Московский государственный университет имени М. В. Ломоносова
Казанский (Приволжский) федеральный университет
Институт прикладной математики имени М. В. Келдыша РАН
ФИЦ "Информатика и управление" РАН

Проблемы теоретической кибернетики

Материалы XIX международной конференции

Казань

2021

ББК 22.18
УДК 519.7
П 78

П78 Проблемы теоретической кибернетики. Материалы XIX международной конференции. Под редакцией Ю. И. Журавлева. — Казань: Казанский федеральный университет, 2021. — 160 с.

Тематика конференции “Проблемы теоретической кибернетики” традиционно включает в себя следующие направления: синтез и сложность управляющих систем, надежность, контроль и диагностика управляющих систем, автоматы и языки программирования, теория графов, комбинаторика, теория кодирования, теория распознавания образов, математическое программирование и исследование операций, математическая теория интеллектуальных систем, прикладная математическая логика, теория функциональных систем, теория оптимального управления, приложения кибернетики в естествознании и технике.

Для научных работников и специалистов в области математической кибернетики, дискретной математики, информатики и их приложений.

Научное издание

ПРОБЛЕМЫ ТЕОРЕТИЧЕСКОЙ КИБЕРНЕТИКИ
МАТЕРИАЛЫ XIX МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ

Под общей редакцией академика РАН Ю. И. Журавлева

Редакционная группа
Ф. М. Аблаев, В. В. Кочергин, С. А. Ложкин

Ответственный за выпуск: Ф. М. Аблаев

Оглавление

Квантовое хеширование на состояниях высокой размерности

Аблаев Фарид Мансурович[1], **Васильев Александр Валерьевич**[2]

[1] Казанский федеральный университет,
Казанский физико-технический институт им. Е.К. Завойского, e-mail: fablayev@gmail.com
[2] Казанский федеральный университет,
Казанский физико-технический институт им. Е.К. Завойского, e-mail: vav.kpfu@gmail.com

В работе [1] нами проведена формализация понятия квантового хеширования и положено начало исследований свойств квантового хеширования. В дальнейших работах нами проведен анализ устойчивости квантовых хеш-функций [2], построены семейства алгоритмов [3] и протоколов [4], основанных на квантовом хешировании, а также предложен обобщенный вариант для входных данных из произвольной конечной абелевой группы [5].

Приведем некоторые необходимые в данной работе определения.

Пусть G – это конечная абелева группа (далее для примера в основном будем рассматривать $G = \mathbb{Z}_q$), χ_a являются ее характерами, проиндексированными элементами $a \in G$. Например, для $a \in \mathbb{Z}_q$ характер χ_a группы $\mathbb{Z}_q$ – это гоморфизм $\chi_a : \mathbb{Z}_q \to \mu_q$, где μ_q – это мультипликативная группа корней из единицы. Т.е. $\chi_a(x) = \omega^{ax}$, где $\omega = e^{\frac{2\pi i}{q}}$. Характер $\chi_0 \equiv 1$ называют тривиальным характером.

Определение. Множество $S \subseteq G$ называется множеством с ε-отклонением, если для любого нетривиального характера χ_a выполняется

$$\frac{1}{|S|}\left|\sum_{x\in S}\chi_a(x)\right| \leq \varepsilon. \tag{1}$$

В работе [6] показано, что множество S из $O(\log|G|/\varepsilon^2)$ элементов группы G, выбранных независимо и случайно в соответствии с равномерным распределением, является множеством с ε-отклонением с ненулевой вероятностью (т.е. демонстрируется их существование). В работе [7] приводятся явные конструкции таких множеств, основанные на кодах с исправлением ошибок.

Пусть G – это конечная абелева группа и пусть $S \subseteq G$ является множеством с ε-отклонением для некоторого $\varepsilon \in (0,1)$.

Определение. Определим квантовую функцию $\psi_S : G \to (\mathcal{H}^2)^{\otimes \log|S|}$ следующим образом

$$|\psi_S(a)\rangle = \frac{1}{\sqrt{|S|}}\sum_{x_j\in S}\chi_a(x_j)\,|j\rangle\,. \tag{2}$$

Соответственно, для $G = \mathbb{Z}_q$ эта функция имеет вид

$$|\psi_S(a)\rangle = \frac{1}{\sqrt{|S|}} \sum_{x_j \in S} e^{\frac{2\pi i}{q} a x_j} |j\rangle .$$

Покажем, что описанная выше квантовая функция ψ_S является (δ, ϵ)-устойчивой квантовой функцией для подходящих δ и ϵ.

Действительно, приведенная квантовая функция создает для произвольного элемента $a \in G$ его квантовый хеш-код, который является квантовым состоянием из $\log |S|$ кубит. Как было отмечено выше, размер S может иметь порядок $O(\log |G|/\varepsilon^2)$, и, таким образом, квантовое хеширование преобразует входные данные в экспоненциально меньшие по размеру образы. Другими словами, для произвольного $a \in G$, представленного $\log |G|$-битной последовательностью, число кубит в его квантовом хеш-коде будет описываться формулой.

$$\log |S| = O(\log \log |G| - \log \varepsilon). \tag{3}$$

Криптографические свойства квантовой хеш-функции, полностью определяются характеристиками множества с ε-отклонением $S \subseteq G$.

В частности, все попарные скалярные произведения хеш-кодов различных сообщений (что является мерой устойчивости квантового хеширования к коллизиям [1]) ограничены ε согласно [5].

Необратимость функции ψ_S основывается на теореме Холево и возможности построения квантового хеш-кода экспоненциально меньшего, чем его прообраз.

При этом получаемый в результате размер квантового хеш-кода является асимптотически оптимальным ввиду нижней оценки на размер множеств попарно различимых квантовых состояний [8]: для построения множества из 2^k квантовых состояний с попарным скалярным произведением меньше ε, эти состояния должны описывать как минимум $\Omega(\log(k/\varepsilon))$ кубит. Данная оценка влечет ограничение снизу на размер квантового хеш-кода $\Omega(\log \log |G| - \log \varepsilon)$, если квантовая хеш-функция ε-устойчива к коллизиям.

Здесь важно отметить, что описанная выше квантовая хеш-функция обеспечивает сбалансированное сочетание криптографических свойств в том числе благодаря применению так называемых “запутанных” состояний кубит, что обеспечивает “экспоненциальное” сжатие при сохранении устойчивости к коллизиям. При отказе от использования запутанных состояний удается добиться такой же устойчивости к коллизиям лишь при отказе от устойчивости к восстановлению прообраза, что является одним из ключевых свойств криптографической хеш-функции.

Вместе с тем, экспериментальная реализация таких максимально запутанных состояний по-прежнему является, пожалуй, труднейшей задачей при построении квантовых вычислений, поскольку требует реализации более сложных многокубитных операций, возможности попарного взаимодействия между кубитами и повышенной устойчивости к декогеренции.

В связи с этим представляется целесообразной реализация такой квантовой хеш-функции на основе состояний высокой размерности (так называемых "кудитов"). Каждая физическая система, описываемая кудитом, имеет $d > 2$ устойчивых базисных состояний, и с ее помощью можно реализовать квантовую хеш-функцию ψ_S, где $|S| = d$. При такой реализации сохраняется внешний вид ψ_S, а также ее устойчивость к коллизиям, а ее устойчивость к инверсии можно оценить сверху величиной δ, где

$$\delta < \frac{d}{|G|}.$$

Работа выполнена при поддержке РНФ (проект № 19-19-00656).

Список литературы

[1] Ablayev F. M., Vasiliev A. V. Cryptographic quantum hashing // Laser Phys. Lett. — 2014. — V. 11, No 2. — P. 025202.

[2] Ablayev F., Ablayev M., Vasiliev A. On the balanced quantum hashing // Journal of Physics: Conference Series. — 2016. — V. 681, No. 1. — P. 012019.

[3] Ablayev F., Vasiliev A. Computing Boolean Functions via Quantum Hashing // Computing with New Resources, C.S. Calude et al. (Eds.): Gruska Festschrift, Lecture Notes in Computer Science. — 2014. — V. 8808. — P. 149–160.

[4] Vasiliev A. Quantum communications based on quantum hashing // International Journal of Applied Engineering Research. — 2015. — V. 10, No. 12. — P. 31415–31426.

[5] Vasiliev A. Quantum hashing for finite abelian groups // Lobachevskii Journal of Mathematics. — 2016. — V. 37, No. 6. — P. 751–754.

[6] Alon N., Roichman Y. Random Cayley graphs and expanders // Random Structures & Algorithms. — 1994. — V. 5, No. 2. — P. 271–284.

[7] Chen S., Moore C., Russell A. Small-bias sets for nonabelian groups // Lecture Notes in Computer Science. — 2013. — V. 8096. — P. 436–451.

[8] Buhrman H., Cleve R., Watrous J., de Wolf R. Quantum fingerprinting // Phys. Rev. Lett. — 2001. — V. 87, No. 16. — P. 167902.

О замкнутых классах, содержащих все полиномы, в частичной k-значной логике

Алексеев Валерий Борисович

МГУ им. М. В. Ломоносова, e-mail: vbalekseev@rambler.ru

В [1] начато изучение связей между решетками замкнутых классов в k-значной логике P_k и в частичной k-значной логике P_k^*. А именно: для каждого предполного класса в P_2 изучалась мощность семейства замкнутых классов в P_2^*, содержащих этот класс. Было установлено, что для предполного класса линейных булевых функций эта мощность континуальна, а для остальных предполных классов конечна. Несколько изменив задачу, Д. Лау (см. [2]) предложила для замкнутого класса A из P_k рассматривать интервал $\mathcal{I}(A)$ в решетке замкнутых классов в P_k^*, состоящий из тех классов, в которых множество всюду определенных функций совпадает с A. Задача определения мощности семейства $\mathcal{I}(A)$ для различных замкнутых классов («проблема Лау») активно изучалась многими зарубежными математиками. Однако только в 2017 году мощность $\mathcal{I}(A)$ окончательно установлена для всех замкнутых классов из P_2 [3].

В P_k^* при $k \geq 3$ мощность интервала $\mathcal{I}(A)$ установлена для всех предполных классов из P_k, кроме некоторых классов монотонных функций. Для предполных классов монотонных функций найдены необходимые и достаточные условия, при которых интервал $\mathcal{I}(A)$ бесконечен [4-6], однако пока неизвестна точная мощность этих интервалов.

Автором предложено [7,8] изучать более узкий интервал замкнутых классов $Int(A)$ — это семейство всех замкнутых классов (относительно суперпозиции) в частичной k-значной логике, содержащих заданный замкнутый класс A k-значной логики, и состоящих только из функций, доопределимых до какой-нибудь функции из A. Интервалы $\mathcal{I}(A)$ и $Int(A)$ могут существенно различаться. Например, если A — предполный класс монотонных k-значных функций, то семейство $\mathcal{I}(A)$ может быть и конечным и бесконечным, но семейство $Int(A)$ всегда конечно и состоит из 6 замкнутых классов [2]. Отметим также, что для класса L булевских линейных функций мощность $\mathcal{I}(L)$ континуальна, а про $Int(L)$ доказано только, что это семейство бесконечно [1], но точная мощность его неизвестна.

Одним из важных замкнутых классов в P_k является класс Pol_k всех функций, представимых полиномом по модулю k. В данной работе мы исследуем интервал $Int(Pol_k)$. Если k — простое число, то $Pol_k = P_k$. В этом случае интервал $Int(Pol_k)$ содержит ровно 3 замкнутых класса, а именно: P_k, P_k^*, $P_k \cup \{*\}$, где $\{*\}$ — множество всех нигде не определенных функций из P_k^* от любого числа переменных.

Пусть $Pr(k)$ — множество всех предикатов на $E_k = \{0, 1, \ldots, k-1\}$ от любого числа переменных. Пусть A — замкнутый (относительно операции суперпозиции) класс из P_k. Через $Z(A)$ будем обозначать семейство всех подмножеств в $Pr(k)$, содержащих все предикаты $1(x_1, \ldots, x_n) \equiv 1$ от любого числа переменных и замкнутых относительно произвольного переименования переменных в предикатах, добавления и изъятия фиктивных переменных, подстановки в предикат функций из A вместо переменных и конъюнкции предикатов. Для исследования интервалов $Int(A)$ оказывается полезной следующая доказанная автором теорема.

Теорема 1 [8]. *Для любого замкнутого класса A, содержащего все селекторы, семейства $Int(A)$ и $Z(A)$ изоморфны как частично упорядоченные множества относительно включения.*

В [8] с использованием этой теоремы автором получен следующий результат.

Теорема 2 [8]. *Если $k = p_1 \cdot p_2$, где p_1 и p_2 — различные простые числа, то интервал $Int(Pol_k)$ состоит из 7 замкнутых классов.*

В [7] автором доказана следующая теорема.

Теорема 3 [7]. *Пусть натуральное число k имеет хотя бы 3 различных простых делителя. Тогда семейство $Int(Pol_k)$ бесконечно, в частности, в нем есть бесконечно убывающая (относительно вложения) цепочка различных замкнутых классов.*

В данной работе исследуются оставшиеся случаи.

Теорема 4. *Пусть $k = p^2$, где p — простое число. Тогда семейство $Int(Pol_k)$ бесконечно, в частности, в нем есть бесконечно возрастающая (относительно вложения) цепочка различных замкнутых классов.*

Доказательство теоремы 4 основано на следующей конструкции. Через $\bigwedge \bigvee_m R(+)$ обозначим множество всех предикатов на E_{p^2} от любого числа переменных n, которые можно выразить формулой, представляющей собой конъюнкцию любого числа формул, каждая из которых представляет собой дизъюнкцию не более чем m формул, являющихся произвольными одноместными предикатами на E_{p^2}, в которые подставлены произвольные линейные функции на E_{p^2}. Пусть $a_i \in E_p = \{0, 1, \ldots, p-1\}$ для $i = \overline{1, n}$. Под n-мерным p-кубом с параметрами $(a_1, a_2, \ldots, a_n)$ будем понимать множество всех наборов $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in E_{p^2}$ таких, что $\alpha_i \equiv a_i \pmod p$ для всех i. Через T_m обозначим множество всех предикатов на E_{p^2} от любого числа переменных n, значения которых на любом n-мерном p-кубе совпадают со значениями некоторого предиката из семейства $\bigwedge \bigvee_m R(+)$ (на разных p-кубах могут быть разные предикаты). Тогда теорема 3 выводится с использованием теоремы 1 из следующего утверждения.

Теорема 5. *При $k = p^2$, где p — простое число, все классы T_m принадлежат семейству $Z(Pol_k)$, различны и образуют в $Z(Pol_k)$ бесконечно возрастающую цепочку: $T_1 \subset T_2 \subset \ldots \subset T_m \subset \ldots$.*

Из теоремы 4 с учетом теорем 2 и 3 можно получить окончательный результат.

Теорема 6. *Если $k \geq 2$ — простое число или произведение двух различных простых чисел, то семейство $Int(Pol_k)$ содержит конечное число замкнутых классов; в остальных случаях это семейство бесконечно.*

Работа выполнена при поддержке РФФИ (проект № 19-01-00200-а).

Список литературы

[1] Алексеев В. Б., Вороненко А. А. О некоторых замкнутых классах в частичной двузначной логике // Дискретная математика. — 1994. — Т. 6, вып. 4. — С. 58–79.

[2] Lau, D. Function algebras on finite sets: a basic course on many-valued logic and clone theory. Springer Monographs in Mathematics. — Springer, Berlin, 2006. 668 pp.

[3] A solution to a problem of D. Lau: Complete classification of intervals in the lattice of partial Boolean clones / Couceiro M., Haddad L., Schoelzel K., Waldhauser T. // J. Mult.-Valued Logic Soft Comput. — 2017. — V. 28. — P. 47–58.

[4] Дудакова О. С. О классах частичных монотонных функций шестизначной логики // Проблемы теоретической кибернетики: XVIII международная конференция (Пенза, 19-23 июня 2017 г.): Материалы: Под редакцией Ю.И. Журавлева. — М. : МАКС Пресс, 2017. — С. 78-81.

[5] Алексеев В. Б. О замкнутых классах в частичной k-значной логике, содержащих класс монотонных функций // Дискретная математика. — 2018. — Т. 30, вып. 2. — С. 3–13.

[6] Дудакова О. С. Построение бесконечного семейства классов частичных монотонных функций многозначной логики // Вестник Московского университета. Серия 1: Математика. Механика. — 2019. — № 1. — С. 3–7.

[7] Алексеев В. Б. О замкнутых классах в частичной k-значной логике, содержащих все полиномы // Дискретная математика. — 2021. — Т. 33, вып. 2. — С. 6–19.

[8] Alekseev Valeriy B. On some intervals of partial clones // J. Mult.-Valued Logic Soft Comput. (принято в печать).

О мощности слоёв декартовых степеней некоторых ЧУМ и множеств с функцией веса

Андреева Татьяна Владимировна*, Семенов Юрий Станиславович

*Российский университет транспорта, e-mail: t-v-andreeva@mail.ru, yuri_semenoff@mail.ru

В работе найдена асимптотика мощности слоев в декартовых степенях ряда общих и частных примеров некоторых ЧУМ и множеств с функцией веса.

1. Весовые функции на множестве. Весовой функцией (функцией веса) w на множестве X называется некоторое отображение $w : X \to \mathbb{R}$, т.е. каждому $x \in X$ приписывается вес $w(x)$. Заметим, что веса могут быть и отрицательными. Например, если X — ранжированное частично упорядоченное множество (ЧУМ), то обычно функция веса совпадает с функцией ранга на этом множестве. Также заметим, что неизоморфные ранжированные ЧУМ могут иметь одну и ту же весовую функцию.

Пару (X, w) будем называть взвешенным множеством. Часто обозначение весовой функции w будет опускаться. Если k — некоторое значение весовой функции, то k–слоем назовём множество всех $x \in X$ таких, что $w(x) = k$, т.е. k–слой состоит из всех элементов одного и того же веса k. Формула $w(x_1, \dots, x_n) = w(x_1) + \dots + w(x_n)$ определяет весовую функцию (обозначаемую той же буквой w) на декартовой степени X^n.

2. Пример (a, b)-множества. Пусть a, b — натуральные числа и пусть

$$X = \{x_{-2,j}, x_{-1,i}, x_0, x_{1,i}, x_{2,j}\}; \quad i = 1, \dots, a;\ j = 1, \dots, b,$$

— множество из $1 + 2a + 2b$ элементов.

Припишем каждому элементу следующие веса:

$$w(x_{\pm 2,j}) = \pm 2\,, \quad w(x_{\pm 1,i}) = \pm 1\,, \quad w(x_0) = 0\,.$$

Обозначим k-слой в X^n через $L_k^{(n)}$, а его мощность — через $A_k^{(n)}$. Нулевой слой $L_0^{(n)}$ назовём центральным слоем.

Хорошим инструментом для нахождения мощности слоёв служит производящая функция:

$$f_n(z) = \sum_k A_k^{(n)} z^k\,.$$

На самом деле, эта функция (многочлен Лорана) имеет вид

$$f_n(z) = (bz^{-2} + az^{-1} + 1 + az + bz^2)^n\,.$$

Теорема 1. *При $n \to \infty$ справедливо равенство:*

$$A_0^{(n)} = (1+2a+2b)^n \cdot \sqrt{\frac{1+2a+2b}{4a+16b}} \cdot \frac{(2n-1)!!}{(2n)!!} \left(1 - \frac{\gamma(a,b)}{n} + O\Big(\frac{1}{n^2}\Big)\right),$$

где

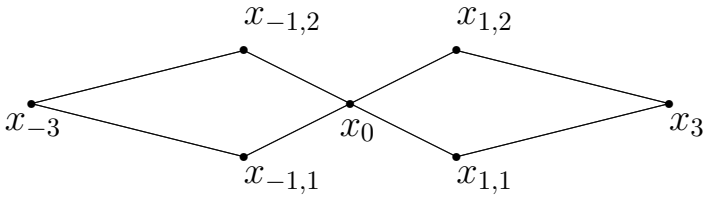
$$\gamma(a,b) = \frac{2a^2 - 2ab + 32b^2 - a - 16b}{16(a+4b)^2}.$$

Этот результат уточняет результат из работы В. Б. Алексеева [1].

3. Пример множества "летучая мышь". Пусть

$$X = \{x_{-3}, x_{-1,i}, x_0, x_{1,i}, x_3\}, \quad i = 1, 2,$$

— множество из 7 элементов:



Припишем каждому элементу следующие веса:

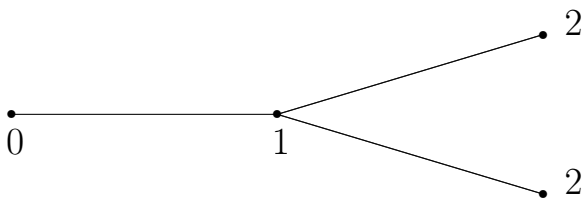
$$w(x_{\pm 3}) = \pm 3\,, w(x_{\pm 1,i}) = \pm 1\,, \quad w(x_0) = 0\,.$$

Отметим, что этот пример – модифицированный вариант одного из примеров (ЧУМ) В. Б. Алексеева.

Теорема 2. *При $n \to \infty$ справедливо равенство:*

$$A_0^{(n)} = 7^n \cdot \sqrt{\frac{7}{44}} \cdot \frac{(2n-1)!!}{(2n)!!} \left(1 + \frac{97}{44^2} \cdot \frac{1}{n} + O\Big(\frac{1}{n^2}\Big)\right),$$

4. О слоях в декартовых степенях 2–метёлок. Рассмотрим 2–метёлку — множество из 4 элементов с весами 0, 1, 2, 2:



Производящая функция 2–метёлки имеет вид $f(z) = 1 + z + 2z^2$. Отметим, что, в отличие от предыдущих примеров, 2–метёлка не является симметричной относительно своего центрального 1–слоя.

Можно показать, что $L_{5n}^{(4n)}$ — это самый мощный слой из слоёв $L_k^{(4n)}$.

Теорема 3. *Для 2–метёлки при $n \to \infty$ справедливо равенство:*

$$A_{5n}^{(4n)} = A_{5n}^{(4n)}(2) = \frac{4 \cdot 4^{4n}}{\sqrt{22}} \cdot \frac{(8n-1)!!}{(8n)!!} \left(1 - \frac{2070}{44^3} \cdot \frac{1}{n} + O\Big(\frac{1}{n^2}\Big)\right).$$

В то же время для t–метёлок с производящей функцией $f(z) = 1+z+tz^2$ Т.В. Андреевой (см., например, [2]) была получена оценка мощности *центральных* слоёв (которые не являются самыми мощными):

$$A_n^{(n)}(t) = \frac{(1+2\sqrt{t})^{n+1/2}}{2t^{1/4}} \cdot \frac{(2n-1)!!}{(2n)!!} \left(1 - \frac{2\sqrt{t}-1}{16\sqrt{t}} \cdot \frac{1}{n} + O\Big(\frac{1}{n^2}\Big)\right).$$

$$A_{n+1}^{(n)}(t) = \frac{t^{1/2}\big(1+2\sqrt{t}\big)^{n+1/2}}{2t^{1/4}} \cdot \frac{(2n-1)!!}{(2n)!!} \left(1 - \frac{3+10\sqrt{t}}{16\sqrt{t}} \cdot \frac{1}{n} + O\Big(\frac{1}{n^2}\Big)\right).$$

В частности, для 2–метёлок мы получим формулы:

$$A_n^{(n)}(2) = \frac{\sqrt{8+2\sqrt{2}}}{4}\big(1+2\sqrt{2}\big)^n \cdot \frac{(2n-1)!!}{(2n)!!} \left(1 - \frac{4-\sqrt{2}}{32} \cdot \frac{1}{n} + O\Big(\frac{1}{n^2}\Big)\right).$$

$$A_{n+1}^{(n)}(2) = \frac{\sqrt{4+\sqrt{2}}}{2}\big(1+2\sqrt{2}\big)^n \cdot \frac{(2n-1)!!}{(2n)!!} \left(1 - \frac{20+3\sqrt{2}}{32} \cdot \frac{1}{n} + O\Big(\frac{1}{n^2}\Big)\right).$$

Таким образом, мощность центрального слоя $L_{4n}^{(4n)} = L_{4n}^{(4n)}(2)$ равна:

$$A_{4n}^{(4n)}(2) = \frac{\sqrt{4+\sqrt{2}}}{2\sqrt{2}}\big(1+2\sqrt{2}\big)^{4n} \cdot \frac{(8n-1)!!}{(8n)!!} \left(1 - \frac{4-\sqrt{2}}{128} \cdot \frac{1}{n} + O\Big(\frac{1}{n^2}\Big)\right).$$

Для доказательства теорем мы рассматриваем производящую функцию мощности слоёв декартовой степени множества, коэффициенты которой выражаются по интегральной формуле Коши. Для оценки коэффициентов мы используем классические методы математического анализа. Подобные оценки можно получать с точностью до $O\big(n^{-s}\big)$ при $s = 3, 4, \ldots$

Авторы благодарят Н.В. Смирнову за помощь в работе.

Список литературы

[1] Алексеев В. Б. О числе монотонных k–значных функций. // Сб. Проблемы кибернетики, М.: Наука — 1974. — Вып. 28, С. 5–24.

[2] Андреева Т. В., Семенов Ю. С. О мощности слоёв некоторых частично упорядоченных множеств. // Учен. зап. Казан. ун-та. Сер. Физ.-матем. науки, 162, № 3, 2020, 269 — 284.

О кодировании и декодировании непомеченных деревьев

Балагура Анна Александровна[1], **Кузьмин Олег Викторович**[2]

[1] Иркутский государственный университет, e-mail: irk25@rambler.ru
[2] Иркутский государственный университет, e-mail: quzminov@mail.ru

Введение

Сегодня значимость исследования деревьев сложно переоценить. Они играют важную роль в современных подходах анализа данных. Задание весовой функции на множестве ребер или вершин дерева позволяет строить различные вероятностные модели и применять их в задачах принятия решения [1]. Рассматриваемые в настоящей статье деревья, в которых из каждой внутренней вершины исходит не менее двух преемников, встречаются в работах Шредера, Стенли [2], Кузьмина [3, 4] и др. В настоящей работе рассматривается перечисление подмножества плоских непомеченных деревьев по числу концевых вершин, преемников корня и последовательностью степеней внутренних вершин дерева при df-обходе. Кодирование осуществляется неубывающими кортежами, присвоением меток внутренним вершинам дерева.

Основные понятия и обозначения

Корневое дерево можно определить рекурсивно. Корневое дерево d есть такое множество вершин, что: одна специально выбранная вершина называется *корнем* дерева d, оставшиеся вершины (исключая корень) разбиты на $m \geq 0$ непересекающихся непустых множеств, каждое из которых является деревом [4]. Вершины, не имеющие преемников, называются *концевыми вершинами*. Вершины, имеющие преемников, называют *внутренними вершинами*. В настоящей работе рассматриваются плоские деревья [2], т.е. поддеревья в любой вершине линейно упорядочены.

Пусть $n \geq 2,\ 2 \leq k \leq n$. Обозначим $\bar{D}(n)$ — множество непомеченных плоских корневых деревьев, имеющих в точности n концевых вершин, у которых из каждой внутренней вершины (и корня) исходит не менее двух вершин; $\bar{D}(n, k)$ — множество корневых непомеченных корневых деревьев, имеющих в точности n концевых вершин и k преемников корня, у которых из каждой внутренней вершины исходит не менее двух вершин. Будем придерживаться введенных в [3] обозначений:
$v(n, k)$ — количество вершин в дереве d, не считая корень,
$w(n, k)$ — количество внутренних вершин в дереве d, не считая корень.

Кортежем длины n называется упорядоченный набор целых неотрицательных чисел $(i_1, ..., i_n)$ [4]. Пусть $a_1, ..., a_n \in N, a_1 \leq ... \leq a_n$. Обозначим $I(a_1, ..., a_n) = \{(i_1, ..., i_n) |\ i_j \leq a_j,\ i_j \leq i_{j+1},\ 1 \leq j \leq n\}$, множество I назо-

вем множеством неубывающих кортежей.

Поскольку в рассматриваемых деревьях важен порядок внутренних вершин, чтобы его зафиксировать будем придерживаться известного способа обхода дерева по правилу 1.

Правило 1: будем обходить дерево в соответствии с df-порядком [2], т.е. в глубину, начиная от корня, совершая обход слева направо.

Назовем типом дерева $d \in \bar{D}(n,k)$ последовательность $(n_1, n_2, ..., n_{w(n,k)})$ степеней внутренних вершин дерева, без учета корня при обходе дерева по правилу 1. Обозначим $\tilde{D}(n,k,n_1,n_2,...,n_{w(n,k)})$ — множество деревьев $d \in \bar{D}(n,\ k)$ типа $(n_1, n_2, ..., n_{w(n,k)})$.

Алгоритмы

Опишем способ кодирования деревьев неубывающими кортежами.

Алгоритм 1 (кодирования дерева)

Вход алгоритма: *диаграмма дерева* $d \in \bar{D}(n,\ k)$.

Выход алгоритма: *код дерева* $a_1, ..., a_{w(n,k)}$.

Обходим дерево по правилу 1, *пусть* $v_1, ..., v_{w(n,k)}$ — *последовательность обхода его внутренних вершин (за исключением корня).*

Для всех внутренних вершин $v_i, 1 \le i \le w(n,k)$ *выполняем:*

считаем $c(v_i)$ — *число концевых вершин пройденных до внутренней вершины* v_i,

кодируем вершину v_i: $a_i := c(v_i) + 1$.

Отметим, что из алгоритма 1 следует, что код дерева является неубывающим кортежем длины $w(n,k)$.

Для описания алгоритма декодирования зададим дерево в виде графа $T =< V, R >$, где V — множство вершин, R — множество ребер. Множество V считаем упорядоченным в соответствии с порядком обхода вершин дерева по правилу 1.

Алгоритм 2 (декодирования дерева)

Вход алгоритма: *код дерева* $a_1, ..., a_{w(n,k)}$, *тип дерева* $(n_1, ..., n_{w(n,k)})$.

Выход алгоритма: *дерево* $T \in \bar{D}(n,\ k)$, $T =< V, R >$.

Пусть $V = \{v_0\}, R$— *пустое множество.*

Строим все k *преемников корня: добавляем вершины* $v_1, ..., v_k$ *в* V, *ребра* $(v_0, v_1), ..., (v_0, v_k)$ — *в* R.

Для $1 \le i \le w(n,k)$ *выполняем:*

обходя по правилу 1 построенное дерево отсчитываем a_i *концевых вершин,*

строим n_i *преемников у* a_i*-й вершины:*

добавляем элементы $v_{k+(n_1+...+n_{i-1})+1}, ..., v_{k+(n_1+...+n_i)+1}$ *после* a_i*-й вершины в* V,

переобозначаем a_i*-ю вершину через* w_{a_i},

добавляем элементы $(w_{a_i}, v_{k+(n_1+\ldots+n_{i-1})+1}), \ldots, (w_{a_i}, v_{k+(n_1+\ldots+n_i)+1})$ *в* R.

Предложенные алгоритмы позволили доказать существование взаимнооднозначного соответствия между изучаемым множеством деревьев и множеством неубывающих кортежей [5]. Для нахождения мощности множества кортежей использован подход, основанный на обобщенной пирамиде Паскаля [4, 6].

Работа выполнена при поддержке РФФИ (проект № 20-41-385001).

Список литературы

[1] Кузьмин О. В., Аталян А. В. Деревья принятия решений в задачах диагностики и прогнозирования // Прикладные задачи дискретного анализа: сб. науч. тр. – Иркутск: Изд-во ИГУ, 2019.— Вып. 5.— С. 64—80.

[2] Стенли Р. Перечислительная комбинаторика. Деревья, производящие функции и симметрические функции. Пер. с англ. — М.: Мир, 2005. — 767 с.

[3] Балагура А. А., Кузьмин О. В. Перечислительные свойства комбинаторных полиномов разбиений // Дискретн. анализ и исслед. опер. — 2011. Т.18 №1. — С. 3-14.

[4] Кузьмин О. В. Обобщенные пирамиды Паскаля и их приложения. — Новосибирск: Наука. Сибирская издательская фирма РАН, 2000. — 294 с.

[5] Balagura A. A., Kuzmin O. V. Encoding and decoding algorithms for unlabeled trees. // Journal of Physics: Conference Series. — 2021. — Vol.1847, Iss.1. — 012027.

[6] Балагура А. А., Кузьмин О. В. Обобщенные пирамиды Паскаля и им обратные // Дискрет. матем. — 2007. — Т.19 №4.—С.108—116.

О локальной ассоциации графов Карриазо–Фернандеса с АСМ-гиперповерхностями 6-мерных эрмитовых подмногообразий алгебры октав

Банару Михаил Борисович[1], **Банару Галина Анатольевна**[2]

[1] Смоленский государственный университет, e-mail: mihail.banaru@yahoo.com
[2] Смоленский государственный университет, e-mail: mihail.banaru@yahoo.com

1. Основным приложением дискретной математики в контактной и эрмитовой геометриях (под которыми мы понимаем геометрии почти кон-

тактных метрических многообразий и почти эрмитовых многообразий, соответственно) долгое время считалась характеризация почти контактных метрических гиперповерхностей почти эрмитовых многообразий в терминах их типовых чисел (характеризация Такаги–Курихары). В последнее время, однако, развивается и другое интересное направление, связанное с использованием теории графов в данных разделах геометрии. Это направление задано, в основном, работами испанских математиков Карриазо, Фернандеса и Родригеса–Идальго (см., например [1, 2]). Карриазо и Фернандес установили, что всякое подмногообразие почти эрмитова многообразия допускает (в локальном смысле) ассоциацию с некоторым графом. Причем конструкция такой ассоциации была определена для многих важных случаев [1].

2. Напомним [3], что почти эрмитовой (almost Hermitian, AH-) структурой на многообразии четной размерности M^{2n} называют пару $\{J,\ g = \langle\cdot,\ \cdot\rangle\}$, где J — почти комплексная структура, а $g = \langle\cdot,\ \cdot\rangle$ — риманова метрика на этом многообразии. При этом почти комплексная структура J и метрика $g = \langle\cdot,\ \cdot\rangle$ должны удовлетворять условию

$$\langle JX,\ JY\rangle = \langle X,\ Y\rangle\,,\quad X, Y \in \aleph(M^{2n}),$$

где $\aleph(M^{2n})$ — модуль гладких векторных полей на многообразии M^{2n} . Многообразие с фиксированной на нем почти эрмитовой структурой называется почти эрмитовым (AH-) многообразием.

Также напомним [3], что под почти контактной метрической (almost contact metric, ACM-) структурой на многообразии N нечетной размерности понимают четверку тензорных полей $\{\Phi, \xi, \eta, g\}$, когда для нее выполняются следующие условия:

$$\eta(\xi) = 1;\ \Phi(\xi) = 0;\ \eta \circ \Phi = 0;\ \Phi^2 = -id + \xi \otimes \eta;$$

$$\langle \Phi X, \Phi Y\rangle = \langle X, Y\rangle - \eta(X)\eta(Y),\ X, Y \in \aleph(N).$$

Здесь $g = \langle\cdot,\ \cdot\rangle$ — риманова метрика, ξ — структурный вектор, Φ — поле тензора типа (1, 1), η — структурная форма. К самым важным видам ACM-структур относятся косимплектическая структура, структуры Эндо, Сасаки и Кенмоцу, а также их многочисленные обобщения. Известно [4], что почти контактная метрическая структура индуцируется на всякой ориентируемой гиперповерхности AH-многообразия.

3. Во многих работах (см., например, [5, 6, 7]) авторами изучались 6-мерные ориентируемые подмногообразия алгебры октав, на которых 3-векторные произведения Брауна–Грея [8] порождают AH-структуру. В основном рассматривались 6-мерные подмногообразия алгебры Кэли с эрмитовой (то есть интегрируемой почти эрмитовой) структурой. Был выделен

особый тип таких 6-мерных эрмитовых подмногообразий алгебры октав — так называемые уплощающиеся (planar) подмногообразия [5, 7]. К АСМ-гиперповерхностям некоторых классов уплощающихся подмногообразий алгебры Кэли оказалось возможным присоединить графы Карриазо–Фернандеса.

Теорема 1. *Граф Карриазо–Фернандеса ассоциируется (в локальном смысле) со всякой косимплектической, слабо косимплектической, сасакиевой и квазисасакиевой гиперповерхностями 6-мерного уплощающегося эрмитова подмногообразия алгебры Кэли.*

При этом установлено, что если АСМ-структура на гиперповерхности 6-мерного уплощающегося эрмитова подмногообразия алгебры октав окажется структурой косимплектического типа (структурой Кириченко–Ускорева [9, 10]), которая не является ни косимплектической структурой, ни структурой Кенмоцу, то присоединение графа Карриазо–Фернандеса к гиперповерхности возможно лишь в том случае, когда эрмитова структура на этом подмногообразии алгебры октав является келеровой.

Теорема 2. *Граф Карриазо–Фернандеса ассоциируется (в локальном смысле) с гиперповерхностью косимплектического типа 6-мерного уплощающегося эрмитова подмногообразия алгебры Кэли в том и только том случае, если эрмитова структура на этом многообразии является келеровой.*

Отметим, что присоединение графа Карриазо–Фернандеса к произвольной гиперповерхности косимплектического типа келерова многообразия размерности не ниже шести — это относительно новый результат. Он был представлен на Международном семинаре «Дискретная математика и её приложения» имени академика О.Б. Лупанова в 2019 году [11].

Список литературы

[1] Carriazo A., Fernandez L. M. Submanifolds associated with graphs // Proc. Amer. Math. Soc. — 2004. — V. 132(11). — P. 3327–3336.

[2] Carriazo A., Fernandez L. M., Rodriguez–Hidalgo A. Submanifolds weakly associated with graphs // Proc. Indian Acad. Sci. (Math. Sci.). V. 119. N3. 2009. P.297–318.

[3] Кириченко В. Ф. Дифференциально-геометрические структуры на многообразиях // Одесса: Печатный дом, — 2013.

[4] Banaru M. B., Kirichenko V. F. Almost contact metric structures on the hypersurface of almost Hermitian manifolds // Journal of Mathematical Sciences (New York). — 2015. — V. 207. N4. — P. 513–537.

[5] Banaru M. B., Banaru G. A. A note on six-dimensional planar Hermitian submanifolds of Cayley algebra // Известия Академии наук Республики Молдова. Математика. — 2014. — N 1(74). — P. 23–32.

[6] Banaru M. B. Geometry of 6-dimensional Hermitian manifolds of the octave algebra // Journal of Mathematical Sciences (New York). — 2015. — V. 207. N3. — P. 354–388.

[7] Banaru M. B., Banaru G. A. 1-cosymplectic hypersurfaces axiom and six-dimensional planar Hermitian submanifolds of the Octonian // SUT Journal of Mathematics. — 2015. — V. 51. N. 1. — P. 1–9.

[8] Brown R., Gray A. Vector cross products // Comm. Math. Helv. — 1967. — V. 42. — P. 222–236.

[9] Кириченко В. Ф., Ускорев И. В. Инварианты конформного преобразования почти контактных метрических структур // Математические заметки. — 2008. — Т. 84. № 6. — С. 838–850.

[10] Банару М. Б., Банару Г. А. О гиперповерхностях со структурой Кириченко–Ускорева // Сибирские электронные математические известия. — 2020. — Т. 17. — С. 1715–1721.

[11] Банару М.Б. Дискретная характеризация гиперповерхностей косимплектического типа келеровых многообразий // Материалы XIII Международного семинара «Дискретная математика и её приложения» им. академика О.Б. Лупанова. М: МГУ. — 2019. — С. 286–288.

Интерпретации обобщенных чисел Каталана одного класса последовательностей

Бондаренко Леонид Николаевич[1], **Шарапова Марина Леонидовна**[2]

[1] Московский университет им С. Ю. Витте, e-mail: leobond5@mail.ru

[2] Московский государственный университет им. М. В. Ломоносова, e-mail: msharapova@list.ru

Числа Каталана и их обобщения применяются в комбинаторике, статистике, информатике и т. д., а их многочисленные интерпретации рассмотрены в трактате [1]. При $n \in \mathbb{N} = \{1, 2, \ldots\}$ числа $C_0^{(r)}$, $C_n^{(r)} = \sum_{k=0}^{n-1} N_{n,k}$, где $N_{n,k} = \frac{1}{n}\binom{n}{k}\binom{n}{k}$ — числа Нараяны, образуют класс последовательностей, включающий при параметре $r \in \mathbb{N}$ числа Каталана, Шредера и т. п., а свойства этих последовательностей при $r = 1, \ldots, 11$ можно найти в [2].

В статье [3] итерацией кодов Лемера перестановок мультимножеств, введенных И. Гесселем и Р. Стенли в 1978 г. (ГС-перестановок), получена интерпретация чисел Фусса — Каталана с помощью 312-избегающих ГС-перестановок. Рассмотрим при $r,\, n \in \mathbb{N}$ новые интерпретации чисел $C_n^{(r)}$ с использованием ГС-перестановок.

Определение 1. а) *ГС-перестановкой порядка* $r \in \mathbb{N}$ *мультимножества* $\{1^r, \ldots, n^r\}$ *называется слово* $\sigma = \sigma_1 \ldots \sigma_{rn}$, *все буквы* σ_i *которого, стоящие при* $i = 1, \ldots, rn$ *между любыми двумя вхождениями символа* τ, *не меньше этого* τ, *а* $\left|\mathfrak{S}_n^{(r)}\right| = 1 \cdot (r+1) \cdot \ldots \cdot (r(n-1)+1)$ — *мощность множества* $\mathfrak{S}_n^{(r)}$ *всех таких перестановок.*

б) *Преобразованием Лемера* $\mathbf{l}$ *ГС-перестановки* $\sigma \in \mathfrak{S}_n^{(r)}$ *называется слово* $\mathbf{l}\sigma = \mathbf{l}\sigma_1 \ldots \mathbf{l}\sigma_{rn}$, *где* $\mathbf{l}\sigma_i = \#\{j : \sigma_j < \sigma_i,\ 0 \le j \le i-1,\ \sigma_0 = 0\}$.

На базе определения 1 и ключа $\kappa = 1^r \ldots n^r$ длины $|\kappa| = rn$ восстановление $\sigma \in \mathfrak{S}_n^{(r)}$ по ее коду Лемера $\mathbf{l}\sigma$ реализуется алгоритмом: на k-м шаге, где $k = 1, \ldots, rn$, символу σ_{rn-k+1} присваивается буква ключа κ с номером $\mathbf{l}\sigma_{rn-k+1}$, а затем она удаляется из κ, и новый κ имеет длину $|\kappa| = rn - k$.

Для исследования однопараметрического класса последовательностей $\{C_n^{(r)}\}$ при $r,\, n \in \mathbb{N}$ используем его T-модель — семейство последовательностей таблиц $T_1^{(r)}$, $T_2^{(r)}$, ... с элементами из $\mathbb{N}$ [4].

Определение 2. *Эта* T*-модель задается первой таблицей* $T_1^{(r)} = (r+1)$, *отображением* $\theta_r : s \mapsto (r+1)^r \ldots s^r(s+r)$, *где* $s \in \{rn+1\}$, *и рекурсивным соотношением* $T_{n+1}^{(r)} = \theta_r(T_n^{(r)})$, *а преобразование* θ_r *каждый элемент* $s \in T_n^{(r)}$ *переводит в строку длины* s *последующей таблицы* $T_{n+1}^{(r)}$.

Например, $T_1^{(2)} = (3)$, $T_2^{(2)} = (335)$, $T_3^{(2)} = \begin{pmatrix} 335 \\ 335 \\ 33557 \end{pmatrix}$.

Степени образов $\theta_r^i(s)$ при $i \ge 0$ элементов $s \in T_n^{(r)}$ называются в [4] блоками i-го ранга таблиц $T_{n+i}^{(r)}$. Вес $|\theta_r^i(s)|$ блока $\theta_r^i(s)$ при $i \ge 1$ равен числу содержащихся в нем блоков $(i-1)$-го ранга, $|\theta_r^0(s)| = s$, а i-й вес таблицы $T_{n+i}^{(r)}$ равен $\left|T_{n+i}^{(r)}\right|_i = \left|\theta_r^i(T_n^{(r)})\right| = \sum_{s \in T_n^{(r)}} |\theta_r^i(s)|$, $i \ge 0$, $n \in \mathbb{N}$.

Имеем $|\theta_r^i(s)| = s$ и $\left|T_{n+i}^{(r)}\right|_i = \left|T_n^{(r)}\right|_0$, а веса $\left|T_n^{(r)}\right|_0$, $\left|T_n^{(r)}\right|_1$, $\left|T_n^{(r)}\right|_2$ равны соответственно сумме и числу элементов, а также числу строк таблицы $T_n^{(r)}$. В частности, $\left|T_1^{(2)}\right|_1 = \left|T_2^{(2)}\right|_2 = 1$, $\left|T_1^{(2)}\right|_0 = \left|T_2^{(2)}\right|_1 = \left|T_3^{(2)}\right|_2 = 3$, $\left|T_2^{(2)}\right|_0 = \left|T_3^{(2)}\right|_1 = 11$, $\left|T_3^{(2)}\right|_0 = 45$.

Теорема 1. *При $r \in \mathbb{N}$ последовательность $\left\{\left|T_n^{(r)}\right|_1\right\}$ для T-модели определения 2 совпадает с последовательностью $\{C_n^{(r)}\}$, где $n \in \mathbb{N}$, так как* $\sum_{n=0}^{\infty}\left|T_{n+1}^{(r)}\right|_1 u^n = (1-(r+1)u-((1-(r-1)u)^2-4u)^{1/2})/(2ru^2) = \sum_{n=0}^{\infty} C_{n+1}^{(r)} u^n$.

Доказательство. Для многочлена $U_n^{(r)}(t)$, получаемого из производящего многочлена $\sum_{s\in T_n^{(r)}} t^s$ таблицы $T_n^{(r)}$ умножением на $t^{-(r+1)}$ и последующей заменой t на $t^{1/r}$, имеем $U_n^{(r)}(1) = \left|T_n^{(r)}\right|_1$. Также при $n \in \mathbb{N}$ справедлива рекуррентная формула $U_1^{(r)}(t) = 1,\ (1-t)U_{n+1}^{(r)}(t)+t(t+r-1)U_n^{(r)}(t) = rU_n^{(r)}(1)$, так как $\theta_r(t^s) = t^{r(r+1)}(t^{r(s-r)}-1)/(t^r-1)+t^{s+r}$, а производящая функция $F^{(r)}(t,\, u) = \sum_{n=0}^{\infty} U_{n+1}^{(r)}(t)u^n = (1-t+ruF^{(r)}(1,\, u))/(1-t+t(t+r-1)u)$ в правой части имеет неопределенность вида $0/0$ при замене t на корень уравнения $rF^{(r)}(1,\, u) = t(t+r-1)$. Поэтому, исключая t в знаменателе $F^{(r)}(t,\, u)$, получим уравнение $ru^2F^{(r)}(1,\, u)^2-(1-(r+1)u)F^{(r)}(1,\, u)+1 = 0$, решение которого дает требуемый результат. **Теорема 1 доказана.**

Определение 3. *Множество номеров $\mathfrak{N}_n^{(r)}$ элементов таблицы $T_n^{(r)}$ при $n \in \mathbb{N}$ зададим рекурсивно: для $T_1^{(r)} = (r+1)$ номер $\nu = 1$; если $\nu = \nu_1 \dots \nu_n$ номер $s \in T_n^{(r)}$, то номер $s' \in T_{n+1}^{(r)}$ равен $\nu\nu_{n+1}$, где ν_{n+1} — порядковый номер элемента s' в строке $\theta_r(s)$.*

Теорема 2. *Пусть $\mathfrak{N}_1^{(r)} = \{1\}$ и $\nu_1 \dots \nu_n \in \mathfrak{N}_n^{(r)}$. Тогда $\nu_1 \dots \nu_n k \in \mathfrak{N}_{n+1}^{(r)}$, где $k = 1, \dots, \omega$, а $\omega = r+1+\nu_n-\tilde{\nu}_n$ и $\tilde{\nu}_n$ — наименьший положительный вычет числа ν_n по* mod$r$. *Например, $\mathfrak{N}_2^{(r)} = \{11,\, 12,\, \dots,\, 1(r+1)\}$.*

В [4] показано, что множество $\mathfrak{N}_n^{(1)}$ совпадает с множеством кодов Лемера 213-избегающих перестановок степени n.

Для каждого слова $\nu = \nu_1 \dots \nu_n \in \mathfrak{N}_n^{(r)}$ при $r,\, n \in \mathbb{N}$ определим две важные статистики: $\mathrm{ima}(\nu)$ — число различных букв и $\rho(\nu) = \sum_{i=1}^{n}(\nu_i - 1)$ — ранг. Аналогично для каждого слова $\mathbf{l}\sigma = \mathbf{l}\sigma_1 \dots \mathbf{l}\sigma_{rn} \in \mathbf{l}\mathfrak{S}_n^{(r)}$ определяется $\mathrm{ima}(\mathbf{l}\sigma)$ и $\tilde{\rho}(\mathbf{l}\sigma) = r^{-1}\sum_{i=1}^{rn}(\mathbf{l}\sigma_i - 1)$.

При $\widetilde{\mathfrak{S}}_n^{(r)} \subset \mathfrak{S}_n^{(r)}$ назовем множества $\mathfrak{N}_n^{(r)}$ и $\mathbf{l}\widetilde{\mathfrak{S}}_n^{(r)}$ изоморфными, если для соответствующих друг другу элементов $\nu \in \mathfrak{N}_n^{(r)}$ и $\mathbf{l}\sigma \in \mathbf{l}\widetilde{\mathfrak{S}}_n^{(r)}$ имеем $\mathrm{ima}(\nu) = \mathrm{ima}(\mathbf{l}\sigma)$ и $\rho(\nu) = \tilde{\rho}(\mathbf{l}\sigma)$. Так как перед буквами слова $\mathbf{l}\sigma = \mathbf{l}\sigma_1 \dots \mathbf{l}\sigma_{rn} \in \mathbf{l}\widetilde{\mathfrak{S}}_n^{(r)}$ имеется $k = 1, \dots, rn$ мест и еще одно место после последней буквы, то в обозначениях теоремы 2 справедливо утверждение.

