Расшифровка арифметических сумм монотонных конъюнкций

З. А. Ниязова

В работе рассматривается задача расшифровки арифметических сумм монотонных конъюнкций, определенных на n-мерном булевом кубе, в модели точной расшифровки при помощи запросов на значение функции. Предложен алгоритм расшифровки, на основе которого получена верхняя оценка сложности расшифровки. Так же предложен алгоритм ответов на запросы, с помощью которого были получены нижние оценки сложности расшифровки исследуемого класса функций для малого числа монотонных конъюнкций.

Ключевые слова: расшифровка дискретных функций, суммы мототонных конъюнкций.

Введение

Если перед нами имеется некоторое физическое явление или техническое устройство и мы хотим его понять, то мы пытаемся на него как-то воздействовать и по его реакциями делаем какие-то заключения. Если техническое устройство является имеет конечную память, то есть является конечным автоматом, то исследованием таких устройств занимается теория автоматов [1]. Среди последних работ по теории автоматов можно отметить [2–16]. При этом в теории автоматов есть специальный раздел, который занимается расшифровкой функций с памятью, называемый экспериментами с автоматами [17–21].

Если исследуемой устройство реализует функцию без памяти, то раздел науки, занимающийся анализом таких устройств называется расшифровкой функций, а в зарубежной литературе Machine Learning (машинное обучение) [22, 23].

В работе исследуется расшифровка дискретных функций, определенных на n-мерном булевом кубе. Неформально задача расшифровки функции из некоторого класса состоит в построении алгоритмов игры между учителем и учеником. Учитель загадывает некоторую функцию из некоторого известного ученику класса. Ученик с помощью алгоритма расшифровки пытается отгадать загаданную функцию, т.е полностью восстановить её таблицу значений, сделав минимальное число запросов на значение функции. При этом мерой сложности служит число запросов на значение функции, требуемое алгоритму для расшифровки данной функции.

Задача расшифровки монотонных булевых функций исследовалась В.К. Коробковым [24]. В 1966 году Ж.Ансель показал [25], что сложность расшифровки монотонных функций в модели точной расшифровки при помощи запросов на значение функции в точности равна $C_n^{[\frac{n}{2}]} + C_n^{[\frac{n}{2}]+1}$. Расшифровкой монотонных функций с несущественными переменными занимались П.Домашке и В.В.Осокин [26–28].

Среди последних работ по расшифровке функций можно отметить [29–38].

В представленной работе нас будет интересовать класс функций сумм монотонных конъюнкций. Значение функции из данного класса на некотором наборе определяется арифметической суммой нижних единиц функции на подкубе, состоящем из наборов не больших чем данный набор.

Похожая задача рассматривалась А.Накамура и Н.Абе [39], в работе которых значение функции определялось линейной комбинацией нижних единиц, где коэффициенты при слагаемых являлись вещественными числами. В случае с положительными вещественными коэффициентами Накамура и Абе предложили алгоритм расшифровки и верхнюю оценку его сложности. Так же ими для данного же случая была получена нижняя оценка.

Таким образом, исследуемая в настоящей работе задача представляет собой частный случай более общей задачи рассмотренной в работе [39], с коэффициентами при слагаемых равными единице.

Учитывая особенности данного класса в предлагаемой работе был предложен алгоритм расшифровки, для которого независимо от результатов работы [39] получена верхняя оценка сложности расшифровки . Так же была доказана нижняя оценка сложности расшифровки исследуемого класса функций для малого числа монотонных конъюнкций, с помощью алгоритма ответов на запросы. Часть результатов данной работы была анонсирована в [40].

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

Основные понятия и формулировка результатов

Обозначим через $E^n = \{(\alpha_1, \ldots, \alpha_n) : \alpha_i \in \{0,1\}, i = 1,2,\ldots,b\}$ — n-мерный булев куб.

Для наборов $\alpha = (\alpha_1, \ldots, \alpha_n), \beta = (\beta_1, \ldots, \beta_n) \in E^n$ введем отношение $\alpha \leq \beta$, если $\alpha_i \leq \beta_i$ для всех $i = 1, \ldots, n$. Будем писать $\alpha < \beta$, если $\alpha \leq \beta$ и $\alpha \neq \beta$.

Если для двух наборов α и β выполнено либо $\alpha \leq \beta$, либо $\beta \leq \alpha$, то мы называем наборы α и β *сравнимыми*, в противном случае — *несравнимыми*.

Функция f называется монотонной, если для любых наборов α) и β, таких что $\alpha \leq \beta$ справедливо $f(\alpha) \leq f(\beta)$.

Набор $\alpha = (\alpha_1, \ldots, \alpha_n)$ называется нижней единицей монотонной функции f, если $f(\alpha) = 1$ и для любого набора β такого, что $\beta < \alpha$ справедливо $f(\beta) = 0$.

Каждому набору $\alpha \in E^n$ можно сопоставим монотонную конъюнкцию

$$K_\alpha(x_1, \ldots, x_n) = \&_{i:\alpha_i=1} x_i$$

Будем говорить, что набор β удовлетворяет конъюнкции, соответствующей набору α, если $K_\alpha(\beta) = 1$, то есть если $\alpha \leq \beta$.

$\mathcal{B}_n$ — множество всех подмножеств булевого куба E^n , состоящих из несравнимых наборов. Понятно, что $|\mathcal{B}_n|$ равно количеству монотонных функций, поскольку каждая монотонная функция однозначно определяется своим множеством нижних единиц. Так, если $B \in \mathcal{B}_n$ — некоторое множество несравнимых наборов, то монотонная функция с множеством нижних единиц B задается формулой $\vee_{\alpha \in B} K_\alpha$.

Если $B \in \mathcal{B}_n$, то функцию $f_B : E^n \to \mathbb{N}$, определяемую соотношением $f_B(x) = |B_x|$, будем называть *арифметической суммой монотонных конъюнкций с нижними единицами из* B.

В работе исследуется класс арифметических сумм монотонных конъюнкций

$$\Phi_n = \{f_B : B \in \mathcal{B}_n\}.$$

Через $\Phi_{n,p}$ обозначим класс арифметических сумм монотонных конъюнкций из $\Phi_{n,p}$ с не более чем p нижними единицами.

Под *запросом* на значение функции будем понимать набор значений переменных функции, под *ответом* на запрос — значение функции на этом наборе. Под *алгоритмом расшифровки* будем понимать условный эксперимент, который последовательно генерирует запросы на значение функции в зависимости от ответов на предыдущие запросы. Будем говорить, что *алгоритм расшифровывает функцию* f *из* Φ_n, если значения функции на наборах, сгенерированных условным экспериментом, однозначно определяют таблицу значений функции f при условии, что $f \in \Phi_n$. Скажем, что *алгоритм расшифровывает класс функций* Φ_n, если он расшифровывает любую функцию из Φ_n при условии, что он получает n в виде входного параметра. Обозначим множество алгоритмов расшифровки класса Φ_n через $\mathcal{A}(\Phi_n)$.

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

Положим

$$\varphi(n, p) = \min_{A \in \mathcal{A}(\Phi_n)} \max_{f \in \Phi_{n,p}} \varphi(A, f).$$

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

В работе предложен алгоритм расшифровки, обеспечивший следующую верхнюю оценку.

