



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
М. В. ЛОМОНОСОВА

Факультет вычислительной математики и кибернетики
Кафедра математической кибернетики

Курсовая работа по теме

«Сложность расшифровки счетчика делимости на три»

Студент 318 группы

М. М. Сакович

Научный руководитель

д.ф-м.н., доцент. С. Н. Селезнева

Москва, 2017

Содержание

Введение	3
1. Основные определения	5
2. Постановка задачи	6
3. Основная часть	7
3.1. Решение	7
3.2. Результаты	10
Литература	11

Введение

В работе рассматривается задача расшифровки функций из некоторого класса. Эта задача состоит в следующем. Нам нужно определить значение на всех наборах некоторой функции f . При этом нам известно, что функция $f(\tilde{x}^n)$ принадлежит некоторому классу функций K . Более того, мы можем задавать вопросы о значении функции f на наборах и получать правильные ответы. Под сложностью расшифровки понимается число вопросов, которое следует задать, чтобы расшифровать любую функцию из класса K . Образно говоря, мы имеем дело с «черным ящиком», у которого n входов и один выход и про который известно, что он реализует некоторую функцию алгебры логики от n переменных из определенного класса K . Нам нужно определить, какую именно функцию он реализует.

Рассмотрим множество $\{A\}$ алгоритмов, решающих поставленную задачу. Любой паре — алгоритму A и функции $f(x_1, \dots, x_n)$ можно сопоставить число $\varphi_K(A, f)$ — число вопросов о значении функции f на наборах с помощью алгоритма A . Под сложностью алгоритма A понимаем функцию $\varphi_K(A, n) = \max_{f \in K^n} \varphi_K(A, f)$, где K^n — множество всех функций от n переменных из класса K . Под сложностью расшифровки класса K понимаем функцию $\varphi_K(n) = \min_F \varphi_K(A, n)$, где минимум берется по всем алгоритмам A , решающим поставленную задачу.

Впервые задача о расшифровке функций из некоторого класса была рассмотрена для монотонных функций. В 1963 году В. К. Коробков [1] получил следующие оценки расшифровки класса монотонных функций M

$$\varphi_M(n) \geq C_n^{\lfloor \frac{n}{2} \rfloor} + C_n^{\lfloor \frac{n}{2} \rfloor + 1}.$$

Окончательное решение этой задачи получил Ж. Ансель в [2]

$$\varphi_M(n) = C_n^{\lfloor \frac{n}{2} \rfloor} + C_n^{\lfloor \frac{n}{2} \rfloor + 1}.$$

В настоящее время важна задача расшифровки следующего вида. Пусть задана последовательность функций $g^k(x_1, \dots, x_k)$, — существенно зависящих от k переменных ($k = 0, 1, \dots$), и рассмотрим класс K функций, в котором K^n содержит функции g^k , существенно зависящие от переменных x_{i_1}, \dots, x_{i_k} , $k = 0, 1, \dots, n$, $1 \leq i_1 < \dots < i_k \leq n$. Поэтому, по сути, такая задача заключается в нахождении существенных переменных функции f .

В частности, в [3] были рассмотрены классы: $K = OR$, где $g^k = x_1 \vee \dots \vee x_k$; $K = PAR$, где $g^k = x_1 \oplus \dots \oplus x_k$; $K = THR$, где g^k — это пороговая функция с пороговым значением t , $0 \leq t \leq k$. Из [3] известно, что

$$\begin{aligned}\varphi_{OR}(n) &= \varphi_{PAR}(n) = n \\ \varphi_{THR}(n) &= n - 1 + \lceil \log_2(n + 1) \rceil.\end{aligned}$$

Кроме того рассматриваются подзадачи расшифровки, когда число существенных переменных k известно. В этом случае:

$$\begin{aligned}\lceil \log_2 \binom{k}{n} \rceil &\leq \varphi_{OR(k)}(n) \leq k \lceil \log_2 \left(\frac{n}{k} \right) \rceil + 2k - 2 \\ \lceil \log_2 \binom{k}{n} \rceil &\leq \varphi_{PAR(k)}(n) \leq O(k \log_2 \left(\frac{n}{k} \right)) \\ \lceil \log_2(kC_n^k + 2) \rceil &\leq \varphi_{THR(k)}(n) \leq 2(k - 1) \log_2 \left(\frac{n - 1}{k - 1} \right) + 6k - 6 + \lceil \log_2(n + 2) \rceil.\end{aligned}$$