Теорема 3. *Пусть $\mathfrak{N}_1^{(r)} = \{1\}$ и $\mathbf{l}\widetilde{\mathfrak{S}}_1^{(r)} = \{1^r\}$. При построенном изоморфизме $\mathfrak{N}_n^{(r)}$ и $\mathbf{l}\widetilde{\mathfrak{S}}_n^{(r)}$ для $\nu = \nu_1 \dots \nu_n \in \mathfrak{N}_n^{(r)}$ имеем $\nu_1 \dots \nu_n k \in \mathfrak{N}_{n+1}^{(r)}$, где $k = 1, \dots, \omega$, и соответственно находим ω требуемых слов в $\mathbf{l}\widetilde{\mathfrak{S}}_{n+1}^{(r)}$*

вставкой подслова k^r *на места* $k = 1, \ldots, \omega$ *в слово* $\mathbf{l}\sigma = \mathbf{l}\sigma_1 \ldots \mathbf{l}\sigma_{rn} \in \mathbf{l}\widetilde{\mathfrak{S}}_n^{(r)}$. *Тогда множества* $\mathfrak{N}_{n+1}^{(r)}$ *и* $\mathbf{l}\widetilde{\mathfrak{S}}_{n+1}^{(r)}$ *также изоморфны.*

Множество $\widetilde{\mathfrak{S}}_n^{(r)}$ однозначно восстанавливается по $\mathbf{l}\widetilde{\mathfrak{S}}_n^{(r)}$ с выполнением соотношения $\left|\widetilde{\mathfrak{S}}_n^{(r)}\right| = S_n^{(r)} < \left|\mathfrak{S}_n^{(r)}\right|$. Поэтому найдены интерпретации чисел $S_n^{(r)}$ с применением ГС-перестановок. Отметим, что $\mathfrak{N}_n^{(1)} \neq \mathbf{l}\widetilde{\mathfrak{S}}_n^{(1)}$ при $n \geq 4$.

Приведенные результаты позволяют получить и ряд других свойств обобщенных чисел Каталана $S_n^{(r)}$, в частности, определить их q-аналоги, построить для них непрерывные дроби и т. д.

Список литературы

[1] Stanley R. P. Catalan numbers. — New York: Cambridge university press, 2015. — 215 p.

[2] Sloane N. J. A. The on-line encyclopedia of integer sequences. — 2021. — http://oeis.org.

[3] Бондаренко Л. Н., Шарапова М. Л. Обобщенные 312-избегающие перестановки и преобразование Лемера // Прикладная дискретная математика. Приложение. — 2017. — N.º10. — С. 7–9.

[4] Бондаренко Л. Н., Шарапова М. Л. Применение T-моделей и T-диаграмм в комбинаторике // Материалы XXIII Международного семинара «Дискретная математика и ее приложения» им. акад. О. Б. Лупанова (Москва, МГУ, 17 — 22 июня 2019 г.). — М.: Изд-во механико-математического факультета МГУ, 2019. — С. 196–199.

О сложности точной расшифровки функций фиксированного веса запросами на сравнение

Быстрыгова Анастасия Викторовна

МГУ имени М. В. Ломоносова, e-mail: anastasiya.bistrigova@yandex.com

Под точной расшифровкой функций будем понимать следующую игру двух участников: учителя и ученика. Перед игрой учитель и ученик фиксируют множество функций и тип запросов. В начале игры учитель в тайне от ученика выбирает одну функцию из зафиксированного множества, затем ученик последовательно отправляет учителю запросы фиксированного типа. На каждый запрос учитель посылает ученику ответ. Ученик анализирует ответы на предыдущие запросы и в итоге либо понимает, какая функция зафиксирована учителем, либо задает следующий запрос, который возможно позволит раскрыть больше информации про загаданную

учителем функцию. Игра завершается, когда ученик в точности понял, какую функцию выбрал учитель в начале игры.

В данной работе будем рассматривать точную расшифровку функций фиксированного веса запросами на сравнение.

Под *функциями фиксированного веса* $F(n,k)$, где $k \in [1, 2^n)$, будем понимать множество булевых функций арности n, у которых число единиц в векторе значений равно k.

Запросы на сравнение были введены Э.Э.Гасановым в работе [1]. *Запрос на сравнение* к функции f — это упорядоченная пара наборов (a, b), а ответ на запрос на сравнение — это знак разности значений функции f на этих наборах, т. е. $sgn(f(a) - f(b))$.

Основным вопросом, представляющим интерес в области точной расшифровки функций, является *сложность расшифровки*. Под сложностью расшифровки обычно понимают наибольшее количество запросов, которое вынужден задать ученик для завершения игры при использовании наилучшей своей стратегии. Под наилучшей стратегией понимается тот алгоритм выбора запросов, который позволит уменьшить их количество.

Если говорить формально, то сложность $\varphi(n,k)$ расшифровки функций фиксированного веса запросами на сравнение определяется следующим образом: $\varphi(n,k) = \min_{A \in \Omega(n,k)} \max_{f \in F(n,k)} Q(A,f)$, где $\Omega(n,k)$ — множество стратегий игры ученика при расшифровке $F(n,k)$, $Q(A,f)$ — количество запросов, которое отправит ученик для завершения игры, в которой учителем выбрана функция f, а учеником при выборе запросов используется стратегия A.

Если a — вещественное число, тогда через $]a[$ будем обозначать наименьшее целое, не меньшее a, через $[a]$ — наибольшее целое, не большее a, через $a \bmod b$ — остаток от деления a на b. Будем писать $A(n) \lesssim B(n)$, если $\lim_{n\to\infty} \frac{A(n)}{B(n)} \leq 1$.

Теорема 1. *Для любого $k = k(n)$, такого, что $k \geq 2, k = o(2^n)$, сложность расшифровки класса $F(n,k)$ запросами на сравнение при $n \to \infty$ удовлетворяет соотношению:* $2/3 \cdot 2^n \lesssim \varphi(n,k) \lesssim k/(k+1) \cdot 2^n$.

Помимо этого, верны следующие оценки: $\varphi(n,1) = 2^{n-1}$ *при* $n \geq 1$, $\varphi(n,2) = [2^{n+1}/3]$ *при* $n \geq 2$, $\varphi(n,3) = 2^n - \Big]3/2 \cdot [2^n/5]\Big[- \big[(2^n \mod 5)/2\big]$ *при* $n \geq 7$.

Верхняя оценка асимптотического неравенства доказана в лемме 1 работы [2], а нижняя — в теореме 6 работы [3]. Точные значения сложности расшифровки для $k \in \{1,2,3\}$ доказаны в теоремах 4, 5, 6 работы [2].

Список литературы

[1] Гасанов Э. Э. Расшифровка линейных функций ранжирования // Материалы XI Международного семинара «Дискретная математика и ее приложения» (Москва, 18-23 июня 2012 г.). — 2012. — С. 332–334.

[2] Быстрыгова А. В. Расшифровка булевых функций фиксированного веса // Интеллектуальные системы. Теория и приложения. — 2020. — Т. 24, № 3. — С. 63–96.

[3] Быстрыгова А. В. Расшифровка булевых функций ограниченного веса // Вестник Московского государственного университета. Серия 1. Математика, механика. — В печати.

Явная формула для числа помеченных тетрациклических графов без мостов

Воблый Виталий Антониевич

Всероссийский институт научно-технической информации РАН, e-mail: vitvobl@yandex.ru

Рассматриваются неориентированные простые связные графы, k-цикли-ческий граф – это связный граф с n вершинами и $n+k-1$ ребрами.

В [1] найдены явные формулы для чисел помеченных бициклических и трициклических графов без мостов. Хенлон и Робинсон [2] получили для производящей функции помеченных графов без мостов функционально-дифференциальное уравнение, а также нелинейное уравнение с частными производными. Однако из этих уравнений не получены вычислительные формулы. В [3] асимптотически перечислены помеченные связные k-циклические графы без мостов.

В докладе получена явная формула для числа помеченных связных тетрациклических графов без мостов с заданным числом вершин.

Обозначим через $\bar{S}(n,k)$ – число помеченных связных n-вершинных k-циклических графов без мостов.

Теорема 1. *Для числа* $\bar{S}(n,4)$ *при* $n \geq 6$ *верна формула*

$$\bar{S}(n,4) = \frac{n!}{1451520}(11n^8 + 165n^7 - 1029n^6 - 8841n^5 + 44856n^4 + 50694n^3 - 152828n^2 - 660048n - 282240\ . \quad (1)$$

Доказательство. Пусть $B(n,k)$ – число помеченных k-циклических блоков с n вершинами, а $B_k(z) = \sum_{n=0}^{\infty} B(n,k)\frac{z^n}{n!}$ – соответствующая экспоненциальная производящая функция. Известна формула [4]

$$\bar{S}(n,k) = \frac{(n-1)!}{nk!}[z^{-1}]Y_k(n1!B_1'(z), n2!B_2'(z)..., nk!B_k'(z))z^{-n}\ ,$$

где $[z^{-1}]$ – оператор формального вычета [5, с. 41], а $Y_k(x_1, ..., x_k)$ – многочлены разбиений с известным выражением [6, с. 173]

$$Y_k(x_1, ..., x_k) = \sum_{\pi(k)} \frac{k!}{m_1!...m_k!} \Big(\frac{x_1}{1!}\Big)^{m_1} ... \Big(\frac{x_k}{k!}\Big)^{m_k} ,$$

где суммирование проводится по всем разбиениям $\pi(k)$ числа k, то есть по всем неотрицательным решениям $(m_1, m_2, \cdots, m_k)$ уравнения

$$m_1 + 2m_2 + ... + km_k = k, m_i \geq 0,\ i = 1, ..., k .$$

Так как [6, с. 246] $Y_4(x_1, x_2, x_3, x_4) = x_1^4 + 6x_1^2x_2 + 3x_2^2 + 4x_1x_3 + x_4$ и $x_i = ni!B_i'(z)$, имеем

$$\bar{S}(n, 4) = \frac{(n-1)!}{24n}[z^{-1}]\Big(n^4(B_1'(z))^4 + 12n^3(B_1'(z))^2B_2'(z) + 12n^2(B_2'(z))^2 +$$

$$24n^2B_1'(z)B_3'(z) + 24nB_4'(z)\Big)z^{-n} . \qquad (2)$$

Райт [7] доказал формулы

$$B(n, 2) = \frac{n!(n-3)(n+2)}{24} , B(n, 3) = \frac{n!(n-3)}{1152}(n^4+4n^3-15n^2-46n-40).$$

$$B(n, 4) = \frac{n!(n-3)(n-4)}{1451520}(11n^6+143n^5+281n^4-1891n^3-8608n^2-12616n-7560).$$

Так как унициклический блок – это простой цикл, то имеем

$$B(n, 1) = (n-1)!/2 , B_1(z) = \sum_{n=3}^{\infty} \frac{1}{2}(n-1)!\frac{z^n}{n!} , B_1'(z) = \frac{z^2}{2(1-z)} ,$$

Таким образом, получим

$$B_2(z) = \frac{z^4(3-2z)}{12(1-z)^3} , B_2'(z) = \frac{12z^3 - 13z^4 + 4z^5}{12(1-z)^4} .$$

$$B_3(z) = \frac{2z^4 + 28z^5 - 47z^6 + 28z^7 - 6z^8}{48(1-z)^6},$$

$$B_3'(z) = \frac{4z^3 + 72z^4 - 127z^5 + 98z^6 - 38z^7 + 6z^8}{24(1-z)^7},$$

Суммирование рядов для $B_2(z)$ и $B_3(z)$, а также дифференцирование было выполнено с помощью пакета программ Maple.

Интегрируя по частям последнее слагаемое в (2), найдем

$$\bar{S}(n,4) = C_1(n) + C_2(n) + C_3(n) + C_4(n) + B(n,4)\ . \qquad (3)$$

С помощью разложения [4, с. 141] $(1-z)^{-p-1} = \sum_{i=0}^{\infty} \binom{i+p}{p} z^i$ получим

$$C_1(n) = \frac{(n-1)!}{24}[z^{-1}]n^3(B_1'(z))^4 z^{-n} = \frac{n!n^2}{24}[z^{-1}]\frac{z^8}{16(1-z)^4} =$$

$$\frac{n!n^2}{384}[z^{-1}]\sum_{i=0}^{\infty}\binom{i+2}{3}z^{i+8-n} = \frac{n!n^2}{384}\binom{n-6}{3}\ .$$

Аналогично имеем

$$C_2(n) = \frac{n!n}{2}[z^{-1}](B_1'(z))^2 B_2'(z)z^{-n} = \frac{n!n}{96}(12\binom{n-3}{5} - 13\binom{n-4}{5} + 4\binom{n-5}{5})\ ,$$

$$C_3(n) = \frac{(n-1)!}{24}[z^{-1}]12n(B_2'(z))^2 z^{-n} = \frac{n!}{288}(16\binom{n-4}{7} - 104\binom{n-3}{7} +$$

$$265\binom{n-2}{7} - 312\binom{n-1}{7} + 144\binom{n}{7}\ .$$

$$C_4(n) = n![z^{-1}]B_1'(z)B_3'(z)z^{-n} = \frac{n!}{48}(6\binom{n-4}{7} - 38\binom{n-3}{7} + 98\binom{n-2}{7} -$$

$$127\binom{n-1}{7} + 72\binom{n}{7} + 4\binom{n+1}{7}\ .$$

Подставляя в (3) выражения для $C_1(n), C_2(n), C_3(n), C_4(n), B(n,4)$ и представляя биномиальные коэффициенты в виде многочленов от n, после приведения подобных членов найдем для $\bar{S}(n,4)$ формулу (1).

Теорема 1 доказана.

Список литературы

[1] Воблый В. А. Перечисление помеченных бициклических и трициклических графов без мостов // Математ. заметки, — 2012. Т. 91, вып. 2, С. 308-311.

[2] Hanlon P., Robinson R. W., Counting bridgeless graphs // J. Combin. Theory, ser. B, 33(1982), С. 276-305.

[3] Воблый В. А. Об асимптотическом перечислении помеченных связных k-циклических графов без мостов // Математ. заметки, — 2020. Т. 107, вып. 2. — С. 304-306.

[4] Воблый В. А., Об одном подходе к перечислению помеченных связных графов: обзор результатов // Итоги науки и техн. Сер. Совр. мат. прилож. Темат. обзоры. — 2020. — Т. 188. — С. 106-118.

[5] Гульден Я., Джексон Д. Перечислительная комбинаторика. — М. : Наука, 1990. — 504 с.

[6] Риордан Дж. Комбинаторные тождества. — М. :Наука, 1982. — 256 с.

[7] Wright E. M. The number of connected sparsely edged graphs.II. Smooth graphs and blocks // J. Graph Theory. 2(1978), No. 4. C. 299–305.

Тестирование бесповторных функций в расширенном обобщенной медианой элементарном базисе

Вороненко Андрей Анатольевич[1], **Кафтан Дарья Владимировна**[2]

[1] Московский Государственный Университет имени М.В. Ломоносова, e-mail: dm6@cs.msu.ru
[2] Московский Государственный Университет имени М.В. Ломоносова, e-mail: blond.programmist@gmail.com

Множество n-мерных булевых наборов T называется *тестом относительно бесповторной альтернативы* [1] в базисе B для функции $f(x_1, \ldots, x_n)$, существенно зависящей от всех переменных, если для любой бесповторной в базисе B функции $h(x_1, \ldots, x_n)$ существует набор $(\alpha_1, \ldots, \alpha_n) \in T$, такой, что $f(\alpha_1, \ldots, \alpha_n) \neq h(\alpha_1, \ldots, \alpha_n)$.

Функция $f_m^s = x_1(x_2 \vee \ldots \vee x_s) \vee x_2 \& \ldots \& x_s$ являтся слабоповторной [2] и при $s = 3$ совпадает с медианой. Обозначим базисы $B_m^s = \{\&, \vee, \neg, f_m^s\}$ и $B_m^{s+} = \{\&, \vee, f_m^s\}$. Обозначим $T_m^s(n)$ функцию Шеннона [3] для длины теста относительно бесповторной альтернативы в базисе B_m^s, а $T_m^{s+}(n)$ — функцию Шеннона для длины проверяющего теста в B_m^{s+}.

Лемма 1. *Справедливо равенство* $T_m^s(n) \leqslant T_m^{s+}(n) + 2n$.

Доказательство. Задача тестирования относительно бесповторной альтернативы функции $f(x_1, \ldots, x_n)$ в базисе B_m^s сводится к проверяющему тестированию в базисе B_m^{s+} функции, получаемой из f заменой антимонотонных переменных на их отрицания. При этом требуется для каждой переменной добавить пару соседних по ней наборов, на которых функция принимает различные значения. **Лемма 1 доказана.**

Каноническим деревом бесповторной в базисе B_m^{s+} функции $f(x_1, \ldots, x_n)$, назовем помеченное корневое дерево, построенное согласно следующим правилам.

1. Листья дерева помечены переменными $x_1, \ldots, x_n$. Разные листья помечены разными переменными.
2. Внутренние вершины дерева соответствуют подформулам бесповторной формулы и помечены функциями из множества $\{\vee, \&, f_m^s\}$. Вершины, помеченные одинаковыми функциями $\&$ или $\vee$, не смежны.
3. Над вершиной, помеченной f_m^s, находятся s инцидентных ребер с выделенным первым ребром при $s \geqslant 4$.
4. Над вершиной, помеченной $\vee$ или $\&$, расположено не менее двух смежных вершин.

Каноническое дерево D реализует функцию f по бесповторной формуле от корня к листьям стандартным образом. При этом арность $\&$ и $\vee$ в формуле совпадает с числом ребер над помеченной ими вершиной. Перестановка переменных является эквивалентной для функции f_m^s при $s \geqslant 4$ тогда и только тогда, когда она сохраняет первую переменную, поэтому каноническое дерево для бесповторной функции в базисе B_m^{s+} является единственным.

Лемма 2. *Пусть бесповторная в базисе B_m^{s+} функция $h(z_1, z_2, z_{k+2}, \ldots, z_n)$ существенно зависит от $n - k + 1 \geqslant 2$ переменных ($k \geqslant 2$), и лист ее канонического дерева, который помечен переменной z_1, смежен с вершиной u, помеченной конъюнкцией или функцией f_m^s, а лист, помеченный z_2, лежит в каком-то другом поддереве над u. Пусть $f(x_1, \ldots, x_n) = h(x_1 \vee \ldots \vee x_k, x_{k+1}, \ldots, x_n)$. Пусть $h(z_1, z_2, \beta_{k+2}, \ldots, \beta_n) = z_1 \& z_2$. Тогда тест T', содержащий наборы теста T функции $f|_{x_k=0}$ с добавленными компонентами $x_k = 0$ и два набора $(1, \ldots, 1, 0, \beta_{k+2}, \ldots, \beta_n)$ и $(0, \ldots, 0, 1, 1, \beta_{k+2}, \ldots, \beta_n)$, является проверяющим тестом для функции f на множестве всех бесповторных в B функций, существенно зависящих от n переменных.*

Лемма 3. *Пусть бесповторная в базисе B_m^{s+} функция $h(z_1, z_{s+1}, \ldots, z_n)$ существенно зависит от $n - s + 1$ переменных, где $n > s$, и лист ее канонического дерева, который помечен переменной z_1, входит в вершину u, помеченную дизъюнкцией или функцией f_m^s.*

Пусть функция f зависит от переменных $x_1, \ldots, x_n$, и для f и h выполнены следующие равенства:

$$f(x_1, \ldots, x_n) = h(f_m^s(x_1, \ldots, x_s), x_{s+1}, \ldots, x_n),$$

$$h(z_1, \beta_{s+1}, \ldots, \beta_n) = z_1.$$

Тогда тест T', содержащий наборы теста T функции $f|_{x_1=0}$ с добавлением компоненты $x_1 = 0$, и наборов $(1, 0, \ldots, 0, \beta_{s+1}, \ldots, \beta_n)$,

$(1, 1, 0, \ldots, 0, \beta_{s+1}, \ldots, \beta_n)$, $(1, 0, 1, \ldots, 0, \beta_{s+1}, \ldots, \beta_n)$, $\ldots$,
$(1, 0, \ldots, 0, 1, \beta_{s+1}, \ldots, \beta_n)$, *является проверяющим тестом для функции* f *на множестве всех бесповторных в* B_m^{s+} *функций, существенно зависящих от* n *переменных.*

Теорема 1. *Справедливо неравенство*:

$$T_m^s(n) \leqslant 5n.$$

Доказательство. При построении теста для функции, бесповторной в B_m^{s+}, применяя леммы 2 или 3, мы уменьшаем число аргументов на единицу. При этом при применении леммы 2 в тест добавляются два набора, а при применении леммы 3 — s наборов.

Однако общее число вершин степени исхода s в ориентированном дереве с n листьями не превосходит $\left[\frac{n-1}{s-1}\right]$. Поэтому при переходе от n аргументов к одному добавляется не более

$$s\left[\frac{n-1}{s-1}\right] + 2(n-1-\left[\frac{n-1}{s-1}\right]) = 2(n-2) + (s-2)\left[\frac{n-1}{s-1}\right] \leqslant 3n$$

наборов.

Так как $T_m^{s+}(1) = 0$, то $T_m^{s+} \leqslant 3n$. По лемме 1 справедливо $T_m^s(n) \leqslant$
$\leqslant 2n + 3n = 5n$.

Теорема 1 доказана.

Список литературы

[1] Вороненко А А. О проверяющих тестах для бесповторных функций // Математические вопросы кибернетики. — Вып. 11. — М.: Физматлит, — 2002. — с. 165–176.

[2] Стеценко В А. О предплохих базисах в P_2 // Математические вопросы кибернетики. — Вып. 4. — М.: Физматлит, — 1992. — с. 139–177.

[3] Shannon C. A symbolic analysis of relay and switching circuits // Trans. AIEE. — 1938. — **57**.— p. 713–723.

Верхняя оценка переключательной мощности реализации периодических последовательностей плоскими автоматными схемами

Воротников Алексей Сергеевич

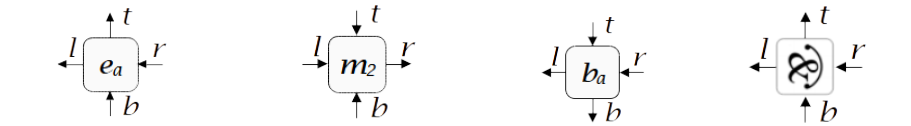
Московский государственный университет имени М. В. Ломоносова, e-mail: vorotnikov.lexa@yandex.ru

Впервые понятие схемы из клеточных элементов, далее так же называемой плоскими схемами, было введено в работе Кравцова С.С. [1]. В работах [2, 3] Г. В. Калачев показал, что порядок потенциала и переключательной мощности плоской схемы, реализующей булеву функцию от n переменных, составляет $2^{n/2}$.

Вводимое определение несколько расширяет определение *плоской схемы*, введённое в работе [2]. Определения *сети из клеточных элементов*, *графа корректной сети из клеточных элементов*, *входов (выходов)*, *узлов*, *подсхемы* так же присутствуют в работе [2].

Клеточным элементом будем называть автомат с не более чем двумя состояниями, у которого в сумме не более четырёх входов и выходов, причём каждому его входу и каждому выходу сопоставлена некоторая метка из множества $\{l, r, t, b\}$, причём метки не повторяются. Клеточный элемент будем изображать в виде единичного квадрата на плоскости. Метки, присвоенные входам (выходам) автомата будем называть входами (выходами) элемента.

Описывать элемент с одним состоянием будем уравнениями, которые задают его оператор, заменяя все переменные в них на сопоставленные им метки (l, r, t или b). Тогда в левой части каждого уравнения будет стоять выходная метка, а в правую будут входить только входные метки. На рисунке 1 приведены примеры клеточных элементов.



Примеры логических элементов.

Всюду далее значок := будет обозначать «по определению равно».

Далее везде используется базис $\mathcal{B}$, состоящий из всех элементов с одним состоянием и множества элементов с двумя состояниями $\{(E, E, E, \phi, \psi_1),$

$(E, E, E^2, \phi, \psi_2)$, $(E, E, E^3, \phi, \psi_3)\}$, где $E = \{0, 1\}$,

$$\begin{cases} \varphi(1) = 0, \\ \varphi(t+1) = a(t). \end{cases}$$

$$\psi_1(t) = q(t), \quad \psi_2(t) = (q(t), q(t)), \quad \psi_3(t) = (q(t), q(t), q(t)),$$

где $a(t)$ — входной сигнал в момент времени t, $q(t)$ — состояние автомата в момент времени t. Будем называть такие элементы *задержками*.

Плоской автоматной схемой на множестве $M \subset \mathbb{Z}^2$ над базисом $\mathcal{E}' \subseteq \mathcal{E}$ будем называть корректную сеть из клеточных элементов, в графе которой все ориентированные циклы содержат хотя бы одну задержку. Множество M будем называть *носителем* схемы K.

Каждой плоской схеме K можно сопоставить структурный автомат $Circ(K)$ следующим образом:

1. каждой функции $f_{s,i}$, которую реализует i-й выход элемента s клеточной схемы, сопоставим функциональный элемент $e_{s,i}$, реализующий $f_{s,i}$; если i-й и j-й выходы являются выходами одной и той же функции, то им будет соответствовать один и тот же функциональный элемент;
2. если i-й выход s_1 подключен к j-му входу s_2, то соединим выход элемента $e_{s_1,i}$ с j-ми входами элементов $e_{s_2,k}$ для всех k, для которых $f_{s_2,k}$ зависит от j-го аргумента;
3. удалим из схемы все тождественные функции, подсоединив их вход ко всем их выходам;
4. аналогично поступаем с задержками.

Сопоставление корректно, так как правила сопоставления вкладываются в правила индуктивного построения структурных автоматов [4]. Правило обратной связи, требующее зависимость со сдвигом от замыкаемой переменной, так же выполнено в силу наличия задержки в каждом ориентированном цикле. Поскольку только ориентированные циклы могут порождать обратную связь, всё верно.

Будем говорить, что схема K реализует автомат A_K, если схема из автоматных элементов $Circ(K)$ реализует A_K.

Далее рассматриваем только плоские автоматные схемы без входов с единственным выходом.

Меры мощности схем

Рассмотрим плоскую автоматную схему K, реализующую периодическую последовательность длины $l \in \mathbb{N}$. Последовательность, реализуемую схемой K обозначим α_K. Для каждой такой схемы K зафиксируем некоторую

нумерацию её узлов. На i-м узле реализуется некоторая автоматная функция g_i. Состоянием схемы K на такте t назовём вектор $s_K(t) := (g_1(t), ..., g_h(t))$.

Величину $c_K(t) := |s_K(t) \oplus s_K(t+1)|$ назовём *затратой энергии на переключение схемы* с такта t на $t+1$. Длина вектора понимается как сумма целых чисел его компонент: $a = (a_1, \cdots, a_n) \in \{0,1\}^n, |a| = \sum_{i=1}^n a_i$.

Схема K *функционирует циклически* с периодом l, если последовательность на её выходе имеет период l.

Переключательной мощностью схемы K, функционирующей циклически с периодом l, назовём $W(K) = \dfrac{1}{l}\sum\limits_{t=0}^{l} c_K(t)$.

Переключательной мощностью последовательности α назовём величину $W(\alpha) = \min\limits_{\alpha_K=\alpha} W(K)$.

Функцией Шеннона для переключательной мощности последовательностей из класса $\{0,1\}^l$ назовём $W(l) = \max\limits_{\alpha\in\{0,1\}^l} W(\alpha)$.

Теорема 1.

$$W(2^n) \preceq 12\frac{2^{n/2}}{n},\ \textit{при}\ n \to \infty$$

Причём достигнуть такой переключательной мощности можно используя асимптотически не более $\frac{3}{2}n$ *задержек при* $n \to \infty$.

Автор выражает благодарность д.ф.-м.н. профессору Э. Э. Гасанову за постановку задачи и внимание к работе, а также к.ф.-м.н., м.н.с. Г. В. Калачёву и А. А. Ефимову за ценные замечания и предложения по тексту работы.

Список литературы

[1] Кравцов С. С. О реализации функций алгебры логики в одном классе схем из функциональных и коммутационных элементов. // Проблемы кибернетики. — 1967. — Т. 19. — С. 285–293.

[2] Калачев Г. В. Порядок мощности плоских схем, реализующих булевы функции. // Дискретная математика. — 2014. — Т. 26, вып. 1. — С. 49–74.

[3] Калачев Г. В. Нижние оценки мощности плоских схем, реализующих частичные булевы операторы. // Интеллектуальные системы. Теория и приложения. — 2014. — Т. 18, № 2. — С. 279-322.

[4] Кудрявцев В. Б., Алёшин С. В., Подколзин А. С. Введение в теорию авоматов: Монография. — М. : Издательство Московского университета, 2019. — 436 с.

Functional specification of quasigroup operations

Галатенко Алексей Владимирович[1], Носов Валентин Александрович[2], Панкратьев Антон Евгеньевич[3]

[1] МГУ имени М.В.Ломоносова, e-mail: agalat@msu.ru
[2] МГУ имени М.В.Ломоносова, e-mail: vnosov40@mail.ru
[3] МГУ имени М.В.Ломоносова, e-mail: apankrat@intsys.msu.ru

Finite quasigroups are becoming a popular platform for implementation of various cryptographic functions (see e.g. [1,2]). In case of public key algorithms the order of quasigroups may be high, so specification of quasigroup operations by the means of Cayley tables is impossible due to memory constraints. A possible solution is switching from tables to functional specification. We present a construction for functional specification of large parametric families of d-quasigroups for arbitrary $d \geq 2$.

Basic definitions

Definition 1. *A finite quasigroup is a pair (Q, f), where is Q a finite set and $f\colon Q \times Q \to Q$ is invertible in both variables.*

Throughout this paper all structures are assumed to be finite, so for the sake of brevity the word "finite" will be omitted.

Definition 1 can be directly extended to the case of operations of greater arity.

Definition 2. *A finite d-quasigroup is a pair (Q, f), where Q is a finite set and $f\colon Q^d \to Q$ is invertible in any variable.*

Assume that $|Q| = k^n$ for some $k, n \in \mathbb{N}$, $k \geq 2$. In this case without loss of generality the elements of Q can be treated as n-tuples from E_k^n, and the operation f can be viewed as a dn-ary vector function $(f_1, \ldots, f_n)$.

Definition 3. *A family $(g_1, \ldots, g_n)$ of n-ary functions of k-valued logic is said to be proper if for any pair α, β of distinct input n-tuples there exists an index i, $1 \leq i \leq n$, such that $\alpha_i \neq \beta_i$, but $g_i(\alpha) = g_i(\beta)$.*

Definition 3 obviously implies that if a family $(g_1, \ldots, g_n)$ is proper, then for any i, $1 \leq i \leq n$, the variable x_i is dummy for the function g_i.

An obvious example of a proper family is a family consisting of constant functions. Other examples are triangular families (i.e. such that the variables $x_1, \ldots, x_i$ are dummy for g_i, $i = 1, \ldots, n$) and orthogonal families (i.e. such that x_i is dummy for g_i, $i = 1, \ldots, n$, and $g_i \cdot g_j \equiv 0$ for all $1 \leq i < j \leq n$).

Quasigroup operation specification

Proper families of functions can be used to specify large families of quasigroups.

Theorem 1 ([3]). *Assume that $h_1, \ldots, h_n$ are 3-quasigroup operations on the set E_k,*

$$\begin{array}{rcl} f_1(x_1, \ldots, x_n, y_1, \ldots, y_n) & = & h_1(x_1, y_1, g_1(\pi_1(x_1, y_1), \ldots, \pi_n(x_n, y_n))) \\ f_2(x_1, \ldots, x_n, y_1, \ldots, y_n) & = & h_2(x_2, y_2, g_2(\pi_1(x_1, y_1), \ldots, \pi_n(x_n, y_n))) \\ & \vdots & \\ f_n(x_1, \ldots, x_n, y_1, \ldots, y_n) & = & h_n(x_n, y_n, g_n(\pi_1(x_1, y_1), \ldots, \pi_n(x_n, y_n))) \end{array} \tag{1}$$

Then the relations (1) specify a quasigroup operation for any choice of the functions $\pi_1, \ldots, \pi_n$ if and only if the family $(g_1, \ldots, g_n)$ is proper.

In the original construction [4] the authors considered h_i of the form $h_i(x, y, z) = x + y + z$ for some Abelian group $(E_k, +)$. In [3] it was shown that switching from Abelian groups to 3-quasigroups allows one to improve subquasigroup structure and thus improve cryptographic strength. Hence the generalization is meaningful; however in order to utilize full power of the construction one needs to be able to generate 3-quasigroup operations of a smaller order. Note that 3-quasigroup operations are starting to attract attention in research cryptographic projects. These facts motivated us to generalize Theorem 1 to the case of d-quasigroup operations, $d \geq 2$. The generalization for the case of Abelian groups was obtained by Plaksina in [5].

Theorem 2. *Assume that $d \in \mathbb{N}$, $d \geq 2$, $h_1, \ldots, h_n$ are $(d+1)$-quasigroup operations on the set E_k. Then the relations*

$$\begin{aligned} f_i\left(x_1^1, \ldots, x_n^1, \ldots, x_1^d, \ldots, x_n^d\right) = \\ = h_i\left(x_i^1, \ldots, x_i^d, g_i\left(\pi_i\left(x_1^1, \ldots, x_1^d\right), \ldots, \pi_n\left(x_n^1, \ldots, x_n^d\right)\right)\right), \end{aligned}$$

$i = 1, \ldots, n$, specify a d-quasigroup operation for all functions $\pi_1, \ldots, \pi_n$ if and only if the family $(g_1, \ldots, g_n)$ is proper.

Permutation construction

N. A. Piven proposed a memory-efficient method [6] for increasing the number of quasigroups generated by proper families of functions and improving some cryptographically important properties (e. g. simplicity). Piven's idea consists in applying arbitrary permutations to the indices of f_i, x_j and y_l in the representation (1). In [6] it is shown that the benefits of the construction start to emerge already for the case $k = n = 2$, e. g. the number of quasigroups generated grows 4 times. Memory occupied by 3 permutations is $3 \cdot n \cdot \lceil \log_2 n \rceil$, which is much smaller than the size of the Cayley table. The construction can be easily extended to the case of functional specification of d-quasigroups.

Theorem 3. *Assume that* $d \in \mathbb{N}$, $d \geq 2$, $h_1, \ldots, h_n$ *are* $(d+1)$*-quasigroup operations on the set* E_k, $\sigma_1, \ldots, \sigma_d, \tilde{\sigma} \in S_n$. *Then the relations*

$$f_{\tilde{\sigma}(i)}\left(x_1^1, \ldots, x_n^1, \ldots, x_1^d, \ldots, x_n^d\right) =$$
$$= h_i\left(x_{\sigma_1(i)}^1, \ldots, x_{\sigma_d(i)}^d, g_i\left(\pi_1\left(x_{\sigma_1(1)}^1, \ldots, x_{\sigma_d(1)}^d\right), \ldots, \pi_n\left(x_{\sigma_1(n)}^1, \ldots, x_{\sigma_d(n)}^d\right)\right)\right),$$

$i = 1, \ldots, n$, *specify a* d*-quasigroup operation for all functions* $\pi_1, \ldots, \pi_n$ *if and only if the family* $(g_1, \ldots, g_n)$ *is proper.*

Storing $d+1$ permutations requires $(d+1) \cdot n \cdot \lceil \log_2 n \rceil$ bits, which is negligible in comparison with the size of the Cayley table consisting of $(k^n)^d$ elements.

References

[1] Глухов М. М. О применениях квазигрупп в криптографии // Прикладная дискретная математика. — 2008. — №2. — С. 28–32.

[2] Shcherbacov V. A. Quasigroups in cryptology // Computer Science Journal of Moldova. — 2009. — Vol. 17, Iss. 2(50). — P. 193–228.

[3] Galatenko A. V., Nosov V. A., Pankratiev A. E. Latin squares over quasigroups // Lobachevskii Journal of Mathematics. — 2020. — Vol. 41, Iss. 2. — P. 194–203.

[4] Nosov V. A., Pankratiev A. E. Latin squares over Abelian groups // Journal of Mathematical Sciences. — 2008. — Vol. 149, Iss. 3. — P. 1230–1234.

[5] Плаксина И. А. Построение параметрического семейства многомерных латинских квадратов // Интеллектуальные системы. Теория и приложения. — 2014. — Т. 18, № 2. — С. 323–329.

[6] Пивень Н. А. Исследование квазигрупп, получаемых с помощью правильных семейств булевых функций порядка 2 // Интеллектуальные системы. Теория и приложения. — 2018. — Т. 22, № 1. — С. 21–35.

Клеточные автоматы с локаторами — новый класс управляющих систем

Гасанов Эльяр Эльдарович[1], Васильев Денис Игоревич[2]

[1] МГУ имени М. В. Ломоносова, e-mail: el_gasanov@mail.ru

[2] МГУ имени М. В. Ломоносова, e-mail: denistryhard@yandex.ru

Понятие клеточных автоматов с локаторами было введено Э.Э.Гасановым в работе [1]. В работе [2] Г.В.Калачев высказал уточнения к определению этого понятия. Ниже мы приводим определение клеточного автомата с локаторами, в котором уже учтены замечания Г.В.Калачева.

Под *телесным углом* в $\mathbb{R}^k$ будем понимать часть пространства $\mathbb{R}^k$, которая является объединением всех лучей, выходящих из данной точки (*вершины угла*) и пересекающих некоторую гиперповерхность в $\mathbb{R}^k$, состоящую из гиперплоскостей, задаваемых линейными уравнениями с целыми коэффициентами. Заметим, что в классическом определении телесного угла нет таких ограничений на гиперповерхности, но для нашего определения эти ограничения важны, поскольку телесные углы, задаваемые вещественными числами, позволяют закодировать в угол бесконечное количество информации. По определению будем считать, что вершина телесного угла не входит в телесный угол.

Клеточным автоматом с локаторами называется восьмерка $\sigma = (\mathbb{Z}^k, Q, V, E, +, L, \varphi, \psi)$, где $\mathbb{Z}^k$ — множество k-мерных векторов с целыми координатами, Q — конечное множество, $V = (\alpha_1, \ldots, \alpha_{h-1})$ — упорядоченный набор попарно различных ненулевых векторов из $\mathbb{Z}^k$, $(E, +)$ — коммутативный моноид, т.е. коммутативная полугруппа с нейтральным элементом, $L = (\nu_1, \ldots, \nu_m)$ — упорядоченный набор попарно различных телесных углов в $\mathbb{R}^k$ с вершиной в начале координат, φ — функция, зависящая от переменных $x_0, x_1, \ldots, x_{h-1}, z_1, \ldots, z_m$, $\varphi : Q^h \times E^m \to Q$, ψ — функция, зависящая от переменных $x_0, x_1, \ldots, x_{h-1}, z_1, \ldots, z_m$, $\psi : Q^h \times E^m \to E$. Элементы множества $\mathbb{Z}^k$ называются *ячейками* клеточного автомата σ; элементы множества Q называются *состояниями ячейки* клеточного автомата σ; среди состояний ячейки есть состояние, которое называется состоянием покоя, и обозначается q_0; набор V называется *шаблоном соседства* клеточного автомата σ; элементы множества E называются *сигналами вещания*; нейтральный элемент множества E обозначается через 0, т.е. для любого x из E выполняется $x + 0 = x$; набор L называется *шаблон локаторов* клеточного автомата σ; функция φ называется *локальной функцией переходов* автомата σ; функция ψ называется *функцией вещания* автомата σ; переменные $x_0, x_1, \ldots, x_{h-1}$ принимают значения из Q, переменные $z_1, \ldots, z_m$ принимают значения из E. Обозначим $\mathbf{q}_0 = (q_0, \ldots, q_0) \in Q^h$, $\mathbf{0} = (0, \ldots, 0) \in E^m$. Ячейки, состояние которых отличается от состояния покоя, будем называть *активными*. Считаем, что на функцию переходов и функцию вещания наложены ограничения: $\varphi(\mathbf{q}_0, \mathbf{0}) = q_0$ — условие сохранения состояния покоя, и $\psi(\mathbf{q}_0, x) = 0$ для любого x из E, то есть неактивная ячейка, у которой нет активных соседей, не может посылать сигналы в эфир.

Если $\alpha \in \mathbb{Z}^k$, ν — телесный угол с вершиной в начале координат, то через $\nu(\alpha)$ обозначим телесный угол, полученный параллельным переносом угла ν в точку α.

Если $\alpha \in \mathbb{Z}^k$ — ячейка клеточного автомата σ, то множество $V(\alpha) = \{\alpha, \alpha + \alpha_1, \ldots, \alpha + \alpha_{h-1}\}$ называется *окрестностью ячейки* α, а множество $L(\alpha) = \{\nu_1(\alpha), \ldots, \nu_m(\alpha_m)\}$ называется *локаторами ячейки* α.

Состоянием клеточного автомата с локаторами σ назовем пару (e, f), где e — произвольная функция, определенная на множестве $\mathbb{Z}^k$, принимающая значения из E, называемая *состоянием эфира*, f — произвольная функция, определенная на множестве $\mathbb{Z}^k$, принимающая значения из Q и называемая *распределением состояний клеточного автомата с локаторами* σ. Такую функцию можно интерпретировать как некую мозаику, возникающую в k-мерном пространстве в результате приписывания каждой точке с целочисленными координатами некоторого состояния из множества Q и некоторого сигнала из множества E. Множество всевозможных состояний клеточного автомата с локаторами обозначим Σ.

Если $\alpha \in \mathbb{Z}^k$, (e, f) — состояние клеточного автомата с локаторами σ, то значение $e(\alpha)$ называем *сигналом ячейки* α*, определяемым состоянием* (e, f), а значение $f(\alpha)$ — *состоянием ячейки* α*, определяемым состоянием* (e, f). Для каждого $i \in \{1, \ldots, m\}$ значение

$$s_i(\alpha) = \sum_{\beta \in \nu_i(\alpha) \cap \mathbb{Z}^k} e(\beta) \tag{1}$$

называем *значением локатора* ν_i*, определяемым состоянием* (e, f). Здесь суммирование сигналов осуществляется с помощью определяющей операции + полугруппы E. В работе [2] показано, что сходимость ряда (1) не зависит от порядка суммирования.

На множестве Σ определим *глобальную функцию переходов* Φ клеточного автомата с локаторами σ, полагая $\Phi(e, f) = (e', f')$, где $(e, f), (e', f') \in \Sigma$ и для любой ячейки $\alpha \in \mathbb{Z}^k$ выполняются тождества

$$f'(\alpha) = \varphi(f(\alpha), f(\alpha + \alpha_1), \ldots, f(\alpha + \alpha_{h-1}), s_1(\alpha), \ldots, s_m(\alpha)),$$

$$e'(\alpha) = \psi(f(\alpha), f(\alpha + \alpha_1), \ldots, f(\alpha + \alpha_{h-1}), s_1(\alpha), \ldots, s_m(\alpha)).$$

Содержательная интерпретация отображения Φ такова, что сигнал каждой ячейки и состояние каждой ячейки "после перехода" определяется по состоянию упорядоченной окрестности ячейки и по значениям локаторов "до перехода" с помощью законов ψ и φ одинаково для всех ячеек.

Поведениями клеточного автомата с локаторами σ называем такие последовательности $(e_0, f_0), (e_1, f_1), (e_2, f_2), \ldots$ его состояний, для которых выполняется $(e_{i+1}, f_{i+1}) = \Phi(e_i, f_i)$ для всех $i = 0, 1, 2, \ldots$, причем (e_i, f_i) называется *состоянием клеточного автомата с локаторами* σ *в момент*

i, а (e_0, f_0) также называется *начальным состоянием клеточного автомата с локаторами* σ.

Состояние клеточного автомата, у которого лишь конечное число ячеек активны, назовем *конфигурацией*.

Клеточный автомат с локаторами назовем *финитным*, если любая конфигурация переводится глобальной функцией переходов Φ в конфигурацию.

Отметим, что ряд (1) в общем случае может расходится, и тогда значение $s_i(\alpha)$ будет не определено. В дальнейшем мы будем рассматривать либо только финитные клеточные автоматы с локаторами, либо клеточные автоматы, для которых $(E, +)$ является идемпотентным моноидом, т.е. для любого $x \in E$ выполнено $x + x = x$. В обоих этих случаях значение $s_i(\alpha)$ всегда будет определено.

Рассмотрим задачу поиска ближайшего соседа на прямой, т.е. ячейки клеточного автомата принадлежат $\mathbb{Z}^1$. В правильной начальной конфигурации одна ячейка отмечена как центральная, еще имеется конечное множество активных ячеек, называемых соседями. В правильной конечной конфигурации должены остаться центральная ячейка и только один сосед, который ближе всего находится к центральной ячейке, и между центральной ячейкой и ближайшим соседом проложен путь из активных ячеек.

Клеточный автомат с локаторами *решает задачу поиска ближайшего соседа*, если для любой правильной начальной конфигурации через конечное число шагов, называемое *временем решения*, клеточный автомат переходит в правильную конечную конфигурацию, которая в дальнейшем не изменяется.

Если задана некоторая правильная начальная конфигурация, то *сложностью данной начальной конфигурации* назовем минимальное время решения, взятое по всем клеточным автоматам с локаторами, решающим задачу поиска ближайшего соседа.

Теорема 1. *Для любой правильной начальной конфигурации ее сложность по порядку равна логарифму расстояния до ближайшего соседа.*

Верхняя оценка теоремы доказана в работе [3], а нижняя — в работе [4].

Отметим, что в общем случае эта задача не может быть решена обычными клеточными автоматами, а для случая ограниченного некоторой константой числа соседей время решения будет пропорционально расстоянию до самого далекого соседа.

Список литературы

[1] Гасанов Э. Э. Клеточные автоматы с локаторами // Интеллектуальные системы. Теория и приложения. — 2020. — Т. 24, № 2. — С. 121–133.

[2] Калачев Г. В. Замечания к определению клеточного автомата с локаторами // Интеллектуальные системы. Теория и приложения. — 2020. — Т. 24, № 4. — С. 47–56.

[3] Васильев Д. И. Поиск ближайшего соседа на прямой с помощью клеточного автомата с локаторами // Интеллектуальные системы. Теория и приложения. — 2020. — Т. 24, № 3. — С. 99–119.

[4] Васильев Д. И. Нижняя оценка сложности задачи поиска ближайшего соседа на прямой с помощью клеточного автомата с локаторами // Вестник Московского государственного университета. Серия 1. Математика, механика. — В печати.

Более точные оценки функции Шеннона глубины схем из функциональных элементов с емкостными параметрами выходов элементов

Данилов Борис Радиславович[1], **Ложкин Сергей Андреевич**[2]

[1] МГУ им. М.В.Ломоносова, e-mail: brdanilov@gmail.com

[2] МГУ им. М.В.Ломоносова, e-mail: lozhkin@cs.msu.ru

Настоящая работа продолжает исследования [1] варианта модели [2, 3] обобщённой глубины схем из функциональных элементов (СФЭ) в произвольном конечном полном базисе $Б = \{\mathcal{E}_1, \ldots, \mathcal{E}_b\}$, где функциональный элемент (ФЭ) типа $\mathcal{E}_i$, $1 \leqslant i \leqslant b$, имеет арность k_i и реализует функцию алгебры логики (ФАЛ) $\varphi_i(x_1, \ldots, x_{k_i})$, которая в случае $k_i \geqslant 2$ существенно зависит от всех своих булевых (т. е. заданных на множестве $E_2 = \{0, 1\}$) переменных. С функциональным элементом $\mathcal{E}_i$ связаны его сложностные характеристики — *базовая глубина* d_i и *множитель глубины* w_i, которые являются действительными числами. Элементы базиса различны, когда они отличаются по своим функциональным или сложностным параметрам.

Рассмотрим понятия обобщённой, классической и структурной глубин, для которых будем использовать термины D-глубина, d-глубина и δ-глубина (соответственно). Нам будет удобно рассматривать понятия глубины не только для ФЭ схемы, но также и для её входов, с которыми для этой цели связываются некоторые (умозрительные) ФЭ базиса Б. Сначала понятие глубины определяется для отдельных элементов. Пусть v — элемент (возможно, вход) типа $\mathcal{E}_i$, $1 \leqslant i \leqslant b$, в схеме Σ. Структурная глубина $\delta(v)$ элемента v равна нулю, если v — вход, и единице для любого другого

элемента. Классическая глубина $d(v)$ элемента v равна базовой глубине d_i связанного с ним ФЭ базиса. Обобщённая глубина $D(v)$ зависит от *степени ветвления* m элемента v в Σ — то есть кратности выхода той подсхемы, которая состоит из одного этого элемента — и определяется по формуле: $D(v) = d_i + (m-1) \cdot w_i$. Элемент назовём *кратным*, если его степень ветвления строго больше единицы.

Далее определение глубины распространяется на подсхемы специального вида — цепи. Назовём *цепной* схемой СФЭ без ветвлений входов, которая представляет собой последовательность ФЭ, в которой каждый элемент, кроме первого, соединён с предыдущим и только с ним, а последний элемент является выходом (возможно, не единственным) схемы. Выделенным выходом цепи назовём выход её последнего ФЭ. Если в цепи выделен (какой-то) вход первого её ФЭ, то такую цепь назовём *инициальной*. Обобщённая (классическая, структурная) глубина инициальной цепи — это сумма обобщённых (базовых, структурных) глубин её выделенного входа и всех последующих элементов этой цепи.

Для СФЭ Σ общего вида её обобщённая (классическая, структурная) глубина $D(\Sigma)$ $(d(\Sigma), \delta(\Sigma))$ — это наибольшая из обобщённых (классических, структурных) глубин её *главных* инициальных цепей, т. е. тех цепных подсхем, выделенным входом которых является какой-то вход схемы Σ, а выделенным выходом — выход схемы Σ.

В работе рассматривается подкласс СФЭ — класс так называемых *усилительных* СФЭ (УСФЭ). Пусть в базисе Б выделен один ФЭ, имеющий единственный вход и реализующий тождественную ФАЛ, который называется усилителем. Без ограничения общности условимся, что усилительным является элемент $\mathcal{E}_1$ ($k_1 = 1$, $\varphi_1(x_1) = x_1$). Усилительная СФЭ — это СФЭ, в которой все кратные элементы являются усилителями, что относится также и к входам схемы.

Пусть $A \in \{D, d, \delta\}$. Выше определена A-глубина схемы. Определим аналогичное понятие для ФАЛ f. Так, под A-глубиной f в классе УСФЭ будем понимать наименьшую из A-глубин УСФЭ над Б, реализующих f. Обозначим через $P_2(n)$ множество всех ФАЛ, зависящих от булевых переменных $x_1, \ldots, x_n$. Функцией Шеннона $A_{\text{Б}}(n)$ для A-глубины ФАЛ из $P_2(n)$ в классе УСФЭ над Б называется наибольшая A-глубина ФАЛ из $P_2(n)$.