Теорема 1. *Имеет место неравенство*

$$\varphi(n,p) \leqslant \begin{cases} np - p - p\lfloor \log_2 p \rfloor + 2^{\lfloor \log_2 p \rfloor + 1}, 0 < p < C_n^{[\frac{n}{2}]}, \\ 1, p = 0. \end{cases}$$

Данная оценка согласуется с оценкой из работы [39].

Предложен алгоритм ответов на запросы алгоритмов расшифровки для малого числа монотонных конъюнкций который заставляет делать алгоритмы расшифровки как можно больше запросов, с помощью которого получена следующая нижняя оценка.

Теорема 2. *При* $p < n+1$ *справедливо* $\varphi(n,p) \geqslant np - \frac{(p-1)p}{2} + 1$.

Следствие 1. *Справедливы равенства* $\varphi(n,1) = n+1$, $\varphi(n,2) = 2n$, $\varphi(n,3) = 3n-2$.

Следствие 2. *Если* $p = \bar{o}(n)$, *то* $\varphi(n,p) \sim np$.

Следствие 3. *Если* $p = \underline{O}(n)$, *то* $\varphi(n,p) = \underline{O}(n^2)$.

Верхняя оценка

Пусть B — множество несравнимых n-мерных векторов, координаты которых неопределены. B соответствует множеству нижних единиц функции.

Задачу расшифровки можно считать разрешенной, если определен каждый элемент из множества B. Для этого необходимо восстановить каждую компоненту каждого n-мерного вектора из данного множества B.

Пусть $\beta = (\beta_1, \ldots, \beta_n)$, $\beta \in E^n$. Обозначим $0_\beta = \{i : \beta_i = 0\}$ — номера нулевых компонент набора β, $1_\beta = \{i : \beta_i = 1\}$ — номера единичных компонент набора β.

Если $\beta \in E^n$, $B \in \mathcal{B}_n$, то обозначим $B_\beta = \{\alpha \in B : \alpha \leq \beta\}$.

Справедливо утверждение.

Лемма 1. *Если* $\beta \in E^n$, $B \in \mathcal{B}_n$, $\alpha \in B_\beta$, *то* $0_\beta \subseteq 0_\alpha$.

Доказательство. Предположим противное, пусть $\beta_j = 0$, а при этом $\alpha_j^i = 1$, тогда $K_\alpha(\beta_1, \ldots, \beta_n) = \&_{i:\alpha_i=1}\beta_i = 0$. Противоречие. Следовательно, из $\beta_j = 0$ следует $\alpha_j = 0$. ■

Данное утверждение позволяет сделать следующий вывод. Если $f_B(\beta) = k$, то в B должно встретиться k наборов, образующих множество B_β, в которых нули стоят в тех же позициях, что и у набора β.

Рассмотрим дополнение к множеству B_β, обозначим его через B'_β, $B'_\beta = B \setminus B_\beta$. Для каждого $\gamma \in B'_\beta$ справедливо $K_\gamma(\beta) = 0$. Это означает, что одна из переменных входящая в монотонную конъюнкцию набора β равна нулю, в то время как данная переменная входящая в конъюнкцию нижней единицы из множества B'_β равна единице.

Данное соображение можно сформулировать в виде утверждения.

Лемма 2. *Если* $\beta \in E^n$, $B \in \mathcal{B}_n$, $\gamma \in B'_\beta$, *то* $0_\beta \cap 1_\gamma \neq \emptyset$.

Отсюда следует, что набор β содержит ровно одну нулевую компоненту, то у всех наборов из B'_β в той же компоненте должна стоять единица.

При этом ясно, что если набор β содержит более чем одну нулевую координату, нельзя однозначно определить какая именно из литер входящих в конъюнкцию $K_\gamma(x_1, \ldots, x_n) = \&_{i:\gamma_i=1}x_i$ равна нулю.

Через 0^j обозначим набор из $n-1$ слоя E^n , в котором j компонента равна нулю, $1 \leq j \leq n$. Пусть $\beta = 0^j$, т.е. $\beta_j = 0$ и $\beta_i = 1$ при $i \neq j$. Опросим набор β, пусть на данном наборе функция будет принимать значение равное k, $k > 0$, т.е. $f_B(\beta) = k$, тогда мы сможем однозначно зафиксировать j координату для каждого из набора, которые принадлежат множествам B_β и B'_β:

1) $\alpha_j^i = 0$, для всех $\alpha^i \in B_\beta$, где $i = \overline{1,k}$,

2) $\gamma_j^i = 1$, для всех $\gamma^i \in B'_\beta$, где $i = \overline{1, |B| - k}$.

В итоге имеем два подкуба $E_{1,j}^{n-1}$ и $E_{0,j}^{n-1}$, объединение которых есть куб E^n. Наборами куба $E_{1,j}^{n-1}$ ($E_{0,j}^{n-1}$) являются все наборы куба E^n с j единичной (нулевой) координатой. Таким образом, множество B_β можно представить как $B_\beta = B \bigcap E_{0,j}^n$, соответственно $B'_\beta = B \bigcap E_{1,j}^n$.

Лемма 3. *Если* $\beta \in E^n$, $B \in \mathcal{B}_n$, $f_B(\beta) = 0$, *то для любого* $\alpha \in B$ *справедливо* $\sum\limits_{i \in 0_\beta} \alpha_i > 0$.

Доказательство. Так как $f_B(\beta) = 0$, то для любого $\alpha \in B$ верно $K_\alpha(\beta) = 0$, т.е. $\alpha \in B'_\beta$. Отсюда, учитывая лемму 2, имеем для любого $\alpha \in B$ справедливо $\sum\limits_{i \in 0_\beta} \alpha_i > 0$. ■

Лемма 4. *Если* $\beta \in E^n$, $B \in \mathcal{B}_n$, $f_B(\beta) = k$, $k > 0$, *то для любых* $\alpha \in B_\beta$ *и* $i \in 0_\beta$ *выполняется* $\alpha_i = 0$, *а для любого* $\gamma \in B'_\beta$ *выполнено* $\sum\limits_{i \in 0_\beta} \gamma_i > 0$.

Данное утверждение является следствием лемм 1 и 2.

Приведем алгоритм расшифровки для класса функций $\Phi_{n,1}$, назовем его $\mathcal{I}_1$. Класс $\Phi_{n,1}$ включает в себя функцию тождественный ноль и функции, которые задаются единственной нижней единицей.

Алгоритм $\mathcal{I}_1$

1) Опросим значение функции f_B на наборе $(11\ldots11)$, пусть $f_B(11\ldots11) = p$. Если $p = 0$, то данная функция тождественный ноль. Расшифровка окончена.

2) Иначе значение $p = 1$, и у функции f_B есть единственная нижняя единица, которую обозначим $\alpha = (\alpha_1, \ldots, \alpha_n)$. Опросим набор $0^n = (111\ldots110)$. Возможны два варианта:

   а) $f_B(0^n) = 0$, тогда из леммы 3 следует, что $\alpha_n = 1$. Далее будем расшифровывать функцию $f_B^1(x_1, ..., x_{n-1}) = f_B(x_1, ..., x_{n-1}, 1)$, учитывая опрошенные наборы.

