n - 1)/(k+1)] + ((2^n - 1) \bmod (k+1)) = k(q-1) + k$, а $G(k, 2^n) = k \cdot [2^n/(k+1)] + (2^n \bmod (k+1)) = kq$.

Если $r > 0$, то $G(k, 2^n - 1) = k \cdot [(2^n - 1)/(k+1)] + ((2^n - 1) \bmod (k+1)) = kq + (r-1)$, а $G(k, 2^n) = k \cdot [2^n/(k+1)] + (2^n \bmod (k+1)) = kq + r$. □

Лемма 6. *Пусть p целое, тогда функция $h(p) = 2 \cdot [p/3] + (p \bmod 3)$ неубывающая.*

Доказательство. Если $p \bmod 3 < 2$, то $h(p+1) = h(p) + 1$, а если $p \bmod 3 = 2$, то $h(p+1) = h(p)$. □

Лемма 7. *Пусть k, n — целые числа, такие, что $0 < k < 2^n$, тогда справедливо неравенство $2 \cdot \left[(2^n - (k+2))/3\right] + ((2^n - (k+2)) \bmod 3) > 2 \cdot \left[2^n/3\right] - \left]2/3 \cdot (k+3)\right[$.*

Доказательство. Пусть $2^n = 3q_1 + r_1, (k+2) = 3q_2 + r_2$, где q_1, q_2, r_1, r_2 — целые неотрицательные числа, такие, что $r_1 \in [1, 2], r_2 \in [0, 2]$.

При $r_1 \geqslant r_2$ верно $2 \cdot [(2^n - (k+2))/3] + ((2^n - (k+2)) \bmod 3) = 2(q_1 - q_2) + r_1 - r_2$, а $2 \cdot [2^n/3] -]2/3 \cdot (k+3)[= 2q_1 - 2q_2 -]2/3 \cdot (r_2 + 1)[$, неравенство справедливо при любом $r_2 \in [0, 2]$.

При $r_1 < r_2$, т.е. $r_1 = 1, r_2 = 2$, верно $2 \cdot [(2^n - (k+2))/3] + ((2^n - (k+2)) \bmod 3) = 2(q_1 - q_2 - 1) + 2$, а $2 \cdot [2^n/3] -]2/3 \cdot (k+3)[= 2q_1 - 2q_2 - 2$ и имеет место искомое неравенство. □

Лемма 8. *Пусть даны четное число $v \geqslant 2$ и кортеж чисел $a_1, a_2, \ldots, a_t$, где $a_i \in \{1, 2\}, i \in [1, t]$, $a_1 \leqslant a_2 \leqslant \ldots \leqslant a_t$, $\sum_{i=1}^{t} a_i \geqslant v + 2$. Тогда существуют два разных подмножества $\{a_{j_1}, \ldots, a_{j_r}\}$ и $\{a_{h_1}, \ldots, a_{h_q}\}$ этого набора, такие, что $\sum_{m=1}^{r} a_{j_m} = \sum_{n=1}^{q} a_{h_n} = v$.*

Доказательство. В качестве первого множества возьмем суффикс этого набора с суммой чисел, равной v, т.е. $\{a_s, a_{s+1}, a_{s+2}, \ldots, a_t\}$, где $a_s + a_{s+1} + a_{s+2} + \ldots + a_t = v$. Тогда в первое множество либо не войдет $a_1 = 2$, либо не войдут $a_1 = 1, a_2 = 1$. Если в первое множество не войдет $a_1 = 2$, то в качестве второго множества возьмем $\{a_1, a_s, a_{s+1}, a_{s+2}, \ldots, a_{t-1}\}$. Если в первое множество не войдет $a_1 = a_2 = 1$, то в качестве второго множества возьмем $\{a_1, a_2, a_s, a_{s+1}, a_{s+2}, \ldots, a_{t-1}\}$ при $a_t = 2$ и $\{a_1, a_2, a_s, a_{s+1}, a_{s+2}, \ldots, a_{t-2}\}$ при $a_{t-1} = a_t = 1$. □

Лемма 9. *Пусть $n \geqslant 3$ и для целых положительных x_2, x_3 верно равенство $2^n = 2x_2 + 3x_3$, причем $x_2 \geqslant 4$. Тогда справедливо неравенство*

$$2[(2^n - 3x_3)/3] + ((2^n - 3x_3) \bmod 3) - 1 \leqslant x_2 + [(x_2 - 1)/2].$$

Доказательство. Пусть $2^n - 3x_3 = 3q + r$, где q, r — целые неотрицательные числа, $r \in [1, 2]$. Соответственно, имеем цепочку равенств $2x_2 = 2^n - 3x_3 = 3q + r$. Перепишем искомое неравенство при помощи введенных обозначений $2[2x_2/3] + (2x_2 \bmod 3) - 1 \leqslant x_2 + [(x_2 - 1)/2]$ или $2q + r - 1 \leqslant x_2 + [(x_2 - 1)/2]$. Рассмотрим два случая в зависимости от делимости x_2 на 2.

- Пусть $x_2 = 2s, s \geqslant 0$. Левая часть искомого неравенства имеет вид $(2q+r)-1 = (3q+r)-q-1 = 4s-(4s-r)/3-1 = 8s/3+r/3-1$. Правая часть искомого неравенства имеет вид $2s+(s-1) = 3s-1$. Рассмотрим разность $8s/3 + r/3 - 1 - 3s + 1 = -s/3 + r/3$. Данное неравенство справедливо при $s \geqslant 2$, что у нас изначально и выполняется, поскольку $x_2 \geqslant 4$.
- Пусть $x_2 = 2s + 1, s \geqslant 0$. Левая часть искомого неравенства имеет вид $(2q+r)-1 = (3q+r)-q-1 = 4s+2-(4s+2-r)/3-1 = 8s/3+(1+r)/3$. Правая часть искомого неравенства имеет вид $2s + 1 + s = 3s + 1$. Рассмотрим разность $8s/3 + (1 + r)/3 - 3s - 1 = -s/3 - 2/3 + r/3$. Приходим к выводу, что искомое неравенство справедливо при $s \geqslant 0$, что у нас изначально и выполняется.

□

Лемма 10. *Пусть $n \geqslant 3$ и для целых неотрицательных x_2, x_3 верно равенство $2^n = 2x_2 + 3x_3$, причем $x_2 < 4 \leqslant x_3$. Тогда имеет неравенство*

$$2[2^n/3] + (2^n \bmod 3) - 1 \leqslant 2^n - (x_2 + x_3) + [x_3/2].$$

Доказательство. Пусть $2^n = 3q + r, q \geqslant 0, r \in [1, 2]$. Рассмотрим 4 случая в зависимости от значений, которые принимает x_2.

1. $x_2 = 0$. Поскольку $x_3 = (2^n - 2x_2)/3$, то такой случай невозможен в силу не равенства $2^n \bmod 3$ нулю.
2. $x_2 = 1$. Поскольку $x_3 = (2^n - 2x_2)/3$, тогда $r = 2$, а $x_3 = q$. Искомое неравенство принимает вид $2q+1 \leqslant 3q+2-(1+q)+[q/2] = 2q+1+[q/2]$, которое верно при любом $q \geqslant 0$.
3. $x_2 = 2$. Поскольку $x_3 = (2^n - 2x_2)/3$, тогда $r = 1$, а $x_3 = q-1$. Искомое неравенство принимает вид $2q \leqslant 3q + 1 - (2 + q - 1) + [(q - 1)/2] = 2q + [(q - 1)/2]$, которое верно при любом $q - 1 \geqslant 0$, что и имеем, поскольку $x_3 = q - 1 \geqslant 4$.
4. $x_2 = 3$. Поскольку $x_3 = (2^n - 2x_2)/3$, то такой случай невозможен в силу не равенства $2^n \bmod 3$ нулю.

□

Лемма 11. *Пусть $n \geqslant 6$, q, r — целые положительные числа из выражения $2^n = 5 \cdot q + r, r \in [1, 4]$. Тогда справедливо неравенство*

$$2^n - 1.5 \cdot q - [0.5 \cdot r] < 3 \cdot 2^{n-2} - 2.$$

Доказательство. Рассмотрим разность $2^n - 1.5 \cdot q - [0.5 \cdot r] - 3 \cdot 2^{n-2} + 2 = 2^{n-2} - 3/10 \cdot (2^n - r) + 2 - [0.5 \cdot r] = 1/10 \cdot (10 \cdot 2^{n-2} - 12 \cdot 2^{n-2} + 3r + 20 - 10 \cdot [0.5 \cdot r]) = 1/10 \cdot (-2^{n-1} + 3r - 10 \cdot [0.5 \cdot r] + 20)$. Последнее выражение отрицательно при $n \geqslant 6$ независимо от того, какое значение принимает r. □

2.3.2 Верхние оценки сложности расшифровки $F(n, k, k)$

Перейдем к демонстрации двух алгоритмов расшифровки функций веса k. Алгоритм леммы 12 нам понадобится при доказательстве верхних оценок сложности расшифровки функций веса 2 и 3 (теорема 7 и лемма 13). Алгоритм теоремы 5 нам пригодится при доказательстве верхней оценки сложности расшифровки функций веса 3 (теорема 7).

Лемма 12. *Справедлива следующая верхняя оценка*

$$\varphi_{CQ}(F, n, k, k) \leqslant k[2^n/(k + 1)] + \max(0, (2^n \bmod (k + 1)) - 1).$$

Доказательство. Пусть $2^n = (k+1)q + r, 0 \leqslant r < (k+1), 0 < q$, q, r — целые числа.

Упорядочим произвольным образом все 2^n двоичных наборов и обозначим их через $x_0, x_1, \ldots, x_{2^n-2}, x_{2^n-1}$. Зададим q групп по k запросов. i-я $(0 \leqslant i < q)$ группа состоит из запросов вида $(x_{i\cdot(k+1)+j}, x_{i\cdot(k+1)+j+1})$, где j меняется от 0 до $k-1$ включительно.

Рассмотрим ответы для i-й группы запросов $(0 \leqslant i < q)$. Возможны два случая.

1. Ответы все запросы равны 0, тогда значение загаданной функции на всех наборах из запросов одно и то же. В силу того, что таких наборов $k+1$, а единиц функции ровно k, значит единицами функциями эти наборы не могут быть и $f(x_{i\cdot(k+1)+j}) = 0$ для всех $j \in \{0, 1, \ldots, k\}$.
2. Существует j_0 — наименьшее целое из интервала $[0, k-1]$ такое, что ответ на запрос отличен от нуля, иными словами, $f(x_{i\cdot(k+1)+j_0}) \neq f(x_{i\cdot(k+1)+j_0+1})$.
    а) Если ответ на запрос $(x_{i\cdot(k+1)+j_0}, x_{i\cdot(k+1)+j_0+1})$ равен 1, то в силу того, что ответы на все запросы $(x_{i\cdot(k+1)+j}, x_{i\cdot(k+1)+j+1}), j < j_0$, равны 0, а $f(x_{i\cdot(k+1)+j_0}) = 1$, следует $f(x_{i\cdot(k+1)+j}) = 1$ для всех $j \in \{0, 1, \ldots, j_0\}$. Более того, из ответов на оставшиеся запросы этой группы можно последовательно восстановить значение на всех наборах этой группы, то есть сначала восстановить значение на наборе $x_{i\cdot(k+1)+j_0+1}$, затем на наборе $x_{i\cdot(k+1)+j_0+2}$ и так далее. Это возможно в силу замечания 1.
    б) Если ответ на запрос $(x_{i\cdot(k+1)+j_0}, x_{i\cdot(k+1)+j_0+1})$ равен -1, то в силу того, что ответы на все запросы $(x_{i\cdot(k+1)+j}, x_{i\cdot(k+1)+j+1}), j < j_0$, равны 0, а $f(x_{i\cdot(k+1)+j_0+1}) = 1$, следует $f(x_{i\cdot(k+1)+j}) = 0$ для всех $j \in \{0, 1, \ldots, j_0\}$. Аналогично предыдущему пункту восстановим значение на всех остальных наборах этой группы.

Итого, будет задано $kq = k\cdot[2^n/(k+1)]$ запросов и восстановленно значение на всех $q \cdot (k+1)$ наборах, останется неизвестным значение на r наборах. Обозначим через x количество единиц, которое будет найдено за эти kq запросов. Тогда если $x = k$, то все единицы уже найдены. Если $x + r = k$, то оставшиеся r наборов также являются единицами и значит вновь функция полностью расшифрована. Иначе, необходимо найти $k - x$ единиц, где $0 < k - x < r$. Зададим $r-1$ запросов, где в каждом запросе первая компонента — это один из непокры-

тых r наборов, а вторая — любой из покрытых ранее. Тогда по ответам на эти запросы мы однозначно восстановим значение функции на всех наборах, кроме оставшегося непокрытого одного. Если после этих запросов найдены все единицы искомой функции, значит непокрытый набор является нулем функции, иначе, этот набор и есть оставшаяся ненайденная единица.

В общей сложности для расшифровки функции будет потрачено следующее количество запросов.

1. Если $r = 0$, то $k \cdot 2^n/(k+1)$ запросов.
2. Если $r \geqslant 1$, то $k \cdot [2^n/(k+1)] + (r-1)$ запросов.

□

Рассмотрим следующую задачу.

Дано: $M_1, M_2, \ldots, M_{u-1}, M_u$ — непустые множества наборов, среди которых равно одно состоит из единиц загаданной функции, а все остальные состоят из нулей загаданной функции.

Цель: найти множество, состоящее из единиц загаданной функции.

Алгоритм 1. *Для нахождения множества единиц среди непустых множеств $M_1, M_2, \ldots, M_{u-1}, M_u$ за $[u/2]$ запросов на сравнение выполнить следующие шаги.*

1. *В каждом из множеств $M_i, i \in [1, u]$, выбрать любой элемент m_i в качестве представителя множества.*
2. *Задать $[u/2]$ запросов вида (m_{2h+1}, m_{2h+2}), где $h = \overline{0, [u/2]-1}$.*
3. *Если u — нечетно и на все запросы получен ответ 0, то множеством единиц является M_u. Иначе, найдется запрос, на который ответ получен не 0, а значит однозначно определено множество, состоящее из единиц.*

Теорема 5. *Пусть $k \leqslant 2^{n-1}$ и для целых положительных $x_m, x_{m+1}, \ldots, x_{k-1}, x_k$, где $m =](k+1)/2[$, верно равенство*

$$2^n = m \cdot x_m + (m+1) \cdot x_{m+1} + \ldots + (k-1) \cdot x_{k-1} + k \cdot x_k.$$

Тогда справедлива следующая верхняя оценка

$$\varphi_{CQ}(F, n, k, k) \leqslant 2^n - (x_m + x_{m+1} + \ldots + x_{k-1} + x_k) + [\max(x_m, x_{m+1}, \ldots, x_{k-1}, x_k)/2].$$

Доказательство. Произвольным образом разобьем 2^n наборов на x_m множеств размера m, x_{m+1} множеств размера $m+1$, ..., x_{k-1} множеств размера $k-1$ и x_k множеств размера k. Заметим, что $k - m < m$. Обозначим все множества следующим образом $A_{i,j}$, где i указывает размер множества, то есть число от m до k, а j — номер множества размера i, то есть число от 1 до x_i. Элементы множества $A_{i,j}$ будем обозначать $\{a^h_{i,j} | h = 1, i\}$.

Для каждого множества $A_{i,j}, i = \overline{m,k}, j = \overline{1,x_i}$ зададим следующие запросы: $(a^1_{i,j}, a^2_{i,j}), (a^2_{i,j}, a^3_{i,j}), \ldots, (a^{i-2}_{i,j}, a^{i-1}_{i,j}), (a^{i-1}_{i,j}, a^i_{i,j})$. Иными словами, мы как будто сцепляем изолированные i вершин ребрами в одну цепочку. Рассмотрим ответы на эти $i-1$ запросов. Возможны два случая.

1. Ответы все запросы равны 0, тогда значение загаданной функции на всех наборах из запросов одно и то же.
2. Существует s — наименьшее целое число из интервала $[1, i-1]$ такое, что ответ на s-й запрос отличен от нуля, иными словами, $f(a^s_{i,j}) \neq f(a^{s+1}_{i,j})$.
    а) Если ответ на запрос $(a^s_{i,j}, a^{s+1}_{i,j})$ равен 1, то в силу того, что ответы на все запросы $(a^t_{i,j}, a^{t+1}_{i,j}), t < s$, равны 0, а $f(a^s_{i,j}) = 1$, следует $f(a^t_{i,j}) = 1$ для всех $t < s$. Более того, из ответов на оставшиеся запросы этой группы можно последовательно восстановить значение на всех наборах этой группы, то есть сначала восстановить значение на наборе $a^{s+1}_{i,j}$, затем на наборе $a^{s+2}_{i,j}$ и так далее. Это возможно в силу замечания 1.
    б) Если ответ на запрос $(a^s_{i,j}, a^{s+1}_{i,j})$ равен -1, то в силу того, что ответы на все запросы $(a^t_{i,j}, a^{t+1}_{i,j}), t < s$, равны 0, а $f(a^s_{i,j}) = 0$, следует $f(a^t_{i,j}) = 0$ для всех $t < s$. Аналогично предыдущему пункту восстановим значение на всех остальных наборах этой группы.

Иными словами, в результате этих запросов про каждое множество $A_{i,j}$ мы знаем либо значение функции на всех наборах этого множества, либо то, что на всех наборах множества функция принимает одинаковое значение.

Подытоживая, получаем, что было задано $(m-1) \cdot x_m + m \cdot x_{m+1} + \ldots + (k-2) \cdot x_{k-1} + (k-1) \cdot x_k$ запросов, все 2^n наборов покрыты. Причем про какие-то наборы уже известно, чему равно значение функции в них, а оставшиеся наборы распались на классы эквивалентности мощности не меньшей m. Обозначим через t количество единиц функции, найденных в результате этих за-

просов. Если $t = k$, то искомая функция расшифрована. Иначе, осталось найти $u = k - t > 0$ единиц и точно известно, что все они лежат ровно в одном из образовавшихся множеств — классов эквивалентности. Поскольку в случае, если бы все ненайденные единицы находились хотя бы в двух множествах A_{i_1,j_1}, A_{i_2,j_2}, тогда получилось бы, что $u = k - t \geqslant i_1 + i_2 > k$.

Следовательно, если после получения ответов на упомянутые выше запросы, не были найдены все единицы, то остается найти $u \in [m, k]$ единиц, которые находятся ровно в одном из множеств $A_{u,1}, A_{u,2}, \ldots, A_{u,x_u-1}, A_{u,x_u}$. Причем, про какие-то из этих множеств мы уже узнали значение функции на них. Не нарушая общности, будем считать, что про последние $(x_u - p) \in [0, x_u - 1]$ множеств $A_{u,p+1}, A_{u,p+2}, \ldots, A_{u,x_u}$ мы знаем значение функции на каждом его элементе. Соответственно, необходимо найти единицу среди множеств $A_{u,1}, A_{u,2}, \ldots, A_{u,p}$. А это осуществимо за $[p/2]$ запросов на сравнение, если воспользоваться алгоритмом 1.

Итого, в худшем случае будет задано $(m-1) \cdot x_m + m \cdot x_{m+1} + \ldots + (k-2) \cdot x_{k-1} + (k-1) \cdot x_k + [x_u/2]$ запросов. Из равенства $2^n = m \cdot x_m + (m+1) \cdot x_{m+1} + \ldots + (k-1) \cdot x_{k-1} + k \cdot x_k$ получаем равенство

$$(m-1) \cdot x_m + m \cdot x_{m+1} + \ldots + (k-2) \cdot x_{k-1} + (k-1) \cdot x_k = \\ = 2^n - (x_m + x_{m+1} + \ldots + x_k).$$

Это и приводит нас к оценке доказываемой теоремы. □

В частности, благодаря подстановке определенных значений $x_m, x_{m+1}, \ldots, x_{k-1}, x_k$ в теорему 5, получаем следующее следствие.

Следствие 1. *Пусть* $3 \leqslant k \leqslant 2^{n-1}, m =](k+1)/2[, s = m + (m+1) + \ldots + (k-1) + k$, *верно равенство* $2^n = s \cdot q + r, r \in [0, s), q \geqslant m$, q, r — *целые положительные числа. Тогда справедлива следующая верхняя оценка*

$$\varphi_{CQ}(F, n, k, k) \leqslant 2^n - (k-m+1)q - c + \Big[0.5 \cdot \max(q - r + (m+1)c, q + r - mc, q)\Big] \\ \leqslant 2^n - k/2 \cdot [2^n/s] + \Big[0.5 \cdot \big([2^n/s] + \big](k+1)/2\big[+ k^2\big)\Big],$$

где c *вычисляется следующим образом*

- $c = [2r/(2m+1)]$ *при* $2r \bmod (2m+1) \leqslant m$,
- $c = [2r/(2m+1)] + 1$ *при* $2r \bmod (2m+1) > m$.

Доказательство. Положим $x_m = q - r + (m+1) \cdot c$, $x_{m+1} = q + r - m \cdot c$, а $x_{m+2} = x_{m+3} = \ldots = x_k = q$. Обозначим через $p = 2r \bmod (2m+1)$. Определим c следующим образом.

– при $p \leqslant m$ положим равным $c = [2r/(2m+1)]$,

– при $p > m$ положим равным $c = [2r/(2m+1)] + 1$.

Заметим, что при $p \leqslant m$ выполнено $(m+1) \cdot c = (m+1) \cdot [2r/(2m+1)] \geqslant (m+1) \cdot [2r/(2m+2)] \geqslant (m+1) \cdot [r/(m+1)] = r - (r \bmod (m+1))$, а при $p > m$ выполнено $(m+1) \cdot c = (m+1) \cdot ([2r/(2m+1)] + 1) \geqslant (m+1) \cdot ([2r/(2m+2)] + 1) \geqslant (m+1) + (r - (r \bmod (m+1)))$. Поэтому при $p \leqslant m$ выполнено $x_m = q - r + (m+1) \cdot c \geqslant q - r + r - (r \bmod (m+1)) \geqslant 0$, при $p > m$ выполнено $x_m = q - r + (m+1) \cdot c \geqslant q - r + (m+1) + (r - (r \bmod (m+1))) \geqslant 0$.

Также заметим, что при $p \leqslant m$ выполнено $m \cdot c = m \cdot [2r/(2m+1)] \leqslant m \cdot 2r/(2m+1) \leqslant m \cdot 2r/(2m) \leqslant r$, а при $p > m$ выполнено $m \cdot c = m \cdot ([2r/(2m+1)] + 1) \leqslant m \cdot 2r/(2m+1) + m \leqslant m \cdot 2r/(2m) + m \leqslant r + m$. Поэтому при $p \leqslant m$ выполнено $x_{m+1} = q + r - mc \geqslant q + r - r \geqslant 0$, при $p > m$ выполнено $x_{m+1} = q + r - mc \geqslant q + r - r - m \geqslant q - m \geqslant 0$.

Подставляя полученные значения x_i в верхнюю оценку теоремы 5, получаем первое неравенство следствия.

Важно отметить, что $|x_m - x_{m+1}| \leqslant m$. При $p \leqslant m$ выполнено $x_{m+1} - x_m = 2r - c(2m+1) = p \leqslant m$. При $p > m$ верна цепочка равенств $x_m - x_{m+1} = c(2m+1) - 2r = 2m + 1 - p \leqslant m$. В силу этого при $x_m < x_{m+1}$ имеем $\max(x_m, x_{m+1}) \leqslant q + r \leqslant q + s \leqslant q + k^2$, при $x_m \geqslant x_{m+1}$ получаем $\max(x_m, x_{m+1}) \leqslant q + r + m \leqslant q + s + m \leqslant q + k^2 + m$.

Теперь оценим точнее параметры первого неравенства следствия. Из формулировки следствия получаем следующие соотношения: $q = [2^n/s]$, $k - m + 1 \geqslant k/2$. В свою очередь, $\max(x_m, x_{m+1}, q) \leqslant q + m + k^2 = [2^n/s] +](k+1)/2[+ k^2$. Все это позволяет нам получить второе неравенство из утверждения следствия. □

Тривиальной верхней оценкой для рассматриваемой задачи является $2^n - 1$, поскольку задав следующией запросы $(x_1, x_2), (x_2, x_3), (x_3, x_4), \ldots, (x_{2^n-1}, x_{2^n})$, где $x_1, x_2, \ldots, x_{2^n}$ — все 2^n бинарных n-местных наборов, можно восстановить значение функции согласно замечанию 1.

Оценим разные комбинации параметров n, k для того, чтобы понять, какой член в оценке следствия 1 главный. Более того, сравним при каких n, k получаемые верхние оценки строго лучше тривиальной верхней оценки $2^n - 1$.

Следствие 2. *При $2^n > k$ справедлива следующая верхняя оценка*

$$\varphi_{CQ}(F,n,k,k) \leqslant 2^n\big(1-\frac{2k}{3(k+1)^2}+\frac{4}{3k^2}\big)+\frac{k^2}{2}+\frac{3k}{4}+1.$$

Причем, при $k \geqslant 5$ эта величина строго меньше 2^n-1.

Доказательство. Предварительно заметим справедливость следующей цепочки неравенств $s = m+(m+1)+\ldots+(k-1)+k = (m+k)(k-m+1)/2 \geqslant (k/2+k)k/4 \geqslant 3/8\cdot k^2$. С другой стороны, $s = (m+k)(k-m+1)/2 \leqslant (k/2+1+k)(k-k/2+1)/2 \leqslant (3k+2)(k+2)/8 < (3k+3)(2k+2)/8 \leqslant 6(k+1)^2/8 = 3/4\cdot(k+1)^2$. Поэтому второе неравенство следствия 1 превратится в неравенство

$$\begin{aligned}
\varphi_{CQ}(F,n,k,k) &\leqslant 2^n - k/2\cdot[2^n/s] + \Big[0.5\cdot\big([2^n/s]+\big](k+1)/2\big[+k^2\big)\Big] \\
&\leqslant 2^n - k/2\cdot\big[4/3\cdot 2^n/(k+1)^2\big] + \Big[0.5\cdot\big(8/3\cdot 2^n/k^2+k/2+1+k^2\big)\Big] \\
&\leqslant 2^n - k/2\cdot\big(4/3\cdot 2^n/(k+1)^2-1\big) + \Big[0.5\cdot\big(8/3\cdot 2^n/k^2+k/2+1+k^2\big)\Big] \\
&\leqslant 2^n - (2/3\cdot 2^n k/(k+1)^2 - k/2) + 4/3\cdot 2^n/k^2 + k/4 + 1 + k^2/2 \\
&\leqslant 2^n\big(1-\frac{2k}{3(k+1)^2}+\frac{4}{3k^2}\big)+\frac{k^2}{2}+\frac{3k}{4}+1.
\end{aligned}$$

Рассмотрим два случая: $2^n \leqslant k^4$ и $2^n > k^4$.