Известно [4, 5], что для для классической глубины поведение функции Шеннона $d_{\text{Б}}(n)$ имеет вид* $d_{\text{Б}}(n) = \tau'_{\text{Б}}(n - \log_2 \log_2 n) \pm O(1)$, и определяется константой $\tau'_{\text{Б}}$, называемой приведённой (классической) глубиной

*Указанный результат и результаты, о которых говорится ниже, получены для СФЭ общего вида, но остаются справедливыми и для более узкого класса УСФЭ

базиса Б:

$$\tau'_{\text{Б}} = \min_{1 \leqslant i \leqslant b,\ k_i \neq 1} \tau'_i, \qquad \tau'_i = \frac{d_i}{\log_2 k_i},$$

где τ'_i — приведённая (классическая) глубина элемента $\mathcal{E}_i$, $k_i \neq 1$. Из этих результатов следует, что функция Шеннона для структурной глубины имеет вид $\delta_{\text{Б}}(n) = n - \log_2 \log_2 n \pm O(1)$, но для структурной глубины в некоторых специальных базисах, состоящих из двухвходовых элементов, известно [6, 7] большее: $\delta_{\text{Б}}(n) = \lceil n - \log_2 \log_2 n \pm o(1) \rceil$.

В работе [1] установлено поведение функции Шеннона $D_{\text{Б}}(n)$ обобщённой глубины вида $D_{\text{Б}}(n) = \tau_{\text{Б}} n \pm O(\log n)$, где константа $\tau_{\text{Б}}$ — приведённая обобщённая глубина базиса Б — определялась по формулам:

$$\tau_{\text{Б}} = \tau'_{\text{Б}} + \tau''_{\text{Б}}, \qquad \tau''_{\text{Б}} = \min_{m \geqslant 2} \frac{d_1 + (m-1) \cdot w_1}{\log_2 m}.$$

В настоящей работе улучшена верхняя оценка для $D_{\text{Б}}(n)$, из которой вместе с соответствующей нижней оценкой из работы [1] вытекает точная по порядку второго члена асимптотического разложения оценка этой функции $D_{\text{Б}}(n) = \tau_{\text{Б}} n - O(\log n)$.

Основным результатом работы является теорема:

Теорема 1. *Имеет место асимптотическая оценка функции Шеннона обобщённой глубины УСФЭ над базисом* Б: $D_{\text{Б}}(n) \leqslant \tau_{\text{Б}} n - O(\log n)$.

Статья опубликована при финансовой поддержке Минобрнауки РФ в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2019-1621.

Список литературы

[1] *Данилов Б. Р., Ложкин С. А.* Асимптотическое оценки функции Шеннона в одной модели глубины схем из функциональных элементов с емкостными параметрами выходов элементов // Прикладная математика и информатика. 2018. № 59. С. 40–49.

[2] *Данилов Б. Р.* Асимптотическое оценки функции Шеннона в одной модели глубины схем из функциональных элементов с емкостными параметрами выходов элементов // Труды X международной конференции «Дискретные модели в теории управляющих систем» (Москва и Подмосковье, 23–25 мая 2018 г.). М.: МАКС Пресс, 2018. С. 112–113.

[3] *Данилов Б. Р.* О поведении функции Шеннона для задержки схем в модели, где задержка соединений определяется типами соединяемых элементов // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2014. Т. 3, № 31. С. 78–100.

[4] *Лупанов О. Б.* О схемах функциональных элементов с задержками // Проблемы кибернетики. Вып. 23. М: Наука, 1970. С. 43–82.

[5] *Ложкин С. А.* О глубине функций алгебры логики в произвольном полном базисе // Вестник Московского университета. Серия 1. Математика. Механика. 1996. № 2. С. 80–82.

[6] *Гашков С. Б.* О глубине булевых функций // Проблемы кибернетики. Вып. 34. М.: Наука, 1978. С. 265–268.

[7] *Ложкин С. А.* О глубине функций алгебры логики в некоторых базисах // Univ. Sci. Budapest. Sec. Comput. 1983. С. 127–131.

Об обобщении проверки однозначности алфавитного декодирования

Дергач Петр Сергеевич

Московский Государственный Университет имени М. В. Ломоносова, e-mail: dergachpes@mail.ru

Аннотация

Данная статья является тезисами к докладу автора на XIX международной конференции «Проблемы Теоретической Кибернетики». В докладе рассказывается о двух способах решения проблемы однозначности алфавитного декодирования — марковский подход и подход автора с использованием теории автоматов. Приводятся более точные оценки, чем были известны ранее. Также приводятся результаты для обобщения этой задачи на случай многоместных склеек — на примере трехэлементной склейки.

Введение

В теории кодирования фундаментальное значение имеет модель алфавитного побуквенного кодирования [1]. Очевидно, что при передаче по каналу связи закодированного этим способом сообщения для последующей дешифровки необходимо, чтобы соответствующая функция кодирования переводила разные слова в разные коды, то есть была инъективной. Можно рассмотреть и более мягкое ограничение — отсутсвие трех слов с одинаковым кодом. Константа 3 здесь взята лишь для примера, чтобы было проще интерпретировать дальнейшие результаты. Подробнее эти и смежные вопросы изложены в диссертации [2].

Основные определения

Понятия схемы алфавитного кодировании, функции алфавитного кодирования широко известны, с ними можно ознакомиться в [1,3]. Также не

будем приводить и понятия регулярного языка и автомата, с которыми можно ознакомиться в [2,4].

Пусть A и B — два конечных непустых алфавита. Через $F(A,B)$ обозначаем множество схем алфавитного кодирования f из алавита A в слова алфавита B. Образы букв в схеме называем элементарными кодами. Для произвольной $f \in F(A,B)$ обозначаем через $\tilde{f}$ доопределение схемы до функции алфавитного кодирования на A^*. Через $L_1(f)$ обозначаем сумму длин элементарных кодов схемы f, а через $L_2(f)$ — сумму квадратов длин элементарных кодов этой же схемы. Через $r(A)$ обозначаем мощность A. Склейкой называем пару различных слов в алфавите A с общим кодом. Если же есть три попарно различных слова с общим кодом, то называем их трехэлементной склейкой. Минимальное количество состояний автомата, распознающего множество $\tilde{f}(A^*)$, обозначим через $Q(f)$. Длину максимального элементарного кода в схеме f обозначим через $l(f)$.

Основные результаты

Теорема 1. *Для проверки однозначности алфавитного кодирования $\tilde{f}$ по произвольной схеме $f \in F(A,B)$ достаточно перебрать все пары слов в алфавите A с суммарной длиной не больше чем $L_1(f) - r(A) + 2$.*

Доказательство. Ввиду ограниченности объема тезисов приводим здесь и далее лишь краткую идею доказательств. За основу доказательства можно взять хорошо известную теорему Маркова, доказательство которой есть, например, в [1]. По двум словам с одинаковым кодом строится общее измельчение этого кода граничными позициями элементарных кодов первого и второго разбиений. Сразу можно считать, что данное разбиение неприводимо, то есть границы одного разбиения не совпадают нигде кроме начала и конца общего кода с границами другого разбиения. Далее, можно заметить (это тоже идея А.А.Маркова), что склейку можно представить как слово в двухъярусном алфавите — либо подается буква из первого слова, либо подается буква из второго слова. Порядок появления букв строго однозначен — пока длинее код уже поданной по второму слову части букв, мы будем подавать очередную букву из первой части. Иначе подаем очередную букву из второй части. Осталось заметить, что при появлении очередной буквы возникает текущий перевес между кодом первой и второй частей. И этот перевес является собственным окончанием какого-то элементарного кода схемы f. Также из доказательства теоремы Маркова известно, что в случае повторения двух таких окончаний склейку можно было бы сократить, выкинув из кода общую часть между этими двумя окончаниями и одно из них. Результат по-прежнему остается склейкой, а значит можно считать, что всего таких окончаний не более чем $L_1(f) - r(A)$ — общее количество собственных окончаний элементарных кодов в схеме

f. Осталось заметить, что лишь первая и последняя поданные буквы (в двухъярусном смысле) не приводят к возникновению собственных окончаний. **Теорема 1 доказана.**

Теорема 2. *Для проверки наличия трехэлементной склейки у алфавитного кодирования $\tilde{f}$ по произвольной схеме $f \in F(A, B)$ достаточно перебрать все тройки слов в алфавите A с суммарной длиной не больше чем* $\frac{L_2(f)+3L_1(f)+2}{2}$.

Доказательство. Доказательство теоремы в целом опирается на доказательство теоремы 1. Отличие в том, что теперь трехэлементная склейка задается последовательностью букв в трехъярусном алфавите. И при подаче очередной такой буквы появляется какой-то элементарный код, задающий самое длинное на данный момент продвижение и две части от него поменьше. Легко заметить, что это размещения с повторениями и, например, для элементарного кода длины l способов отрезать от него 2 окончания не больше чем $C^2_{l+2} = \frac{(l+2)*(l+1)}{2}$. При этом порядок этих слов неважен — по-прежнему можно соединять такие куски, переплетая коды между собой. Остается просуммировать полученную оценку по всем элементарным кодам. **Теорема 2 доказана.**

Теорема 3. *Для проверки однозначности алфавитного кодирования $\tilde{f}$ по произвольной схеме $f \in F(A, B)$ достаточно перебрать все пары слов в алфавите A длины не больше* $1 + l(f) + Q^2(f)$.

Доказательство. Этот результат является простым следствием теоремы 1.1 (страница 26) из работы [2], в которой исследуется более общий случай произвольного входного регулярного языка. Сама идея автоматной техники в том, что мы следим за парой состояний автоматов, распознающих $\tilde{f}(A^*)$. Если пары состояний начинают повторяться, то склейку можно сократить. Остальные два слагаемых из теоремы нужны для оценивания той части, которая гарантирует нам, что после сокращения склейки мы получим склейку, а не пару одинаковых входных слов. **Теорема 3 доказана.**

Теорема 4. *Для проверки наличия трехэлементной склейки у алфавитного кодирования $\tilde{f}$ по произвольной схеме $f \in F(A, B)$ достаточно перебрать все тройки слов в алфавите A в алфавите A длины не больше* $1 + l(f) + Q^2(f) + Q^3(f)$.

Доказательство. Доказательство похоже на доказательство предыдущей теоремы. Слагаемое $1 + l(f)$ возникает для сохранения части, различающей хотя бы пару слов из тройки. Слагаемое $Q^2(f)$ нужно, чтобы гарантированно отличить оставшееся третье слово. И последнее слагаемое $Q^3(f)$ отвечает за неповторение наборов из трех состояний. **Теорема 4 доказана.**

Список литературы

[1] Яблонский С. В. Введение в дискретную математику // М.: Наука, 1986.— 384 с.

[2] Дергач П. С. Алфавитное кодирование регулярных языков с полиномиальной функцией роста : дис. ... канд. физ.-мат. наук : 01.01.09 : защищена 21.10.16 / Дергач Петр Сергеевич. — Москва, 2016. — 213 с.

[3] Марков А. А. Основания общей теории кодов // Проблемы кибернетики. — Вып. 31. — М.: Наука, 1976. — С. 77–108.

[4] В. Б. Кудрявцев, С. В. Алешин, А. С. Подколзин. Введение в теорию автоматов // М.: Наука, 1985.— 320 с.

Об интервалах в решетке замкнутых классов частичных монотонных функций k-значной логики

Дудакова Ольга Сергеевна

МГУ им. М. В. Ломоносова, e-mail: olga.dudakova@gmail.com

В работе исследуются замкнутые классы частично определенных функций многозначной логики [1 ч. II гл. 20]. Изучаются классы частичных функций, содержащих замкнутый класс всюду определенных монотонных функций.

Пусть $E_k = \{0, 1, 2, \ldots, k-1\}, k \geq 2$. Через P_k и P_k^* обозначаются множество всех функций k-значной логики и множество всех частичных функций на E_k, то есть функций, которые наборах из E_k принимают значения из множества $\{0, 1, 2, \ldots, k-1, *\}$, где $*$ трактуется как неопределенность. Известно, что мощность семейства всех замкнутых классов в P_k^* континуальна, поэтому представляет интерес исследование отдельных семейств классов, в частности, классов, содержащих заданный класс всюду определенных функций.

Пусть на множестве E_k задан нетривиальный частичный порядок $\leq$. Обозначим через M класс всех всюду определенных монотонных функций, через $\widehat{M}^*$ — класс всех частичных функций, монотонных на области определенности (то есть на множестве наборов, значение функции на которых отлично от $*$), и через M^* — класс всех частичных функций, которые можно доопределить до функций из M заменой $*$ на некоторые значения из E_k (в книге [1] класс частичных функций, доопределяемых до монотонных, обозначается через $Str(M)$). Легко видеть, что $M \subset M^* \subset \widehat{M}^*$.

Пусть F_1 и F_2 — замкнутые классы в P_k^*. Положим

$$\mathcal{I}(F_1, F_2) = \{G \subseteq P_k^* \mid [G] = G, F_1 \subseteq G \subseteq F_2\}.$$

Известно [1–4], что если M — предполный класс в P_k, то интервал $\mathcal{I}(M, \widehat{M^*})$ содержит конечное число классов тогда и только тогда, когда частично упорядоченное множество $(E_k, \leq)$ является решеткой. В случае, когда $(E_k, \leq)$ — решетка, число замкнутых классов в интервале $\mathcal{I}(M, \widehat{M^*})$ равно 6 для любого $k \geq 2$ (см. [1, 2]); отметим, что в этом случае классы M^* и $\widehat{M^*}$ совпадают. Если частичный порядок таков, что M — предполный класс, но $(E_k, \leq)$ не является решеткой, то интервал $\mathcal{I}(M, M^*)$ содержит 6 классов, а в интервале $\mathcal{I}(M^*, \widehat{M^*})$ содержится бесконечное число классов. Если частичный порядок таков, что класс M всех монотонных функций не является предполным в P_k, то интервал $\mathcal{I}(M, \widehat{M^*})$ содержит бесконечное число замкнутых классов [5]. В данной работе показано, что если класс M не является предполным в P_k, то число замкнутых классов в интервале $\mathcal{I}(M, M^*)$ конечно и равно 6 или 7 в зависимости от задания частичного порядка на E_k.

Произвольному частично упорядоченному множеству $\mathcal{E}$ можно поставить в соответствие граф $G(\mathcal{E})$ — диаграмму Хассе (см., например, [1] ч. I гл. 2). В случае, когда $G(\mathcal{E})$ является несвязным, множество $\mathcal{E}$ будем называть несвязным, а подмножества $\mathcal{E}$, соответствующие компонентам связности графа $G(\mathcal{E})$, будем называть компонентами связности множества $\mathcal{E}$. Очевидно, что множество $\mathcal{E}$ несвязно тогда и только тогда, когда для любого $n \geq 1$ множество E^n несвязно.

Рассмотрим следующие классы частичных функций (см. также [1, 2]):

$M \cup \{*\}$, где $\{*\}$ — множество всех функций, принимающих только значение $*$;

M_1 — множество всех таких функций $f(x_1, \ldots, x_n) \in M^*$, что для любых наборов $\widetilde{a}$ и $\widetilde{b}$ из E_k^n, таких что $\widetilde{a} \leq \widetilde{b}$, выполняется либо $f(\widetilde{a}), f(\widetilde{b}) \neq *$ (и в этом случае $f(\widetilde{a}) \leq f(\widetilde{b})$), либо $f(\widetilde{a}) = *$;

M_2 — множество всех таких функций $f(x_1, \ldots, x_n) \in M^*$, что для любых наборов $\widetilde{a}$ и $\widetilde{b}$ из E_k^n, таких что $\widetilde{a} \leq \widetilde{b}$, либо $f(\widetilde{a}), f(\widetilde{b}) \neq *$ (и в этом случае $f(\widetilde{a}) \leq f(\widetilde{b})$), либо $f(\widetilde{b}) = *$;

M_3 — множество всех таких функций $f(x_1, \ldots, x_n) \in M^*$, что для любых трех наборов $\widetilde{a}, \widetilde{b}, \widetilde{c} \in E_k^n$, таких что $\widetilde{a} \leq \widetilde{b} \leq \widetilde{c}$, либо $f(\widetilde{b}) \neq *$, либо $f(\widetilde{b}) = *$ и тогда по крайней мере на одном из наборов $\widetilde{a}, \widetilde{c}$ функция принимает значение $*$.

Кроме того, определим множество M_0 всех функций $f(x_1, \ldots, x_n) \in M^*$, удовлетворяющих следующему условию: пусть $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_s$ — все связные

компоненты множества $(E_k^n, \leq)$, $s \geq 1$. Если для некоторого $i \in \{1, \ldots, s\}$ найдется набор $\widetilde{a} \in \mathcal{E}_i$, такой что $f(\widetilde{a}) = *$, то f принимает значение $*$ на всех наборах множества $\mathcal{E}_i$. Нетрудно показать, что множество M_0 является замкнутым классом в P_k^*. Если $(E_k, \leq)$ — связное множество, то класс M_0 совпадает с $M \cup \{*\}$, а если $(E_k, \leq)$ несвязно, то эти классы различны.

Утверждение 1. *Выполняются следующие включения:* $M \subset M \cup \{*\} \subseteq M_0$; $M_0 \subset M_1, M_2$; $M_1, M_2 \subset M_3 \subset M^*$. *Строгое включение* $M \cup \{*\} \subset M_0$ *выполняется тогда и только тогда, когда множество* $(E_k, \leq)$ *несвязно.*

Теорема. *Пусть на* E_k *задан нетривиальный частичный порядок* $\leq$, *такой что множество* $(E_k, \leq)$ *не содержит наибольшего или наименьшего элемента. Тогда интервал* $\mathcal{I}(M, M^*)$ *содержит 6 или 7 классов в зависимости от задания частичного порядка на* E_k, *и это все классы, перечисленные в утверждении 1.*

Доказательство теоремы во многом аналогично доказательству теоремы 2 из работы [2], основное отличие заключается в появлении дополнительного класса M_0 в случае, когда множество $(E_k, \leq)$ несвязно.

Список литературы

[1] Lau D. Function algebras on finite sets: a basic course on many-valued logic and clone theory. — Springer Monographs in Mathematics. — Berlin. Springer, 2006. — 668 p.

[2] Алексеев В. Б., Вороненко А. А. О некоторых замкнутых классах в частичной двузначной логике // Дискретная математика. — 1994. — Т. 6, вып. 4. — С. 58–79.

[3] Алексеев В. Б. О замкнутых классах в частичной k-значной логике, содержащих класс монотонных функций // Дискретная математика. — 2018. — Т. 30, вып. 2. — С. 3–13.

[4] Дудакова О. С. Построение бесконечного семейства классов частичных монотонных функций многозначной логики // Вестник Московского университета. Серия 1: Математика. Механика. — 2019. — N.º 1. — С. 3–7.

[5] Дудакова О. С. О классах частичных монотонных функций трехзначной логики // Материалы XIII Международного семинара «Дискретная математика и ее приложения» им. акад. О. Б. Лупанова (Москва, МГУ, 17—22 июня 2019 г.). — Изд-во мех.-матем. ф-та МГУ. — С. 173–175.

Оценки мощности объёмных схем для класса частичных булевых операторов

Ефимов Алексей Андреевич[1], **Калачёв Глеб Вячеславович**[2]

[1] МГУ имени М. В. Ломоносова, e-mail: efimovqwerty@yandex.ru

[2] МГУ имени М. В. Ломоносова, e-mail: gleb.kalachev@yandex.ru

В данной работе рассматриваются схемы функциональных элементов (СФЭ) специального вида. На практике большой интерес вызывают схемы, где учитываются различные пространственные характеристики (размеры проводов и функциональных элементов, расположение их в пространстве). Одним из первых такие модели СФЭ (с ограничениями на длину проводников) стал рассматривать А.Д. Коршунов в работе [1].

В этой работе мы будем рассматривать такую модель СФЭ, как объёмные схемы. *Объёмной схемой* K или *схемой из кубических элементов* будем называть такую укладку СФЭ в трёхмерную целочисленную решётку $\mathbb{Z}^3$, чтобы в каждое ребро решётки попадало не более одного ребра СФЭ. Таким образом в каждой вершине решётки реализуется булев оператор, у которого в сумме не более 6 входов и выходов. Будем говорить, что схема K *реализует* булев оператор F, если соответствующая СФЭ реализует F. Через $\mathrm{Impl}(F)$ обозначим множество всех объёмных схем, реализующих оператор F.

Узлами объёмной схемы K будем называть рёбра решётки $\mathbb{Z}^3$, в которые уложены провода СФЭ. Отметим, что в результате такой укладки вершины СФЭ (в вершинах которых реализуются только булевы функции) склеиваются в вершины решётки $\mathbb{Z}^3$ (в вершинах которой реализуются булевы операторы), поэтому узлы объёмной схемы K соответствуют вершинам СФЭ, что важно для дальнейших определений. Для каждой схемы K зафиксируем некоторую нумерацию её узлов. Функцию, реализуемую в i-м узле, обозначим через g_i (на входах схемы считаем, что реализуются тождественные функции).

Потенциалом схемы K *на входном наборе* $x \in \{0,1\}^n$ назовём величину $u_K(x) = \sum_{i=1}^{\ell} g_i(x)$, где ℓ — число узлов в схеме K. Таким образом, потенциал схемы на наборе — количество узлов схемы, значение в которых равно 1 на этом наборе.

Максимальным потенциалом схемы K назовём величину

$$\hat{U}(K) := \max_{x \in \{0,1\}^n} u_K(x).$$

Пусть $f : \{0,1\}^n \to \{0,1\}^m$ – булев оператор. Тогда

$$\hat{U}(f) := \min_{K \in \mathrm{Impl}(f)} \hat{U}(K).$$

Средним потенциалом схемы K с n входами на множестве входных наборов $D \subseteq \{0,1\}^n$ назовём величину

$$U_D(K) := \frac{1}{|D|} \sum_{x \in D} u_K(x).$$

Пусть $f : D \to \{0,1\}^m$ – частичный булев оператор. Тогда

$$U(f) := \min_{K \in \mathrm{Impl}(f)} U_D(K).$$

Множество булевых операторов $f : \{0,1\}^n \to \{0,1\}^m$ обозначим $P_2(n,m)$. Введём функцию Шеннона для среднего и максимального потенциала:

$$U(n,m) := \max_{f \in P_2(n,m)} U(f), \quad \hat{U}(n,m) := \max_{f \in P_2(n,m)} \hat{U}(f).$$

В работе [3] для всюду определённых операторов получена верхняя оценка потенциала

$$\hat{U}(n,m) = \mathcal{O}\left(\frac{m2^{n/3}}{\min^{2/3}(m,n)}\right).$$

В данной работе получена нижняя оценка потенциала для класса частичных булевых операторов, сформулированная в следующей теореме.
Теорема 1. *Существует константы $C_1 > 0$, $C_2 > 0$ такие, что если $D \subseteq \{0,1\}^n$, то доля операторов $f : D \to \{0,1\}^m$, для которых выполнено неравенство*

$$U(f) \geq C \frac{m\sqrt[3]{|D|}}{\min^{2/3}(m, \log_2 |D|)},$$

не меньше $\alpha(n,m,|D|)$, причём

$$\alpha(n,m,d) \to 1 \quad \textit{при} \quad d \to \infty,\ n\log_2 n = o(d),\ \log_2 m \leq C_2 d.$$

Другими словами, в теореме утверждается, что если область определения D содержит существенно больше $n \log n$ наборов и число выходов m не близко к числу всевозможных частичных функций на D, то потенциал почти всех операторов $f : D \to \{0,1\}^m$ по порядку не меньше $\frac{m\sqrt[3]{|D|}}{\min^{2/3}(m,\log_2|D|)}$.

В ходе доказательства теоремы 1 была разработана модификация метода расслоения, изначально предложенного в работе [2]. Идея метода расслоения состоит в том, чтобы считать потенциал схемы по слоям. Для

данной схемы строится семейство её разрезов такое, чтобы множества проводов в разных разрезах не пересекались. Под разрезом схемы здесь понимается множество проводов, при удалении которых схема распадается на 2 части. Далее оценивается снизу среднее число единиц на проводах, входящих в каждый разрез по-отдельности. Итоговая оценка потенциала складывается из оценок для всех разрезов.

В доказательстве нижней оценки потенциала плоских схем для класса частичных операторов в работе [2] такой подход приводит к существенным техническим трудностям, связанными с тем, что при построении расслоения характеристики подсхем изменяются дискретно, что сильно осложняет точные оценки и ведёт к необходимости некоторых огрублений.

В данной работе предлагается модифицированный подход, позволяющий строить «непрерывное» расслоение. Он основан на геометрическом представлении схемы в пространстве таким образом, что элементы расположены в узлах целочисленной решётки, а провода — отрезки, соединяющие соседние вершины. При этом, если провод активен (значение на нём равно 1), то энергия выделяется равномерно по всей его длине. Вместо того, чтобы рассматривать расслоение, в котором разрезом является множество проводов, мы будем рассматривать пересечение бесконечного семейства слоёв с проводами схемы. Вместо суммирования мы будем интегрировать по элементам расслоения таким образом, чтобы потенциал каждого провода получался как интеграл по множеству слоев, пересекающих этот провод.

В качестве следствия из Теоремы 1 с учётом верхней оценки из работы [3] можно получить порядок функции Шеннона для всюду определенных операторов.

Следствие 1.

$$U(n,m) \asymp \hat{U}(n,m) \asymp \frac{m2^{n/3}}{\min^{2/3}(m,n)}$$

при $n \to \infty, \log_2 m = o(2^n)$.

Список литературы

[1] Коршунов А. Д. Об оценках сложности из объемных функциональных элементов и объемных схем из функциональных элементов // Проблемы кибернетики. Наука — 1967. — Т. 19. — С. 275–283.

[2] Калачёв Г. В. Порядок мощности плоских схем, реализующих булевы функции // Дискретная математика — 2014. — Т. 26, № 1. — С. 49–74.

[3] Ефимов А. А. Верхняя оценка энергопотребления объемных схем, реализующих булевы операторы // Интеллектуальные системы. Теория и приложения. — 2019. — Т. 23, № 2. — С. 105–124.

Синтез некоторых типов бинарных программ, допускающих рекурсивный вызов процедур ограниченной глубины

Жуков Владимир Владимирович

Московский государственный университет им. М. В. Ломоносова, e-mail: zhvv117@gmail.com

Задача синтеза, которая впервые была рассмотрена Шенноном [1], состоит в поиске наиболее оптимальных методов построения дискретных управляющих систем для произвольной булевой функции или систем таких функций. Для оценки оптимальности метода синтеза вводится функция Шеннона, которая для заданного значения n равна сложности самой сложной функции, зависящей от n переменных. При этом сложностью функции называют наименьшую сложность управляющей системы, реализующей данную функцию. Под сложностью управляющей системы чаще всего понимают количество элементов в ней или их суммарный вес.

О. Б. Лупановым [2] был предложен асимптотически наилучший метод синтеза схем из функциональных элементов (СФЭ) в полных конечных базисах. С его помощью была получена асимптотически точная верхняя оценка функции Шеннона для сложности реализации булевых функций в классе СФЭ в произвольном полном конечном базисе $Б$, равная $\rho_Б \cdot 2^n/n$, где $\rho_Б$ — константа, называемая приведённым весом базиса $Б$.

С. В. Грибком [3] была рассмотрена модель бинарных программ, реализующих булевы функции, которые состоят из одного или нескольких модулей, содержащих вычислительные и переадресующие команды, а также команды вызова процедур. В указанной работе была установлена асимптотика функций Шеннона для сложности реализации булевых функций в различных классах бинарных программ, а в некоторых случаях удалось получить для них асимптотические оценки высокой степени точности.

Одним из ограничений модели бинарных программ, рассматриваемых С. В. Грибком и другими авторами, является отсутствие возможности рекурсивного вызова процедур. В настоящей работе рассматривается модель бинарных программ, в которой такой вызов процедур возможен, т. е. в процессе выполнения программы процедуры могут непосредственно или через другие процедуры вызывать сами себя. Для данной модели предложены методы синтеза программ, реализующие произвольные булевы функции, и методы получения нижних оценок функций Шеннона для их сложности, с помощью которых при определённых ограничениях установлена асимптотика указанных функций Шеннона.

Возьмём счётное множество булевых переменных (БП) $X = \{x_1, \ldots, x_n, \ldots\}$, которые будут являться аргументами рассматриваемых далее буле-

вых функций. Каждая булева переменная принимает значения из множества $B = \{0, 1\}$. Булевым кубом размерности n, $n \in \mathbb{N}$, называется n-я декартова степень множества B и обозначается B^n, а функцией алгебры логики или булевой функцией $f(x_1, \ldots, x_n)$ — отображение булева куба B^n на множество B. Множество всех булевых функций, зависящих от n переменных обозначим $P_2(n)$.

Рассмотрим полный конечный базис $Б = \{\varphi_1, \ldots, \varphi_b\}$, состоящий из b булевых функций $\varphi_1, \ldots, \varphi_b$, в котором каждая функция φ_i, $i = \overline{1, b}$, существенно зависит от r_i, $r_i \geq 0$, переменных. Также припишем каждой функции φ_i, $i = \overline{1, b}$, некоторый положительный вещественный вес L_i.

Бинарной программой Σ назовём набор подпрограмм $\{\Sigma_1, \ldots, \Sigma_s\}$, для каждой из которых задан набор из $in(i)$, $i = \overline{1, s}$, входных аргументов, $out(i)$, $i = \overline{1, s}$, выходных аргументов, упорядоченный набор команд Γ_i, а также конечный размер области памяти M_i, $M_i \geq in(i) + out(i)$, используемой данной подпрограммой.

Команды подпрограмм могут быть трёх типов:

1. Вычислительные команды $\{\varphi_j; m_1, \ldots, m_{r_j}; m_{out}\}$ описываются символом булевой функции $\varphi_j(x_1, \ldots, x_{r_j})$ из заданного полного конечного базиса $Б$, номерами входных ячеек памяти $m_1, \ldots, m_{r_j}$ и номером выходной ячейки памяти $m_{out} > in(i)$.
2. Переадресующие команды $\{m; c_{false}, c_{true}\}$ описываются номером ячейки памяти m и номерами команд текущей подпрограммы c_{false} и c_{true}, на которые осуществляется условный переход.
3. Команды вызова подпрограмм $\{p; m_1, \ldots, m_{in(p)}; m'_1, \ldots, m'_{out(p)}\}$ описываются номером вызываемой подпрограммы p, номерами ячеек памяти текущей подпрограммы $m_1, \ldots, m_{in(p)}$, значения которых будут использованы как входные аргументы вызываемой подпрограммы, а также номерами ячеек памяти текущей подпрограммы $m'_1, \ldots, m'_{out(p)}$, куда будут записаны результаты выполнения подпрограммы Σ_p. Подпрограмма Σ_p, как уже говорилось, тоже может содержать команды вызова подпрограмм, включая, в общем случае, и саму себя, т. е. допускается рекурсивный вызов подпрограмм.

Будем рассматривать бинарные программы с ограниченной глубиной рекурсии $r \geq 2$, которые функционируют таким образом, что при превышении данной глубины, т. е. при попытке выполнения команды вызова подпрограммы находясь на уровне r вложенности вызовов подпрограмм, данный вызов не происходит, а выполнение текущей подпрограммы продолжается со следующей команды. Напомним, что случай $r = 1$, который соответствует классу т. н. одномодульных программ, рассмотрен в [3].

Выполнение бинарной программы Σ начинается с первой подпрограммы Σ_1 с некоторыми заданными значениями входных аргументов $x_1, \dots x_n$, $n \le in(1)$. Результатом работы бинарной программы Σ является набор значений, записанных в $out(1)$ выходных ячеек памяти первой вызванной подпрограммы. Будем говорить, что бинарная программа Σ вычисляет булеву функцию $f(x_1, \dots, x_n) \in P_2(n)$, если вычисление Σ для всех 2^n наборов входных аргументов $x_1, \dots, x_n$ завершается и результатом является значение функции $f(x_1, \dots, x_n)$.

Сложностью $\mathcal{L}(\Sigma)$ бинарной программы Σ называется сумма весов всех команд её подпрограмм, где вес вычислительной команды равен весу L_j соответствующей ей булевой функции φ_j, вес переадресующей команды равен 2λ, а вес команды вызова подпрограммы равен ν, где λ, ν — некоторые положительные вещественные константы. Обозначим класс бинарных программ в базисе $Б$ с параметрами λ, ν и ограниченной глубиной рекурсии r как $\mathcal{U}^{\Pi(r)}_{Б,\lambda,\nu} = \mathcal{U}^{\Pi(r)}$. Сложность булевой функции в классе $\mathcal{U}^{\Pi(r)}$, а также соответствующая функция Шеннона $\mathcal{L}^{\Pi(r)}(n)$ для сложности реализации булевых функций из $P_2(n)$ в этом классе определяются обычным образом.

Удельным весом базиса $Б$ назовём константу

$$\pi_Б = \min_{i=\overline{1,b}} \frac{L_i}{r_i + 1}.$$

Основным результатом работы является следующее утверждение.

Теорема. *В случае $\pi_Б \ge 2\lambda$ и $r \ge 2$ справедливо асимптотическое равенство*

$$\mathcal{L}^{\Pi(r)}(n) \sim r\nu^{\frac{r-1}{r}}(r-1)^{-\frac{r-1}{r}}\sqrt[r]{2\lambda}\frac{2^{\frac{n}{r}}}{\sqrt[r]{n}}.$$

Работа выполнена при финансовой поддержке Минобрнауки РФ в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению № 075–15–2019–1621.

Список литературы

[1] Шеннон К. Работы по теории информации и кибернетике // Пер. с англ. — М. : ИЛ, 1963. — 829 с.

[2] Лупанов О. Б. Асимптотические оценки сложности управляющих систем // — М.: Изд-во МГУ, 1984.

[3] Грибок С. В. О реализации функций алгебры логики в некоторых классах программ // Диссертация на соискание ученой степени кандидата физико-математических наук. — МГУ им. М. В. Ломоносова. — 2003.

Количество разметок графов дефинитных автоматов

Ищенко Роман Андреевич

МГУ имени М. В. Ломоносова, e-mail: ishchenko.roman1@gmail.com

Дефинитные автоматы представляют собой один из основополагающих классов автоматов, изучавшихся в том числе в работах [1-4]. Если мы сотрем отметки на ребрах в диаграмме Мура автомата (предположим, информация потеряна), то получим ориентированный граф. В работе [4] был приведен критерий того, что ориентированный граф может быть размечен до диаграммы Мура некоторого дефинитного автомата (восстановление информации), а также приведен алгоритм, который осуществляет эту разметку (в таком случае граф называется *дефинитным*, а разметка — *д-разметкой*). В данном докладе изучается вопрос в каких случаях данное восстановление единственно, и каково максимальное число д-разметок в зависимости от структуры дефинитного графа в случае неединственности восстановления.

Доклад разделен на три части. В первой части доклада рассматривается структура дефинитного графа (в продолжение вопроса структуры дефинитного автомата, изученного в [2]) и показывается, что дефинитный граф единственным образом раскладывается в виде сильно-связной компоненты (ядра G_0) и набора дополнительных слоев. При этом показывается, что любая разметка f дефинитного графа G, такая что ее сужение на ядро G_0 графа G являтся д-разметкой графа G_0, является д-разметкой графа G, т.е. произвольная раскраска слоев не влияет на свойство разметки быть дефинитной. Во второй части показывается, что в случае алфавита из двух элементов разметка сильно-связного дефинитного графа всегда единственная и приводится ряд достаточных условий неединственности разметки в общем случае. В третьей части показывается, что в общем случае максимальное число разметок сильно-связного дефинитного автомата экспоненциально зависит от количества вершин.

Введем необходимые понятия и определения.

Пусть $V = (A, Q, \varphi)$ – конечный автомат без выходов. Автомат V называется *k-дефинитным*, если для любого входного слова α длины k существует такое состояние $q(\alpha) \in Q$, что для любого состояния $q \in Q$ выполняется $\varphi(q, \alpha) = q(\alpha)$. Состояния q_1, q_2 автомата $V = (A, Q, \varphi)$ называются *1-эквивалентными*, если для любого символа $a \in A$ выполняется $\varphi(q_1, a) = \varphi(q_2, a)$. *Сжатым автоматом* $V(q_1, q_2) = (A, Q \setminus \{q_1\}, \varphi_{q_1,q_2})$ будем называть автомат с функцией переходов, определенной следующим

образом:

$$\varphi_{q_1,q_2}(q,a) = \begin{cases} \varphi(q,a), & \text{если } \varphi(q,a) \neq q_1, \\ q_2, & \text{если } \varphi(q,a) = q_1, \end{cases}$$

где $q \in Q \setminus \{q_1\}, a \in A$.

Графом автомата $V = (A, Q, \varphi)$ называется размеченный ориентированный граф $G = (Q, W, f)$, вершины которого соответствуют состояниям автомата, при этом

$$e = (q_i, q_j) \in W,\ f(e) = a \Leftrightarrow \varphi(q_i, a) = q_j,$$

где $f : W \to A$, $a \in A$.

Определим *множество смежности* $\Gamma(q)$ вершины q графа $G = (Q, W)$ как $\Gamma(q) = \{q' \in Q \mid (q, q') \in W\}$. Назовем вершины q_1 и q_2 ориентированного графа $G = (Q, W)$ *псевдоэквивалентными*, если $\Gamma(q_1) = \Gamma(q_2)$. По аналогии со сжатым автоматом, определим *сжатый граф* $G(q_1, q_2) = (Q \setminus \{q_1\}, W(q_1, q_2))$, образованный из графа $G = (Q, W)$ следующим образом: $W(q_1, q_2) = W \setminus \{(q_1, q) \mid q \in Q\} \cup \{(q, q_2) \mid (q, q_1) \in W\}$, вершины q_1 и q_2 — псевдоэквивалентны. Для сохранения преемственности вершину q_2 в графе $G(q_1, q_2)$ будем также обозначать $[q_1, q_2]$.

Разметку $f : W \to A$ ориентированного графа $G = (Q, W)$ такую, что $G = (Q, W, f)$ — граф дефинитного автомата, будем называть *д-разметкой* ориентированного графа G. Граф G, у которого исходящая степень каждой вершины равна 1 и существует натуральное m, что кратность каждого ребра равна m называется *транзитным* графом кратности m.

Структура графов дефинитных автоматов

Теорема 1. *Пусть граф $G = (Q, W)$ имеет д-разметку. Тогда множество вершин Q может быть единственным образом представлено в виде непересекающегося разложения $\{Q_0, Q_1, \ldots, Q_t\}, Q = \bigcup_{i=\{0,\ldots,t\}} Q_i$, с соответствующим множеством ребер $W_i = \{(v, u) \in W | v \in Q_i\}$ и графами $G_i = (Q_i \cup \{u \in Q | \exists$ ребро $(v, u) \in W_i\}, W_i), i \in \{1, \ldots, t\}$, что:*

- *G_0 – сильно-связный граф и для любого ребра (v, u) в графе G_0 выполнено $u \in Q_0$*
- *Для любого ребра (v, u) в графе $G_i, i \in \{1, \ldots, t\}$, выполнено $u \in \bigcup_{j=\{0,\ldots,i-1\}} Q_j$*
- *Для любого числа $i \in \{1, \ldots, t\}$ для любой вершины $v \in Q_i$ в графе G_i существует ребро (v, u), что $u \in Q_{i-1}$*

Таким образом можно говорить о дефинитной структуре $\{G_i\}$ дефинитного графа G.

Теорема 2. *Пусть G — граф с д-разметкой с дефинитной структурой $\{G_i, i = 0, \ldots, t\}$. Пусть f_0 — произвольная д-разметка подграфа G_0 и функция f – произвольная разметка графа G, что $f|G_0 = f_0$. Тогда f — д-разметка графа G.*

Условие единственности разметки дефинитного сильно-связного графа.

Теорема 3. *Если у сильно-связного графа G существует д-разметка в алфавите из двух элементов, то она единственна.*

Теорема 4. *Если сжатый граф $G(p, q)$ имеет неединственную д-разметку, то граф G также имеет неединственную д-разметку.*

Теорема 5. *Пусть в дефинитном графе G существуют вершины v, u и w, что выполнено:*

- *v и u псевдоэквивалетны и принадлежат множеству $\Gamma(w)$*
- *В сжатом графе $G(v, u)$ нет транзитного подграфа G_1, что в G_1 выполнено $\Gamma(w) = [v, u]$ и кратность G_1 равна кратности ребра $(w, [v, u])$ в графе $G(v, u)$.*

Тогда граф G имеет неединственную д-разметку.

Теорема 6. *Пусть в дефинитном графе G существуют такие вершины v, u, q и p, что v и u псевдоэквивалентны и принадлежат множеству $\Gamma(q) \cap \Gamma(p)$. Тогда G имеет неединственную д-разметку.*

Теорема 7. *Если д-разметка f дефинитного графа G такова, что существует пара псевдоэквивалентных вершин, не являющихся 1-эквивалентными в f, то f – не единственная д-разметка графа G.*

Максимальное число разметок дефинитных графов

Теорема 8. *Для любых натуральных чисел m и n существует сильно-связный дефинитный автомат G в алфавите $|A| = m$ с количеством вершин n, у которого не менее $(m-1)!^{(n-m+1)}$ д-разметок*

Список литературы

[1] Perles M., Rabin M., Shamir E. Theory of definite automata //IEEE Trans. Electronic Computers. EC-12 — 1963. — С. 233–243.

[2] Ito M., Duske J. On cofinal and definite automata //Acta Cybernetica — 1983. — Т. 6, № 2. — С. 181–189.

[3] Жук Д. О классификации автоматных базисов Поста по разрешимости свойств А-полноты для дефинитных автоматов //Дискрет. матем — 2010. — Т. 22, № 2. — С. 80–95.

[4] Ищенко Р. О разметке графов дефинитных автоматов // Вестн. Моск. ун-та. Сер. 1. Матем., мех. — 2019. — Т. 74, № 5. — С. 44-48.

Квантовый алгоритм для задачи о самом длинном пути

Капралов Руслан Илнарович[1], **Хадиев Камиль Равилевич**[2]

[1] Казанский федеральный университет, e-mail: kapralov_ruslan@mail.ru

[2] Казанский федеральный университет, Казанский физико-технический институт им. Е.К. Завойского ФИЦ Казанский научный центр РАН, e-mail: kamilhadi@gmail.com

В рамках работы была рассмотрена задача о самом длинном пути во взвешенном графе. Формально задача звучит следующим образом. Дан взвешенный граф $G = (V, E)$, где V — множество вершин, а E — множество ребер. При этом w_e — это вес ребра $e \in E$. Необходимо построить простой путь максимального суммарного веса. Простым путем мы будем называть путь, в котором не может встречаться повторяющихся вершин. Данная задача является NP-трудной [9]. В работе задача исследуется с точки зрения квантовых вычислений, в частности модели запросов [1, 2]. Известно, что существуют задачи, для которых разработаны квантовые алгоритмы, имеющие меньшую запросную сложность, чем классические аналоги [3]. В том числе, квантовые алгоритмы разрабатывались и для NP-полных задач [4].

Для данной задачи был разработан квантовый алгоритм, сложность которого составляет $O^*(1.728^n)$, где $n = |V|$ — число вершин в графе, O^* — не учитывает логарифмический множитель. Алгоритм базируется на алгоритме Гровера [5], его модификации — квантовом алгоритме поиска максимума [6]. Кроме того, алгоритм использует идею квантового алгоритма для метода динамического программирования по подмножествам [4].

Опишем основную концепцию алгоритма. Для этого рассмотрим функцию $F : 2^V \times V \times V \to \mathbb{R}$, где 2^V — это множество всех подмножеств множества V. При этом $F(S, v, u)$ — это длина максимального простого пути, в который можно включать только вершины из множества S, путь должен начинаться с вершины v и заканчиваться вершиной u.

Отметим, что $F(\{v\}, v, v) = 0$ для любой $v \in V$. В то же время для остальных случаев функцию можно вычислить по следующей формуле:

$$F(S, v, u) = \max_{y \in S, y \neq u} \left\{ \max_{y \neq v} F(S \backslash \{y\}, v, u), \max_{(y,u) \in E} \left(F(S \backslash \{u\}, v, y) + w_{(y,u)} \right) \right\},$$

где $v, u \in S, S \subset V, |S| > 1$. В этом случае ответом будет $\max\limits_{v,u \in V} F(V, v, u)$.

Заметим, что у функции F есть следующее свойство:

Свойство 1. *Рассмотрим множество $S \subset V$ и целое число $k \in \{1, \ldots, |S|\}$, тогда верно следующее:*

$$F(S, v, u) = \max_{S' \subset S, |S'| = k, y \in S'} \left(F(S', v, y) + F((S \backslash S') \cup \{y\}, y, u) \right)$$

На основе этого свойства строится следующий алгоритм.

Шаг 1. Выберем $\alpha = 0.055$, тогда посчитаем классически все $F(S, v, u)$ для всех S таких, что $|S| = (1-\alpha)\frac{n}{4}$.

Шаг 2. Пусть $n = |V|$. Рассмотрим следующие функции.

Для $T \subset V$ таких, что $T = \frac{n}{4}$

$$F(T, u, v) = \max_{T' \subset T, |T'| = (1-\alpha)n/4, y \in T'} \left(F(T', v, y) + F((T \backslash T') \cup \{y\}, y, u)\right).$$

Для $S \subset V$ таких, что $S = \frac{n}{2}$

$$F(S, u, v) = \max_{S' \subset S, |S'| = n/4, y \in S'} \left(F(S', v, y) + F((S \backslash S') \cup \{y\}, y, u)\right).$$

А затем,

$$F(V, u, v) = \max_{S' \subset V, |S'| = n/2, y \in S'} \left(F(S', v, y) + F((V \backslash S') \cup \{y\}, y, u)\right).$$

Таким образом, мы можем вычислить $F(V, u, v)$ как рекурсивный запуск трех процедур поиска максимума, используя квантовый алгоритм поиска максимума Дюра-Хоера [6]. Заметим, что ввиду вложенности вызовов, алгоритм Дюра-Хоера должен использовать форму алгоритма Гровера с вероятностным оракулом [7].

Для такого алгоритма справедлива следующая сложность

Теорема 1. *Приведённый квантовый алгоритм решает задачу поиска самого длинного пути. Его сложность* $O^*\left(1.728^n\right)$ *и вероятность ошибки константа не превышающая* $1/3$.

Доказательство. Сложность классического предподсчета составляет

$O^*(C_n^{(1-\alpha)\frac{n}{4}}) = O^*(1.728^n)$.

Сложность трех вложенных запусков квантовых алгоритмов поиска максимума составляет

$$O^*\left(\sqrt{C_n^{n/2} C_{n/2}^{n/4} C_{n/4}^{\alpha n/4}}\right) = O^*(1.728^n).$$

Таким образом, итоговая сложность равна $O^*(1.728^n)$.

Квантовый алгоритм представляет из себя запуск трех вложенных алгоритмов Дюра-Хоера, использующих форму алгоритма Гровера с вероятностным оракулом [7]. Каждый из алгоритмов имеет константную ошибку не превышающую $1/3$. В связи с этим можно доказать, что итоговый алгоритм также имеет константную ошибку не превышающую $1/3$. Для этого можно использовать идею из [4,8]. **Теорема 1 доказана.**

Работа выполнена за счёт средств субсидии, выделенной Казанскому федеральному университету для выполнения государственного задания в сфере научной деятельности, проект 0671-2020-0065.

Список литературы

[1] Ambainis A. Understanding quantum algorithms via query complexity // Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018. — 2018. — С. 3265–3285.

[2] Ablayev F., Ablayev M., Huang J.Z., Khadiev K., Salikhova N., Wu D. On quantum methods for machine learning problems part I: Quantum tools //Big Data Mining and Analytics. — 2019. — Т. 3. — №. 1. — С. 41–55.

[3] Stephen Jordan. Quantum algorithms zoo/http://quantumalgorithmzoo.org/ — 2021.

[4] Ambainis, A., Balodis, K., Iraids, J., Kokainis, M., Prusis, K., Vihrovs, J., Quantum speedups for exponential-time dynamic programming algorithms //Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms. — Society for Industrial and Applied Mathematics, 2019. — С. 1783-1793.

[5] Grover L. K. A fast quantum mechanical algorithm for database search //Proceedings of STOC'96. — 1996. — С. 212–219.

[6] Durr C., Hoyer P. A quantum algorithm for finding the minimum //arXiv preprint quant-ph/9607014. — 1996.

[7] Hoyer P., Mosca M., De Wolf R. Quantum search on bounded-error inputs //International Colloquium on Automata, Languages, and Programming. — Springer, Berlin, Heidelberg, 2003. — С. 291–299.

[8] Ambainis A., Balodis K., Iraids J., Khadiev K., Klevickis V., Prusis K., Shen Y., Smotrovs J., Vihrovs J. Quantum Lower and Upper Bounds for 2D-Grid and Dyck Language//MFCS 2020. — LIPIcs. — 2020.— Т. 170. — С. 8:1–8:14.

[9] Schrijver A. Combinatorial optimization: polyhedra and efficiency. — Springer Science and Business Media, 2003. — Т. 24.

О поиске на произвольном изображении подизображений, аффинно эквивалентных данному

Козлов Вадим Никитович

Московский университет, e-mail: vnkozlov@mail.ru

Изображением (двумерным) называем конечное (непустое) множество точек на плоскости. Считаем, что любую фигуру можно «аппроксимировать» конечным множеством точек, которые уже сами по себе делают фигуру вполне узнаваемой. При этом если точек много, то такая совокупность точек практически неотличима от исходной фигуры. Так же можно представлять и полутоновые, черно-бело-серые изображения, при этом разная плотность точек в разных частях изображения дает разные оттенки «серого цвета».

Перенумеруем некоторым образом точки изображения A так, чтобы номера разных точек были попарно различны. Обозначим через M_A множество этих номеров. Пусть S_{mnu} и S_{kps} — площади треугольников с вершинами в тройках точек с номерами m, n, u и k, p, s и пусть $\rho_{mnu,kps} = S_{mnu}/S_{kps}$. Полагаем, что порядок номеров в тройках не важен, сами тройки различны и при $S_{kps} = 0$ значение $\rho_{mnu,kps}$ не определено. Множество индексированных чисел $\rho_{mnu,kps}$ для всех таких пар троек обозначим через T_A. Код изображения A — пара $< M_A, T_A >$. Изображения A и B с кодами $< M_A, T_A >$ и $< M_B, T_B >$ назовем эквивалентными, если существует такая биекция $\psi\colon M_A \to M_B$, что для любых m, n, u и k, p, s из M_A выполнено $\rho_{mnu,kps} = \rho_{\psi(m)\psi(n)\psi(u),\psi(k)\psi(p)\psi(s)}$. Ясно, что эквивалентность изображений содержательно означает одинаковость их кодов с точностью до перенумерации точек.