Отметим, что алгоритм A расшифровки функций из класса K можно представить в виде двоичного дерева, которое называется дерево решений [4]. Дерево решений D_A — это корневое ориентированное дерево, из любой вершины которого исходит не более двух дуг. Вершины, из которых дуги не исходят, помечены функциями из K^n и называются листьями, остальные вершины помечены наборами, которые задаются в вопросе. Если из вершины исходит две дуги, то они помечены 0 и 1 (условный переход) и если исходит одна дуга, то она без пометки (безусловный переход). Тогда сложностью $\varphi_K(A, n)$ — это длина самой длинной цепи из корня в лист, не включая лист. Из такого представления алгоритмов расшифровки сразу следует, что для любого класса K

$$\varphi_K(n) \geq \lceil \log_2 |K^n| \rceil,$$

т.к. в дереве решений D_A для каждой функции $f \in K^n$ должен быть лист, помеченный этой функцией.

В настоящей работе рассматривается сложность задачи расшифровки счетчика делимости на 3.

1. Основные определения

Пусть $E_2 = \{0, 1\}$. Набор $(\alpha_1, \alpha_2, \dots, \alpha_n)$, где $\alpha_i \in E_2, 1 \leq i \leq n$, называется булевым или двоичным набором (вектором). Элементы набора называют компонентами. Число n называется длиной набора. Далее двоичный набор длины n будем обозначать $\tilde{\alpha}$. Множество всех двоичных наборов длины n образует n -мерный булев (или двоичный) куб, который называют также единичным n -мерным кубом и обычно обозначают B^n . Весом набора $\tilde{\alpha}$ (обозначение $|\tilde{\alpha}|$) называют число его координат, равных 1, т.е.

$$|\tilde{\alpha}| = \sum_{i=1}^n \alpha_i.$$

Наборы $\tilde{\alpha} \in B^n$ называют вершинами куба B^n . Множество всех вершин куба B^n , имеющих вес k , называется k -м слоем куба B^n (обозначение B_k^n). Набор, все координаты которого равны 0, будем называть нулевым. Набор, все координаты которого равны 1 — единичным. Вес нулевого набора равен 0, а вес единичного — n . Наборы будем называть соседними, если они различаются только в одной координате.

Функция $f(x_1, \dots, x_n)$, определенная на множестве $B^n = \{0, 1\}^n$ и принимающая значения из множества $\{0, 1\}$, называется функцией алгебры логики (булевой функцией). Множество всех булевых функций обозначим через P_2^n .

Переменная x_i ($1 \leq i \leq n$) функции $f(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n)$ называется существенной, если можно указать такие наборы $\tilde{\alpha}$ и $\tilde{\beta}$, соседние по i -й компоненте, (т.е. $\tilde{\alpha} = (\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n)$ и $\tilde{\beta} = (\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n)$), что $f(\tilde{\alpha}) \neq f(\tilde{\beta})$. В противном случае переменная называется фиктивной. Если у функции все переменные фиктивные, то она является константой.

2. Постановка задачи

Рассмотрим функции: $\tau_0^k(x_1, \dots, x_k), \tau_1^k(x_1, \dots, x_k), \tau_2^k(x_1, \dots, x_k) \in P_2^k$, такие что

$$\tau_i^k(\tilde{\alpha}) = \begin{cases} 1, & |\tilde{\alpha}| \bmod 3 = i, \\ 0, & |\tilde{\alpha}| \bmod 3 \neq i. \end{cases}$$

Пусть $A_i^n = \{\tau_i^k(x_{i_1}, \dots, x_{i_k}) \mid 1 \leq i_1 < i_2 < \dots < i_k \leq n, k = 0, \dots, n\} \mid i \in \{0, 1, 2\}$.

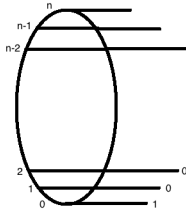


Рис. 2.1: Класс A_0^n .

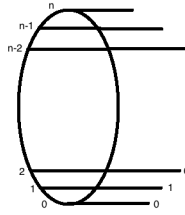


Рис. 2.2: Класс A_1^n .