Подстановкой $2^n \leqslant k^4$, получаем следующую цепочку неравенств

$$\begin{aligned}
\varphi_{CQ}(F,n,k,k) &\leqslant 2^n\big(1-\frac{2k}{3(k+1)^2}+\frac{4}{3k^2}\big)+\frac{k^2}{2}+\frac{3k}{4}+1 \\
&\leqslant k^4\big(1-\frac{2k}{3(k+1)^2}+\frac{4}{3k^2}\big)+\frac{k^2}{2}+\frac{3k}{4}+1.
\end{aligned}$$

Определим, при каких k последнее выражение строго меньше тривиальной верхней оценки 2^n-1, иными словами необходимо решить неравенство $k^4\big(1-\frac{2k}{3(k+1)^2}+\frac{4}{3k^2}\big)+\frac{k^2}{2}+\frac{3k}{4}+1 < 2^n-1 < k^4$, иными словами необходимо решить неравенство $k^4\big(\frac{2k}{3(k+1)^2}-\frac{4}{3k^2}\big)-\frac{k^2}{2}-\frac{3k}{4}-1>0$, или равносильное ему $\frac{2k^5}{3(k+1)^2}-\frac{11k^2}{6}-\frac{3k}{4}-1>0$.

Поскольку при $k \geqslant 5$ верно $\frac{2k^5}{3(k+1)^2} \geqslant \frac{2k^5}{3(k+0.2k)^2} = \frac{25k^3}{54}$, что достаточно показать, что $\frac{25k^3}{54}-\frac{11k^2}{6}-\frac{3k}{4}-1>0$.

График функции $y(k) = \frac{25k^3}{54}-\frac{11k^2}{6}-\frac{3k}{4}-1$ представлен на рисунке 2.1. Покажем, что функция возрастает при $k \geqslant 5$ и строго положительна при $k=5$.

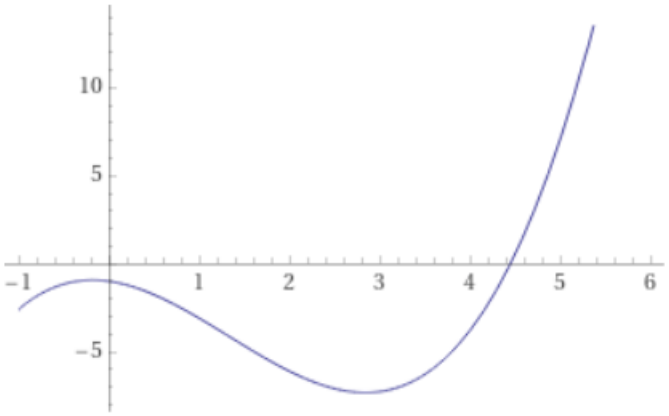

Рисунок 2.1 — График функции $y(k) = \frac{25k^3}{54} - \frac{11k^2}{6} - \frac{3k}{4} - 1$.

При $k = 5$ значение функции равно $\frac{25 \cdot 5^3}{54} - \frac{11 \cdot 25}{6} - \frac{15}{4} - 1 = \frac{25 \cdot 5^3 \cdot 2 - 11 \cdot 25 \cdot 18 - 15 \cdot 27 - 108}{108} = \frac{787}{108} > 0$. Покажем, что производная функции $y(k)$ неотрицательна при $k \geqslant 5$.

$$y'(k) = \frac{25k^2}{18} - \frac{11k}{3} - \frac{3}{4} = \frac{50k^2 - 132k - 27}{36}$$

При положительных k выражение $50k^2 - 132k - 27$ неотрицательно при $k \geqslant 3$.

Следовательно, при таких значениях k полученная в предыдущем следствии верхняя оценка строго меньше $2^n - 1$.

Теперь рассмотрим случай $2^n > k^4$. Найдем значения k, при которых полученная верхняя оценка строго меньше $2^n - 1$, для этого необходимо решить неравенство

$$2^n\big(1 - \frac{2k}{3(k+1)^2} + \frac{4}{3k^2}\big) + \frac{k^2}{2} + \frac{3k}{4} + 1 < 2^n - 1,$$

или равносильное ему

$$2^n\big(\frac{2k}{3(k+1)^2} - \frac{4}{3k^2}\big) - \frac{2k^2 + 3k + 8}{4} > 0.$$

Вспоминая, что $2^n > k^4$, остается показать, при каких значениях k верно неравенство

$$k^4\big(\frac{2k}{3(k+1)^2} - \frac{4}{3k^2}\big) - \frac{2k^2 + 3k + 8}{4} > 0,$$

тогда при этих же значениях k будет верно и неравенство

$$2^n\big(\frac{2k}{3(k+1)^2} - \frac{4}{3k^2}\big) - \frac{2k^2 + 3k + 8}{4} > 0.$$

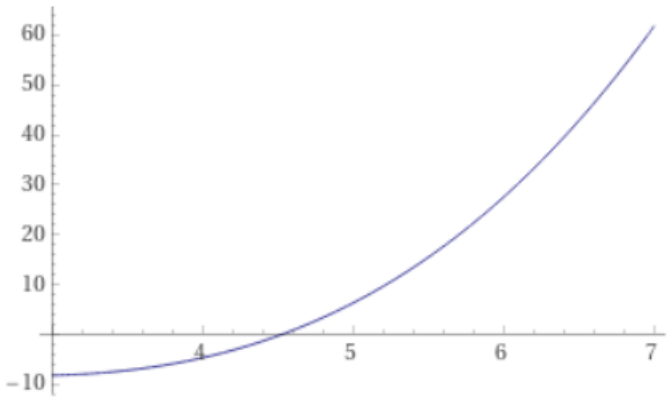

Рисунок 2.2 — График функции $g(k) = \frac{50k^3-198k^2-81k-216}{108}$.

При $k \geqslant 5$ верно неравенство

$$k^4 \frac{2k}{3(k+1)^2} \geqslant k^4 \frac{2k}{3(k+0.2k)^2} = \frac{25k^3}{54}.$$

Поэтому осталось показать, что при $k \geqslant 5$ всегда неотрицательно выражение

$$\frac{25k^3}{54} - \frac{4k^2}{3} - \frac{2k^2+3k+8}{4} = \frac{50k^3-144k^2-54k^2-81k-216}{108} = \frac{50k^3-198k^2-81k-216}{108}$$

График функции $g(k) = \frac{50k^3-198k^2-81k-216}{108}$ представлен на рисунке 2.2.

Аналогично первой половине доказательства, покажем возрастание функции $g(k) = \frac{50k^3-198k^2-81k-216}{108}$ при $k \geqslant 5$ и положительность при $k = 5$.

$$g(5) = \frac{50 \cdot 5^3 - 198 \cdot 5^2 - 81 \cdot 5 - 216}{108} = \frac{679}{108} > 0.$$

$$g'(k) = \frac{150k^2 - 396k - 81}{108}.$$

Производная $g'(k)$ неотрицательна при $k \geqslant 3$, а значит мы доказали, что при $k \geqslant 5$ полученная в предыдущем следствии верхняя оценка строго меньше $2^n - 1$.

□

2.3.3 Точная оценка сложности расшифровки для функций веса 1

Для функций веса 1 справедлива следующая теорема о сложности расшифровки запросами на сравнение.

Теорема 6. *Сложность расшифровки класса* $F(n,1,1)$ *запросами на сравнение равна* $\varphi_{CQ}(F,n,1,1) = 2^{n-1}$.

Доказательство. Нижняя оценка. Пусть ученик задаст $2^{n-1}-1$ запросов, на каждый ответим 0. В итоге, суммарно будет опрошено не более $(2^{n-1}-1)\cdot 2 = 2^n - 2$ наборов. Про каждый из них ученик поймет, что значение функции на нем равно 0. Остается как минимум два набора, на которых неизвестно значение функции, следовательно нужно задать хотя бы еще один запрос.

Верхняя оценка. Зададим такие запросы на сравнение, чтобы 2^n наборов распались на компоненты связности размера на менее 2. В силу леммы 3 это можно сделать, если задать 2^{n-1} запросов на сравнение. □

2.3.4 Точная оценка сложности расшифровки для функций веса 2

Перейдем к доказательству теоремы о сложности расшифровки функций веса 2. Неформально говоря, оказывается, для расшифровки функций веса выгодно разбить 2^n наборов на группы размера 3, для каждой группы из наборов a, b, c задать два запроса на сравнение (a,b), (b,c), по ответам на эти запросы можно однозначно восстановить загаданную функцию.

Теорема 7. *При* $n \geqslant 2$ *сложность расшифровки класса* $F(n,2,2)$ *запросами на сравнение равна* $\varphi_{CQ}(F,n,2,2) = [2^{n+1}/3]$.

Доказательство. Докажем следующую нижнюю оценку $\varphi_{CQ}(F,n,2,2) \geqslant 2[2^n/3] + (2^n \bmod 3) - 1$. Возможны два случая: $2^n = 3x+1$ и $2^n = 3x+2$, где x — целое.

Рассмотрим первый случай: $2^n = 3x+1$. Необходимо доказать, что ученик задаст хотя бы $2x$ запросов. Пусть он задаст $2x-1$ запросов, на каждый

запрос ответим 0, тогда покажем, что либо ему недостаточно информации, полученной по этим запросам, чтобы восстановить загаданную функцию, либо по заданным запросам можно восстановить порядок объединения вершин графа в компоненты связности размера не менее 3 за меньшее число запросов, чем утверждается в лемме 3. Итак, ученик задал $2x - 1$ запросов. Рассмотрим все возможные случаи.

1. $2x - 1$ запросов покрывают не более $3x$ наборов. Обозначим через a_0 один из непокрытых наборов. Согласно лемме 3, после не менее $2x$ запросов $3x$ наборов могут объединиться в x компонент связности размера 3. Но раз задано на один запрос меньше, то в лучшем случае сформировано уже $x - 1$ множеств мощности 3, а для образования еще одного такого не хватает одного запроса. Следовательно, среди всех наборов, за исключением набора a_0, имеется еще
   а) либо три непокрытых набора a_1, a_2, a_3,
   б) либо один непокрытый a_1 и одна компонента связности, состоящяя из наборов a_2, a_3,
   в) либо две компоненты связности размера 2.

   В первом случае 2 единицы функции могут находиться в любых двух наборах из a_0, a_1, a_2, a_3. Во втором случае 2 единицы либо a_2, a_3, либо a_0, a_1. В третьем случае, обе единицы могут находиться в любом из этих двух двухэлементных множеств.

   В любом случае, ученик вынужден задать еще хотя бы один запрос, чтобы избавиться от возникшей неоднозначности.
2. $2x - 1$ запросов покрывают все $2^n = 3x + 1$ наборов. Поскольку отсутствуют непокрытые наборы и ответы на все запросы равны 0, то все наборы разбились на компоненты связности размера хотя бы 2. Возможны следующие три случая.
   а) Имеются как минимум две компоненты связности размера 2. Тогда учитель может спрятать обе единицы в одну из них, поэтому ученик вынужден задать еще хотя бы один запрос.
   б) Имеются ровна одна компонента связности размера 2. Иными словами, после $2x - 1$ операций объединения исходные $3x + 1$ вершины объединились в компоненты связности, среди которых ровно одна размера 2, а остальные размера не менее 3. Но этот случай невозможен, так как противоречит лемме 3,

потому что один запрос из $2x-1$ потрачен на образование компоненты связности размера 2, а остальные $2x-2$ потрачены на объединение $(3x+1)-2=3(x-1)+2$ вершин в компоненты связности размера не менее 3, а в лемме 3 утверждается, что для этого необходимо $2x$ запросов.

в) Все компоненты связности имеют размер строго больше 2. Этот случай невозможен, так как в этом случае все $3x+1$ вершин за $2x-1$ операций объединились в компоненты связности размера не менее 3, что противоречит лемме 3.

Следовательно, в случае $2^n=3x+1$ ученик вынужден задать как минимум $2x$ запросов.

Рассмотрим случай $2^n=3x+2$. Необходимо доказать, что ученик вынужден задать хотя бы $2x+1$ запросов.

Пусть ученик задал $2x$ запросов. На первые $2x-1$ запросов отвечаем 0.

Если $2x$ запросов покрывают ровно $3x$ наборов и к запросу с номером $2x$ в точности $3x-3$ наборов объединились в компоненты связности размера 3, а запрос с номером $2x$ объединит еще три вершины в компоненту связности размера 3, то есть один набор этого запроса — изолированная вершина, второй — вершина компоненты связности размера 2, тогда ответим -1, если первый набор запроса соответствует вершине компоненты связности размера 2, и ответим 1, если второй набор запроса соответствует вершине компоненты связности размера 2.

Иначе, на запрос с номером $2x$ ответим 0.

Рассмотрим все возможные случаи.

1. $2x$ запросов покрыли не более $3x-1$ наборов. Тогда хотя бы 3 набора не покрыты. Ответы на все запросы были равны 0. Учитель может спрятать обе единицы в двух из трех непокрытых наборах. Поэтому ученик вынужден задать хотя бы еще один запрос.
2. $2x$ запросов покрыли ровно $3x$ наборов. Тогда возможна одна из следующих двух ситуаций.
   а) Ответ на запрос с номером $2x$ был отличен от 0. Следовательно, учитель раскрыл информацию о ровно одной единице и двух нулях функции. Осталось найти еще одну единицу. В силу того, что ответы на первые $2x-1$ запросов были равны 0, то все остальные наборы из рассматриваемых $3x$ объеди-

нились в компоненты связности размера не меньше 2. В этих компонентах связности вторая единиица не может находиться, значит она находится среди непокрытых двух наборов. Следовательно, ученик вынужден задать еще один запрос для однозначного восстановления функции.

б) Ответ на запрос с номером $2x$ был равен 0. Следовательно, рассматриваемые $3x$ наборов не объединились в x компонент связности размера 3. Возможна одна из следующих ситуаций.

1) Среди компонент связности, в которые объединились $3x$ наборов, есть компонента размера 2. Тогда учитель может спрятать обе единицы либо в два непокрытых набора, либо в эту компоненту связности размера 2. Ученик вынужден задать еще один запрос для восстановления функции.

2) Среди компонент связности, в которые объединились $3x$ наборов, нет компоненты размера 2. Но не все компоненты связности размера ровно 3. Следовательно, $3x$ наборов объединились в p компонент связности размера не менее 3, причем существует хотя бы одна компонента размера строго больше 3. На образование p компонент связности размера ровно 3 необходимо не менее $2p$ запросов, и суммарно останется $3x - 3p > 0$ наборов распределить по этим компонентам связности. На добавление каждого такого набора в какую-то из p компонент связности тратится не менее 1 запроса. Исходя из этого, получаем цепочку неравенств $2x \geqslant 2p + (3x - 3p)$, $p \geqslant x$, $3p \geqslant 3x$, что противоречит неравенству $3x - 3p > 0$.

3. $2x$ запросов покрыли хотя бы $3x + 1$ наборов. Тогда возможна одна из следующих трех ситуаций.

а) Не менее $3x + 1$ наборов объединились в компоненты связности размера не менее 3. Этот случай невозможен, так как противоречит лемме 3.

б) Среди компонент связности, в которые объединились не менее $3x + 1$ наборов, имеются хотя бы две размера 2. Тогда обе

единицы можно спрятать в любой из них и ученик вынужден использовать еще один дополнительный запрос для понимания, какая из этих компонент содержит обе единицы.

в) Среди компонент связности, в которые объединились не менее $3x+1$ наборов, имеется ровно одна размера 2. Для рассмотрения этой ситуации отдельно рассмотрим случай, когда $2x$ запросов покрыли $3x+1$ наборов и когда $2x$ запросов покрыли $3x+2$ набора.

- $2x$ запросов покрыли $3x+1$ наборов. Соответственно, $3x-1 = 3(x-1)+2$ наборов за $2x-1 = 2(x-1)+1$ запросов объединились в компоненты связности размера не менее 3, что противоречит лемме 3. Значит, этот случай невозможен.
- $2x$ запросов покрыли $3x+2$ набора. Получается, что $3x$ наборов за $2x-1$ запросов объединились в компоненты связности размера не менее 3, что противоречит лемме 3. Следовательно, и этот случай невозможен.

Исходя из рассмотренных случаев становится ясно, что в случае $2^n = 3x+2$ ученик вынужден задать как минимум $2x+1$ запросов.

Учитывая нижнюю оценку, полученную в доказательстве данной теоремы, и верхнюю оценку, полученную в лемме 12 при подстановке $k=3$, имеем следующее равенство $\varphi_{CQ}(F,n,2,2) = 2[2^n/3] + (2^n \bmod 3) - 1$.

Покажем, что последняя величина равна $[2^{n+1}/3]$. Для этого рассмотрим два случая.

1. Если $2^n = 3x+1$, тогда $\varphi_{CQ}(F,n,2,2) = 2x$, но с другой стороны $[2^{n+1}/3] = [2(3x+1)/3] = [2x+2/3] = 2x$.
2. Если $2^n = 3x+2$, тогда $\varphi_{CQ}(F,n,2,2) = 2x+1$, но с другой стороны $[2^{n+1}/3] = [2(3x+2)/3] = [2x+4/3] = 2x+1$.

□

2.3.5 Точная оценка сложности расшифровки для функций веса 3

В данном разделе доказывается, что для расшифровки функций арности n и веса 3 необходимо задать в точности $2^n - \left]3/2 \cdot [2^n/5]\right[- \left[(2^n \bmod 5)/2\right]$ запросов на сравнение (теорема 8). Опишем кратко схему доказательства верхней оценки.

1. Сначала доказывается лемма 13 о том, что в случае функций веса 3 оценку теоремы 5, равную $2^n - (x_2 + x_3) + [\max(x_2, x_3)/2]$, можно заменить на $2^n - (x_2 + x_3) + \left[\max(x_2 - 1, x_3)/2\right]$ при целых положительных x_2, x_3.
2. Далее ищутся подходящие значения x_2, x_3, чтобы максимизировать значение $(x_2 + x_3) - \left[\max(x_2 - 1, x_3)/2\right]$. Для этого показывается, что максимум функции $M(x_2, x_3, \ldots, x_{2^n}) = (x_2 + x_3 + \ldots + x_{2^n}) - \left[\max(x_2 - 1, x_3)/2\right]$ при $2x_2 + 3x_3 + \ldots + 2^n x_{2^n} = 2^n, x_2 + x_3 > 1$ с целыми неотрицательными $x_2, x_3, \ldots, x_{2^n}$ достигается при $x_4 = x_5 = \ldots = x_{2^n} = 0$ (леммы 14 и 15), то есть максимум "расширенной функции" M совпадает с тем, который мы ищем, и $x_2 \geqslant x_3$ (леммы 16 и 17).
3. Далее в лемме 18 определяются значения x_2, x_3, который доставляют нужный максимум $\left]3/2 \cdot [2^n/5]\right[+ \left[(2^n \bmod 5)/2\right]$ функции M.

Теперь кратко изложим схему доказательства нижней оценки.

1. Сначала в лемме 20 показывается, что если хотя бы один из 2^n наборов не будет покрыт запросами ученика, то для расшифровки функции ученику понадобится задать не менее $3 \cdot 2^{n-2} - 2$ запросов.
2. Далее в лемме 11 демонстрируется, что верно неравенство

$$2^n - 1.5 \cdot q - [0.5 \cdot r] < 3 \cdot 2^{n-2} - 2$$

для q, r таких, что $2^n = 5 \cdot q + r, r \in [1, 4]$. Но указанная в лемме 19 верхняя оценка $2^n - \left]3/2 \cdot [2^n/5]\right[- \left[(2^n \bmod 5)/2\right]$ не меньше $2^n - 1.5 \cdot q - [0.5 \cdot r]$. Иными словами, если ученик использует для расшифровки алгоритм, покрывающий не все 2^n наборов, тогда он вынужден задать строго больше запросов, чем в алгоритме, приведенном при доказательстве верхней оценки в лемме 19. Поэтому для минимизации числа запросов ученик будет использовать алгоритм, покрывающий все наборы.

3. Далее в лемме 21 доказывается, что если после ответов на все запросы ученика компонент связности размера 2 и 3 образуются не более одной, то было задано не менее $3 \cdot 2^{n-2} - 2$ запросов. А это, как было сказано выше, является неоптимальной стратегией действий ученика. Поэтому ученик точно должен использовать алгоритм, в котором число образованных компонент связности размера 2 и 3 не менее двух.
4. Затем приводится стратегия, как учитель должен отвечать на запросы ученика, чтобы все три единицы функции оказались только в компонентах связности размера 2 или 3, и показывается, что число запросов, заданных учеником, будет равно значению

$$2^n - (x_2 + x_3 + \ldots + x_{2^n}) - \big[\max(x_2 - 1, x_3)/2\big].$$

5. Далее используется результат леммы 18 о максимуме последней указанной функции.

Теперь перейдем к формальному доказательству упомянутых лемм.

Лемма 13. *Пусть $n \geqslant 4$ и для целых положительных x_2, x_3 верно равенство $2^n = 2x_2 + 3x_3$. Тогда справедлива следующая верхняя оценка*

$$\varphi_{CQ}(F, n, 3, 3) \leqslant 2^n - (x_2 + x_3) + [\max(x_2 - 1, x_3)/2].$$

Доказательство. Зададим $2x_3 + x_2 = 2^n - x_2 - x_3$ запросов, образовав x_3 компонент связности размера 3 и x_2 компонент размера 2 соответственно. Причем, сначала будем задавать запросы, относящиеся к компонентам размера 3, а лишь затем к компонентам размера 2.

При этом заметим что, если ответ на каждый из первых $(2^n - x_2 - x_3)$ запросов был 0, тогда все единицы находятся в одной из x_3 компонент связности размера 3 и для нахождения нужной компоненты надо дополнительно задать $[x_3/2]$ запросов, если воспользоваться алгоритмом 1, то есть суммарно будет задано $2^n - (x_2 + x_3) + [x_3/2]$ запросов. Если хоть раз был получен ответ отличный от 0, тогда возможны следующие случаи.

– Если первый ответ отличный от 0 был получен на запросах, относящихся к формированию компонент размера 2, тогда все компоненты связности размера 3 уже сформированы, в них точно не находятся единицы, а значит остальные единицы функции будут находиться в одной из $x_2 - 1$ компонент размера 2. Соответственно, суммарно придется потратить не более $2^n - (x_2 + x_3) + [(x_2 - 1)/2]$ запросов.

– Если первый ответ отличный от 0 был получен на запросах, относящихся к формированию компонент связности размера 3, тогда возможно три ситуации.

1. Запрос объединял две изолированные вершины. Соответственно, найдена только одна единица, и сформирована компонента размера 2. Будем считать, что мы отошли на один запрос от нашего плана сначала только формировать компоненты связности размера 3, а затем размера 2. Иными словами, мы с опережением сформировали одну компоненту связности размера 2 и про нее уже все знаем. Также нам известно, что останется найти еще две единицы. Поэтому если мы запросим оставшиеся запросы, чтобы сформировать в итоге x_2 компонент размера 2 и x_3 компонент размера 3, и всегда в ответ получим 0, то две единицы будут содержаться в одной из x_2-1 компонент размера 2. В этом случае, суммарно будет задано $2^n-(x_2+x_3)+[(x_2-1)/2]$ запроса. Если в процессе формирования этих компонент мы еще раз получим ответ отличный от 0, то мы за $2^n-(x_2+x_3)$ запросов найдем все единицы, дополнительные запросы не понадобятся.
2. Запрос объединял одну- и двухвершинную компоненты связности. По ответу на запрос мы однозначно поймем, какая из этих компонент состоит из единиц функции. Если они в компоненте размера 2, тогда останется найти оставшуюся единицу и она раскроется в течение $2^n-(x_2+x_3)$ запросов при формировании x_2 компонент размера 2 и x_3 компонент размера 3, в силу этого дополнительные запросы не пригодятся и суммарно будет задано всего лишь $2^n-(x_2+x_3)$ запросов.

   Если единица — это изолированная вершина, тогда очевидно, что осталось найти две единицы. Доопросим все запросы, необходимые для формирования x_3 компонент связности размера 3, и затем вместо формирования x_2 компонент связности размера 2 будем формировать компоненты связности размера 3. Суммарно будет задано $2x_3$ запросов для формирования x_3 компонент связности размера 3. Затем будет задано $2[(2^n-3x_3)/3]+((2^n-3x_3) \bmod 3)-1 = 2[2^n/3]-2x_3+(2^n \bmod 3)-1$

запросов, то есть оставшиеся $2^n - 3x_3$ наборов при помощи запросов объединим в $[(2^n - 3x_3)/3]$ компонент связности размера 3 и одну компоненту размера $\Big((2^n - 3x_3) \bmod 3\Big)$. То есть будет создано много компонент связности размера 3 и возможно одна компонента размера 1-2, поскольку осталось найти только две единицы функции, они однозначно будут определены по ответам на эти запросы. Значит, суммарно было задано $2x_3 + 2[2^n/3] - 2x_3 + (2^n \bmod 3) - 1 = 2[2^n/3] + (2^n \bmod 3) - 1$ запросов.

Соответственно, в случае, когда запрос с ответом отличном от 0 относился к формированию компонент связности размера 3, ученик будет вынужден задать не более $\max(2[2^n/3] + (2^n \bmod 3) - 1, 2^n - (x_2 + x_3) + [(x_2 - 1)/2])$ запросов.

Определим условия на x_2, x_3, при которых этот максимум равен $2^n - (x_2 + x_3) + [(x_2 - 1)/2]$. Тогда в целом при таком алгоритме расшифровки с учетом оценки $2^n - (x_2 + x_3) + [x_3/2]$ случая, описанного выше, понадобится не более $2^n - (x_2 + x_3) + [\max(x_2 - 1, x_3)]/2$ запросов, что и докажет оценку данной леммы.