Два изображения называем аффинно эквивалентными, если они переводимы друг в друга аффинными преобразованиями. Если все точки изображения не лежат на одной прямой или двух параллельных прямых, то изображение называем плоским.

Теорема 1 [1]. *Два плоских изображения эквивалентны точно тогда, когда они аффинно эквивалентны.*

Содержательно теорема означает, что код изображения задает его с точностью до аффинных преобразований.

Назовем изображения A и B пропорциональными, если существует такая биекция $\psi\colon M_A \to M_B$, при которой для любых точек с номерами m, n, u из M_A (не лежащих на одной прямой), число $\rho_{mnu,\psi(m)\psi(n)\psi(u)}$ есть константа, не зависящая от выбора точек m, n, u.

Теорема 2 [2]. *Два плоских изображения пропорциональны тогда и только тогда, когда они аффинно эквивалентны.*

Изображение B назовем частью изображения A, если множетво точек B является подмножеством множества точек A. Изображение B назовем подизображением изображения A, если B аффинно эквивалентно с некоторой частью в A.

В алгоритмах распознавания, основывающихся на представленных понятиях эквивалентных и пропорциональных изображений, важна и повторяется задача определения того, является ли B подизображением для A. Главным образом именно она — предмет настоящего рассмотрения. Конечно эта задача может быть решена тривиальным перебором, однако это не рационально. Процедура проверки может быть упрощена. Построим изображение по его коду. Выберем на A три точки a, b и c, не лежащие на одной прямой. Затем будем поочередно брать все возможные тройки точек a_0, b_0 и c_0 на B (не лежащие на одной прямой) и считать их совмещенными с точками a, b и c. Если A и B эквивалентны, ψ — биекция из определения эквивалентности, и $a_0 = \psi(a)$, $b_0 = \psi(b)$, $c_0 = \psi(c)$, то при построении по коду и все остальные точки изображения B совместятся с соответствующими точками изображения A. Это и определит полностью биекцию ψ. Число вариантов для проверки при такой процедуре, очевидно, не превышает n^3. Можно еще больше упростить процедуру проверки. Назовем точку x изображения A внутренней, если найдутся такие точки a, b и c из A, отличные от x, что образованный ими треугольник содержит x. Пусть V_A — множество всех таких точек из A. Очевидно, что по коду изображения для каждой его точки можно проверить включение ее в V_A. Точки из A без V_A назовем внешними или контурными, а часть изображения A, образованную этими точками — контуром. Ясно, что биекция ψ должна порождать взаимно однозначное соответствие точек контуров изображений A и B. Затем на изображении, образованном точками из V_A, можно выделить его контур, и т. д. В результате изображение однозначно «расслоится» на контуры: первый, второй и т. д. Далее для изображений A и B нужно сопоставлять друг другу только точки из контуров с одинаковыми номерами.

Полного аффинного совпадения B с частью в A можно и не требовать, только «приблизительного совпадения», с «зазором ϵ», и называть это ϵ-эквивалентностью. Выяснение, является ли некоторое B подизображением (с точностью до ϵ-эквивалентности) некоторого изображения A, выглядит следующим образом. Выбирают на B три точки b_1, b_2, b_3 (произвольные, но не лежащие на одной прямой), ставят им в соответствие три произвольные (но не на одной прямой) точки a_1, a_2, a_3 изображения

A. Берут на B точку x_1, сопоставляют ей некоторую точку y_1 на A. Если теперь элементы кода изображения из точек b_1, b_2, b_3 и x_1 отличаются от соответствующих элементов кода изображения из точек a_1, a_2, a_3 и y_1 на A не более, чем на ϵ, то точку y_1 называют приемлемой при соответствии с точкой x_1. Если точка y_1 оказалась не приемлемой, то берут другую точку из A в качестве точки y_1 и повторяют рассуждение. Затем то же делают для точки x_2, и т. д. Пусть B состоит из точек b_1, b_2, b_3, x_1, …, x_k, и для точек x_1, …, x_k найдены соответствующие приемлемые точки y_1, …, y_k на A. Тогда для изображения B из точек b_1, b_2, b_3, x_1, …, x_k и изображения A' из точек a_1, a_2, a_3, y_1, …, y_k (при указанном соответствии) проверяют в целом различие всех соответствующих элементов их кодов не более, чем на ϵ. При положительном результате проверки подизображение A' изображения A называют искомым. Если первоначально выбранные тройки точек a_1, a_2, a_3 и b_1, b_2, b_3 не дали возможность для каждой из точек x_1, …, x_k подобрать приемлемые точки y_1, …, y_k, то проделывается описанное со всеми тройками точек на A и B и всеми вариантами соответствия между точками в тройках. В результате либо находят искомое изображение, либо делают вывод, что такого изображения нет.

В частном случае при $\epsilon = 0$ процедура с очевидностью превращается в поиск на A части, аффинно эквивалентной с B.

Аналогичные построения с использованием ϵ-эквивалентности можно проделать и для пропорциональных изображений.

Список литературы

[1] Козлов В. Н. Введение в математическую теорию зрительного восприятия. — М. : Издательство Центра прикладных исследований при механико-математическом факультете МГУ, 2007.

[2] Козлов В. Н. Алгоритмы формирования системы взаимосвязанных образов // Интеллектуальные системы. — 2014. — Т. 18, № 2. — С. 99–114.

Хопфовость унитарных и неунитарных полигонов над группами

Кожухов Игорь Борисович[1], **Колесникова Ксения Анатольевна**[2]

[1] Национальный исследовательский университет «МИЭТ», МЦ ФПМ МГУ, e-mail: kozhuhov_i_b@mail.ru

[2] Национальный исследовательский университет «МИЭТ», МЦ ФПМ МГУ, e-mail: ksenya.koless@gmail.com

Пусть заданы некоторая полугруппа S и множество X. Напомним, что X называется полигоном над полугруппой S, если задано отображение

$\cdot : X \times S \to X$ такое, что $x \cdot (st) = (xs) \cdot t$ для любого $x \in X$ и для любых s, $t \in S$ (см. [1]). Полигон над полугруппой интересен тем, что является алгебраической моделью автомата. Полигон X над S называется *хопфовым*, если любой его сюръективный эндоморфизм является автоморфизмом. Хопфовость является условием конечности, поскольку конечные полигоны, очевидно, обладают этим свойством. Для полигонов над полугруппами известен следующий факт: всякий конечно порождённый коммутативный полигон над полугруппой хопфов (см. [2]).

Если S — полугруппа с единицей e и для всех $x \in X$ выполнено $x \cdot e = x$, полигон X над S называют *унитарным*. Мы будем рассматривать случай, когда полугруппа S является группой, и исследовать унитарные и неунитарные полигоны над группой.

Известно, что унитарный циклический полигон X над группой G изоморфен полигону вида G/H (H — подгруппа в G, не обязательно нормальная). Кроме того, произвольный унитарный полигон представим в виде копроизведения циклических унитарных полигонов $\coprod_{i\in I}(G/H_i)$ (см. [3]).

Можно рассмотреть отношение $\preccurlyeq$ на множестве индексов I унитарного полигона $\coprod_{i\in I}(G/H_i)$, полагая $i \preccurlyeq j$, если существует $a \in G$ такое, что $H_i \subseteq a^{-1}H_j a$. Это означает, что $i \preccurlyeq j$, если существует гомоморфизм $G/H_i \to G/H_j$. Можно показать, что отношение $\preccurlyeq$ является квазипорядком.

Нами были найдены необходимые и достаточные условия хопфовости унитарного полигона над группой.

Теорема 1. *Пусть X – унитарный полигон над группой G, $X = \coprod\limits_{i\in I} X_i$, где $X_i \cong G/H_i$. Квазипорядок $\preccurlyeq$ на I имеет тот же смысл, что и выше. Тогда полигон X хопфов, если и только если выполняются следующие условия:*

1. *каждое G/H_i хопфово;*
2. *множество I не содержит последовательностей таких, что*

$$\ldots \preccurlyeq i_{-n} \preccurlyeq \ldots \preccurlyeq i_{-1} \preccurlyeq i_0,$$

где все i_k различны.

Строение неунитарных полигонов над группой было выяснено в [3]. Всякий полигон X над группой G состоит из унитарной части $Y = Xe$ и множества $A = X \setminus Xe$ с заданным отображением $\mu : A \to X$, $\mu(a) = ae$. Результат действия элемента $g \in G$ на элемент $x \in X$ определяется следующим образом:

$$x \cdot g = \begin{cases} xg, & \text{если } x \in Y, \\ \mu(x)g, & \text{если } x \in A. \end{cases}$$

Для полигона с циклической унитарной частью можно доказать следующие теоремы.

Теорема 2. *Полигон $X = (G/H) \cup A$ над группой G хопфов в том и только том случае, если выполнены условия:*

1. $\left|\mu^{-1}(Hg)\right| < \infty$ *при всех* $g \in G$;
2. *для любого* $a \in G$ *такого, что* $a^{-1}Ha \supset H$, *для некоторого* $g \in G$ *выполнено неравенство*

$$\sum_{i \in I} \left|\mu^{-1}(Ha^{-1}g_i g)\right| < \left|\mu^{-1}(Hg)\right|,$$

   где $H = \bigcup_{i \in I} aHa^{-1}g_i$ — *разложение группы* H *в правые смежные классы по подгруппе* aHa^{-1};
3. *для любого* $a \in G$ *такого, что* $a^{-1}Ha = H$, *верно неравенство*

$$\left|\mu^{-1}(Ha^{-1}g)\right| \geqslant \left|\mu^{-1}(Hg)\right|$$

   для всех $g \in G$, *и верно неравенство*

$$\left|\mu^{-1}(Ha^{-1}g)\right| > \left|\mu^{-1}(Hg)\right|$$

   при некотором $g \in G$.

Теорема 3. *Конечно порождённый полигон $X = X_1 \sqcup \ldots \sqcup X_n$ над группой G хопфов в том и только том случае, если X_i хопфов для каждого $i \in \{1, \ldots, n\}$.*

Список литературы

[1] Kilp M., Knauer U., Mikhalev A.V. Monoids, acts and categories // Berlin – New York: W. de Gruyter — 2000.

[2] Карташов В. К. Независимые системы порождающих и свойство Хопфа для унарных алгебр. Дискретная матем., 2008, т. 20, вып. 4, с. 79-84.

[3] Максимовский М. Ю. О биполигонах и мультиполигонах над полугруппами. Мат. заметки, 2010, т.87, №6, с.855-866.

О некоторых аспектах развития обобщенной схемы размещения с примерами применения

Колчин Андрей Валентинович, Безродный Борис Федорович, Леева Марина Александровна

Московский автомобильно-дорожный государственный технический университет (МАДИ), e-mail: math@madi.ru

Комбинаторика сыграла существенную роль в начале развития теории вероятностей, и эти два раздела математики продолжают развиваться в тесном взаимодействии. В настоящее время теория вероятностей, предлагая новые подходы к решению задач дискретной математики, как бы отдает долги комбинаторике. Среди этих новых подходов упомянем хорошо развитые в теории вероятностей методы асимптотического анализа. Комбинаторные задачи и методы занимают значительное место в исследованиях по теории вероятностей. В этой области можно выделить несколько направлений: комбинаторные задачи в теории случайных процессов; задачи, связанные со случайными отображениями и случайными графами; задачи размещения частиц по ячейкам.

Для решения широкого круга комбинаторных задач весьма плодотворным оказывается *вероятностный подход* [1]. Если на множестве рассматриваемых комбинаторных структур задать распределение вероятностей, то числовые характеристики этих структур можно рассматривать как случайные величины и анализировать их вероятностными методами.

Среди тех, трудами которых развивалась вероятностная комбинаторика в России, были В. Л. Гончаров, С. Н. Бернштейн, Н. В. Смирнов, В. Е. Степанов, ее успехи тесно связаны с блестящей Российской вероятностной школой, школой А. А. Маркова, П. Л. Чебышёва, А. М. Ляпунова, А. Я. Хинчина, А. Н. Колмогорова, Ю. В. Прохорова.

В вероятностной комбинаторике находит успешное применение *обобщенная схема размещения*, позволяющая сводить ряд комбинаторных задач к задачам о суммах независимых случайных величин, классическому объекту изучения в теории вероятностей (см., например, [4–5]). Она оказалась удобным средством исследования таких интереснейших объектов, как случайные графы, случайные леса, системы линейных уравнений со случайными коэффициентами, случайные подстановки, в том числе в связи с построением и анализом вычислительных алгоритмов. В настоящее время активные исследования асимптотического поведения различных комбинаторных объектов с использованием обобщенной схемы размещения ведутся, в частности, Ю. Л. Павловым в Карельском научном центре

РАН, А. Н. Чупруновым вначале в Казанском федеральном, а затем в Чувашском государственном университете, и И. Фазекашем в Дебреценском университете (Венгрия).

Напомним, что в обобщенной схеме размещения частиц распределение заполнений ячеек представимо как условное распределение *независимых* случайных величин при условии, что их сумма принимает фиксированное значение. Пусть $\eta_1, \dots, \eta_N$ — неотрицательные целочисленные случайные величины, рассматриваемые как некоторые числовые характеристики комбинаторной структуры из N компонент, состоящей из n элементов, такие, что $\eta_1 + \cdots + \eta_N = n$. Если существуют независимые случайные величины $\xi_1, \dots, \xi_N$ такие, что совместное распределение $\eta_1, \dots, \eta_N$ допускает представление

$$\mathsf{P}\{\eta_1 = k_1, \dots, \eta_N = k_N\} = \mathsf{P}\{\xi_1 = k_1, \dots, \xi_N = k_N \mid \xi_1 + \cdots + \xi_N = n\},$$

где $k_1, \dots, k_N$ — произвольные целые числа, то говорят, что $\eta_1, \dots, \eta_N$ образуют обобщенную схему размещения с параметрами n и N и независимыми случайными величинами $\xi_1, \dots, \xi_N$. Случайные величины $\eta_1, \dots, \eta_N$ интерпретируются как заполнения ячеек.

А. Н. Чупруновым активно исследуются обобщенные схемы размещения *случайного* числа частиц по N ячейкам. Ее частными случаями являются обобщенная схема размещения с неполным комплектом частиц, а также некоторые другие аналоги обобщенной схемы размещения.

И. Фазекашем изучаются расширения обобщенных схем размещения, где в N ячеек размещаются либо *по крайней мере* n частиц, либо *не более* n частиц. Также им предложено расширение обобщенной схемы размещения, где ячейки берутся из некоторого множества.

Изучение многих характеристик обобщенной схемы размещения сводится к задачам о суммах независимых случайных величин. В большинстве применений возникает необходимость в *локальных предельных теоремах в схеме серий.*

Как правило, распределение $\xi_1, \dots, \xi_N$ допускает представление

$$\mathsf{P}\{\xi_1 = k\} = \frac{b_k \theta^k}{k!\, B(\theta)},$$

где $b_0, b_1, b_2, \dots$ — некоторая последовательность неотрицательных чисел, $B(\theta) = \sum_{k=0}^{\infty} b_k \theta^k / k!$, и θ — параметр, принимающий положительные значения из области сходимости ряда $B(\theta)$. Для изучения характеристик обобщенной схемы размещения требуются локальные предельные теоремы при всех значениях параметра θ (см. [2–5]). В простейшем случае значения θ от-

делены от 0 и не приближаются к значению радиуса сходимости $B(\theta)$. Отметим феномен *перехода* распределений сумм случайных величин с одной решетки на другую в контексте обобщенной схемы размещения: именно, при $N \to \infty$ и различных "промежуточных" скоростях стремления к нулю параметра θ распределение суммы $S_N = \sum_{k=1}^{N} \xi_k$ переходит с решетки целых чисел, где имеет место сходимость к нормальному распределению, на решетку целых неотрицательных чисел с некоторым шагом r, где имеет место сходимость уже к распределению Пуассона. В случаях, когда значения параметра θ приближаются к границе сходимости ряда $B(\theta)$, могут появляться и другие предельные распределения, как можно показать на примерах.

В иллюстративных целях, упомянем несколько простых, но интересных с методической точки зрения примеров сведения комбинаторных задач к обобщенным схемам размещения частиц по ячейкам (см., например, [5]). Так, обобщенную схему размещения частиц по ячейкам можно использовать для изучения размеров деревьев в случайном лесе из корневых деревьев. Также оказывается возможным изучать поведение числа разбиений целого положительного числа n на N целочисленных слагаемых, не превосходящих некоторого $r \geq 0$. Еще одним примером является изучение поведения случайных подстановок и уравнений в подстановках. Наконец, изучение различных характеристик разнообразных вариантов урновых схем естественным образом сводится к применению обобщенной схемы размещения.

Список литературы

[1] Erdős P., Spencer J. H. Probabilistic Methods in Combinatorics. — New York: Academic Press, 1974.

[2] Колчин А. В. Предельные теоремы для обобщенной схемы размещения // Дискретная математика. — 2003. — Т. 15, № 4. — С. 148–157.

[3] Колчин А. В. Предельные теоремы в обобщенной схеме размещения // Обозрение прикладной и промышленной математики. — 2009. — Т. 16, № 3. — С. 432–435.

[4] Колчин А. В., Безродный Б. Ф., Леева М. А. Обобщенная схема размещения: некоторые аспекты развития // Обозрение прикладной и промышленной математики. — 2018. — Т. 25, № 2. — С. 97–102.

[5] Колчин А. В., Безродный Б. Ф. О некоторых аспектах развития обобщенной схемы размещения // Труды Карельского научного центра Российской академии наук. — 2019. — № 7. — С. 21–29.

О сложности реализации систем одночленов двух переменных схемами композиции

Корнеев Сергей Александрович
Московский государственный университет им. М. В. Ломоносова, e-mail: korneev.sa.42@gmail.com

Исследуется задача о сложности вычисления системы мономов схемами композиции, т. е. задача о нахождении величины

$$l_{sh}(\{{x_1}^{a_{11}}{x_2}^{a_{12}}\dots{x_q}^{a_{1q}}, {x_1}^{a_{21}}{x_2}^{a_{22}}\dots{x_q}^{a_{2q}}, \dots, {x_1}^{a_{p1}}{x_2}^{a_{p2}}\dots{x_q}^{a_{pq}}\}) \text{ —}$$

минимального числа операций композиции, достаточного для вычисления системы мономов по переменным (при этом допускается многократное использование промежуточных результатов). Операция композиции была предложена А. И. Ширшовым [1] как обобщение операции умножения. Схемы композиции исследовались в работах [2–5]. Так как система из p мономов от q переменных однозначно задаётся целочисленной матрицей A размера $p \times q$, бывает удобно использовать термин «сложность реализации матрицы» вместо «сложность реализации системы мономов» и соответствующее обозначение $l_{sh}(A)$.

Для реализации матрицы размера $2 \times q$ в работе [4] установлено точное значение сложности, а также показано, что эта сложность, как и для классической модели [6], определяется некоторой подматрицей размера 2×2. Аналогичное утверждение для сложности реализации матриц размера $p \times 2$ в классической модели справедливо в силу принципа двойственности (см., например, [7]). Однако для схем композиции соображения двойственности не работают в достаточной степени [3]. Следующий пример демонстрирует, что при фиксированном p даже удаление одного произвольного монома может изменить асимптотику роста сложности реализации матрицы размера $p \times 2$ схемами композиции, и следовательно, эта асимптотика не только не определяется подматрицей размера 2×2, но даже, вообще говоря, не определяется никакой подматрицей размера $(p-1) \times 2$.

Пусть p — чётное число (это непринципиально, пример легко обобщается на случай нечётного p). Рассмотрим матрицу

$$A = \begin{pmatrix} 2^{2n} & 1 \\ 2^{2n} & 2^{3n} \\ 2^{4n} & 2^{3n} \\ 2^{4n} & 2^{5n} \\ \dots & \dots \\ 2^{pn} & 2^{(p-1)n} \\ 2^{pn} & 2^{(p+1)n} \end{pmatrix}.$$

Обозначим через $A^{(k)}$ матрицу, полученную из матрицы A удалением k-й строки. Тогда

$$l_{sh}(A) = (2p+1)n + 1, \quad l_{sh}(A^{(k)}) = (2p-1)n + 1.$$

Рассмотрим последовательности матриц A_n и $A_n^{(k_n)}$, которые получаются из матриц A и $A^{(k)}$, соответственно, при $n = 1, 2, 3, \dots$ и произвольных $k_n \in \{1, 2, \dots, p\}$. Тогда

$$\lim_{n\to\infty} \frac{l_{sh}(A_n)}{l_{sh}(A_n^{(k_n)})} = 1 + \frac{1}{p - \frac{1}{2}},$$

т. е. асимптотика роста сложности последовательности матриц $A_n^{(k_n)}$ отличается от асимптотики роста сложности последовательности матриц A_n.

Пусть $a^+ = \max(a, 1)$ для любого числа a. Будем считать максимум по пустому множеству чисел равным единице. Для матрицы $A = (a_{ij})$ размера $p \times 2$ положим

$$G(A) = \max_{(i_1,\dots,i_p)\in S_p} \sum_{k=1}^{p} \left\lceil \log_2 \max \left(\frac{a^+_{i_k 1}}{\max\limits_{l:l<k} a^+_{i_l 1}}, \frac{a^+_{i_k 2}}{\max\limits_{l:l<k} a^+_{i_l 2}}, 1 \right) \right\rceil.$$

Следующая теорема с точностью до слагаемого порядка p устанавливает сложность реализации матрицы размера $p \times 2$ схемами композиции.

Теорема. *Пусть в матрице $A = (a_{ij})$ размера $p \times 2$ нет нулевых строк и столбцов, и все её элементы — целые неотрицательные числа. Тогда*

$$G(A) \le l_{sh}(A) \le G(A) + 2p - 3.$$

Список литературы

[1] *Ширшов А. И.* Некоторые алгоритмические проблемы для алгебр Ли // Сиб. матем. журнал. — 1962. — Т. 3. — С. 292–296.

[2] *Мерекин Ю. В.* О порождении слов с использованием операции композиции // Дискретн. анализ и исслед. опер. Сер. 1 — 2003. — Т. 10, № 4. — С. 70–78.

[3] *Трусевич Е. Н.* О сложности вычисления некоторых систем одночленов схемами композиции // Вестник московского университета. Сер. 1. Математика. Механика. — 2014. — № 5. — С. 18–22.

[4] *Корнеев С. А.* О сложности реализации системы из двух мономов схемами композиции // Дискретная математика. — 2020. — Т. 32, № 2. — С. 15–31.

[5] *Корнеев С. А.* Об асимптотическом поведении функции шенноновского типа, характеризующих сложность вычисления систем мономов // Учёные записки Казанского университета. Серия Физико-математические науки. — 2020. — Т. 162, № 3. — С. 300–311.

[6] *Кочергин В. В.* О сложности вычисления систем одночленов от двух переменных // Труды VII Международной конференции «Дискретные модели в теории управляющих систем» (Покровское, 4-6 марта 2006 г). — 2006. — С. 185–190.

[7] *Сидоренко А. Ф.* Сложность аддитивных вычислений семейства целочисленных линейных форм // Записки научных семинаров ЛОМИ. — 1981. —Т. 105. — С. 53–61.

Об оценке констант при задании классов функций, определяемых кусочно-линейной мажорантой

Коротченко Анатолий Григорьевич[1], **Сморякова Валентина Михайловна**[2]

[1] Нижегородский государственный университет им. Н.И. Лобачевского, e-mail: koangr@yandex.ru
[2] Нижегородский государственный университет им. Н.И. Лобачевского, e-mail: smorykov@mail.ru

Рассматриваются классы функций, имеющие следующую структуру: они содержат вогнутые, выпуклые, удовлетворяющие условию Липшица функции и замкнуты относительно ряда естественных операций, таких, например, как операции взятия минимума, максимума, суммирования с неотрицательными коэффициентами по конечному набору функций. Выше упомянутые классы функций могут возникать, например, при решении

многокритериальных задач [1]. Для указанных классов функций одной переменной построены алгоритмы поиска экстремума с учетом таких критериев оценки их эффективности как простота в реализации и точность в отыскании наибольшего значения функции [2-4]. Данные алгоритмы могут использоваться при решении более сложных задач.

Будем говорить, что непрерывная функция $f(x)$, определенная на отрезке $[a, b]$, принадлежит классу функций $F_1(a, b, K_1, K_2)$, если выполняются следующие соотношения:

$$\frac{f(x_2) - K_1}{x_2 - a} \leq \frac{f(x_1) - K_1}{x_1 - a}, \frac{f(x_2) - K_2}{b - x_2} \geq \frac{f(x_1) - K_2}{b - x_1}, \tag{1}$$

где $x_2 > x_1$, $x_1, x_2 \in (a, b)$, $K_1 > K_2$.

Непрерывную функцию $f(x)$, определенную на отрезке $[a, b]$, будем относить к классу функций $F_2(a, b, K_1, K_2)$, если соотношения (1) выполняются при $K_2 > K_1$.

Непрерывную функцию $f(x)$, определенную на отрезке $[a, b]$, будем относить к классу функций $F(a, b, K)$, если соотношения (1) выполняются при $K = K_2 = K_1$.

Заметим здесь, что в [2-4] описываются некоторые способы задания констант K_1, K_2, K с помощью которых функции, образованные из вогнутых, выпуклых с помощью операций взятия минимума, максимума по их конечному набору, можно отнести к соответствующим классам $F_1(a, b, K_1, K_2)$, $F_2(a, b, K_1, K_2)$, $F(a, b, K)$.

Для упрощения изложения, будем рассматривать класс функций $F(a, b, K)$. Отметим, что если $f(x) \in F(a, b, K)$, то $f(x) \geq K$, и если $f(x) \in F(a, b, K')$, то $f(x) \geq F(a, b, K)$, $K \leq K'$.

Если значение константы K не удаётся установить априори, то для задания её начального значения используем следующую процедуру. Вычислим функцию $f(x)$ в точках a, b, $\widetilde{x_1}$, $\widetilde{x_2}$, $(a < \widetilde{x_1} < \widetilde{x_2} < b)$ и определим величины:

$$\widetilde{K}_{1i} = f(b) + \frac{f(\widetilde{x_i}) - f(b)}{b - \widetilde{x_i}}(b - a), \widetilde{K}_{2i} = f(a) + \frac{f(\widetilde{x_i}) - f(a)}{\widetilde{x_i} - a}(b - a), i = 1, 2,$$

$$\widetilde{K}_{13} = f(\widetilde{x_1}) + \frac{f(\widetilde{x_2}) - f(\widetilde{x_1})}{\widetilde{x_2} - \widetilde{x_1})}(a - \widetilde{x_1}), \widetilde{K}_{23} = f(\widetilde{x_2}) + \frac{f(\widetilde{x_1}) - f(\widetilde{x_2})}{\widetilde{x_1} - \widetilde{x_2})}(b - \widetilde{x_2}).$$

Пусть $\widetilde{K}_1 = \min\limits_{i=1,2,3} \widetilde{K}_{1i}$, $\widetilde{K}_2 = \min\limits_{i=1,2,3} \widetilde{K}_{2i}$, $\widetilde{K} = min\{f(a), f(b), \widetilde{K}_1, \widetilde{K}_2\}$,

$$K = \widetilde{K} - \Delta, \Delta \geq 0. \tag{2}$$

Такой выбор начального значения K обеспечивает выполнение соотношений (1), для точек $\widetilde{x_1}$, $\widetilde{x_2}$.

Предположим, сначала, что $f(x) \in F(a,b,K)$ с константой K, определенной в (2). Пусть функция $f(x)$ вычислена в N точках x_i и $y_i = f(x_i)$, $i = 1,\ldots,N$ $(a < x_1 < \ldots < x_N < b)$. При этом используется алгоритм α, описанный в [2-4]. В качестве первых двух точек здесь можно выбрать точки $\widetilde{x}_1$, $\widetilde{x}_2$. В алгоритме α используется точная верхняя мажоранта класса $F(a,b,K,N)$. Под классов $F(a,b,K,N)$ здесь понимается подкласс всех таких функций, принадлежащих $F(a,b,K)$ и принимающих в точках x_i значения y_i, $i = 1,\ldots,N$.

Пусть $\phi_{1i}(x) = \frac{(y_i-K)}{(x_i-a)}(x-a)+K$, $\phi_{2i}(x) = \frac{(y_i-K)}{(b-x_i)}(b-x)+K$, $\phi_i(x) = max\{\phi_{1i}(x), \phi_{2i}(x)\}$, $i = 1,\ldots,N$, а $\Psi_N(x) = \min\limits_{i=1,\ldots,N} \phi_i(x)$, $x \in [a,b]$. Кусочно-линейная функция $\Psi_N(x)$ является точной верхней мажорантой класса $F(a,b,K,N)$. Пусть $h = \max\limits_{i=1,\ldots,N} y_i$ и $\overline{x} = \operatorname{argmax}\limits_{x\in[a,b]} \Psi_N(x)$, где $x_{p-1} \le \overline{x} \le x_p$, $1 \le p \le N+1$. Здесь $x_0 = a, x_{N+1} = b$. Пусть $[u,v]$, где $x_{p_1} \le u < v < x_p$, такой отрезок, что $\Psi_N(x) \ge h$, $x \in [u,v]$. Тогда в алгоритме α точка очередного вычисления $f(x)$ есть $\widetilde{x} = \frac{u+v}{2}$. Свойства алгоритма α и его связь с оптимальным одношаговым алгоритмом поиска наибольшего значения функций из класса $F(a,b,K)$ описаны в [2-4].

Если константа K априори не установлена, то в процессе работы алгоритма α может возникнуть необходимость в её уточнении.

Рассмотрим эвристическую процедуру оценки этой константы. Пусть функция $f(x)$ вычислена в точке $\widetilde{x}$ и $\widetilde{y} = f(\widetilde{x})$. Возможны следующие случаи.

Случай 1. Если $\widetilde{y} \le \Psi_N(\widetilde{x})$, то строится точная верхняя мажоранта класса $F(a,b,K,N+1)$ и, как и выше, находится отрезок для следующего вычисления $f(x)$ по алгоритму α.

Случай 2. В случае, когда $\widetilde{y} > \Psi_N(\widetilde{x})$, вычислим $K_1' = y_{p-1} + \frac{\widetilde{y}-y_{p-1}}{\widetilde{x}-x_{p-1}}(a-x_{p-1})$, $K_2' = y_p + \frac{\widetilde{y}-y_p}{\widetilde{x}-x_p}(b-x_{p_1})$,

$$K = min\{K_1', K_2'\} - \Delta, \Delta > 0. \tag{3}$$

Далее, точная верхняя мажоранта класса $F(a,b,K,N)$ перестраивается для нового значения K, см. (3), и строится точная верхняя мажоранта для класса $F(a,b,K,N+1)$ с этим новым значением K. После этого поступаем как и в случае 1.

Случай 3. Отметим, что в силу (1), если $f(x) \in F(a,b,K)$ и $\widetilde{f}(x) = max\{f(x), h\}$, то $\widetilde{f}(x) \in F(a,b,K)$. Тогда, если $f(\widetilde{x}) < h$, то имеются две возможности продолжения вычислений. Первая из них состоит в том, что поступаем также, как в случае 1, а вторая - в том, что полагаем $f(\widetilde{x}) = h$ и уже далее поступаем как в случае 1.

Завершение вычислений по алгоритму α в соответствии с описанной процедурой оценки K может производится либо по исчерпанию вычислительного ресурса, либо, если начиная с какого-то шага получаем константу K, для которой $f(x) \in F(a, b, K)$, то по достижению заданной точности в отыскании наибольшего значения функции.

Отметим, что описанная процедура оценки K может дополняться, если реализуется случай 2, процедурой перехода к подотрезкам, на которых определяются свои константы, аналогичные K. Эвристические процедуры, подобные описанной выше могут быть построены и для классов $F_1(a, b, K_1, K_2)$, $F_2(a, b, K_1, K_2)$.

Список литературы

[1] Коротченко А. Г., Кумагина Е. А., Сморякова В. М., Введение в многокритериальную оптимизацию. //Нижний Новгород. —2017.

[2] Коротченко А. Г., Об одном алгоритме поиска наибольшего значения одномерных функций //Журнал вычислительной математики и математической физики. — 1978. — Т. 18, № 3. — С. 563–573.

[3] Коротченко А. Г., Сморякова В. М., Об оценке погрешности алгоритмов поиска экстремума в классах функций, определяемых кусочно-линейной мажорантой //Вестник Нижегородского университета им. Н.И. Лобачевского. — 2013. № 3-1. С. 188-194.

[4] Коротченко А. Г., Сморякова В. М., Об одном алгоритме поиска максимума в классе функций, определяемом кусочно-линейной мажорантой //Вестник Нижегородского университета им. Н.И. Лобачевского. — 2014. № 4-1. С. 409-415.

Уточненные оценки немонотонной сложности функций многозначной логики

Кочергин Вадим Васильевич[1], **Михайлович Анна Витальевна**[2]

[1] МГУ имени М. В. Ломоносова, e-mail: vvkoch@yandex.ru

[2] НИУ ВШЭ, e-mail: anna@mikhaylovich.com

Пусть P_k — множество всех функций k-значной логики ($k \geq 2$), M — класс всех функций из P_k, монотонных относительно порядка

$$0 < 1 < \ldots < k - 1.$$

Исследуется сложность реализации функций k-значной логики схемами из функциональных элементов над базисами B, имеющими вид:

$$B = M \cup \{\omega_1, \ldots, \omega_p\}, \quad \omega_i \in P_k \setminus M,\ i = 1, \ldots, p, \qquad (*)$$

причем функциям из множества M приписан нулевой вес, а функциям $\omega_1, \ldots, \omega_p$ — единичный.

Определим *немонотонную сложность* $I_B(S)$ *схемы* S *над базисом* B как число немонотонных элементов схемы S.

Немонотонную сложность над базисом B функции k-значной логики f, обозначаемую $I_B(f)$, определим как минимальную немонотонную сложность схем, вычисляющих над базисом B функцию f.

Обозначим множество $\{0, 1, \ldots, k-1\}$ через E_k. Последовательность

$$\tilde{\alpha}_1 = (\alpha_{11}, \ldots, \alpha_{1n}),\ \tilde{\alpha}_2 = (\alpha_{21}, \ldots, \alpha_{2n}),\ \ldots,\ \tilde{\alpha}_r = (\alpha_{r1}, \ldots, \alpha_{rn})$$

наборов из множества E_k^n назовем *возрастающей цепью относительно порядка* $0 < 1 < \ldots < k-1$ или просто *цепью*, если все наборы $\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_r$ различны и выполняются неравенства

$$\alpha_{ij} \le \alpha_{i+1,j}, \quad i = 1, \ldots, r-1, \quad j = 1, \ldots, n.$$

Пусть $f(x_1, \ldots, x_n)$ — функция k-значной логики. Упорядоченную пару наборов $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ и $\tilde{\beta} = (\beta_1, \ldots, \beta_n)$, $\tilde{\alpha}, \tilde{\beta} \in E_k^n$, будем называть *обрывом для функции* f, если выполнены условия:

1) $\alpha_j \le \beta_j,\ j = 1, \ldots, n$;

2) $f(\tilde{\alpha}) > f(\tilde{\beta})$.

Пусть $F = \{f_1, \ldots, f_m\}$, $m \ge 1$, — система функций k-значной логики от переменных $x_1, \ldots x_n$, а C — цепь, имеющая вид $\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_r$. Под *падением* $d_C(f)$ *функции* f *на цепи* C, имеющей вид $\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_r$, будем понимать число обрывов функции f на парах вида $(\tilde{\alpha}_i, \tilde{\alpha}_{i+1})$.

Спад $d(f)$ функции f определим равенством $d(f) = \max d_C(f)$, где максимум берется по всем цепям C.

Для произвольной функции k-значной логики $f(x_1, x_2, \ldots, x_n)$ и произвольной цепи $C = (\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_r)$ наборов из E_k^n определим величину $u_C(f)$ как наибольшую длину t подпоследовательности $\tilde{\beta}_1, \tilde{\beta}_2, \ldots, \tilde{\beta}_t$ последовательности C, удовлетворяющей условию $f(\tilde{\beta}_1) > f(\tilde{\beta}_2) > \ldots > f(\tilde{\beta}_t)$.

Определим *инверсионную силу* $u(f)$ *функции* f равенством

$$u(f) = \max u_C(f),$$

где максимум берется по всем цепям C наборов из E_k^n.

Для базиса B вида $(*)$ положим $u(B) = \max u(f)$, где максимум берется по всем функциям f из базиса B.

Исчерпывающее описание немонотонной сложности булевых функций в базисе $B_0 = M \cup \{\overline{x}\}$ (т. е. инверсионной сложности) было получено

А. А. Марковым [1]: для любой булевой функции f установлено равенство

$$I_{B_0}(f) = \lceil \log_2(d(f)+1) \rceil .$$

В работе [2] для произвольной функции k-значной логики установлено точное значение немонотонной сложности над двумя естественными базисами $B_P = M \cup \{N_P(x)\}$ и $B_L = M \cup \{N_L(x)\}$, где $N_P(x)$ — отрицание Поста, т. е. функция $x+1 \pmod k$, а $N_L(x)$ — отрицание Лукасевича, т. е. функции $k-1-x$:

$$I_{B_P}(f) = \lceil \log_2(d(f)+1) \rceil , \qquad I_{B_L}(f) = \lceil \log_k(d(f)+1) \rceil .$$

В случае произвольного базиса B вида $(*)$ из результатов работ [2, 3] следует, что найдется такая константа $c(B)$, что для любой функции k-значной логики f выполняются неравенства

$$\left\lceil \log_{u(B)}(d(f)+1) \right\rceil - c(B) \le I_B(f) \le \left\lceil \log_{u(B)}(d(F)+1) \right\rceil .$$

Однако эти оценки далеки от окончательных, так как константа $c(B)$ может оказаться соль угодно большой: для любого заданного значения N найдется базис B_N вида $M \cup \{h_N\}$ и функция g_N, для которых справедливо неравенство

$$\left\lceil \log_{u(B)}(d(g_N)+1) \right\rceil - I_{B_N}(g_N) > N.$$

В булевом случае удалось получить окончательный результат [4]: для любой булевой функции f и любого базиса B, имеющего вид

$$B = M \cup \{\omega_1, \ldots, \omega_p\}, \quad \omega_i \in P_2 \setminus M,\ i = 1, \ldots, p,$$

справедливо равенство

$$I_B(f) = \left\lceil \log_2 \left(\frac{d(f)}{D(B)} + 1 \right) \right\rceil ,$$

где $D(B) = \max\{d(\omega_1), \ldots, d(\omega_p)\}$.

Для случая реализации функций k-значной логики не удалось установить точное значение немонотонной сложности, однако получены верхняя и нижняя оценка, отличающиеся на константу, не зависящую от базиса.

Теорема. *Для любой функции k-значной логики f и для произвольного базиса B вида*

$$B = M \cup \{\omega_1, \ldots, \omega_p\}, \quad \omega_i \in P_k \setminus M,\ i = 1, \ldots, p,$$

выполняются неравенства

$$\left\lceil \log_{u(B)} \left(\frac{d(f)}{D(B)} + 1 \right) \right\rceil - (\log_2 k + 2) \le I_B(f) \le \left\lceil \log_{u(B)} \left(\frac{d(f)}{D(B)} + 1 \right) \right\rceil + k^2,$$

где $D(B) = \max\{d(\omega_1), \ldots, d(\omega_p)\}$.

Список литературы

[1] Марков А. А. Об инверсионной сложности систем функций // ДАН СССР. — 1957. — Т. 116, N.º 6. — С. 917–919.

[2] Кочергин В. В., Михайлович А. В. О минимальном числе отрицаний при реализации систем функций k-значной логики // Дискретная математика. — 2016. — Т. 28, вып. 4. — С. 80–90.

[3] Kochergin V. V., Mikhailovich A. V. Asymptotics of growth for non-monotone complexity of multi-valued logic function systems // Siberian Electronic Mathematical Reports (`http://semr.math.nsc.ru`). — 2017. — Т. 14. — С. 1100–1107.

[4] Кочергин В. В., Михайлович А. В. Точное значение немонотонной сложности булевых функций // Математические заметки. — 2019. — Т. 105, вып. 1. — С. 32–41.

Совмещенные комбинаторные блок-схемы и их применение

Кочетова Наталья Петровна[1], **Темников Дмитрий Юрьевич**[2], **Фролов Александр Борисович**[3]

[1] НИУ МЭИ Кафедра МиКМ, e-mail: `inatashka99@yandex.ru`
[2] НИУ МЭИ Кафедра МиКМ, e-mail: `dnstnt@mail.ru`
[3] НИУ МЭИ Кафедра МиКМ, e-mail: `abfrolov@mail.ru`

Рассматриваемые в настоящей работе комбинаторные схемы образуются совмещением определенной комбинаторной блок-схемы и соответствующей двойственной блок-схемы. Комбинаторная схема [1] состоит из b блоков, составляющих множество B и содержащих по k элементов из множества V, $|V| = v$. Имеется множество D дуальных блоков, находящееся во взаимно однозначном соответствии с множеством V. Дуальный блок состоит из блоков, содержащих элемент, соответствующий этому дуальному блоку. Блоки комбинаторной блок-схемы являются элементами двойственной блок-схемы, а дуальные блоки – ее блоками.

Элементы множеств V, B и D пронумерованы начальными неотрицательными числами. В числовом виде блок представляется как множество

номеров его элементов, дуальный блок рассматривается как совокупность номеров блоков, содержащих его номер (то есть номер соответствующего ему элемента). Состав блока вычисляется по его номеру [2]. Совмещенная комбинаторная блок-схема состоит из элементов, представляемых парой чисел (<номер блока>,<номер дуального блока>) и составляющих множество идентификаторов элементов V_c, $|V_c| = bk$.

Ее блоки $A_i = \{(j, s), s$ - номер элемента j-го блока комбинаторной схемы (т.е. номер дуального блока, содержащего этот элемент)$\}$.

Ее дуальные блоки $C_i = \{(g, i), g =$ элемент дуального блока комбинаторной схемы (т.е. номер блока, содержащего $i)\}$.

Совмещенные комбинаторные блок-схемы удобно использовать для компьютерного моделирования различного рода сетей. Удалением той или иной компоненты идентификатора элемента отражаются возможные исключения отдельных элементов или связей между элементами.

Пример 1. Совмещенная проективная геометрия CPG(2,n) получается совмещением проективной геометрии PG(2,n) [1] и двойственной проективной геометрии DPG(2,n). Она является моделью компьютерной сети из $n^3 + 2n^2 + 2n + 1$ компьютеров. Сеть построена из $b = n^2 + n + 1$ узлов, каждый из которых соответсвует блоку PG(2,n) и представляет собой полносвязную сеть из $k = n + 1$ компьютеров. Каждый компьютер при этом входит в одну из v полносвязных подсетей, соответсвующих дуальным блокам DPG(2,n) и включающих по k компьютеров. Таким образом, каждый компьютер представляется как элемент (<номер блока>,<номер дуального блока>) рассматриваемой совмещенной комбинаторной блок-схемы. Если идентификаторы двух компьютеров из разных узлов имеют одинаковые числа <номер блока> или <номер дуального блока>, то эти компьютеры связаны непосредственно, так как принадлежат одной полносвязной сети из k компьютеров. Иначе, по свойству проективной геометрии в разных узлах, в которые они входят, должны быть компьютеры с идентификаторами указанного вида. Тогда для соединения потребуются два или три шага.

В табл. 1 приведены элементы, блоки и дуальные блоки CPG(2,2).

Пример 2. Совмещенная линейная трансверсальная комбинаторная блок-схема CTD(k,n) получается на основе трансверсальной блок-схемы TD(2,n) [1] и двойственной трансверсальной блок-схемы DTD(2,n). Она является моделью компьютерной сети из kn^2 компьютеров. Сеть построена из $b = n^2$ узлов, каждый из которых соответствует блоку TD(k,n) и представляет собой полносвязную сеть из k компьютеров. Каждый компьютер при этом

Таблица 1.
V_C={(0,0), (0,1), (0,5), (1,1), (1,2), (1,6), (2,2), (2,3), (2,0), (3,3), (3,4), (3,1), (4,4), (4,5), (4,2), (5,5), (5,6), (5,3), (6,6), (6,0), (6,4)}.

Блоки CPG(2,2):	Дуальные блоки CPG(2,2):
$(0, [(0,0),(0,1),(0,5)])$	$(0, [(0,0),(6,0),(2,0)])$
$(1, [(1,1),(1,2),(1,6)])$	$(1, [(1,1),(0,1),(3,1)])$
$(2, [(2,2),(2,3),(2,0)])$	$(2, [(2,2),(1,2),(4,2)])$
$(3, [(3,3),(3,4),(3,1)])$	$(3, [(3,3),(2,3),(5,3)])$
$(4, [(4,4),(4,5),(4,2)])$	$(4, [(4,4),(3,4),(6,4)])$
$(5, [(5,5),(5,6),(5,3)])$	$(5, [(5,5),(4,5),(0,5)])$
$(6, [(6,6),(6,0),(6,4)])$	$(6, [(6,6),(5,6),(1,6)])$

входит в одну из $v = kn$ полносвязных подсетей, соответсвующих дуальным блокам DTD(k,n) и включающих по n компьютеров. Любые два компьютера из разных узлов принадлежат одной полносвязной сети из k или n компьютеров (если в их идентификаторах есть одинаковые <номер блока> или <номер дуального блока>), т.е. связаны непосредственно. Иначе в разных узлах, в которые они входят, могут быть компьютеры с идентификаторами указанного вида. Тогда для соединения потребуются два или три шага. Если же таких компьютеров в узлах нет, то по свойству трансверсальной блок-схемы найдется узел, имеющий идентификатор, подходящий для одного узла и идентификатор, подходящий для второго узла и соединение возможно за три или четыре шага.

В табл. 2 приведены элементы, блоки и дуальные блоки CTD(2,4).

По совмещенным комбинаторным блок-схемам CPG(2,n) и CTD(k,n) вычисляются маршруты для соединения компьютеров с учетом возможных исключений элементов или связей. Например, по табл. 1 для компьютеров (1,1) и (0,1) это маршрут $[(1,1),(0,1)]$, для компьютеров (1,1) и (0,5) это маршрут $[(1,1),(0,1),(0,5)]$, а если при этом вместо (0,1) имеется $(0,\_)$, то вычисляется маршрут $[(1,1),(1,2),(2,2),(2,0),(0,0),(0,5)]$.

Список литературы

[1] Холл М. Комбинаторика. - М: Мир. 1970.

[2] Фролов А.Б., Клягин А О., Кочетова Н.П., Темников Д.Ю. Распределенное вычисление комбинаторных блок-схем // Проблемы теоретической кибернетики. Матер. заочного семинара XIX Междунар. конф. / Под ред. Ю.И. Журавлева. Казань, 2020. С. 126–129.

Таблица 2.
V_C={(0,0), (0,4), (1,0), (1,6), (2,0), (2,7), (3,0), (3,5), (4,1), (4,5), (5,1), (5,7), (6,1), (6,6), (7,1), (7,4), (8,2), (8,6), (9,2), (9,4), (10,2), (10,5), (11,2), (11,7), (12,3), (12,7), (13,3), (13,5), (14,3),(14,4), (15,3), (15,6)}.

Блоки CTD(2,2):	Дуальные блоки CTD(2,2):
(0, [(0,0), (0,4)]), (1, [(1,0), (1,6)])	(0, [(0,0), (1,0), (2,0), (3,0)])
(2, [(2,0), (2,7)]), (3, [(3,0), (3,5)])	(1, [(4,1), (5,1), (6,1), (7,1)])
(4, [(4,1), (4,5)]), (5, [(5,1), (5,7)])	(2, [(8,2), (9,2), (10,2), (11,2)])
(6, [(6,1), (6,6)]), (7, [(7,1), (7,4)])	(3, [(12,3), (13,3), (14,3), (15,3)])
(8, [(8,2), (8,6)]), (9, [(9,2), (9,4)])	(4, [(0,4), (7,4), (9,4), (14,4)])
(10, [(10,2), (10,5)])	(5, [(3,5), (4,5), (10,5), (13,5)])
(11, [(11,2), (11,7)])	(6, [(1,6), (6,6), (8,6), (15,6)])
(12, [(12,3), (12,7)])	(7, [(2,7), (5,7), (11,7), (12,7)])
(13, [(13,3), (13,5)])	
(14, [(14,3), (14,4)])	
(15, [(15,3), (15,6)])	

Моделирование двунаправленного движения на луче клеточными автоматами

Кузнецова Екатерина Викторовна

Московский государственный университет им. М. В. Ломоносова,
Механико-математический факультет, e-mail: kuz.net.sova@mail.ru

Пусть S — множество конечных и бесконечных последовательностей, состоящих из элементов $\alpha_n \in \{sf, s, b\}$, в префиксе любой длины которых количество символов b не превышает количества символов f. Элементы множества S будем называть *законами движения*. Символ f подразумевает движение на одну клетку вправо, s — остаться на месте, b — на одну клетку влево.

Экраном будем называть следующую конструкцию.

Пусть имеется бесконечная в правую сторону полоса шириной в одну клетку. В каждую клетку полосы поместим по одному экземпляру одного и того же конечного автомата. К входам этого автомата присоединим выходы автоматов, стоящих в двух соседних с ним клетках, то есть у автомата имеется *левый* вход, *правый* вход и текущее состояние автомата. Выходом автомата в заданный момент времени является его состояние в этот момент времени. Для автомата, стоящего в самой левой клетке полосы левый вход не определён. Будем называть его *управляющим входом* и подавать на него *управляющие сигналы*. Управляющая последователь-

ность сигналов вырабатывается некоторым управляющим устройством в соответствии с законом движения.

Значения состояний клеточного автомата, при которых считается, что клетка, находящаяся в данном состоянии, видима (чёрная) будем называть *метками*.

