Math-Net.Ru

Общероссийский математический портал

В. В. Осокин, О расшифровке монотонных булевых функций с несущественными переменными, *Дискрет. матем.*, 2010, том 22, выпуск 3, 134–145

DOI: <https://doi.org/10.4213/dm1113>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 35.173.219.12

26 апреля 2023 г., 05:46:50

О расшифровке монотонных булевых функций с несущественными переменными

© 2010 г. В. В. Осокин

Задача расшифровки функции состоит в ее восстановлении с помощью условного эксперимента, состоящего из запросов на значение функции в заданной точке. Нами получен порядок сложности расшифровки булевых монотонных функций, имеющих также фиктивные переменные.

1. Введение

Неформально задача расшифровки функций из некоторого класса Φ состоит в построении ученика (алгоритма), который смог бы успешно играть с учителем в следующую игру. Учитель загадывает некоторую функцию из класса Φ . Ученик должен расшифровать загаданную функцию, задав учителю минимальное число вопросов. Под вопросом понимается точка из области определения функции. Ответом учителя является значение функции в данной точке.

Пусть $\varphi(F, f)$ — число запросов на значение функции, требуемое алгоритму (ученику) F , чтобы разгадать функцию $f \in \Phi$, $L(\Phi)$ — множество всех алгоритмов расшифровки функций из Φ . Тогда требуется построить алгоритм F , для которого величина $\max_{f \in \Phi} \varphi(F, f)$ минимальна среди всех алгоритмов из $L(\Phi)$. Положив

$$\varphi(n) = \min_{F \in L(\Phi)} \max_{f \in \Phi} \varphi(F, f),$$

получаем, что наилучший алгоритм должен делать не более $\varphi(n)$ запросов на значение функции при расшифровке самой худшей для себя функции из Φ . Величина $\varphi(n)$ называется сложностью расшифровки функций из класса Φ .

В 80-е годы были предложены общие модели расшифровки, в рамках которых и сосредоточились дальнейшие исследования. Так, в 1984 г. Л. Валиант предложил модель вероятно примерно точной расшифровки [3]. В 1988 г. Д. Англуин предложила модель точной расшифровки [4]. Именно эти две модели считаются за рубежом основополагающими. В отечественной литературе среди направлений, близких к рассматриваемому, можно выделить теорию тестового распознавания [5, 6].

Первые результаты в рамках точной модели расшифровки были получены задолго до 1988 г. Так в начале 60-х годов В. К. Коробков получил порядок сложности расшифровки монотонных функций [1]. В 1968 г. Ж. Ансель предложил алгоритм [2], который позволил ему получить точную оценку сложности расшифровки монотонных функций от n

переменных, т.е., показать, что

$$\varphi(n) = \min_{F \in L(M^n)} \max_{f \in M^n} \varphi(F, f) = \binom{n}{\lfloor n/2 \rfloor} + \binom{n}{\lfloor n/2 \rfloor + 1},$$

где M^n — класс монотонных булевых функций от n переменных.

Введем в рассмотрение величину

$$\varphi(k, n) = \min_{F \in L(M^n)} \max_{f \in M^{k,n}} \varphi(F, f),$$

где $M^{k,n}$ — множество монотонных функций от n переменных, существенно зависящих от не более чем k своих переменных. Отметим, что минимум берется по классу алгоритмов, расшифровывающих все монотонные функции от n переменных. Тем самым предполагается, что число k неизвестно алгоритму расшифровки.

В серии работ П. Дамашке [7, 8, 9] показано, что $\varphi(k, n)$ по порядку не превосходит $2^k + k \log n$. Там же (ошибочно) показано, что $\varphi(k, n)$ по порядку больше 2^k . На самом деле, как показано в настоящей работе, $\varphi(k, n)$ по порядку равно $2^k / \sqrt{k} + k \log n$.

Автор выражает благодарность профессору Э. Э. Гасанову за постановку задачи и помощь в работе и академику В. Б. Кудрявцеву за многочисленные критические замечания, способствовавшие существенному улучшению текста.

2. Постановка задачи и формулировка результатов

Обозначим $E = \{0, 1\}$. Для $\alpha = (\alpha_1, \dots, \alpha_n), \beta = (\beta_1, \dots, \beta_n) \in E^n$ будем писать $\alpha \leq \beta$, если $\alpha_i \leq \beta_i$ для всех $i = 1, \dots, n$. Пишем $\alpha < \beta$, если $\alpha \leq \beta$ и существует такой номер $i \in \{1, \dots, n\}$, что $\alpha_i < \beta_i$.