Перейдем доказательству неравенства $2[2^n/3] + (2^n \bmod 3) - 1 \leqslant 2^n - (x_2 + x_3) + [(x_2 - 1)/2] = x_2 + 2x_3 + [(x_2 - 1)/2]$. Если из обеих частей отбросим общие $2x_3$ запросы, останется лишь доказать неравенство $2[(2^n - 3x_3)/3] + ((2^n - 3x_3) \bmod 3) - 1 \leqslant x_2 + [(x_2 - 1)/2]$. В силу леммы 9, это неравенство имеет место при $x_2 \geqslant 4$.

Следовательно можно заключить, что необходимо задать в худшем случае $\max(2[2^n/3] + (2^n \bmod 3) - 1, 2^n - (x_2 + x_3) + [\max(x_2 - 1, x_3)/2])$. При этом, при $x_2 \geqslant 4$ число запросов равно $2^n - (x_2 + x_3) + [\max(x_2 - 1, x_3)/2]$, как и утверждается в данной лемме.

При $x_2 < 4$ число запросов равно $\max(2[2^n/3] + (2^n \bmod 3) - 1, 2^n - (x_2 + x_3) + [x_3/2])$. Поскольку $n \geqslant 4$, то $(2^n - 2x_2) = 3x_3 \geqslant 10, x_3 \geqslant 4$, то есть $\max(x_3, x_2 - 1) = x_3$. Осталось показать, что $2[2^n/3] + (2^n \bmod 3) - 1 \leqslant 2^n - (x_2 + x_3) + [x_3/2]$, а это было сделано в лемме 10.

Следовательно, оценка данной леммы доказана.

□

Будем говорить, что $x_2, x_3, \ldots, x_{2^n}$ *удовлетворяют условию* $\mu(n)$, если $x_2, x_3, \ldots, x_{2^n}$ — целые неотрицательные числа, а также выполнено

$$2x_2 + 3x_3 + 4x_4 + \ldots + 2^n x_{2^n} = 2^n, x_2 + x_3 > 1.$$

Рассмотрим следующую задачу. Пусть n — целое число, не меньшее 4. При условии, что $x_2, x_3, \ldots, x_{2^n}$ удовлетворяют условию $\mu(n)$, требуется максимизировать функцию

$$M(x_2, x_3, \ldots, x_{2^n}) = \begin{cases} x_2 + x_3 + x_4 + \ldots + x_{2^n} - [x_3/2], & \text{если } x_2 \leqslant x_3, \\ x_2 + x_3 + x_4 + \ldots + x_{2^n} - [(x_2 - 1)/2], & \text{если } x_2 > x_3. \end{cases}$$

Эту функцию можно переписать в следующем виде

$$M(x_2, x_3, \ldots, x_{2^n}) = x_2 + x_3 + x_4 + \ldots + x_{2^n} - [\max(x_2 - 1, x_3)/2].$$

В доказательстве леммы 9 работы [43] имеются неточности, которые не приводят к изменению результатов. Поэтому далее приведем уточненное доказательство.

Лемма 14. *Пусть* $a_2, a_3, \ldots, a_{2^n}$ *удовлетворяют условию* $\mu(n)$, *помимо этого хотя бы для одного* $t \in [4, 2^n]$ *верно* $a_t > 0$. *Тогда существуют* $b_2, b_3, \ldots, b_{2^n}$, *такие что они удовлетворяют условию* $\mu(n)$ *и* $b_4 = a_4, \ldots, b_{t-1} = a_{t-1}, b_t = a_t - 1, b_{t+1} = a_{t+1}, \ldots, b_{2^n} = a_{2^n}$, *и для которых справедливо неравенство* $M(a_2, a_3, \ldots, a_{2^n}) \leqslant M(b_2, b_3, \ldots, b_{2^n})$.

Доказательство. Рассмотрим два случая в зависимости от четности t.

1. Пусть $t = 2 \cdot p$, p — целое. Тогда положим $b_2 = a_2 + p, b_3 = a_3, b_4 = a_4, \ldots, b_{t-1} = a_{t-1}, b_t = a_t - 1, b_{t+1} = a_{t+1}, \ldots, b_{2^n} = a_{2^n}$. Заметим, что $b_2, b_3, \ldots, b_{2^n}$ удовлетворяют $\mu(n)$ условию. Рассмотрим значение $M(b_2, b_3, \ldots, b_{2^n}) = b_2 + b_3 - [0.5 \cdot \max(b_2 - 1, b_3)] + (b_4 + b_5 + \ldots + b_{2^n}) = M(a_2, a_3, \ldots, a_{2^n}) + p - 1 - [0.5 \cdot \max(a_2 + p - 1, a_3)] + [0.5 \cdot \max(a_2 - 1, a_3)]$. Осталось показать, что $p - 1 - [0.5 \cdot \max(a_2 + p - 1, a_3)] + [0.5 \cdot \max(a_2 - 1, a_3)] \geqslant 0$.

   Заметим, что $p \geqslant 2$, так как $t > 3$.

   Если $a_2 + p - 1 \leqslant a_3$, то $[0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)] = 0$, а значит $p - 1 - [0.5 \cdot \max(a_2 + p - 1, a_3)] + [0.5 \cdot \max(a_2 - 1, a_3)] > 0$.

   Если $a_2 - 1 > a_3$, то необходимо рассмотреть три случая в зависимости от четности p и a_2.

а) При четном p выполняется неравенство $[0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_2 + p - 1)] - [0.5 \cdot (a_2 - 1)] = [0.5(a_2 - 1)] + [0.5p] - [0.5(a_2 - 1)] = [0.5p]$, соответственно $p - 1 - [0.5p] \geqslant 0$ при $p \geqslant 2$.

б) При нечетном p и четном a_2 выполняется неравенство $[0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_2 + p - 1)] - [0.5 \cdot (a_2 - 1)] = [0.5a_2] + [0.5p] - [0.5a_2] + 1 = [0.5p] + 1$, соответственно $p - 1 - [0.5p] - 1 = [0.5p] + 1 - 2 = [0.5p] - 1 \geqslant 0$ при $p \geqslant 2$.

в) При нечетном p и нечетном a_2 выполняется неравенство $[0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_2 + p - 1)] - [0.5 \cdot (a_2 - 1)] = [0.5(a_2 - 1)] + [0.5p] - [0.5(a_2 - 1)] = [0.5p]$, соответственно $p - 1 - [0.5p] \geqslant 0$ при $p \geqslant 2$.

Если $a_2 - 1 \leqslant a_3 < a_2 + p - 1$, то $[0.5 \cdot \max(a_2 + p - 1, a_3)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_2 + p - 1)] - [0.5 \cdot a_3] \leqslant [0.5 \cdot (a_2 + p - 1)] - [0.5 \cdot (a_2 - 1)]$. Но выше было показано, что $p - 1 - \Big([0.5 \cdot (a_2 + p - 1)] - [0.5 \cdot (a_2 - 1)]\Big) \geqslant 0$. Приходим к выводу, что $p - 1 - [0.5 \cdot \max(a_2 + p - 1, a_3)] + [0.5 \cdot \max(a_2 - 1, a_3)] \geqslant 0$.

2. Пусть $t = 2 \cdot p + 1$, p — целое. Заметим, что $t > 3$, поэтому $p \geqslant 2$. Тогда положим $b_2 = a_2 + (p - 1), b_3 = a_3 + 1, b_4 = a_4, \ldots, b_{t-1} = a_{t-1}, b_t = a_t - 1, b_{t+1} = a_{t+1}, \ldots, b_{2^n} = a_{2^n}$. Заметим, что $b_2, b_3, \ldots, b_{2^n}$ удовлетворяют $\mu(n)$ условию.

Рассмотрим значение $M(b_2, b_3, \ldots, b_{2^n}) = b_2 + b_3 - [0.5 \cdot \max(b_2 - 1, b_3)] + (b_4 + b_5 + \ldots + b_{2^n}) = M(a_2, a_3, \ldots, a_{2^n}) + p - 1 + 1 - 1 - [0.5 \cdot \max(a_2 + p - 2, a_3 + 1)] + [0.5 \cdot \max(a_2 - 1, a_3)]$. Осталось показать, что $p - 1 - [0.5 \cdot \max(a_2 + p - 2, a_3 + 1)] + [0.5 \cdot \max(a_2 - 1, a_3)] \geqslant 0$.

Если $a_2 + p - 2 < a_3 + 1$, следовательно, $a_2 - 1 \leqslant a_3$, то $[0.5 \cdot \max(a_2 + p - 1, a_3 + 1)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_3 + 1)] - [0.5 \cdot a_3] \in \{0, 1\}$. Поэтому $p - 1 - ([0.5 \cdot (a_3 + 1)] - [0.5 \cdot a_3]) \geqslant 0$ при $p \geqslant 2$.

Обратим внимание, что в случае $a_2 + p - 2 = a_3$ выполняется неравенство $a_2 + p - 2 < a_3 + 1$.

Если $a_2 - 1 > a_3$, то $[0.5 \cdot \max(a_2 + p - 2, a_3 + 1)] - [0.5 \cdot \max(a_2 - 1, a_3)] = [0.5 \cdot (a_2 + p - 2)] - [0.5 \cdot (a_2 - 1)]$. Необходимо рассмотреть три случая в зависимости от четности p и a_2.

а) При четном a_2 выполняется равенство $[0.5\cdot(a_2+p-2)]-[0.5\cdot(a_2-1)]=[0.5a_2]-1+[0.5p]-[0.5a_2]+1=[0.5p]$, соответственно $p-1-[0.5p]\geqslant 0$ при $p\geqslant 2$.

б) При нечетном a_2 и четном p выполняется равенство $[0.5\cdot(a_2+p-2)]-[0.5\cdot(a_2-1)]=[0.5a_2]+[0.5p]-1-[0.5a_2]=[0.5p]-1$, соответственно $p-1-[0.5p]+1=[0.5p]>0$ при $p\geqslant 2$.

в) При нечетном a_2 и нечетном p выполняется неравенство $[0.5\cdot(a_2+p-2)]-[0.5\cdot(a_2-1)]=[0.5a_2]+[0.5p]+1-1-[0.5a_2]=[0.5p]$, соответственно $p-1-[0.5p]=[0.5p]+1-1>0$ при $p\geqslant 2$.

Если $a_2-1\leqslant a_3<a_2+p-2$, то $[0.5\cdot\max(a_2+p-2,a_3+1)]-[0.5\cdot\max(a_2-1,a_3)]=[0.5\cdot(a_2+p-2)]-[0.5\cdot a_3]\leqslant[0.5\cdot(a_2+p-2)]-[0.5\cdot(a_2-1)]$. Выше было показано, что $p-1-([0.5\cdot(a_2+p-2)]-[0.5\cdot(a_2-1)])\geqslant 0$ при $p\geqslant 2$.

Приходим к выводу, что $p-1+[0.5\cdot\max(a_2+p-2,a_3+1)]-[0.5\cdot\max(a_2-1,a_3)]\geqslant 0$.

□

Из леммы 14 вытекает следующая лемма.

Лемма 15. *Пусть $a_2,a_3,\ldots,a_{2^n}$ удовлетворяют условию $\mu(n)$, причем $a_4+a_5+\ldots+a_{2^n}>0$. Тогда существуют целые неотрицательные b_2,b_3, удовлетворяющие условиям $2b_2+3b_3=2^n, b_2+b_3>1$, такие что справедливо неравенство $M(a_2,a_3,a_4,\ldots,a_{2^n})\leqslant M(b_2,b_3,0,\ldots,0)$.*

В доказательстве леммы 10 работы [43] содержатся неточности, которые не влияют на конечный результат. Поэтому далее приведем уточненное доказательство.

Лемма 16. *Пусть $n\geqslant 4$, $a_2,a_3,\ldots,a_{2^n}$ удовлетворяют условию $\mu(n)$ и выполнено $a_4=a_5=\ldots=a_{2^n}=0, a_2<a_3$. Тогда существуют $b_2,b_3,\ldots,b_{2^n}$, такие что $b_4=b_5=\ldots=b_{2^n}=0, b_3<a_3$, которые удовлетворяют условию $\mu(n)$, и для которых справедливо неравенство $M(a_2,a_3,\ldots,a_{2^n})\leqslant M(b_2,b_3,\ldots,b_{2^n})$.*

Доказательство. Поскольку $a_3+a_2>1, a_2<a_3$, то либо $a_2=0, a_3\geqslant 2$, либо $a_2\geqslant 1, a_3\geqslant 2$. Если $a_3=2$, а $a_2\leqslant 1$, $3a_3+2a_2\leqslant 8<2^n$ при $n\geqslant 4$. Следовательно, если $a_2<a_3$, то $a_3>2$. Положим равными $b_2=a_2+3, b_3=$

$a_3 - 2, b_4 = b_5 = \ldots = b_{2^n} = 0$. Очевидно, что $b_2, b_3, b_4, \ldots, b_{2^n}$ удовлетворяют условию $\mu(n)$. Рассмотрим значение выражения $M(b_2, b_3, b_4, \ldots, b_{2^n}) = b_2 + b_3 - [0.5 \cdot \max(b_2 - 1, b_3)] + (b_4 + \ldots + b_{2^n}) = M(a_2, a_3, a_4, \ldots, a_{2^n}) + 1 - [0.5 \cdot \max(a_2 + 3 - 1, a_3 - 2)] + [0.5 \cdot a_3]$. Осталось показать, что $1 - [0.5 \cdot \max(a_2 + 2, a_3 - 2)] + [0.5 \cdot a_3] \geqslant 0$.

Если $a_2 + 2 < a_3 - 2$, то $1 - [0.5 \cdot \max(a_2 + 2, a_3 - 2)] + [0.5 \cdot a_3] = 1 - [0.5 \cdot (a_3 - 2)] + [0.5 \cdot a_3] = 1 - [0.5 \cdot a_3] + 1 + [0.5 \cdot a_3] = 2 > 0$.

Если $a_2 + 2 > a_3$, то $1 - [0.5 \cdot \max(a_2 + 2, a_3 - 2)] + [0.5 \cdot a_3] = 1 - [0.5 \cdot (a_2 + 2)] + [0.5 \cdot a_3] = 1 - [0.5 \cdot a_2] - 1 + [0.5 \cdot a_3] = [0.5 \cdot a_3] - [0.5 \cdot a_2] \geqslant 0$, так как $a_3 > a_2$.

Если $a_3 - 2 \leqslant a_2 + 2 \leqslant a_3$, то $1 - [0.5 \cdot \max(a_2 + 2, a_3 - 2)] + [0.5 \cdot a_3] = 1 - [0.5 \cdot (a_2 + 2)] + [0.5 \cdot a_3] \geqslant 1 - [0.5 \cdot a_3] + [0.5 \cdot a_3] = 1 > 0$. □

Из лемм 16 и 15 получаем следующую лемму.

Лемма 17. *Чтобы найти максимальное значение функции* $M(b_2, b_3, \ldots, b_{2^n})$ *при* $n \geqslant 4$, *достаточно перебрать удовлетворяющие условию* $\mu(n)$ $b_2, b_3, \ldots, b_{2^n}$, *для которых выполнены ограничения* $b_2 \geqslant b_3, b_4 = b_5 = \ldots = b_{2^n} = 0$.

Лемма 18. *Если* $n \geqslant 4$, *то максимальное значение функции* $M(x_2, x_3, \ldots, x_{2^n})$ *равно* $]3/2 \cdot [2^n/5][+[(2^n \bmod 5)/2]$ *и достигается при следующих* $x_2, x_3, \ldots, x_{2^n}$:

- $x_2 = [2^n/5] + 2, x_3 = [2^n/5] - 1, x_4 = \ldots = x_{2^n} = 0$, *если* $2^n \bmod 5 = 1$,
- $x_2 = [2^n/5] + 1, x_3 = [2^n/5], x_4 = \ldots = x_{2^n} = 0$, *если* $2^n \bmod 5 = 2$,
- $x_2 = [2^n/5] + 3, x_3 = [2^n/5] - 1, x_4 = \ldots = x_{2^n} = 0$, *если* $2^n \bmod 5 = 3$,
- $x_2 = [2^n/5] + 2, x_3 = [2^n/5], x_4 = \ldots = x_{2^n} = 0$, *если* $2^n \bmod 5 = 4$.

Доказательство. Согласно лемме 17, для поиска $x_2, x_3, \ldots, x_{2^n}$ достаточно перебрать $x_2, x_3, \ldots, x_{2^n}$, удовлетворяющие условию $\mu(n)$ и следующим ограничениям

1. $x_4 = x_5 = \ldots = x_{2^n} = 0$;
2. $x_2 \geqslant x_3$.

Отсюда следует, что $2^n = 2x_2 + 3x_3$, следовательно $2^n - 2x_2$ должен быть кратен 3.

1. Если $2^n \bmod 3 = 1$, то должно выполняться $x_2 \bmod 3 = 2$.
2. Если $2^n \bmod 3 = 2$, то должно выполняться $x_2 \bmod 3 = 1$.

Из равенства $2^n = 2x_2 + 3x_3$ следует то, что $x_2 \neq x_3$, а значит $x_2 > x_3$.

Соответственно, x_2 подберем в соответствии с этими требованиями. Учитывая, что $x_3 = (2^n - 2x_2)/3 < x_2$, получаем $x_2 \geqslant]2^n/5[$.

Рассмотрим значение функции $M(x_2, x_3, x_4, \ldots, x_{2^n}) = x_2 + x_3 - [0.5 \cdot \max(x_2 - 1, x_3)] =](x_2 + 1)/2[+ x_3 =](x_2 + 1)/2[+ (2^n - 2x_2)/3$. Рассмотрим два случая в зависимости от четности x_2.

1. Если x_2 — нечетное, тогда $M(x_2, x_3, x_4, \ldots, x_{2^n}) = (x_2 + 1)/2 + (2^n - 2x_2)/3 = (2^{n+1} - x_2 + 3)/6$. Учитывая ограничение $x_2 \geqslant]2^n/5[$, получаем, что максимум $M(x_2, x_3, x_4, \ldots, x_{2^n})$ достигается при x_2 наименьшем целом числе, неменьшим $]2^n/5[$ и дающий остаток от деления на 3 равный $3 - 2^n \bmod 3$.
2. Если x_2 — четное, тогда $M(x_2, x_3, x_4, \ldots, x_{2^n}) = x_2/2 + 1 + (2^n - 2x_2)/3 = (2^{n+1} - x_2 + 6)/6$. Учитывая ограничение $x_2 \geqslant]2^n/5[$, получаем, что максимум $M(x_2, x_3, x_4, \ldots, x_{2^n})$ достигается при x_2 наименьшем целом числе, неменьшим $]2^n/5[$ и дающий остаток от деления на 3 равный $3 - 2^n \bmod 3$.

Заметим, что если $n \bmod 4 = 0$, то $2^n \bmod 3 = 1, 2^n \bmod 5 = 1$. Если $n \bmod 4 = 1$, то $2^n \bmod 3 = 2, 2^n \bmod 5 = 2$. Если $n \bmod 4 = 2$, то $2^n \bmod 3 = 1, 2^n \bmod 5 = 4$. Если $n \bmod 4 = 3$, то $2^n \bmod 3 = 2, 2^n \bmod 5 = 3$.

Пусть $2^n = 3q + 1 = 5t + r$, где q, t, r — целые неотрицательные числа, $r \in \{1, 4\}$. Соответственно, $]2^n/5[= t + 1$. Учитывая справедливость равенства $]x[= [x] + 1$ в случае нецелого x, получаем $t = [2^n/5]$.

Представим t в виде суммы $3p + w$, где p — целое неотрицательное число, $w \in \{0, 1, 2\}$.

Рассмотрим два случая относительно остатков от деления 2^n на 5.

- $r = 1$. Заметим, что t — нечетное, так как $5t = 2^n - 1$. Теперь рассмотрим возможные значения w. Если $w = 1$, то $2^n = 5(3p + 1) + 1 = 3 \cdot 5p + 6 \neq 3q + 1$. Если $w = 2$, то $2^n = 5(3p + 2) + 1 = 3 \cdot 5p + 11 \neq 3q + 1$. Если $w = 0$, то $2^n = 5 \cdot 3p + 1 = 3 \cdot 5p + 1 = 3q + 1$. Отсюда следует, что $w = 0$. Тогда $x_2 = t + 2, x_3 = (2^n - 2x_2)/3 = (3t + r - 4)/3 = t - 1$. Соответственно, $M(x_2, x_3, x_4, \ldots, x_{2^n}) = 2t + 1 - [(t + 2 - 1)/2] = 2t + 1 - (t + 1)/2 = (3t + 1)/2 =]3/2 \cdot [2^n/5][$. Последнее следует из равенства $]3x/2[= (3x + 1)/2$, которое имеет место при нечетных x.
- $r = 4$. Обратим наше внимание, что t — четное, так как $5t = 2^n - 4$. Теперь рассмотрим возможные остатки от деления t на 3, то есть все

значения w. Если $w = 1$, то $2^n = 5(3p+1)+4 = 3\cdot 5p+9 \neq 3q+1$. Если $w = 2$, то $2^n = 5(3p+2)+4 = 3\cdot 5p+14 \neq 3q+1$. Если $w = 0$, то $2^n = 5\cdot 3p+4 = 3q+1$. Отсюда следует, что $w = 0$.

Тогда $x_2 = t+2, x_3 = (2^n - 2x_2)/3 = (3t+r-4)/3 = t$. Соответственно, $M(x_2, x_3, x_4, \ldots, x_{2^n}) = 2t+2-[(t+2-1)/2] = 2t+2-t/2 = 3t/2+2 =]3/2\cdot[2^n/5][+2$.

Пусть $2^n = 3q+2 = 5t+r$, где q, t, r — целые неотрицательные числа, $r \in \{2, 3\}$. Соответственно, $]2^n/5[= t+1$.

Рассмотрим два случая относительно остатков на деление 2^n на 5.

– $r = 2$. Заметим, что t — четное, так как $5t = 2^n - 2$. Теперь рассмотрим возможные значения w. Если $w = 1$, то $2^n = 5(3p+1)+2 = 3\cdot 5p+7 \neq 3q+2$. Если $w = 2$, то $2^n = 5(3p+2)+2 = 3\cdot 5p+12 \neq 3q+2$. Если $w = 0$, то $2^n = 5\cdot 3p+2 = 3\cdot 5p+2 = 3q+2$. Отсюда следует, что $w = 0$.

Тогда $x_2 = t+1, x_3 = (2^n - 2x_2)/3 = (3t+r-2)/3 = t$. Соответственно, $M(x_2, x_3, x_4, \ldots, x_{2^n}) = 2t+1-[(t+1-1)/2] = 2t+1-t/2 = 3t/2+1 =]3/2\cdot[2^n/5][+1$.

– $r = 3$. Также заметим, что t — нечетное, так как $5t = 2^n - 3$. Теперь рассмотрим значения w. Если $w = 1$, то $2^n = 5(3p+1)+3 = 3\cdot 5p+8 = 3q+2$. Если $w = 2$, то $2^n = 5(3p+2)+3 = 3\cdot 5p+13 \neq 3q+2$. Если $w = 0$, то $2^n = 5\cdot 3p+3 \neq 3q+2$. Отсюда следует, что $w = 1$.

Тогда $x_2 = t+3, x_3 = (2^n-2x_2)/3 = (3t+r-6)/3 = t-1$. Соответственно, $M(x_2, x_3, x_4, \ldots, x_{2^n}) = 2t+2-[(t+3-1)/2] = 2t+2-(t+1)/2 = (3t+3)/2 =]3/2\cdot[2^n/5][+1$.

□

Лемма 19. *При $n \geqslant 4$ справедлива следующая верхняя оценка*

$$\varphi_{CQ}(F, n, 3, 3) \leqslant 2^n -]3/2\cdot[2^n/5][-[(2^n \bmod 5)/2].$$

Доказательство. Подставим в оценку леммы 13 приводимые в лемме 18 значения x_2, x_3, а $x_4, \ldots, x_{2^n}$ положим равными 0. Заметим, что они удовлетворяют условию теоремы 5. Причем, получаемая верхняя оценка равна $2^n - M(x_2, x_3, 0, \ldots, 0)$. Согласно лемме 18, при выбранных таким образом x_2, x_3 достигается минимум $2^n -]3/2\cdot[2^n/5][-[(2^n \bmod 5)/2]$. □

Лемма 20. *Если для расшифровки функции $f \in F(n, 3, 3)$, где $n \geqslant 4$, ученик использует алгоритм, покрывающий не все наборы, тогда для однозначного восстановления функции ему потребуется как минимум $3 \cdot 2^{n-2} - 2$ запросов.*

Доказательство. Пусть $2^n = 4x + 4$, где x — целое неотрицательное. Пусть алгоритмом расшифровки не покрыто $r > 0$ наборов, при этом задано y запросов и ученик по ответам на свои запросы однозначно может восстановить загаданную функцию. Заметим, что $3x + 1 = 3 \cdot 2^{n-2} - 2$. В роли учителя на все первые $3x + 1$ запросов будем отвечать 0. Ответы на последующие запросы определим позднее при рассмотрении разных случаев.

Обратим внимание, что возможен один из двух случаев: либо все непокрытые наборы нули функции, либо все они являются ее единицами. Поскольку если бы часть непокрытых наборов была нулями, а оставшиеся непокрытые наборы единицами, то невозможно было без дополнительных запросов понять, какие из них и есть единицы. После y запросов $2^n - r$ наборов распались на классы эквивалентности, то есть подразбились на множества наборов, про которые известно, что значение функции на всех элементах одного множества одинаковое. Про все множества мощности не меньшее четырех ученик сразу понял, что в них нет единиц функции. Рассмотрим всевозможные значения r.