б) $f_B(0^n) = 1$, тогда из леммы 4 следует, что $\alpha_n = 0$, и мы расшифровываем функцию $f_B^0(x_1, ..., x_{n-1}) = f_B(x_1, ..., x_{n-1}, 0)$ по тому же алгоритму рекурсивно.

Опишем алгоритм для расшифровки функций в общем случае. Обозначим его через $\mathcal{I}$, данный алгоритм будет определять значения функции $f_B(x_1, \ldots, x_n)$ из Φ_n, а точнее будет определять множество B.

Алгоритм $\mathcal{I}$

1) Определим значение функции $f_B(x_1, \ldots, x_n)$ на наборе $(111\ldots11)$. Пусть значение функции на данном наборе равно p. Если $p = 0$, то функция полностью расшифрована. Если $p = 1$ и размерность куба равна единице, то функция так же полностью расшифрована. Если $p = C_n^{[\frac{n}{2}]}$, и n — четное, то B совпадает со средним слоем куба и функция полностью определена, если n — нечетное, то B один из двух средних слоев. Опросим значение функции на любом наборе из нижнего среднего слоя и полностью определим функцию. Иначе переходим к пункту 2.

2) Далее рекурсивно применяем алгоритм $\mathcal{I}$ к кубу меньшей размерности, где на n месте находится ноль, т.е будем расшифровывать функцию $f_B(x_1, \ldots, x_{n-1}, 0)$. При этом восстанавливаем литеры нижних единиц.

3) Далее запускаем алгоритм $\mathcal{I}$ к кубу $x_n = 1$ и расшифровываем функцию $f_B^1(x_1, \ldots, x_{n-1}) = f_B(x_1, \ldots, x_{n-1}, 1) - f_B(x_1, \ldots, x_{n-1}, 0)$. При этом значение функции $f_B^1(11\ldots1)$ мы уже знаем:

$$f_B^1(11\ldots1) = f_B(11\ldots1) - f_B(11\ldots10).$$

Обозначим

$$\phi(\mathcal{I}, n, p) = \max_{f \in \Phi_{n,p}} \varphi(\mathcal{I}, f).$$

Предположим на верхнем наборе куба E^n значение функции равно p. Далее согласно алгоритму $\mathcal{I}$ мы опрашиваем набор

$(11\ldots10)$, предположим, что значение функции на данном наборе будет равно q. Сложность расшифровки в подкубе $x_n = 0$ будет равна $\phi(\mathcal{I}, n-1, q)$, а в подкубе $x_n = 1$ соответственно $\phi(\mathcal{I}, n-1, p-q)$, при этом учитываем, что набор $(11\ldots11)$ уже опрошен. Необходимо выбрать q, таким образом, чтобы сумма $\phi(\mathcal{I}, n-1, q) + \phi(\mathcal{I}, n-1, p-q)$ достигала максимального значения.

Таким образом, можем задать сложность алгоритма $\mathcal{I}$ в виде рекурсивной формулы

$$\begin{cases} \phi(\mathcal{I}, 0, 1) = 1, \\ \phi(\mathcal{I}, n, 0) = 1, \\ \phi(\mathcal{I}, n, p) = \max_{0 \le q \le [n/2]}(\phi(\mathcal{I}, n-1, q) + \phi(\mathcal{I}, n-1, p-q)). \end{cases}$$

Покажем, что максимум достигается при $q = [\frac{p}{2}]$.

Лемма 5. *Справедливо равенство*

$$\begin{aligned} \phi(\mathcal{I}, n, p) &= \max_{0 \le q \le [p/2]}(\phi(\mathcal{I}, n-1, q) + \phi(\mathcal{I}, n-1, p-q)) = \\ &= \phi(\mathcal{I}, n-1, \lfloor p/2 \rfloor) + \phi(\mathcal{I}, n-1, \lceil p/2 \rceil). \end{aligned}$$

Доказательство. Проведем доказательство по индукции.

Базис индукции. Для $n = 0, 1, 2$ лемма верна.

Индуктивный переход.

Предположим, что лемма верна для n. Т.е справедливо равенство

$$\phi(\mathcal{I}, n-1, q) + \phi(\mathcal{I}, n-1, p-q) \le \phi(\mathcal{I}, n-1, \lfloor \frac{p}{2} \rfloor) + \phi(\mathcal{I}, n-1, \lceil \frac{p}{2} \rceil)$$

для $0 \le q \le [p/2]$.

Докажем утверждение для $n+1$.

По предположению индукции справедливо

$$\begin{aligned} &\phi(\mathcal{I}, n, q) + \phi(\mathcal{I}, n, p-q) \le \phi(\mathcal{I}, n-1, \lfloor \frac{q}{2} \rfloor) + \\ &+ \phi(\mathcal{I}, n-1, \lceil \frac{q}{2} \rceil) + \phi(\mathcal{I}, n-1, \lfloor \frac{p-q}{2} \rfloor) + \phi(\mathcal{I}, n-1, \lceil \frac{p-q}{2} \rceil). \end{aligned}$$

1) Пусть $p = 2k$, $q = 2m$. Тогда имеем

$$\begin{aligned}\phi(\mathcal{I}, n, q) + \phi(\mathcal{I}, n, p-q) \le \phi(\mathcal{I}, n-1, m) + \phi(\mathcal{I}, n-1, m) + \\ + \phi(\mathcal{I}, n-1, k-m) + \phi(\mathcal{I}, n-1, k-m) = \\ = \phi(\mathcal{I}, n, k) + \phi(\mathcal{I}, n, k) = \phi(\mathcal{I}, n, \frac{p}{2}) + \phi(\mathcal{I}, n, \frac{p}{2}).\end{aligned}$$

2) Для $p = 2k+1$, $q = 2m+1$ получим

$$\begin{aligned}\phi(\mathcal{I}, n, q) + \phi(\mathcal{I}, n, p-q) \le \phi(\mathcal{I}, n-1, [\frac{2m+1}{2}]) + \\ + \phi(\mathcal{I}, n-1, [\frac{2m+1}{2}]+1) + + \phi(\mathcal{I}, n-1, [\frac{2k+1-2m-1}{2}]) + \\ + \phi(\mathcal{I}, n-1, [\frac{2k+1-2m-1}{2}]) = \phi(\mathcal{I}, n-1, m) + \\ + \phi(\mathcal{I}, n-1, m+1) + \phi(\mathcal{I}, n-1, k-m) + \\ + \phi(\mathcal{I}, n-1, k-m) = \phi(\mathcal{I}, n, k) + \phi(\mathcal{I}, n, k+1) = \\ = \phi(\mathcal{I}, n, [\frac{p}{2}]) + \phi(\mathcal{I}, n, [\frac{p}{2}]+1).\end{aligned}$$