Напомним, что булева функция $f(y_1, \dots, y_n)$ называется монотонной, если соотношение $f(\alpha_1, \dots, \alpha_n) \leq f(\beta_1, \dots, \beta_n)$ имеет место для любых наборов $(\alpha_1, \dots, \alpha_n) \leq (\beta_1, \dots, \beta_n)$.

Если $f(\alpha_1, \dots, \alpha_n) = 1$, то набор $(\alpha_1, \dots, \alpha_n)$ называется единицей монотонной функции f . Единица монотонной функции $f(\alpha_1, \dots, \alpha_n)$ называется нижней единицей функции f , если для любого $(\beta_1, \dots, \beta_n)$, такого, что $(\beta_1, \dots, \beta_n) < (\alpha_1, \dots, \alpha_n)$, выполняется равенство $f(\beta_1, \dots, \beta_n) = 0$.

Весом набора α называют число его координат, равных единице. Наборы $\alpha \in E^n$ называют вершинами куба E^n . Множество всех вершин куба, имеющих вес k , называют k -м слоем куба E^n . Гранью куба E^n называется множество

$$E_{\sigma_1, \dots, \sigma_s}^{n, i_1, \dots, i_s} = \{(\alpha_1, \dots, \alpha_n) \in E^n : \alpha_{i_1} = \sigma_1, \dots, \alpha_{i_s} = \sigma_s\}.$$

Через 0_G и 1_G будем обозначать наименьший и наибольший наборы из E^n , принадлежащие грани G . Формально, $\delta_G = (\alpha_1^\delta, \dots, \alpha_n^\delta) \in E^n$, $\delta \in \{0, 1\}$, $\alpha_{i_1}^\delta = \sigma_1, \dots, \alpha_{i_s}^\delta = \sigma_s$ и $\alpha_j^\delta = \delta$ при $j \notin \{i_1, \dots, i_s\}$. Булев куб E^n мы будем схематически изображать в виде фигуры, приведенной на рис. 1 (слева). Заметим, что любая пара наборов (α, β) , $\alpha < \beta \in E^n$, задает грань G в E^n , такую что $\alpha = 0_G$, $\beta = 1_G$. Схематически этот факт будем изображать как на рис. 1 (справа). Мы будем говорить, что грань G образована наборами α и β .

Переменная y_i ($1 \leq i \leq n$) функции $f(y_1, \dots, y_n)$ называется существенной, если существуют такие наборы α и β , отличающиеся только i -й компонентой, что $f(\alpha) \neq f(\beta)$.

Рис. 1. Булев куб и грань булевого куба

В противном случае переменная y_i называется фиктивной (несущественной) переменной функции f . Функции f и g называются равными, если множества их существенных переменных совпадают и на любых двух наборах, различающихся, быть может, только значениями несущественных переменных, значения функций одинаковы.

Пусть $Y = \{y_1, \dots, y_n\}$, $X = \{x_1, \dots, x_m\} \subseteq Y$, $Z = \{z_1, \dots, z_{n-m}\} \subseteq Y$ и $Y = X \cup Z$. Подфункцией булевой функции $f(y_1, \dots, y_n)$, полученной из f путем фиксации переменных $z_1, \dots, z_{n-m}$ значениями $\sigma_1, \dots, \sigma_{n-m}$, $\sigma_i \in \{0, 1\}$, $i \in \{1, \dots, n-m\}$, назовем функцию $f'(x_1, \dots, x_m)$, полученную подстановкой для каждого $i \in \{1, \dots, n-m\}$ константы σ_i в функцию f на место переменной z_i .

Пусть M^n — множество монотонных булевых функций от n переменных, $L(M^n)$ — множество алгоритмов, решающих задачу расшифровки функций из M^n с помощью запросов на значение функции. Под запросом на значение функции понимается набор значений переменных функции, а под ответом на запрос — значение расшифровываемой функции на этом наборе. Каждый алгоритм из $L(M^n)$ должен для любой функции f из M^n , сделав некоторое число запросов на значение функции, полностью восстановить таблицу значений функции f . Алгоритмы расшифровки предполагаются условными, то есть при формировании очередного запроса на значения функции алгоритм расшифровки может пользоваться ответами на все предыдущие запросы.

Через $\varphi(F, f)$ обозначим число запросов на значение функции, требуемое алгоритму $F \in L(M^n)$, чтобы полностью восстановить таблицу значений функции $f \in M^n$.

Рассмотрим класс $M^{k,n}$ всех монотонных функций от n переменных, существенно зависящих от не более, чем k своих переменных. Функции из этого класса будем называть k -существенными монотонными функциями.

В работе исследуется сложность расшифровки функций из $M^{k,n}$ в худшем случае, то есть величина

$$\varphi(k, n) = \min_{F \in L(M^n)} \max_{f \in M^{k,n}} \varphi(F, f).$$

Получен порядок величины $\varphi(k, n)$.