1. $r \geqslant 4$. Учитель может спрятать все единицы функции среди этих четырех непокрытых наборов. В силу этого, ученику недостаточно y запросов для однозначного восстановления функции, соответственно такой случай невозможен.
2. $r = 3$. Если среди множеств имеется множество мощности 3, тогда ученик не поймет единицы лежат в этом множестве или среди непокрытых наборов. Следовательно, множеств мощности 3 не должно быть. Если среди множеств имеется множество мощности 2, тогда ученик поймет, что две единицы лежат в каком-то из множеств мощности 2 или все единицы лежат среди непокрытых наборов, но в первом случае, не поймет какой из непокрытых наборов является третьей единицей. Соответственно, множеств мощности 2 также не должно быть. Приходим к выводу, что раз ученик однозначно восстановил функцию после y запросов, значит все множества имеют мощность не меньше 4, а непокрытые наборы и есть единицы функции. Согласно лемме 3, для объединения

$4x + 1$ наборов во множества мощности не меньшей 4, необходимо не менее $3x + 1$ запросов, иными словами, $y \geqslant 3 \cdot 2^{n-2} - 2$.

3. $r = 2$. Если среди множеств имеется множество мощности 2, тогда ученик поймет, что две единицы лежат в каком-то из множеств мощности 2, но не поймет какой из непокрытых наборов является третьей единицей. Соответственно, множеств мощности 2 также не должно быть. Если среди множеств имеются хотя бы два множества мощности 3, тогда ученик не определит, в каком из этих множеств лежат единицы функции. Исходя из этого возможные следующие два случая.
   - Имеется ровно одно множество мощности 3 и несколько множество мощности не меньшей 4. На образование множества мощности 3 необходимо 2 запроса, а на покрытие оставшихся $4x - 1$ наборов, согласно лемме 3, понадобится $3(x - 1) + 3$ запроса. Отсюда следует, что $y \geqslant 3x + 2 \geqslant 3 \cdot 2^{n-2} - 2$.
   - Отсутствуют множества мощности 3 и все наборы разбились на множества мощности не меньшей 4. Согласно лемме 3, для этого необходимо $3x + 2$ запроса. На первые $3x + 1$ запросов учитель отвечает 0. Если же ученик задает $(3x + 2)$-й запрос и учитель видит, что если ответ на этот запрос будет равен 0, то покрытыми станут $4x+2$ набора и они распадутся на множества мощности не меньшей 4, тогда в этом случае учитель отвечает не 0, раскрывая единицы в той компоненте запроса, которая относится ко множеству меньшей мощности. Тем самым, учитель гарантирует, что не обманывал ученика и действительно его ответы соответствуют какой-то функции из $F(n, 3, 3)$.

     Рассмотреть ответ на $(3x + 2)$-й запрос необходимо было лишь для последней цели, а так и этот случай демонстрирует, что ученику потребуется как минимум $3x + 1$ запросов, то есть $y \geqslant 3 \cdot 2^{n-2} - 2$.
4. $r = 1$. Если имеется и множество мощности 2, и множество мощности 3, тогда ученик не поймет, единицы функции лежат во множестве мощности 3 или во множестве мощности 2 и непокрытом наборе. Если имеется несколько множеств мощности 3, тогда ученик не поймет, в каком из них лежат единицы функции. Если имеются несколько множеств мощности 2, тогда ученик может понять, что одна единица —

это непокрытый набор, но не сможет определить, в каком из множеств мощности 2 лежат оставшиеся единицы. Следовательно, возможны следующие три случая.

- Имеется ровно одно множество мощности 2, нет множеств мощности 3 и имеются несколько множество мощности не меньшей 4. На образование множества мощности 2 необходим 1 запрос, а на покрытие оставшихся $4x+1$ наборов, согласно лемме 3, понадобится $3x+1$ запросов. Отсюда следует, что $y \geqslant 3x+2 \geqslant 3 \cdot 2^{n-2} - 2$.
- Имеется ровно одно множество мощности 3, нет множеств мощности 2 и имеются несколько множество мощности не меньшей 4. На образование множества мощности 3 необходимо 2 запроса, а на покрытие оставшихся $4x$ наборов, согласно лемме 3, понадобится $3x$ запросов. Отсюда следует, что $y \geqslant 3x+2 \geqslant 3 \cdot 2^{n-2} - 2$.
- Отсутствуют множества мощности 2 и 3 и все наборы разбились на множества мощности не меньшей 4. Согласно лемме 3, для этого необходимо $3x+3$ запроса. На первые $3x+2$ запросов учитель отвечает 0. Если же ученик задает $(3x+3)$-й запрос и учитель видит, что если ответ на этот запрос будет равен 0, то покрытыми станут $4x+3$ набора и они распадутся на множества мощности не меньшей 4, тогда в этом случае учитель отвечает не 0, раскрывая единицы в той компоненте запроса, которая относится ко множеству меньшей мощности. Тем самым, учитель гарантирует, что не обманывал ученика и действительно его ответы соответствуют какой-то функции из $F(n,3)$. Рассмотреть ответ на $(3x+3)$-й запрос необходимо было лишь для последней цели, а так и этот случай демонстрирует, что ученику потребуется как минимум $3x+1$ запросов, то есть $y \geqslant 3 \cdot 2^{n-2} - 2$.

□

Пусть A — алгоритм расшифровки $F(n,k,k)$, покрывающий все 2^n наборы. Тогда через $C(A)$ будем обозначать такое число q, что первые $q-1$ запросов алгоритма A покрывают не все наборы, а q запросов покрывают все 2^n набо-

ров. Под $N(A, x, y)$ будем понимать количество компонент связности размера y, которые образовались после отправки первых x запросов алгоритма A.

Лемма 21. *Пусть* $2^n = 5 \cdot q + r, r \in [1,4], n \geqslant 4$, *и для расшифровки функции* $f \in F(n,3,3)$ *ученик использует алгоритм расшифровки* A, *покрывающий все* 2^n *наборов. Если* $N(A, C(A), 2) + N(A, C(A), 3) \leqslant 1$, *ученик задаст не менее* $3 \cdot 2^{n-2} - 2$ *запросов.*

Доказательство. Пусть $2^n = 4x + 4$. Рассмотрим три случая.

1. $N(A, C(A), 2) + N(A, C(A), 3) = 0$
   После $C(A)$ запросов образовались множества мощности строго больше 3. Согласно лемме 3, для этого потребуется не менее $3 \cdot 2^{n-2}$ запросов.
2. $N(A, C(A), 2) = 1, N(A, C(A), 3) = 0$
   После $C(A)$ запросов образовались множества мощности строго больше 3 и ровно одно множество мощности 2. Согласно лемме 3, для образования множеств мощности не меньшей 4 из $2^n - 2 = 4x + 2$ потребуется не менее $3x + 2 = 3 \cdot 2^{n-2} - 1$ запросов.
3. $N(A, C(A), 2) = 0, N(A, C(A), 3) = 1$
   После $C(A)$ запросов образовались множества мощности строго больше 3 и ровно одно множество мощности 3. Согласно лемме 3, для образования множеств мощности не меньшей 4 из $2^n - 3 = 4x + 1$ потребуется не менее $3x + 1 = 3 \cdot 2^{n-2} - 2$ запросов.

□

Лемма 22. *При* $n \geqslant 6$ *верно неравенство*

$$\varphi_{CQ}(F, n, 3, 3) \geqslant 2^n -]3/2 \cdot [2^n/5][-[(2^n \bmod 5)/2].$$

Доказательство. Из лемм 11, 20 вытекает то, что для расшифровки функции с тремя единицами невыгодно использовать алгоритм расшифровки, непокрывающий все наборы. Поскольку верхняя оценка леммы 19 утверждает, что $\varphi_{CQ}(F, n, 3, 3) \leqslant 2^n -]3/2 \cdot [2^n/5][-[(2^n \bmod 5)/2] \leqslant 2^n - 1.5 \cdot [2^n/5] - [(2^n \bmod 5)/2]$.

Соответственно, чтобы задать как можно меньше запросов, ученик вынужден использовать алгоритм, покрывающий все наборы. Пусть первый запрос, после ответа на который окажутся покрытыми все 2^n наборов, имеет номер w. Пусть после w запросов исходные 2^n наборов объединятся в x_2 компоненты

связности размера 2, x_3 компоненты связности размера 3, ..., x_{2^n} компоненты связности размера 2^n, иными словами верно соотношение $2^n = 2x_2 + 3x_3 + 4x_4 + \ldots + 2^n \cdot x_{2^n}$. На первые w запросов ученика (a, b) учитель будет отвечать следующим образом:

- На первые $w - 1$ запросов учитель отвечает 0.
- Ответ на w-й запрос определяется следующим образом.

  Если $x_2 \leqslant x_3$, учитель отвечает 0. Если $x_2 > x_3$, учитель отвечает

  1. 1, если a — один из непокрытых наборов, покрываемых запросом (a, b),
  2. -1, в противном случае.

  Из леммы 21 следует $x_2 + x_3 > 1$.

Спустя w запросов ученик понимает, что во всех $x_4 + x_5 + \ldots + x_{2^n}$ компонентах связности размера строго больше 3 нет единиц.

Если на w-й запрос ученик в ответ получит 0, тогда он поймет, что все единицы находятся в одной из компонент связности размера 3. Для определения нужной компоненты ученику потребуется $[x_3/2]$ запросов.

Если на w-й запрос ученик в ответ получит отличный от 0, тогда он найдет в точности одну единицу. Не нарушая общности будем считать, что на w-й запрос (a, b) ученик получил в ответ 1. Тогда он понимает, что a — единица, а компонента, представителем которой является набор b, состоит полностью из нулей загаданной функции. Следовательно, ученику остается найти оставшиеся 2 единицы и они очевидно лежат в какой-то из компонент связности размера 2. Для определения нужной ученику потребуется

- $[x_2/2]$ запросов, если и b был уже покрыт первыми $w - 1$ запросами, иными словами, суммарно компонента связности с представителем в b и набор a образуют компоненту размера хотя бы 3,
- $[(x_2 - 1)/2]$, если и a, и b оба не были покрыты первыми $w - 1$ запросами, а значит суммарно после w запросов образуют компоненту связности размера 2, соответственно, про одну из x_2 компонент ученик полностью знает значение функции на каждом ее элементе, поэтому остается искать компоненту с двумя единицами среди меньшего числа компонент.

Заметим, что $w = x_2 + 2x_3 + 3x_4 + \ldots + (i - 1)x_i + \ldots + (2^n - 1)x_{2^n} = 2^n - (x_2 + x_3 + \ldots + x_{2^n})$, поскольку для образования компоненты связности размера i необходимо $i - 1$ запросов. Соответственно, если $x_2 \leqslant x_3$ ученик задаст не

менее $2^n - (x_2 + x_3 + \ldots + x_{2^n}) + [x_3/2]$, а если $x_2 > x_3$, то вынужден будет задать не менее $2^n - (x_2 + x_3 + \ldots + x_{2^n}) + [(x_2 - 1)/2]$ запросов.

Цель ученика подобрать такие $x_2, x_3, \ldots, x_{2^n}$, чтобы минимизировать это количество запросов, а значит максимизировать величину $(x_2+x_3+\ldots+x_{2^n})-[x_3/2]$ при $x_2 \leqslant x_3$ и $(x_2+x_3+\ldots+x_{2^n})-[(x_2-1)/2]$ при $x_2 > x_3$, соответственно максимизировать функцию $M(x_2, x_3, \ldots, x_{2^n}) = (x_2+x_3+\ldots+x_{2^n})-[\max(x_2-1, x_3)/2]$. Согласно лемме 18, максимальное значение этой функции равно $]3/2 \cdot [2^n/5][+[(2^n \bmod 5)/2]$, а значит ученик вынужден задать не менее $2^n-]3/2 \cdot [2^n/5][-[(2^n \bmod 5)/2]$.

□

Теорема 8. *При $n \geqslant 6$ сложность расшифровки класса $F(n, 3, 3)$ запросами на сравнение равна*

$$\varphi_{CQ}(F, n, 3, 3) = 2^n-]3/2 \cdot [2^n/5][-[(2^n \bmod 5)/2].$$

Доказательство. Доказательство теоремы следует из лемм 19 и 22. □

2.3.6 Порядок сложности расшифровки $F(n, k, i)$

Вопрос исследования значения сложности расшифровки запросами на сравнение класса функций арности n, имеющих вес из отрезка $[i, k]$, в общем случае довольно непрост. Для отдельных случаев ($i = 0$, $i = 1$) удалось получить точное значение сложности расшифровки или верхнюю и нижнюю оценку этой функции, отличающиеся не более чем на один. С доказательства этих фактов и начинается данный раздел. Далее в разделе приводится лемма 23, утверждающая, что для расшифровки функций веса k необходимо задать не менее $2 \cdot \left[2^n/3\right] - \left]2/3 \cdot (k+3)\right[$ запросов на сравнение. Завершает раздел доказательство теоремы 11 о порядке расшифровке функций ограниченного веса.

Теорема 9. *При $n \geqslant 2, 2^{n-1} \geqslant k$ сложность расшифровки класса $F(n, k, 0)$ запросами на сравнение равна*

$$\varphi_{CQ}(F, n, k, 0) = G(k, 2^n).$$

Доказательство. Докажем верхнюю оценку. Пусть $2^n = q \cdot (k+1) + r, r \in [0, k]$, q, r — неотрицательные целые числа, тогда создадим q компонент связности размера $k+1$ за $q \cdot k$ запросов, как это делается в доказательстве леммы 12. Заметим, что $q \geqslant 1$, так как $k \leqslant 2^{n-1}$. Значение на каждом из покрытых наборов однозначно восстанавливается. Осталось восстановить значение на оставшихся r наборах. Уже задали $kq = k \cdot [2^n/(k+1)]$ запросов, зададим еще r запросов, где в каждом запросе первая компонента — это один из непокрытых r наборов, а вторая — любой из покрытых ранее. Тогда по ответам на эти запросы мы однозначно восстановим значение функции на всех наборах.

Значит, для класса $F(n, k, 0)$ верхняя оценка не более $k \cdot [2^n/(k+1)] + (2^n \bmod (k+1))$, что равно $G(k, 2^n)$.

Докажем нижнюю оценку. На каждый запрос ученика будем отвечать числом 0. Если в какой-то момент у ученика остается компонента связности размера не более k, то он не знает, загадана функция положительного веса или нулевого, поэтому вынужден продолжать задавать запросы. Ученик разгадает функцию, если задаст такое количество запросов, которое необходимо, чтобы все имеющиеся наборы разбить на компоненты связности размера строго больше k. Согласно лемме 3 это возможно сделать не менее чем за $k \cdot [2^n/(k+1)] + (2^n \bmod (k+1))$ запросов. □

Теорема 10. *При $n \geqslant 2, 2^{n-1} \geqslant k \geqslant 1, 2^n \bmod (k+1) = k$ сложность расшифровки класса $F(n, k, 1)$ запросами на сравнение равна*

$$\varphi_{CQ}(F, n, k, 1) = G(k, 2^n) - 1.$$

При $n \geqslant 2, 2^{n-1} \geqslant k \geqslant 1, 2^n \bmod (k+1) = 0$ сложность расшифровки класса $F(n, k, 1)$ запросами на сравнение равна

$$\varphi_{CQ}(F, n, k, 1) = G(k, 2^n).$$

При $n \geqslant 2, 2^{n-1} \geqslant k \geqslant 1, 2^n \bmod (k+1) \in (0, k)$ сложность расшифровки класса $F(n, k, 1)$ запросами на сравнение удовлетворяет следующим ограничениям:

$$G(k, 2^n) - 1 \leqslant \varphi_{CQ}(F, n, k, 1) \leqslant G(k, 2^n).$$

Доказательство. Верхняя оценка для случая $2^n \bmod (k+1) < k$ следует из того, что $F(n, k, 1) \subset F(n, k, 0)$, поэтому можно применить алгоритм расшифровки функций класса $F(n, k, 0)$.

Пусть $2^n \bmod (k+1) = k$, тогда создадим $[2^n/(k+1)]$ компонент связности размера $k+1$ и одну компоненту связности размера k. Для этого потребуется выполнить в точности $k[2^n/(k+1)] + (k-1)$ запросов, что соответствует определению $G(k, 2^n) - 1$. Утверждается, что этих запросов достаточно для полного восстановления загаданной функции. Действительно, в силу замечания 1 значение функции однозначно восстановится на всех наборах, попавших в компоненты связности размера $k+1$. Если все наборы, попавшие в компоненту связности размера k, лежат в одном классе эквивалентности (т.е. при формированиии компоненты размера k в ответ на запросы были получены только нули), то они точно являются единицами в случае, когда среди остальных наборов лежат только нули, и точно нулями в случае, когда среди остальных наборов найдена хотя бы одна единица. Если при формированиии компоненты размера k хотя бы раз в ответ был получен не нуль, то значения на всех наборах в силу замечания 1 будут восстановлены.

Докажем нижнюю оценку. На каждый запрос ученика будем отвечать следующим образом. Если после текущего запроса остаются компоненты связности размера не более k, то отвечаем числом 0. Если после текущего запроса не останется компонент связности размера не более k, а сам запрос объединяет две компоненты связности размера не более k, то можем ответить как числом 1, так и -1, тогда ученик поймет, какая функция загадана и что у нее вес положительный. Если после текущего запроса не останется компонент связности размера не более k, а сам запрос объединяет компоненту размера строго больше k и компоненту размера не более k, то ответим так, чтобы ученик понял, что все единицы функции лежат в компоненте размера не более k, после этого ученик опять-таки узнает загаданную функцию. Но заметим, что в последнем случае он мог и не посылать этот запрос, так как все компоненты связности, кроме этой, точно не содержат единиц функции, поскольку они размера строго больше k, значит, все единицы лежат в компоненте размера не более k.

При такой стратегии ответов учителя ученик разгадает функцию, если задаст такое количество запросов, чтобы компонент связности размера не более k стало равно 0 или 1. Первое согласно лемме 3 возможно сделать только не менее чем за $k \cdot [2^n/(k+1)] + (2^n \bmod (k+1))$ запросов. Второе же возможно не менее чем за $k \cdot [(2^n - x)/(k+1)] + ((2^n - x) \bmod (k+1)) + (x-1)$ запросов, где $x \in [1, k]$ — количество наборов, которое будет в компоненте связности размера не более k. Согласно лемме 4 минимальное значение этой величины

достигается при $x = 1$ и равно $k \cdot [(2^n - 1)/(k + 1)] + ((2^n - 1) \bmod (k + 1))$. В силу леммы 5 справедливо неравенство $G(k, 2^n - 1) = k \cdot [(2^n - 1)/(k + 1)] + ((2^n - 1) \bmod (k + 1)) \leqslant k \cdot [2^n/(k + 1)] + (2^n \bmod (k + 1)) = G(k, 2^n)$, поэтому левая часть и взята в качестве нижней оценки для данного класса. Это количество соответствует определению $G(k, 2^n - 1)$, а согласно лемме 5 выполнено $G(k, 2^n - 1) \geqslant G(k, 2^n) - 1$. □

Лемма 23. *При $k \geqslant 2$ сложность расшифровки класса $F(n, k, k)$ запросами на сравнение не меньше $2 \cdot \left[2^n/3\right] - \left]2/3 \cdot (k + 3)\right[$.*

Доказательство. Обозначим $Q = 2 \cdot \left[2^n/3\right] - \left]2/3 \cdot (k+3)\right[$. Для начала поймем, какие компоненты связности могут образоваться, если ученик задаст ровно Q запросов. Воспользуемся известным неравенством $E + K \geqslant V$, где E, V, K — число ребер, вершин и компонент связности графа соответственно. В нашем случае $V = 2^n$, $E = Q$, подставляя эти значения в упомянутое неравенство, получаем $K \geqslant V - E \geqslant \left[1/3 \cdot 2^n\right] + \left]2/3 \cdot (k + 3)\right[$. Значит, средний размер компоненты связности $V/K \leqslant 2^n/\left(\left[1/3 \cdot 2^n\right] + \left]2/3 \cdot (k + 3)\right[\right) < 3$. Обозначим через S сумму размеров всех компонент связности, имеющих размер 1 или 2. Докажем справедливость неравенства $S \geqslant k + 3$. Предположим противное, т.е. $S < k + 3$. Тогда сумма размеров остальных компонент связности не меньше $2^n - (k+2)$. Согласно лемме 3, чтобы p вершин разбить на компоненты связности размера хотя бы 3, необходимо не менее $2 \cdot [p/3] + (p \bmod 3)$ запросов. В силу леммы 6 для того, чтобы не менее $2^n - (k + 2)$ вершин разбить на компоненты связности размера хотя бы 3, необходимо не менее $2 \cdot [(2^n - (k + 2))/3] + ((2^n - (k + 2)) \bmod 3)$ запросов. Следовательно, на образование компонент связности размера хотя бы 3 необходимо больше запросов, чем Q, поскольку имеет место неравенство $2 \cdot \left[(2^n - (k+2))/3\right] + ((2^n - (k+2)) \bmod 3) > 2 \cdot \left[2^n/3\right] - \left]2/3 \cdot (k+3)\right[$ согласно лемме 7. Противоречие. Значит общий размер маленьких компонент связности (т.е. компонент связности размера 1 и 2) не менее $k + 3$.

Теперь приведем стратегию ответов учителя на первые Q запросов. Также покажем, что хотя бы две функции из класса $F(n, k, k)$ будут удовлетворять ответам учителя на эти запросы, т.е. это будет означать, что, во-первых, ответы учителя корректны и действительно соответствуют какой-то функции из класса $F(n, k, k)$, во-вторых, ученик вынужден продолжить задавать запросы для однозначного восстановления функции. Рассмотрим два случая.

Случай 1: k четное. На каждый из Q запросов ученика ответим числом 0. Соответственно все наборы ученика распадутся на компоненты связности, где внутри каждой компоненты связности значение функции на всех ее вершинах одинаковое. Чтобы показать, что как минимум две функции из класса $F(n, k, k)$ удовлетворяют этим ответам, достаточно предъявить два способа спрятать все единицы функции только в компонентах размера 1 и 2. Это осуществимо в силу леммы 8, если положить $v = k$, а набор a положить равным $1, \ldots, 1, 2, \ldots, 2$, где количество единиц равно числу компонент связности размера 1 и количество двоек равно числу компонент связности размера 2.

Случай 2: k нечетное. Если после ответа на текущий запрос не окажутся покрытыми все 2^n наборов, то отвечаем на запрос числом 0. Иначе отвечаем так, что последним покрываемым набором станет единица функции. Если в запрос объединяются два последних непокрытых набора, то единицу помещаем в любой из них. Соответственно после такого запроса ученик однозначно определяет ровно одну единицу и понимает, чему равно значение на компоненте, которая содержалась в последнем запросе. На все последующие запросы, объединяющие компоненту связности G с компонентой связности H, о которой все известно, будем отвечать так, чтобы оказалось, что и в компоненте связности G точно нет единиц. На все последующие запросы, объединяющие компоненту связности G с компонентой связности H, ни в одной из которых не удалось узнать значение функции на наборах компоненты, будем отвечать 0.

Рассмотрим случай, когда ученик задал Q запросов и в ответ на каждый из них получил 0. Тогда существует как минимум один непокрытый набор, а может, только ровно один, в любой из этих наборов положим ровно одну единицу. Соответственно одну из компонент связности размера 1 учитель для себя определил как одну единицу. Теперь его цель — выбрать из оставшихся компонент размера 1 и 2 два подмножества компонент суммарного размера $k - 1$. Это осуществимо в силу леммы 8, если считать $v = k - 1$, а набор a положить равным $1, \ldots, 1, 2, \ldots, 2$, где количество единиц равно числу оставшихся компонент связности размера 1 и количество двоек равно числу компонент связности размера 2.

Рассмотрим случай, когда ученик своими Q запросами покрыл все наборы, значит, в какой-то момент он раскрыл одну единицу, причем все остальные элементы, попавшие с этой единицей в одну компоненту связности, точно нули функции. Возможны две ситуации: раскрытая единица лежит в компоненте

связности размера 2 или хотя бы 3. Если раскрытая единица лежит в компоненте размера хотя бы 3, то среди компонент размера 1 и 2, суммарный размер которых $S \geqslant k+3$, необходимо двумя способами спрятать $k-1$ единицу. Это возможно в силу леммы 8, если считать $v = k-1$, а набор a положить равным $1, \ldots, 1, 2, \ldots, 2$, где количество единиц равно числу компонент связности размера 1 и количество двоек равно числу компонент связности размера 2. Если раскрытая единица лежит в компоненте размера 2, то среди других компонент связности размера 1 и 2, суммарный размер которых не менее $k+1$, необходимо спрятать $k-1$ единицу двумя способами, что также легко осуществимо в силу леммы 8, если считать $v = k-1$, а набор a положить равным $1, \ldots, 1, 2, \ldots, 2$, где количество единиц равно числу компонент связности размера 1 и количество двоек равно числу оставшихся компонент связности размера 2. □

Теорема 11. *Для любого $k = k(n)$, такого, что $k \geqslant 2, k = o(2^n)$, сложность расшифровки класса $F(n,k,i)$ запросами на сравнение при $n \to \infty$ удовлетворяет следующим соотношениям:*

$$\begin{cases} 7/10 \cdot 2^n \lesssim \varphi_{CQ}(F,n,k,i) \lesssim k/(k+1) \cdot 2^n & \text{при } i \leqslant 3 \leqslant k, \\ 2/3 \cdot 2^n \lesssim \varphi_{CQ}(F,n,k,i) \lesssim k/(k+1) \cdot 2^n & \text{при } i > 3 \text{ или } k = 2. \end{cases}$$

Доказательство. Учитывая вложение $F(n,k,i) \subseteq F(n,k,0)$, по теореме 9 получаем $\varphi_{CQ}(n,k,i) \leqslant \varphi_{CQ}(n,k,0) = k \cdot [2^n/(k+1)] + (2^n \bmod (k+1))$. Вспоминая, что $(2^n \bmod (k+1)) \in [0,k]$ и $k = o(2^n)$, приходим к соотношению $\varphi_{CQ}(n,k,i) \lesssim k/(k+1) \cdot 2^n$.

Учитывая вложение $F(n,k,k) \subseteq F(n,k,i)$, по лемме 23 получаем $\varphi_{CQ}(n,k,i) \geqslant \varphi_{CQ}(n,k,k) \geqslant 2[2^n/3] - \big]2/3 \cdot (k+3)\big[$. Следовательно, справедливо соотношение $2/3 \cdot 2^n \lesssim \varphi_{CQ}(n,k,i)$.