3) Пусть $p = 2k+1$, $q = 2m$, тогда

$$\begin{aligned}\phi(\mathcal{I}, n, q) + \phi(\mathcal{I}, n, p-q) \le \phi(\mathcal{I}, n-1, [\frac{2m}{2}]) + \\ + \phi(\mathcal{I}, n-1, [\frac{2m}{2}]) + \phi(\mathcal{I}, n-1, [\frac{2k+1-2m}{2}]) + \\ + \phi(\mathcal{I}, n-1, [\frac{2k+1-2m}{2}]+1) = \phi(\mathcal{I}, n-1, m) + \\ + \phi(\mathcal{I}, n-1, m) + \phi(\mathcal{I}, n-1, k-m) + \\ + \phi(\mathcal{I}, n-1, k-m+1) = \phi(\mathcal{I}, n, k) + \phi(\mathcal{I}, n, k+1) = \\ = \phi(\mathcal{I}, n, [\frac{p}{2}]) + \phi(\mathcal{I}, n, [\frac{p}{2}]+1).\end{aligned}$$

4) Для $p = 2k$, $q = 2m + 1$ имеем

$$\phi(\mathcal{I}, n, q) + \phi(\mathcal{I}, n, p - q) \le \phi(\mathcal{I}, n-1, [\frac{2m+1}{2}]) +$$
$$+ \phi(\mathcal{I}, n-1, [\frac{2m+1}{2}] + 1) + \phi(\mathcal{I}, n-1, [\frac{2k-2m-1}{2}]) +$$
$$+ \phi(\mathcal{I}, n-1, [\frac{2k-2m-1}{2}] + 1) = \phi(\mathcal{I}, n-1, m) +$$
$$+ \phi(\mathcal{I}, n-1, m+1) + \phi(\mathcal{I}, n-1, k-m-1) + \phi(\mathcal{I}, n-1, k-m) =$$
$$= \phi(\mathcal{I}, n, k) + \phi(\mathcal{I}, n, k) = \phi(\mathcal{I}, n, \frac{p}{2}) + \phi(\mathcal{I}, n, \frac{p}{2}).$$

Тем самым доказательство леммы завершается. ■

Выведем аналитическую формулу из полученного рекуррентного соотношения.

Лемма 6. *Справедливо равенство*

$$\phi(\mathcal{I}, n, p) = \begin{cases} np - p - p\lfloor \log_2 p \rfloor + 2^{\lfloor \log_2 p \rfloor + 1}, & 0 < p < C_n^{[\frac{n}{2}]}, \\ 1, \quad p = 0. \end{cases}$$

Доказательство. Распишем рекуррентную формулу

$$\phi(\mathcal{I}, n, p) = \phi(\mathcal{I}, n-1, \lfloor \frac{p}{2} \rfloor) + \phi(\mathcal{I}, n-1, \lceil \frac{p}{2} \rceil) =$$
$$= \phi(\mathcal{I}, n-2, \lfloor \frac{\lfloor \frac{p}{2} \rfloor}{2} \rfloor) + \phi(\mathcal{I}, n-2, \lceil \frac{\lceil \frac{p}{2} \rceil}{2} \rceil) +$$
$$+ \phi(\mathcal{I}, n-2, \lceil \frac{\lfloor \frac{p}{2} \rfloor}{2} \rceil) + \phi(\mathcal{I}, n-2, \lfloor \frac{\lceil \frac{p}{2} \rceil}{2} \rfloor) =$$
$$= \ldots = \phi(\mathcal{I}, n-k_1, 1) + \phi(\mathcal{I}, n-k_2, 1) + \ldots + \phi(\mathcal{I}, n-k_p, 1).$$

Рекурсивное дерево будет выглядеть, как изображено на рисунке 1.

Используя соотношение $\phi(\mathcal{I}, n, 1) = n + 1$, получим

$$\phi(\mathcal{I}, n, p) = (n - k_1) + 1 + \ldots (n - k_p) + 1 = np + p - k_1 \ldots - k_p$$

Рассмотрим бинарное дерево, изображенное на рисунке 2.

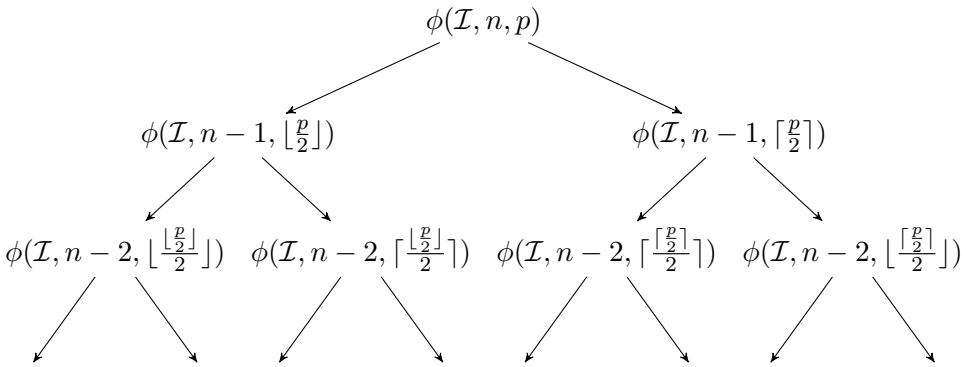

Рис. 1: Рекурсивное дерево.

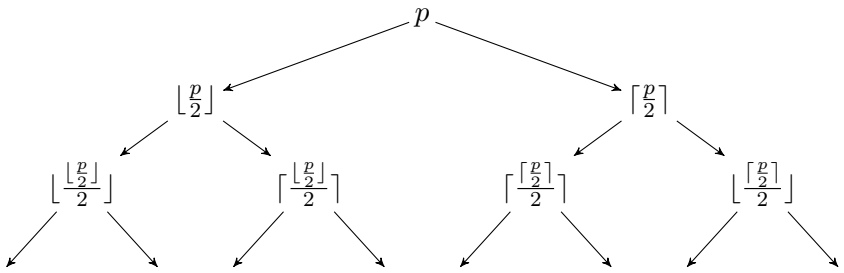

Рис. 2: Бинарное дерево.

Значения на узлах рассматриваемого дерева будут соответствовать значениям функции на опрашиваемых наборах булевого куба. Значения на узлах, являющихся листами данного дерева, будут равны единице и при этом уровень каждого листа будет отличаться не более, чем на единицу. Дерево имеет p листов.

Уровню k соответствует 2^k узлов. То есть $k_1, ..., k_p$ равны уровню каждого листа бинарного дерева, изображенного на рисунке 2.

Если p степень двойки, получаем

$$k_1 + \ldots + k_p = \log_2 p + \ldots + \log_2 p = p \log_2 p.$$

Следовательно, $\phi(\mathcal{I}, n, p) = np + p - p \log_2 p$.

Выведем формулу в общем случае. Формула выглядит следующим образом

$$\phi(\mathcal{I}, n, p) = (n - k_1) + 1 + \ldots (n - k_p) + 1 = np + p - k_1 \ldots - k_p.$$

Обозначим сумму отрицательных членов через K, т.е $K = k_1 + \ldots + k_p$. Определим K.

Уровень каждого листа дерева, изображенного на рисунке 2, будет равен либо $\lfloor \log_2 p \rfloor$, либо $\lfloor \log_2 p \rfloor + 1$. Так как строго бинарное дерево с n листами всегда содержит $2n - 1$ узлов, имеем

$$2p - 1 = 2^0 + 2^1 + 2^2 + \ldots + 2^{\lfloor \log_2 p \rfloor} + x,$$

где x — число узлов на уровне $\lfloor \log_2 p \rfloor + 1$.