Теорема 1. При $k, n \rightarrow \infty$ справедливо соотношение

$$\varphi(k, n) \asymp \frac{2^k}{\sqrt{k}} + k \log n.$$

3. Нижняя оценка

Лемма 1. Сложность расшифровки булевых функций из произвольного класса мощности N с помощью запросов на значение функции не меньше $\log N$.

Доказательство. Ответом на любой запрос на значение функции может быть только 0 или 1. Отсюда, если для расшифровки любой функции из заданного класса алгоритм ограничивается m запросами на значение функции, то различные результаты работы этого алгоритма вместе могут задавать не более чем 2^m различных функций. Значит, $2^m \geq N$ и, следовательно, $m \geq \log N$. Лемма доказана.

Лемма 2. Для любых $k \leq n$ при $k, n \rightarrow \infty$ справедливо соотношение

$$\varphi(k, n) \gtrsim k \log \frac{n}{k} + \sqrt{\frac{2}{\pi}} \frac{2^k}{\sqrt{k}}.$$

Доказательство. Согласно нижней оценке, полученной Э. Н. Гильбертом [10], $|M^k| \geq 2^{\binom{k}{\lfloor k/2 \rfloor}}$. Здесь $\binom{k}{\lfloor k/2 \rfloor}$ — число точек на среднем слое k -мерного булевого куба, где под средним слоем k -мерного булевого куба мы понимаем слой с номером $\lfloor k/2 \rfloor$. Оценим число $|M^k| - |M^{k-1}|$ монотонных функций, существенно зависящих от всех k своих переменных.

Пусть a, b, c — три произвольных набора из среднего слоя k -мерного булевого куба, таких что поэлементная дизъюнкция этих трех наборов дает в точности единичный набор: $a \vee b \vee c = (1, 1, \dots, 1)$. Рассмотрим множество M' монотонных функций от k переменных, такое что для каждой функции из M' наборы a, b, c являются нижними единицами этой функции и все нижние единицы этой функции располагаются в среднем слое. Легко заметить, что функции из множества M' существенно зависят от всех k своих переменных. Действительно, для того чтобы монотонная функция существенно зависела от некоторой своей переменной, достаточно, чтобы эта переменная входила хотя бы в одну конъюнкцию минимальной ДНФ этой функции. Но по определению нижних единиц a, b, c , соответствующие им три конъюнкции уже содержат все k переменных. Понятно, что мощность множества M' равна $2^{\binom{k}{\lfloor k/2 \rfloor} - 3}$. Поскольку каждая функция из M' существенно зависит от всех k своих переменных, получаем, что

$$|M^k| - |M^{k-1}| > |M'| > 2^{\binom{k}{\lfloor k/2 \rfloor} - 3}.$$

Значит, для числа $|M^{n,k}|$ монотонных функций от n переменных, существенно зависящих не более чем от k переменных, справедливо соотношение

$$|M^{n,k}| > 2^{\binom{k}{\lfloor k/2 \rfloor} - 3} \binom{n}{k}.$$

Логарифмируя, получаем мощностную нижнюю оценку для $\varphi(k, n)$:

$$\varphi(k, n) > \log \binom{n}{k} + \left(\binom{k}{\lfloor k/2 \rfloor} - 3 \right).$$

Воспользовавшись известным неравенством для биномиальных коэффициентов

$$\left(\frac{n}{k} \right)^k \leq \binom{n}{k} \leq \left(\frac{ne}{k} \right)^k$$

и асимптотическим равенством для центральных биномиальных коэффициентов

$$\binom{k}{k/2} \sim \sqrt{\frac{2}{\pi}} \frac{2^k}{\sqrt{k}}, \quad k \rightarrow \infty,$$

получаем утверждение леммы.

Второе слагаемое в полученной в лемме 2 оценке $\varphi(k, n)$ лишь в два раза меньше асимптотики сложности расшифровки монотонных функций от k переменных, существенно зависящих от не более чем k переменных (то есть функций из M^k). Докажем, что число запросов на значение функции, указанное в первом слагаемом, необходимо любому алгоритму расшифровки функций из $M^{k,n}$ только для того, чтобы найти все существенные переменные некоторой функции из $M^{k,n}$ (даже если k наперед известно алгоритму расшифровки).

Если из значений на запрошенных некоторым алгоритмом наборах следует существенность некоторой переменной неизвестной функции, будем говорить, что алгоритм расшифровал эту существенную переменную. При этом сложностью определения существенных переменных функции данным алгоритмом назовем минимальное число запросов, необходимое этому алгоритму для расшифровки всех переменных этой функции.