Будем говорить, что *на экране реализуется движение по закону* $A \in S$, если выполняются следующие условия:

1) в некоторый момент времени в самой левой клетке экрана появляется метка (до этого на экране нет меток), этот момент будем называть *моментом начала движения* или *началом движения*;
2) изменение позиции метки на экране в i-й момент от начала движения соответствует i-й букве в слове или сверхслове A, а именно, если $A(i) = s$, то в $(i+1)$-й момент метка остается в той же клетке, где была в текущий момент, если $A(i) = f$, то в $(i + 1)$-й момент метка сдвинется на одну ячейку вправо, если $A(i) = b$, то в $(i + 1)$-й момент метка сдвинется на одну клетку влево, по сравнению со своим текущим положением;
3) в каждый момент времени после начала движения на экране есть ровно одна метка.

Экран будем называть *универсальным* для множества законов движения S, если для любого закона движения из S существует такая последовательность управляющих сигналов, что на экране формируется такое изображение, что метка движется по закону S.

Рассмотрим клеточный автомат, заданный на бесконечной в правую сторону полосе. Причем левый вход первой слева клетки является управляющим, туда подаются управляющие сигналы (его нельзя изменить через функцию переходов φ, определяющую зависимость состояния клетки от состояний в предыдущий момент времени этой самой клетки и соседних с ней клеток; просто на каждом следующем такте туда подается следующий элемент управляющей последовательности).

Изначально в данном клеточном автомате одни нули. Затем управляющее устройство начинает подавать ему на вход управляющие сигналы (управляющая последовательность). В какой-то момент в самой левой клетке экрана появится метка, которая интерпретируется как точка, движение которой мы и изучаем.

Таким образом, под появлением точки на экране будем подразумевать переключение клетки автомата, соответствующей самой левой клетке экрана, в состояние, соответствующее состоянию метки.

После того, как точка (метка) появилась на экране, то она никуда не исчезает и двух точек (меток) на экране быть не может (поэтому, если

метка движется, например, вправо, то на изначальном месте она должна затереться, т. е. клетка, в которой была метка, должна перейти в состояние, не соответствующее метке).

Законы движения из S, обладающие тем свойством, что в них не встречается двух подряд идущих символов f, можно реализовать клеточным автоматом с пятью состояниями, причем оценка на количество состояний не улучшаема.

Ранее Титовой Е. Е. был получен аналогичный результат для законов движения без движения назад.

Теорема. [Титова [1]] *Пусть* $F = ((sf) \vee (s))^{\infty}$ *— множество законов движения, состоящих из элементов множества* $\{sf, s\}$. *Тогда мощность наименьшего множества состояний клеточного автомата, необходимого для осуществления любого закона из* F *равна 4.*

Теорема 1 по сути является расширением результата Титовой.

Теорема 1. *Пусть* $S = ((sf) \vee (s) \vee (b))^{\infty}$ *— множество законов движения, состоящих из элементов множества* $\{sf, s, b\}$ *таких, что в префиксе* S *любой длины количество символов* b *не превышает количества символов* f. *Тогда мощность наименьшего множества состояний клеточного автомата, необходимого для осуществления любого закона из* S *равна 5.*

Верхняя и нижняя оценки доказаны в работе [2].

Автор выражает благодарность научному руководителю, д. ф.-м. н., профессору Э. Э. Гасанову за постановку задачи и научное руководство.

Список литературы

[1] Титова Е. Е. Конструирование движущихся изображений клеточными автоматами // Интеллектуальные системы. — 2014. — Т. 18, №1. — С. 153–180.

[2] Кузнецова Е. В. Число состояний универсального автомата бесконечного экрана, реализующего двунаправленное движение на луче // Интеллектуальные системы. Теория и приложения. — 2021. — Т. 25, №1. — С. 127–148.

Комбинаторные алгоритмы на конечных графах

Кузьмин Олег Викторович[1], **Лавлинский Максим Викторович**[2]

[1] Иркутский государственный университет, e-mail: quzminov@mail.ru

[2] Иркутский государственный университет, e-mail: lavlinskimv@mail.ru

Мы исследуем проблему решения традиционных задач комбинаторного конечного графа с использованием безопасных методов вычислений, уделяя особое внимание задачам кратчайшего пути.

Проблема вычисления функции на наборе секретных входных данных без их утечки (кроме выходных данных функции) находится в центре исследований в области криптографии. Уже созданы безопасные и эффективные алгоритмы для аукционов [2], голосования [4], сравнительного анализа [1], распознавания лиц [6]. Общей чертой всех этих алгоритмов является то, что процесс оценки функции не принимает во внимание входные данные, на которых функция должна быть оценена. Вычисление наибольшей из n ставок или суммирование n голосов производится путем выполнения n сравнений или суммирования независимо от рассматриваемых значений.

Однако существуют задачи, для которых процесс оценки зависит от входных данных. В этом случае, даже если все обрабатываемые данные зашифрованы, потока выполнения может быть достаточно для утечки скрытой информации.

Обычно это имеет место в комбинаторных задачах, в которых проблемы с графами являются одними из наиболее распространенных [5]. Рассмотрим, например, компании, занимающиеся доставкой грузов. Эти компании могут быть заинтересованы в вычислении наиболее быстрого способа доставки груза из одного места в другое, но не хотят делиться друг с другом точными перемещениями, которые они используют, и характеристиками своих грузовиков. Их проблема может быть решена путем надежной реализации традиционных алгоритмов кратчайшего пути, таких как алгоритмы Беллмана-Форда или Дейкстры. Непосредственным способом безопасного вычисления кратчайшего пути было бы скрыть (зашифровать или разделить) вес всех ребер соответствующего графа. Однако и этот подход может привести к утечке значительного количества скрытой информации.

Для решения проблемы безопасной реализации комбинаторных алгоритмов на конечных графах можно использовать протоколы на основе арифметической функции черного ящика FABB Дамгарда и Нильсена [3]. Эта функция позволяет n сторонам безопасно хранить элементы кольца Z_m, многократно выполнять кольцевые операции сложения и умно-

жения над этими элементами и открывать результат вычислений, когда это необходимо. Следуя Тофту [7], можно рассмотреть расширенную и абстрактную версию этой функции, которая предлагает возможность выполнять безопасное сравнение и рассматривать любое возможное кольцо. Таким образом, сохранение, открытие, сложение, умножение и сравнение будут единственными безопасными операциями, на которые будут опираться комбинаторные алгоритмы.

Проблема кратчайшего пути с одним источником – одна из основных проблем теории графов. Алгоритмы кратчайшего пути также используются в качестве подалгоритмов для более сложных задач, таких как проблема китайского почтальона или проблема максимального потока. Мы рассмотрим два стандартных алгоритма поиска кратчайшего пути из одного источника в конечном графе с взвешенными ребрами: алгоритм Дейкстры и алгоритм Беллмана-Форда. Первый требует, чтобы все веса ребер были положительными, а второй только предполагает, что во входном графе нет цикла с отрицательным весом.

Алгоритм Беллмана-Форда

Алгоритм Беллмана-Форда продолжает повторное сканирование всех ребер в поисках добавления ребер, которые уменьшают текущее расстояние от источника до различных вершин. Если проход по краям не улучшил текущее решение, или если края были просканированы V раз, то выполнение алгоритма останавливается. Особенностью этого алгоритма является то, что его последовательность операций зависит только от структуры графа, но не от веса ребер. Его недостатком является временная сложность: классическая реализация выполняется за O (|V||E|) времени.

Безопасный алгоритм поиска кратчайшего пути отличается от классического алгоритма Беллмана-Форда двумя аспектами:

- ветвление, соответствующее обнаружению более короткого пути, обрабатывается посредством арифметики Дамгарда-Нильсена и Тофта,
- условие раннего завершения алгоритма Беллмана-Форда, которое срабатывает, если внутренний цикл не действует в течение одного прохода, удаляется, так как это может привести к утечке информации.

Эти аспекты не отменяют правильность алгоритма, а только увеличивает время его работы.

Алгоритм Дейкстры

Алгоритм Дейкстры вычисляет кратчайший путь от источника до всех вершин графа, то есть дерево кратчайших путей с корнем в источнике. Алгоритм жадный. На каждой итерации одна вершина (с наименьшей меткой расстояния) постоянно обновляется до статуса сканированной.

Основные различия между классическим Алгоритмом Дейкстры и безопасной версией алгоритма Дейкстры находятся во вложенном цикле:

- цикл проходит через все вершины, а не только на соседей текущей вершины,
- алгоритм проходит через все элементы строки или вектора, даже если мы знаем, что только один из них будет обновлен.

Эти две модификации способствуют увеличению сложности алгоритма Дейкстры. Мы рассмотрели два алгоритма для безопасного вычисления кратчайших путей. Эти протоколы вызывают интерес во многих аспектах, ведь проблема сохранения конфиденциальных данных является очень актуальной в наше время.

Работа выполнена при поддержке РФФИ и Правительства Иркутской области (проект № 20-41-385001).

Список литературы

[1] Барни, М., Файлла, П., Колесников, В., Лаззеретти, Р., Садеги, А. Р., Шнайдер, Т .: Безопасная оценка частных программ линейного ветвления с медицинскими приложениями. В кн .: ESORICS. Том 5789 из LNCS., Springer (2009) с. 424-439.

[2] Богетофт, П., Дамгард, И., Якобсен, Т. П., Нильсен, К., Пагтер, Дж., Тофт, Т .: Практическая реализация безопасных аукционов, основанная на многосторонних целочисленных вычислениях. В кн .: Финансовая криптография. Том 4107 журнала LNCS., Springer (2006) с. 142-147.

[3] Дамгард, И., Нильсен Д.,: Универсально компонуемое эффективное многостороннее вычисление на основе порогового гомоморфного шифрования. В: CRYPTO. Том 2729 LNCS., Springer (2003) с. 247-264.

[4] Крамер, Р., Франклин, М.К., Шенмакерс, Б., Юнг, М .: Выборы тайным голосованием с участием нескольких органов власти с линейной работой. В: EUROCRYPT. Том 1070 LNCS., Springer (1996) с. 72-83.

[5] Кузьмин, О. В. Комбинаторные методы дискретного анализа: учебное пособие / О. В. Кузьмин. — Иркутск: Издательство ИГУ, 2013. — 126 с.

[6] Садеги, А. Р., Шнайдер, Т., Веренберг, И.: Эффективное распознавание лиц с сохранением конфиденциальности. В: ICISC. Том 5984 из LNCS., Springer (2009) с. 229-244.

[7] Тофт, Т .: Безопасные структуры данных на основе многосторонних вычислений. В: PODC, ACM (2011) с. 291-292.

Самодуальные обобщённые бент-функции и их свойства

Куценко Александр Владимирович

Новосибирский государственный университет, Институт математики им. С.Л. Соболева СО РАН, e-mail: alexandrkutsenko@bk.ru

Через $\mathbb{F}_2^n$ обозначим линейное пространство всех двоичных векторов длины n над полем $\mathbb{F}_2$. Пусть q — натуральное число; *обобщённой булевой функцией* от n переменных называется отображение вида $\mathbb{F}_2^n \to \mathbb{Z}_q$. Множество всех обобщённых булевых функций от n переменных обозначим $\mathcal{GF}_n^q$. Для каждой пары $x, y \in \mathbb{F}_2^n$ через $\langle x, y\rangle$ обозначается значение $\bigoplus\limits_{i=1}^{n} x_i y_i$. *Весом Хэмминга* $\mathrm{wt}(x)$ вектора $x \in \mathbb{F}_2^n$ называется число его ненулевых координат. Ортогональной группой порядка n над полем $\mathbb{F}_2$ называется группа $\mathcal{O}_n = \left\{L \in \mathrm{GL}\,(n, \mathbb{F}_2) : LL^{\mathrm{T}} = I_n\right\}$, где L^{T} — транспонирование L; I_n — единичная матрица порядка n над полем $\mathbb{F}_2$.

Обобщённым преобразованием Уолша—Адамара функции $f \in \mathcal{GF}_n^q$ называется функция $H_f : \mathbb{F}_2^n \to \mathbb{C}$, заданная равенством

$$H_f(y) = \sum_{x \in \mathbb{F}_2^n} \omega^{f(x)} (-1)^{\langle x, y\rangle}, \quad y \in \mathbb{F}_2^n,$$

где $\omega = e^{2\pi i/q}$. Функция $f \in \mathcal{GF}_n^q$ называется *обобщённой бент-функцией*, если $|H_f(y)| = 2^{n/2}$ для каждого $y \in \mathbb{F}_2^n$ [4]. Обзор различных обобщений бент-функций представлен в работе [1]. Множество обобщённых бент-функций обозначается через $\mathcal{GB}_n^q$. *Весом Ли* вектора $x \in \mathbb{Z}_q$ называется число $\mathrm{wt}_L(x) = \min\{x, q - x\}$. *Расстояние Ли* $\mathrm{dist}_L(f, g)$ между функциями $f, g \in \mathcal{GF}_n^q$ определяется как $\mathrm{dist}_L(f, g) = \sum\limits_{x \in \mathbb{F}_2^n} \mathrm{wt}_L\left(\delta(x)\right)$, где $\delta \in \mathcal{GF}_n^q$ и $\delta(x) = f(x) + (q-1)g(x)$, $x \in \mathbb{F}_2^n$.

Пусть $f \in \mathcal{GB}_n^q$, тогда если существует функция $\widetilde{f} \in \mathcal{GF}_n^q$, такая, что $H_f(y) = \omega^{\widetilde{f}(y)} 2^{n/2}$, то обобщённая бент-функция f называется *регулярной*, а функция $\widetilde{f}$ — дуальной к f. Дуальная функция также является регулярной обобщённой бент-функцией. Если $f = \widetilde{f}$, то f называется *самодуальной* обобщённой бент-функцией. Если $f = \widetilde{f} + q/2$, то f называется *анти-самодуальной* обобщённой бент-функцией. Всюду далее считается, что q — чётное натуральное число.

Открытой проблемой является классификация булевых самодуальных бент-функций ($q = 2$). Подробную информацию о бент-функциях, известных результатах, связанных с ними, а также и их приложениях можно найти в книге [7]. В ряде работ исследованы свойства самодуальных

бент-функций в рамках различных обобщений бент-функций: так, в работах [2, 3] рассматривается обобщение вида $\mathbb{F}_p^n \to \mathbb{F}_p$, где p простое. Получен ряд результатов, в частности, представлена полная классификация квадратичных самодуальных бент-функций. Связь самодуальных обобщённых бент-функций вида $\mathbb{F}_2^n \to \mathbb{Z}_4$ и самодуальных булевых бент-функций исследована в работе [6].

Одной из самых известных конструкций (обобщённых) бент-функций является конструкция Мэйорана—МакФарланда. В настоящей работе получены необходимые и достаточные условия самодуальности обобщённых бент-функций, построенных с помощью конструкции Мэйорана—МакФарланда.

Утверждение 1. *Обобщённая бент-функция Мэйорана—МакФарланда*

$$f(x,y) = \frac{q}{2}\langle x, \pi(y)\rangle + g(y),\ x, y \in \mathbb{F}_2^{n/2},$$

является (анти-)самодуальной тогда и только тогда, когда

$$\pi(y) = L\left(y \oplus b\right), \quad g(y) = \frac{q}{2}\langle b, y\rangle + d, \quad y \in \mathbb{F}_2^{n/2},$$

где $L \in \mathcal{O}_{n/2}$; $b \in \mathbb{F}_2^{n/2}$; $\mathrm{wt}\,(b)$ — чётное (нечётное) число; $d \in \mathbb{Z}_q$.

Далее представлен спектр расстояний Ли между (анти-)самодуальными обобщёнными бент-функциями из класса Мэйорана—МакФарланда; для данного спектра используется обозначение Sp_L.

Теорема 1. *Справедливо*

$$\mathrm{Sp}_L = \left\{q \cdot 2^{n-2}\right\} \cup \bigcup_{w=0}^{q/2} \bigcup_{r=0}^{n/2-1} \left\{q \cdot 2^{n-2}\left(1 \pm \frac{1}{2^r}\right) \mp w \cdot 2^{n-r}\right\}.$$

Более того, все приведённые расстояния достижимы.

Из Теоремы 1 следует, что минимальное расстояние Ли между самодуальными обобщёнными бент-функциями из класса Мэйорана—МакФарланда равно $q \cdot 2^{n-3}$.

Утверждение 2. *Пусть $n \geqslant 4$ — чётное число и f — самодуальная обобщённая бент-функция от n переменных. Для характеристического вектора $\omega^f = \left(F^{00}, F^{01}, F^{10}, F^{11}\right)$, где $F^{jk} \in \left\{1, \omega, \omega^2, \ldots, \omega^{q-1}\right\}^{2^{n-2}}$, $j, k = 0, 1$, справедливо*

$$\left\langle F^{00}, F^{01}\right\rangle + \left\langle F^{10}, F^{11}\right\rangle = 0, \qquad \left\langle F^{00}, F^{10}\right\rangle + \left\langle F^{01}, F^{11}\right\rangle = 0.$$

Хорошо известно, что булева бент-функция не может быть аффинной. Но для случая обобщенных бент-функций ситуация нетривиальна: например, в работе [5] показано, что для случая, когда q кратно 4, существуют аффинные обобщённые бент-функции. Оставался открытым вопрос о существовании аффинных самодуальных обобщённых бент-функций для различных q.

Теорема 2. *Для любого положительного чётного q и произвольного натурального n не существует самодуальных обобщённых бент-функций вида $f(x) = \sum_{i=1}^{n} \lambda_i x_i + \lambda_0$, где $\lambda_0, \lambda_1, \ldots, \lambda_n \in \mathbb{Z}_q$.*

Далее представлен класс отображений, сохраняющих самодуальность обобщенной бент-функции.

Теорема 3. *Отображения множества всех обобщённых булевых функций от n переменных в себя, имеющие вид $f(x) \longrightarrow f\left(L\left(x \oplus c\right)\right) + \frac{q}{2}\langle c, x\rangle + d$, $x \in \mathbb{F}_2^n$, где $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$ — чётное число, $d \in \mathbb{Z}_q$, сохраняют самодуальность обобщённой бент-функции.*

Заметим, что каждое такое отображение сохраняет расстояние Хэмминга и растояние Ли между обобщёнными бент-функциями, то есть является изометричным. С помощью отображений данного вида получена уточнённая классификация кватернарных самодуальных бент-функций от 4 переменных, представленная в работе [6].

Работа выполнена в рамках госзадания ИМ СО РАН (проект № 0314-2019-0017), при поддержке РФФИ (проект № 20-31-70043) и лаборатории криптографии JetBrains Research.

Список литературы

[1] Токарева Н. Н. Обобщения бент-функций. Обзор работ // Дискрет. анализ исслед. опер. — 2010. — Т. 17, № 1. — С. 33–62.

[2] Çeşmelioğlu A., Meidl W., and Pott A. On the dual of (non)-weakly regular bent functions and self-dual bent functions // Adv. Math. Commun. — 2013. V. 7. № 4. — P. 425–440.

[3] Hou X.-D. Classification of p-ary self dual quadratic bent functions, p odd // J. Algebra. — 2013. № 391. — P. 62–81.

[4] Schmidt K.-U. Quaternary constant-amplitude codes for multicode CDMA // IEEE Trans. Inform. Theory. — 2009. V. 55. № 4. — P. 1824–1832.

[5] Singh B. K. On cross-correlation spectrum of generalized bent functions in generalized Maiorana—McFarland class // Inform. Sci. Lett. — 2013. V. 2. № 3. — P. 139–145.

[6] Sok L., Shi M., and Solé P. Classification and construction of quaternary self-dual bent functions // Cryptogr. Commun. — 2018. V. 10. №2. — P. 277–289.

[7] Tokareva N. Bent Functions: Results and Applications to Cryptography. — Acad. Press, Elsevier, 2015. — 230 p.

О верхней оценке количества дополнительных рёбер в минимальных вершинных 1-расширениях двухмерных решёток

Лобов Александр Андреевич[1], **Абросимов Михаил Борисович**[2]

[1] ФГБОУ ВО «Саратовский национальный исследовательский государственный университет имени Н.Г.Чернышевского», e-mail: aisanekai@mail.ru

[2] ФГБОУ ВО «Саратовский национальный исследовательский государственный университет имени Н.Г.Чернышевского», e-mail: mic@rambler.ru

Граф G^* называется вершинным k-расширением (В-k-Р) графа G если G вкладывается в каждый граф, полученный из G^* удалением k вершин. Граф $G = (V, \alpha)$ вкладывается в граф $H = (U, \beta)$ если существует отображение $\phi : V \to U$ такое, что $\forall u, v \in V : \{u, v\} \in \alpha \Rightarrow \{\phi(u), \phi(v)\} \in \beta$.

Вершинные k-расширения — это математическая модель на основе графов задачи построения k-отказоустойчивой реализации системы [1,2,3]. Суть задачи – добавление в исходную систему дополнительных элементов и связей так, чтобы при отказе какого-либо элемента отказоустойчивой системы в ней содержалась копия исходной.

Так как каждый дополнительный элемент и каждая связь имеют свою цену, важно минимизировать их количество. Минимальное количество элементов, которые нужно добавить в систему, равно k. Количество дополнительных связей при этом может может отличаться. Вершинное k-расширение называется минимальным (МВ-k-Р) если количество дополнительных вершин в нём равно k, а количество дополнительных рёбер минимально среди всех таких расширений. Количество дополнительных рёбер в МВ-k-Р графа G обозначается $ec_k(G)$.

Прямоугольная двухмерная решётка — удобная с точки зрения размещения элементов топология. Далее рассматриваются только прямоугольные решётки. Решётку $n \times m$ будем обозначать $M_{n,m}$. Пример решётки 3×4 $M_{3,4}$ можно увидеть на рисунке 1 слева.

Теорема 1 описывает В-1-Р решётки. Расширение рассмотренного ранее графа можно увидеть на рисунке 1 (справа). Ранее было показано [4], что

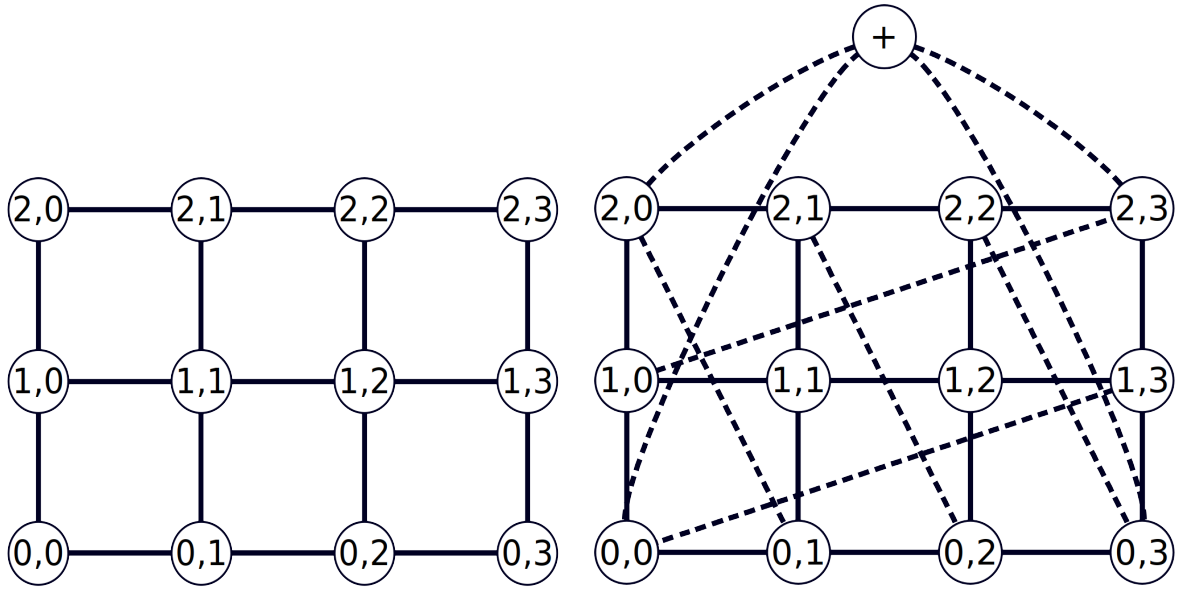


Рис. 1: Пример двухмерной решётки 3×4 (слева) и её расширения (справа).

расширения такого типа могут быть построены алгоритмом A2 [5]. Можно предложить более простой способ.

Теорема 1. *Для каждой двухмерной решётки $M_{n,m}$ существует B-1-P, количество дополнительных рёбер в котором равно $n+m+2$, которое может быть построено по следующей схеме:*

1. *найти вершины со степенями 2 и 3. Данные вершины будут образовывать цикл $(a_1, ..., a_{n-1}; b_1, ..., b_{m-1}; c_1, ..., c_{n-1}; d_1, ..., d_{m-1})$, где степени вершин a_1, b_1, c_1, d_1 равны 2;*
2. *добавить в граф вершину w и рёбра $\{w, a_1\}, \{w, b_1\}, \{w, c_1\}, \{w, d_1\}$;*
3. *добавить рёбра $\{a_i, c_{n-i}\}$ и $\{b_j, d_{m-j}\}$ для $1 \le i \le n-1$ и $1 \le j \le m-1$.*

Также у каждого графа существует получающееся добавлением в граф одной вершины и рёбер между ней и каждой вершиной исходного графа тривиальное вершинное 1-расширение [3].

Из всего описанного следует теорема 2.

Теорема 2. *Количество дополнительных рёбер в MB-1-P двухмерной решётки $n \times m$ не превосходит минимума из $n \times m$ и $n+m+2$, т. е. $ec_1(M_{n,m}) \le min\{n+m+2, n \times m\}$.*

Таким образом, была получена оценка сверху количества дополнительных рёбер в минимальном вершинном 1-расширении двухмерных решёток.

Стоит отметить, что данная оценка является именно оценкой сверху, пример этого изображён на рисунке 2.

На рисунке 2 слева изображено MB-1-P графа $M_{3,3}$. Количество дополнительных рёбер в нём меньше, чем в расширении того же графа, по-

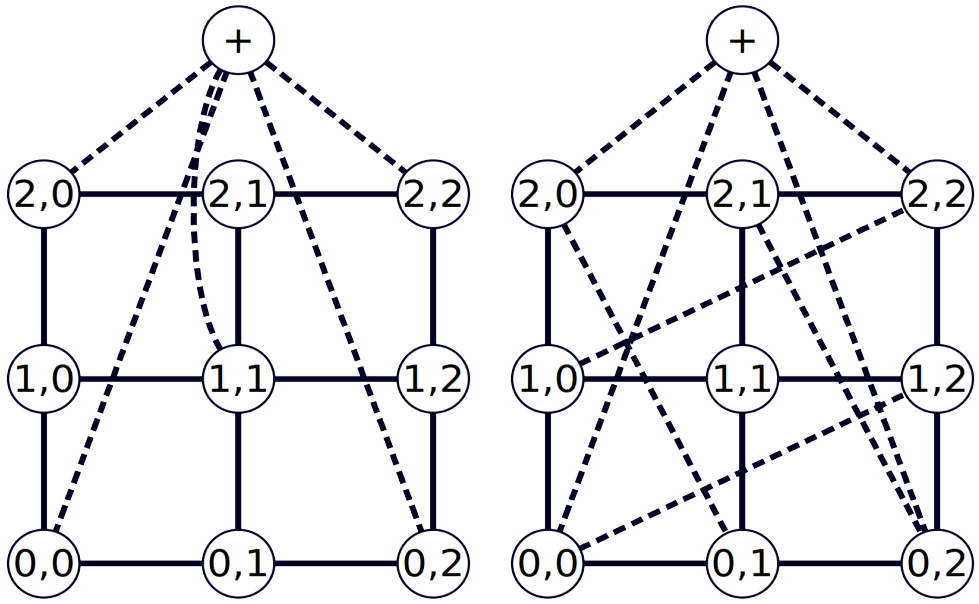


Рис. 2: Расширения двухмерной решётки $M_{3,3}$: минимальное (слева) и построенное по теореме 1 (справа).

строенном по схеме из теоремы 2 (изображено на том же рисунке справа). Данное минимальное вершинной 1-расширение было представлено в каталоге в работе [6].

Список литературы

[1] Hayes J. P. A graph model for fault-tolerant computing system // IEEE Trans. Comput. 1976. № 9. — P. 875–884.

[2] Harary F., Hayes J. P. Node fault tolerance in graphs // Networks. — 1996. — Vol. 27. — P. 19–23.

[3] Абросимов М. Б. Графовые модели отказоустойчивости // Саратов : Изд-во Сарат. ун-та, — 2012. — 192 с.

[4] Каравай М. Ф., Минимизированное вложение произвольных гамильтоновых графов в отказоустойчивый граф и реконфигурация при отказах. II. решетки и k-отказоустойчивость // Автоматика и телемеханика. — 2004. — № 2. — С. 175–189.

[5] Каравай М. Ф., Минимизированное вложение произвольных гамильтоновых графов в отказоустойчивый граф и реконфигурация при отказах. I // Автоматика и телемеханика. — 2004. — № 12. — С. 159–178.

[6] Камил И. А. К. Вычислительный эксперимент по построению отказоустойчивых реализаций графов с числом вершин до 9 // International Journal of Open Information Technologies. — 2020. — Т 8. № 9. — С. 43–47.

Оценки площади мультиплексорных функций в одной модели клеточных схем

Ложкин Сергей Андреевич[1], **Зизов Вадим Сергеевич**[2]

[1] ВМК МГУ им. М.В. Ломоносова, e-mail: lozhkin@cs.msu.ru
[2] ВМК МГУ им. М.В. Ломоносова, e-mail: vzs815@gmail.com

Модель клеточных схем (**КС**) является математической моделью интегральных схем (ИС), учитывающей особенности физического синтеза. Наличие требований на геометрию схемы, обеспечивающих учёт необходимых трассировочных ресурсов при создании ИС, представляет собой принципиальное отличие от хорошо изученной модели «обычных» схем из функциональных элементов (СФЭ). Модель **КС** в некотором конкретном базисе из коммутационных и функциональных элементов впервые была предложена в 1967 году С.С. Кравцовым в работе [1], где для $n = 1, 2, \ldots$ был получен порядок роста вида 2^n функции Шеннона $A(n)$, характеризующей сложность (площадь) самой «сложной» функции алгебры логики (ФАЛ) от n переменных. В работе А. Альбрехта [2] доказано, что функция Шеннона $A(n)$ при $n = 1, 2, \ldots$ асимптотически равна $\sigma 2^n$, где σ - некоторая константа, точное значение которой неизвестно в настоящее время. При этом из мощностных соображений и работ [1, 2] следует, что σ находится в сегменте $[\frac{1}{4}, \frac{9}{2}]$. Заметим, что в работе [3] был предложен специальный базис **КС**, для которого константа, аналогичная константе σ, равна 1, то есть соответствующая функция Шеннона для площади КС, реализующих ФАЛ от n переменных, асимптотически равна 2^n.

В рамках данной модели были получены нижние и верхние оценки сложности для некоторых специальных ФАЛ и систем ФАЛ. Так, максимальная по порядку нижняя оценка для последовательности ФАЛ от $n, n = 1, 2, \ldots$ булевых переменных (БП), равная n^2, получена в работе Ю. Хромковича, С.А. Ложкина и др. [4].

В работе [5] для одного специального базиса **КС** - базиса $Б_0'$ были получены асимптотические точные верхние и нижние оценки для площади схем, реализующих дешифратор порядка n, которые совпадают в первом члене разложения, имеют вид $n2^{n-1}\left(1 \pm O\left(\frac{1}{n}\right)\right)$, и аналогично работе [6], могут считаться асимптотическими оценками высокой степени точности (АОВСТ). В настоящей работе установлены асимптотически точные верхние и нижние оценки для площади **КС** над базисом $Б_0'$, реализующих т.н. мультиплексорные ФАЛ.

Напомним, что **КС** является прямоугольной решёткой на плоскости, состоящей из клеток - единичных квадратов, каждый из которых представляет собой либо *коммутационный* элемент (КЭ), либо *функциональный*

элемент (ФЭ) исходного базиса. При этом *функциональным* называется элемент, который реализует хотя бы одну нетождественную ФАЛ, а *коммутационные элементы*, тем самым, реализуют только тождественные ФАЛ и их назначением является передача сигналов. Входами и выходами элемента являются контакты, сопоставленные сторонам его квадрата, при этом *изолированными* называются контакты элемента, которые не подсоединены к входам либо выходам других элементов.

Клеточная схема представляет собой своего рода вложение СФЭ над базисом из ФЭ исходного базиса Б в прямоугольную решётку, что обусловлено необходимостью учёта трассировочных ресурсов при размещении СФЭ в «пространстве» (на кристалле) ИС. Будем предполагать, что функционирование **КС** Σ определяется функционированием соответствующей ей СФЭ S, причем ее входы и выходы располагаются однократно на границе Σ (см., например, [5]). Под сложностью (площадью) $A_{\text{Б}}(F)$ системы ФАЛ F в классе **КС** над базисом Б понимается минимальная площадь **КС** в базисе Б, реализующих F.

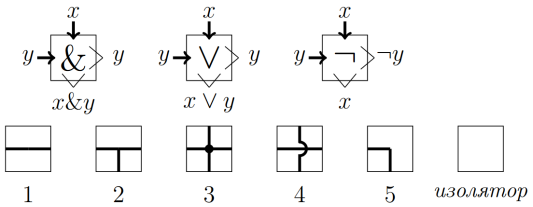


Рис. 1. Базис $\text{Б}_0'$: *ФЭ* конъюнкции (&), дизъюнкции ($\vee$) и отрицания ($\neg$); *КЭ* - проводник (1), Т-образный разветвитель (2), разветвитель (3), пересечение без соединения (4), поворот (5) и изолятор, т.е. КЭ с изолированными контактами

В данной работе, как и в [5], рассматривается базис $\text{Б}_0'$ - один из возможных базисов **КС**, связанных с элементами стандартного базиса алгебры логики $\text{Б}_0 = \{x_1x_2, x_1 \vee x_2, \overline{x}_1\}$, который состоит из 3 функциональных и 7 коммутационных элементов (см. Рис. 1).

Под стандартным *мультиплексором порядка* n или, иначе говоря, под стандартной *мультиплексорной ФАЛ порядка* n понимается ФАЛ μ_n с n адресными входами $x_1, \ldots, x_n$, 2^n информационными входами $y_0, \ldots, y_{2^n-1}$, которая при наборе $\sigma = (\sigma_1, \ldots, \sigma_n)$ на адресных входах тождественно равна $y_{\nu(\sigma)}$, то есть информационному входу с номером $\nu(\sigma) = \sum_{i=1}^{n} \sigma_i 2^{n-i}$. Иначе говоря, мультиплексорная функция может быть представлена в виде следующей ДНФ, где, как обычно, $x_i^0 = \overline{x_i}, x_i^1 = x_i$:

$$\mu_n(x_1, \ldots, x_n, y_0, \ldots, y_{2^n-1}) = \bigvee_{\sigma=(\sigma_1,\ldots,\sigma_n)} x_1^{\sigma_1} \cdots x_n^{\sigma_n} y_{\nu(\sigma)}.$$

Теорема 1 (о верхней оценке) *Для $n = 1, 2, \ldots$ выполняется неравенство*

$$A_{Б_0'}(M_n) \leqslant n2^{n-1} + O(2^{n-1}).$$

Теорема 2 (о нижней оценке) *Для $n = 1, 2, \ldots$ в модели **КС** над произвольным полным базисом Б:*

$$A_{Б}(\mu_n) \geqslant (n - 3 - 2\log n)2^{n-1} + O(n^2).$$

Следствие (асимптотическая оценка) *Из теоремы 1 и теоремы 2 следует, что при $n = 1, 2, \ldots$ в модели **КС** справедливы оценки*

$$n2^{n-1} - O(\log n2^n) \leqslant A_{Б_0'}(\mu_n) \leqslant n2^{n-1} + O(2^n),$$

которые можно считать близкими к т.н. АОВСТ оценками [6].

Таким образом, в настоящей работе установлены нижние и верхние асимптотически точные оценки сложности **КС** в базисе $Б_0'$, реализующих мультиплексорную ФАЛ (схемных мультиплексоров). Более того, полученные оценки являются близкими к АОВСТ оценками.

Статья опубликована при финансовой поддержке Минобрнауки в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению №075-15-2019-1621.

Список литературы

[1] Кравцов С. С. О реализации функций алгебры логики в одном классе схем из функциональных и коммутационных элементов // Проблемы кибернетики. М.: Наука. — 1967. — Т. 19. — С. 285–292.

[2] Альбрехт А. О схемах из клеточных элементов // Проблемы кибернетики. М.: Наука. — 1975. — Т. 33. — С. 209–214.

[3] Грибок С. В. Об асимптотике сложности клеточного контактного дешифратора // Вестник ННГУ — 2012. — Т. 1, № 4. — С. 225–231.

[4] Lower Bounds on the Area Complexity of Boolean Circuit / J. Hromkovic, S. Lozkin, A. Rybko, A. Sapozhenko, N. Skalikova // Theor. Comput. Sci — 1992. — Т. 97 — С. 285–300.

[5] Ложкин С. А., Зизов В. С. Уточненные оценки сложности дешифратора в модели клеточных схем из функциональных и коммутационных элементов // Учен. зап. Казан. ун-та. Сер. Физ.-матем.науки. — 2020. — Т. 162, № 3. — С. 322–334.

[6] Ложкин С. А. Оценки высокой степени точности для сложности управляющих систем из некоторых классов // Математические вопросы кибернетики М., Наука, Физматлит. — 2005. — Т. 6. — С. 189–214.

О сложности реализации стандартных мультиплексорных функций в одном классе контактных схем

Ложкин Сергей Андреевич, Хзмалян Давид Эдвардович
МГУ имени М.В.Ломоносова, e-mail: lozhkin@cs.msu.ru, david.khzmalian@gmail.com

В настоящей статье рассматривается задача индивидуального синтеза контактных схем (КС) для мультиплексорной функции алгебры логики (ФАЛ) порядка n, то есть для функции μ_n от $n + 2^n$ булевых переменных (БП), первые n из которых называются «адресными», а оставшиеся 2^n — «информационными». Значение этой функции равно значению той ее информационной переменной, номер которой поступил на адресные входы.

Сложность мультиплексорной ФАЛ изучалась в ряде работ. Так, из работы [1], известно, что сложность мультиплексорной ФАЛ в классе параллельно-последовательных схем асимптотически равна

$$2^{n+1} + \frac{2^n}{n} \pm O\Big(\frac{2^n}{n \log n}\Big). \tag{1}$$

В работе [2] устанавливается нижняя оценка сложности реализации мультиплексора порядка n в классе схем из функциональных элементов (СФЭ) в так называемом унимодальном базисе U_2, состоящем из всех двухместных элементарных конъюнкций и дизъюнкций, равная $2^{n+1} - 2$. В работе [3] приводится реализация стандартного мультиплексора порядка n с помощью СФЭ в этом же базисе со сложностью $2^{n+1} + O(2^{\frac{n}{2}})$ и глубиной $n + \lceil \log n \rceil + 1$. Также известно (см., например, [4]), что сложность реализации ФАЛ μ_n, $n = 1, 2, ...$, как формулами, так и СФЭ в стандартном базисе $Б_0$, асимптотически равна 2^{n+1}. В работе [5] получена нижняя оценка вида $2^{n+1} + c_1(n) \cdot 2^{\frac{n}{2}} - O(2^{\frac{n}{4}})$ и верхняя оценка вида $2^{n+1} + c_2(n) \cdot 2^{\frac{n}{2}} + O(2^{\frac{n}{4}})$ для сложности реализации мультиплексора μ_n в классе СФЭ над базисом $Б_0$, где $c_1(n) = \frac{1}{3}$, $c_2(n) = 2$, если n четно, и $c_1(n) = 0,32$, $c_2(n) = \frac{3}{\sqrt{2}}$ если n нечетно.

Напомним некоторые определения и факты, а также введем обозначения, связанные с реализацией ФАЛ в классе контактных схем. Те понятия, которые в данной работе не определяются, могут быть найдены в [6].

Сложностью контактной схемы Σ будем называть количество контактов $L(\Sigma)$ в ней.

Сложностью ФАЛ f будем называть количество контактов $L^K(f)$ в схеме, которая реализует f с минимальной сложностью.

Будем говорить, что непустое подмножество U БП X ФАЛ f *забивает* ее БП x, $x \notin U$, если подстановкой некоторых констант вместо БП мно-

жества U из ФАЛ f можно получить ФАЛ, не зависящую существенно от x.

Множество X, состоящее из БП ФАЛ f, будем называть *незабиваемым*, если $|X| \geq 2$ и любая БП x, $x \in X$, не забивается множеством $X \setminus \{x\}$. Переменная, принадлежащая некоторому незабиваемому множеству БП ФАЛ f, считается *незабиваемой* переменной этой ФАЛ.

Из определений следует, что если U — незабиваемое множество БП ФАЛ f и $U' \subset U$, $|U'| \geq 2$, то при любой подстановке констант вместо БП из множества $U \setminus U'$ в ФАЛ f получается ФАЛ f', для которой множество U' является незабиваемым множеством. Заметим, что информационные БП образуют незабиваемое множество переменных ФАЛ μ_n, и, следовательно, в соответствии с [7], $L^K(\mu_n) \geq 2^{n+1} - 1$.

Любую ФАЛ, получающуюся из ФАЛ μ_n в результате некоторой подстановки констант вместо ее информационных БП, будем называть *квазимультиплексорной ФАЛ порядка n и ранга r*, где r — число информационных БП, вместо которых не были подставлены константы.

Будем говорить, что контактная схема Σ, реализующая мультиплексорную (квазимультиплексорную) ФАЛ μ_n, *корректна*, если любая проводящая цепь схемы Σ, содержащая вершину инцидентную некоторому контакту информационной переменной, содержит также контакт какой-либо информационной переменной инцидентный данной вершине.

Множество K', которое состоит из всевозможных корректных контактных схем, реализующих мультиплексорные или квазимультиплексорные функции, будем называть классом *корректных мультиплексорных контактных схем.*

Сложностью $L^{K'}(f)$ функции f в классе K' будем, называть сложность минимальной контактной схемы из класса K', которая реализует данную функцию.

Используя незабиваемость множества информационных переменных ФАЛ μ_n и верхнюю оценку (1), доказаны следующие утверждения.

Лемма. *Пусть Σ — минимальная контактная схема, реализующая мультиплексорную ФАЛ μ_n порядка $n, n \geq 2$. Тогда число тех информационных БП, которые имеют в схеме Σ ровно два контакта, не превосходит $8\frac{2^n}{n} - 8k + O(\frac{2^n}{n \log n})$, где k — это число тех информационных БП ФАЛ μ_n, которые имеют в схеме Σ не менее трех контактов.*

Теорема. *Для мультиплексорной ФАЛ μ_n порядка $n, n \geq 2$, справедливо следующее неравенство:*

$$L^{K'}(\mu_n) \geq 2^{n+1} + \frac{2^n}{2n} - O(\frac{2^n}{n^2}). \qquad (2)$$

Заметим, что класс параллельно-последовательных схем является частным случаем класса контактных схем. Следовательно, верхняя оценка сложности мультиплексорной ФАЛ в классе контактных схем не превосходит верхней оценки сложности мультиплексорной ФАЛ в классе параллельно-последовательных схем, то есть согласно (1)

$$L^K(\mu_n) \le 2^{n+1} + \frac{2^n}{n} + O\Big(\frac{2^n}{n\log n}\Big). \tag{3}$$

Данная оценка достигается построением корректной контактной схемы, реализующей мультиплексорную ФАЛ. Таким образом, в силу (2) и (3) верны следующие неравенства:

$$2^{n+1} + \frac{2^n}{2n} - O(\frac{2^n}{n^2}) \le L^{K'}(\mu_n) \le 2^{n+1} + \frac{2^n}{n} + O\Big(\frac{2^n}{n\log n}\Big).$$

Статья опубликована при финансовой поддержке Минобрнауки РФ в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению №075-15-2019-1621.

Список литературы

[1] Ложкин С. А., Власов Н. В. О сложности мультиплексорной функции в классе π-схем // Учен. зап. Казан. гос. ун-та. Физико-математические науки. — 2009. — Т. 151, № 2. — С. 98–106.

[2] Paul W. J. A 2,5n lower bound on the combinational complexity of Boolean functions // SIAM, Philadelphia: SIAM Journal on Computing. — 1977. — Т. 6, — С. 427–443.

[3] Klein P., Paterson M. S. Asymptotically optimal circuit for a storage access function // IEEE Trans, on Computers, IEEE Computer Society. — 1980. Т. 29, № 8. — С. 737–738.

[4] Коровин В. В. О сложности реализации универсальной функции схемами из функциональных элементов // Дискретная математика. — 1995. — Т. 7, вып. 2. — С. 95–102.

[5] Румянцев П. В. О сложности реализации мультиплексорной функции схемами из функциональных элементов // Тезисы докладов XIV международной конференции (Пенза, 23-28 мая 2005 г.). — 2005. — С. 133.

[6] Ложкин С. А. Лекции по основам кибернетики. — М. : Изд. отдел фак. ВМиК МГУ, 2004. — 256 с.

[7] Ложкин С. А. Дополнительные главы кибернетики и теории управляющих систем [Электронный ресурс]. — 2013. — 76 с. — Способ доступа: https://mk.cs.msu.ru/images/3/39/Лекции_ДГКТУС_Часть_1-2.pdf

Полная классификация сложности задачи о реберной раскраске для классов субкубических графов, определяемых 8-реберными запретами

Малышев Дмитрий Сергеевич

Национальный исследовательский университет «Высшая школа экономики», Нижегородский государственный университет им. Н. И. Лобачевского, e-mail: dsmalyshev@rambler.ru

В работе рассматриваются только *обыкновенные графы*, т.е. неориентированные графы без петель и кратных ребер. Класс графов называется *наследственным*, если он замкнут относительно удаления вершин. Любой наследственный класс $\mathcal{X}$ (и только наследственный класс) графов может быть задан множеством своих *запрещенных порожденных подграфов* $\mathcal{Y}$, при этом принята запись $\mathcal{X} = \mathcal{F}ree(\mathcal{Y})$. *Сильно наследственный* (или *монотонный*) класс графов — наследственный класс, замкнутый еще и относительно удаления ребер. Любой монотонный класс $\mathcal{X}$ может быть задан множеством своих *запрещенных подграфов* $\mathcal{Y}$, при этом пишем $\mathcal{X} = \mathcal{F}ree_s(\mathcal{Y})$.

Реберной k-раскраской графа G называется произвольное отображение $c : E(G) \longrightarrow \{1, 2, \ldots, k\}$ такое, что $c(e_1) \neq c(e_2)$ для любых соседних ребер e_1 и e_2. Минимальное k, такое что существует реберная k-раскраска графа G, называется *хроматическим индексом* G и обозначается через $\chi'(G)$. *Задача о реберной k-раскраске* (кратко, *задача* k-PP) для заданного графа G состоит в том, чтобы проверить, выполняется ли неравенство $\chi'(G) \leq k$ или нет. *Задача о реберной раскраске* (кратко, *задача* PP) для заданных графа G и числа k состоит в том, чтобы проверить, выполняется ли неравенство $\chi'(G) \leq k$ или нет. Задачи 3-PP и PP являются классическими NP-полными задачами [1].

В работе [2] при любом k была получена полная сложностная дихотомия (т.е. полная классификация сложности) для задачи k-PP и всех классов вида $Free(\{H\})$. В статье [3] была получена полная классификация алгоритмической сложности задачи 3-PP для множеств запрещенных порожденных структур, каждая с не более чем 6 вершинами, среди которых не более двух имеют ровно 6 вершин. В работе [4] рассматривались задача PP и семейство монотонных классов, задаваемых запрещением подграфов, каждый из которых имеет не более чем 6 ребер или не более чем 7 вершин. В ней была получена полная классификация сложности задачи PP для классов графов из данного семейства. В работе [5] была получена полная классификация сложности задачи PP для монотонных классов, за-

даваемых запрещением подграфов, каждый из которых имеет не более чем 7 ребер. В настоящей работе в определенном смысле улучшается результат из [4]. Чтобы его сформулировать, нам понадобятся несколько определений.

Преобразование, называемое *заменой вершины треугольником*, хорошо известно. Оно применяется к вершине x графа, окрестность которой состоит в точности из вершин x_1, x_2, x_3, и определяется следующим образом. Удаляется вершина x, добавляются вершины x'_1, x'_2, x'_3 и ребра x'_1x_1, x'_2x_2, $x'_3x_3, x'_1x'_2, x'_2x'_3, x'_1x'_3$.

Определим преобразование *замены вершины (2,3)-бикликой*. Оно применяется к вершине x графа, окрестность которой состоит в точности из вершин x_1, x_2, x_3, и определяется следующим образом. Удаляется вершина x, добавляются вершины y_1, y_2 и ребра $x_1y_1, x_2y_1, x_3y_1, x_1y_2, x_2y_2, x_3y_2$.

Граф называется *кубическим*, если степени всех его вершин равны 3. Граф называется *субкубическим*, если степени всех его вершин не более чем 3.

Через $\mathcal{Z}_k$ обозначим множество кубических графов, не содержащих порожденных циклов длины до k включительно. Ясно, что $\mathcal{Z}_1$ и $\mathcal{Z}_2$ совпадают с множеством всех кубических графов. Обозначим через $\mathcal{Z}_k^*$ множество графов, которые получаются из графов класса $\mathcal{Z}_k$ последовательной заменой их вершин треугольниками. Через $\mathcal{Z}_k^{**}$ обозначим множество графов, которые получаются из графов класса $\mathcal{Z}_k$ последовательной заменой их вершин (2,3)-бикликами. Уточним, что в определениях $\mathcal{Z}_k^*$ и $\mathcal{Z}_k^{**}$ речь идет только о заменах вершин, принадлежащих графам из $\mathcal{Z}_k$. *Монотонное замыкание* класса графов $\mathcal{Z}$ — множество графов, являющихся подграфами графов из $\mathcal{Z}$. Оно обозначается через $[\mathcal{Z}]_s$.

Через O_n и P_n обозначаются пустой граф и простой путь на n вершинах. Через $G_1 + G_2$ обозначается дизъюнктное объединение графов G_1 и G_2 с непересекающимися множествами вершин.