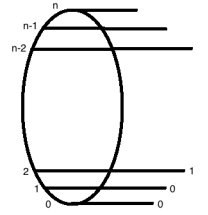


Рис. 2.3: Класс A_2^n .

Положим $A_i = \bigcup_{n \geq 0} A_i^n$. Найдем сложность $\varphi_{A_i}(n)$ расшифровки функции из класса A_i , $i \in \{0, 1, 2\}$.

3. Основная часть

3.1. Решение

Чтобы дать нижнюю оценку, посчитаем мощность классов A_i^n ($i = 0, 1, 2$).

Лемма 1. $\forall n \geq 1$ справедливо $|A_0^n| = |A_1^n| = 2^n$.

Доказательство. Рассмотрим класс $A_0(n)$. Он порождается функцией $\tau_0(x)$, зависящей от n переменных. Из этих n переменных можем выбрать 0 существенных переменных, т.е. C_n^0 , 1 существенную переменную, т.е. C_n^1 и т.д.. Таким образом,

$$|A_0(n)| = C_n^0 + C_n^1 + \dots + C_n^n = \sum_{k=0}^n C_n^k = 2^n.$$

Аналогично, $|A_1(n)| = 2^n$. □

Лемма 2. Мощность класса $A_2(n)$ равна $2^n - n$.

Доказательство. Класс $A_2(n)$ порождается функцией $\tau_2(x)$, которая зависит от n переменных. В отличие от классов $A_0(n)$ и $A_1(n)$, если функция из класса $A_2(n)$ существенно зависит от одной переменной, то она никогда не примет значение 1, значит мы не сможем распознать её. Таким образом

$$|A_2(n)| = C_n^0 + C_n^2 + \dots + C_n^n = \sum_{k=0}^n C_n^k - C_n^1 = 2^n - n.$$

□

Зная мощности классов, можно дать нижнюю оценку

1. $\lceil \log_2 |A_0(n)| \rceil = \log_2 2^n = n$
2. $\lceil \log_2 |A_1(n)| \rceil = \log_2 2^n = n$
3. $\lceil \log_2 |A_2(n)| \rceil = \lceil \log_2 (2^n - n) \rceil$.

Лемма 3. При любых $n > 2$ верно:

$$\lceil \log_2 (2^n - n) \rceil = n$$

Доказательство. Из свойств логарифмов:

$$\begin{aligned}\lceil \log_2(2^n - n) \rceil &= \lceil \log_2(2^n) + \log_2(1 - \frac{n}{2^n}) \rceil = \\ &= \lceil n + \log_2(1 - \frac{n}{2^n}) \rceil\end{aligned}$$

При $n > 2$ значение $\frac{n}{2^n} < 1$, убывает и ограничено нулем снизу.

Значит $-1 < \log_2(1 - \frac{n}{2^n}) < 0$. Отсюда следует, что $\lceil \log_2(1 - \frac{n}{2^n}) \rceil = 0$, откуда $\lceil n + \log_2(1 - \frac{n}{2^n}) \rceil = n$. \square

Посмотрим, достигаются ли нижние оценки для классов $A_i(n)$ ($i = 0, 1, 2$).

Построим алгоритм из n запросов для функции из класса $A_0(n)$.

Алгоритм A_0 .

Шаг 1. Рассмотрим функцию f из класса $A_0(n)$ на нулевом наборе

$$\tau_0(0, 0, \dots, 0) = \{0 \bmod 3 = 0\} = 1.$$

Не задавая вопроса, мы знаем, что любая функция из класса $A_0(n)$ на нулевом наборе имеет значение 1.

Шаг i . Рассмотрим набор $\alpha_i = (0, 0, \dots, 0, 1, 0, \dots, 0)$, в котором на i -м месте стоит 1, а на всех остальных местах 0. Если $f(\alpha_i) = 1$, то переменная x_i не влияет на значение функции, т.е. x_i - фиктивная переменная, в противном случае x_i - существенная

Применим шаг i для всех $i = 1, 2, \dots, n$.

Т.о. делаем n запросов $\alpha_1, \alpha_2, \dots, \alpha_n$ и для каждой из переменных x_1, x_2, \dots, x_n узнаем, является она существенной или фиктивной. Отметим, что данный алгоритм является безусловным.