Предложение 1. *Для произвольного алгоритма расшифровки существует такая функция из $M^{1,n}$, что для определения существенной переменной этому алгоритму требуется асимптотически не менее $\log n$ запросов на значение функции.*

Доказательство. Рассмотрим случай 1-существенной монотонной функции $f = x_1$. Она зависит от n переменных $y_1, \dots, y_n$, одна из которых является существенной (мы ее обозначили через x_1). Покажем, что любому алгоритму потребуется по крайней мере $\log n$ запросов на значение функции для того, чтобы определить существенную переменную, т.е. чтобы понять, какая переменная из $y_1, \dots, y_n$ скрывается за обозначением x_1 . Пусть J_{temp} — множество номеров переменных, которые на данном шаге алгоритма все еще могут быть существенными. Пока алгоритм не запросил ни одного набора, любая переменная может быть существенной, т.е. $J_{\text{temp}} = \{1, \dots, n\}$. Пусть алгоритм определения существенных переменных на m -м шаге запросил значение функции на наборе a . Если среди элементов набора a с номерами из J_{temp} нулей больше, чем единиц, положим $f(a) = 0$ и удалим из J_{temp} номера всех переменных, которые равны единице в наборе a . Иначе положим $f(a) = 1$ и удалим из J_{temp} номера всех переменных, которые равны нулю в наборе a .

Легко видеть, что на каждом шаге мощность множества J_{temp} уменьшается не более чем в два раза. Значит, алгоритму определения существенных переменных потребуется как минимум $\lceil \log n \rceil$ запросов на значение функции, чтобы мощность J_{temp} стала равна единице, то есть чтобы определить существенную переменную x_1 . Предложение доказано.

Стоит заметить, что k -мерный случай не следует из одномерного непосредственно, как это могло бы показаться. Действительно, отвечая на запросы алгоритма определения существенных переменных, мы, вообще говоря, раскрываем информацию сразу о многих существенных переменных.

Лемма 3. *Для любых $k \leq n$ и для любого алгоритма расшифровки существует такая функция из $M^{k,n}$, что число запросов на значение функции, необходимое этому алгоритму для определения всех ее существенных переменных асимптотически не меньше $(k/2) \log n$.*

Рис. 2. Функция $f(y_1, \dots, y_n)$

Доказательство. Применим следующий подход: множество существенных переменных разбивается на пары таким образом, чтобы ответ на любой запрос алгоритма раскрывал информацию только о некоторой паре переменных. Если на определение каждой пары требуется $\log n$ наборов, то мы получаем требуемую оценку. Чтобы обеспечить это, фиксируется некоторая k -существенная функция $f(y_1, \dots, y_n)$ с существенными переменными $y_{i_1} = x_1, \dots, y_{i_k} = x_k$, подфункция f' которой, полученная из f произвольной фиксацией ее несущественных переменных, обладает следующим свойством. На наборах k -мерного куба, первые две компоненты которых равны, функция f' вырождается в функцию x_1 ; на наборах, первые две компоненты которых не равны, а третья и четвертая — равны, функция f' вырождается в функцию x_3 и так далее. В связи со сказанным, мы можем последовательно “прятать” сначала пару переменных x_1, x_2 , затем, когда эта пара раскрыта, пару переменных x_3, x_4 , затем другие пары, пока алгоритм не определит все существенные переменные. При этом, разгадывая очередную пару x_{i_1}, x_{i_2} переменных алгоритм пытается определить существенную переменную 1-существенной функции x_{i_1} , и потому, согласно оценке в одномерном случае, тратит на эту переменную $\log n$ запросов на значение функции.

Построим такую функцию $f \in M^{k,n}$, что число запросов на значение функции, требуемое F для определения k ее существенных переменных, асимптотически не меньше $(k/2) \log n$. В качестве функции f возьмем функцию, задаваемую ДНФ из леммы 1 работы [7]. Без ограничения общности будем полагать, что k четное. Пусть F — алгоритм расшифровки функций из $M^{k,n}$. Пусть $y_{i_1}, y_{i_2}, \dots, y_{i_k}$ — существенные переменные функции f (номера $i_1, \dots, i_k$ пока не фиксированы). Положим

$$f(y_1, \dots, y_n) = y_{i_1} \vee y_{i_2} (y_{i_3} \vee y_{i_4} (y_{i_5} \vee y_{i_6} (\dots (y_{i_{k-1}} \vee y_{i_k}))))).$$

Обозначим $x_1 = y_{i_1}, \dots, x_k = y_{i_k}$. На рис. 2 изображен n -мерный булев куб. Каждому направлению n -мерного куба соответствует некоторая переменная из множества $\{y_1, \dots, y_n\}$.