Через T_1 обозначается дерево с множествами вершин и ребер

$$\{x_1, x_2, x_3, z_1, z_2, y_1, y_2, y_3\}, \{x_1x_2, x_2x_3, y_1y_2, y_2y_3, z_1z_2, x_2z_2, z_2y_2\},$$

соответственно. Через T_2, T_3, T_4 обозначаются деревья на множестве вершин $\{x_1, x_2, x_3, z_1, z_2, z_3, y_1, y_2, y_3\}$ с множествами ребер

$$\{x_1x_2, x_2x_3, y_1y_2, y_2y_3, x_2y_2, x_1z_1, y_1z_2, y_1z_3\},$$

$$\{x_1x_2, x_2x_3, y_1y_2, y_2y_3, x_2y_2, y_3z_1, y_1z_2, y_1z_3\},$$

$$\{x_1x_2, x_2x_3, y_1y_2, y_2y_3, x_2z_1, z_1z_2, z_1z_3, z_3y_2\},$$

соответственно. Справедливо следующее утверждение:

Теорема. *Пусть $\mathcal{Y}$ — множество графов, каждый из которых имеет не более чем 8 ребер. Тогда задача* PP *является полиномиально разрешимой для субкубических графов из $\mathcal{X} = Free_s(\mathcal{Y})$, если*

1. либо $\mathcal{Y}$ содержит субкубический лес, не принадлежащий множеству $A \cup B$, где

$$A = \{T_1 + P_2 + O_n : \ n \geq 0\} \cup \{T_2 + O_n : \ n \geq 0\} \cup \{T_4 + O_n : \ n \geq 0\},$$

$$B = \{T_3 + O_n : \ n \geq 0\}.$$

2. либо $\mathcal{Y}$ одновременно содержит граф из A и граф из $[\mathcal{Z}_4^*]_s$,
3. либо $\mathcal{Y}$ одновременно содержит граф из B и графы из $[\mathcal{Z}_4^*]_s$ и $[\mathcal{Z}_4^{**}]_s$.
Во всех остальных случаях она является NP*-полной для графов из $\mathcal{X}$.*

Данная теорема дает полную сложностную дихотомию для задачи о реберной раскраске и монотонных классов субкубических графов, заданных запрещенными подграфами с 8 ребрами. Кажется, что ограничение субкубичности в упомянутой теореме можно снять.

Работа выполнена при поддержке РФФИ и БРФФИ, проект № 20-51-04001 (Ф21РМ-001).

Список литературы

[1] Holyer I. The NP-completeness of edge-coloring // SIAM Journal on Computing. — 1981. — V. 10, № 4. — P. 718–720.

[2] Galby E., Lima P. T., Paulusma D., Ries B. Classifying k-edge colouring for H-free graphs // Information Processing Letters. — 2019. — V. 146. — P. 39–43.

[3] Malyshev D. S. The complexity of the edge 3-colorability problem for graphs without two induced fragments each on at most six vertices // Siberian Electronic Mathematical Reports. — 2014. — V. 11. — P. 811–822.

[4] Malyshev D. S. Complexity classification of the edge coloring problem for a family of graph classes // Discrete Mathematics and Applications. — 2017. V. 27, № 2. — P. 97–101.

[5] Malyshev D. S. Complete complexity dichotomy for 7-edge forbidden subgraphs in the edge coloring problem // Journal of Applied and Industrial Mathematics. — 2020. V. 14, № 4. — P. 706–721.

Анализ работы реализации квантового алгоритма построения деревьев решений на квантовом симуляторе

Маннапов Ильназ Магсумович[1], **Хадиев Камиль Равилевич**[2], **Сафина Лилия Ильхамовна**[3]

[1] Казанский федеральный университет, e-mail: ilnaztatar5@gmail.com
[2] Казанский федеральный университет, e-mail: kamilhadi@gmail.com
[3] Казанский федеральный университет, e-mail: liliasafina94@gmail.com

В данной работе мы проводим анализ результатов программной реализации квантовой версии построения деревьев решений на квантовом симуляторе. В работе [1] нами был предложен алгоритм построения деревьев решений, основанный на такой структуре данных, как сбалансированное дерево, и квантовом алгоритме поиска минимального (максимального) значения Дюрра-Хойера [2], за счёт чего мы добились асимптотического ускорения алгоритма. Однако любой квантовый алгоритм является вероятностным алгоритмом, из-за чего деревья, построенные классической и нашей квантовой версиями, могут отличаться. В связи с чем мы решили написать программную реализацию нашего алгоритма, построили дерево решений на открытом множестве данных для обучения, проверили схожесть построенных деревьев, и получили интересные результаты.

Введение

Алгоритмы машинного обучения работают с большими данными и требуют мощных вычислительных ресурсов, в связи с чем скорость работы алгоритма имеет большое значение. Будущие квантовые компьютеры потенциально полезны для обработки больших объёмов данных.

Цель нашей работы — провести анализ полученных классическим и квантовым способами построения деревьев решений. Модели деревьев решений [3] на сегодняшний день не являются актуальными способами решений для задач машинного обучения. Однако деревья решений используются в ансамблевых моделях. Например, такие алгоритмы, как случайный лес и градиентный бустинг [4] строятся на классических деревьях решений и используются в решениях реальных задач, поэтому ускорение алгоритма в будущем может быть полезно для индустрии.

Идея алгоритма и реализация

Весь теоретический подход нашего алгоритма описан в работе [1], где мы предлагаем квантовую версию алгоритма построения деревьев решений для задач классификации. Для построения каждого узла дерева алгоритму необходимо выбрать атрибут и его значение порога разбиения обучаю-

щего множества на подмножества, используя некий критерий разбиения. Алгоритм максимизирует функцию критерия разбиения. Самые популярные критерии основаны на энтропии, что является мерой неопределённости количества информации в данных. Для подсчёта функции критерия разбиения алгоритм каждый раз итеративно вычисляет некоторые значения, к примеру, количество элементов подмножества, относящиеся к каждому классу, их значение функции смешанности и другие, для которых мы предлагаем сделать предподсчёт и хранить их в сбалансированном дереве, что позволит ускорить процесс получения необходимых значений даже в классическом случае. Для максимизации критерия отбора мы применяем алгоритм поиска максимального значение Дюрра-Хойера.

В данной работе мы реализовали классический алгоритм построения деревьев решений без применения классического ускорения с предподсчётом данных и хранением их в бинарном дереве. Эксперимент был реализован на языке Kotlin. В качестве обучающей выборки была взята открытая база данных с описательными признаками абалонов (род моллюсков), где предлагается предсказать их возраст. В представленной базе данных для обучения более 4000 данных с категориальными и числовыми значениями.

Нами были реализованы базовый классический алгоритм построения деревьев решений C5.0 [5] и его квантовое улучшение, где поиск максимального значения функции критерия разбиения реализован с помощью алгоритма Дюрра-Хойера. Алгоритм максимизации функции критерия разбиения запускается константное количество раз, чтобы увеличить вероятность схожести деревьев. Однако даже при таких условиях мы получаем временное ускорение. Мы не использовали существующие квантовые симуляторы, так как наш алгоритм состоит как из классических вычислений, так и квантовых подпрограмм, поэтому было принято решение самостоятельно написать симуляцию работы квантового алгоритма Дюрра-Хойера. К сожалению, на симуляторе мы не можем оценить фактическое временное ускорение работы алгоритма. Теоретически ускорение квантовой версии доказано нами в работе [1] вместе с теоретической вероятностью совпадения классических и квантовых деревьев.

Результаты

Для оценки схожести деревьев мы использовани отношение совпадающих вершин, полученных квантовым способом и классическим, к общему количеству вершин:

$$Q_{tree} = \frac{B_{eq}}{B},$$

где B — общее количество вершин, B_{eq} — количество совпадающих вершин.

Критерий остановки построения дерева зависит от самих данных, а также заданной высоты дерева. В эксперименте мы построили деревья с $h = 3$, $h = 5$, $h = 7$, $h = 10$.

Мы получили следующие значения Q_{tree}:

Высота дерева h	Q_{tree}
$h = 3$	1
$h = 5$	1
$h = 7$	1
$h = 10$	0.99

Результаты таблицы демонстрируют, что наш квантовый алгоритм выстраивает дерево максимально схожее на дерево, построенное классическим способом.

Заключение

Нами предложен алгоритм с использованием квантовой подпрограммы максимизации функции критерия разбиения множества данных для обучения на подмножества в каждой вершине дерева. Однако мы сомневались в схожести построенных деревьев квантовым и классическим способами из-за вероятностного поведения квантовой системы. Эксперимент доказал, что наш квантовый алгоритм выстраивает такое же дерево с гиперпараметром высотой дерева $h = 3$, $h = 5$ и $h = 7$ и довольно близкое с высотой $h = 10$. Мы теоретически доказали асимптотическое ускорение нашего алгоритма. Такие результаты подтверждают, что будущие квантовые компьютеры могут быть полезны на серверах, используемых для задач классификации с большими данными для обучения.

Работа выполнена за счёт средств субсидии, выделенной Казанскому федеральному университету для выполнения государственного задания в сфере научной деятельности, проект 0671-2020-0065.

Список литературы

[1] Khadiev K, Mannapov I, Safina L., Classical and quantum improvements of generic decision tree constructing algorithm for classification problem// CEUR Workshop Proceedings. Vol.2842, Is.. - P.83-93, — 2021

[2] Quinlan J. R. , Induction of decision trees. // Machine learning, —1986.

[3] Durr C., Høyer P., A quantum algorithm for finding the minimum // arXiv:quant-ph/9607014, — 1996.

[4] Ting K. M., Witten I. H. Stacked Generalization: when does it work?. — 1997.

[5] C5.0: An informal tutorial. // https://www.rulequest.com/see5-unix.html, — 2019.

О сложности схем ограниченной толщины для некоторых вычислительных задач

Медведев Дмитрий Сергеевич

Московский государственный университет имени М. В. Ломоносова, e-mail: diman7771111991@ya.ru

Исследуется схемная сложность для двух задач — задачи возведения в степень (см., например, в [1] задачу об аддитивных цепочках) и задачи сборки двоичных слов (см., например, [2]) — при константных ограничениях на число одновременно хранящихся результатов промежуточных вычислений.

Пусть $L_e(x^n)$ — минимальное число операций умножения, достаточное для вычисления по переменной x степени x^n, $L_{ed}(x^n)$ — минимальное число операций умножения и деления, достаточное для вычисления по переменной x степени x^n, а $L_c(\tilde{\alpha})$ — минимальное число операций конкатенации (склейки), достаточное для вычисления двоичного слова $\tilde{\alpha}$, исходя из символов «0» и «1»

Положим

$$L_e(n) = \max_{k:\ k\le n} L_e(x^k), \quad L_{ed}(n) = \max_{k:\ k\le n} L_{ed}(x^k), \quad L_c(n) = \max_{\tilde{\alpha}:\ |\tilde{\alpha}|=n} L_c(\tilde{\alpha}).$$

Асимптотическое поведение при $n \to \infty$ функций Шеннона $L_e(n)$, $L_{ed}(n)$ и $L_c(n)$, характеризующих соответствующие сложности вычисления при отсутствии ограничений на используемую память, известно (см., например, [3 , 4]):

$$L_e(n) \sim \log_2 n, \quad L_{ed}(n) \sim \log_2 n, \quad L_c(n) \sim \frac{n}{\log_2 n}.$$

Пусть t — фиксированный натуральный параметр. Обозначим через $L_e^{(t)}(x^n)$ (соответственно $L_{ed}^{(t)}(x^n)$) минимальное число операций умножения (умножения и деления), достаточное для вычисления по переменной x степени x^n в случае, когда при вычислении в каждый момент времени хранится для дальнейшего использования не более t степеней (при этом считается, что сама переменная x всегда доступна и не требует хранения).

Через $L_c^{(t)}(\tilde{\alpha})$ обозначим минимальное число операций конкатенации, достаточное для сборки слова $\tilde{\alpha}$ схемами конкатенации в случае, когда при вычислении в каждый момент времени хранится для дальнейшего использования не более t слов (при этом считается, что символы «0» и «1» всегда доступны и не требуют хранения).

Положим

$$L_e^{(t)}(n) = \max_{k:\ k\le n} L_e^{(t)}(x^k), \quad L_{ed}^{(t)}(n) = \max_{k:\ k\le n} L_{ed}^{(t)}(x^k), \quad L_c^{(t)}(n) = \max_{\tilde{\alpha}:\ |\tilde{\alpha}|=n} L_c^{(t)}(\tilde{\alpha}).$$

Для любого натурального n положим

$$\alpha(n) = \begin{cases} 2, & \text{если найдется такое } p, \text{ что } 2^p \leqslant n \leqslant 2^p + 2^{p-1} - 1; \\ 1, & \text{если найдется такое } p, \text{ что } 2^p + 2^{p-1} - 1 \leqslant n \leqslant 2^{p+1} - 1; \\ 0, & \text{если найдется такое } p, \text{ что } n = 2^{p+1} - 1. \end{cases}$$

С использованием результатов работ [5, 6] установлены некоторые оценки сложности вычислений в описанных моделях при константных ограничениях на используемую память.

Теорема. *Справедливы следующие утверждения:*

1. *Для задачи сборки двоичных слов схемами конкатенации при* $t = 1$ *выполняется равенство*

$$L_c^{(1)}(n) = \begin{cases} n - 1, & \textit{если } 1 \leqslant n \leqslant 18; \\ n - 2, & \textit{если } n \geqslant 19. \end{cases}$$

2. *Для задачи сборки двоичных слов схемами конкатенации при* $t = 2$ *верны следующие оценки сложности*

$$\frac{n}{\log_2 24} \leqslant L_c^{(2)}(n) \leqslant \frac{4n}{5} + O(1).$$

3. *Для задачи возведения в степень в модели «умножение» при* $t = 1$ *выполняются равенства*

$$L_e^{(1)}(x^n) = \lfloor \log_2 n \rfloor + (N_1 - 1), \quad L_e^{(1)}(n) = 2 \cdot \lfloor \log_2 n \rfloor - \alpha(n),$$

где N_1 *— количество единиц в двоичном разложении числа* n.

4. *Для задачи возведения в степень в модели «умножение» при* $t = 2$ *верна следующая верхняя оценка сложности:*

$$L_e^{(2)}(n) \leqslant \frac{3}{2} \log_2 n + O(1).$$

5. *Для задачи возведения в степень в модели «умножение» при любом фиксированном* $t \geq 2$ *найдётся такое* $\varepsilon = \varepsilon(t) > 0$, *что при* $n \to \infty$ *выполняется асимптотическое соотношение*

$$L_e^{(t)}(n) \gtrsim (1 + \varepsilon) \log_2 n.$$

6. *Для сложности возведения в степень в модели «схемы умножения и деления» при* $t = 1$ *имеет место следующее равенство:*

$$L_{ed}^{(1)}(x^n) = \lfloor \log_2 n \rfloor + p(n) + r(n),$$

где через $p(n), r(n)$ *обозначены следующие характеристики двоичного представления числа* n *без учета старшего разряда:* $p(n)$ *— число блоков единиц,* $r(n)$*— число блоков, содержащих не менее* 2*-х единиц.*

7. *Для задачи возведения в степень в модели «схемы умножения и деления» при* $t = 1$ *имеет место следующее равенство:*

$$L_{ed}^{(1)}(n) = 3\lfloor \log_8 n \rfloor + \beta(n),$$

где $\beta(n)$ *— некоторая величина, удовлетворяющая неравенству* $-1 \leqslant \beta(n) \leqslant 2$.

Замечание 1. *Нижняя оценка в пункте* 2 *теоремы является следствием более общей оценки: при любом фиксированном* $t \geqslant 2$ *справедливо неравенство*

$$L_c^t(n) \geqslant \frac{n}{\log_2(t^3 + 8t)}.$$

Замечание 2. *Точное значение величины* $\beta(n)$ *из пункта* 7 *теоремы установлено, однако аккуратное задание этой величины громоздко (содержит разбор* 8 *случаев) и поэтому в данной работе опущено.*

Список литературы

[1] Кнут Д. Е. Искусство программирования для ЭВМ, т. 2. 1-е издание. — М.: Мир, 1977.

[2] Мерекин Ю. В. Нижняя оценка сложности для схем конкатенации слов // Дискретный анализ и исследование операций. — 1996. — Т. 3, № 1. — С. 52–56.

[3] Brauer A. On addition chains // Bull. Amer. Math. Soc. — 1939. — V. 45. — P. 736–739.

[4] Кочергин В. В., Кочергин Д. В. Уточнение асимптотического поведения сложности сборки слов схемами конкатенации // Вестник Московского университета. Сер. 1. Математика. Механика. — 2016, № 2. — С. 12–18.

[5] Aviezri S. Fraenkel, R. Jamie Simpson. How many squares must a binary sequence contain? // The electronic journal of combinatorics. — 1995. — V. 2. — #R.

[6] Балакин К. С. Нетривиальная верхняя оценка сложности возведения в степень с использованием 4 ячеек памяти // Материалы VIII молодежной научной школы по дискретной математике и ее приложениям (Москва 24–29 октября 2011 г.). Часть I. — С. 7–10.

О хроматическом числе 2-псевдокубических графов

Мельник Марина Владимировна

Московский Государственный Университет имени М.В. Ломоносова, e-mail: melnikmv@cs.msu.ru

Рассматривается задача 3-раскраски вершин графа. Она состоит в том, чтобы выяснить можно ли вершины графа раскрасить так, чтобы у смежных вершин были различные цвета. Сейчас задачу 3-раскраски решают для графов, на структуру которых накладывают различные ограничения. Например, ограничения на степени вершин или запрет порожденных подграфов определенной структуры [1 – 3]. Граф $G = (V, E)$ назовем псевдокубическими, если степени всех его вершин за исключением одной не превосходят 3-х, а степень исключительной не превосходит 4-х. Такие графы являются расширением понятия субкубических графов, то есть графов у которых все степени вершин не превосходят 3-х. Известно, что любой субкубический граф, не содержащий подграфов K_4, можно раскрасить в 3 цвета [4]. В [5] показано, что любой псевдокубический граф с одной исключительной вершиной можно раскрасить в три цвета, если он не содержит подграфов K_4 и K_4^-. Графом S назовем граф изображенный на рисунке 1.

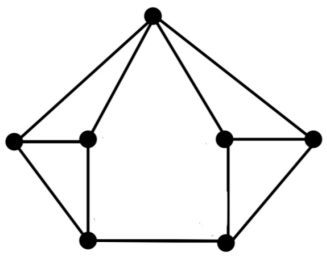


Рис. 1: Граф S.

В данной работе меняется ограничение на псевдокубические графы, а именно ограничение на подграфы K_4^- заменяется на другое.

Теорема 1. *Если $G = (V, E)$ связный псевдокубический граф с исключительной вершиной v_0, $d_G(v_0) \leqslant 4$, не содержащий подграфов K_4 и S, то $\chi(G) \leqslant 3$.*

Далее в работе рассматривается ослабление структуры графа, а именно количество исключительных вершин увеличивается. Граф $G = (V, E)$ назовем 2-псевдокубическими, если степени всех его вершин за исключением двух не превосходят 3-х, а степени исключительных не превосходят 4-х. Для 2-псевдокубических графов получен следующий результат.

Теорема 2. *Если $G = (V, E)$ — связный 2-псевдокубический граф с исключительными вершинами $v_0, u_0 \in V$, $d_G(u_0) \leqslant 4$ и $d_G(v_0) \leqslant 4$, не содержащий подграфов K_4 и K_4^-, то $\chi(G) \leqslant 3$.*

Статья опубликована при финансовой поддержке Минобрнауки РФ в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2019-1621 («Оценки сложностных характеристик булевых функций и графов»).

Список литературы

[1] Kochol M., Lozin V. V., Randerath B. The 3-colorability problem on graphs with maximum degree four // SIAM J. Comput. — 2003. — V. 32.

[2] Malyshev D. The complexity of the 3-colorability problem in the absence of a pair of small forbidden induced subgraphs // Discrete Mathematics. — 2015. — Vol. 338, No. 11. — P. 1860–1865.

[3] Sirotkin D., Malyshev D. On 3-colouring of graphs with short faces and bounded maximum vertex degree // Lobachevskii Journal of Mathematics. — 2021. —Vol. 42, No. 4. — P. 760–766.

[4] Brooks R. L. On coloring the nodes of network // Proc. Cambridge Philos. Soc. — 1941. — Vol. 37. — P. 194–197.

[5] Селезнева С. Н., Мельник М. В., Астахова А. В. Раскраска в три цвета псевдокубических графов // Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. — 2019. — № 2. — С. 39–45.

Отношения делимости чисел и включения замкнутых классов многозначных функций

Мещанинов Дмитрий Германович

НИУ «МЭИ», e-mail: MeshchaninovDG@mpei.ru

k-значная логика

Рассматриваются замкнутые классы k-значной логики, характеризуемые зависящими от делителей числа k аддитивными формулами, т. е. суммами с однозначно определенными слагаемыми [1]. В качестве слагаемых часто

используются так называемые *d-периодические функции*, они удовлетворяют условию

$$\tilde{a} \equiv \tilde{b} \pmod{d} \Rightarrow f(\tilde{a}) = f(\tilde{b}).$$

Будем применять следующие обозначения:
$E_m = \{0, 1, \ldots, m-1\}$ для m;
$l(\tilde{x}) = a_0 + a_1 x_1 + \cdots + a_n x_n$ для линейной функции;
L для замкнутого класса всех линейных функций;
$G_d(\tilde{x})$ для d-периодической функции;
$d \cdot F(\tilde{x})$ для функции, все значения которой кратны d;

$$g_d(\tilde{x}) = \begin{cases} 1, & \tilde{x} \equiv \tilde{0} \pmod{d}, \\ 0, & \tilde{x} \not\equiv \tilde{0} \pmod{d}, \end{cases} \qquad \chi_{d,j}(\tilde{x}) = x_j g_d(x_1, \ldots, x_j, \ldots, x_n).$$

Все рассматриваемые классы являются подклассами в классах $C(d)$ сохранения сравнений по модулю d, $d|k$. Они определяются наиболее общими аддитивными формулами

$$f(\tilde{x}) = l(\tilde{x}) + G_d(\tilde{x}) + d \cdot F(\tilde{x}).$$

Если d собственный делитель числа k, то каждый класс $C(d)$ является предполным в P_k. Также имеют место равенства $C(1) = C(k) = P_k$. Рассмотрим некоторые семейства подклассов в $C(d)$.

1. Классы $C_e(d)$, где $e|d$, $d|k$ [1] состоят из функций вида

$$f(\tilde{x}) = l(\tilde{x}) + eG_d(\tilde{x}) + d \cdot F(\tilde{x}).$$

При фиксированном d классы $C_e(d)$ образуют решетку, антиизоморфную решетке делителей e числа d ($e_1|e_2 \Leftrightarrow C_{e_1}(d) \supseteq C_{e_2}(d)$). Максимумом решетки является класс $C_1(d) = C(d)$, минимумом — класс $C_d(d)$ функций вида

$$f(\tilde{x}) = l(\tilde{x}) + d \cdot F(\tilde{x}).$$

2. Классы $R(d)$ сохранения d-разностей [2] состоят из функций вида

$$f(\tilde{x}) = l(\tilde{x}) + G_d(\tilde{x}) + H_d(\tilde{x}), \quad H_d(\tilde{x}) = \sum_{\mu \in E_d^n} \sum_{j=1}^{n} a_j(\mu) \chi_{d,j}(\tilde{x} - \tilde{\mu}). \qquad (1)$$

3. Классы $L(d)$ абсолютного сохранения d-разностей [2] состоят из функций вида

$$f(\tilde{x}) = l(\tilde{x}) + G_d(\tilde{x}).$$

Имеют место включения $L \subseteq L(d) \subseteq R(d) \subseteq C(d)$.

Классы $R(d)$ образуют решетку, изоморфную решетке делителей d числа k. Такую же решетку образуют классы $L(d)$ (условия $d_1|d_2$, $R(d_1) \subseteq R(d_2)$, $L(d_1) \subseteq L(d_2)$ эквивалентны). Минимумом решетки является класс $R(1) = L(1) = L$, максимумом — класс $L(k) = R(k) = C(k) = P_k$.

4. Классы $K(e, d)$ при $d|k, e|k$ [3] состоят из функций вида

$$f(\tilde{x}) = l(\tilde{x}) + eG_d(\tilde{x}) + d \cdot F(\tilde{x}).$$

При фиксированном d и условии $e|d$ классы $K(e, d)$ образуют решетку, антиизоморфную решетке делителей e числа d. При фиксированном e и условии $e|d$ классы $K(e, d)$ образуют решетку, антиизоморфную решетке делителей d числа k, кратных e.

5. Классы $S(d) = C_d(d) \cap R(d)$, $d|k$, состоят из функций вида

$$f(\tilde{x}) = l(\tilde{x}) + d \cdot G_d(\tilde{x}) + H_d(\tilde{x}),$$

где функция $H_d(\tilde{x})$ определена в (1).

Если $d_1|d_2$, то классы $S(d_1)$ и $S(d_2)$ не сравнимы по включению.

6. Пусть $e|d, d|k$. Введем классы $S_e(d) = S(d) \cap R(e) = C_d(d) \cap R(e)$ всех функций вида

$$f(\tilde{x}) = l(\tilde{x}) + d \cdot G_e(\tilde{x}) + \frac{d}{e} \cdot H_e(\tilde{x}).$$

При фиксированном d классы $S_e(d)$ образуют решетку, антиизомрорфную решетке делителей e числа d.

Счетнозначная логика

Решетки, изоморфные решетке $\mathbb{N}$ с отношением делимости, образуют и замкнутые классы в функциональных системах $P(\mathbb{Z})$ всех полиномов над кольцом $\mathbb{Z}$) и $L(\mathbb{Z})$ полиномов первой степени , а также $L(\mathbb{N}_0)$ полиномов первой степени с неотрицательными коэффициентами. Эти системы — замкнутые классы в счетнозначной логике.

7. Пусть $k \in \mathbb{Z}_+$. В функциональной системе $P(\mathbb{Z})$ рассмотрим замкнутые классы $F(k)$ полиномов со свободным коэффициентом, кратным k.

8. Пусть $k \in \mathbb{Z}_+$, $b \in \{0, 1, 2, \ldots\}$. В функциональной системе $L(\mathbb{Z})$ рассмотрим замкнутые классы $U(b, k)$ [4,5] сохранения множеств

$$\{c \in \mathbb{Z} \mid c \equiv b \pmod{k}\}.$$

9. Пусть $k \in \mathbb{Z}_+$. В функциональной системе $L(\mathbb{Z})$ рассмотрим замкнутые классы [4,5]

$$S(k) = \{a_0 + a_1x_1 + \cdots + a_nx_n \mid a_1 + \cdots + a_n \equiv 1 \pmod{k}\}.$$

10. Пусть $d \in \mathbb{N}$, $m \in \mathbb{Z}_+$. В функциональной системе $L(\mathbb{N}_0)$ рассмотрим замкнутые классы [6]

$$\Sigma_1(dm) = [\mathbb{N} \cup \{0, x, x + dm\}].$$

Условия $d_1|d_2$ и $\Sigma_1(d_1m) \supseteq \Sigma_1(d_2m)$ эквивалентны.

Классы $F(k)$, $U(b, k)$ при фиксированном b, $S(k)$, $\Sigma_1(dm)$ при фиксированном m образуют решетки, антиизоморфные решетке $(\mathbb{N}; |)$.

Список литературы

[1] Мещанинов Д. Г. Некоторые семейства замкнутых классов в P_k, задаваемых аддитивными формулами // Дискретная математика. — 2021. — Т. 33, вып. 2. — С. 100–116.

[2] Мещанинов Д. Г. О первых d-разностях функций k-значной логики // Математические вопросы кибернетики. Вып. 7. — М.: Наука, 1998. — С. 265–280.

[3] Мещанинов Д. Г. Об одном семействе замкнутых классов в k-значной логике // Вестник Московского ун-та. Сер.15. Выч. математика и кибернетика. — 2019, №1. — С. 44–53.

[4] Мамонтов А. И., Мещанинов Д. Г. Проблема полноты в функциональной системе линейных полиномов с целыми коэффициентами // Дискретная математика. — 2010. — Т. 22, вып. 4. — С. 64–82.

[5] Мамонтов А. И. О некоторых решетках замкнутых классовв функциональной системе линейных полиномов с целыми коэффициентами // Труды IX Междунар. конф. «Дискретные модели в теории управляющих систем» 20–22 мая 2015 г. Москва и Подмосковье. — М.: МАКС Пресс, 2015. — С. 154–156.

[6] Мещанинов Д. Г., Никитин И. В. Классы сохранения пороговых разбиений в функциональных системах полиномов // Вестник МЭИ. — 2012, №6. — С. 132–141.

Построение проективной плоскости генетическим алгоритмом

Наумов Илья Евгеньевич, Хворостухина Екатерина Владимировна

Саратовский государственый технический университет имени Гагарина Ю.А., e-mail: ilya.naumov.99@mail.ru, khvorostukhina85@gmail.com

Задача построения проективной плоскости не является тривиальной (см., например, [1]). В частности, предложенный в работе [2] алгоритм

построения p-гиперграфов и плоскостей, как их частного случая, имеет факториальную сложность. Открытой остается и проблема существования плоскостей многих порядков. Например, в 1989 с помощью компьютера было доказано [3] несуществование плоскости 10 порядка, что заняло около 2000 часов компьютерного времени. Мы продолжаем исследования в данном направлении и предлагаем использовать для построения проективной плоскости генетический алгоритм [4], который позволяет найти решение задачи поиска глобального минимума или максимума даже для NP-полных задач за относительно небольшое число итераций. Генетические алгоритмы применяются во многих областях науки и техники: в биологии, в финансах, в робототехнике. Генетические алгоритмы не гарантируют нахождения решения за полиномиальное время [4], но они бывают весьма полезны при решении задач с большим пространством поиска. В простейшей вариации, генетический алгоритм включает следующие операторы: селекция (данный оператор выбирает из популяции хромосомы для дальнейшего скрещивания, вероятность выбора хромосом зависит от функции приспособленности (целевой функции) — чем значение функции лучше, тем вероятнее соответствующая хромосома будет выбрана), кроссинговер (данный оператор отвечает за обмен частей хромосом для создания потомка), мутация (данный оператор симулирует проявление мутаций: с некоторой, обычно достаточно малой вероятностью, в хромосоме случайный ген может изменить свое значение).

Согласно [5], конечная проективная плоскость порядка q содержит $q^2 + q + 1$ линий (прямых) и $q^2 + q + 1$ точек, где $q \in N$, $q > 1$, удовлетворяющих следующим аксиомам: (A1) каждая линия содержит $q + 1$ точек; (A2) каждая точка инцидентна $q + 1$ линии; (A3) любые две различные прямые пересекаются ровно в одной точке; (A4) любые две точки лежат только на одной прямой. Обозначим $n = q^2 + q + 1$.

Мы предлагаем следующую схему построения проективной плоскости порядка q, основанную на генетическом алгоритме. Во-первых, необходимо определить способ кодирования решения (хромосому). Итак, хромосома представляет из себя матрицу инцидентности размерности $n \times n$. Строкам матрицы соответствуют прямые, а столбцам — точки плоскости. Соответственно, в каждой строке матрицы отображены инцидентные ей точки, а в каждом столбце матрицы — инцидентные этой точке прямые. Строки матрицы представляют из себя случайные наборы нулей и единиц, с тем условием, что выполняется аксиома (A1) — каждая прямая инцидентна ровно $q + 1$ точке, в данном случае это значит, что в каждой строке ровно $q + 1$ единиц. Учитывая то, что при кроссинговере будет сохраняться выполнимость аксиомы (A1), то целевая функция подсчитывает количество

нарушений аксиом (A2)—(A4). Когда значение целевой функции равно 0, то это означает, что проективная плоскость построена. Далее определим селекцию (отбор) родителей. В данном случае используется дополнительный гиперпараметр n_c, который определяет, что n_c лучших особей (матрицы с минимальным значением целевой функции) будут выбраны для участия в кроссинговере. Все выбранные кандидаты попадут в следующее поколение. Данный выбор селекции обусловлен тем, что часто бывает полезно сохранять наиболее успешных особей в будущих поколениях. Значение этого параметра зависит от конкретной задачи.

В предлагаемом алгоритме нами был выбран одноточечный кроссинговер. Точка кроссинговера определяется случайным образом. После чего, первая часть матрицы инцидентности (первые строки матрицы) потомка принадлежит первому родителю, до выбранной точки кроссинговера, вторая же часть матрицы берется от второго родителя. В том случае, когда кроссинговер не произошел, потомок оказывается точной копией одного из родителей. Вероятность кроссинговера определяется гиперпараметром p_c. Обычно данному параметру присваивают как можно большее значение. В разрабатываемой программе $p_c \in [0.9, 0.99]$.

Для сохранения выполнимости аксиомы (A1), мутация реализована следующим образом: в пределах одной случайно выбранной прямой выбираются две случайные точки, после чего выбранные точки меняются местами. В этом случае, аксиома (A1) не нарушается. Вероятность мутации определяется гиперпараметром p_m. Данная вероятность обычно принимает небольшие значения. В разрабатываемой программе она берется из отрезка $[0.01, 0.2]$.

На данный момент написанная на языке программирования Python 3.9 программа строит проективные плоскости малых порядков. На рисунке 1 изображена зависимость значения целевой функции от числа поколений на примере построения плоскости порядка 2. В данном случае были использованы следующие гиперпараметры: число поколений, равное 100, численность популяции, равная 10000, число лучших особей для образования нового поколения, равное 3000, вероятность кроссинговера и мутации, равные 0.95 и 0.2, соответственно.

Далее планируется усовершенствовать данный алгоритм, эмпирически подобрать более оптимальные гиперпараметры, а также модифицировать разработанный алгоритм для построения p-гиперграфов [2].

Список литературы

[1] Картези Ф. Введение в конечные геометрии. — М. : Наука, 1980. — 320 с.

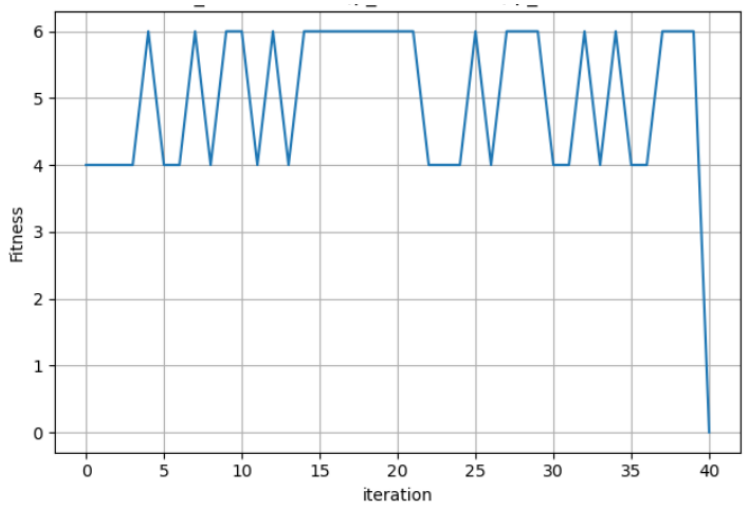


Рис. 1: Зависимость значения целевой функции от числа поколений

[2] Курбатов М. С, Молчанов В. А., Хворостухина Е. В. Разработка алгоритма построения p-гиперграфов // Математика. Механика : сб. науч. тр. — 2019. — Вып. 21. — С. 38–41.

[3] Lam C. W. H. The search for a Finite Projective Plane of Order 10 // The American Mathematical Monthly. — 1991. — Vol. 98, № 4. — P. 305–318.

[4] Батищев Д. И., Неймарк Е. А., Старостин Н. В. Применение генетических алгоритмов к решению задач дискретной оптимизации. — Н.Новогород. : НИ НГУ им. Н.И. Лобачевского, 2006. — 136 с.

[5] Хартсхорн Р. Основы проективной геометрии. — М. : Наука, 1970. — 160 с.

Свободное наращение матроидов и слабые отображения

Ревякин Александр Михайлович

Национальный исследовательский университет МИЭТ, e-mail: arevyakin@mail.ru

Матроидом (см. [1, 2]) называется система $\mathcal{I}$ подмножеств конечного множества S, удовлетворяюшая условиям: (1) $\oslash \in \mathcal{I}$; (2) если $A \subseteq B$ и $B \in \mathcal{I}$, то $A \in \mathcal{I}$; (3) если $A, B \in \mathcal{I}$ и $|A| > |B|$, то найдется $a \in A \setminus B$ такое, что $B \cup b \in \mathcal{I}$. Множества из $\mathcal{I}$ в M называются независимыми, максимальные по включению независимые множества – базами. Ранговой функцией называется целочисленная функция $r(A)$, определенная для всех $A \subseteq S$, такая, что $r(A) = \max\{|X| : X \subseteq A \text{ и } X \in \mathcal{I}\}$.

Конечное множество S с оператором замыкания $A \to \overline{A}$ ((1) $A \subseteq \overline{A}$ для всех $A \subseteq S$; (2) если $A, B \subseteq S$ и $A \subseteq B$, то $\overline{A} \subseteq \overline{B}$; (3) $\overline{A} = \overline{\overline{A}}$ для

всех $A \subseteq S$), удовлетворяющим свойству замены: (4) для любых $x, y \in S$ и всякого $A \subseteq S$ из $y \in \overline{A \cup \{x\}}$ и $y \notin \overline{A}$ следует, что $x \in \overline{A \cup \{y\}}$, образует матроид $(S, ^-)$. Причем семейство $\mathcal{I}$ подмножеств $A \subseteq S$ таких, что из $x \in A$ следует, что $x \notin A \setminus \{x\}$, образует матроид $M = (S, \mathcal{I})$.

Если $\overline{A} = A$, то A называется замкнутым (или поверхностью) в M. Пара $(S, \mathcal{F})$, где $\mathcal{F}$ – семейство поверхностей из S образует матроид, если: (1) $S \in \mathcal{F}$; (2) если $F_1, F_2 \in \mathcal{F}$, то $F_1 \cap F_2 \in \mathcal{F}$; (3) если $F_1, F_2, \ldots, F_k \in \mathcal{F}$ и F_i покрывает F (т.е. $F_i \supseteq F$ и не существует поверхности $F' \in \mathcal{F}$ такой, что $F_i \supset F' \supset F$) для всех i, $i = 1, 2, \ldots, k$, то $\{F_1 \setminus F, \ldots, F_k \setminus F\}$ – разбиение множества $S \setminus F$. Причем семейство $\mathcal{I}$ подмножеств $A \subseteq S$ таких, что для всех $a \in A$ найдется подмножество $F \in \mathcal{F}$, для которого $A \setminus F = \{a\}$, образует матроид $(S, \mathcal{I})$.

Комбинаторной геометрией [3] (далее: геометрией) называется матроид, у которого все одноэлементные подмножества, а также пустое множество являются замкнутыми. Задание комбинаторной геометрии на множестве S отличается от задания топологии на множестве S тем, что для замыкания не требуется выполнения условия $\overline{A} \cup \overline{B} = \overline{A \cup B}$ для всех $A, B \subseteq S$, но имеет место свойство (4), которое, вообще говоря, может не выполняться для замыкания топологии.

Решетка L называется геометрической [4], если L – полумодулярная алгебраическая решетка, в которой компактными элементами являются конечные объединения атомов и только они. Если $G = (S, ^-)$ – матроид и $\mathcal{L}(G)$ – множество всех его поверхностей, упорядоченных по включению, то $\mathcal{L}(G)$ является геометрической решеткой, в которой $A \vee B = \overline{A \cup B}$ и $A \wedge B = A \cap B$, для всех $A, B \in \mathcal{L}(G)$.

Пусть $M = (S, \mathcal{I})$ – матроид на множестве S с ранговой функцией $r(A)$, $A \subseteq S$, и k-натуральное число, $0 < k \leq r(S)$. Тогда семейство $\mathcal{I}_k = \{A \subseteq S : A \in \mathcal{I} \text{ и } |A| \leq k\}$ – семейство независимых множеств некоторого матроида M_k на множестве S. Матроид M_k называется $k-$усечением матроида M. Решетка поверхностей M_k получается из решетки поверхностей матроида M вычеркиванием всех поверхностей ранга $\geq k$ и заменой всех их единственным максимальным элементом решетки. Если некоторый матроид M на множестве S изоморфен $(r(H) - 1)-$усечению матроида H на том же множестве S, то говорят, что H есть наращение матроида M. Решетка поверхностей наращения H получается из решетки поверхностей матроида M включением уровня новых коатомов, который лежит между старыми коатомами и единичным элементом решетки.

Подмножество $A \subseteq S$ матроида M называется $k-$замкнутым, если оно содержит замыкания всех его $j-$элементных подмножеств для всякого $j \leq k$. Опрелелим $k-$замыкание множества A как наименьшее $k-$замкну-

тое подмножество множества S, содержащее A. Подмножество A частично упорядоченного множества P называется антицепью, если A тривиально упорядочено в P, т.е. если $x, y \in A$ и $x \leq y$, то $x = y$. Множество антицепей в P, будем обозначать $\mathcal{A}(P)$. Если $A, B \in \mathcal{A}(P)$, то $A \leq B$ тогда и только тогда, когда всякого $x \in A$ найдется $y \in B$ такой, что $x \leq y$. Антицепь A разбивает B, если каждый $x \in B$ ограничен сверху единственным элементом $y \in A$. При это A разбивает B замкнуто, если каждый элемент из A представим как решеточный супремум элементов, им ограниченных в B.

Г. Крапо [5] установил, что необходимыми и достаточными условиями для того, чтобы множество подмножеств A множества S матроида $M = (S, \mathcal{I})$ ранга r было наращением, являются:

а) A – антицепь в булевой алгебре 2^S;

б) все элементы A являются $r-1$ замкнутыми в матроиде M;

в) антицепь A разбивает антицепь всех баз матроида M.

Различные наращения решетки матроида M, упорядоченные как антицепи образуют полную решетку. Наименьший элемент этой решетки называется свободным наращением [5, 6] матроида M. Если свободное наращение матроида M известно, то все другие наращения могут быть получены путем разбиения свободного наращения.

Пусть $A, B, C \in \mathcal{A}(2^S)$, φ – оператор, A в $C \leq A$, являющуюся наименьшей среди элементов разбивающих антицепь T всех баз матроида $M(S)$, а ψ – оператор, отображающий A в B, элементами которой являются $(r-1)-$замыкания элементов антицепи A. Нетрудно заметить, что операторы φ и ψ являются операторами замыкания.

Теорема 1. *Пусть $M(S)$ – матроид ранга r на конечном множестве S, T – антицепь всех баз матроида $M(S)$ и $\varphi\psi(T) \neq S$. Тогда максимальные $\varphi\psi$ – замыкания всех баз матроида $M(S)$ являются всеми коатомами свободного наращения матроида $M(S)$.*

Полученная теорема обобщает результат Г.Крапо, который нашел свободное наращение для случая, когда операторы φ и ψ коммутируют.

Говорят, что тождественное отображение множества S на себя индуцирует слабое отображение геометрии $G(S)$ в геометрию $H(S)$ (обозначение: $G \to H$), если каждое независимое подмножество из H независимо в G.

Теорема 2. *Пусть $H(S)$ и $K(S)$ – наращения геометрии $G(S)$ ранга r на конечном множестве S, а H и K – соответствующие им антицепи коатомов наращений. Тогда $H \leq K$ в решетке 2^S тогда и только тогда, когда тождественное отображение множества S на себя индуцирует слабое отображение $K(S) \to H(S)$.*

Другие свойства отображений и их связи с различными матроидными конструкциями можно найти, например в [1, 2].

Список литературы

[1] Oxley J. G. Matroid theory. — N.Y., Oxford University Press, 2006. — 532 с.

[2] Revyakin A. M. Matroids. // J. Math. Sci., 2002, V. 108, N 1. — С. 71–130.

[3] Crapo H. H., Rota G.-C. On the foundations of combinatorial theory. II. Combinatorial geometries // Stud. Appl. Math. — 1970. — 49, N 2. — С. 109–133.

[4] Скорняков Л. А. Элементы теории структур. М.: Наука, 1970. — 148 с.

[5] Crapo H. H. Erecting geometries. University of Waterloo, Canada, !973 (препринт).

[6] Ревякин А. М. О наращениях комбинаторных геометрий // Вестн. Моск. ун-та. Мат., мех., 4, 1976. — С. 59-62.

Квантовый поиск ближайшего соседа

Салихова Наиля Маратовна

Казанский (Приволжский) Федеральный университет, e-mail: nailyasalikhova66@gmail.com

Задача поиска вхождений заданной подстроки в тексте – одна из основных задач поиска информации.

В некоторых случаях нахождение точного совпадения не нужно или невозможно, а требуется найти наиболее похожие элементы. Поэтому среди задач машинного обучения очень распространена задача поиска ближайших соседей. В данной работе мы приводим характеристики известного классического алгоритма линейного поиска $\mathcal{A}1$ и квантового алгоритма $\mathcal{A}2$ [1]. Мы предлагаем наш квантовый алгоритм $\mathcal{A}$ и обсуждаем его характеристики.

Задача поиска. Дана обучающая выборка $D = \{b_1, \ldots b_N\}$, состоящая из N объектов. Объекты b_i представлены двоичными последовательностями длины n: $b_i \in \{0,1\}^n$. Дана двоичная строка $w = w_1 \ldots w_n$. Требуется найти индекс t объекта b_t такого, что Хэммингово расстояние между объектом b_t из обучающей выборки D и w – наименьшее.

Известные классический и квантовый алгоритмы

- Классический линейный алгоритм $\mathcal{A}1$ рассчитывает Хэммингово расстояние между каждым элементом D и w и находит номер объекта из D, соответствующего минимальному расстоянию. Временная сложность и сложность по памяти такого алгоритма $O(nN)$.

- Квантовый алгоритм $\mathcal{A}2$ [1] (2021 года) имеет следующие характеристики:
  - Временная сложность алгоритма $T^{\mathcal{A}2}(n, N) = O(\sqrt{N} \log N)$.
  - Авторы [1] не анализируют объем памяти алгоритма – число требуемых кубит. Наш анализ показывает, что сложность по памяти составляет $S^{\mathcal{A}2}(n, N) = 2n + 2 \log C + \log n$, где C – это количество классов.

Квантовый алгоритм $\mathcal{A}$

Мы применяем предлагаемый нами квантовый поиск на основе локально-чувствительного хеширования (LSH). Отметим, что LSH - это один из широко распространенных "классических" подходов к решению задачи поиска ближайшего соседа [2,3,4,5,6].

Классическое локально-чувствительное хеширование

Пусть M – метрическое пространство конечной размерности с метрикой ρ. S - множество корзин. Семейство функций $\mathcal{F} = \{f : M \to S\}$ называется d_1, d_2, p_1, p_2 - чувствительным, если для всех $x, y \in M$ имеет место следующее:

- Для множества $\mathcal{F}_{x,y} = \{f \in \mathcal{F} : f(x) = f(y)\}$ выполняется
  1. Если $\rho(x, y) \leq d_1$, то $|\mathcal{F}_{x,y}|/|\mathcal{F}| \geq p_1$
  2. Если $\rho(x, y) \geq d_2$, то $|\mathcal{F}_{x,y}|/|\mathcal{F}| \leq p_2$

Характеристики квантового алгоритма:

- *Сложность запросов.* Число $Q(\mathcal{A})$ запросов (число применений оракула) является "запросной" мерой сложности квантового алгоритма $\mathcal{A}$ [12].
- *Сложность по памяти.* Число $S(\mathcal{A})$ используемых кубит является мерой сложности памяти квантового алгоритма $\mathcal{A}$.
- *Временная сложность.* Число $T(\mathcal{A})$ применяемых элементарных квантовых операторов является мерой временной сложности квантового алгоритма $\mathcal{A}$.

Далее излагается квантовый алгоритм $\mathcal{A}$ поиска ближайшего соседа в постановке, когда размер объектов (количество бит) много больше количества объектов в обучающей выборке.

Алгоритм $\mathcal{A}$.

- *Первый этап алгоритма(классический):* Выбираем семейство

$$\mathcal{F} = \{f : f(x) \in \{0, 1\}\}$$

$(R, cR, 1-\frac{R}{n}, 1-\frac{cR}{n})$-локально-чувствительных функций для Хэммингова расстояния.

"Правильным" образом выбирается параметр k. По $\mathcal{F}$ определяется функция h

$$h(x) = (f_1(x), \ldots, f_k(x)),$$

где $f_1, \ldots, f_k$ выбираются случайным образом из $\mathcal{F}$.

- *Второй этап (подготовка квантового состояния):* По обучающей выборке D готовится начальное $(\log N + k + 1)$ кубитное состояние $|D\rangle$ на основе последовательности D

$$|D\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle \otimes |h(b_j)\rangle \otimes |1\rangle.$$

- *Третий этап (классический):* По w находится $h(w)$.
- *Четвертый этап (квантовый):* К состоянию $|D\rangle$ применяется алгоритм Гровера [7]. Квантовый поиск Гровера ведется с оракулом, определяемым $h(w)$.

  В результате измерения первых $\log N$ кубит будет получен номер t, для которого реализуется равенство $h(w) = h(b_t)$.

Теорема 1. Для алгоритма $\mathcal{A}$, для случая $N = o(n)$, выбранного R, для "правильно" подобранного параметра k алгоритма выполняется

- $Er^{\mathcal{A}}(string, w)) \leq 1/N + \sqrt[k]{1/N} + R/n$
- $T^{\mathcal{A}}(n, N) = O(\sqrt{N} \log N)$,
- $Q^{\mathcal{A}}(n, N) = O(\sqrt{N})$,
- $S^{\mathcal{A}}(n, N) = O(\log N)$.

Заключение

В заключение отметим, что алгоритм $\mathcal{A}2$ хорош в случае, когда количество N объектов в выборке много больше размера n объекта , а алгоритм $\mathcal{A}$ дает наилучший выигрыш в количестве используемых кубит для задачи поиска ближайшего соседа в случае, когда размер n объекта в обучающей выборке много больше количества N объектов в ней. При этом наш алгоритм $\mathcal{A}$ сразу обеспечивает хороший уровень достоверности результата.

Список литературы

[1] Li J. et al. Quantum K-nearest neighbor classification algorithm based on Hamming distance //arXiv preprint arXiv:2103.04253. – 2021.

[2] Manber U. et al. Finding Similar Files in a Large File System //Usenix Winter. – 1994. – Т. 94. – С. 1-10.

[3] Broder A. Z. On the resemblance and containment of documents //Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171). – IEEE, 1997. – С. 21-29.

[4] Gionis A. et al. Similarity search in high dimensions via hashing //Vldb. – 1999. – Т. 99. – №. 6. – С. 518-529.

[5] Andoni A., Indyk P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions //2006 47th annual IEEE symposium on foundations of computer science (FOCS'06). – IEEE, 2006. – С. 459-468.

