УДК 519.716

РАСШИФРОВКА БУЛЕВЫХ ФУНКЦИЙ ОГРАНИЧЕННОГО ВЕСА

А. В. Быстрыгова[1]

Работа посвящена изучению сложности точной расшифровки булевых функций ограниченного веса четырьмя типами запросов: на значение, сравнение, ограниченную и расширенную эквивалентность. Для всех рассматриваемых классов функций получены точные значения сложности расшифровки тремя типами запросов: на значение и на ограниченную и расширенную эквивалентность. Для запросов на сравнение приводятся верхние и нижние оценки, совпадающие по порядку. Помимо этого для класса функций веса, ограниченного только сверху, получена точная оценка сложности расшифровки запросами на сравнение.

Ключевые слова: функции ограниченного веса, запросы на значение, запросы на сравнение, запросы на ограниченную эквивалентность, запросы на расширенную эквивалентность, точная расшифровка.

The complexity of exact learning of bounded-weight Boolean functions is considered in the paper using separately four types of queries: membership queries, equivalence queries, extended equivalence queries, and comparation queries. The values of the complexity for the first three types are obtained. Upper and lower bounds of the complexity for comparation queries being of the same order are presented. Moreover, the exact value of the complexity of learning upper bounded-weight Boolean functions with the help of comparation queries is obtained.

Key words: bounded-weight functions, membership queries, comparation queries, equivalence queries, extended equivalence queries, exact learning.

Введение. Одной из задач теории обучения (computational learning theory) является расшифровка функций. Это игра между "учителем" и "учеником ", в которой учитель выбирает функцию из класса, известного ученику, а ученик, отправляя учителю запросы определенного типа, должен однозначно понять по ответам учителя, какая функция выбрана. Аналогом функции Шеннона для такой задачи является сложность расшифровки. Расшифровка функций фиксированного класса интересна исследователям с точки зрения того, как меняется сложность его расшифровки при использовании разных типов запросов.

Одними из ведущих в этой области стали работы Д. Англуин [1, 2], в которых изучалась сложность расшифровки некоторых классов сразу для трех типов запросов: запросов на значение, на ограниченную и расширенную эквивалентность. В работе [3] (или в ее англоязычной версии [4]) исследовалась сложность расшифровки замкнутых классов Поста запросами на значение. В [5] показано, что в задаче расшифровки замкнутых классов Поста сложность расшифровки запросами на сравнение незначительно больше сложности расшифровки запросами на значение, а для некоторых классов и строго меньше. Работа [6] посвящена исследованию сложности расшифровки класса булевых функций фиксированного веса для четырех типов запросов: запросов на значение, сравнение и ограниченную и расширенную эквивалентность; были получены точные значения сложности расшифровки запросами на значение, на оба типа эквивалентности, а также точные значения сложности расшифровки запросами на сравнение для функций малого веса (1, 2, 3).

В настоящей работе исследуется этот же вопрос для более широкого класса — класса функций ограниченного веса. Демонстрируется тот факт, что при расширении изучаемого класса функций сложность расшифровки для одного типа запросов может сильно уменьшиться (как это будет показано для запросов на ограниченную эквивалентность), а для других типов остаться почти неизменной (как это будет видно на примере запросов на значение).

Основные понятия и формулировка результатов. Будем обозначать через $P(n)$ множество булевых функций арности n. 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 $k \in [1,2^n)$, $0 \leqslant i \leqslant k$,

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

Bystrygova Anastasiya Viktorovna — Postgraduate, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

будем понимать множество $\{f \in P(n) : i \leqslant |f| \leqslant k\}$, иными словами, $F(n,k,i)$ — множество булевых функций арности n, вес которых лежит в диапазоне $[i,k]$.

Пусть учителем загадана функция $f \in F(n,k,i)$. 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 понимать функцию $g \in F(n,k,i)$, 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 $g \in P(n)$, а 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 $A^T_{n,k,i}$ 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 $f \in F(n,k,i)$. Через $\mathcal{A}^T_{n,k,i}$ будем обозначать множество всех алгоритмов $A^T_{n,k,i}$ 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: $\varphi_T(n,k,i) = \min_{A \in \mathcal{A}^T_{n,k,i}} \max_{f \in F(n,k,i)} q(A,f)$.

Чтобы доказать неравенство $\varphi_T(n,k,i) \leqslant p(n,k,i)$, достаточно привести алгоритм расшифровки A для запросов типа T, такой, что для любой функции $f \in F(n,k,i)$ будет выполняться неравенство $q(A,f) \leqslant p(n,k,i)$. В то же время, чтобы доказать неравенство $\varphi_T(n,k,i) \geqslant p(n,k,i)$, достаточно показать, что какой бы алгоритм расшифровки A для запросов типа T ученик не использовал в игре с учителем, учитель для этого алгоритма загадает такую функцию $f \in F(n,k,i)$, что $q(A,f) \geqslant p(n,k,i)$. 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.

Будем считать, что ученику известны все три параметра: n, k и i.