Откуда

$$x = 2p - 1 - 2^0 - 2^1 - 2^2 - \ldots - 2^{\lfloor \log_2 p \rfloor},$$

$$x = 2p - 2^{\lfloor \log_2 p \rfloor + 1}.$$

В общем виде имеем

$$K = (\lfloor \log_2 p \rfloor + 1)(2p - 2^{\lfloor \log_2 p \rfloor + 1}) + \lfloor \log_2 p \rfloor (p - (2p - 2^{\lfloor \log_2 p \rfloor + 1})) =$$
$$= \lfloor \log_2 p \rfloor 2p + 2p - \lfloor \log_2 p \rfloor 2^{\lfloor \log_2 p \rfloor + 1} - 2^{\lfloor \log_2 p \rfloor + 1} + \lfloor \log_2 p \rfloor 2^{\lfloor \log_2 p \rfloor + 1} -$$
$$- p \lfloor \log_2 p \rfloor = p \lfloor \log_2 p \rfloor + 2p - 2^{\lfloor \log_2 p \rfloor + 1}.$$

Таким образом,

$$\phi(\mathcal{I}, n, p) = np + p - p\lfloor \log_2 p \rfloor - 2p + 2^{\lfloor \log_2 p \rfloor + 1} = \\ = np - p - p\lfloor \log_2 p \rfloor + 2^{\lfloor \log_2 p \rfloor + 1}.$$

Лемма доказана. ■

Теорема 1 является следствием леммы 6 и неравенства $\varphi(n, p) \leq \phi(\mathcal{I}, n, p)$.

Нижняя оценка

Задача учителя заключается в построении алгоритма ответов на запросы на значение функции. При этом выбирать ответы на запросы необходимо таким образом, чтобы максимизировать общее число запросов на значение функции, которое должен сделать алгоритм расшифровки, для однозначного определения данной функции. То есть каждый раз необходимо отвечать так, чтобы алгоритм получил как можно меньше информации о наборах, являющихся нижними единицами.

Обозначим множество нижних единиц, задающих функцию следующим образом: $B = \{\alpha^i = (\alpha^i_1, \ldots, \alpha^i_n) : i = 1, 2, \ldots, p\}$. Считаем, что множество нижних единиц восстановлено, если восстановлены каждая из координат векторов α^i.

Пусть есть некий алгоритм ответов на запросы $\mathcal{W}_1$, который для каждого запроса на значение функции на некотором наборе, возвращать значение функции, учитывая при этом предыдущие запросы. На каждом шаге алгоритму $\mathcal{W}_1$ необходимо следить за тем, чтобы выполнялись условия лемм 3 и 4, при этом каждый ответ на вопрос можно вызвать фиксацию некоторых компонент векторов из B.

Построим алгоритм ответов на запросы функции для классов $\Phi_{n,1}$ и $\Phi_{n,2}$ тем самым докажем нижнюю оценку.

Лемма 7. *Имеет место неравенство* $\varphi(n, 1) \geq n + 1$.

Доказательство. Алгоритм ответов на запросы будет “прятать” искомую функцию в множестве

$$A = \{0, x_1 \ldots x_{n-1}, x_1 \ldots x_{n-2} x_n, x_2 \ldots x_n, x_1 \ldots x_n\}.$$

Все эти функции на наборах ниже $(n-1)$-го слоя принимают значение 0, поэтому для любого запроса β, где $|1_\beta| < n-1$ (т.е. β лежит ниже $(n-1)$-го слоя), выдаем ответ 0. В этом случае алгоритм расшифровки ничего не узнает о расшифровываемой функции, и множество A не меняется.

Если алгритм расшифровки опросил набор $\beta = (11\ldots11)$, т.е. $|0_\beta| = 0$, то выдаем ответ 1, тем самым исключаем функцию 0 из множества A, поскольку только функция 0 принимает значени 0 на наборе $(11\ldots11)$.

Если алгоритм расшифровки опрашивает набор $\beta = 0^j$ из $(n-1)$-го слоя, то выдаем ответ 0 и, тем самым, исключаем функцию $x_1\ldots x_{j-1}x_{j+1}\ldots x_n$ из множества A, поскольку только эта функция принимает значение 1 на наборе 0^j.

Тем самым любой запрос из n-го или $(n-1)$-го слоя исключают из множества A ровно одну функцию, т.е. опросить не все наборы из n-го и $(n-1)$-го слоя, то в множестве A останется по крайней мере 2 функции, и алгоритм расшифровки не будет знать, какая из этих 2-х функций искомая.

Лемма доказана. ■

Лемма 8. *Справедливо неравенство* $\varphi(n,2) \geq 2n$.

Доказательство. Пусть $B = \{\alpha^1 = (\alpha^1_1,\ldots,\alpha^1_n), \alpha^2 = (\alpha^2_1,\ldots,\alpha^2_n)\}$ — множество нижних единиц, с помощью которых задается функция. На данном этапе эта информация недоступна алгоритму расшифровки, для этого необходимо будет опросить вершину куба. Поэтому считаем, что набор $(11,\ldots,11)$ опрошен изначально. Покажем, что алгоритму расшифровки придется сделать как минимум еще $2n-1$ запросов на значение функции, чтобы однозначно восстановить все нижние единицы.

Если $|0_\beta| > 1$, то возвращаем значение функции равное нулю, откуда согласно лемме 3 справедливы неравенства

$$\begin{cases} \sum\limits_{i\in 0_\beta} \alpha^1_i > 0, \\ \sum\limits_{i\in 0_\beta} \alpha^2_i > 0. \end{cases} \tag{1}$$

В этом случае, мы не позволим алгоритму расшифровки однозначно зафиксировать литеры α^1 и α^2, для этого будут необходимы дополнительные запросы на значение функции.

Если алгоритм опрашивает наборы на $n-1$ слое, будем формировать значение функции равное единице. Согласно 4, если $f_B(0^j) = 1$, то справедливо $\alpha_j^1 + \alpha_j^2 = 1$. Если ни одна из литер α^1, α^2 не определена, можем принять на первом шаге $\alpha_j^1 = 1, \alpha_j^2 = 0$. Иначе чтобы решить $\alpha_j^1 + \alpha_j^2 = 1$, необходимо будет сделать дополнительный запрос на значение функции. Чтобы однозначно зафиксировать координаты нижних единиц алгоритму потребуется опросить $n-1$ слой, и сделать дополнительно $n-1$ запросов, чтобы решить уравнения вида $\alpha_j^1+\alpha_j^2 = 1$. Для этого нужно опросить наборы вида $0^{i,j}$ — набор в котором ровно 2 нуля на i-м и j-м месте.

Так если предположить, что уже опрошен набор 0^i, причем $\alpha_i^1 = 0, \alpha_i^2 = 1$, и опрошен набор $0^{i,j}$ и $f_B(0^{i,j}) = 0$, то есть

$$\begin{cases} \alpha_i^1 + \alpha_j^1 > 0, \\ \alpha_i^2 + \alpha_j^2 > 0, \end{cases} \tag{2}$$

то, чтобы однозначно восстановить α_j^2, необходимо будет опросить еще и набор - 0^j, иначе α_j^2 может принимать значение как ноль, так и один.