Теорема 1. *Сложность расшифровки класса $A_0(n)$ равна n .*

Доказательство. Как показано выше, нижняя оценка равна n . Алгоритм A_0 дает верхнюю оценку, которая равна n . Следовательно сложность расшифровки класса $A_0(n)$ равна n . \square

Построим алгоритм из n запросов для функции из класса $A_1(n)$.

Алгоритм A_1 .

Шаг 1. Рассмотрим функцию f из класса $A_1(n)$ на нулевом наборе

$$\tau_1(0, 0, \dots, 0) = \{0 \bmod 3 = 0\} = 0.$$

Из этого следует, что не задавая вопроса, мы знаем, что любая функция из класса $A_1(n)$ на нулевом наборе имеет значение 0. Заметим, что если функция f существенно зависит от всех своих переменных, то на любом наборе из первого слоя, функция f принимает значение 1.

Шаг i . Рассмотрим набор $\alpha_i = (0, 0, \dots, 0, 1, 0, \dots, 0)$, в котором на i -м месте стоит 1, а на всех остальных местах 0. Если $f(\alpha_i) = 0$, то переменная x_i не влияет на значение функции, т.е. x_i - фиктивная переменная, в противном случае x_i - существенная

Применим шаг i для всех $i = 1, 2, \dots, n$.

Т.о. делаем n запросов $\alpha_1, \alpha_2, \dots, \alpha_n$ и для каждой из переменных x_1, x_2, \dots, x_n узнаем, является она существенной или фиктивной. Отметим, что данный алгоритм также является безусловным.

Теорема 2. *Сложность расшифровки класса $A_1(n)$ равна n .*

Доказательство. Как показано выше, нижняя оценка равна n . Алгоритм A_1 дает верхнюю оценку, которая равна n . Следовательно сложность расшифровки класса $A_1(n)$ равна n . \square

Построим алгоритм распознавания функции из класса $A_2(n)$

Алгоритм A_2 .

Шаг 1. Рассмотрим функцию f из класса $A_2(n)$ на наборе $\tilde{\alpha} = (1, 1, 0, \dots, 0)$. Если $f(\tilde{\alpha}) = 1$, то переходим к шагу 3. В противном случае переходим к шагу 2.

Шаг 2. Заменяем в наборе $\tilde{\alpha}$ первый встречающийся 0 на 1. Если $f(\tilde{\alpha}) = 1$, то переходим к шагу 3. Если набор $\tilde{\alpha}$ - единичный и $f(\tilde{\alpha}) = 0$, то все переменные функции f - фиктивные, иначе повторяем шаг 2.

Шаг 3. Не нарушая общности рассуждений, пусть последняя единица в наборе $\tilde{\alpha}$ стоит на k -м месте ($k \leq n$). Обозначим такой набор как $\tilde{\alpha}_k$. Тогда переменная x_k - существенная. Среди переменных x_1, x_2, \dots, x_{k-1} ровно одна существенная, т.к. $\tau_2(\tilde{\alpha})$ может принять первый раз значение 1 только в том случае, если на месте двух первых встретившихся существенных переменных в наборе $\tilde{\alpha}$ стоят единицы, а так как добавление единицы на k -е место обращает функцию $\tau_2(\tilde{\alpha})$ в 1, то x_k - вторая существенная переменная, а первая находится среди первых $k - 1$ переменных.

Шаг 4. Найдем первую существенную переменную. Для этого построим набор

$$\alpha_{\lceil \frac{k-1}{2} \rceil_{k-1}} = (1, 1, \dots, 1, 0, \dots, 0, 1, 0, \dots, 0),$$

который получается из набора $\tilde{\alpha}$ заменой 1, стоящих на местах $\lceil \frac{k-1}{2} \rceil, \lceil \frac{k-1}{2} \rceil + 1, \dots, k - 1$ на 0.

Если $f(\alpha_{\lceil \frac{k-1}{2} \rceil_{k-1}}) = 0$, то существенная переменная находится среди $x_{\lceil \frac{k-1}{2} \rceil}, \dots, x_{k-1}$, в противном случае существенная переменная среди $x_1, \dots, x_{\lfloor \frac{k-1}{2} \rfloor}$. Т.о мы сузили область поиска вдвое. Аналогичным образом будем заменять половину оставшихся 1 на 0, в зависимости того, в какую половину попадает существенная переменная, до тех пор, пока не останется одна 1, соответствующая переменной x_i .