[6] Lee K. M. Locality-sensitive hashing techniques for nearest neighbor search //International Journal of Fuzzy Logic and Intelligent Systems. – 2012. – Т. 12. – №. 4. – С. 300-307.

[7] Grover L. K. A fast quantum mechanical algorithm for database search //Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. – 1996. – С. 212-219.

[8] Boyer M. et al. Tight bounds on quantum searching //Fortschritte der Physik: Progress of Physics. – 1998. – Т. 46. – №. 4-5. – С. 493-505.

О сложности проверки периодичности функций алгебры логики, заданных многочленами Жегалкина

Селезнева Светлана Николаевна

Московский государственный университет имени М. В. Ломоносова, e-mail: selezn@cs.msu.ru

В настоящей работе рассматривается задача проверки периодичности и поиска периодов функции алгебры логики по ее многочлену Жегалкина. Периодические функции обладают линейной структурой, что означает определенную слабость этих функций с криптографической точки зрения (см. [1], с. 107). В [2] доказано, что проверить периодичность функции алгебры логики относительно периода из всех единиц по ее многочлену Жегалкина можно с полиномиальной сложностью относительно произведения числа переменных функции и числа слагаемых в ее многочлене Жегалкина. Из этого сразу следует, что проверить периодичность функции алгебры логики относительно любого заданного периода по ее многочлену Жегалкина можно также с полиномиальной сложностью. В [3] описаны возможные периоды симметрических функций. Изучались также свойства периодических функций (см. [3–5]). В [6] показано, что поиск периодов функции алгебры логики по ее многочлену Жегалкина можно свести к решению некоторой системы алгебраических уравнений (в общем случае,

нелинейной) над полем из двух элементов. В [7] описан квантовый алгоритм решения задачи о проверке периодичности функции алгебры логики. В настоящей работе доказано, что поиск периодов функции алгебры логики по ее многочлену Жегалкина можно свести к решению некоторой однородной системы линейных алгебраических уравнений над полем из двух элементов. При этом такое сведение можно выполнить со сложностью $n^{O(d)}$, где n — число переменных, а d — степень этой функции. Как следствие, получено, что по многочлену Жегалкина функции алгебры логики ограниченной степени базис пространства всех ее периодов можно найти с полиномиальной сложностью относительно числа переменных функции.

Пусть $E_2 = \{0, 1\}$, $P_2^{(n)} = \{f \mid f : E_2^n \to E_2\}$ — множество функций алгебры логики n переменных, $n \geqslant 0$, и $P_2 = \bigcup\limits_{n \geqslant 0} P_2^{(n)}$ — множество всех функций алгебры логики. Пусть $F_2 = (E_2; +, \cdot)$ — поле и $PF_2[x_1, \ldots, x_n]$ обозначает множество всех многочленов из $F_2[x_1, \ldots, x_n]$, в которых степень любой переменной не выше единицы. Известно, что для любой функции $f(x_1, \ldots, x_n) \in P_2^{(n)}$ найдется единственный задающий ее многочлен $p_f \in PF_2[x_1, \ldots, x_n]$, который называется ее многочленом Жегалкина. В дальшейшем будем отождествлять понятие функции алгебры логики и ее представления многочленом Жегалкина, если это не приводит к путанице. Если $f \in P_2$, то пусть $d(f)$ обозначает степень функции f, т. е. степень многочлена Жегалкина функции f.

Набор $a \in E_2^n$ называется *периодом* функции $f \in P_2^{(n)}$, если верно равенство

$$f(x_1 + a_1, \ldots, x_n + a_n) = f(x_1, \ldots, x_n).$$

Понятно, что нулевой набор $0 = (0, \ldots, 0) \in E_2^n$ является периодом любой функции из $P_2^{(n)}$. Функция $f \in P_2^{(n)}$ называется *периодической*, если найдется ненулевой набор $a \in E_2^n$, являющийся ее периодом. Множество всех периодов функции $f \in P_2$ обозначим $T(f)$. Множество $T(f)$ является линейным пространством над полем F_2. Следовательно, для любой функции $f \in P_2$ множество $T(f)$ можно определить как множество решений некоторой однородной системы линейных уравнений над полем F_2.

Основной результат работы состоит в доказательстве следующей теоремы.

Теорема 1. *Задача поиска базиса пространства всех периодов функции $f \in P_2^{(n)}$, заданной многочленом Жегалкина, может быть решена со сложностью $n^{O(d(f))}$.*

Из теоремы 1 получаем следствие.

Следствие 1. *Пусть $d \geqslant 1$ — заданное число. Задача поиска базиса пространства всех периодов функции алгебры логики степени не выше d по ее многочлену Жегалкина может быть решена полиномиальным алгоритмом относительно числа переменных этой функции.*

Работа поддержана РФФИ в рамках научного проекта № 19-01-00200-а и Минобрнауки РФ в рамках выполнения программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2019-1621.

Список литературы

[1] Логачев О. А., Сальников А. А., Смышляев С. В., Ященко В. В. Булевы функции в теории кодирования и криптологии. — М.: МЦНМО, 2012. — 584 с.

[2] Селезнева С. Н. О сложности распознавания полноты множеств булевых функций, реализованных полиномами Жегалкина // Дискретная математика. — 1997. — Т. 9, вып. 4. — С. 24–31.

[3] Dawson E., Wu C.-K. On the linear structure of symmetric Boolean functions // Australasian Journal of Combinatorics. — 1997. — V. 16. — P. 239–243.

[4] Леонтьев В. К. О некоторых задачах, связанных с булевыми полиномами // Журнал вычислительной математики и математической физики. — 1999. — Т. 39, вып. 6. — С. 1045–1054.

[5] Charpin P., Kyureghyan G. M. On a class of permutation polynomials over F_{2^n} // Lecture Notes of Computer Science. — 2008. — V. 5203. — P. 368–376.

[6] Бухман А. В. О свойствах полиномов периодических функций и сложности распознавания периодичности по полиному булевой функции // Дискретная математика. — 2014. — Т. 26, вып. 1. — С. 21–31.

[7] Yang L., Li H.-W. Investigating the linear structure of Boolean functions based on Simon's period-finding quantum algorithm // https://arxiv.org/pdf/1306.2008.pdf

Программы с запаздыванием

Сергеев Игорь Сергеевич

ФГУП «НИИ «Квант», e-mail: isserg@gmail.com

В современной электронике многие преобразования выполняются конвейерными схемами. Конвейерная схема считывает значения входов и обновляет значения выходов на каждом такте рабочей частоты, но соответствующий входному набору аргументов результат вычисляется с задерж-

кой (латентностью) в несколько тактов. Это побуждает рассмотреть следующую модель вычислений, которую мы назовем *программами с запаздыванием*. Программа с запаздыванием t над базисом B и множеством входных переменных X определяется как разновидность неветвящейся программы:

$$g_1 = f_1(Y_1), \quad g_2 = f_2(Y_2), \quad \ldots, \quad g_k = f_k(Y_k),$$

где $f_i \in B$ и $Y_i \subset X \cup \bigcup_{j \le i-t}\{g_j\}$. При $t = 1$ имеем обычную неветвящуюся программу. При $t > 1$ выбор аргументов для выполнения очередной операции ограничен результатами, полученными не менее t шагов назад.

Как обычно, k (длина последовательности) называется сложностью программы. Программа реализует оператор F, если каждая компонента оператора функционально эквивалентна некоторой функции g_i. Через $C_B^{(t)}(F)$ обозначим *сложность* оператора F — минимальную длину реализующей его программы с запаздыванием t.

Основной вопрос: насколько вычисление конкретного оператора сложнее в модели программ с запаздыванием относительно модели обычных неветвящихся программ. Заметим, что в модели с запаздыванием любые t последовательных функций $g_{i+1}, g_{i+2}, \ldots, g_{i+t}$ вычисляются независимо. Таким образом, модели присуща локальная параллельность.

Очевидно,

$$C_B^{(1)}(F) \le C_B^{(t)}(F) \le t \cdot C_B^{(1)}(F). \tag{1}$$

Рассмотрим несколько простых примеров вычислений над базисом из одной операции умножения $B = \{*\}$. Первые два примера демонстрируют достижимость как оптимистической, так и пессимистической границ (1). Параметр t далее считаем постоянным.

Пример 1. Возведение в N-ю степень.

Утверждение 1. $C_B^{(t)}\left(x^N\right) \sim t \log_2 N$.

Доказательство. Верхнюю оценку дает метод Брауэра [2] и (1). Для доказательства нижней достаточно заметить, что за серию из t шагов (напомним, что они выполняются независимо) максимум вычисленных показателей степени x^M может быть не более чем удвоен. □

Пример 2. Вычисление последовательных степеней.

Утверждение 2. $C_B^{(t)}\left(x, x^2, \ldots, x^N\right) \sim N$.

Доказательство. За $t-1$ серий по t шагов легко получить степени $x^2, \ldots, x^t$ (на самом деле, достаточно $\log_2 t$ серий). Далее, в каждой очередной серии из t шагов можно последовательно вычислять степени $x^{(k+1)t}$, $x^{(k+1)t-1}, \ldots,$ x^{kt+1}. Длина описанной программы не превосходит $N + t^2$. □

Из общих соображений ясно, что запаздывание не замедляет существенно вычисление операторов, допускающих эффективные параллельные реализации. Напомним, что *глубина* программы определяется как глубина изображающего ее графа (схемы), т. е. как длина максимального ориентированного пути от входа к выходу.

Лемма 1. *Если оператор F вычисляется неветвящейся программой над базисом B со сложностью C и глубиной D, то $C_B^{(t)}(F) \le C + tD$.*

Доказательство. Программу можно разбить на D подпрограмм (слоев), состоящих из независимых (параллельно выполняемых) шагов. Подпрограмма из k независимых шагов тривиально реализуется $\lceil k/t \rceil$ сериями из t шагов в модели с запаздыванием t, откуда следует требуемая оценка. □

Утверждение 2 выводится из леммы 1, поскольку последовательные N степеней переменной x легко вычислить программой сложности $N-1$ и глубины $\lceil \log_2 N \rceil$.

В свете сказанного вопрос о замедлении вычислений в модели программ с запаздыванием содержателен для операторов, программы оптимальной сложности для которых имеют глубину, по порядку совпадающую со сложностью. Следующие два примера иллюстрируют эту ситуацию.

Пример 3. Вычисление префиксов $\pi_i = x_1 \cdot \ldots \cdot x_i$, $1 \le i \le N$.

Утверждение 3. $C_B^{(t)}(\pi_1, \ldots, \pi_N) \sim 2tN/(t+1)$.

Доказательство. Докажем нижнюю оценку. Разобьем программу на серии из t операций. Предположим, что вычисление укладывается в k серий. Пусть программа вычисляет максимальное произведение $x_1 \cdot \ldots \cdot x_N$ как $x_1 \cdot p_1 \cdot \ldots \cdot p_k$, где p_i — множитель, добавляемый в i-й серии. Допускается, что некоторые p_i могут быть равны 1.

Вычисление всех p_i требует не менее $N-1-k$ шагов программы. Учитывая, что еще минимум $N-1$ шагов требуется для вычисления собственно префиксов, длина программы оценивается как $2(N-1)-k$. Получаем оценку $kt \ge 2(N-1)-k$, откуда следует $k \ge 2(N-1)/(t+1)$.

Иначе, нижнюю оценку можно получить, отталкиваясь от известного соотношения $C + D \ge 2N - 2$ для сложности C и глубины D программ, реализующих префиксы N переменных, см. [3]. Поскольку для программы с запаздыванием t выполнено $D \le C/t + 1$, то $(1 + 1/t)C \gtrsim 2N$.

Покажем, что оценка достижима. Пусть для простоты $N = k(t+1)$. Максимальный префикс вычисляется по формуле

$$\pi_N = x_1 \cdot p_1 \cdot x_{t+2} \cdot p_2 \cdot x_{2(t+1)+1} \cdot \ldots \cdot p_k, \qquad p_i = x_{(i-1)(t+1)+2} \cdot \ldots \cdot x_{i(t+1)}.$$

Обозначим $p_{i,\,j} = x_{(i-1)(t+1)+2} \cdot \ldots \cdot x_{(i-1)(t+1)+j+1}$ — промежуточные стадии вычисления p_i (при этом $p_i = p_{i,\,t}$). Остальные префиксы получаются как

$$\pi_{i(t+1)+1} = \pi_{i(t+1)} \cdot x_{i(t+1)+1}, \quad \pi_{i(t+1)+j} = \pi_{i(t+1)+1} \cdot p_{i+1,\,j-1}, \quad j = 2, \ldots, t+1.$$

Программу можно составить, последовательно объединяя серии следующего вида при i от $2-t$ до $k-1$:

$$p_{i+t-1,\,2},\ p_{i+t-2,\,3},\ \ldots,\ p_{i+1,\,t},\ \pi_{i(t+1)+1},$$
$$\pi_{i(t+1)-1},\ \pi_{i(t+1)-2},\ \ldots,\ \pi_{i(t+1)-t+1},\ \pi_{(i+1)(t+1)}.$$

Неопределенные величины из программы исключаются (заменяются чем угодно). Длина программы равна $2t(k+t-2) = 2N/(t+1) + O(t^2)$. $\square$

Сделаем еще одно элементарное наблюдение. Пусть $X_1, \ldots, X_t$ — равномощные группы переменных. Очевидно,

$$C_B^{(t)}(F(X_1), \ldots, F(X_t)) \le t \cdot C_B^{(1)}(F). \tag{2}$$

В следующем примере (несмотря на схожесть с предыдущим) свойство (2) позволяет избежать существенных потерь во времени при переходе к модели программ с запаздыванием.

Пример 4. Вычисление дополняющих произведений $c_i = \prod_{j \ne i} x_j$, $1 \le i \le N$.

Утверждение 4. $C_B^{(t)}(c_1, \ldots, c_N) \sim 3N$.

Доказательство. Сложность системы в модели без запаздывания равна $3N-6$, см. [1]. Схема, доставляющая оптимальную оценку в этой модели, состоит из последовательной части (вычисление префиксов $\pi_i = x_1 \cdot \ldots \cdot x_i$ и суффиксов $\sigma_i = x_i \cdot \ldots \cdot x_N$) и параллельной части (произведения суффиксов и префиксов). Но эту схему можно перестроить в более параллельную.

Пусть $N = tk$. Разобьем множество переменных на t групп $X_j = (x_{(j-1)k+1}, \ldots, x_{jk})$, $j = 1, \ldots, t$, мощности k. Обозначим префиксные и суффиксные произведения в каждой из групп через $\pi_{i,\,j} = \pi_i(X_j)$ и $\sigma_{i,\,j} = \sigma_i(X_j)$. Положим формально $\pi_{0,\,j} = \sigma_{k+1,\,j} = 1$. Через $p_j = \pi_{k,\,j} = \sigma_{1,\,j}$ обозначим произведение всех переменных группы X_j.

Если $i = (j-1)k + l$, где $1 \le l \le k$, то c_i можно вычислить по формуле

$$c_i(x_1, \ldots, x_N) = c_j(p_1, \ldots, p_t) \cdot \pi_{l-1,\,j} \cdot \sigma_{l+1,\,j}. \tag{3}$$

Множество всех суффиксов k переменных вычисляется тривиально неветвящейся программой длины $k-1$. Поэтому согласно (2) все $\sigma_{i,\,j}$, и среди них p_j, могут быть вычислены программой с запаздыванием t длины $t(k-1)$.

Далее, за $t-1$ серий по t шагов можно вычислить все дополняющие группы X_j произведения $u_j = c_j(p_1, \ldots, p_t)$.

При каждом j произведения $u_j \cdot \pi_{l-1,\,j}$, $1 \le l \le k$, образуют систему префиксов $\{\pi_i(u_j, X_j) \mid 1 \le i \le k\}$. Согласно (2), они вычисляются программой длины $t(k-1)$.

Теперь любое произведение (3) может быть получено одним умножением выражения $u_j \cdot \pi_{l-1,\,j}$, найденного на предыдущем шаге, и суффикса $\sigma_{l+1,\,j}$. Эти умножения независимы, поэтому могут быть выполнены за N шагов. Общая длина программы не превосходит $2t(k-1) + N + t^2 = 3N + O(t^2)$. □

Иначе, этот результат можно получить как следствие из леммы 1: система дополняющих произведений N переменных вычисляется неветвящейся программой сложности $3N + o(N)$ и глубины $o(N)$, что по сути и доказано в утверждении 4.

Список литературы

[1] Чашкин А. В. О сложности булевых матриц, графов и соответствующих им булевых функций // Дискретная математика. — 1994. — Т. 6(2). — С. 43–73.

[2] Brauer A. On addition chains // Bull. AMS. — 1939. — V. 45. — P. 736–739.

[3] Snir M. Depth-size trade-offs for parallel prefix computation // J. Algorithms. — 1986. — V. 4. — P. 185–201.

О длине минимальных единичных проверяющих тестов относительно замен функциональных элементов на инверторы в произвольном полном базисе

Темербекова Гульгайша Габдуловна, Романов Дмитрий Сергеевич

Московский государственный университет имени М. В. Ломоносова, e-mail: gulgaisha93@mail.ru, romanov@cs.msu.ru

Рассмотрим тестирование схем из функциональных элементов (СФЭ), реализующих произвольные булевы функции. Пусть имеется СФЭ S, реализующая булеву функцию $f(\tilde{x}^n)$, где $x^n = (x_1, x_2, \ldots, x_n)$. Пусть на S воздействует источник неисправностей U так, что один или несколько элементов схемы S переходят в неисправное состояние. Тогда схема S вместо исходной функции $f(\tilde{x}^n)$ будет реализовывать некоторую, возможно, от-

личную от f, булеву функцию $g(\tilde{x}^n)$. Полученная функция $g(\tilde{x}^n)$ называется функцией неисправности схемы S.

Введём следующие определения. Проверяющим тестом для схемы S называется такое множество T наборов значений переменных $x_1, x_2, \ldots, x_n$, что для любой отличной от $f(\tilde{x}^n)$ функции неисправности схемы S в T найдётся набор α, на котором $f(\alpha) \neq g(\alpha)$. Число наборов в T называется длиной теста. Тест называется единичным, если может быть неисправен только один элемент схемы. Нетривиальной будем называть неисправность СФЭ, при которой хотя бы на одном входном наборе меняется (по сравнению со случаем отсутствия неисправностей) значение на выходе хотя бы одного элемента в схеме. Схема называется неизбыточной, если всякая нетривиальная неисправность любого одного элемента приводит к отличной от исходной функции неисправности.

Пусть зафиксирован вид неисправности «закорачивание входа элемента на его выход с инвертированием» (собственно, замена функционального элемента на инвертор одного из входов этого элемента) для СФЭ. Пусть B — произвольный конечный схемный базис, обладающий функциональной полнотой в P_2, а T — единичный проверяющий тест для некоторой схемы S в базисе B. Введём следующие обозначения: $L(T)$ — длина теста T; $L(S) = \min L(T)$, где минимум берётся по всем единичным проверяющим тестам T для схемы S; $L(f) = \min L(S)$, где минимум берётся по всем неизбыточным схемам S в базисе B, реализующим функцию $f(\tilde{x}^n)$; $L(n) = \max L(f)$, где максимум берётся по всем булевым функциям f от n переменных, для которых определено значение $L(f)$. Функция $L(n)$ называется функцией Шеннона длины единичного проверяющего теста.

В данной работе найдены оценки функции Шеннона длины единичного проверяющего теста при неисправностях вида «закорачивание входа элемента на его выход с инвертированием» для СФЭ в произвольном конечном схемном базисе, обладающем функциональной полнотой в P_2.

Справедливо следующее утверждение.

Теорема. *Любую булеву функцию $f(\tilde{x}^n)$, где $x^n = (x_1, x_2, \ldots, x_n)$, можно реализовать неизбыточной схемой из функциональных элементов в произвольном конечном схемном базисе, обладающем функциональной полнотой в P_2, допускающей в случае неисправностей вида «закорачивание входа элемента на его выход с инвертированием» единичный проверяющий тест длины не более* 4.

Работа выполнена при поддержке Московского Центра фундаментальной и прикладной математики (проект «Оценки сложностных характеристик булевых функций и графов») и госбюджетной темы НИР №5.4.19 факультета ВМК МГУ имени М. В. Ломоносова.

Radius of almost all n-vertex graphs of given diameter

Fedoryaeva Tatiana Ivanovna

Sobolev Institute of Mathematics, e-mail: fti@math.nsc.ru

We study finite labeled ordinary n-vertex graphs. For a connected graph $G = (V, E)$, the *distance* $\rho_G(u, v)$ between its vertices $u, v \in V$ is defined as the length of the shortest path connecting these vertices. In this case, $e_G(v) = \max_{u \in V} \rho_G(v, u)$ is the *eccentricity* of the vertex v of the graph G, $d(G) = \max_{v \in V} e_G(v)$ is the *diameter* of the graph G, and $r(G) = \min_{v \in V} e_G(v)$ is the *radius* of the graph G. A vertex is called *central* if its eccentricity is equal to the radius. A graph is *self-centered* if all its vertices are central.

The relation $r(G) \leq d(G) \leq 2r(G)$ between the radius and the diameter of an arbitrary connected graph G is well known. Moreover, for every r and d satisfying the relation $r \leq d \leq 2r$ and $n \geq d + r$, F. Ostrand's theorem implies the existence of an n-vertex graph G with $r(G) = r$ and $d(G) = d$ [1]. From the result of J.W. Moon and L. Moser [2], it is easy to obtain that almost all n-vertex graphs have diameter and radius equal to 2 (see, for example, [3]). Yu.D. Burtin, considering a random n-vertex graph $G_p(n)$ with the probability $p = p(n)$ of the presence of an edge, showed that $L_p(n) + 1 \leq r(G_p(n)) \leq d(G_p(n)) \leq L_p(n) + 2$ under natural restrictions on the growth of $p(n)$ if $n \to \infty$, i.e. radius and diameter of the random graph $G_p(n)$ can take only two values $L_p(n) + 1$ and $L_p(n) + 2$, which are calculated in [4], with probability tending to 1 as $n \to \infty$. The question naturally arises about a possible radius of almost all n-vertex graphs of fixed diameter k. In the present paper, based on the found typical properties of metric balls contained in a graph and constructed typical n-vertex graphs of given diameter (Theorem 1), we establish the radius of almost all graphs of a given fixed diameter (Theorems 2 and 3).

Let us give the necessary definitions. To estimate the measure of the number of graphs with a certain property, the concept of *almost all* is often used; in this approach, the studied property is considered for graphs with a large number of vertices. Let $\mathcal{J}_n$ be the class of labeled n-vertex graphs with the fixed set of vertices $V = \{1, 2, \ldots, n\}$, $n \in \mathbb{N}$. Consider some property $\mathcal{P}$, by which each graph may or may not possess. Through $\mathcal{J}_n^{\mathcal{P}}$ denote the set of all graphs from $\mathcal{J}_n$ that possess the property $\mathcal{P}$. *Almost all graphs possess the property* $\mathcal{P}$ if $\lim_{n\to\infty} \frac{|\mathcal{J}_n^{\mathcal{P}}|}{|\mathcal{J}_n|} = 1$, i.e. $|\mathcal{J}_n^{\mathcal{P}}| \sim |\mathcal{J}_n|$ (here $\sim$ denotes *the asymptotic equality* as $n \to \infty$), and *there are almost no graphs with the property* $\mathcal{P}$ if $\lim_{n\to\infty} \frac{|\mathcal{J}_n^{\mathcal{P}}|}{|\mathcal{J}_n|} = 0$. In the study and selection of almost all graphs in a class Ω of graphs under consideration it is often useful to define not characteristic

properties themselves for the notion of almost all, but directly select a subclass of typical graphs itself. A subclass $\Omega^* \subseteq \Omega$ is the *class of typical graphs of the class* Ω if $|\Omega_n^*| \sim |\Omega_n|$, where $\Omega_n = \Omega \cap \mathcal{J}_n$ and $\Omega_n^* = \Omega^* \cap \mathcal{J}_n$. A property of graphs from a class under consideration is *typical* if almost all graphs from this class have the given property.

Let $\mathcal{J}_{n,\,d=k}$, $\mathcal{J}_{n,\,d\geq k}$, $\mathcal{J}^*_{n,\,d\geq k}$ be the following classes of labeled n-vertex graphs: graphs of diameter k; connected graphs of diameter at least k; graphs (not necessarily connected) with a shortest path of length at least k, respectively. When studying the variety of metric balls in graphs, the author proved that all these three classes of graphs have the same asymptotic cardinality for every fixed $k \geq 2$ [5,6]. Moreover, a class $\mathcal{F}_{n,k}$ of typical graphs for each of the classes $\mathcal{J}_{n,\,d=k}$, $\mathcal{J}_{n,\,d\geq k}$, $\mathcal{J}^*_{n,\,d\geq k}$, $k \geq 3$, with a number of nontrivial metric properties is constructed in [7]. It turned out that almost all n-vertex graphs of given diameter $k \geq 3$ have a unique pair of diametral vertices (this is not the case for $k = 1, 2$) and a number of typical properties related to the variety of metric balls contained in the graph is fulfilled.

In this paper, in the class $\mathcal{F}_{n,k}$, $k \geq 3$, we distinguish a family $\mathcal{F}_{n,k,p}$, $p \geq 1$ of nested subclasses of n-vertex graphs. Each of these classes preserves the already established properties of graphs from $\mathcal{F}_{n,k}$; in addition, the graphs of the introduced classes have a property of metric spheres, which ensures the presence of a predetermined number of vertices in the intersection of spheres of radius 1. This condition also turns out to be useful in studying typical properties of n-vertex graphs associated with various metric characteristics.

Theorem 1. *Let $k \geq 3$, $0 < \varepsilon < 1$ and $p \geq 1$ do not depend on n. Then there is a constant $c > 0$ independent of n and such that for every $n \in \mathbb{N}$ the following inequalities hold*

$$2^{\binom{n}{2}}\xi_{n,k}\Big(1 - c\Big(\frac{5+\varepsilon}{6}\Big)^{n-k+1}\Big) \leq |\mathcal{F}_{n,k,p}| \leq |\mathcal{F}_{n,k}| \leq |\mathcal{J}_{n,\,d=k}| \leq |\mathcal{J}_{n,\,d\geq k}|$$

$$\leq |\mathcal{J}^*_{n,\,d\geq k}| \leq 2^{\binom{n}{2}}\xi_{n,k}\Big(1 + c\Big(\frac{5+\varepsilon}{6}\Big)^{n-k+1}\Big), \ \textit{where}$$

$\xi_{n,k} = q_k(n)_{k-1}\Big(\frac{3}{2^{k-1}}\Big)^{n-k+1}$, $q_k = \frac{1}{2}(k-2)2^{-\binom{k-1}{2}}$, $(n)_k = n(n-1)\cdots(n-k+1)$.

Corollary 1. *Let $k \geq 3$ and $p \geq 1$ be independent of n. Then $\mathcal{F}_{n,k,p}$ is the class of typical graphs of each of the following classes of n-vertex graphs $\mathcal{J}_{n,\,d=k}$, $\mathcal{J}_{n,\,d\geq k}$ and $\mathcal{J}^*_{n,\,d\geq k}$. Moreover, $|\mathcal{F}_{n,k,p}| \sim |\mathcal{F}_{n,k}| \sim |\mathcal{J}_{n,\,d=k}| \sim |\mathcal{J}_{n,\,d\geq k}| \sim |\mathcal{J}^*_{n,\,d\geq k}| \sim 2^{\binom{n}{2}}\xi_{n,k}$ for $n \to \infty$.*

Note that complete graph K_n is the unique n-vertex graph of diameter 1 and $r(K_n) = 1$. Therefore, almost all graphs of diameter $k = 1$ have radius

equal to the diameter. A similar fact for graphs of diameter 2 also trivially follows from well-known theorems.

Theorem 2. *Almost all graphs of diameter* 2 *have radius* 2.

The main case $k \geq 3$ is considered in the following theorem.

Theorem 3. *For every fixed integer* $k \geq 3$*, almost all* n*-vertex graphs of diameter* k *have radius* $\lceil \frac{k}{2} \rceil$.

Corollary 2. *There are almost no self-centered* n*-vertex graphs of fixed diameter* $k \geq 3$*, while almost all graphs of diameter* $k = 1, 2$ *are self-centered.*

Corollary 3. *Almost all* n*-vertex graphs of odd fixed diameter have at least two central vertices.*

All obtained typical properties for n-vertex graphs of fixed diameter $k \geq 2$ remain typical for connected graphs of diameter at least k, as well as for graphs in the class $\mathcal{J}^*_{n,\,d\geq k}$. In particular, the following corollary hold.

Corollary 4. *For every fixed* $k \geq 3$*, almost all* n*-vertex graphs of each of the classes* $\mathcal{J}_{n,\,d\geq k}$, $\mathcal{J}^*_{n,\,d\geq k}$ *are connected and have radius* $\lceil \frac{k}{2} \rceil$*, diameter* k.

The work was carried out within the framework of the state contract of the Sobolev Institute of Mathematics (project no. FWNF-2022-0018).

References

[1] Ostrand P. A. Graphs with specified radius and diameter // Discrete Math. — 1973. — Vol. 4, Iss. 1. — P. 71–75.

[2] Moon J. W., Moser L. Almost all (0,1) matrices are primitive // Stud. Sci. Math. Hung. — 1966. — P. 153–156.

[3] Emelichev V. A., Melnikov O. I., Sarvanov V. I., Tyshkevich R. I. Lectures on Graph Theory // B.I.Wissenschaftsverlag, Mannheim. — 1994.

[4] Burtin Yu. D. On Extreme Metric Parameters of a Random Graph. I. Asymptotic Estimates // Theory Probab. Appl. — 1975. — Vol. 19, Iss. 4. — P. 710–725.

[5] Fedoryaeva T. I. The diversity vector of balls of a typical graph of small diameter // Diskretn. Anal. Issled. Oper. — 2015. — Vol. 22, Iss. 6. — P. 43–54.

[6] Fedoryaeva T. I. Asymptotic approximation for the number of n-vertex graphs of given diameter // J.Appl.Ind.Math. — 2017. — Vol. 11, Iss. 2. — P. 68–86.

[7] Fedoryaeva T. I. Structure of the diversity vector of balls of a typical graph with given diameter // Siber. Electr. Math. Reports — 2016. — Vol. 13. — P. 375–387.

Квантовый алгоритм для распознавания языка Дика для нескольких типов скобок

Хадиев Камиль Равилевич[1], **Кравченко Дмитрий Александрович**[2]

[1] Казанский федеральный университет, Казанский физико-технический институт им. Е.К. Завойского ФИЦ Казанский научный центр РАН, e-mail: kamilhadi@gmail.com

[2] Латвийский университет, Центр квантовой информатики, e-mail: kravchenko@gmail.com

Квантовые вычисления [1, 2] являются активно развивающейся областью теоретической кибернетики. Известно множество квантовых алгоритмов, демонстрирующих преимущество перед классическими [3]. В рамках данной работы мы рассмотрели язык Дика для нескольких типов скобок. Говоря формально, рассматривается функция $\textsc{Dyck}_{n,k,t}$ для некоторых положительных целых n, k и t, на вход которой поступает строка длины n, состоящая из символов множества $\{1, \ldots, 2t\}$, где символы со значениями $2i$ и $2i-1$ соответствуют открывающейся и закрывающейся скобкам i-го типа. Значение функции равно 1, если и только если входная строка образует правильную скобочную последовательность глубины не более k.

Мы рассматриваем задачу в терминах запросной сложности, в частности квантовой запросной сложности [2], при этом под термином «сложность» мы будем понимать именно запросную сложность.

Для данной задачи известно, что классическая сложность равна $\Theta(n)$. При этом решения для одного типа скобок и для нескольких типов скобок различаются. Для одного типа скобок используется счётчик и $O(\log k)$ памяти, а для нескольких типов скобок используется стек и $O(k)$ памяти.

В квантовом случае известен алгоритм для случая одного типа скобок [5,6]. Его сложность составляет $O\left(\sqrt{n}(\sqrt{\log n})^k\right)$, нижняя же оценка [5,7] составляет $O(\sqrt{n}c^k)$ для некоторой константы c.

В данной работе мы строим квантовый алгоритм для случая нескольких типов скобок. Его сложность совпадает со сложностью квантового алгоритма для одного типа скобок – в отличие от ситуации с классическими алгоритмами, где разные типы скобок требуют экспоненциально большего размера памяти. Мы также отмечаем, что оценки снизу, справедливые для случая одного типа скобок $O(\sqrt{n}c^k)$, справедливы и для нескольких типов скобок.

Алгоритм

Решение задачи состоит из трёх этапов.

- **Этап 1** Проверка того, что нет скобок с номером типа более чем t. Для этого мы можем нейти максимальный номер типа скобок с помощью квантового алгоритма поиска максимума [8] – разновидности алгоритма

Гровера [4]. Сложность данного этапа $O(\sqrt{n})$, а вероятность ошибки – некоторая константа.

- **Этап 2** Проверка того, что строка является правильной скобочной последовательностью (без учёта типов скобок) и что при этом глубина вложенности скобок не превышает k. Мы используем алгоритм из работ [5,6]. Сложность данного этапа $O\left(\sqrt{n}(\log n)^{0.5k}\right)$, а вероятность ошибки – некоторая константа.
- **Этап 3** Проверка специального условия для любого целого v. Пусть $s' = (s_l, \ldots, s_r)$ – подстрока входной строки $s = (s_1, \ldots, s_n)$, а $(s_{l+1}, \ldots, s_{r-1})$ – пустая подстрока либо правильная скобочная последовательность глубины $v-1$. Тогда s_l и s_r должны быть, соответственно, открывающейся и закрывающейся скобками одного типа. Для этого этапа был разработан алгоритм, имеющий сложность $O\left(\sqrt{n}(\log n)^{0.5(k-1)}\right)$ и константную вероятность ошибки.

В случае, если все три этапа проверки пройдены, то строка принадлежит языку и значение Dyck равно 1. Если же хотя бы один этап не пройден, то результат равен 0.

Рассмотрим немного подробнее Этап 3. Мы последовательно проверим выполнение условия для всех v от 1 до k. Для проверки условия в случае $v = 1$, достаточно запустить алгоритм Гровера для поиска пары двух символов s_i, s_{i+1} таких, что они являются открывающейся и закрывающейся скобками разного типа. Если алгоритм Гровера найдёт такую пару, то условие не пройдено, если же не найдёт, то пройдено. Для проверки $v > 1$ достаточно найти минимальные $(v-1)$-подстроку и следующую за ней $-(v-1)$-подстроку – так, чтобы между ними не было никакой $\pm(v-1)$-подстроки, – а затем проверить, что их окаймляют открывающая и закрывающая скобки разных типов. Минимальной t-подстрокой ($\pm t$-подстрокой) называется подстрока, у которой абсолютная величина разности между числом открывающихся и закрывающихся скобок равна t и которая не содержит подстрок с аналогичным свойством. Если алгоритм найдёт такую пару, то условие не пройдено, если же не найдёт, то пройдено. Для реализации алгоритма достаточно модифицировать алгоритм из [5].

Теорема 1. *Приведённый квантовый алгоритм решает задачу* $\text{Dyck}_{n,k,t}$. *Его сложность* $O\left(\sqrt{n}(\log n)^{0.5k}\right)$ *и вероятность ошибки константна.*

Доказательство. Этапы выполняются последовательно, поэтому суммарная сложность алгоритма составляет

$$O\left(\sqrt{n}\right) + O\left(\sqrt{n}(\log n)^{0.5k}\right) + O\left(\sqrt{n}(\log n)^{0.5(k-1)}\right) = O\left(\sqrt{n}(\log n)^{0.5k}\right).$$

Вероятность ошибки каждого этапа константна, поэтому итоговая вероятность ошибки – тоже константа. Заметим, что даже если эта константа

велика (близка к 1), мы можем повторить алгоритм несколько раз. Тогда вероятность возникновения ошибок сразу во всех повторениях будет экспоненциально убывать. **Теорема 1 доказана.**

Работа выполнена за счёт средств субсидии, выделенной Казанскому федеральному университету для выполнения государственного задания в сфере научной деятельности, проект 0671-2020-0065; а также при поддержке Европейского фонда регионального развития в рамках проекта 1.1.1.5/18/A/020 «Квантовые алгоритмы: от теории сложности к экспериментам».

Список литературы

[1] Ambainis A. Understanding quantum algorithms via query complexity // Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018. — 2018. — С. 3265–3285.

[2] Ablayev F., Ablayev M., Huang J.Z., Khadiev K., Salikhova N., Wu D. On quantum methods for machine learning problems part I: Quantum tools //Big Data Mining and Analytics. — 2019. — Т. 3. — №. 1. — С. 41–55.

[3] Stephen Jordan. Quantum algorithms zoo/http://quantumalgorithmzoo.org/ — 2021.

[4] Grover L. K. A fast quantum mechanical algorithm for database search //Proceedings of STOC'96. — 1996. — С. 212–219.

[5] Ambainis A., Balodis K., Iraids J., Khadiev K., Klevickis V., Prusis K., Shen Y., Smotrovs J., Vihrovs J. Quantum Lower and Upper Bounds for 2D-Grid and Dyck Language//MFCS 2020. — LIPIcs. — 2020.— Т. 170. — С. 8:1–8:14.

[6] Хадиев К., Shen Y., Квантовый алгоритм для распознавания языка Дика константной глубины //Проблемы теоретической кибернетики. Материалы заочного семинара XIX международной конференции. — 2020. — С. 129–132.

[7] Buhrman H., Patro S., Speelman F. A Framework of Quantum Strong Exponential-Time Hypotheses //STACS 2021. — LIPIcs. — 2021.— Т. 187. — С. 19:1–19:19.

[8] Durr C., Hoyer P. A quantum algorithm for finding the minimum //arXiv preprint quant-ph/9607014. — 1996.

Квантовая версия предсказания результирующего класса ансамблевыми методами для задач бинарной классификации

Хадиев Камиль Равилевич[1], **Сафина Лилия Ильхамовна**[2]

[1] Казанский федеральный университет, e-mail: kamilhadi@gmail.com

[2] Казанский федеральный университет, e-mail: liliasafina94@gmail.com

В данной работе мы предлагаем комбинацию квантового и вероятностного алгоритмов для ускорения процесса предсказывания результирующего класса для задачи бинарной классификации. Пусть $O(T)$ — асимптотическая временная сложность предсказывания в некоторой модели машинного обучения. Рассмотрим ансамблевую модель машинного обучения, состоящего из N небольших классификаторов. Тогда временная сложность предсказания результирующего класса составит $O(N \cdot T)$. Временная сложность нашего алгоритма равна $O\left(\sqrt{N} \cdot T\right)$.

Введение

Небольшие классификаторы имеют тенденцию переобучаться, в связи с чем на практике используются ансамлебвые методы, где результаты, полученные маленькими классификаторыми, усредняются по некоторому правилу. Наш алгоритм предсказания работает для задачи бинарной классификации, такая задача актуальна и часто встречается на практике: предсказание решения почтовой системы — является ли письмо спамом или нет, предсказание пола, есть ли аварийная ситуация или нет угрожающих факторов и так далее. К тому же, иногда задачи многоклассовой классификации сводятся к задаче бинарной классификации.

Мы предлагаем применить квантовую подпрограмму для предсказания номера класса и его вероятности. Наш алгоритм имеет временное ускорение по сравнению с классической версией. Это может быть актуально, когда количество маленьких классификаторов — очень большое число.

Вероятностный и квантовый алгоритмы

Рассмотрим следующим алгоритм предсказания номера класса. Пусть каждый маленький классификатор возвращает вероятность того, что входной объект принадлежит первому классу. Пусть p_i — вероятность принадлежности входного объекта первому классу, полученная i-ым классификатором. Среди N классификаторов равновероятностно выберем номер классификатора, например, j, через который пропустим входной объект, и получим вероятность того, что входной объект принадлежит первому классу,

равную $p = \frac{1}{N} \cdot p_j$. В классическом случае такой подход неэффективен. Однако рассмотрим квантовую систему, где каждый классификатор вернёт номер первого и второго класса с некоторыми амплитудами: $\alpha_i|0\rangle + \beta_i|1\rangle$, где $|0\rangle$ — состояние первого класса, α_i — его амплитуда, причем $\alpha_i^2 = p_i$. Соответственно, $|1\rangle$ — состояние второго класса.

Тогда состояние, полученное ансамблевой моделью, будет:

$$|q\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \left(\alpha_i|0\rangle + \beta_i|1\rangle\right).$$

Таким образом, вероятность получить первый класс составит $p = \frac{1}{N} \sum_{i=1}^{N} \alpha_i^2$.

Мы предлагаем применить квантовые алгоритмы для оценки амплитуды квантового состояния.

Квантовый алгоритм для вычисления вероятности

В работе [1] авторы предлагают два способа для оценки амплитуды — подпрограммы **QSearch**, основанный на их алгоритме **Amplitude Amplification**, и **Amp_Est**, основанный на прямом и обратном квантовом преобразовании Фурье [3]. В работе авторы делят элементы на «хорошие» и «плохие». В нашем случае «хорошим» элементом будет считаться первый класс.

Лемма 1. *Метод* ***QSearch*** *работает за* $\Theta\left(\frac{1}{\sqrt{p}}\right)$*, где* p *— это вероятность получить первый класс.*

Лемма 2. *Метод* ***Amp_Est*** *работает за* $\Theta\left(\sqrt{N}\right)$*, где* N *— это количество классификаторов.*

В работе [2] мы предлагаем свой способ вычислить амплитуду «хорошего» элемента. Он так же основан на методе **Amplitude Amplification**.

Лемма 3. *Алгоритм оценки амплитуды, основанный на методе* ***Amplitude Amplification****, работает за* $O\left(\frac{1}{\sqrt{p}}\right)$*, где* p *— это вероятность получить первый класс.*

На начальное квантовое состояние мы подействуем некоторым квантовым (или классическим) алгоритмом A. В нашем случае A — это алгоритм предсказывания на каждом классификаторе. За счёт свойства квантового параллелизма алгоритм A отработает параллельно на всех классификаторах. Алгоритм A поменяет суммарные амплитуды первого и второго классов. Пусть $O(T)$ — асимптотическая сложность алгоритма A, время работы алгоритма для предсказания на одном классификаторе в классическом случае (в квантовом случае работает параррельно).

Далее, чтобы найти усреднённую между всеми классификаторами вероятность получить первый класс, нам необходимо запустить одну из квантовых подпрограмм: наш алгоритм оценки амплитуды, **QSearch**, или **Amp_Est**. Эти алгоритмы позволят найти вероятность получить первый класс. Мы можем установить граничное значение p, при котором результатом бинарной классификации будет первый класс. Например, если $p \geq 0.5$. Временная сложность нашего квантового предсказания на N классификаторах ансамблевого метода составит $O\left(\sqrt{\frac{1}{p}} \cdot T\right)$.

Теорема 1. *Квантовая версия предсказывания результирующего класса для задачи бинарной классификации работает за* $O\left(\sqrt{N} \cdot T\right)$*, где* $O(T)$ *— время работы предсказывания на одном классификаторе, а* N *— количество классификаторов.*

Доказательство. Наш метод оценки амплитуды и метод **QSearch** работают за $O\left(\sqrt{\frac{1}{p}}\right)$, нам необходимо доказать, что $\frac{1}{p} \leq N$, из этого следует, что $\sqrt{\frac{1}{p}} \leq \sqrt{N}$.

Данные методы основаны на алгоритме **Amplitude Amplification**, который за $O\left(\sqrt{\frac{1}{p}}\right)$ находит «хороший» элемент в последовательности. Допустим, что хотя бы один классификатор вернул вероятность перового класса, близкую к 1, тогда $p \geq \frac{1}{N}$, отсюда следует, что $\frac{1}{p} \leq N$.

Если же ни один классификатор не вернул высокую вероятность первого класса, значит, за $O\left(\sqrt{N}\right)$ запусков **Amplitude Amplification** ни разу не нашёл «хороший» элемент, из этого следует, что $p < \frac{1}{N}$, данная вероятность очень мала и позволяет нам вернуть в качестве ответа второй класс.

Теорема 1 доказана.

Заключение

В данной работе мы кратко описали идею ускорения предказания результирующего класса, основанную на квантовых алгоритмах оценки амплитуды квантового состояния, для задачи бинарной классифкации.

Работа выполнена за счёт средств субсидии, выделенной Казанскому федеральному университету для выполнения государственного задания в сфере научной деятельности, проект 0671-2020-0065.

Список литературы

[1] Brassard G., Hoyer P., Mosca M., Tapp, A., Quantum amplitude amplification and estimation //Contemporary Mathematics. — Т. 305, — С. 53-74, — 2002.

[2] Khadiev K., Safina L. The Quantum Version of Random Forest Model for Binary Classification Problem // CEUR, — С. 30–35, — 2021.

[3] Гайнутдинова А. Ф. Квантовые вычисления //метод. пособие.–Казань: Казанский государственный университет, — Т. 272, — С. 88-92, — 2009.

Квантовый алгоритм для распознавания конкатенации двух палиндромов

Хадиев Камиль Равилевич[1], Серов Данил Юрьевич[2]

[1] Казанский федеральный университет, Казанский физико-технический институт им. Е.К. Завойского ФИЦ Казанский научный центр РАН, e-mail: kamilhadi@gmail.com

[2] Казанский федеральный университет, e-mail: serovdanilru@gmail.com

В данной работе рассматривается задача о распознавании языка конкатенации двух палиндромов. Формально задача звучит следующим образом. Рассмотрим язык $L_{rev} = \{uu^r vv^r\, u, v \in \Sigma^*\}$, где u^r — строка u, записанная в обратном порядке, Σ — алфавит входных символов. Рассмотрим входную строку s длинны n. Для некоторого наперед известного числа $0 < \varepsilon < 1$ выполняется одно из следующих утверждений:

- Строка s принадлежит языку L_{rev};
- Строка s отличается от любой из строк из языка L_{rev} как минимум в $n \cdot \varepsilon$ символах.

В работе задача исследуется с точки зрения квантовых вычислений, в частности модели запросов [1, 2]. В рамках работы, говоря о сложности, мы будем подразумевать запросную сложность. Известно, что существуют задачи, для которых разработаны квантовые алгоритмы, имеющие меньшую запросную сложность чем классические аналоги [3]. В том числе квантовые алгоритмы разрабатывались и для задач по обработке строк [4,5].

Для рассматриваемой задачи известна верхняя [6] и нижняя [7] оценки на вероятностную, а значит классическую сложность. Верхняя оценка составляется $O\left(\frac{1}{\varepsilon} n^{\frac{1}{2}} \log n\right)$, нижняя же оценка равна $\Omega\left(n^{\frac{1}{2}}\right)$. Таким образом они достаточно близки.

В данной работе был разработан квантовый алгоритм, сложность которого равна $O\left(\frac{1}{\varepsilon} n^{\frac{1}{3}} \log n\right)$. Таким образом, квантовый алгоритм эффективнее чем любой классический. Алгоритм использует в качестве составной части квантовый алгоритм Гровера [8]. В связи с этим, также как и для других алгоритмов базирующихся на алгоритме Гровера, запросная сложность алгоритма отличается от временной сложности в логарифм раз.

Приведем краткое описание алгоритма. Рассмотрим строку s длины n и два множества $I = \{0, 1, ..., n^{1/3} - 1\}$ и $J = \{0, n^{1/3}, 2 \cdot n^{1/3}, ..., n^{2/3} \cdot n^{1/3}\}$, а также число $m = \frac{1}{\epsilon} 2 \log_2 n$. Алгоритм состоит из следующих шагов:

1. Выбираем m чисел $p_1, ..., p_m$ равновероятно из набора $\{0, ..., n\}$, т.е. $p_i \in \{0, ..., n\}$, $i \in \{1, ..., m\}$;
2. Создадим префиксное дерево [9] T и добавим в него строки

$$x^i = (s_{(i-p_1) \ mod \ n}, ..., s_{(i-p_m) \ mod \ n})$$

для каждого $i \in I$;

3. Найдем индекс $j \in J$ такой, что для некоторого $i \in I$ выполнилось условие $y^j = x^i$, где

$$y^j = (s_{(j+p_1) \ mod \ n}, ..., s_{(j+p_m) \ mod \ n})$$

Поиск выполняется с помощью Алгоритма Гровера.

Теорема 1. *Приведённый квантовый алгоритм решает задачу распознавания языка L_{REV}. Его сложность равна $O\left(\frac{1}{\varepsilon} n^{\frac{1}{3}} \log n\right)$ и вероятность ошибки константа не превышающая $\frac{1}{2}$.*

Доказательство. Временная сложность выбора m случайных чисел составляет $O(m)$. Далее для каждого $i \in I$, добавляется строка x^i в префиксное дерево. Длина каждой строки равна m, общее количество строк равно $|I|$. Сложность добавления всех строк x^i равна $O(|I| \cdot m)$. Сложность поиска $j \in J$ такого, что y^j есть в дереве T алгоритмом Гровера составляет $O(\sqrt{|J|} \cdot m)$. Итоговая сложность алгоритма равна

$$O\left((|I| + \sqrt{|J|}) \cdot m\right) = O\left((n^{1/3} + \sqrt{n^{2/3}}) \cdot \tfrac{1}{\epsilon} \log n\right) = O\left(\tfrac{1}{\epsilon} n^{1/3} \log n\right)$$

В случае, если входной набор принадлежит языку L_{REV}, обязательно найдется искомый j, таким образом, вероятность ошибки равна вероятности ошибки алгоритма Гровера, т.е. $\frac{1}{2}$. Если же входной набор не принадлежит языку, то тогда из-за условия, что входной набор отличается от ближайшего представителя языка как минимум в εn позициях, легко показать, что при достаточно большом n вероятность ошибки можно ограничить константой $\frac{1}{4}$. **Теорема 1 доказана.**

Работа выполнена за счёт средств субсидии, выделенной Казанскому федеральному университету для выполнения государственного задания в сфере научной деятельности, проект 0671-2020-0065.

Список литературы

[1] Ambainis A. Understanding quantum algorithms via query complexity // Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018. — 2018. — С. 3265–3285.

[2] Ablayev F., Ablayev M., Huang J.Z., Khadiev K., Salikhova N., Wu D. On quantum methods for machine learning problems part I: Quantum tools //Big Data Mining and Analytics. — 2019. — Т. 3. — №. 1. — С. 41–55.

[3] Stephen Jordan. Quantum algorithms zoo/http://quantumalgorithmzoo.org/ — 2021.

[4] Ramesh H., Vinay V. String matching in $O(\sqrt{n} + \sqrt{m})$ quantum time //Journal of Discrete Algorithms. — 2003. — Т. 1. — №. 1. — С. 103–110.