Отвечая подобным способом, мы каждый раз позволяем фиксировать алгоритму одну координату нижней единицы, что зафиксировать литеры α^1 алгоритму расшифровки придется опросить минимум n наборов, а для α^2 минимум $n-1$ наборов, так как она будет находится в другом подкубе $E_{0,j}^{n-1}$. ■

Постороим алгоритм ответов на запросы $\mathcal{W}_1$ для класса $\Phi_{n,p}$, где $p < n$.

Договоримся считать, что первый запрос, который делает алгоритм расшифровки, это запрос $(1, 1, \ldots, 1)$, который дает информацию о количестве нижних единиц функции, и пусть эточисло равно p. Поэтому алгоритм ответов на запросы будет иметь дело с множеством из p векторов $B = \{\alpha^i = (\alpha_1^i, \ldots, \alpha_n^i) : i = 1, \ldots, p\}$, интерпретируемых как нижние единицы функции. Каждая компонента этих векторов будет принимать значения

из множества $\{*, *0, *1, 0, 1\}$. Символ $*$ будет пониматься как компонента не известна, $*0$ и $*1$ будут интерпретироваться как условные ноль и единица соответственно, а 0 и 1 будут означать, что значение компоненты полностью определено. В начальный момент все p векторов полностью не известны, т.е. $\alpha^i = (*, \ldots, *)$, $i = 1, \ldots, p$. Алгоритм ответов на запросы старается отвечать так, чтобы после ответа как можно меньшее количество компонент векторов из множества B стало известно. Когда все компоненты становятся известными, алгоритм прекращает свою работу, поскольку функция полностью определена.

Идея алгоритма ответов на запросы будет состоять в следующем. Если поступает запрос β из $(n-1)$-го слоя, т.е. $\beta = 0^i$, то стараемся ответить 1, если это ничему не противоречит. Для запросов ниже $(n-1)$-го слоя стараемся ответить 0, если это ничему не противоречит. Противоречия могут возникнуть из-за того, что желаемое значение функции может не согласовываться с уже известными на данный момент компонентами векторов α^i, $i = 1, \ldots, p$.

Прежде чем строго определять алгоритм ответов на запросы, определин две вспомогательные процедуры.

Процедура ПроверкаЗначенияНоль будет проверять, можно ли установить значение искомой функции на опрашиваемом запросе в 0, и если да, то вносит в значения векторов из B изменения, навязываемые данным значением функции. На вход процедуры ПроверкаЗначенияНоль поступает опрашиваемый набор β и один из векторов $\alpha^i = (\alpha^i_1, \ldots, \alpha^i_n)$ из B. Если для всех компонент j таких, что $\beta_j = 0$ выполнено $\alpha^i_j = 0$, то не выполняется условие леммы 2, и, значит, значение функции $f_B(\beta)$ не может быть равным 0, поэтому процедура возвращает значение 0, что означает неуспех. Если существует такой индекс j, что $\beta_j = 0$ и $\alpha^i_j = *$, а для всех остальных индексов l таких, что $l \neq j$, $\beta_l = 0$ выполняется $\alpha^i_l = 0$, то устанавливаем $\alpha^i_j = *1$. Во всех остальных случаях значения компонент вектора α^i не меняется. Процедура ПроверкаЗначенияНоль возращает значение 1, что означает успех.

Вторая вспомогательная процедура называется ПроверкаЗначенияЕдиница. Она проверяет, можно ли установить значение искомой функции на опрашиваемом запросе в 1, и если можно, то изменяет множество B в соответствии с этим значением. Процедура состоит из следующих шагов.

1) Если все компоненты всех векторов из B не известны и $\beta = 0^j$, то устанавливаем $\alpha_j^1 = 0$, а $\alpha_j^i = 1$, $i = 2, \ldots, p$, возвращаем значение 1, что означает успех. Например, если $\beta = 0^1$, то после применения процедуры имеем

$$\begin{gathered}\alpha^1 = (0, *, \ldots, *),\\ \alpha^2 = (1, *, \ldots, *),\\ \ldots\\ \alpha^p = (0, *, \ldots, *),\end{gathered}$$

2) Если таких наборов α^i, что из $\beta_j = 0$ следует $\alpha_j^i = 0$, больше одного, то возвращаем значение 0, что означает неуспех.

3) Если для всех α^i из B выполняется $0_\beta \cap 1_{\alpha^i} \neq \emptyset$, то возвращаем значение 0.

4) В случае, когда имеется ровно один вектор α^i , для которого из $\beta_j = 0$ следует $\alpha_j^i 0$, тогда для остальных векторов из B запускаем процедуру ПроверкаЗначенияНоль. Тот факт, что процедура не завершилась на шаге 2, гарантирует, что все процедуры ПроверкаЗначенияНоль завершатся успешно. Возвращаем значение 1, что означает успех. Например, если опрашиваем набор $\beta = 0^1$ и имеем следующее множество нижних единиц

$$\begin{gathered}\alpha^1 = (0, \alpha_2^1, \ldots, \alpha_n^1),\\ \alpha^2 = (\alpha_2^2, \alpha_2^2, \ldots, \alpha_n^2),\\ \ldots\\ \alpha^p = (\alpha_2^2, \alpha_2^p, \ldots, \alpha_n^p),\end{gathered}$$

то после применения процедуры ПроверкаЗначенияЕдиница получим

$$\alpha^1 = (0, \alpha_2^1, \ldots, \alpha_n^1),$$
$$\alpha^2 = (*1, \alpha_2^2, \ldots, \alpha_n^2),$$
$$\ldots$$
$$\alpha^p = (*1, \alpha_2^p, \ldots, \alpha_n^p).$$

5) Если для одного вектора α^i из B выполнено условие, что из $\beta_j = 0$ следует $\alpha_j^i = *$, а для остальных векторов α^l, $l \neq i$, выполнено $0_\beta \cap 1_{\alpha^l} \neq \emptyset$, то для каждого индекса j такого, что $\beta_j = 0$ устанавливаем $\alpha_j^i = *0$, и возвращаем значение 1. Например, опрашиваем набор $\beta = 0^{0,1}$, и имеем следующее множество нижних единиц

$$\alpha^1 = (*, *, \alpha_3^1, \ldots, \alpha_n^1),$$
$$\alpha^2 = \{1, 0, \alpha_3^2, \ldots, \alpha_n^2),$$
$$\ldots$$
$$\alpha^p = \{\alpha_2^p, 1, \alpha_3^p, \ldots, \alpha_n^p),$$

т.е. у всех α^l, $l = 2, \ldots, p$, одна из координат α_1^l, α_2^l равна единице, то после применения процедуры ПроверкаЗначенияЕдиница получим

$$\alpha^1 = (*0, *0, \alpha_3^1, \ldots, \alpha_n^1),$$
$$\alpha^2 = \{1, 0, \alpha_3^2, \ldots, \alpha_n^2),$$
$$\ldots$$
$$\alpha^p = \{\alpha_2^p, 1, \alpha_3^p, \ldots, \alpha_n^p).$$

6) Если в множестве нижних единиц все кроме одного набора схожи, т.е. одинаково проставлены координаты , то тогда первому проставляем условный ноль, для остальных запускаем процедуру ПроверкаЗначенияНоль и возвращаем значение 1.