Переменная x_i - является существенной. Пусть $\alpha_{ik} = (0, \dots, 0, 1, 0, \dots, 0, 1, 0, \dots, 0)$, где на i -м и k -м местах стоят 1, а на всех

остальных 0. Т.к. x_i и x_k - существенные, то $f(\alpha_{ik}) = 1$.

Отметим, что на шагах 1-4 алгоритм является условным.

Шаг 5. Осталось найти все существенные переменные среди $n - k$ оставшихся. Для этого составим $n - k$ безусловных запросов:

$$\begin{aligned}\alpha_{ik_1} &= (0, \dots, 0, 1, 0, \dots, 0, 1, 1, 0, \dots, 0) \\ \alpha_{ik_2} &= (0, \dots, 0, 1, 0, \dots, 0, 1, 0, 1, \dots, 0) \\ &\dots \\ \alpha_{ik_n} &= (0, \dots, 0, 1, 0, \dots, 0, 1, 0, 0, \dots, 1)\end{aligned}$$

Если $f(\alpha_{ik_j}) = 1$ (где $j = k + 1, \dots, n$), то переменная x_j - фиктивная. В противном случае x_j - существенная.

Посчитаем, сколько запросов нам понадобилось.

На первом шаге 1 запрос. На втором шаге нам понадобилось $k - 2$ запроса, где $k \leq n$. На 5-м шаге $n - k$. Четвертый шаг по сути является алгоритмом бинарного поиска, сложность которого $\leq \log_2(n)$. Таким образом, в худшем случае сложность приведенного алгоритма равна $1 + k - 2 + n - k + \log_2(n) = n - 1 + \log_2(n)$.

Теорема 3. *Асимптотическая сложность расшифровки класса $A_2(n)$ равна n .*

Доказательство. Как показано выше, нижняя оценка при $n > 2$ равна n . Рассмотрим случаи, когда $n = 1$ и $n = 2$.

При $n = 1$ функция $f(|\alpha|) \in A_2(n)$ является константой, так как никогда не примет значение 1. При $n = 2$ нижняя оценка равна 1. Получим верхнюю оценку. Сделаем запрос $\alpha(1, 1)$. Если $f(\alpha) = 1$, то обе переменные существенные, в противном случае функция является константой. Таким образом для $n = 1, 2$ нижняя и верхняя оценки совпадают и равны $n - 1$.

При $n > 2$ алгоритм A_2 дает верхнюю асимптотическую оценку, которая равна n . Следовательно сложность расшифровки класса $A_2(n)$ асимптотически равна n . \square

3.2. Результаты

Для классов $A_0(n)$ и $A_1(n)$ получена нижняя оптимальная оценка в n шагов, которая достигается на представленном безусловном алгоритме. Для класса $A_2(n)$ получена сложность асимптотически равная n .

Литература

- [1] Коробков В.К. *О монотонных функциях алгебры логики*. Сб. "Проблемы кибернетики вып 13, М., "Наука 1965, 5-27.
- [2] Ансель Ж. *О числе монотонных булевых функций от n переменных*. В кн. Кибернетический сборник. Новая серия. Вып. 5. М.: Мир, 1968, с. 53-57.
- [3] Ryuhei Uehara, Kensei Tsuchida, Ingo Wegener *Optimal attribute-efficient learning of disjunction, parity, and threshold functions*. 1996.
- [4] Ахо А., Хопкрофт Дж., Ульман Дж. *Построение и анализ вычислительных алгоритмов*. М.:Мир, 1979.
- [5] Гаврилов Г.П., Сапоженко А.А. *Сборник задач по дискретной математике*. М.:ФИЗМАТЛИТ, 2004.
- [6] Слезнева С.Н. *Лекции по "Избранным вопросам дискретной математики"* 3-й курс, 318 группа. 2016.
- [7] Алексеев В.Б. *Введение в теорию сложности алгоритмов*. М.: Изд. отдел ф-та ВМиК МГУ, 2002.