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 писать $A(n)=o(B(n))$, если $\lim_{n\to\infty} \frac{A(n)}{B(n)}=0$, и $A(n) \lesssim B(n)$, если $\lim_{n\to\infty} \frac{A(n)}{B(n)} \leqslant 1$.

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

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

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

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

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

ность равна

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

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

Теорема 4. *Сложность расшифровки класса $F(n,k,0)$ запросами на сравнение равна*

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

Теорема 5. *При $2^n \bmod (k+1)=k$ сложность расшифровки класса $F(n,k,1)$ запросами на сравнение равна*

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

При $2^n \bmod (k+1)<k$ сложность расшифровки класса $F(n,k,1)$ запросами на сравнение удовлетворяет следующим ограничениям:

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

Теорема 6. *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:*

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

Запросы на значение, ограниченную и расширенную эквивалентность. Для удобства приведем в виде лемм формулировки теорем из работы [6], на которые мы будем ссылаться: лемма 1 соответствует теореме 1, лемма 2 — теореме 3, лемма 3 — теореме 2.

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

Лемма 2. *Сложность расшифровки класса $F(n,k,k)$ запросами на расширенную эквивалентность равна $\min(k,2^n-k)$.*

Лемма 3. *Сложность расшифровки класса $F(n,k,k)$ запросами на ограниченную эквивалентность равна 2^n-1.*

Доказательство теоремы 1. Случай $i=k$ следует из леммы 1. Нам осталось рассмотреть случай $i<k$.

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 доказана.

Доказательство теоремы 2. *Верхняя оценка.* Алгоритм расшифровки $F(n,k,k)$, используемый в доказательстве верхней оценки леммы 2, является и алгоритмом расшифровки $F(n,k,i)$. Приведем его здесь, поскольку он также пригодится для доказательства верхней оценки теоремы 3.

Пусть $k\leqslant 2^n-k$. В роли ученика в качестве первого запроса мы отправим учителю константу 0. В силу того что $k>0$, в ответ мы получим 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 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 $k>2^n-k$, тогда можно применять приведенный выше алгоритм (т.е. раскрывать за каждый запрос одну единицу). Но если мы применим его, заменив везде 0 на 1 (т.е. теперь будем раскрывать за каждый запрос нуль функции), то восстановим функцию за меньшее число запросов.

Нижняя оценка. В силу того что $F(n,k,k)\subseteq F(n,k,i)$, учитель может загадать любую функцию веса ровно k, поэтому нижняя оценка $\varphi_{XEQ}(n,k,i)$ не меньше нижней оценки для $\varphi_{XEQ}(n,k,k)$, следующей из леммы 2. Теорема доказана.

Доказательство теоремы 3. Случай $i = k$ является результатом леммы 3. Докажем оценку сложности расшифровки для случаев $i = 0$ и $i \in (0, k)$.

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

Случай $0 < i < k$. Зададим 2^n запросов. Пусть A — множество наборов с неизвестным значением загаданной функции в них; A_1 — наборы, значение функции в которых равно 1. Изначально A_1 — пустое множество, а A состоит из всех 2^n наборов. Для формирования очередного запроса выберем любой набор a из A. Тогда запрос — это функция, равная 1 на всех наборах из A_1 и на наборе a и равная 0 на всех остальных наборах. В ответ на запрос будет либо раскрыта какая-то новая единица b, отличная от a, и тогда удалим b из A и добавим в A_1, либо станет известно, что набор a на самом деле является нулем, и тогда удалим a из A. Следовательно, за каждый запрос мы узнаем значение функции ровно на одном наборе.

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.

Случай $i = 0$. На каждый из первых k запросов ученика будем возвращать в ответ любой набор, на котором функция, присланная учеником, равна 0. Тем самым за один запрос мы раскроем ученику одну единицу.

Случай $0 < i < k$. На каждый из первых $2^n - k$ запросов будем возвращать набор, на котором функция ученика равна 1. 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, 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. Теорема доказана.

Запросы 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.

Следующая лемма приводится для удобства и объединяет формулировки лемм 2 и 5 из работы [6].

Лемма 4. *Для того чтобы Q наборов объединить в компоненты связности размера не менее p, необходимо задать не менее $G(p - 1, Q) = (p - 1) \cdot [Q/p] + (Q \bmod p)$ запросов на сравнение.*

Доказательство теоремы 4. Верхняя оценка следует из адаптации леммы 1 статьи [6] к специфике класса $F(n, k, 0)$. Пусть $2^n = q \cdot (k + 1) + r, q \geqslant 0, r \in [0, k]$, тогда создадим q компонент связности размера $k + 1$ за $q \cdot k$ запросов. Значение на каждом из покрытых наборов в силу замечания 1 однозначно восстанавливается. Осталось восстановить значение на оставшихся r наборах. В упомянутой статье мы присоединяли с помощью $r - 1$ запросов на сравнение к существующей компоненте размера $k+1$ оставшиеся $r-1$ наборов и тем самым также однозначно восстанавливали значение на них. Далее по тому, нашли мы уже $k - 1$ или k единиц, делали вывод, является ли последний непокрытый набор единицей или нет. В классе же $F(n, k, 0)$ лежат как функции веса k, так и функции меньшего веса, поэтому если после покрытия $q \cdot (k+1)+r-1$ наборов будет найдено не k единиц, то для однозначного восстановления функции придется покрыть и последний набор, а значит, для класса $F(n, k, 0)$ верхняя оценка $k \cdot [2^n/(k + 1)] + \max(0, (2^n \bmod (k + 1)) - 1)$ из работы [6] заменяется на $k \cdot [2^n/(k + 1)] + (2^n \bmod (k + 1))$, 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 множество мощности 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 мощности строго больше k. Согласно лемме 4 это возможно сделать не менее чем за $k \cdot [2^n/(k+1)] + (2^n \bmod (k+1))$ запросов. Теорема доказана.