Например, если $\beta = 0^2$ и имеем следуещее множество B

$$\alpha^1 = (0, 1, \alpha_3^1, \ldots, \alpha_n^1),$$
$$\alpha^2 = (1, *, \alpha_3^2, \ldots, \alpha_n^2),$$
$$\ldots$$
$$\alpha^p = (1, *, \alpha_3^p, \ldots, \alpha_n^p),$$

то после применения процедуры ПроверкаЗначенияЕдиница получим

$$\alpha^1 = (0, 1, \alpha_3^1, \ldots, \alpha_n^1),$$
$$\alpha^2 = (1, *0, \alpha_3^2, \ldots, \alpha_n^2),$$
$$\alpha^3 = (1, *1, \alpha_3^3, \ldots, \alpha_n^3),$$
$$\ldots$$
$$\alpha^p = (1, *1, \alpha_3^p, \ldots, \alpha_n^p).$$

Теперь опишем алгоритм ответов на запросы. Считаем, что алгоритм запоминает все опрошенные наборы и ответы на них. На вход алгоритма поступаем опрашиваемый набор β. Алгоритм состоит из следующих шагов.

1) Если набор β находится на n слое, то возвращаем значение функции равное p.

2) Если набор β находится ниже $(n-1)$-го слоя, то выполняем следующие действия.

   а) Проверяем опрошен ли данный набор, если да, то возвращаем то значение, которое уже возвращали.

   б) Для каждого набора из запускаем процедуру ПроверкаЗначенияНоль.

   в) Если на каком-то наборе процедура ПроверкаЗначенияНоль вернула значение 0, то запускаем процедуру ПроверкаЗначенияЕдиница и возвращаем значение единица.

   г) Для всех опрошенных наборов булевого куба, мы запускаем процедуры ПроверкаЗначенияНоль и ПроверкаЗначенияЕдиница в зависимости от значения функции на данном наборе, которое мы ранее вернули алгоритму расшифровки. Если для каждого из наборов проверка прошла успешно, то возвращаем значение равное нулю, и все координаты вида $*1, *0$ преобразуем в $1, 0$. Иначе если хотя бы для одного из наборов условие не выполняется, все координаты вида $*1, *0$ преобразуем в $*$, и возвращаем значение функции, равное единице.

3) Если набор β находится на $(n-1)$-м слое, то выполняем следующие действия.

   а) Проверяем опрошен ли данный набор, если да, то возвращаем то значение, которое уже возвращали.

   б) Запускаем процедуру ПроверкаЗначенияЕдиница.

   в) Если проверка завершается неуспешно, то запускаем процедуру ПроверкаЗначенияНоль для каждого набора из B, и возвращаем значение ноль.

   г) Если проверка завершается неуспешно, то для всех опрошенных наборов булевого куба, мы запускаем процедуры ПроверкаЗначенияНоль и ПроверкаЗначенияЕдиница в зависимости от значения функции на данном наборе, которое мы ранее вернули алгоритму расшифровки. Если для каждого из наборов проверка прошла успешно, то возвращаем значение равное единица, и все координаты вида $*1, *0$ преобразуем в $1, 0$. Иначе если хотя бы для одного из наборов условие не выполняется, все координаты вида $*1, *0$ преобразуем в $*$, и возвращаем значение функции, равное ноль.

Перейдем к доказательству теоремы 2.

Согласно алгоритму ответов на запросы, приведенному выше, если алгоритм будет опрашивать наборы на нижних слоях булевого куба, то будем формировать значение функции равное нулю, т.е $f_B(\beta) = 0$, в этом случае алгоритму не удастся разгадать ни одну из координат $\alpha \in B$. То есть алгоритм расшифровки не получит информации о нижних единицах функции, так как в этом случае будем располагать нижние единицы в верхних слоях куба E_n. Итак, если $f_B(\beta) = 0$, то для любого $\alpha \in B$ справедливо $\sum\limits_{j \in 0_\beta} \alpha_j > 0$. Если нулей более чем один, чтобы однозначно восстановить $|0_\beta|$ нулевых литер для одной α_j, необходимо будет сделать дополнительные запросы на значение функции. В случае, когда невозможно вернуть ноль, это значит что координаты уже зафиксированы таким образом, что для некоторого α из B выполняется $\sum\limits_{j:\beta_j=0} \alpha_j = 0$.

Если опрашиваются наборы на $(n-1)$-м слое, то обычно $f_B(\beta) = 1$. В случае когда ни одна из литер $\alpha \in B$ не определена, то можем отнести α^1 к B_β, а остальные к B'_β, и зафиксировать одну координату для нижних единиц. Иначе необходимо будет задать дополнительные вопросы, чтобы зафиксировать координату нижней единицы.

Таким образом, как правило мы позволяем фиксировать одну координату за один запрос алгоритма расшифровки.

Если алгоритм будет достаточно умен, то будет опрашивать $(n-1)$-й и $(n-2)$-й слой, в этом случае располагаем нижние единицы на $(n-1)$-м слое. Если одна нижняя единица находится в подкубе $E^{n-1}_{0,j}$, то очевидно что остальные находятся в подкубе $E^{n-1}_{1,j}$, исходя из $f_B(0^j) = 1$. Далее алгоритм будет расшифровывать нижниее единицы в подкубе меньшей размерности $E^{n-1}_{1,j}$, где следующие нижние единицы так же будут принадлежать $(n-1)$-му слою. Таким образом, чтобы зафиксировать все литеры $\alpha^1 \in E^{n-1}_{0,j}$ необходимо будет задать n запросов, при этом для остальных $\alpha \in E^{n-1}_{1,j}$ мы зафиксируем одну литеру, так как они будут находится в разных подкубах, т.е уже необходимо будет задать $n-1$ запрос, для следующих $\alpha \in B$ действуем аналогично, откуда получаем

$$\varphi(n,p) \geq 1+n+n-1+n-2+n-3+n-(p-1)) = np-\frac{(p-1)p}{2}+1.$$

Список литературы

[1] В.Б.Кудрявцев, С.В. Алешин, А.С. Подколзин. Введение в теорию автоматов. Издательство «Наука», Москва, 1985.

[2] Алешин С.В. Полугруппы и группы автоматов // Интеллектуальные системы. — 2013. — Т. 17, вып. 1–4. — С. 129–141.

[3] Иванов И.Е. О некоторых свойствах автоматов с магазинной памятью // Интеллектуальные системы. — 2014. — Т. 18, вып. 1. — С. 243–252.

[4] Часовских А.А. Условия полноты линейно-p-автоматных функций // Интеллектуальные системы. — 2014. — Т. 18, вып. 3. — С. 203–252.

[5] Александров Д.Е. Об оценках автоматной сложности распознавания классов регулярных языков // Интеллектуальные системы. — 2014. — Т. 18, вып. 4. — С. 161–190.