На рис. 2 переменные приписаны лишь направлениям, необходимым для иллюстрации функции f . На заштрихованных гранях функция f принимает значение 1. Можно считать, что алгоритм расшифровки знает указанную ДНФ. Единственное, что ему неизвестно — это какие переменные функции f скрываются за обозначениями $x_1, \dots, x_k$.

Будем на каждом шаге алгоритма анализировать вновь запрошенный им набор, и в зависимости от набора и от значений на предыдущих наборах выбирать значение функции f на поданном наборе.

Через J_{temp} обозначим множество натуральных чисел, которое мы будем использовать для хранения номеров переменных, которые все еще могут быть существенными, но при этом могут оказаться и несущественными. Изначально положим $J_{\text{temp}} = \{1, \dots, n\}$, то есть множество J_{temp} содержит номера всех переменных функции f . Через $b \in \{0, 1, \dots, n\}^k$ обозначим набор $b = (b_1, \dots, b_k)$ длины k . Этот набор будет хранить номера известных существенных переменных функции f . Изначально положим его нулевым (все существенные переменные неизвестны). Через s обозначим номер первой нулевой компоненты набора b . Изначально $s = 1$. Номера существенных переменных будем открывать алгоритму последовательно по парам. В связи с этим s всегда нечетное.

Пусть на некотором шаге алгоритм F запросил значение f на наборе a . Пусть i — наименьшее нечетное натуральное число, такое что $a_{b_i} = a_{b_{i+1}}$ и $b_i \neq 0$. Если такое i существует, то есть если существует такая пара уже найденных алгоритмом переменных y_{b_i} и $y_{b_{i+1}}$, что они фиксированы в наборе a одинаковым образом, то полагаем $f(a) = a_{b_i}$ и переходим к следующему поданному алгоритмом набору. Заметим, что при этом мы не сообщаем алгоритму никакой новой информации о существенных переменных, поскольку значение функции f на наборе a может быть использовано только для определения (уже “раскрытых” нами) переменных y_{b_i} и $y_{b_{i+1}}$.

Пусть теперь приведенное выше равенство не выполняется ни для каких i . Пусть $J_0 \subseteq J_{\text{temp}}$ — номера компонент набора a , равных 0; $J_1 \subseteq J_{\text{temp}}$ — номера компонент набора a , равных 1. Если $|J_0| > |J_1|$, полагаем $f(a) = 0$ и $J_{\text{temp}} = J_{\text{temp}} \setminus J_1$. Иначе полагаем $f(a) = 1$ и $J_{\text{temp}} = J_{\text{temp}} \setminus J_0$. Если $|J_{\text{temp}}| \neq 2$, переходим к следующему набору.

Пусть $J_{\text{temp}} = \{j_1, j_2\}$. Полагаем $b_s = j_1$, $b_{s+1} = j_2$ и $s = s + 2$. Если $s < k$, полагаем $J_{\text{temp}} = \{1, 2, \dots, n\} \setminus \{b_1, \dots, b_{s-1}\}$, и продолжаем строить значения функции f на подаваемых алгоритмом F наборах в соответствии с приведенным выше описанием. Если же $s > k$, достраиваем функцию f на оставшихся наборах произвольным образом (так, чтобы она оставалась k -существенной), и на всех новых наборах, запрашиваемых алгоритмом F , выдаем значения f на этих наборах.

Заметим, что значение s меняется $k/2$ раз. Таким образом, если показать, что между двумя последовательными увеличениями значения s алгоритм F делает $\lceil \log n \rceil$ запросов на значения функции, то утверждение будет доказано. Для этого достаточно заметить, что для определения i -й или $(i + 1)$ -й существенной переменной (i — нечетное) алгоритму F нужно запрашивать наборы a , такие что для каждой пары b_j, b_{j+1} , j нечетное, $j < i$ выполнено $a_{b_j} \neq a_{b_{j+1}}$. Но приведенное построение гарантирует, что впервые такие наборы алгоритм F выдаст как раз при $s = i$. Этим доказательство завершается.

4. Верхняя оценка

Для того, чтобы завершить доказательство теоремы 1, остается показать, что $\varphi(k, n)$ по порядку не превосходит $2^k / \sqrt{k} + k \log n$.

Как будет показано в лемме 5, какое-то число $k' \leq k$ существенных переменных неизвестной функции можно найти примерно за $\log n$ запросов на каждую переменную. После этого найденные переменные можно фиксировать таким образом, чтобы получившаяся подфункция исходной функции существенно зависела хотя бы от одной переменной, и снова применить алгоритм из леммы 5. Так можно продолжать, пока не будут найдены

Рис. 3. Грань G с некоторой f -разбивающей G парой вершин (a, b)

все существенные переменные. После этого остается применить алгоритм Анселя для окончательной расшифровки.