Если $i \leqslant 3 \leqslant k$, то, учитывая вложение $F(n,3,3) \subseteq F(n,k,i)$, на основании леммы 22 заключаем, что $\varphi_{CQ}(n,k,i) \geqslant \varphi_{CQ}(n,3,3) \geqslant 2^n - \Big]3/2 \cdot [2^n/5]\Big[- \big[(2^n \bmod 5)/2\big]$. Поэтому при $i \leqslant 3 \leqslant k$ справедлива оценка $7/10 \cdot 2^n \lesssim \varphi_{CQ}(n,k,i)$. □

Глава 3. Расшифровка функций замкнутых классов Поста запросами на значение

Глава посвящена исследованию значения сложности точной расшифровки замкнутых классов Поста (рисунок 1) запросами на значение. При этом учитель выбирает функцию из одного из классов решетки, а ученику известен сам класс и то, что у функции n переменных и не более k из них являются существенными. При рассмотрении задачи для многих классов учитывается малость k относительно n. Например, при расшифровке классов функций логических сумм (S_1, S_3, S_5, S_6) рассматриваются $k = o(n)$, а для классов линейных функций считается, что $\log_2 k = o(\log_2 n)$.

В ходе доказательств в этой главе демонстрируется связь теории расшифровки с другими областями математики. Так например, при расшифровке класса всех булевых функций (C_1) и классов функций "счетной этажерки" (F_j^i, где $j \in \{1, 2, 3, 4\}$, i — целое число, не меньшее 2) показано, что задача расшифровки функций арности n с не более k существенными переменными сводится к задаче построения бинарных матриц с n столбцами, удовлетворяющих следующему свойству: для любых k столбцов в матрице имеются все 2^k бинарных строк. Эти матрица являются объектом изучения теории тестирования. Несмотря на долгую историю исследования этого вопроса, до сих пор неизвестен размер $\alpha(n, k)$ (количество строк) этих матриц. Экспертами этой области удается лишь получать точные значения $\alpha(n, k)$ для фиксированных значений параметров n, k и некоторые общие оценки, упомянутые во введении данной работы. Но ни порядок, а тем более асимптотика данной функции еще неизвестны. Поскольку в данной главе показывается, что сложность расшифровки некоторых классов равна $\alpha(n, k)$, то для таких классов мы будем говорить, что известна условная асимптотика, равная $\alpha(n, k)$ в том смысле, что когда станет известна асимптотика функции $\alpha(n, k)$, тогда будет определена и асимптотика сложности расшифровки этих классов.

Кроме того, что существует связь между задачей расшифровки и задачей построения упомянутых выше матриц, оказывается имеется отношение между задачей расшифровки и задачей построения линейных кодов. Это отношение косвенно демонстрируется в этой главе, но явно используется в доказательстве

утверждения в работе [29], а на само утверждение опирается доказательство леммы 52 данной работы.

Напомним, что довольно многие классы Поста уже исследовались с точки зрения сложности расшифровки запросами на значение. Все полученные результаты приводятся в этой главе для демонстрации общей картины результатов для всей решетки замкнутых классов Поста (рисунок 1). Если класс выделен треугольником на рисунке, то это результат, полученный ранее в других работах, далее все такие результаты будут называться утверждениями с указанием источника и автора. Если класс выделен квадратом, то результат ранее в литературе не встречался и впервые упоминается в данной работе, далее такие результаты в работе будут называться теоремами.

Глава состоит из девяти разделов. В первом разделе приведены все вспомогательные определения и утверждения, используемые далее в доказательствах. Следующие семь разделов демонстрируют результаты исследования сложности расшифровки для классов, сгруппированных по букве в их обозначении: $C_i, A_i, D_i, F_j^i, S_i, L_i, O_i$. Глава завершается разделом с теоремами, в которые объединены результаты семи разделов.

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.

3.1 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 го-

ворить, что *запрос содержит фиксацию*, если переменные в фиксации имеют такие же значения, как и эти же переменные в запросе.

Через $M(n,k)$ будем обозначать любую бинарную матрицу с наименьшим числом строк и n столбцами, такую, что в любых k столбцах встретятся все 2^k двоичных наборов-строк. Число строк в такой матрице обозначим через $\alpha(n,k)$, а саму такую матрицу будем называть покрывающей.

Для упрощения выкладок будем считать $\alpha(n,0) = 0$ для любого натурального n.

Также напомним обозначение

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

Под $a \vee b$, где a, b — векторы одной размерности, будем понимать вектор, каждая компонента которого получается дизъюнкцией соответствующих компонент векторов a, b.

Лемма 24. *(P. Damaschke [22]) Пусть f — булевская функция, и пусть известны два набора $X = x_1x_2\ldots x_n$ и $Y = y_1y_2\ldots y_n$, что $f(x_1x_2\ldots x_n) = 0$, $f(y_1y_2\ldots y_n) = 1$. Тогда за не более $\log_2 n$ запросов на значение можно найти номер одной из существенных переменных.*

Приведем доказательство данного факта из [22], поскольку оно облегчает понимание дальнейших выкладок.

Доказательство. По двум наборам X, Y построим третий $Z = z_1z_2\ldots z_n$. Если $x_i = y_i$, то положим $z_i = x_i$. Иначе примерно половину переменных в Z зафиксируем как зафиксированы они в наборе X, остальные — как в наборе Y. Запросим значение на наборе Z. Если значение равно 0, то положим $X = Z$, иначе положим $Y = Z$. Заметим, что новая пара наборов X, Y отличается как минимум в 2 раза меньшем числе переменных, чем было до этого. Продолжаем этот процесс, пока наборы отличаются в более чем одной переменной. Эта отличающая переменная и будет существенной. □

Лемма 25. *Справедливо неравенство $\beta_i(n,k) \geqslant \alpha(n,k) - 1, i \in \{0,1\}$.*

Доказательство. Докажем для $\beta_1(n,k)$, для $\beta_0(n,k)$ доказательство аналогичное. Для этого покажем, что $\alpha(n,k)$ ограничена сверху величиной $\beta_1(n,k) + 1$. Любая почти покрывающая матрица типа 1 либо уже какая-то бинарная покрывающая матрица быть может с неминимальным числом строк $\alpha_1(n,k)$, либо

становится бинарной покрывающей матрицей после добавления к ее строкам строки из всех единиц. Следовательно,

$$\beta_1(n,k)+1 \geqslant \alpha_1(n,k) \geqslant \alpha(n,k).$$

Значит $\beta_1(n,k) \geqslant \alpha(n,k)-1$. □

Лемма 26. *Справедливо соотношение* $k\log_2 n = o(\alpha(n,k))$ *при* $k = o(n), k, n \to \infty$.

Доказательство. Из условий леммы и определения $\alpha(n,k)$ имеем

$$\frac{k\log_2 n}{\alpha(n,k)} \geqslant 0.$$

При оценке отношения $\frac{k\log_2 n}{\alpha(n,k)}$ используем нижнюю оценку из работы [34], которая имеет место для $k \geqslant 2$ и $n \to \infty$:

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

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

Учитывая, что $k = o(n)$, $k, n \to \infty$, применим теорему 3.14 [6] ("теорему о двух милиционерах") к неравенствам и получим соотношение леммы. □

Лемма 27. *Справедливо соотношение* $k\log_2 n = o((2^p-1)\alpha(n-p,k-p))$ *при* $1 \leqslant p < k-1, k = o(n), k, n \to \infty$.

Доказательство. Из условий леммы и определения $\alpha(n,k)$ имеем

$$\frac{k\log_2 n}{(2^p-1)\alpha(n-p,k-p)} \geqslant 0.$$

Используем нижнюю оценку на $\alpha(n,k)$ [34] при оценке отношения $\frac{k\log_2 n}{(2^p-1)\alpha(n-p,k-p)}$.

$$\frac{k\log_2 n}{(2^p-1)\alpha(n-p,k-p)} \leqslant \frac{k\log_2 n}{(2^p-1)2^{k-p-2}\log_2(n-k+2)(1+o(1))}$$

Учитывая, что $k = o(n), p < k-1$, значит $p = o(n)$, $k, n \to \infty$, применим теорему 3.14 [6] ("теорему о двух милиционерах") к неравенствам и получим соотношение леммы. □

Лемма 28. *При $k \geqslant 2$ справедливо соотношение $\alpha(n,k) \asymp \log n$ при $n \to \infty$.*

Доказательство. Доказательство следует из нижней оценки с [34] и верхней оценки с [27], которые имеют место для $k \geqslant 2$ и $n \to \infty$:

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

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

□

Лемма 29. *При $k,n \to \infty, k = o(n)$*

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

Доказательство. Рассмотрим $(2^p-1)\alpha(n-p,k-p)$ при $p = k-1$, $p = k-2$ и $k,n \to \infty, k = o(n)$.

$$(2^{k-1}-1)\alpha(n-(k-1),k-(k-1)) = (2^{k-1}-1)\alpha(n-(k-1),1) = 2^k - 2$$

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

Последнее равенство вытекает из [26].

Получается при $k,n \to \infty, k = o(n)$

$$(2^{k-1}-1)\alpha(n-(k-1),k-(k-1)) < (2^{k-2}-1)\alpha(n-(k-2),k-(k-2))$$

Следовательно, при $k,n \to \infty, k = o(n)$ максимум выражения $(2^p-1)\alpha(n-p,k-p)$ достигается точно не на $p = k-1$, поэтому $S_{n,k} > (2^{k-1}-1)\alpha(n-(k-1),k-(k-1))$.

□

Лемма 30. *При $k \geqslant 2$ справедливо соотношение $S_{n,k} \asymp \log n$ при $n \to \infty$.*

Доказательство. Доказательство следует из определения $S_{n,k}$ и леммы 28. □

Лемма 31. *Пусть $Q_1(n,k), Q_2(n,k)$ — классы решетки Поста, причем Q_1 вкладывается в класс Q_2. Пусть $T \in \{MQ, CQ\}$. Пусть $\varphi_T(Q_2,n,k) \leqslant R_2$ и $R_1 \leqslant \varphi_T(Q_1,n,k)$, тогда $\varphi_T(Q_1,n,k) \leqslant R_2$ и $R_1 \leqslant \varphi_T(Q_2,n,k)$.*

Доказательство. Для доказательства леммы достаточно показать, что $\varphi_T(Q_1,n,k) \leqslant \varphi_T(Q_2,n,k)$.

Пусть $\mathcal{A}^T_{Q_i(n,k)}$, где $i \in \{1,2\}$, — множество алгоритмов расшифровки класса $Q_i(n,k)$ запросами типа T.

Заметим, что для всех функций из $Q_1(n,k)$ алгоритм $A \in \mathcal{A}^T_{Q_2(n,k)}$ является также алгоритмом расшифровки из $\mathcal{A}^T_{Q_1(n,k)}$, так как представляет собой последовательность запросов, которые однозначно определяют загаданную функцию $f \in Q_1(n,k)$, причем быть может какие-то запросы ненужные, так как не позволяют уменьшить мощность множества функций-кандидатов на ответ, то есть такие запросы q_i, для которых верно $|W_{i-1}| = |W_i|$, где $W_0 = Q_2(n,k), W_1, W_2, \ldots, W_{\varphi_T(A,f)} = \{f\}$, где $W_i (i > 0)$ — подмножество функций из $Q_2(n,k)$ — кандидатов на ответ после заданных первых i вопросов алгоритмом A.

Учитывая указанное выше соотношение, определение сложности расшифровки и свойство функций $\max, \min$, получаем цепочку неравенств, из которой и следует утверждение леммы.

$$\varphi_T(Q_2,n,k) = \min_{A \in \mathcal{A}^T_{Q_2(n,k)}} \max_{f \in Q_2(n,k)} \varphi_T(A,f) \geqslant \min_{A \in \mathcal{A}^T_{Q_2(n,k)}} \max_{f \in Q_1(n,k)} \varphi_T(A,f) \geqslant$$
$$\geqslant \min_{B \in \mathcal{A}^T_{Q_1(n,k)}} \max_{f \in Q_1(n,k)} \varphi_T(B,f) = \varphi_T(Q_1,n,k).$$

□

Лемма 32. *(мощностная нижняя оценка) Сложность расшифровки булевых функций из произвольного класса $F(n,k)$ мощности N с помощью запросов на значение не меньше $\log_2 N$, то есть $\varphi_{MQ}(F,n,k) \geqslant \log_2 N$.*

Обоснование можно найти, например, в [14].

3.2 Классы C_i

Для класса всех булевых функций C_1 задача сложности расшифровки была исследована еще в работе [22]. Исследование этой задачи для класса функций, сохраняющих 1 (класс C_2) или сохраняющих и 0, и 1 (класс C_4), опирается на решение для класса C_1. Поэтому для всех классов этой группы задача сложности расшифровки сводится к задаче построения бинарных покрывающих матриц.

Утверждение 1. *(P. Damaschke [22]) Справедливы неравенства*

$$\alpha(n,k) \leqslant \varphi_{MQ}(C_1,n,k) \leqslant \alpha(n,k) + k\log_2 n.$$

Следовательно, $\varphi_{MQ}(C_1,n,k) = \alpha(n,k)\cdot(1+o(1))$ *при* $k,n\to\infty$.

Приведем доказательство неравенств из [22], так как понимание алгоритма, используемого для доказательства верхней оценки понадобится далее.

Доказательство. Сначала получим верхнюю оценку. Для этого опросим $\alpha(n,k)$ строк какой-то покрывающей матрицы.

1. Если существуют два набора, на котором загаданная функция принимает разные значения, то перейти к шагу 3, иначе — загаданная функция константа, СТОП.
2. Пусть уже найдено q существенных переменных. Рассмотрим 2^q подфункций загаданной функции, получающихся при всевозможных заменах нулями и единицами уже найденных существенных переменных. Если существует подфункция, которая не константа, тогда возьмем любые два набора, на которых соответствующая подфункция принимает разные значения и перейти к шагу 3. Иначе, СТОП.
3. При помощи алгоритма леммы 24 найдем номер новой существенной переменной, потратив не более $\log_2 n$ запросов. Перейти к шагу 2.

Покажем, что в момент, когда произошел СТОП на шаге 2, найдены все существенные переменные и известен весь вектор значений функции. Пусть есть хотя бы одна ненайденная существенная переменная. По определению существенной переменной, найдутся два набора, отличающиеся только в этой переменной, значения функции на которых будут различаться. Но такие два

набора будут относиться к одной и той же подфункции из 2^q штук, так как уже найденные существенные переменные в этих наборах будут зафиксированы одинаково. Но раз произошел СТОП на шаге 2, все 2^q подфункций были константы, пришли к противоречию. Значит, нет ненайденных существенных переменных. Вектор значений загаданной функции уже известен в силу того, что известны номера существенных переменных и значения на всех фиксациях k переменных из n, в том числе на фиксациях существенных переменных с быть может некоторыми фиктивными переменными, если $q < k$.

Чтобы получить нижнюю оценку, достаточно на все запросы ученика отвечать 1 и тогда он до последнего запроса не поймет, перед ним константа или не константа. Цель учителя тем самым заставить ученика опросить все строки почти покрывающей матрицы. Действительно, если ученик не опросит хотя бы одну фиксацию каких-то k переменных из n, тогда ответам учителя на предыдущие запросы ученика будет удовлетворять как константа 1, так и функция, у которой ровно k существенных переменных, входящих в неопрошенную фиксацию, и у которой в векторе значений ровно один нуль, а именно функция равна нулю на неопрошенной фиксации. Заметим, что у функции, у которой в векторе значений ровно один нуль, все переменные существенные. Могло бы показаться, что при таком выставлении нуля получится не функция с ровно k существенными переменными, а с бОльшим числом существенных переменных. Но если у так заданной функции имеется $(k+1)$-я существенная переменная, то существуют два n-местных набора, отличающие только в этой переменной, на которых значение функции отличаются. Если эта фиксация этих k переменных встретилась в первых $\alpha(n,k) - 1$ запросах учителя, то значение на обоих наборах равно 1. Если эта фиксация является той последней неопрошенной, вошедшей в запрос с номером $\alpha(n,k)$, то значение на обоих наборах равно 0. Поэтому если учитель на последний запрос ученика ответит 0, то образуется функция с ровно k существенными переменными.

Учитывая лемму 26, имеем $\varphi(C_1, n, k) = \alpha(n,k) \cdot (1 + o(1))$ при $k, n \to \infty$. □

Лемма 33. *Имеет место неравенство*

$$\alpha(n,k) - 1 \leqslant \varphi_{MQ}(C_2, n, k) \leqslant \alpha(n,k) + k \log n.$$

Следовательно, $\varphi_{MQ}(C_2, n, k) = \alpha(n,k) \cdot (1 + o(1))$ *при* $k, n \to \infty$.

Доказательство. Верхняя оценка, согласно лемме 31, наследуется от класса $C_1(n,k)$ (утверждения 1).

Для получения нижней оценки на все запросы ученика будем отвечать 1. 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. Если он и задаст запрос, содержащий такую фиксацию, то он все равно получит в ответ 1, то есть он не поймет, именно переменные этой фиксации из одних единиц и есть набор существенных переменных или нет. Следовательно, ученик должен опросить все строки почти покрывающей матрицы типа 1. Учитывая лемму 25, получаем $\alpha(n,k)-1 \leqslant \varphi_{MQ}(C_2,n,k)$. □

Лемма 34. *При $n>1, k>1$ имеет место неравенство*

$$2\alpha(n-1,k-1)-2 \leqslant \varphi_{MQ}(C_4,n,k) \leqslant 2\alpha(n-1,k-1)+k\log n.$$

Следовательно, $\varphi(C_4,n,k)=2\alpha(n-1,k-1)\cdot(1+o(1))$ при $k,n\to\infty$.

Доказательство. Получим сначала верхнюю оценку.

$f(0,\ldots,0)=0, f(1,\ldots,1)=1$. Воспользуемся алгоритмом леммы 24 для нахождения существенной переменной x_j по этим двум наборам. После этого возьмем покрывающую матрицу $M(n-1,k-1)$, ее столбцы будут соответствовать остальным переменным. Опросим строки этой матрицы, считая сначала, что $x_j=0$, затем аналогично проделаем, считая, что $x_j=1$. В итоге будет опрошена вся бинарная покрывающая матрица возможно с неминимальным числом строк $2\alpha(n-1,k-1)$. По этой матрице затратив быть может еще $(k-1)\log n$ запросов для нахождения существенных переменных алгоритмом утверждения 1, мы расшифруем всю функцию.

Получим нижнюю оценку леммы. Играем за учителя, сдадим ученику сразу информацию, что переменная x_1 существенная. На все его запросы $a_1a_2\ldots a_n$ будем отвечать a_1. Покажем, что он будет вынужден опросить строки покрывающей матрицы типа 0 $M_0(n-1,k-1)$, где столбцам матрицы соответствуют переменные $x_2,\ldots,x_n$, а $x_1=0$, и строки покрывающей матрицы типа 1 $M_1(n-1,k-1)$, где столбцам матрицы соответствуют переменные $x_2,\ldots,x_n$, а $x_1=1$.

Для этого покажем, что следующие функции лежат в $C_4(n,k)$.

1. Функция, которую получит ученик, если сделает все $\beta_0(n-1,k-1)+\beta_1(n-1,k-1)$ запросов, равна x_1. Она, очевидно, лежит в $C_4(n,k)$.
2. Пусть ученик задал строго меньше $\beta_0(n-1,k-1)+\beta_1(n-1,k-1)$ запросов. Рассмотрим набор $(a_1,a_2,\ldots,a_n)$, содержащий неопрошенную фиксацию $k-1$ переменных из переменных $x_2,\ldots,x_n$. Хотя бы одна такая фиксация точно найдется, потому что были опрошены не все строки матрицы $M_0(n-1,k-1)$ при $x_1=0$, либо не все строки матрицы $M_1(n-1,k-1)$ при $x_1=1$. Если таких несколько, то в случае $M_0(n,k)$ можно взять любую отличную от фиксации из всех нулей, в случае $M_1(n,k)$ можно взять любую отличную от фиксации из всех единиц. Переменные, вошедшие в эту неопрошенную фиксацию, обозначим за $x_{i_1},x_{i_2},\ldots,x_{i_{k-1}}$ $(1<i_1<i_2<\ldots<i_{k-1})$. Рассмотрим функцию от переменных $x_1,x_2,\ldots,x_n$, у которой множество существенных переменных в точности равно $\{x_1,x_{i_1},x_{i_2},\ldots,x_{i_{k-1}}\}$. Вектор значений этой функции при удалении всех фиктивных переменных имеет следующий вид: значение на наборе $(a_1,a_{i_1},a_{i_2},\ldots,a_{i_{k-1}})$ равно $\overline{a_1}$, а на остальных наборах значение совпадает со значением переменной x_1. Следует проверить, что действительно все переменные $x_1,x_{i_1},x_{i_2},\ldots,x_{i_{k-1}}$ являются существенными. Переменная x_1 существенная, так как на любой паре наборов, отличающихся только в переменной x_1, где ни один набор из пары не равен набору $(a_1,a_{i_1},a_{i_2},\ldots,a_{i_{k-1}})$, значения отличаются в силу задания вектора значений функции. Такая пара наборов существует, так как $n>1$. Для переменной $x_{i_j}, j=\overline{1,k-1}$ в качестве такой пары наборов возьмем наборы $(a_1,a_{i_1},a_{i_2},\ldots,a_{i_{k-1}})$ и $(a_1,a_{i_1},a_{i_2},\ldots,a_{i_{j-1}},\overline{a_{i_j}},a_{i_{j+1}},\ldots,a_{i_{k-1}})$. Значение функции на первом наборе равно $\overline{a_1}$, на втором — a_1. Поэтому переменные $x_{i_1},x_{i_2},\ldots,x_{i_{k-1}}$ существенные.

   Заметим, что набор $(a_1,a_2,\ldots,a_n)$ не может быть $(0,0,\ldots,0,0)$. Докажем от противного. Пусть $(a_1,a_2,\ldots,a_n)=(0,0,\ldots,0,0)$, следовательно, в матрице $M_0(n,k)$ выбрана фиксация из всех нулей, а значит среди всех фиксаций набора $(a_1,a_2,\ldots,a_n)$ она была единственная неопрошенная. А значит, эту строку можно удалить с матрицы $M_0(n,k)$ и получить почти покрывающую матрицу типа 0 с меньшим числом строк, противоречие. Аналогично показывается, что набор $(a_1,a_2,\ldots,a_n)$ не может быть $(1,1,\ldots,1,1)$.

Поэтому значение функции на наборе $(0,\ldots,0)$ равно 0, а наборе $(1,\ldots,1)$ — 1. Значит заданная функция лежит в $C_4(n,k)$ и она не равна x_1 при $n>1$.

Из этого следует, что ученик вынужден задать все $\beta_0(n-1,k-1)+\beta_1(n-1,k-1)$ запросов, потому что до последнего момента он не знает, у загаданной функции ровно одна существенная переменная или больше. □

Лемма 35. *При $k \geqslant 2$ справедливы следующие соотношения.*

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

Доказательство. Доказательство следует из утверждения 1, лемм 33 и 34 и леммы 28. □

3.3 Классы A_i

Вопрос сложности расшифровки разных множеств монотонных булевых функций, в том числе и для класса всех монотонных булевых функций A_1, обстоятельно был изучен в работе [14]. Поскольку остальные классы рассматриваемой группы A_2, A_4 отличаются от A_1 только константами, то результат для класса A_1 легко обобщается и на классы A_2, A_4.

Утверждение 2. *(В. Осокин [14]) Справедливо соотношение*

$$\varphi_{MQ}(A_1,n,k) \asymp \frac{2^k}{\sqrt{k}} + k\log n$$

при $k,n\to\infty$.

Лемма 36. *Справедливы соотношения*

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

Доказательство. Поскольку $A_2(n,k), A_4(n,k)$ — подмножества класса $A_1(n,k)$, то верхняя оценка для них согласно 31 также же, как и для класса $A_1(n,k)$, по порядку не превосходит $\frac{2^k}{\sqrt{k}} + k\log n$.

Нижняя оценка, полученная в [14], для класса $A_1(n,k)$ мощностная. Если покажем, что мощность классов $A_2(n,k), A_4(n,k)$ почти такая же, как и для класса $A_1(n,k)$, то нижняя оценка класса $A_1(n,k)$ будет верна и для классов $A_2(n,k), A_4(n,k)$.

Действительно, класс $A_2(n,k) = A_1(n,k) \setminus \{f(x_1,x_2,\ldots,x_n) \equiv 0\}$, $A_4(n,k) = A_2(n,k) \setminus \{f(x_1,x_2,\ldots,x_n) \equiv 1\}$ □

Лемма 37. *Справедливы соотношения*

- $\varphi_{MQ}(A_1,n,k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{MQ}(A_2,n,k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{MQ}(A_4,n,k) \asymp \log n$ *при* $n \to \infty$.

Доказательство. В доказательстве леммы 3.2 работы [14] приводится нижняя оценка $\varphi_{MQ}(A_1,n,k) \geqslant \log_2 C_n^k + C_k^{[k/2]} - 3$. Далее используется известные соотношения $\left(\frac{n}{k}\right)^k \leqslant C_n^k \leqslant \left(\frac{n\cdot e}{k}\right)^k$ и асимптотическое равенство центральных биномиальных коэффициентов $C_n^k \sim \sqrt{\frac{2}{\pi}}\frac{2^k}{\sqrt{k}}$ при $k \to \infty$, тем самым получается нижняя оценка из утверждения 2. В случае фиксированного k верна следующая цепочка неравенств

$$\varphi_{MQ}(A_1,n,k) \geqslant \log_2 C_n^k + C_k^{[k/2]} - 3 \geqslant k\log_2 n - k\log_2 k + 2^{[k/2]} - 3.$$

В доказательстве леммы 4.4 работы [14] приводится верхняя оценка $\varphi_{MQ}(A_1,n,k) \leqslant 8k\log_2 n + 6C_k^{[k/2]}$. В случае фиксированного k верна следующая цепочка неравенств

$$\varphi_{MQ}(A_1,n,k) \leqslant 8k\log_2 n + 6C_k^{[k/2]} \leqslant 8k\log_2 n + (2e)^{[k/2]}.$$

Соответственно, при $n \to \infty$ верна следующая оценка по порядку

$$\varphi_{MQ}(A_1,n,k) \asymp \log n.$$

Поскольку $A_2(n,k), A_4(n,k)$ — подмножества класса $A_1(n,k)$, то верхняя оценка для них согласно 31 также же, как и для класса $A_1(n,k)$, при $n \to \infty$ по порядку не превосходит $\log n$.