Лемма 5. *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 $r \in [0, k]$. 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 доказана.

Лемма 6. *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 доказана.

Доказательство теоремы 5. 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 формировании 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 формировании 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 4 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 5 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 6 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 6 выполнено $G(k, 2^n - 1) \geqslant G(k, 2^n) - 1$. Теорема доказана.

Лемма 7. *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 доказана.

Лемма 8. *Пусть k, n — целые числа, такие, что $0 < k < 2^n$, тогда справедливо неравенство* $2 \cdot \big[(2^n - (k+2))/3\big] + ((2^n - (k+2)) \bmod 3) > 2 \cdot \big[2^n/3\big] - \big]2/3 \cdot (k+3)\big[$.

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 неравенство имеет место. Лемма доказана.

Лемма 9. *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 доказана.

Лемма 10. *При $k \geqslant 2$ сложность расшифровки класса $F(n,k,k)$ запросами на сравнение не меньше* $2 \cdot \big[2^n/3\big] - \big]2/3 \cdot (k+3)\big[$.

Доказательство. Обозначим $Q = 2 \cdot \big[2^n/3\big] - \big]2/3 \cdot (k+3)\big[$. 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 $K \geqslant V - E \geqslant \big[1/3 \cdot 2^n\big] + \big]2/3 \cdot (k+3)\big[$. Значит, средний размер компоненты связности $V/K \leqslant 2^n / \big(\big[1/3 \cdot 2^n\big] + \big]2/3 \cdot (k+3)\big[\big) < 3$. 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 4, 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 7 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 $2 \cdot \big[(2^n - (k+2))/3\big] + ((2^n - (k+2)) \bmod 3) > 2 \cdot \big[2^n/3\big] - \big]2/3 \cdot (k+3)\big[$ согласно лемме 8. 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 все вершины одной компоненты связности принадлежат одному классу эквивалентности. 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 множествах размера 1 и 2. Это осуществимо в силу леммы 9, 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 связ-

ности 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 9, 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 9, 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 9, 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. Лемма доказана.

Перед тем как перейти к доказательству теоремы 6, приведем в виде леммы 11 формулировку теоремы 6 из работы [6], на которую мы будем опираться при доказательстве.

Лемма 11. *При $n > 6$ сложность расшифровки запросами на сравнение класса $F(n,3,3)$ равна* $2^n - \left]3/2 \cdot [2^n/5]\right[- \left[(2^n \bmod 5)/2\right]$.

Доказательство теоремы 6. Учитывая вложение $F(n,k,i) \subseteq F(n,k,0)$, по теореме 4 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 10 получаем $\varphi_{CQ}(n,k,i) \geqslant \varphi_{CQ}(n,k,k) \geqslant 2[2^n/3] - \left]2/3 \cdot (k+3)\right[$. 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 11 заключаем, что $\varphi_{CQ}(n,k,i) \geqslant \varphi_{CQ}(n,3,3) \geqslant 2^n - \left]3/2 \cdot [2^n/5]\right[- \left[(2^n \bmod 5)/2\right]$. Поэтому при $i \leqslant 3 \leqslant k$ справедлива оценка $7/10 \cdot 2^n \lesssim \varphi_{CQ}(n,k,i)$. Теорема доказана.

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

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

СПИСОК ЛИТЕРАТУРЫ

1. *Angluin D.* Queries and concept learning // Mach. Learning. 1988. **2**. 319–342.
2. *Angluin D.* Queries revisited // Theor. Comput. Sci. 2001. **313**, N 2. 175–194.
3. *Быстрыгова А.В.* Параметро-эффективная расшифровка булевых функций из замкнутых классов Поста // Дискретн. матем. 2019. **31**, № 2. 34–58.
4. *Bistrigova A.V.* Attribute-efficient learning of Boolean functions from Post closed classes // Discrete Math. and Appl. 2020. **30**, N 5. 285–301.
5. *Быстрыгова А.В.* Запросы на сравнение в задаче параметро-эффективной расшифровки булевых функций // Интеллектуальные системы. Теория и приложения. 2019. **23**, № 4. 115–124.
6. *Быстрыгова А.В.* Расшифровка булевых функций фиксированного веса // Интеллектуальные системы. Теория и приложения. 2020. **24**, № 3. 63–96.

Поступила в редакцию
23.11.2020