Для отыскания упомянутой выше подходящей фиксации найденных переменных может использоваться полный перебор. Для задачи расшифровки разбиения булевого куба на подкубы [11] можно показать, что при поиске подходящей подфункции нельзя избежать полного перебора. Как показано в лемме 6, при расшифровке k -существенных монотонных функций от полного перебора при выборе очередной фиксации найденных переменных можно отказаться, используя алгоритм Анселя расшифровки n -существенных монотонных функций от n переменных при выборе очередной фиксации. Для этого при помощи алгоритма Анселя последовательно расшифровываются две функции: подфункция исходной функции, полученная фиксацией всех переменных, кроме уже найденных, единицами, и подфункция исходной функции, полученная фиксацией всех переменных, кроме уже найденных, нулями. Если эти две подфункции равны, то найдены все существенные переменные исходной функции. В противном случае любой набор, на котором эти две подфункции принимают разные значения, дает искомую подходящую фиксацию найденных существенных переменных.

Нам потребуется следующее определение. Пусть G — грань n -мерного булевого куба, 0_G — набор этой грани с максимальным числом нулей, 1_G — набор этой грани с максимальным числом единиц. Пусть a — произвольная вершина этой грани, не совпадающая с 0_G и 1_G , b получается из a взятием поэлементного отрицания набора a , $b \neq 0_G$, $b \neq 1_G$. Пусть f — некоторая монотонная функция, определенная на E^n . Пару (a, b) будем называть f -разбивающей грань G , если $f(a) = f(b)$, $f(0_G) = 0$, $f(1_G) = 1$. Далее предполагаем, что $f(a) = f(b) = 0$. В противном случае в последующих рассуждениях достаточно заменить 1_G на 0_G . Пусть G_a — грань, образованная точками 1_G и a , G_b — грань, образованная точками 1_G и b . На рис. 3 изображена грань G с некоторой f -разбивающей G парой вершин (a, b) . Цифры, приписанные вершинам суть значения функции f в этих вершинах.

Лемма 4. Если $f \in M^n$ и пара вершин (a, b) f -разбивает грань G куба E^n , то подфункция f_1 исходной функции f , определенная на G_a , существенно зависит хотя бы от одной переменной, подфункция f_2 исходной функции f , определенная на G_b , существенно

зависит хотя бы от одной переменной, и множества переменных, от которых зависят функции f_1 и f_2 , не пересекаются.

Доказательство. Утверждение сразу следует из того, что переменные, фиксированные единицами в наборах a и b , суть различные переменные.

Следующая лемма основана на теореме 4.1 работы [9] и ее доказательстве.

Лемма 5. Пусть функция $f \in M^{k,n}$ существенно зависит хотя бы от одной своей переменной. Тогда существует алгоритм, который определяет некоторое число k' существенных переменных функции f , $1 \leq k' \leq k$, за не более чем $8k' \log n$ запросов на значение функции.

Доказательство. Для доказательства леммы мы построим бинарное дерево с k' концевыми вершинами, в каждой из которых определяется некоторая (уникальная) существенная переменная. Каждой вершине этого дерева сопоставлена некоторая грань исходного n -мерного булевого куба. Корню сопоставлена грань, совпадающая с самим n -мерным кубом. Построение дерева осуществляется следующим образом. Если в грани, соответствующей некоторой уже построенной вершине дерева, мы находим грань единичной размерности, в двух вершинах которой функция f принимает различные значения, то объявляем данную вершину дерева концевой (мы нашли существенную переменную). В противном случае в этой грани мы находим грань G (возможно) меньшей размерности и пару точек (a, b) , которая f -разбивает грань G , добавляем к дереву две новые вершины, сопоставив одной вершине грань G_a , а другой — G_b и соединяем эти две вершины ребрами с текущей вершиной.

Без ограничения общности будем предполагать, что n является степенью двойки. В противном случае добавим такое число фиктивных переменных, чтобы число переменных, от которых зависит получившаяся в результате функция f' , было бы степенью двойки. Каждой переменной функции f поставим в соответствие бинарную кодировку ее номера, то есть слово $b_1 \dots b_r$, где $r = \log_2 n$. Если рассматривать эти слова как столбцы некоторой матрицы, то в полученной матрице A размера $r \times n$ нет одинаковых столбцов. Запросим значения функции f на строках матрицы A и на их отрицаниях (всего $2 \log n$ запросов). Возможны два варианта.