Нижняя оценка, полученная в [14], для класса $A_1(n,k)$ мощностная. Вспоминаем, что класс $A_2(n,k) = A_1(n,k) \setminus \{f(x_1,x_2,\ldots,x_n) \equiv 0\}$, $A_4(n,k) = A_2(n,k) \setminus \{f(x_1,x_2,\ldots,x_n) \equiv 1\}$. Также вспомним, что при $X \geqslant 3$ верно $\log_2(X-1) \geqslant \log_2 X - 1$ и $\log_2(X-2) \geqslant \log_2 X - 1$. Отсюда следует, что оценка

$\varphi_{MQ}(A_1, n, k) \geqslant \log_2 C_n^k + C_k^{[k/2]} - 3$ для классов $A_2(n,k), A_4(n,k)$ преобразуется в следующие.

$$\varphi_{MQ}(A_2, n, k) \geqslant \log_2 C_n^k + C_k^{[k/2]} - 4.$$
$$\varphi_{MQ}(A_4, n, k) \geqslant \log_2 C_n^k + C_k^{[k/2]} - 4.$$

Поэтому для классов $A_2(n,k), A_4(n,k)$ также верны оценки по порядку $\varphi_{MQ}(A_2, n, k) \asymp \log n$ и $\varphi_{MQ}(A_4, n, k) \asymp \log n$ при $n \to \infty$. □

3.4 Классы D_i

С точки зрения особенностей расшифровки группу классов самодвойственных функций можно разбить на две. Задача расшифровки класса всех самодвойственных функций D_3 и класса всех самодвойственных, сохраняющих 0, D_1 чем-то напоминает задачу расшифровки всех булевых функций, а задача расшифровки класса D_2 самодвойственных, монотонных, являющихся подмножеством класса F_2^2, похожа на задачу расшифровки монотонных функций. Поэтому в этом разделе показано два характера результатов: условная асимптотика, как это было свойственно для результатов раздела 3.2, и порядок, как это было свойственно для результатов раздела 3.3.

Лемма 38. *Для $k > 1$ имеет место неравенство*

$$\varphi_{MQ}(D_3, n, k) \geqslant \alpha(n-1, k-1).$$

Доказательство. В качестве отгадываемой функции выберем такую, что у нее подфункция $f(0, x_2, \ldots, x_n)$ — константа 0, подфункция $f(1, x_2, \ldots, x_n)$ — константа 1.

На запрос ученика $(x_1, x_2, \ldots, x_n)$ будем отвечать x_1.

Покажем, что ученик будет вынужден задать при такой системе ответов на его запросы как минимум $\alpha(n-1, k-1)$ запросов.

Рассмотрим все запросы, что задаст ученик. Каждый запрос преобразуем следующим образом:

– если $x_1 = 0$, то оставим запрос без изменений;

– если $x_1 = 1$, то инвертируем все переменные в запросе.

Теперь рассмотрим преобразованные запросы. Пусть различных запросов среди них строго меньше $\alpha(n-1,k-1)$, значит ученик не опросил какую-то фиксацию $k-1$ переменной из переменных $x_2, x_3, \ldots, x_n$. Если бы все фиксации $k-1$ переменных из переменных $x_2, x_3, \ldots, x_n$ были б опрошены, тогда $\alpha(n-1,k-1)$ изначально было не минимально. Значит, ученик действительно одну из фиксаций не опросил. Фиксируем одну такую фиксацию. Рассмотрим любой набор, содержащий ее, мы могли ответить на этот набор в качестве запроса как x_1, так и $\overline{x_1}$. Если бы мы ответили на этот набор-запрос значение переменной x_1, тогда загаданная функция была равна $x_1 \in D_3(n,k)$. Если бы мы ответили $\overline{x_1}$, то все переменные из фиксации и переменная x_1 — существенные, а остальные — фиктивные. Получилась бы самодвойственная функция, у которой при выбрасывании всех фиктивных переменных ($n-k$ штук) вектор значений в первой половине содержит ровно одну единицу, то есть функция из $D_3(n,k)$. Таким образом, не задав последнего $\alpha(n-1,k-1)$-го запроса ученик не знает, какая из двух функций перед ним. □

Лемма 39. *Имеет место неравенство*

$$\varphi_{MQ}(D_3,n,k) \leqslant \alpha(n-1,k-1) + k\log n + 1.$$

Доказательство. Запрашиваем значение на наборе веса 0. В силу самодвойственности f знаем, чему равно ее значение на наборе веса n. Пользуемся алгоритмом леммы 24 для нахождения существенной переменной по двум наборам, на которых значение функции отличается. Соответственно, за $\log_2 n$ найдем одну существенную переменную x_i.

Строим покрывающую матрицу $M(n-1,k-1)$, вставляем в нее i-й столбец из нулей, опрашиваем все строки этой матрицы. Теперь возьмем матрицу, где строки — это инвертированные строки нашей матрицы. Ответы на запросы-строки второй матрицы в силу самодвойственности функции восстанавливаются по ответам по первой матрице. Заметим, что все фиксации переменных $x_1, x_2, \ldots, x_n$ кроме x_i содержатся в первой матрице и во второй матрице. Все фиксации, в которых участвует переменная x_i, если она равна 0, есть в первой матрице, если она равно 1, во второй.

Теперь у нас есть найденная одна существенная переменная и построена какая-то покрывающая матрица с быть может не минимальным числом строк,

далее найдем все остальные существенные переменные алгоритмом утверждения 1.

Итого, задали $1 + k \log n + \alpha(n-1, k-1)$ запросов. □

Лемма 40. *Для любого $k > 1$ имеет место неравенство*

$$\varphi_{MQ}(D_1, n, k) \geqslant \alpha(n-1, k-1) - 1.$$

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

В качестве отгадываемой функции выберем такую, что у нее подфункция $f(0, x_2, \ldots, x_n)$ — константа 0, подфункция $f(1, x_2, \ldots, x_n)$ — константа 1.

На запрос ученика $(x_1, x_2, \ldots, x_n)$ будем отвечать x_1.

Покажем, что ученик будет вынужден задать при такой системе ответов на его запросы как минимум $\beta_0(n-1, k-1)$ запросов.

Рассмотрим все запросы, что задаст ученик. Каждый запрос преобразуем следующим образом:

– если $x_1 = 0$, то оставим запрос без изменений;

– если $x_1 = 1$, то инвертируем все переменные в запросе.

Теперь рассмотрим преобразованные запросы. Пусть различных запросов среди них строго меньше $\beta_0(n-1, k-1)$, значит ученик не опросил какую-то фиксацию $k-1$ переменной из переменных $x_2, x_3, \ldots, x_n$, где не все переменные одновременно равны 0. Если бы все такие фиксации $k-1$ переменных из переменных $x_2, x_3, \ldots, x_n$ были б опрошены, тогда $\beta_0(n-1, k-1)$ изначально было не минимально. Значит, ученик действительно одну из фиксаций не опросил. Фиксируем любую такую фиксацию отличную от фиксации из всех нулей. Рассмотрим любой набор, содержащий ее, мы могли ответить на этот набор в качестве запроса как x_1, так и $\overline{x_1}$. Если бы мы ответили на этот набор-запрос значение переменной x_1, тогда загаданная функция была равна $x_1 \in D_1(n, k)$. Если бы мы ответили $\overline{x_1}$, то все переменные из фиксации и переменная x_1 — существенные, а остальные — фиктивные. Получилась бы самодвойственная функция, у которой при выбрасывании всех фиктивных переменных ($n-k$ штук) вектор значений в первой половине содержит ровно одну единицу, то есть функция из $D_1(n, k)$. Таким образом, не задав последнего $\beta_0(n-1, k-1)$-го запроса ученик не знает, какая из двух функций перед ним.

Следовательно, ученик вынужден опросить $\beta_0(n-1, k-1)$ запросов, что согласно лемме 25, не меньше $\alpha(n-1, k-1) - 1$.

□

Лемма 41. *Имеет место неравенство*

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

при $k, n \to \infty$, $k < n$.

Доказательство. Посчитаем число функций из D_2 арности k_2, где k_2 существенных переменных и k_2 — самое большое четное число, не превосходящее k. Зададим значение функции на вершинах среднего слоя с номером $k_2/2$ в k_2-мерном булевом кубе. Если задаем значение функции на наборе X этого слоя, то противоположный набор лежит в этом же слое. Поэтому если каким-то образом определим значение на половине вершин среднего слоя (попарно непротивоположных), то весь слой будет задан. Значение функции на наборах выше среднего слоя положим равным 1. Противоположные к ним наборы лежат ниже среднего слоя, значение функции на них положим равным 0. Полученная функция самодвойственная, монотонная, сохраняет 0 и 1. Проверим, что полученная функция лежит в F_2^2, для этого проверим для каждой пары наборов, на которых функция обращается в нуль, имеют ли они общую нулевую компоненту.

1. Возьмем любые два набора a, b, такие что либо они оба ниже среднего слоя, либо один ниже среднего слоя, а второй — со среднего слоя. Вес набора $a \vee b$ строго меньше k_2. Отсюда следует, что у наборов a, b есть хотя бы одна общая нулевая компонента.
2. Возьмем любые два набора a, b со среднего слоя, на которых функция равна 0. Вес каждого в точности равен $k_2/2$. Вес набора $a \vee b$ будет равен k_2 лишь в случае, когда a и b противоположные наборы, но тогда значение на одном из них равно 1, чего не может быть в силу выбора a, b. Значит, вес набора $a \vee b$ строго меньше k_2 и поэтому у наборов a, b имеется общая нулевая компонента.

Теперь посчитаем число способов верным образом задать значение функции на половине вершин среднего слоя, чтобы получилась функция, существенно зависящая от всех своих переменных. Сделаем это аналогично доказательству леммы 3.2 из [14].

Пусть a, b, c — три произвольных набора из слоя $k_2/2$ k_2-мерного булевого куба таких, что их поэлементная дизъюнкция дает в точности единичный набор. Заметим, что такая тройка наборов существует, можно, например, взять наборы следующие

$111\ldots1100\ldots000$

$000\ldots0111\ldots110$

$000\ldots0111\ldots101$

Причем они попарно не являются противоположными, то есть входят в ту половину серединного слоя, что мы собираемся задавать.

Рассмотрим множество функций A из D_2 от k_2 переменных, для которых эти a, b, c являются нижними единицами. Чтобы переменная была существенной для монотонной функции надо, чтобы она входила хотя бы в одну конъюнкцию минимальной ДНФ этой функции. Конъюнкции, соответствующие наборам a, b, c, содержат все k_2 переменных, следовательно все переменные существенные. Значит число функций в D_2 от k_2 переменных, где все переменные существенные, не меньше чем мощность A, которая равна $2^{\frac{1}{2}\cdot C_{k_2}^{k_2/2}-3}$, то есть каждой вершине среднего слоя из той половины, что собираемся задавать, кроме наборов a, b, c, можно присвоить любое из двух значений.

Следовательно, всего функций от n переменных, не более k из которых существенные, в D_2 не меньше, чем $C_n^{k_2} \cdot 2^{\frac{1}{2}\cdot C_{k_2}^{k_2/2}-3}$. Пользуемся известным неравенством: $\left(\frac{n}{k}\right)^k \leqslant C_n^k$.

Получаем мощностную нижнюю оценку

$$\varphi_{MQ}(D_2, n, k) \geqslant \log_2\left(C_n^{k_2} \cdot 2^{\frac{1}{2}C_{k_2}^{k_2/2}-3}\right) = \log_2 C_n^{k_2} + \frac{1}{2}\cdot C_{k_2}^{[k_2/2]} - 3 \geqslant$$
$$\geqslant k_2 \log_2(n/k_2) + \frac{1}{2}\cdot C_{k_2}^{k_2/2} - 3.$$

Аналогично [14] воспользуемся асимптотическим равенством для центральных биномиальных коэффициентов $C_k^{k/2} \sim \sqrt{\frac{2}{\pi}} \cdot \frac{2^k}{\sqrt{k}}$ при $k \to \infty$ для получения оценки доказываемой леммы. □

Лемма 42. *Имеют место следующие соотношения.*

1. $\varphi_{MQ}(D_3, n, k) = \alpha(n-1, k-1)\cdot(1+o(1))$ *при* $k, n \to \infty, k = o(n)$.
2. $\varphi_{MQ}(D_1, n, k) = \alpha(n-1, k-1)\cdot(1+o(1))$ *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. Соотношение, связанное с классом D_3, следует из лемм 38, 39, 27 при $p = 1$.

Соотношение, связанное с классом D_1, следует из лемм 40, 39, 27 при $p = 1$.

Соотношение, связанное с классом D_2, следует из леммы 41 и утверждения 2.

□

Лемма 43. *Верны следующие соотношения.*

1. $\varphi_{MQ}(D_1, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$.
2. $\varphi_{MQ}(D_2, n, k) \asymp \log n$ *при* $n \to \infty$.
3. $\varphi_{MQ}(D_3, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$.

Доказательство. Доказательство следует из лемм 42 и 28. □

3.5 Классы $F_j^i, 1 \leqslant j \leqslant 4$

Заметим, что подобно группе классов D_i, в рассматриваемой группе классов также имеется множество классов монотонных функций F_3^i, F_2^i и множество классов необязательно монотонных функций F_4^i, F_1^i. Поэтому результаты получаемые для классов "счетной этажерки" по характеру бывают двух видов: условная асимптотика для F_4^i, F_1^i и порядок для F_3^i, F_2^i.

Прежде чем перейти к доказательствам результатов этого раздела, заметим следующее свойство всех классов "счетной этажерки".

Замечание 2. *Ученику заведомо известно, что на наборе из всех единиц значение функции из этих классов равно 1, так как они также лежат в классе C_2. Более того, если на каком-то наборе значение равно 0, то на противоположном значение равно 1, поскольку противоположные наборы не имеют общей нулевой компоненты.*

Лемма 44. *Имеет место неравенство* $\varphi_{MQ}(F_4^i, n, k) \geqslant \alpha(n, k) - 1$.

Доказательство. В качестве загаданной функции выберем константу 1. Тогда ученик будет вынужден опросить все строки почти покрывающей матрицы типа 1. Потому что если ученик не опросит хотя бы одну строку, будет не опрошена хотя бы одна фиксация. На этот запрос учитель мог ответить как 0, так и 1.

Если бы учитель ответил 0, то ученик бы понял, что одна из опрошенных этим запросом фиксаций и содержит все номера существенных переменных. □

Лемма 45. *Справедливо неравенство* $\varphi_{MQ}(F_1^i, n, k) \leqslant S_{n,k} + k \log n + k$.

Доказательство. Найдем первую существенную переменную по наборам $(0,\ldots,0)$ и $(1,\ldots,1)$ алгоритмом леммы 24, за не более $\log n$ запросов. Далее следуем к шагу 1.

1. Положим $i = 1$. Перейти к шагу 2.
2. Запросить значение на наборе веса $n-i$, где уже найденные существенные переменные зафиксированы 0. Если оно равно 1, то перейти к шагу 3, иначе перейти к 4.
3. Алгоритмом леммы 24 по набору из предыдущего шага и набору $(0,0,\ldots,0,0)$ находим номер новой существенной переменной. Меняем $i = i + 1$. Перейти к шагу 2.
4. Пусть найденные существенные переменные имеют номера $n_1, n_2, \ldots, n_i$. Тогда раз на опрошенном только что наборе функция равна 0, то на любом наборе, где $x_{n_1} = x_{n_2} = \ldots = x_{n_i} = 1$, функция равна 1 из-за свойств функций класса F_1^i. Поэтому мы не тратим запросы на опрос строк матрицы $M(n-i, k-i)$ на переменных с номерами $\{1,2,\ldots,n\} \setminus \{n_1, n_2, \ldots, n_i\}$. СТОП.

Допустим после этого нашли p существенных переменных. Пусть номера найденных существенных переменных $j_1 < j_2 < \ldots < j_p$. Если $p = k$, то останется опросить значение функции на $2^k - 2 - k$ наборах и тогда полностью расшифруем загаданную функцию. Минус 2 делаем потому, что значение на наборе из всех единиц и на наборе из всех 0, значение функции известно. Минус k, потому что их уже опросили в предыдущем алгоритме.

Иначе, сделаем запрос по строкам следующих $2^p - 1$ матриц.

Q-я матрица $(0 \leqslant Q < 2^p - 1)$ строится так:

– Берем любую покрывающую матрицу $M(n-p, k-p)$. Столбцы этой матрицы будут соответствовать переменным с номерами $\{1,2,\ldots,n\} \setminus \{j_1, \ldots, j_p\}$
– Значение найденных существенных переменных будет соответствовать битам двоичного разложения номера Q.

Опросим все такие матрицы.

После этого получится, что мы опросили все фиксации из n столбцов по k. По этой информации согласно алгоритму, описанному в утверждении 1, мы сможем полностью восстановить всю функцию, затратив по не более $\log n$ запросов на нахождение каждой из оставшихся существенных переменных.

Итого оценим количество запросов

– $p \neq k$

$$p + (2^p - 1)\alpha(n-p, k-p) + k\log n \leqslant k + S_{n,k} + k\log n.$$

– $p = k$

$$\alpha(n-k+1, 1) = 2.$$

$$k + (2^k - 2 - k) + k\log n \leqslant (2^k - 2) + k\log n \leqslant$$
$$\leqslant (2^{k-1} - 1)\alpha(n-k+1, 1) + k\log n \leqslant S_{n,k} + k\log n.$$

□

Лемма 46. *Пусть $i > 1, k > 1, 1 \leqslant p < k$, p — целое. Тогда справедливо неравенство*

$$\varphi_{MQ}(F_1^i, n, k) \geqslant (2^p - 1)\alpha(n-p, k-p).$$

Доказательство. Для получения этой оценки рассмотрим такую f. Переменные $x_1, x_2, \ldots, x_p$ объявим существенными, а есть ли еще существенные переменные или нет определимся позже. Если ученик подаст запрос $a = (a_1, a_2, \ldots, a_p, a_{p+1}, \ldots, a_n)$, то будем различать следуюшие три категории запросов:

1. $a_1 = a_2 = \ldots = a_p = 0$
2. $a_1 = a_2 = \ldots = a_p = 1$
3. остальные наборы

Будем считать, 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 $x_1, x_2, \ldots, x_p$.

Отвечать на запросы будем следующим образом.

1. Если $a_1 = a_2 = \ldots = a_p = 0$
   а) Если опрошены все группы наборов 3-й категории, то
      1) Если после ответа на текущий запрос, будет опрошена группа первой категории, отвечаем 1.

2) Иначе, отвечаем 0.

б) Иначе, отвечаем 0.

2. Если $a_1 = a_2 = \ldots = a_p = 1$, будем отвечать 1.

3. а) Если после ответа на текущий запрос, будут опрошены все группы запросов 3-й категории,

1) если опрошена вся группа первой категории, отвечаем 0,

2) иначе, отвечаем 1.

б) Иначе, отвечаем 1.

Заметим, что если бы на запросы группы первой категории мы всегда б отвечали 0, а на запросы групп остальных категорий 1, то это получилась в точности функция с ровно p существенными переменными $x_1|x_2|\ldots|x_p$, так как все подфункции, соответствующие зафиксированным наборам p первых переменных, являлись константами. Эта функция принадлежит $F_1^\infty(n,k)$, так как все наборы, на которых функция обращается в 0 имеют общую нулевую компоненту x_1.

Но мы в самый последний момент, когда опрошены все запросы первой и третьей категорий, отвечаем иначе, и получаем иную функцию. После ответа на этот запрос становятся опрошенными все фиксации по k переменных, куда входят переменные с номерами $x_1, \ldots, x_p$, значит любая из фиксаций, которая была наконец опрошена этим запросом, и определяет набор существенных переменных, а их в фиксации ровно k штук. Зафиксируем одну из таких фиксаций и переменные, помимо переменных $x_1, \ldots, x_p$, вошедшие в нее, объявим существенными, обозначим их $x_{i_{p+1}}, x_{i_{p+2}}, \ldots, x_{i_k}$, где $p < i_{p+1} < i_{p+2} < \ldots < i_k$.

Не нарушая общности будем считать, что в зафиксированной фиксации переменным $x_{i_{p+1}}, x_{i_{p+2}}, \ldots, x_{i_q}$ присвоено значение 1, а переменным $x_{i_{q+1}}, x_{i_{q+2}}, \ldots, x_{i_k}$ — значение 0.

Также не нарушая общности будем считать, что в зафиксированной фиксации переменным $x_1, x_2, \ldots, x_t$ присвоено значение 1, а переменным $x_{t+1}, x_{t+2}, \ldots, x_p$ — значение 0.

Покажем, что получившаяся функция лежит в $F_1^\infty(n,k)$.

1. Пусть сработал случай 1.(а).i. Тогда загаданная функция имеет вид

$$x_1|x_2|\ldots|x_p|(x_{i_{p+1}}\&x_{i_{p+2}}\&\ldots x_{i_q}\&\overline{x_{i_{q+1}}}\&\overline{x_{i_{q+2}}}\&\ldots\&\overline{x_{i_k}}).$$

Очевидно, что все переменные $x_1, x_2, \ldots, x_p, x_{i_{p+1}}, x_{i_{p+2}}, \ldots, x_{i_k}$ существенные.

В этом случае $f \in F_1^\infty(n,k)$, так как все наборы, на которых f равна 0, имеют общую нулевую компоненту x_1.

2. Пусть сработал случай 3.(а).i. Тогда загаданная функция имеет вид

$$(x_1|x_2|\ldots|x_p)\&(\overline{x_1}|\overline{x_2}|\ldots|\overline{x_t}|x_{t+1}|x_{t+2}|\ldots| \\ |x_p|\overline{x_{i_{p+1}}}|\overline{x_{i_{p+2}}}|\ldots|\overline{x_{i_q}}|x_{i_{q+1}}|x_{i_{q+2}}|\ldots|x_{i_k}).$$

Необходимо показать, что все переменные $x_1, x_2, \ldots, x_p, x_{i_{p+1}}, x_{i_{p+2}}, \ldots, x_{i_k}$ существенные. Рассмотрим переменную x_j, где $j \in \{1, 2, \ldots, p, i_{p+1}, i_{p+2}, \ldots, i_k\}$.

- Если в последнем запросе $(a_1, a_2, \ldots, a_n)$ $a_j = 0$, то переменная x_j — существенная, так как на наборе $(a_1, a_2, \ldots, a_n)$ функция равна 0, а на наборе $(a_1, a_2, \ldots, a_{j-1}, \overline{a_j}, a_{j+1}, \ldots, a_n)$ — 1.
- Если в последнем запросе $(a_1, a_2, \ldots, a_n)$ $a_j = 1$, то
  - если $j \in \{i_{p+1}, i_{p+2}, \ldots, i_k\}$, то число ненулевых компонент в наборе $(a_1, a_2, \ldots, a_n)$ строго больше 1, так как есть еще хотя бы одна единица среди значений переменных $x_1, x_2, \ldots, x_p$. Поэтому в качестве пары наборов, отличающихся только в переменной x_j, можно взять наборы $(a_1, a_2, \ldots, a_n)$ и $(a_1, a_2, \ldots, a_{j-1}, \overline{a_j}, a_{j+1}, \ldots, a_n)$. На первом наборе значение функции равно 0, на втором — 1. Следовательно, переменная x_j — существенная.
  - если $j \in \{1, 2, \ldots, p\}$ и среди чисел $a_1, a_2, \ldots, a_p$ строго больше одной единицы, то в качестве пары соседних наборов, отличающихся в переменной x_j, возьмем наборы $(a_1, a_2, \ldots, a_n)$ и $(a_1, a_2, \ldots, a_{j-1}, \overline{a_j}, a_{j+1}, \ldots, a_n)$. На первом наборе значение функции равно 0, на втором — 1. Следовательно, переменная x_j — существенная.
  - если $j \in \{1, 2, \ldots, p\}$ и среди чисел $a_1, a_2, \ldots, a_p$ ровно одна единица, тогда в качестве пары соседних наборов, отличающихся в переменной x_j, возьмем наборы $(a_1, a_2, \ldots, a_p, \overline{a_{p+1}}, \overline{a_{p+2}}, \ldots, \overline{a_n})$ и $(a_1, a_2, \ldots, a_{j-1}, \overline{a_j}, a_{j+1}, \ldots, a_p, \overline{a_{p+1}}, \overline{a_{p+2}}, \ldots, \overline{a_n})$. На

первом наборе значение функции равно 1, на втором — 0. Следовательно, переменная x_j — существенная.

Заметим, что в последнем запросе не все $a_1, a_2, \ldots, a_p$ одновременно равны 1. Пусть отлична от единицы $a_q, 1 \leqslant q \leqslant p$. Тогда все наборы, на которых f обращается в 0, имеют общую нулевую компоненту x_q. Следовательно $f \in F_1^\infty(n,k)$.

Заметим, что мы могли ученику подсказать, что значение на всех наборах группы второй категории равно 1, и он мог не задавать в принципе эти запросы, но все запросы групп первой и третьей категории мы его заставили задать, потому что не задав хотя бы один из этих запросов, у него нет уверенности, p или больше существенных переменных у загаданной функции.

Итого, ученик вынужден задать как минимум $(2^p-1)\alpha(n-p,k-p)$ запросов. □

Лемма 47. *Имеет место неравенство*

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

при $k, n \to \infty$, $k < n$.

Доказательство. Оценим снизу количество функций из F_2^∞ арности k, где все переменные существенные.

Рассмотрим функции f, обладающие обоими свойствами:

1. $f(0, x_2, \ldots, x_k)$ — монотонная функция арности $k-1$, у которой все переменные существенные.
2. $f(1, x_2, \ldots, x_k) \equiv 1$.

Заметим, что каждая такая функция f монотонная и лежит в F_4^∞, так как у всех наборов, на которых функция может обратиться в 0, имеется общая нулевая компонента x_1.