[6] Гасанов Э.Э. Прогнозирование периодических сверхсобытий автоматами // Интеллектуальные системы. — 2015. — Т. 19, вып. 1. — С. 23–34.

[7] Иванов И.Е. О сохранении периодических последовательностей автоматами с магазинной памятью с однобуквенным магазином // Интеллектуальные системы. — 2015. — Т. 19, вып. 1. — С. 145–160.

[8] Летуновский А.А. Выразимость линейных автоматов относительно расширенной суперпозиции // Интеллектуальные системы. — 2015. — Т. 19, вып. 1. — С. 161–170.

[9] Гербус В.Г. О связи функций автомата и автоматной функции // Интеллектуальные системы. — 2015. — Т. 19, вып. 2. — С. 109–116.

[10] Миронов А.М. Критерий реализуемости функций на строках вероятностными автоматами Мура с числовым выходом // Интеллектуальные системы. — 2015. — Т. 19, вып. 2. — С. 149–160.

[11] Терехина И.Ю. Модель невлияния для квантовых автоматов // Интеллектуальные системы. — 2015. — Т. 19, вып. 2. — С. 183–190.

[12] Бабин Д.Н., Летуновский А.А. О возможностях суперпозиции, при наличии в базисе автоматов фиксированной добавки из булевых функций и задержки // Интеллектуальные системы. — 2015. — Т. 19, вып. 3. — С. 71–78.

[13] Бабин Д.Н. Автоматы с суперпозициями, пример нерасширяемости до предполного класса // Интеллектуальные системы. — 2015. — Т. 19, вып. 3. — С. 87–94.

[14] Э.Э.Гасанов, А.А.Мастихина Прогнозирование общерегулярных сверхсобытий автоматами // Интеллектуальные системы. — 2015. — Т. 19, вып. 3. — С. 127–154.

[15] Иванов И.Е. Нижняя оценка на максимальную длину периода выходной последовательности автономного автомата с магазинной памятью // Интеллектуальные системы. — 2015. — Т. 19, вып. 3. — С. 175–194.

[16] А.А.Часовских. Критериальные системы в классах линейно-автоматных функций над конечными полями // Интеллектуальные системы. — 2015. — Т. 19, вып. 3. — С. 195–207.

[17] Пантелеев П.А. Об отличимости состояний решетчатых автоматов // Интеллектуальные системы. — 2004. — Т. 8, вып. 1–4. — С. 529–542.

[18] Кирнасов А.Е. Об отношении сложностей условного и безусловного установочного экспериментов // Интеллектуальные системы. — 2005. — Т. 9, вып. 1–4. — С. 433–444.

[19] Уваров Д.В. О сложности кратных диагностических экспериментов для подмножеств состояний автоматов // Интеллектуальные системы. — 2005. — Т. 9, вып. 1–4. — С. 485–504.

[20] Кудрявцев В.Б., Грунский И.С., Козловский В.А. Анализ и синтез автоматов по их поведению // Интеллектуальные системы. — 2006. — Т. 10, вып. 1–4. — С. 345–448.

[21] Пантелеев П.А. Об отличимости состояний автомата при искажениях на входе // Интеллектуальные системы. — 2007. — Т. 11, вып. 1–4. — С. 653–678.

[22] Angluin D. Queries and Concept Learning // Machine Learning. Vol. 2. 1988. P. 319–342.

[23] В.Б.Кудрявцев, Э.Э.Гасанов, А.С.Подколзин. Введение в теорию интеллектуальных систем. — М.: Издательский отдел факультета ВМиК МГУ, 2006.

[24] Коробков В. К. О монотонных функциях алгебры логики — М.: Проблемы кибернетики. Вып. 13. М.: Наука, 1965.

[25] G. Hansel. Sur le nombre des fonctions booleennes monotones de n variables, C.R. Acad. Sci. Paris 262 (1966), no. 20, 1088–1090 (French).

[26] Damaschke, P. (2003). On Parallel Attribute-Efficient, Learning // Journal of Computer and System Sciences, Volume 67, Issue 1, August 2003, 46-62.

[27] В. В. Осокин. О расшифровке монотонных булевых функций с несущественными переменными // Дискретная математика, 22:3 (2010), 134–145.

[28] V.V.Osokin. On learning monotone Boolean functions with irrelevant variables // Discrete Mathematics and Applications. Volume 20, 2010. Issue 3, 307-320.

[29] Adam R. Klivans, Rocco A. Servedio. Toward Attribute Efficient Learning of Decision Lists and Parities // The Journal of Machine Learning Research Volume 7, 2006.

[30] Vitaly Feldman. On Attribute Efficient and Non-adaptive Learning of Parities and DNF Expressions // The Journal of Machine Learning Research Volume 8, 2007

[31] Осокин В.В. Асимптотически оптимальный алгоритм расшифровки разбиения булевого куба на подкубы // Интеллектуальные системы. — 2007. — Т. 11, вып. 1–4. — С. 635–652.

[32] В. В. Осокин. О сложности расшифровки разбиения булевого куба на подкубы // Дискретная математика, 20:2 (2008), 46–62.

[33] Воронин Б.В., Осокин В.В. О сложности расшифровки существенных переменных функции, задающей разбиение булевого куба // Интеллектуальные системы. — 2008. — Т. 12, вып. 1–4. — С. 159–178.

[34] Осокин В.В. О параллельной расшифровке разбиений булевого куба // Интеллектуальные системы. — 2009. — Т. 13, вып. 1–4. — С. 427–454.

[35] Осокин В.В. О параллельной параметро-эффективной расшифровке псевдо-булевских функций // Интеллектуальные системы. — 2010. — Т. 14, вып. 1–4. — С. 429–458.

[36] Гасанов Э.Э. Расшифровка линейных функций ранжирования // Материалы XI Международного семинара "Дискретная математика и ее приложения посвященного 80-летию со дня рождения академика О.Б.Лупанова (Москва, 18-23 июня 2012 г.). Изд-во мех-мат фак-та МГУ. 2012. С. 332-334.

[37] Хегай С.И. Расшифровка полиномиальных функций ранжирования // Интеллектуальные системы. 2015. 19:1. 213–230.

[38] Быстрыгова А.В. Сложность расшифровки линейных булевых функций // Интеллектуальные системы. 2015. 19:3. 101–126.

[39] Nakamura A., Abe N. (1995) Exact learning of linear combinations of monotone terms from function value queries // Theoretical Computer Science, Volume 137, Issue 1, 159-176, 1995.

[40] Гасанов Э.Э., Ниязова З.А. Расшифровка арифметических сумм малого числа монотонных конъюнкций // Материалы XI Международного семинара Дискретная математика и ее приложения (Москва, 18-23 июня 2012 г.). — Изд-во механико-математического ф-та МГУ Москва, 2012. — С. 335–337.

Learning of arithmetic sum of monotone conjuctions

Z. A. Niyazova

The problem of learning of the arithmetic sums of monotone conjunctions is considered. It is offered algorithm of learning on the basis of which the upper bound of learning complexity is obtained. The algorithm of replies to the requests is proposed. With the help of this algorithm, lower bound of learning complexity is obtained.

Keywords: learning of discrete functions, sum of monotone conjuctions.