- (1) Среди пар противоположных наборов есть пара (a_1, a_2) , такая что $f(a_1) = f(a_2)$. Тогда пара (a_1, a_2) f -разбивает n -мерный булев куб.
- (2) Для всех пар (a_1, a_2) противоположных наборов $f(a_1) \neq f(a_2)$. Заменим все строки матрицы A , значения функции f на которых равны нулю, на противоположные. Тогда для любой строки a матрицы A выполнено $f(a) = 1$. Определим матрицу B , положив ее i -ю строку равной поэлементной конъюнкции первых i строк матрицы A : $B_i = A_1 \wedge \dots \wedge A_i$ (см. рис. 4). Запросим значения функции f на строках матрицы B . Заметим, что набор B_r имеет только одну единичную компоненту, и если $f(B_r) = 1$, то переменная, соответствующая этой компоненте, является существенной. Далее предполагаем, что существует номер i , такой что $f(B_i) = 1$, а $f(B_{i+1}) = 0$.

Обозначим грань куба, образованную вершинами B_i и $(00 \dots 00)$, через G . Рассмотрим набор $B_i \wedge \overline{B_{i+1}}$ (то есть набор, противоположный набору B_i в грани G ; конъюнкция здесь снова поэлементная), см. рис. 4. Получаем, что

$$\begin{cases} B_{i+1} \leq B_i \\ B_{i+1} \leq A_{i+1} \end{cases} \Rightarrow B_i \wedge \overline{B_{i+1}} = B_i \wedge \overline{A_{i+1}} \leq \overline{A_{i+1}}.$$

Рис. 4. Наборы $B_i \wedge \overline{B_{i+1}}$ и B_{i+1} f -разбивают грань G

Поскольку $f(\overline{A_{i+1}}) = 0$, то $f(B_i \wedge \overline{B_{i+1}}) = 0$. Таким образом, пара вершин $(B_{i+1}, B_i \wedge \overline{B_{i+1}})$ f -разбивает грань G .

В результате мы либо определяем существенную переменную функции f , либо находим две грани G_a и G_b , полученные f -разбиением некоторой грани G , такие что подфункции функции f , определенные на этих гранях, являются существенными функциями и зависят от разных переменных. Для каждой из подфункций мы можем повторить описанные действия. В итоге мы определим некоторое число $k' \leq k$ существенных переменных. При этом будет построено бинарное дерево с ровно k' концевыми вершинами, в каждой вершине которого делается не более чем $4 \log n$ запросов на значение функции f . Таким образом, общее число запросов не превышает $8k' \log n$. Лемма доказана.

Заметим, что запросы на значение функции, которые делает описанный алгоритм в некоторой вершине упомянутого бинарного дерева, не зависят от ответов на запросы в той же вершине. В этом заключается преимущество описанного алгоритма над алгоритмом, который определяет существенную переменную неизвестной функции при помощи бинарного поиска.

Пусть $f \in M^{k,n}$, Y_f — множество всех ее переменных, $|Y_f| = n$, X_f — множество всех ее существенных переменных, $X'_f \subseteq X_f$, $|X'_f| = k'$.

Лемма 6. Существует алгоритм, который за $2 \left(\binom{k'}{\lfloor k'/2 \rfloor} + \binom{k'}{\lfloor k'/2 \rfloor + 1} \right)$ запросов на значение функции определяет, что $|X'_f| = |X_f|$, либо находит такую фиксацию переменных множества X'_f , что подфункция функции f , задаваемая этой фиксацией, существенно зависит хотя бы от одной переменной.

Доказательство. Определим набор a_1 длины n , фиксируя все его компоненты, соответствующие переменным из $Y_f \setminus X'_f$, единицами. Аналогично все компоненты набора a_0 , соответствующие переменным из $Y_f \setminus X'_f$, положим нулями. Рассмотрим грань G_1 , образованную точками $(11 \dots 11)$ и a_1 , и грань G_0 , образованную точками $(00 \dots 00)$ и a_0 (см. рис. 5). Воспользуемся алгоритмом Анселя [2] для расшифровки подфункций f_0 и f_1 функции f , определенных на гранях G_0 и G_1 , соответственно. Для расшифровки каждой подфункции потребуется не более $\binom{k'}{\lfloor k'/2 \rfloor} + \binom{k'}{\lfloor k'/2 \rfloor + 1}$ запросов на значение функции f .

Рис. 5. Наборы b_0 и b_1 задают искомую подходящую фиксацию

Возможны два случая.