Соответственно, надо оценить число монотонных функций арности $k-1$, у которых все переменные существенные. Это было сделано в доказательстве леммы 3.2 работы [14]. Таких функций не меньше, чем $2^{C_{k-1}^{[(k-1)/2]}-3}$.

Значит, всего функций в $F_2^\infty(n,k)$ не меньше, чем $C_n^k \cdot 2^{C_{k-1}^{[(k-1)/2]}-3}$. Проделываем шаги леммы 41 и получаем нижнюю мощностную оценку из утверждения леммы. □

Лемма 48. *Справедливы соотношения*

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

Доказательство. Соотношение, касающееся класса $F_4^i(n,k)$, следует из леммы 44, верхней оценки утверждения 1, леммы 26.

Соотношение, касающееся класса $F_1^i(n,k)$, следует из лемм 46, 45, 27, 29.

Соотношения, касающееся классов $F_2^i(n,k), F_3^i(\mathrm{n},\ \mathrm{k})$, по лемме 31 следуют из верхней оценки утверждения 2 и леммы 47, потому что класс $F_2^\infty(n,k)$ вкладывается в классы $F_2^i(n,k), F_3^i(n,k)$. □

Лемма 49. *Справедливы соотношения*
– *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_1^i, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
– *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_2^i, n, k) \asymp \log n$ *при* $n \to \infty$;
– *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_3^i, n, k) \asymp \log n$ *при* $n \to \infty$;
– *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_4^i, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$.

Доказательство. Доказательство первого пункта следует из леммы 46, 45 и леммы 30. Доказательство второго и третьего пунктов следует из верхней оценки, получаемой при доказательстве утверждения 2, и леммы 47. Доказательство последнего пункта следует из леммы 44, верхней оценки утверждения 1 и леммы 28. □

3.6 Классы S_i

Асимптотика сложности расшифровки класса всех логических сумм S_1 была получена в работе [39]. Результаты сложности расшифровки для остальных классов (S_3, S_5) рассматриваемой группы получаются адаптацией результата класса S_1.

Утверждение 3. *(Uehara, Tsuchida, Wegener [39]) Справедливы неравенства*

$$\left]\log_2 C_n^k\right[\leqslant \varphi_{MQ}(S_1, n, k) \leqslant k\left]\log\left(n/k\right)\right[+ 2k - 2.$$

Следовательно, $\varphi_{MQ}(S_1, n, k) \sim k \log(n/k), n, k \to \infty, k = o(n)$.

Лемма 50. *Справедливы соотношения*

- $\varphi_{MQ}(S_3, n, k) \sim k \log_2(n/k), n, k \to \infty, k = o(n)$;
- $\varphi_{MQ}(S_6, n, k) \sim k \log_2(n/k), n, k \to \infty, k = o(n)$.

Доказательство. Получим верхнюю оценку сложности расшифровки для класса $S_3(n, k)$. Запросив значение на наборе $(0,\ldots,0)$, можно понять $f(x_1, \ldots, x_n) \equiv 1$ или нет. Если нет, то разгадываем обычную функцию из $S_1(n, k)$. Таким образом, получаем верхнюю оценку $1 + k]\log(n/k)[+2k - 2$. Нижняя оценка мощностная $\rceil \log_2(C_n^k + 1)\lfloor$. Отсюда и вытекает первый пункт леммы.

Получим верхнюю оценку сложности расшифровки для класса $S_6(n, k)$. Запросив значение на наборе $(0,\ldots,0)$, можно понять $f(x_1, \ldots, x_n) \equiv 1$ или нет. Если нет, запросив значение на наборе $(1, \ldots, 1)$, можно понять $f(x_1, \ldots, x_n) \equiv 0$ или нет. Если нет, то разгадываем обычную функцию из $S_1(n, k)$. Таким образом, получаем верхнюю оценку $2 + k\rceil \log(n/k)\lfloor + 2k - 2$. Нижняя оценка мощностная $\rceil \log_2(C_n^k + 2)\lfloor$. Отсюда и вытекает второй пункт леммы.

□

Лемма 51. *Справедливы соотношения*

- $\varphi_{MQ}(S_1, n, k) \asymp \log(n), n \to \infty$;
- $\varphi_{MQ}(S_3, n, k) \asymp \log(n), n \to \infty$;
- $\varphi_{MQ}(S_6, n, k) \asymp \log(n), n \to \infty$.

Доказательство. Доказательство следует из утверждения 3 и леммы 50. □

3.7 Классы L_i

Основное продвижение в задаче сложности расшифровки класса линейных функций было сделано в работе [29], когда была получена асимптотика сложности расшифровки линейных функций, сохраняющих 0 (класс L_3). Результаты для остальных классов этой группы: всех линейных функций (класс L_1), линейных функций, сохраняющих 0 (класс L_2), самодвойственных (класс L_5), самодвойственных и сохраняющих 0 и 1 (класс L_4) — несложно получаются из результата для класса L_3.

Утверждение 4. *(Hofmeister [29]) Справедливо неравенство* $\varphi_{MQ}(L_3, n, k) \leqslant k \log_2 n + k$.

Утверждение 5. *(Uehara, Tsuchida, Wegener [39]) Справедливо неравенство*

$$\varphi_{MQ}(L_3, n, k) \geqslant \rceil \log_2 C_n^k \lceil.$$

Как следствие утверждений 4, 5, получаем следующее утверждение.

Утверждение 6. *(Uehara, Tsuchida, Wegener [39], Hofmeister [29]) Справедливо соотношение*

$$\varphi_{MQ}(L_3, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n).$$

Лемма 52. *Справедливы соотношения*

- $\varphi_{MQ}(L_1, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
- $\varphi_{MQ}(L_2, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
- $\varphi_{MQ}(L_4, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
- $\varphi_{MQ}(L_5, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$.

Доказательство. Получим верхнюю оценку сложности расшифровки для класса $L_1(n, k)$. Задав один запрос $(0, \ldots, 0)$, определим свободный член c_0 загаданной функции $f(x_1, \ldots, x_n) = x_{i_1} \oplus \ldots \oplus x_{i_k} \oplus c_0$. Если $c_0 = 0$, то разгадываем функцию из класса $L_3(n, k)$, иначе разгадываем $\overline{f}$ из класса $L_3(n, k)$. Итого, на расшифровку функции потратим не более $1 + k \log_2 n + k$ запросов.

Для получения нижней оценки заметим, что ученик первым же запросом может узнать значение свободного члена. Поэтому нижняя оценка практически совпадает с нижней оценкой для класса $L_3(n, k)$: $\varphi_{MQ}(L_1, n, k) \geqslant 1 + \rceil \log_2 C_n^k \lceil$.

В качестве верхней оценки сложности расшифровки функции из класса $L_4(n, k)$ по лемме 31 можно взять верхнюю оценку сложности расшифровки функции из класса $L_3(n, k)$, так как класс $L_4(n, k)$ вкладывается в класс $L_3(n, k)$. В качестве нижней оценки возьмем мощностную оценку. В классе $L_4(n, k)$ лежат линейные функции с нулевым свободным членом с нечетным числом существенных переменных. Положим k_o — наибольшее нечетное число, которое не больше, чем k. Тогда получим нижнюю мощностную оценку: $\varphi_{MQ}(L_4, n, k) \geqslant \rceil \log_2 C_n^{k_o} \lceil$.

Сложность расшифровки классов $L_2(n, k), L_5(n, k)$ асимптотически по лемме 31 равна сложности расшифровки классов $L_1(n, k)$ и $L_4(n, k)$ в силу того, что они вкладываются в $L_1(n, k)$ и содержат $L_4(n, k)$.

□

Лемма 53. *Справедливы соотношения*

- $\varphi_{MQ}(L_1, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_2, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_3, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_4, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_5, n, k) \asymp \log n, n \to \infty$.

Доказательство. Доказательство следует из утверждений 4 и 5 и леммы 52. □

3.8 Классы O_i

Данный раздел посвящен расшифровке самых маленьких замкнутых классов решетки Поста. Для класса селекторов в работе [39] получена точная оценка сложности расшифровки. Для остальных классов этой группы приводятся нижние и верхние оценки, отличающиеся на небольшую константу, которые в итоге позволяют получить асимптотику сложности расшифровки.

Утверждение 7. *(Uehara, Tsuchida, Wegener [39]) При $n > 1$ справедливо неравенство*

$$\varphi_{MQ}(O_1, n, 1) =]\log_2 n[.$$

Лемма 54. *При $n > 1$ справедливы соотношения*

- $\varphi_{MQ}(O_4, n, 1) =]\log_2 n[$;
- $\varphi_{MQ}(O_5, n, 1) \sim \log_2 n, n \to \infty$;
- $\varphi_{MQ}(O_6, n, 1) \sim \log_2 n, n \to \infty$;
- $\varphi_{MQ}(O_8, n, 1) \sim \log_2 n, n \to \infty$;
- $\varphi_{MQ}(O_9, n, 1) \sim \log_2 n, n \to \infty$.

Доказательство. Рассмотрим класс $O_4(n, 1)$. Задав запрос $(0,\ldots,0)$, определим $f(x_1, \ldots, x_n) \equiv x_i$ или $f(x_1, \ldots, x_n) \equiv \overline{x_i}$. Затем, согласно утверждению 7, потратим не более $]\log_2 n[$ запросов на нахождение номера существенной переменной. Таким образом, верхняя оценка равна $1+]\log_2 n[$ запросов. Нижняя оценка мощностная: $]\log_2 2n[=]1 + \log_2 n[= 1+]\log_2 n[$.

Рассмотрим класс $O_6(n,1)$. Задав запрос $(0,\ldots,0)$, определим $f(x_1,\ldots,x_n) \equiv 1$ или нет. Если нет, зададим запрос $(1,\ldots,1)$, определим $f(x_1,\ldots,x_n) \equiv 0$ или нет. Если нет, то, согласно утверждению 7, потратим не более $]\log_2 n[$ запросов на нахождение номера существенной переменной. Таким образом, верхняя оценка равна $2+]\log_2 n[$ запросов. Нижняя оценка мощностная: $\left]\log_2(2+n)\right[$.

Сложность расшифровки классов $O_5(n,1), O_8(n,1)$ по лемме 31 асимптотически равна сложности расшифровки классов $O_1(n,1)$ и $O_6(n,1)$ в силу того, что $O_5(n,1), O_8(n,1)$ вкладываются в $O_6(n,1)$ и содержат $O_1(n,1)$.

Рассмотрим класс $O_9(n,1)$. Зададим запросы $(0,\ldots,0)$ и $(1,\ldots,1)$.

- Если $f(0,\ldots,0)=0, f(1,\ldots,1)=0$, то $f(x_1,\ldots,x_n)\equiv 0$.
- Если $f(0,\ldots,0)=1, f(1,\ldots,1)=1$, то $f(x_1,\ldots,x_n)\equiv 1$.
- Если $f(0,\ldots,0)=0, f(1,\ldots,1)=1$, то $f(x_1,\ldots,x_n)=x_i$. Согласно утверждению 7, потратим не более $]\log_2 n[$ запросов на нахождение номера существенной переменной.
- Если $f(0,\ldots,0)=1, f(1,\ldots,1)=0$, то $f(x_1,\ldots,x_n)=\overline{x_i}$. Согласно утверждению 7, потратим не более $]\log_2 n[$ запросов на нахождение номера существенной переменной.

Следовательно, верхняя оценка $2+]\log_2 n[$. Нижняя оценка — мощностная: $\left]\log_2(2+2n)\right[$.

□

3.9 Теорема о сложности расшифровки для всех классов Поста

Для облегчения понимания общей картины результатов точной расшифровки замкнутых классов решетки Поста запросами на значение перейдем к теоремам, в которой собраны результаты утверждений 1, 2, 3, 6 и лемм 33, 34, 36, 42, 48, 50, 52, 54. В теореме 12 собраны результаты для случая $n,k\to\infty$, в теореме 13 — для случая $n\to\infty$ с фиксированным k.

Напомним, что под условной асимптотической оценкой будем понимать то, что оценка асимптотически равна величине, связанной с $\alpha(n,k)$, асимптотику которой мы не знаем.

Теорема 12. *Замкнутые классы решетки Поста по характеру известной на данный момент сложности точной расшифровки запросами на значение разделены на четыре группы в случае* $n, k \to \infty$*:*

1. *точная оценка*
   - $\varphi_{MQ}(O_4, n, 1) = \left]\, \log_2 n \,\right[$ *при* $n > 1$;
2. *асимптотика*
   - $\varphi_{MQ}(S_1, n, k) \sim k \log_2 (n/k), n, k \to \infty, k = o(n)$;
   - $\varphi_{MQ}(S_3, n, k) \sim k \log_2 (n/k), n, k \to \infty, k = o(n)$;
   - $\varphi_{MQ}(S_6, n, k) \sim k \log_2 (n/k), n, k \to \infty, k = o(n)$;
   - $\varphi_{MQ}(L_1, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
   - $\varphi_{MQ}(L_2, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
   - $\varphi_{MQ}(L_3, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
   - $\varphi_{MQ}(L_4, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
   - $\varphi_{MQ}(L_5, n, k) \sim k \log_2 n, n, k \to \infty, \log_2 k = o(\log_2 n)$;
   - $\varphi_{MQ}(O_5, n, 1) \sim \log_2 n, n \to \infty$;
   - $\varphi_{MQ}(O_6, n, 1) \sim \log_2 n, n \to \infty$;
   - $\varphi_{MQ}(O_8, n, 1) \sim \log_2 n, n \to \infty$;
   - $\varphi_{MQ}(O_9, n, 1) \sim \log_2 n, n \to \infty$;
3. *условная асимптотика*
   - $\varphi_{MQ}(C_1, n, k) = \alpha(n, k) \cdot (1 + o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - $\varphi_{MQ}(C_2, n, k) = \alpha(n, k) \cdot (1 + o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - $\varphi_{MQ}(C_4, n, k) = 2\alpha(n-1, k-1) \cdot (1 + o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - $\varphi_{MQ}(D_3, n, k) = \alpha(n-1, k-1) \cdot (1+o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - $\varphi_{MQ}(D_1, n, k) = \alpha(n-1, k-1) \cdot (1+o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_4^i, n, k) = \alpha(n, k) \cdot (1 + o(1))$ *при* $k, n \to \infty, k = o(n)$;
   - *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_1^i, n, k) = S_{n,k} \cdot (1 + o(1))$ *при* $k, n \to \infty, k = o(n)$;
4. *порядок*
   - $\varphi_{MQ}(A_1, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $k, n \to \infty$;
   - $\varphi_{MQ}(A_2, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $k, n \to \infty$;
   - $\varphi_{MQ}(A_4, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $k, n \to \infty$;
   - $\varphi_{MQ}(D_2, n, k) \asymp \frac{2^k}{\sqrt{k}} + k \log n$ *при* $k, n \to \infty$;

– *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_2^i, n, k) \asymp k \log n + \frac{2^k}{\sqrt{k}}$ *при* $k, n \to \infty$;
– *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_3^i, n, k) \asymp k \log n + \frac{2^k}{\sqrt{k}}$ *при* $k, n \to \infty$.

Доказательство. Теорема является объединением результатов утверждений 1, 2, 3, 6, 7 и лемм 33, 34, 36, 42, 48, 50, 52, 54, а также лемм 26, 29. □

На рисунке 1 схематично приведены основные результаты данной главы. Если класс выделен вертикальной чертой в квадрате или треугольнике, то имеется точная оценка. Если класс выделен белым, то для него получена асимптотическая оценка, если обозначен косой линией внутри квадрата или треугольника, то оценка по порядку, если черным, то условная асимптотическая оценка. Если класс выделен треугольником, то это результат, полученный ранее в других работах. Если класс выделен квадратом, то результат ранее в литературе не встречался и впервые упоминается в данной работе.

Теорема 13. *Замкнутые классы решетки Поста по характеру известной на данный момент сложности точной расшифровки запросами на значение в случае, когда* n *растет, а* k *не меняется, разделены на три группы:*

1. *точная оценка*
   – $\varphi_{MQ}(O_4, n, 1) = \rceil \log_2 n \lfloor$ *при* $n > 1$;
2. *асимптотика*
   – $\varphi_{MQ}(O_5, n, 1) \sim \log_2 n, n \to \infty$;
   – $\varphi_{MQ}(O_6, n, 1) \sim \log_2 n, n \to \infty$;
   – $\varphi_{MQ}(O_8, n, 1) \sim \log_2 n, n \to \infty$;
   – $\varphi_{MQ}(O_9, n, 1) \sim \log_2 n, n \to \infty$;
3. *порядок*
   – $\varphi_{MQ}(C_1, n, k) \asymp \log n, n \to \infty, k \geqslant 2$;
   – $\varphi_{MQ}(C_2, n, k) \asymp \log n, n \to \infty, k \geqslant 2$;
   – $\varphi_{MQ}(C_4, n, k) \asymp \log n, n \to \infty, k \geqslant 2$;
   – $\varphi_{MQ}(A_1, n, k) \asymp \log n$ *при* $n \to \infty$;
   – $\varphi_{MQ}(A_2, n, k) \asymp \log n$ *при* $n \to \infty$;
   – $\varphi_{MQ}(A_4, n, k) \asymp \log n$ *при* $n \to \infty$;
   – $\varphi_{MQ}(D_1, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
   – $\varphi_{MQ}(D_2, n, k) \asymp \log n$ *при* $n \to \infty$;

- $\varphi_{MQ}(D_3, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_1^i, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_2^i, n, k) \asymp \log n$ *при* $n \to \infty$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_3^i, n, k) \asymp \log n$ *при* $n \to \infty$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{MQ}(F_4^i, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
- $\varphi_{MQ}(S_1, n, k) \asymp \log_2 n, n \to \infty$;
- $\varphi_{MQ}(S_3, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(S_6, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_1, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_2, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_3, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_4, n, k) \asymp \log n, n \to \infty$;
- $\varphi_{MQ}(L_5, n, k) \asymp \log n, n \to \infty$.

Доказательство. Теорема является объединением результатов утверждений 7 и лемм 35, 37, 43, 49, 51, 53, 54. □

Глава 4. Расшифровка функций замкнутых классов Поста запросами на сравнение

В данной главе исследуется значение сложности точной расшифровки замкнутых классов Поста (рисунок 2) запросами на сравнение. Также как и в предыдущей главе, учитель выбирает функцию из одного из классов решетки, а ученику известен сам класс и то, что у функции n переменных и не более k из них являются существенными. Аналогично второй главе при рассмотрении задачи 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 $\log_2 k = o(\log_2 n)$.

Напомним, что довольно многие классы Поста уже исследовались с точки зрения сложности расшифровки запросами на значение, но никакие из них не изучались с точки зрения запросов на сравнение, поскольку этот тип запросов был введен в литературу сравнительно недавно [4]. Поэтому интерес представляет разница результатов рассматриваемой задачи для запросов на значение, демонстрируемые во второй главе, и запросов на сравнение, рассматриваемые в этой.

Глава состоит из девяти разделов. В первом разделе приведены все вспомогательные определения и утверждения, используемые далее в доказательствах. Следующие семь разделов демонстрируют результаты исследования сложности расшифровки для классов, сгруппированных по букве в их обозначении: $C_i, A_i, D_i, F_j^i, S_i, L_i, O_i$. Глава завершается разделом с теоремами, в которых представлены результаты по всем классам Поста для двух случаев: оба n, k устремлены к бесконечности, только n устремлен к бесконечности.

Основное различие между запросами на значение и запросами на сравнение демонстрируется в разделе со вспомогательными определениями и утверждениями, в котором 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. 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.

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.

4.1 Вспомогательные утверждения и определения

В разделе приводятся вспомогательные утверждения, позволяющие понять, для каких классов решетки Поста возможно рассматривать задачу точной расшифровки запросами на сравнение, а также показывается, как преобразовать алгоритм расшифровки запросами на значение в алгоритм расшифровки запросами на сравнение для классов решетки Поста.

Непосредственно из теоремы 4 получаем следующее следствие.

Следствие 3. *Если* $F \in \{C_1, A_1, L_1, S_6, O_9, O_6\}$, *то расшифровать запросами на сравнение класс* F *невозможно. Если* $F \in \{C_2, C_3, C_4, A_2, A_3, A_4, F_4^i, F_3^i, F_1^i, F_2^i, D_1, D_2, S_1, S_3, S_5, L_2, L_3, L_4, O_8, O_5, O_1, O_4, D_3, L_5\}$, *то класс* F *можно расшифровать.*

Чтобы не терять полностью из рассмотрения замкнутые классы $C_1, A_1, L_1, S_6, O_9, O_6$ 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 C_1^*, L_1^*, O_9^* — замкнутые классы, а $A_1^* = A_2, S_6^* = S_3, O_6^* = O_5$.

Лемма 55. *Если для некоторого набора* a, *некоторого числа* $b \in \{0,1\}$ *для любой* $f \in F(n,k)$ *известно, что* $f(a) = b$, *и* $\{0,1\} \not\subseteq F(n,k)$, *тогда* $\varphi_{CQ}(F,n,k) \leqslant \varphi_{MQ}(F,n,k)$.

Доказательство. В силу теоремы 4, класс $F(n,k)$ можно расшифровать запросами на сравнение, поэтому дальнейшие рассуждения имеют смысл.

Рассмотрим алгоритм расшифровки A запросами на значение, на котором достигается минимум выражения $max_{f \in F(n,k)} \varphi_{MQ}(A, f)$. По алгоритму A построим алгоритм расшифровки B запросами на сравнение следующим образом: первая компонента каждого запроса на сравнение есть набор из алгоритма A, а вторая компонента запроса на сравнение – это набор a. Тогда для любой $f \in F(n,k)$ верно соотношение $\varphi_{CQ}(B, f) = \varphi_{MQ}(A, f)$. Отсюда следует равенство $max_{f \in F(n,k)} \varphi_{CQ}(A, f) = max_{f \in F(n,k)} \varphi_{MQ}(A, f)$. Учитывая, что в $\mathcal{A}^{CQ}_{F(n,k)}$ помимо предложенного алгоритма расшифровки возможно имеются еще какие-то, получаем неравенство леммы. □

Лемма 56. *Если* $F(n,k) \subseteq D_3(n,k)$, *тогда* $\varphi_{CQ}(F,n,k) \leqslant \varphi_{MQ}(F,n,k)$.

Доказательство. Рассмотрим алгоритм расшифровки A запросами на значение, на котором достигается минимум выражения $max_{f \in F(n,k)} \varphi_{MQ}(A, f)$. По алгоритму A построим алгоритм расшифровки B запросами на сравнение следующим образом: первая компонента каждого запроса на сравнение есть набор из алгоритма A, а вторая компонента запроса на сравнение – это противоположный ему набор. Ответ на любой такой запрос на сравнение однозначно восстановит значение функции на обоих наборах в силу того, что наборы противоположные, а функция самодвойственная. Поэтому для любой $f \in F(n,k)$ верно соотношение $\varphi_{CQ}(B, f) = \varphi_{MQ}(A, f)$. Отсюда следует $max_{f \in F(n,k)} \varphi_{CQ}(A, f) = max_{f \in F(n,k)} \varphi_{MQ}(A, f)$. Учитывая, что в $\mathcal{A}^{CQ}_{F(n,k)}$ помимо указанного алгоритма расшифровки возможно имеются еще какие-то, получаем неравенство леммы. □

Лемма 57. *(Нижняя мощностная оценка) Если* $\{0,1\} \not\subseteq F(n,k)$, *то* $\varphi_{CQ}(F,n,k) \geqslant \left\rceil \log_3 |F(n,k)| \right\lfloor$.

Доказательство. В силу теоремы 4, класс $F(n,k)$ можно расшифровать запросами на сравнение, поэтому дальнейшие рассуждения имеют смысл.

Доказательство аналогично случаю расшифровки запросами на значение. Пусть задано x запросов на сравнение, на каждый запрос возможны 3 варианта ответа, следовательно, всего возможных векторов ответов не более 3^x (“не более” в силу того, что какие-то комбинации ответов на запросы могут быть между собой не согласованы, то есть не определяют функцию из заданного класса).

Необходимо задать такое количество запросов, чтобы однозначно восстановить функцию, поэтому должно выполняться следующее неравенство $3^x \geqslant |F(n,k)|$. Отсюда следует неравенство леммы. □

Лемма 58. *Если известно, что среди множества переменных* $A = \{x_{i_1}, x_{i_2}, \ldots, x_{i_q}\}, 1 \leqslant i_1 < i_2 < \ldots < i_q \leqslant n$, *нечетное число существенных переменных функции* $f(x_1, x_2, \ldots, x_n) \in L_3$. *Тогда одну существенную переменную можно найти, задав не более* $\rceil \log_3 q \lceil$ *запросов на сравнение.*

Доказательство. Шаг 1. Если $|A| = 1$, тогда СТОП, переменная из множества и есть существенная. Иначе, разделить множество A на непересекающиеся множества A_1, A_2, A_3, отличающиеся по мощности не более, чем на один элемент. Перейти к шагу 2.

Шаг 2. Запросить значение на сравнение на множествах A_1, A_2. Если значение равно 1, тогда положить $A = A_1$, перейти к шагу 1. Если значение равно -1, тогда положить $A = A_2$, перейти к шагу 1. Иначе, положить $A = A_3$, перейти к шагу 1.

Если на шаге 2 ответ на запрос равен 0, то значит, либо в обоих A_1, A_2 нечетное число существенных из множества A и значение функции на наборах, сформированных по каждому из множеств A_1, A_2 равно 1, либо в обоих A_1, A_2 четное число существенных из множества A и значение функции на наборах, сформированных по каждому из множеств A_1, A_2, равно 0. В обоих случаях, суммарно в A_1, A_2 содержится четное число существенных переменных, значит нечетное число существенных содержится в A_3.

После каждого шага 2 мощность множества A уменьшается как минимум в 3 раза.

□

Лемма 59. *Сложность расшифровки запросами на сравнение функций из классов* C_2, C_3, C_4, A_2, A_3, A_4, $F_4^i, F_3^i, F_1^i, F_2^i$, D_1, D_2 *и классов* S_1, S_3, S_5, L_2, L_3, L_4, O_8, O_5, D_3, L_5 *не хуже, чем сложность расшифровки запросами на значение.*

Доказательство. Если $F \in \{C_2, C_3, C_4, A_2, A_3, A_4, F_4^i, F_3^i, F_1^i, F_2^i\}$ или $F \in \{D_1, D_2, S_1, S_3, S_5, L_2, L_3, L_4, O_8, O_5\}$, то согласно лемме 55 получаем неравенство $\varphi_{CQ}(F,n,k) \leqslant \varphi_{MQ}(F,n,k)$.

Если $F \in \{D_3, L_5\}$, то в силу леммы 56 получаем неравенство $\varphi_{CQ}(F, n, k) \leqslant \varphi_{MQ}(F, n, k)$.

□

Стоит отметить, что верхние оценки для классов получаются использованием леммы 59. Нижние оценки получаются адаптацией доказательств нижних оценок этих классов с третьей главы. Если в случае запросов на значение для класса нижняя оценка мощностная, то в случае запросов на сравнение по порядку она такая же. Если нижняя оценка получалась конструктивным образом, то и для запросов на сравнение она конструктивная, причем в качестве загадываемой функции учитель выбирает для запросов на сравнение такие же, как и в случае запросов на значение.

4.2 Классы C_i

Покажем, что сложность расшифровки этих классов по порядку равна сложности построения бинарных покрывающих матриц с соответствующими параметрами.

Лемма 60. *Имеет место неравенство*

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

Доказательство. Для получения нижней оценки на первые запросы $[0.5(\alpha(n, k) - 1)] - 1$ 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 25 и факта, что в один запрос на сравнение можно включить две строки почти покрывающей матрицы типа 1. □

Лемма 61. *При $n > 1, k > 1$ имеет место неравенство*

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

Доказательство. По сути доказательство этой леммы повторяет доказательство нижней оценки леммы 34. То есть необходимо вынудить 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 $x_2, \ldots, x_n$, а $x_1 = 0$,
2. строки покрывающей матрицы $M_1(n-1, k-1)$ типа 1, где столбцам матрицы соответствуют переменные $x_2, \ldots, x_n$, 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 $(a_1a_2a_3 \ldots a_n, b_1b_2b_3 \ldots b_n)$, тогда ответ на него $sign(a_1 - b_1)$.

Учитывая лемму 25, 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 $\beta_0(n, k) + \beta_1(n, k) \geqslant 2\alpha(n, k) - 2$. 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 леммы 34, будут удовлетворять ответам учителя на $\alpha(n-1, k-1) - 2$ запросов на сравнение. □

Лемма 62. *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 леммы следует из леммы 59 и лемм 33, 60, 26.

Доказательство третьего пункта леммы следует из леммы 59 и лемм 34, 61, 27 при $p = 1$. □

Лемма 63. *При $k \geqslant 2$ 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 лемм 33, 60 и 34, 61, а также леммы 28. □

4.3 Классы A_i

В следующих леммах покажем, что для обоих классов A_2, A_4 порядок расшифровки запросами на сравнение совпадает с порядком расшифровки запросами на значение как в случае $n, k \to \infty$, так и в случае только $n \to \infty$

Лемма 64. *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 лемм 59 и 36.

Нижняя оценка, полученная в [14], для класса $A_1(n, k)$, как и для классов A_2, A_4 в лемме 36 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 $\frac{2^k}{\sqrt{k}} + k \log n$. □

Лемма 65. *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* $n \to \infty$.

Доказательство. Верхняя оценка для обоих классов следует из леммы 59.

В доказательстве леммы 64 для классов A_2, A_4 были приведены нижние мощностные оценки, которые в случае фиксированного k и устремлении n к бесконечности превращаются в оценки по порядку не меньшие $\log n$. □

4.4 Классы D_i

Все изучаемые в этом разделе классы (D_1, D_2, D_3) можно расшифровать запросами на сравнение согласно следствию 3. В разделе приводятся нижние оценки для упомянутых классов, опирающиеся на нижние оценки их расшифровки запросами на значение (леммы 38, 40, 41).

Лемма 66. *Для $k > 1$ имеет место неравенство*

$$\varphi_{CQ}(D_3, n, k) \geqslant \left[0.5\alpha(n-1, k-1)\right].$$

Доказательство. На все $\left[0.5\alpha(n-1,k-1)\right]-1$ запросов ученика будем отвечать следующим образом. Если запрос на сравнение имеет вид $(a_1a_2\ldots a_n, b_1b_2\ldots b_n)$, тогда ответим $sign(a_1 - b_1)$. Тогда аналогично доказательству леммы 38 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 $2[0.5\alpha(n-1, k-1)] - 2$. Значит ученик не опросил какую-то фиксацию $k-1$ переменной из переменных $x_2, x_3, \ldots, x_n$. 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 $[0.5\alpha(n-1, k-1)] - 1$ 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 $\left[0.5\alpha(n-1,k-1)\right]-1$ 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 $\big[0.5\alpha(n-1,k-1)\big]$-го запроса ученик не знает, какая функция загадана: x_1 или функция с ровно k существенными переменными. □

Лемма 67. *Для любого $k>1$ имеет место неравенство*

$$\varphi_{CQ}(D_1,n,k) \geqslant \big[0.5(\alpha(n-1,k-1)-1)\big].$$

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

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

Далее доказательство повторяет доказательство леммы 66 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 $x_2,x_3,\ldots,x_n$.

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 $2\big[0.5(\alpha(n-1,k-1)-1)\big]-2$. Согласно лемме 25 верно соотношение $\beta_0(n-1,k-1)\geqslant\alpha(n-1,k-1)-1$, поэтому своими $\big[0.5(\alpha(n-1,k-1)-1)\big]-1$ 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 матрицы 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 66. □

Лемма 68. *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 41). □

Лемма 69. *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. Верхняя оценка для классов D_1, D_3 следует из лемм 59 и 42. Нижняя оценка для класса D_3 следует из лемм 66, 27 при $p = 1$. Нижняя оценка для класса D_1 следует из лемм 67, 27 при $p = 1$.

Верхняя оценка для класса D_2 следует из лемм 59 и 42, а нижняя — из леммы 68.

□

Лемма 70. *Верны следующие соотношения.*

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

Доказательство. Для классов D_1, D_3 доказательство следует из лемм 59, 42 и лемм 66, 67 и 28.

Для класса D_2 доказательство следует из лемм 59, 42 и 68. □

4.5 Классы $F_i^j, 1 \leqslant j \leqslant 4$

Перейдем к рассмотрению задачи расшифровки группы классов — "счетной этажерке" решетки Поста, среди которых все классы можно расшифровать запросами на сравнение согласно следствию 3.

Лемма 71. *Имеет место неравенство*

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

Доказательство. Аналогично доказательству леммы 44, в качестве загаданной функции выберем константу 1 и отвечать на все запросы на сравнение 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 опрошенных этим запросом ненулевых фиксаций 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 25, получаем неравенство доказываемой леммы. □

Лемма 72. *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 73. *Пусть* $i > 1, k > 1, 1 \leqslant p < k$, p *— целое. Тогда справедливо неравенство*

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

Доказательство. Доказательство этой нижней оценки опирается на доказательство нижней оценки сложности расшифровки этого же класса запросами на значение (лемма 46). Учитель отвечает на запросы ученика так, чтобы ответам на все запросы кроме последнего заданного удовлетворяли две функции, которые использовались в доказательстве леммы 46: $x_1|x_2|\ldots|x_p$ и функция, отличающаяся от $x_1|x_2|\ldots|x_p$ ровно в одном наборе и равная 1 на всех наборах с $x_1 = x_2 = \ldots = x_p = 1$.

Напомним, что в доказательстве леммы 46 различались следуюшие три категории n-местных наборов $x_1x_2\dots x_n$:

1. $x_1 = x_2 = \dots = x_p = 0$,
2. $x_1 = x_2 = \dots = x_p = 1$,
3. остальные наборы.

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

Лемма 74. *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. Соотношение, касающееся класса $F_4^i(n, k)$, следует из лемм 71, 31, леммы 59, верхней оценки утверждения 1, а также леммы 26.

Соотношение, касающееся класса $F_1^i(n, k)$, следует из лемм 73, 45, 27, 29 и 59.

Соотношения, касающееся классов $F_2^i(n, k), F_3^i$(n, k), по лемме 31 следуют из леммы 59 и верхней оценки утверждения 2, а также леммы 72, потому что класс $F_2^\infty(n, k)$ вкладывается в классы $F_2^i(n, k), F_3^i(n, k)$. □

Лемма 75. *Справедливы соотношения*

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

– *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* $n \to \infty$, $k \geqslant 2$.

Доказательство. Доказательство всех пунктов следует из леммы 74 и лемм 28, 30. □

4.6 Классы S_i

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.

Лемма 76. *Справедливы соотношения*
– $\varphi_{CQ}(S_1, n, k) \asymp k \log(n/k), n, k \to \infty, k = o(n)$;
– $\varphi_{CQ}(S_3, n, k) \asymp k \log(n/k), n, k \to \infty, k = o(n)$;
– $\varphi_{CQ}(S_5, n, k) \asymp k \log(n/k), n, k \to \infty, k = o(n)$.

Лемма 77. *Справедливы соотношения*
– $\varphi_{CQ}(S_1, n, k) \asymp \log(n), n \to \infty$;
– $\varphi_{CQ}(S_3, n, k) \asymp \log(n), n \to \infty$;
– $\varphi_{CQ}(S_5, n, k) \asymp \log(n), n \to \infty$.

4.7 Классы L_i

Для классов линейных функций в случае запросов на значение (лемма 52) 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.

Лемма 78. *Справедливы соотношения*

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

Лемма 79. *Справедливы соотношения*

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

4.8 Классы O_i

Перейдем к результатам расшифровки самых маленьких по мощности классов решетки Поста. В этом разделе наконец приводятся классы, для которых сложность расшифровки запросами на сравнение строго меньше таковой для запросов на значение.

Лемма 80. *Справедливы следующие соотношения:*

1. $\varphi_{CQ}(O_1, n, 1) = \rceil \log_3 n \lceil \leqslant \rceil \log_2 n \lceil = \varphi_{MQ}(O_1, n, 1)$,
2. $\varphi_{CQ}(O_4, n, 1) \leqslant \rceil \log_3 n \lceil + 1 \leqslant \rceil \log_2 n \lceil + 1 = \varphi_{MQ}(O_4, n, 1)$.

Доказательство. Соотношение $\varphi_{CQ}(O_1, n, 1) = \rceil \log_3 n \lceil$ следует из лемм 57 и 58. Учитывая соотношение $\varphi_{MQ}(O_1, n, 1) = \rceil \log_2 n \lceil$ из [39], получаем первый пункт леммы.

Для получения верхней оценки для $\varphi_{CQ}(O_4, n, 1)$ предлагается следующий алгоритм расшифровки: узнаем значение на запросе $(0 \ldots 0, 1 \ldots 1)$, если ответ равен -1, то загадана функция $x_i (1 \leqslant i \leqslant n)$, иначе загадана функция $\overline{x_i} (1 \leqslant i \leqslant n)$. В случае функции x_i применяем алгоритм расшифровки, описанный в доказательстве леммы 58, в случае функции $\overline{x_i}$ применяем похожий алгоритм. В результате, получаем неравенство $\varphi_{CQ}(O_4, n, 1) \leqslant 1 + \rceil \log_3 n \lceil$. Учитывая соотношение $\varphi_{MQ}(O_4, n, 1) = \rceil \log_2 n \lceil + 1$ леммы 54, получаем второй пункт доказываемой леммы.

□

Лемма 81. *Для $n > 1$ справедливы следующее соотношение:*

1. *соотношение $\rceil \log_3 n + \log_3 2 \lceil = \rceil \log_3 n \lceil$ 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 $\rceil \log_3 n + \log_3 2 \lceil = \rceil \log_3 n \lceil + 1$ 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 $\log_3 n = \Big[\log_3 n\Big] = \Big[\log_3 n + \log_3 2\Big]$. Равенство второго пункта леммы верно.

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

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

Равенство второго пункта леммы верно.

□

Лемма 82. *При $n > 1$ справедливы соотношения*

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

Доказательство. Точная оценка для класса O_1 и верхняя оценка для класса O_4 следует из леммы 80.

Нижняя мощностная оценка для класса O_4 следующая:

$$\varphi_{CQ}(O_4, 1, n) \geqslant \rceil \log_3 (2n) \lceil = \rceil \log_3 n + \log_3 2 \lceil.$$

Согласно лемме 81, при определенных значениях n оценка $\rceil \log_3 n + \log_3 2 \lceil$ равна значению $\rceil \log_3 n \lceil + 1$, 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 леммы 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 $(0 \ldots 0, 1 \ldots 1)$, если ответ равен 0, то загадана единственная в классе константа, иначе загадана функция $x_i (1 \leqslant i \leqslant n)$. В случае функции x_i применяем алгоритм расшифровки, описанный в доказательстве леммы 58. Поэтому сложность расшифровки классов O_5, O_8 не превосходит $\rceil \log_3 n \lceil + 1$. Нижняя оценка для обоих классов мощностная и не меньше $\rceil \log_3 (n+1) \lceil$, что совпадает с $\rceil \log_3 n \lceil + 1$ при целом $\log_3 n$ и на 1 меньше $\rceil \log_3 n \lceil + 1$ в остальных случаях. Третий и четвертый пункты леммы доказаны.

Нижняя мощностная оценка для класса O_9^* следующая:

$$\varphi_{CQ}(O_9^*, 1, n) \geqslant \rceil \log_3 (2n+1) \lceil \geqslant \rceil \log_3 n + \log_3 2 \lceil.$$

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

$$\varphi_{CQ}(O_9^*, 1, n) \leqslant 1 + \rceil \log_3 n \lceil.$$

□

4.9 Теорема о сложности расшифровки для всех классов Поста

В этом разделе приводятся теоремы, объединяющие леммы предыдущих разделов данной главы и позволяющие увидеть общую картину результатов точной расшифровки замкнутых классов решетки Поста запросами на сравнение. В теореме 14 собраны результаты для случая $n, k \to \infty$, в теореме 15 — для случая $n \to \infty$ с фиксированным k.

Напомним, что 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.

Теорема 14. *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*
   - $\varphi_{CQ}(O_1, n, 1) = \left]\log_3 n\right[$;
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* $\varphi_{CQ}(F_4^i, n, k) \asymp \alpha(n,k)$ *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* $n, k \to \infty, k = o(n)$.

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* $n, k \to \infty, \log k = o(\log n)$.

Доказательство. Теорема является объединением результатов лемм 62, 64, 69, 74, 76, 78, 82. □

На рисунке 2 схематично приведены основные результаты данной главы. Если класс выделен квадратом с обеими диагоналями, тогда этот класс 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 15. *Замкнутые классы решетки Поста по характеру известной на данный момент сложности точной расшифровки запросами на сравнение в случае, когда* n *растет, а* k *не меняется, разделены на три группы:*

1. *точная оценка*
   - $\varphi_{CQ}(O_1, n, 1) = \left] \log_3 n \right[$;
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* $n \to \infty$;
- $\varphi_{CQ}(C_2, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
- $\varphi_{CQ}(C_4, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
- $\varphi_{CQ}(A_2, n, k) \asymp \log n$ *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* $n \to \infty, k \geqslant 2$;
- $\varphi_{CQ}(D_2, n, k) \asymp \log n$ *при* $n \to \infty$;
- $\varphi_{CQ}(D_3, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
- *если* $i > 1$ *или* $i = \infty$, *то* $\varphi_{CQ}(F_1^i, n, k) \asymp \log n$ *при* $n \to \infty, k \geqslant 2$;
- *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* $n \to \infty$, $k \geqslant 2$;
- $\varphi_{CQ}(S_1, n, k) \asymp \log n$ *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* $n \to \infty$.

Доказательство. Теорема является объединением результатов лемм 63, 65, 70, 75, 77, 79, 82. □

Заключение

Основные результаты работы заключаются в следующем.

1. Получены значения сложности параметро-эффективной точной расшифровки класса функций ограниченного веса для трех типов запросов в отдельности: на значение, на расширенную и ограниченную эквивалентность.
2. Получены значения сложности точной расшифровки запросами на сравнение класса функций малого веса: 1, 2, 3.
3. Получена практически точная оценка сложности расшифровки запросами на сравнение классов функций веса ограниченного снизу нулем и ограниченного снизу единицей.
4. Доказан порядок сложности точной расшифровки запросами на сравнение класса функций ограниченного веса в случае, когда растет арность функций, а ее вес не меняется.
5. Доказаны оценки сложности параметро-эффективной точной расшифровки запросами на значение замкнутых классов самодвойственных функций и классов “счетной этажерки” решетки Поста.
6. Доказаны оценки сложности параметро-эффективной точной расшифровки запросами на сравнение всех замкнутых классов решетки Поста.

Дальнейшее изучение сложности параметро-эффективной точной расшифровки произвольных множеств как булевых функций, так и функций многозначной логики разными типами запросов является перспективным направлением исследований на междисциплинарном уровне, поскольку для определения значения сложности расшифровки затрагиваются как разделы дискретной математики, так и теории тестирования, и теории кодирования. Особый интерес представляет получение хотя бы асимптотики размера бинарных покрывающих матриц, что позволит понять асимптотику сложности расшифровки многих замкнутых классов Поста. Также полезным представляется изучение того, какие типы или комбинации типов запросов оптимальнее использовать для восстановления из частичных сведений наиболее чаще встречающихся на практике функций.

Список литературы

1. *Анселъ, Ж.* О числе монотонных булевых функций от n переменных / Ж. Ансель // Кибернетический сборник. Новая серия. Т. 5. — Мир, 1968. — С. 53—57.
2. *Блохина, Г. Н.* О спектрах классов Поста булевских функций / Г. Н. Блохина, В. Б. Кудрявцев // Интеллектуальные системы. — 2010. — Т. 14, № 1—4. — С. 279—298.
3. *Вороненко, А. А.* Расшифровка бесповторных функций запросами тождественности / А. А. Вороненко, Д. В. Чистиков // Проблемы теоретической кибернетики. Материалы XVI Международной конференции. — 2011. — С. 105—108.
4. *Гасанов, Э. Э.* Расшифровка линейных функций ранжирования / Э. Э. Гасанов // Материалы XI Международного семинара «Дискретная математика и ее приложения» (Москва, 18-23 июня 2012 г.) — 2012. — С. 332—334.
5. *Золотых, Н. Ю.* Расшифровка пороговых функций k-значной логики / Н. Ю. Золотых, В. Н. Шевченко // Дискретный анализ и исследование операций. — 1995. — Т. 2, № 3. — С. 18—23.
6. *Ильин, В. А.* Математический анализ. Начальный курс / В. А. Ильин, В. А. Садовничий, Б. Х. Сендов. — Издательство Московского университета, 1985.
7. *Калачев, Г. В.* Об оценках мощности плоских схем для замкнутых классов булевых функций / Г. В. Калачев // Интеллектуальные системы. Теория и приложения. — 2016. — Т. 20, № 3. — С. 52—57.
8. *Калачев, Г. В.* Оценки мощности плоских схем, реализующих функции с ограниченным числом единиц / Г. В. Калачев // Интеллектуальные системы. Теория и приложения. — 2017. — Т. 21, № 1. — С. 28—96.
9. *Кибкало, М. А.* Об автоматной сложности классов Поста булевых функций / М. А. Кибкало // Интеллектуальные системы. — 2010. — Т. 15, № 1—4. — С. 379—400.
10. *Комков, С. А.* Мощности генерирующих множеств по операциям из классов решетки Поста / С. А. Комков // Дискретная математика. — 2018. — Т. 30, № 1. — С. 19—38.

11. *Коробков, В. К.* О монотонных функциях алгебры логики / В. К. Коробков // Сб. Проблемы кибернетики. Т. 13. — Москва : Наука, 1965. — С. 5—28.
12. *Кулямин, В. В.* Обзор методов построения покрывающих наборов / В. В. Кулямин, А. А. Петухов // Программирование. Т. 37. — Москва : Российская академия наук, 2011. — С. 3—41.
13. *Осокин, В. В.* О параллельной параметро-эфективной расшифровке псевдо-булевых функций / В. В. Осокин // Интеллектуальные системы. — 2010. — Т. 14, № 1—4. — С. 429—458.
14. *Осокин, В. В.* О расшифровке монотонных булевых функций с несущественными переменными / В. В. Осокин // Дискретная математика. — 2010. — Т. 22, № 3. — С. 134—145.
15. *Селезнева, С. Н.* Расшифровка монотонных функций с исправлением одной ошибки / С. Н. Селезнева, Ю. Лю // Дискретная математика. — 2019. — Т. 31, № 4. — С. 53—69.
16. *Хегай, С. И.* Расшифровка полиномиальных функций ранжирования / С. И. Хегай // Интеллектуальные системы. Теория и приложения. — 2015. — Т. 19, № 1. — С. 213—230.
17. *Членова, Т. С.* О слоистости замкнутых классов булевых функций и функций k-значной логики / Т. С. Членова // Интеллектуальные системы. Теория и приложения. — 2014. — Т. 18, № 1. — С. 259—262.
18. A survey of Binary Covering Arrays / J. Lawrence [et al.] // The electronic journal of combinatorics. — 2011. — Vol. 18, no. 1. — P. 1—30.
19. *Angluin, D.* Queries and Concept Learning / D. Angluin // Machine Learning. — 1988. — Vol. 2. — P. 319—342.
20. *Angluin, D.* Queries Revisited / D. Angluin // Theoretical Computer Science. — 2004. — Vol. 313, no. 2. — P. 175—194.
21. *Bertoni, A.* Efficient learning with equivalence queries of conjunctions of modulo functions / A. Bertoni, N. Cesa-Bianchi, G. Fiorino // Information Processing Letters. — 1995. — Vol. 56, no. 1. — P. 15—17.
22. *Damaschke, P.* Adaptive Versus Nonadaptive Attribute-Efficient Learning / P. Damaschke // Machine Learning. — 2000. — Vol. 41. — P. 197—215.

23. *Damaschke, P.* On parallel attribute-efficient learning / P. Damaschke // Journal of Computer Science. — 2003. — Vol. 67. — P. 46—62.

24. *Das, S.* A Semi-Random Construction of Small Covering Arrays / S. Das, T. Mészáros. — 2017.

25. *Gal, S.* Rendezvous search on a line / S. Gal // Operations Research. — 1999. — Vol. 47. — P. 974—976.

26. *Gargano, L.* Sperner capacities / L. Gargano, J. Korner, U. Vaccaro // Graph. Combinator. — 1993. — Vol. 9. — P. 31—46.

27. *Godbole, A.* t-Covering Arrays: Upper Bounds and Poisson Approximations / A. Godbole, D. Skipper, R. Sunley // Combinatorics, Probability & Computing. — 1996. — Vol. 5. — P. 105—117.

28. *Hartman, A.* Software and Hardware Testing Using Combinatorial Covering Suites / A. Hartman // Graph Theory, Combinatorics and Algorithms. Vol. 34. — 2006. — P. 237—266. — (Operations Research/Computer Science Interfaces Series).

29. *Hofmeister, T.* An Application of Codes to Attribute-Efficient Learning / T. Hofmeister // Proceedings of the 4th European Conference on Computational Learning Theory. — 1999. — P. 101—110. — (EuroCOLT '99).

30. *Kleitman, D. J.* Families of k - independent sets / D. J. Kleitman, J. Spencer // Discrete Mathematics. — 1973. — Vol. 6. — P. 255—262.

31. *Lim, W. S.* Minimax rendezvous on the line / W. S. Lim, S. Alpern // SIAM J. Control and Optim. — 1996. — Vol. 34. — P. 1650—1665.

32. *Littlestone, N.* Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm / N. Littlestone // Machine Learning. — 1988. — Vol. 2. — P. 285—318.

33. Oracles and Queries That Are Sufficient for Exact Learning / N. H. Bshouty [et al.] // Journal of Computer and System Sciences. — 1996. — Vol. 52, no. 3. — P. 421—433.

34. Partial Covering Arrays: Algorithms and Asymptotics / K. Sarkar [et al.] // International Workshop on Combinatorial Algorithms IWOCA 2016: Combinatorial Algorithms. — 2016. — P. 437—448.

35. *Post, E.* Determination of all closed systems of truth tables / E. Post // Bull. Amer. Math. Soc. — 1920.

36. *Post, E.* Two-valued iterative systems of mathematical logic / E. Post // Annals of Mathematics studies, Princeton Univ. Press, Princeton. — 1941. — No. 5.

37. *Sloane, N. J. A.* Covering arrays and intersecting codes / N. J. A. Sloane // Journal of Combinatorial Designs. — 1993. — Vol. 1. — P. 51—63.

38. The AETG System: An Approach to Testing Based on Combinatorial Design / D. M. Cohen [et al.] // IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. — 1997. — Vol. 23, no. 7. — P. 437—444.

39. *Uehara, R.* Optimal Attribute-Efficient Learning of Disjunction, Parity and Threshold Functions / R. Uehara, K. Tsuchida, I. Wegener // Proceedings of the Third European Conference on Computational Learning Theory. — 1997. — P. 171—184. — (EuroCOLT '97).

40. *Valiant, L. G.* A theory of the learnable / L. G. Valiant // STOC '84. — 1984. — P. 436—445.

Список публикаций автора по теме диссертации

41. *Быстрыгова, А. В.* Запросы на сравнение в задаче параметро-эффективной расшифровки булевых функций / А. В. Быстрыгова // Интеллектуальные системы. Теория и приложения. — 2019. — Т. 23, № 4. — С. 115—124.

42. *Быстрыгова, А. В.* Параметро-эффективная расшифровка булевых функций из замкнутых классов Поста / А. В. Быстрыгова // Дискрет. матем. — 2019. — Т. 31, № 2. — С. 34—58.

43. *Быстрыгова, А. В.* Расшифровка булевых функций фиксированного веса / А. В. Быстрыгова // Интеллектуальные системы. Теория и приложения. — 2020. — Т. 24, № 3. — С. 63—96.

44. *Быстрыгова, А. В.* Расшифровка булевых функций ограниченного веса / А. В. Быстрыгова // Вестник Московского университета. Серия 1: Математика. Механика. — 2021. — № 6. — С. 14—20.

45. *Быстрыгова, А. В.* Запросы на сравнение в задаче точной расшифровки замкнутых классов Поста / А. В. Быстрыгова // Интеллектуальные системы. Теория и приложения. — 2022. — Т. 26, № 3. — С. 88—108.