- (1) Выполнено $f_0 = f_1$. Это значит, что множество X'_f содержит все существенные переменные функции f , то есть $|X'_f| = |X_f|$.
- (2) Существует набор b k' -мерного куба, такой что $f_0(b) \neq f_1(b)$ или, что то же, существуют наборы $b_0 < b_1$, $b_0 \in G_0$, $b_1 \in G_1$, отличающиеся только компонентами, соответствующими переменным из $Y_f \setminus X'_f$, такие что $f(b_0) \neq f(b_1)$ (см. рис. 5). Тогда набор b дает искомую подходящую фиксацию переменных из X'_f , поскольку подфункция функции f , определенная на грани G , образованной наборами b_0 и b_1 , зависит существенно хотя бы от одной переменной и не зависит от переменных из X'_f .

Лемма доказана.

Лемма 7. При $k \leq n$ и $k, n \rightarrow \infty$ справедливо соотношение

$$\varphi(k, n) \lesssim \frac{2^k}{\sqrt{k}} + k \log n.$$

Доказательство. Алгоритм F расшифровки произвольной k -существенной монотонной функции f выглядит следующим образом.

Шаг 1. Запускаем алгоритм F_1 из леммы 5 (ищем какие-нибудь существенные переменные).

Шаг 2. Запускаем алгоритм F_2 из леммы 6 (ищем подходящую фиксацию всех найденных к настоящему моменту существенных переменных). Если такая фиксация найдена, переходим к шагу 1, иначе заканчиваем работу.

Подсчитаем число запросов на значение функции, которое требуется алгоритму F для расшифровки некоторой функции f .

Суммарно выполнение шага 1 алгоритма F может потребовать не более $8k \log n$ запросов на значение функции.

Шаг 2 алгоритма F может выполняться не более k раз, поэтому суммарное число запросов на значение функции не может превышать $2 \sum_{i=1}^k \left(\binom{i}{\lfloor i/2 \rfloor} + \binom{i}{\lfloor i/2 \rfloor + 1} \right)$. Положим $g(n) = \binom{2n}{n}$. Воспользовавшись тождеством Паскаля для центральных биномиальных коэффициентов

$$\binom{2n}{n} = \frac{4n-2}{n} \binom{2n-2}{n-1},$$

получаем что $g(n) > 3g(n-1)$. Значит,

$$\sum_{i=1}^n g(i) < g(n) \left(1 + \frac{1}{3} + \frac{1}{9} + \dots \right) = \frac{3}{2} \left(1 - \frac{1}{3^n} \right) < \frac{3}{2} g(n).$$

Вспомяная, что $\binom{2n}{n} = O(2^n / \sqrt{n})$, получаем, что суммарно выполнение шага 2 алгоритма F приведет к по порядку не более чем $2^k / \sqrt{k}$ запросам на значение функции.

Итого, алгоритму F потребуется по порядку не более $2^k / \sqrt{k} + k \log n$ запросов, чтобы полностью расшифровать заданную k -существенную функцию f .

Справедливость теоремы 1 непосредственно следует из лемм 2 и 7.

Список литературы

1. Коробков В. К., О монотонных функциях алгебры логики. *Проблемы кибернетики* (1965) **13**, 5–28.
2. Hansel G., Sur le nombre des fonctions booléennes monotones de n variables. *C. R. Acad. Sci., Paris, Sér. B* (1966) **262**, 1088–1090.
3. Valiant L., A theory of the learnable. *Comm. ACM* (1984) **27**, 1134–1142.
4. Angluin D., Queries and concept learning. *Mach. Learn.* (1988) **2**, 319–342.
5. Кудрявцев В. Б., Андреев А. Е., Гасанов Э. Э., *Теория тестового распознавания*. Физматлит, Москва, 2007.
6. Кудрявцев В. Б., Гасанов Э. Э., Долотова О. А., Погосян Г. Р., *Теория тестирования логических устройств*. Физматлит, Москва, 2006.
7. Damaschke P., Adaptive versus nonadaptive attribute-efficient learning. *Mach. Learn.* (2000) **41**, 197–215.
8. Damaschke P., On parallel attribute-efficient learning. *J. Comput. Syst. Sci.* (2003) **67**, 46–62.
9. Damaschke P., Parallel attribute-efficient learning of monotone Boolean functions. *Lecture Notes Computer Sci.* (2000) **1851**, 504–512.
10. Gilbert E. N., Lattice theoretic properties of frontal switching functions. *J. Math. Phys.* (1954) **33**, 57–97.
11. Осокин В. В., Асимптотически оптимальный алгоритм расшифровки разбиения булевого куба на подкубы. *Интеллектуальные системы* (2007) **11**, 587–606.
12. Осокин В. В., О сложности расшифровки разбиения булевого куба на подкубы. *Дискретная математика* (2008) **20**, №2, 46–62.

Статья поступила 28.07.2009.