[5] Montanaro A. Quantum pattern matching fast on average //Algorithmica. — 2017. — Т. 77. — №. 1. — С. 16–39.

[6] Parnas M., Ron D., Rubinfeld R. Testing membership in parenthesis languages //Random Structures and Algorithms. — 2003. — Т. 22. — №. 1. — С. 98–138.

[7] Alon, N., Krivelevich, M., Newman, I., Szegedy, M., Regular languages are testable with a constant number of queries //SIAM Journal on Computing. — 2001. — Т. 30. — №. 6. — С. 1842–1862.

[8] Grover L. K. A fast quantum mechanical algorithm for database search //Proceedings of STOC'96. — 1996. — С. 212–219.

Об аффинных автоматах

Хадиева Алия Ихсановна[1], **Abuzer Yakaryilmaz**[2]

[1] Казанский Федеральный Университет, Университет Латвии, e-mail: aliya.khadi@gmail.com
[2] Университет Латвии, e-mail: abuzer.yakaryilmaz@gmail.com

Введение

Рассматривается модель конечных автоматов, расширяющая вероятностные. В работе [1] содержится описание конечной аффинной системы как нелинейного обобщения вероятностной системы, допускающей отрицательные значения вероятностей. Пусть $E = \{e_1, \dots e_n\}$ - множество базисных состояний $n-$мерной вероятностной системы. Аффинное состояние — линейная комбинация из E. $v = (a_1, a_2 \dots a_n)^T$ - вектор аффинных состояний, где элементы вектора могут быть любыми вещественными числами, при этом сумма элементов сохраняет единицу. Любое аффинное преобразование — линейный оператор, то есть матрицы преобразования таковы, что в каждом столбце сумма элементов равна 1. В качестве измерения системы можно принять аналог измерения квантовой системы с некоторыми поправками. Это преобразование назовем оператором взвешивания. Пусть $|v| = |a_1| + |a_2| + \ldots + |a_n| \geq 1$. Тогда вероятность обозревания состояния

a_k равна $\frac{|a_k|}{v}$ для $1 \leq k \leq n$. После проведения операции взвешивания система схлопывается в одно детерминированное состояние. Введем понятие Аффинного Конечного Автомата, исходя из определения, данного в работе [1]. Положим, Σ — входной алфавит, не содержащий признаков начала и конца строки $\odot, \$$. $\tilde{\Sigma} = \Sigma \bigcup \{\odot, \$\}$. $\tilde{w} = \odot w \$$, где $w \in \Sigma^*$. Будем считать $|w|$ длиной входного слова w, $|w|_\sigma$ — числом вхождений символа $\sigma \in \Sigma$ в слове w, w_j — j-ым элементом в строке w. Для заданного автомата M $f_M(w)$ является обозначением вероятности (значения) принять слово w автоматом M. Аффинный конечный автомат (АКА) — набор из 5 элементов $M = (E, \Sigma, \{A_\sigma | \sigma \in \tilde{\Sigma}\}, e_s, E_a)$, такой что E — множество детерминированных состояний, $e_s \in E$ — начальное состояние, $E_a \subset E$ — множество принимающих состояний. A_σ — матрица аффинного преобразования системы для входного символа $\sigma \in \Sigma$. Пусть $w \in \Sigma$ — некоторое входное слово. АКА M считывает $\tilde{w}$ посимвольно начиная с крайнего левого символа. После считываения автоматом j- ого символа аффинная система описывается вектором состояний $v_k = A_{\tilde{w}_j} v_{k-1} = A_{\tilde{w}_j} A_{\tilde{w}_{j-1}} \dots A_{\tilde{w}_1} v_0$, где v_0 — вектор вида $(e_1, e_2, \dots e_{|E|})^T$, такой что $e_s = 1$, все остальные элементы нулевые. $1 \leq j \leq |\tilde{w}|$. Финальное состояние обозначим за $v_f = v_{|\tilde{w}|}$. После прочтения всего слова $\tilde{w}$ применяется оператор взвешивания, после чего вероятность принятия слова w автоматом M будет равна $f_M(w) = \sum\limits_{e_k \in E_a} \frac{|v_f[k]|}{|v_f|} \in [0, 1]$

Данная работа посвящена построению аффинных автоматов по распознаванию языка $SQUARE = \{w \in \Sigma^* | |w|_0 = |w|_1^2\}$ в алфавите $\Sigma = \{0, 1\}$

Также рассматривается модель недетерминированного аффинного автомата, дополненного классическими состояниями (Н+А)КА. Эта модель совмещает аффинную и классическую системы так, что аффинная система контролируется классическим состоянием, в котором находится автомат. Введем определение (Н+А)КА. Недетерминированный аффинный конечный автомат (Н+А)КА Q — набор из 8 элементов:

$Q = (E, S, \Sigma, \{A_\sigma | \sigma \in \tilde{\Sigma}\}, \delta, e_s, s_0, E_a, S_a)$.

К уже известному определению аффинного конечного автомата добавляется множество классический состояний S, функции переходов для классических состояний δ и начальное классическое состояние s_0. Находясь в некотором классическом состоянии $s_i \in S$, автомат производит преобразования аффинной системы, соответствующие s_i. В свою очередь, переход классической системы из состояния в другие состояния производится недетерминированно, то есть $\delta : \tilde{\Sigma} \times S \to [S]$, где $[S]$ — множество всех подмножеств S. S_a — множество принимающих классических состояний.

В данной работе рассматривается построение (Н+А)КА для NP-полной задачи $SUBSETSUM$.

АКА с ограниченной односторонней ошибкой

Построим аффинный конечный автомат R, распознающий язык $SQUARE$. Для этого введем 5-мерный вектор аффинных состояний $v = (1 \quad 0 \quad 0 \quad 0 \quad 0)^T$. Множество принимающих состояний — $\{e_1\}$. R будет принимать входное слово с вероятностью $p_{acc} = \frac{|v[1]|}{\sum\limits_{i=\overline{1,5}} |v[i]|}$. Вектор v преобразуется путем домножения на матрицы S_1 или S_0 в зависимости от считанного автоматом входного символа. При прочтении символа начала строки автоматом вектор v не видоизменяется. При прочтении символа 0 вектор состояний умножается на матрицу S_0, при этом $v[2]$ инкрементируется. При прочтении единицы вектор домножается на S_1, тогда $v[3]$ увеличивается на 1, а $v[4]$ сохраняет значение квадрата числа считанных единиц. $v[5]$ необходим для нормализации вектора и хранит значение, равное $1 - \sum\limits_{i=\overline{1,4}} v[i]$. После прочтения признака конца строки вектор состояний преобразуется матрицей $A_\$$, $v[3]$ принимает значение $v[2] - v[4]$. В случае $|w|_0 = |w|_1^2$ финальный вектор v_f будет равным $(1 \quad 0 \quad 0 \quad 0 \quad 0)^T$ и входное слово принимается с вероятностью 1, иначе $v_f = (1 \quad t(|w|_0 - |w|_1^2) \quad 0 \quad 0 \quad -t(|w|_0 - |w|_1^2))^T$, и входное слово примется с вероятностью $p = \frac{1}{1+2t(||w|_0 - |w|_1^2|)}$, где t — некоторое целое число, взятое для уменьшения вероятности ошибки вычисления. Матрицы пробразования представлены ниже.

$$A_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & -1 & -1 & 0 \end{pmatrix}. \; A_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 2 & 1 & 0 \\ 0 & -1 & -1 & -1 & 0 \end{pmatrix}. \; A_\$ = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & t & 0 & -t & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -t & 0 & t & 0 \end{pmatrix}$$

Построение (Н+А)КА для задачи $SUBSETSUM$

В задаче $SUBSETSUM$ дано число K и множество чисел X. Необходимо определить, существует ли такое подмножество Y множества A, что сумма элементов Y будет равна числу K. Формально язык $SUBSETSUM$ в алфавите $\Sigma = \{0, 1, \sharp\}$ представим множеством строк $\{k\sharp x_1\sharp x_2\sharp \dots x_n\}$, где $k, x_1, x_2, \dots x_n$ - некоторые числа, заданные в бинарном виде, для которых выполняется следующее условие: существует такое $Y \subset X$, что $\sum\limits_{x_j \in Y} x_j = k$. Эта задача относится к NP-полным.

Построим (Н+А)КА R для задачи $SUBSETSUM$. Множество классических состояний содержит 6 состояний $s_{init}, s_1, s_2, s_3, s_4, s_{exit}$. 4-мерный вектор аффинных состояний обозначим за v. Изначально $v = v_0 = (1000)^T$. Множество принимающих аффинных состояний $\{e_1\}$. Множество принимающих классических состояний $S_a = \{s_2, s_3\}$. Считав число k, автомат

сохраняет величину k в $v[2]$. После чтения каждого символа-разделителя $\sharp$ автомат недетерминированно "решает" , включать ли следующее число в множество Y или пропустить. Таким образом из $v[2]$ вычитаются те числа x_j, из которых автомат "решил"сформировать множество Y. После того как будет считан символ конца строки, производится операция взвешивания. При нулевом значении $v[2]$ автомат принимает входное слово с вероятностью 1. В противном случае слово принимается с вероятностью $\frac{1}{1+2(|k-\sum\limits_{x_j\in Y} x_j|)}$. Автомат R устроен следующим образом. s_{init} — начальное классическое состояние. $v = v_0 = (1 \quad 0 \quad 0 \quad 0)^T$ На входном символе 0 (1) из состояния s_{init}: автомат переходит в состояние s_1, вектор v преобразуется умножением на матрицу A_0 (A_1, соответственно). На входном символе $\sharp$ из s_{init} происходит переход в состояние s_{exit}, входное слово отвергается с вероятностью 1. На входном нуле (единице) в состоянии s_1 автомат ведет себя так же, как в состоянии s_{init}, но на входном $\sharp$ переходит в s_2. Из s_2 осуществляется недетерминированный переход в состояние s_3 (при этом на входном нуле (единице) v умножается на A'_0 (A'_1, соответственно)) или в s_4 без изменения вектора v. На входном символе $\sharp$ из s_2: переход в состояние s_{exit}, входное слово отвергается с вероятностью 1. В s_3 на входном нуле (единице) состояние не меняется, вектор v умножается на A'_0 (A'_1). Но на входном $\sharp$ автомат переходит в s_2, а вектор умножается на $A_\sharp$. Находясь в s_4 автомат меняет состояние, получив на вход $\sharp$, и переходит в s_2. Вектор v остается неизменным.

$$A_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 \end{pmatrix}. \; A_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 \end{pmatrix}. \; A'_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & -1 & -1 & 0 \end{pmatrix}.$$

$$A'_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & -1 & -1 & 0 \end{pmatrix}. \; A_\sharp = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \end{pmatrix}.$$

Работа выполнена при поддержке РФФИ (проект № 20-37-70080).

Список литературы

[1] Díaz-Caro A., Yakaryılmaz A. Affine computation and affine automaton //International Computer Science Symposium in Russia. // Springer, Cham — 2016. — С. 146-160.

[2] Villagra M., Yakaryılmaz A. Language recognition power and succinctness of affine automata //Natural Computing. — 2018. — Т. 17. — №. 2. — С. 283-293.

[3] Hirvensalo M., Moutot E., Yakaryılmaz A. On the computational power of affine automata //International Conference on Language and Automata Theory and Applications. — Springer, Cham, 2017. — С. 405-417.

[4] Ibrahimov R. et al. Error-free affine, unitary, and probabilistic OBDDs //International Conference on Descriptional Complexity of Formal Systems. — Springer, Cham, 2018. — С. 175-187.

[5] Nakanishi M. et al. Exact Affine Counter Automata //EPTCS 252. – С. 205. — 2017.

Saliency Map Generation through Lateral Inhibition Mechanism

Jiang Lei

Moscow state university, e-mail: lei.jiang@hipasus.com

Saliency map highlights portion of the image that contributes to a classification decision, hence it is an effective tool for the interpretation of convolutional neural networks (CNN). In recent years, various saliency detection methods have been proposed: Guided Backpropagation(GBP) [1], Class Activation Mapping(CAM) [2], Grad-CAM [3], Grad-CAM++ [4] and so on. Although some saliency detection methods are able to generate the saliency map, they are independent of the model and data generation process, and may not best explain the relationship between the inputs and outputs of the model during learning or to debug the model.

Inspired by the lateral inhibition(LI) mechanism [5] in the brain, we introduced it in artificial neural networks and applied it to saliency detection. Experiments show that our approach has significant advantages in comparison with the mainstream methods.

Algorithm of Saliency Map

During backpropagation, gradients are generated for all feature maps; these then work together to update the corresponding weights. However, previous work [6] showed that not every gradient is important for training. Our model assigns importance to the feature gradients of each convolutional layer with the Laplacian of Gaussian (LoG) operator, which has a similar Mexican hat distribution as attention modulation in the biological brain. During backpropagation, LoG inhibits part of the gradients that are less important.

Initialize G^l = Tensor of gradients in layer l, q = Quantile, σ = Sigma, s = Kernel size, C = Number of channels , K = Number of sets, P= Prediction of interest category. Let l_c be the c-th feature map of layer l, a_{ij} is the output of

neuron in the coordinate $\{i, j\}$:

$$G^l = [\frac{\partial P}{\partial a_{ij}^{l_c}}]_{C \times u \times v}, \quad i = 1, \dots u, j = 1, \dots, v, c = 1, \dots, C \tag{4}$$

Divide gradients into minicolumns:

$$M_{ij}^l(k) = [\frac{\partial P}{\partial a_{ij}^{l_{c'}}}]_{\frac{C}{K} \times 1}, \quad c' = 1, \dots, \frac{C}{K}, k = 1, \dots, K \tag{5}$$

and compute l_2 norm of each minicolumn: $||M_{i,j}^l(k)||_2$ to get K matrices:

$$D^l(k) = [||M_{ij}^l(k)||_2]_{u \times v}, \quad k = 1, \dots, K \tag{6}$$

Perform LoG convolution on each matrix $D^l(k)$ with given parameters σ and s:

$$\delta_{ij}^l(k) = LoG_{\sigma,s}(D^l(k)), \quad k = 1, \dots, K \tag{7}$$

For every activation layer A^l, get δ^l. Set $K = 1$, i.e. each pixel channel can be seen as one minicolumn. Then resize δ^l from $u \times v$ to $H \times W$, where H and W are the height and width of the input image:

$$\delta^l = LoG_{\sigma,s}(D^l) \tag{8}$$

$$\delta^l \in \mathbb{R}^{u \times v} \rightarrow \delta^l \in \mathbb{R}^{H \times W} \tag{9}$$

Finally, in order to combine information from each layer, we sum the δ^l of all activation layers to get the saliency map F:

$$F = \sum_{l=1}^{t} \delta^l, \quad l \in [1, t] \tag{10}$$

where t is the number of activation layers.

With the aim of quantitatively comparing various saliency map methods, we propose a measure based on Intersection over Union (IoU) scores. We select 15% of pixels with the largest F values (F is calculated by Equation 10) in the saliency map generated by the different methods as the region of interest. The smallest and largest coordinate points of these pixels are used to form a rectangular bounding box and the IoU score is calculated with the real bounding box where the target is located. As shown in Table 1, we verify the superiority of our method on the PASCAL VOC 2007 and ImageNet datasets.

Dataset	Grad-CAM	Grad-CAM++	LI-Map
PASCAL VOC 2007	0.44	0.45	**0.5**
ImageNet	0.46	0.46	**0.49**

Table 1: The IoU results on PASCAL VOC 2007 and ImageNet

References

[1] Springenberg J T, Dosovitskiy A, Brox T, et al. Striving for simplicity: The all convolutional net[J]. arXiv preprint arXiv:1412.6806, 2014.

[2] Zhou B, Khosla A, Lapedriza A, et al. Learning deep features for discriminative localization[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2921-2929.

[3] Selvaraju R R, Cogswell M, Das A, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization[C]//Proceedings of the IEEE international conference on computer vision. 2017: 618-626.

[4] Chattopadhay A, Sarkar A, Howlader P, et al. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks[C]//2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2018: 839-847.

[5] Amari S. Dynamics of pattern formation in lateral-inhibition type neural fields[J]. Biological cybernetics, 1977, 27(2): 77-87.

[6] Lan J, Liu R, Zhou H, et al. Lca: Loss change allocation for neural network training[J]. arXiv preprint arXiv:1909.01440, 2019.

О реализации булевых функций программами без памяти

Чашкин Александр Викторович

МГУ им. М. В. Ломоносова, e-mail: chashkin_oo@inbox.ru

Неветвящейся программой с условной остановкой над множеством независимых булевых переменных $X = \{x_1, \ldots, x_n\}$ назовем список $p_1, \ldots, p_L$ последовательно выполняемых команд двух видов — вычислительных команд и команд остановки. Если p_i — вычислительная команда, то она присваивает внутренней переменной y_i значение $f_i(z_1, z_2)$, где f_i — двухместная булева функция и $z_1, z_2 \in X \bigcup \{y_1, \ldots, y_{i-1}\}$. Если p_i — команда остановки $\texttt{Stop}(z_1, z_2)$, где $z_1, z_2 \in X \bigcup \{y_1, \ldots, y_{i-1}\}$, то эта команда останавливает вычисления если $z_1 = 1$ и объявляет результатом работы значение

z_2. Если $z_1 = 0$, то выполняется следующая команда программы. Если ни одна команда остановки не остановила программу, то ее значением объявляется значение последней внутренней переменой. Программы, в которых каждая внутренняя переменная используется в качестве аргумента какой-либо команды только один раз, будем называть программами без памяти, а символы, отвечающие таким программам, будем отмечать нижним индексом 1. Сложностью программы P называется число $C(P)$ ее команд. Величина

$$T(P) = 2^{-n} \sum_{x \in \{0,1\}^n} T_P(x),$$

где $T_P(x)$ — число команд, выполненных программой P на наборе x до ее остановки, называется средним временем работы P. Если для булевой функции f и любого булева набора x справедливо равенство $f(x) = P(x)$, то программа P вычисляет функцию f. Величину

$$T_1(f) = \min T(P),$$

где минимум берется по всем программам без памяти, вычисляющим f, назовем средней формульной сложностью f. Пусть $T_1(n) = \max_{f \in P_2(n)} T_1(f)$ — функция Шеннона средней формульной сложности. В [2] показано, что

$$T_1(n) \sim \frac{2^{n-1}}{\log_2 n}.$$

Пусть $M(n)$ — множество всех n-местных монотонных булевых функций. Для функции $T_1^M(n) = \max_{f \in M(n)} T_1(f)$ справедливо следующее утверждение.

Теорема 1. *Пусть $n \to \infty$. Тогда*

$$T_1^M(n) = \Theta\left(\frac{2^n}{n \log_2 n}\right).$$

Нижняя оценка теоремы доказывается мощностным методом. Верхняя оценка основана на формулируемой ниже лемме 1, доказанной в [5], и близко к доказательству верхних оценок средней сложности монотонных функций из [3, 5].

Пусть $\boldsymbol{i} = (i_1, i_2, \ldots, i_k)$ и $\boldsymbol{\alpha} = (\alpha_1 \alpha_2 \ldots \alpha_k)$. Символом $f_{\boldsymbol{i}}^{\boldsymbol{\alpha}}(x)$ обозначим n-местную функцию с k фиктивными переменными, получающуюся из n-местной булевой функции f подстановкой констант α_j вместо ее i_j-х аргументов, а символом $x_{\boldsymbol{i}}^{\boldsymbol{\alpha}}$ — булев набор длины n, у которого i_j-е разряды равны величинам α_j.

Лемма 1. *При $n \to \infty$ и $k = o(n)$ для любой n-местной монотонной булевой функции f найдется такой набор $\boldsymbol{i}$ длины k, что $f^{\mathbf{0}}_{\boldsymbol{i}}(x) \neq f^{\mathbf{1}}_{\boldsymbol{i}}(x)$ не более чем для*

$$\frac{k \cdot 2^n}{\sqrt{\pi n/2}}(1 + o(1))$$

различных наборов x длины n.

Положим $s = \lceil \log_2 \log_2 n \rceil$. В силу леммы 1 для любого $k \in \{1, \ldots, s\}$ найдется такой набор $\boldsymbol{i}_k$ длины 2^k, что $f^{\mathbf{0}}_{\boldsymbol{i}_k}(y) \neq f^{\mathbf{1}}_{\boldsymbol{i}_k}(y)$ не более чем для

$$\frac{2^k \cdot 2^n}{\sqrt{\pi n/2}}(1 + o(1))$$

различных наборов y длины n. Пусть A_k — множество всех таких наборов, $A_{s+1} = \{0,1\}^n$. Нетрудно видеть, что функция $f^{\mathbf{0}}_{\boldsymbol{i}_k}(x) \oplus f^{\mathbf{1}}_{\boldsymbol{i}_k}(x)$ будет характеристической функцией множества A_k, и что функции $f^{\mathbf{0}}_{\boldsymbol{i}_k}(x)$ и $f^{\mathbf{1}}_{\boldsymbol{i}_k}(x)$ можно вычислить формулами, сложность которых [1, 4] есть

$$O\left(\frac{2^{n-2^k}}{\sqrt{n-2^k}\log_2(n-2^k)}\right) = O\left(\frac{2^{n-2^k}}{\sqrt{n}\log_2 n}\right). \tag{1}$$

Вычисляющую функцию $f(x)$ программу P представим в виде последовательности подпрограмм

$$P = P_s P_{s-1} \ldots P_k \ldots P_1 P_0.$$

Здесь для $k = s, \ldots, 1$ каждая подпрограмма P_k выполняет следующие действия: 1) вычисляет значения функций $f^{\mathbf{0}}_{\boldsymbol{i}_k}(x)$ и $f^{\mathbf{0}}_{\boldsymbol{i}_k}(x) \oplus f^{\mathbf{1}}_{\boldsymbol{i}_k}(x)$; 2) объявляет значением вычисляемой функции значение $f^{\mathbf{0}}_{\boldsymbol{i}_k}(x)$; 3) останавливает вычисления если $f^{\mathbf{0}}_{\boldsymbol{i}_k}(x) \oplus f^{\mathbf{1}}_{\boldsymbol{i}_k}(x) = 1$. Подпрограмма P_0 является формулой, вычисляющей $f(x)$.

Так как в программе P каждая подпрограмма P_k работает только на наборах множества A_{k+1}, то

$$T(P) = 2^{-n} \sum_{k=s}^{0} |A_{k+1}| C(P_k). \tag{2}$$

В силу (1) и неравенства $k+1 \le 2^k - k + 1$ слагаемые в (2) удовлетворяют неравенствам

$$|A_{s+1}|C(P_s) = 2^n \cdot O\left(\frac{2^{n-2^s+1}}{\sqrt{n}\log_2 n}\right) = O\left(\frac{2^{2n}}{n\sqrt{n}\log_2 n}\right),$$
$$|A_{k+1}|C(P_k) \lesssim \frac{2^{k+1}2^n}{\sqrt{\pi n/2}} \cdot O\left(\frac{2^{n-2^k+1}}{\sqrt{n}\log_2 n}\right) = O\left(\frac{2^{2n-k}}{n\log_2 n}\right),\ k = 1, \ldots, s-1$$
$$|A_1|C(P_0) \lesssim \frac{2 \cdot 2^n}{\sqrt{\pi n/2}} \cdot O\left(\frac{2^n}{\sqrt{n}\log_2 n}\right) = O\left(\frac{2^{2n}}{n\log_2 n}\right).$$

Таким образом,

$$T(P) = O\left(\frac{2^n}{n\sqrt{n}\log_2 n} + \sum_{k=s-1}^{1} \frac{2^{n-k}}{n\log_2 n} + \frac{2^n}{n\log_2 n}\right) = O\left(\frac{2^n}{n\log_2 n}\right).$$

Список литературы

[1] Андреев А. Е. О сложности монотонных функций // Вестник Московского государственного университета. Серия 1. Математика. Механика, — 1985, № 4. — С. 83–87.

[2] Забалуев Р. Н. О реализации булевых функций программами одного типа // Вестник Московского государственного университета. Серия 1. Математика. Механика, — 2005, № 5. — С. 9—13.

[3] Забалуев Р. Н. О средней сложности монотонных функций // Дискретая математика. — 2006, № 2. — С. 71–83.

[4] Чашкин А. В. О сложности реализации булевых функций формулами // Дискретный анализ и исследование операций. Серия 1. — 2005, № 2. — С. 56–72.

[5] Чашкин А. В. Оценки средней сложности монотонных булевых функций // Дискретная математика. — 2016, № 2. — С. 146–153.

Вероятность идентификации цепей Маркова на основе структурных признаков эталонных стохастических матриц

Шалагин Сергей Викторович[1], **Нурутдинова Алсу Рафаиловна**[2]

[1] Казанский национальный исследовательский технический университет им. А.Н. Туполева, e-mail: sshalagin@mail.ru

[2] Казанский федеральный университет, e-mail: nurutdinova@mail.ru

Решена задача определения вероятности идентификации цепей Маркова, задаваемой на основе определенной эргодической стохастической матрицы (ЭСМ). В качестве эталонных объектов для идентификации служат ЭСМ, имеющие нулевые элементы в заданных позициях. Определены вероятности того, что цепь Маркова будет идентифицирована как не принадлежащая каждому из эталонных объектов с заданными структурными признаками.

При идентификации цепей Маркова (ЦМ) [1] модифицированным методом Рабинера [2, 3] существует вероятность их ошибочного соотнесения с ЦМ, порождаемыми на основе заданных, эталонных ЭСМ при использовании известных автоматных моделей [4, 5, 6]. Вместе с тем, наличие в эталонных объектах таких структурных признаков, как нулевых элементов в заданных позициях, позволяет однозначно идентифицировать ЦМ как не принадлежащую заданному эталонному объекту. Предложен метод определения вероятности гипотезы о том, что идентифицируемая ЦМ не будет порождена заданным l-м эталонным объектом из множества мощности m – H_l , $l = \overline{1,\ m}$. При этом каждый из m объектов содержит хотя бы одни нулевой элемент.

Пусть для каждого из m эталонных объектов предварительно вычислены двоичные матрицы $Z_l = \left(z_{ij}^{(l)}\right)_{n\times n}$: $z_{ij}^{(l)} = 1$, если элемент l-й эталонной ЭСМ $P_l = \left(p_{ij}^{(l)}\right)_{n\times n}$ является нулевым, в противном случае – $z_{ij}^{(l)} = 0$, $i,\ j = \overline{1,\ n}$, $l = \overline{1,\ m}$. Когда ЦМ задана согласно [1] как $(S,\ P,\ \pi_0)$, где $|\,S\,| = n$, $P = (p_{ij})_{n\times n}$ – ЭСМ, то для нее вычисляем предельный вектор распределения состояний π_f.

Предложен метод определения вероятностей принятия H_l относительно ЦМ $(S,\ P,\ \pi_0)$ для каждой из m эталонных ЭСМ. Метод включает в себя два этапа и на входе которого имеем: $P = (p_{ij})_{n\times n}$, π_f, $Z_l = \left(z_{ij}^{(l)}\right)_{n\times n}$, $i,\ j = \overline{1,\ n}$, $l = \overline{1,\ m}$.

Этап 1. Вычисление матрицы $X_l = (P \circ Z_l)_{n\times n}$, $l = \overline{1,\ m}$.

Этап 2. Определение вероятности принятия H_l согласно формуле:

$$h_l = (\pi_f \cdot X_l) \cdot I_{n\times 1}$$

$l = \overline{1,\ m}$, $I_{n\times 1}$ – единичный вектор-столбец.

Замечание 1.

Если на этапе 1 метода получаем $X_l : \forall x_{ij}^{(l)} = 0$, $i,\ j = \overline{1,\ n}$, *то* $h_l = 0$, $l = \overline{1,\ m}$.

В случае, если ЦМ задана как последовательность длины N, то модификация предложенного метода включает три этапа.

Этап 1. Вычисляем вектор распределения ЦМ длины N по состояниям $\bar{\pi}_N$ и частотную матрицу $\bar{P} = (\bar{p}_{ij})_{n\times n}$.

Этап 2. Вычисление матрицы $\bar{X}_l = \left(\bar{P} \circ Z_l\right)_{n\times n}$, $l = \overline{1,\ m}$.

Этап 3. Определение наличия отклонения идентификации согласно формуле: $\bar{h}_l = \left(\bar{\pi}_N \cdot \bar{X}_l\right) \cdot I_{n\times 1}$; если $\bar{h}_l > 0$, то заданная последовательность не относится к l-й эталонной ЭСМ, $l = \overline{1,\ m}$.

Замечание 2.

Если на этапе 2 метода получаем $\bar{X}_l : \forall \bar{x}_{ij}^{(l)} = 0$, $i,\ j = \overline{1,\ n}$, *то* $\bar{h}_l = 0$, $l = \overline{1,\ m}$.

Определены оценки сложности реализации указанного метода и его модификации в зависимости от количества состояний и способа задания идентифицируемой ЦМ, а также от структурных признаков элементов множества эталонных ЭСМ.

Введем следующие базовые операции: поразрядное выполнение операций конъюнкции с общим входом над двоичным вектором заданной длины (AND), умножение (Mlt) и сложение (Sm) чисел, операция увеличения заданного числа на единицу (Inc) и операция деления (Div). Для метода определения вероятности принятия H_l относительно ЦМ $(S,\ P,\ \pi_0)$, $l = \overline{1,\ m}$, вычислительная сложность этапа 1 вычислена как $m \cdot n^2$ AND, а этапа 2 – как $d \cdot n^2$ Mlt и $2d \cdot n\,(n-1)$ Sm, где d “— количество $\bar{X}_l : \bar{h}_l \neq 0$, $d \in [0,\ m]$. Для модификации предложенного метода сложность этапа 1 – N операций Inc, и n операций Div для вычисления $\bar{\pi}_N$ и N операций Inc, $n\,(n-1)$ Sm и n^2 Div для вычисления $\bar{P} = (\bar{p}_{ij})_{n\times n}$. Сложность этапов 2 и 3 модификации метода соответствует сложности этапов 1 и 2 исходного метода. Справедливы

Утверждение 1.

Оценка сложности метода определения вероятности принятия H_l *относительно ЦМ* $(S,\ P,\ \pi_0)$ *имеет вид* $l = \overline{1,\ m}$, – $m \cdot n^2$ *AND,* $d \cdot n^2$ *Mlt и* $2d \cdot n\,(n-1)$ *Sm,* $d \in [0,\ m]$.

Утверждение 2.

Оценка сложности модификации метода определения вероятности принятия H_l относительно ЦМ, заданной как последовательность длины N, имеет вид $l = \overline{1,\ m}$, – $2N$ Inc, $n\,(n+1)$ Div, $n\,(n-1)\,(2d-1)$ Sm, $m \cdot n^2$ AND и $d \cdot n^2$ Mlt, $d \in [0,\ m]$.

Список литературы

[1] Кемени Дж., Снелл Дж. Конечные цепи Маркова. — М.: Наука, 1970. — 272 с.

[2] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition// Proc. IEEE. — 1989. — V. 77, No 2. — P. 257–286.

[3] Нурутдинова А. Р., Шалагин С. В. Идентификация и классификация автоматных марковских моделей методами многопараметрического анализа: монография. — Казань: Изд-во КНИТУ-КАИ, 2019. — 176 с.

[4] Захаров В. .М., Нурмеев Н. Н., Салимов Ф. И. и др. Анализ стохастических матриц методами многомерной классификации // Дискретная математика и ее приложения: материалы 7-го Междунар. семинара 29 янв. — 2 февр. 2001. — В 3 ч. Ч. II. — М.: МГУ, 2001 . — С. 156–159.

[5] Бухараев Р. Г. О представимости событий в вероятностных автоматах // Учен. зап. Казан. ун-та. Сер. Физ.-матем. науки. — 1967. — Т. 127, кн. 3. — С. 7– 20.

[6] Поспелов Д. А. Вероятностные автоматы. – М.: Энергия. — 1970. — 88 с.

Построение архитектуры нейронной сети, достаточной для приближения всякой кусочно-линейной функции с любой наперед заданной точностью

Шишляков Владимир Геннадьевич

Московский Государственный Университет им. М.В. Ломоносова, Механико-математический факультет, e-mail: bolotmaks@yandex.ru

Основные определения

Базисом нейронной сети называется некоторое конечное число функциональных элементов, где каждый функциональный элемент представляет из себя пару $(S, f(x_1, ..., x_n))$, в которой $f(x_1, ..., x_n) : \mathbb{R}^n \to \mathbb{R}$, а S — сопоставленный ей графический объект с n входными стрелками и одной

выходной (кратко – входы и выход объекта S). Входам объекта S приписаны слева направо переменные $x_1, ..., x_n$ функции f, выходу приписан выход функции $f(x_1, ..., x_n)$.

Далее в тексте графические обозначения объектов опускаются.

Функцию $\psi : \mathbb{R} \to [0; 1]$ будем называть сигмоидной, если ψ не убывает на $\mathbb{R}$ и $\lim_{x\to-\infty} \psi(x) = 0$, $\lim_{x\to+\infty} \psi(x) = 0$.

В теореме 1, сформулированной ниже, рассматривается построение нейронных сетей над видоизмененным базисом:

$$B_1 = \{c, \gamma \cdot x, \sum_n (x_1, ..., x_n), \prod_n (x_1, ..., x_n), \psi(x)\} \tag{1}$$

В базисе (1) используются следующие виды функций:

- Сумматор — это класс функций $\sum_n (x_1, ..., x_n) = x_1 + ... + x_n$
- Продуктор — это класс функций $\prod_n (x_1, ..., x_n) = x_1 \cdot ... \cdot x_n$
- Константа — выдает константу (на входе ничего нет, выдает только заранее определенную константу)
- Умножение на константу — функция умножает пришедший на вход аргумент x на фиксированную константу γ
- Функция активации — это произвольная сигмоидная функция $\psi(x)$

Нейроном в базисе (1) назовем всякую схему, вычисляющую одну из функций $\varphi(\sum_{i=1}^n w_i \cdot x_i + c)$ или $\varphi(\prod_{i=1}^n w_i \cdot x_i + c)$, где функция $\varphi(x)$ полагается равной либо $\psi(x)$, либо x.

Все нейроны первого типа будем называть нейронами-сумматорами, а нейроны второго типа — нейронами-продукторами.

В дальнейшем для удобства мы будем оперировать именно нейронами, а не отдельными элементами базиса нейронной сети. Так, комбинируя нейроны друг с другом, мы можем получать различные функции.

Индуктивно введем понятие слоя нейронной сети.

База индукции. Множество нейронов, все входы которых не подсоединены ни к каким выходам каких-либо функциональных элементов, назовем нейронами первого слоя

Шаг индукции. Пусть определено множество нейронов n-го слоя. Тогда $n + 1$-ым слоем называются все нейроны, у которых хотя бы один вход подсоединен к выходу нейрона n-го слоя, а все оставшиеся входы подсоединены либо к выходам нейронов n-го слоя, либо не подсоединены ни к каким нейронам (тогда считается, что на вход принимаются входные данные).

Определения рассматриваемых в дальнейшем кусочно-постоянных, кусочно-параллельных и кусочно-линейных функций можно найти в [1].

Результаты и выводы

В [1] была доказана теорема о том, что любую кусочно-линейную функцию (не обязательно непрерывную) можно представить в виде схемы функциональных элементов над следующим базисом:

$$B_2 = \{c, \gamma \cdot x, \sum_n (x_1, ..., x_n), \theta(x), F(x, y)\} \tag{2}$$

В выражении (2) полагается, что $F(x, y) = \begin{cases} x, & y \geqslant 0 \\ 0, & y < 0 \end{cases}$.

Однако схемы, построенные над базисом (2), не соотносятся с классической теорией нейронных сетей из-за использования функции $F(x, y)$.

Кроме того, в работе [1] не вводилось понятие слоя нейронной сети, а оценка количества нейронов была нечеткой, т.к. количество нейронов приравнивалось к количеству функций $\theta(x)$.

В данной работе понятие нейрона является более формализованным и более удобным для целей практической реализации, а базис (2) заменен на лучшее приближение (1) к классическому базису, используемому в нейронных сетях прямого распространения.

Очевидно, что в новом базисе (1) задача восстановления кусочно-линейной функции в общем виде невозможна. Поэтому теорема 1 решает задачу аппроксимации кусочно-линейных функций при помощи нейронных сетей над измененным базисом (1).

В целом, подобные задачи уже решались в работе [2] и ряде других работ, таких как [3, 4]. Однако использованные там кусочно-постоянные аппроксимации хорошо применимы для задач классификации и управления, однако плохо совместимы с задачами регрессии.

Теорема 1 является аналогом теоремы Цыбенко [2], но только для нейронных сетей, построенных над видоизмененным базисом, который удобен как для обучения нейронных сетей классическими градиентными методами (при выборе, например, $\psi(x) = \sigma(x) = \frac{1}{1+e^{-x}}$), так и для решения задач регрессии при помощи нейронных сетей. Формулировка теоремы 1 — ниже.

Теорема 1. *Пусть $l_1, ..., l_k$ — гиперплоскости, которые разбивают пространство $\mathbb{R}^n$ на s классов эквивалентности $R^1, ..., R^s$, а $f(x_1, ..., x_n)$ - кусочно-линейная функция, заданная над данными классами эквивалентности.*

Тогда $\forall \varepsilon > 0, \forall \xi > 0, \forall R > 0$ *найдется нейронная сеть* $G(x_1, ..., x_n)$ *над базисом (2) такая, что выполняется* $\sup\limits_{x \in O_R(\overline{0}) \setminus L_\xi} |G(x) - f(x)| < \varepsilon$, *где* $L_\xi = \bigcup\limits_{i=1}^{k} \{(x_1, ..., x_n) \in \mathbb{R}^n || l_i(x_1, ..., x_n)| < \xi\}$.

Причем данная нейронная сеть обладает следующей архитектурой:

1. На первом слое потребуется не более $2k$ *нейронов-сумматоров, имеющих функцию активации* $\varphi(x) = \psi(x)$

2. На втором слое потребуется $2s$ *нейронов, из которых* s *штук имеют функцию активации* $\varphi(x) = \psi(x)$, *а еще* s *штук — тождественную*

3. На третьем слое потребуется s *нейронов-продукторов с тождественной функцией активации*

4. На четвертом слое потребуется один нейрон-сумматор с тождественной функцией активации.

Автор выражает благодарность доценту Часовских А. А. и младшему научному сотрдунику Половникову В. С. за постановку задачи.

Список литературы

[1] Половников В. С. Об оптимизации структурной реализации нейронных сетей : дис. ... канд. физ.-мат. наук : 01.01.09 : защищена 18.04.2008 : утв. 18.03.2008 / Иванов А.О. — Москва, 2008. — 108 с.

[2] Cybenko G. Approximations by superpositions of sigmoidal functions // Math. Control, Signals, Systems. — 1989. — Т. 2, № 5. — С. 303–314.

[3] Funahashi K. On the approximate realization of continuous mappings by neural networks // Neural Networks. — 1989. — Т. 2, № 3. — С. 183–192.

[4] Hecht-Nielsen K. Kolmogorov's Mapping Neural Networks Existence Theorem // IEEE First Annual Int. Conf. on Neural Networks. San Diego — 1987. — Т. 3, № 1. — С. 11–13.

Распознавание униграфов и быстрое вычисление их кликовых чисел

Шкатов Владимир Михайлович

Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского, e-mail: vmshkatov@gmail.com

Здесь и далее используются определения графа, степени вершины, вектора степеней графа, подграфа, соединения, данные в [1]. Все рассматриваемые графы неориентированные.

Вектором степеней графа называется невозрастающая последовательность степеней его вершин. Будем называть граф униграфом, если не существует никакого другого неизоморфного графа с таким же вектором степеней. Существуют эффективный алгоритм ответа на вопрос, является ли заданный граф униграфом.

Кликой графа называется любой полный подграф, содержащийся в данном графе. Независимым множеством графа называется любое множество попарно несмежных вершин графа. Кликовым числом графа называется число вершин в наибольшей клике. Для краткости будем обозначать кликовое число графа G как $clique(G)$. Задача о клике является классической NP-полной задачей [2], поэтому эффективных универсальных алгоритмов для поиска клик заданного размера и кликового числа графа неизвестно. Однако, для униграфов данная задача может быть решена за полиномиальное время.

Распознавание униграфов

Расщепляемым графом называется граф G, множество вершин которого можно разделить на два непересекающихся множества A и B, где вершины из A образуют клику, а вершины из B образуют независимое множество. Известно [3], что для степеней расщепляемых графов $(d_1, d_2, \ldots, d_n)$ (и только для них) выполняется соотношение $\sum\limits_{i=1}^{p} d_i = p(p-1) + \sum\limits_{i=p+1}^{n} d_i$, где p — число вершин в кликовой части. Для произвольных расщепляемых графов, вообще говоря, не исключена ситуация, когда это соотношение выполняется для нескольких p, то есть возможно несколько вариантов расщепления.

Расщепляемой тройкой называется тройка (G, A, B), где $G = (V, \alpha)$ — расщепляемый граф, A — клика, B — независимое множество, $A \cup B = V$ и $A \cap B = \emptyset$. Таким образом, расщепляемая тройка представляет собой расщепляемый граф с фиксированным расщеплением. Будем считать две тройки (G_1, A_1, B_1) и (G_2, A_2, B_2) изоморфными, если существует изоморфизм ϕ графов G_1 и G_2 и при этом $\phi(A_1) = A_2$, $\phi(B_1) = B_2$.

Пусть есть расщепляемая тройка (G, A, B) и произвольный граф H (при этом множества вершин G и H не пересекаются). Тогда композицией $F = (G, A, B) \circ H$ будем называть граф, полученный добавлением в объединение графов $G \cup H$ рёбер между каждой вершиной из A и каждой вершиной из H. Произвольный граф L называется разложимым, если его можно представить в виде подобной композиции, и неразложимым в противном случае.

Теорема о декомпозиции [3]. *Любой граф F можно представить в виде канонического разложения $F = (G_1, A_1, B_1) \circ \ldots \circ (G_k, A_k, B_k) \circ H$, где*

H — неразложимый нерасщепляемый граф, G_i — неразложимые расщепляемые графы. При этом декомпозиция определяет граф с точностью до изоморфизма.

Критерий униграфа [3]. *Граф F является униграфом тогда и только тогда, когда все графы в его каноническом разложении являются униграфами.*

К этой теореме в работе [3] также прилагается описание всех неразложимых униграфов в виде нескольких параметризованных классов. Структура этих графов известна, и по их вектору степеней можно легко определить класс и параметры графа, если он принадлежит к одному из них. Согласно работе [3], декомпозицию и распознание можно провести за линейное время (от длины вектора степеней), т. о. униграфичность при найденном разложении можно проверить за линейное время. В то же время, согласно статье [4], данное каноническое разложение ищется также за полиномиальное время.

Быстрое вычисление кликовых чисел для униграфов

В основе предлагаемой возможности быстрого вычисления кликовых чисел для униграфов лежит следующее предложение.

Предложение (о кликовых числах разложимых графов). *Для композиции $F = (G, A, B) \circ H$, где G неразложим, верно следующее соотношение: $clique(F) = clique(H) + |A|$.*

На основе вышеизложенных теорем и предложения, можно сформулировать следующую теорему-результат.

Теорема о быстром вычислении кликового числа униграфов. *Для униграфов возможно вычисление кликового числа по вектору степеней за полиномиальное время от числа элементов в нём.*

Для этого необходимо за линейное время получить декомпозицию графа (алгоритм описан в [4]), распознать каждый элемент полученной декомпозиции за линейное время и применить формулу из предложения. При этом, поскольку структура нерасщепляемых униграфов известна, кликовое число определяется для них немедленно после определения их класса и параметров.

Автор выражает благодарность своему научному руководителю Абросимову М. Б. за постановку задачи и помощь в оформлении результатов.

Список литературы

[1] Богомолов А. М. Алгебраические основы теории дискретных систем — М. : Наука. Физматлит, 1997. — 368 с.

[2] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи — М. Мир, 1982. — 416 с.

[3] Tyshkevich R. Decomposition of graphical sequences and unigraphs // Discrete Mathematics. — 2000. — Vol. 220. — С. 201–238

[4] Тышкевич Р. И., Суздаль С. В., Декомпозиция графов // Избранные труды Белорусского Государственного Университета. — 2001. — Т. 6. — С. 482–500.

О необходимых условиях бесповторности функций над $\mathbb{Z}_3$

Яшунский Алексей Дмитриевич

ИПМ им. М. В.Келдыша, e-mail: yashunsky@keldysh.ru

Функция F называется *бесповторной* над некоторой системой операций B, если над B существует формула, выражающая F и содержащая каждый из символов переменных не более одного раза. По-видимому, одним из первых бесповторные функции рассматривал А. В. Кузнецов [1]. В дальнейшем вопросы бесповторной выразимости рассматривались как для булевых функций (см. обзор в [2]), так и для функций на конечных множествах, и даже для арифметических функций (см., например, [3, 4]). Указанные работы преимущественно посвящены алгоритмам проверки бесповторности функций, заданных каким-то образом.

Задачи о детерминированных преобразованиях дискретных случайных величин оказываются связанными с бесповторностью функций, а именно — совокупность итеративных преобразований независимых случайных величин операциями из заданной системы B в точности совпадает с классом бесповторных функций над B. В связи с этим некоторые результаты о выразимых распределениях дискретных случайных величин над системой B могут быть интерпретированы как необходимые условия бесповторности функций над указанной системой. В частности, из результатов работы [5] вытекают необходимые условия бесповторности функций k-значной логики над системой $\{+ \pmod k, \times \pmod k\}$ для простых k. Отметим, что методы из [5] не допускают включения в преобразующую систему константных функций. В настоящей работе устанавливаются необходимые условия бесповторности над $\mathbb{Z}_3$, т. е. над системой $\{+ \pmod 3, \times \pmod 3, 1\}$.

Каждая функция $F(x_1, \ldots, x_n)\colon \{0,1,2\}^n \to \{0,1,2\}$ задает набор чисел (p_0, p_1, p_2), где p_i — доля наборов, на которых функция принимает значение $i = 0, 1, 2$. Этот набор далее называем *распределением* функции F. Распределение функции, заданной бесповторной формулой Φ, однозначно определяются распределениями функций, задаваемых главными подформулами формулы Φ. Как следствие, все распределения бесповторных

над $\mathbb{Z}_3$ функций лежат в классе распределений, замкнутом относительно бесповторных сумм и произведений функций $\bmod 3$, а также содержащем распределения тождественной и константных функций.

Каждому распределению (p_0, p_1, p_2) поставим в соответствие комплексное число $\mathbf{p} = p_0 + p_1 e^{2\pi i/3} + p_2 e^{4\pi i/3}$, действительную и мнимую часть которого будем обозначать $\Re\mathbf{p}$ и $\Im\mathbf{p}$ соответственно. Далее будем отождествлять число $\mathbf{p}$ с распределением (p_0, p_1, p_2). Его можно рассматривать как барицентрические координаты в треугольнике, вершины которого — распределения константных функций: $\mathbf{e}^{(0)} = 1$, $\mathbf{e}^{(1)} = e^{2\pi i/3}$, $\mathbf{e}^{(2)} = e^{4\pi i/3}$.

Для функций F и G с непересекающимися наборами переменных и распределениями $\mathbf{p}$ и $\mathbf{q}$ распределения бесповторной суммы $F + G \pmod 3$ и произведения $F \times G \pmod 3$ будем обозначать $\mathbf{p} \mathbin{\widehat{+}_3} \mathbf{q}$ и $\mathbf{p} \mathbin{\widehat{\times}_3} \mathbf{q}$ соответственно. Они удовлетворяют соотношениям:

$$\mathbf{p} \mathbin{\widehat{+}_3} \mathbf{q} = \mathbf{p} \cdot \mathbf{q}, \quad 1 - \Re(\mathbf{p} \mathbin{\widehat{\times}_3} \mathbf{q}) = \frac{2}{3}(1 - \Re\mathbf{p})(1 - \Re\mathbf{q}), \quad \Im(\mathbf{p} \mathbin{\widehat{\times}_3} \mathbf{q}) = \frac{2}{\sqrt{3}} \Im\mathbf{p} \Im\mathbf{q}.$$

Используя эти выражения для операций $\widehat{+}_3$ и $\widehat{\times}_3$, построим множество распределений, которое сохраняется указанными операциями, содержит распределения константных функций и равномерное распределение, и является при этом топологически замкнутым.

Функцию $f(t)$, определенную для действительных $t \in [0; \sqrt{3}/2]$, будем называть *граничной функцией*, если она удовлетворяет условиям $f(t) \leqslant 0$, $f'(t) \leqslant 0$, $f''(t) \leqslant 0$, $f(\sqrt{3}/2) = -1/2$ и $f'(\sqrt{3}/2) \geqslant -1/\sqrt{3}$. По заданной граничной функции f определим множество $\mathcal{A}(f)$ следующим образом:

$$\mathcal{A}(f) = \{e^{2\pi i k/3}(f(t) \pm it) \colon t \in [0; \sqrt{3}/2]; k = -1, 0, 1\}.$$

Пример множества $\mathcal{A}(f)$ приведен на Рис. 1. Часть комплексной плоскости, ограниченную линией $\mathcal{A}(f)$, будем обозначать $\mathbf{A}(f)$.

Имеет место следующее утверждение.

Теорема 1. *Для любой граничной функции f множество $\mathbf{A}(f)$ сохраняется операцией $\widehat{+}_3$, т. е. для всех $\mathbf{p}, \mathbf{q} \in \mathbf{A}(f)$ выполнено $\mathbf{p} \mathbin{\widehat{+}_3} \mathbf{q} \in \mathbf{A}(f)$.*

Специальный выбор граничной функции позволяет доказать замкнутость множества $\mathbf{A}(f)$ и относительно операции $\widehat{\times}_3$.

Теорема 2. *Пусть граничная функция $f(t)$ задана параметрически соотношениями $t = \sqrt{3}(u^3 + 3u - 2)/4$, $f(t) = -(3u^3 - 3u + 1)/4$, где $u \in [u_0; 1]$, а u_0 — корень уравнения $u_0^3 + 3u_0 - 2 = 0$, лежащий на отрезке $[0; 1]$. Тогда множество $\mathbf{A}(f)$ сохраняется операцией $\widehat{\times}_3$, т. е. для всех $\mathbf{p}, \mathbf{q} \in \mathbf{A}(f)$ выполнено $\mathbf{p} \mathbin{\widehat{\times}_3} \mathbf{q} \in \mathbf{A}(f)$.*

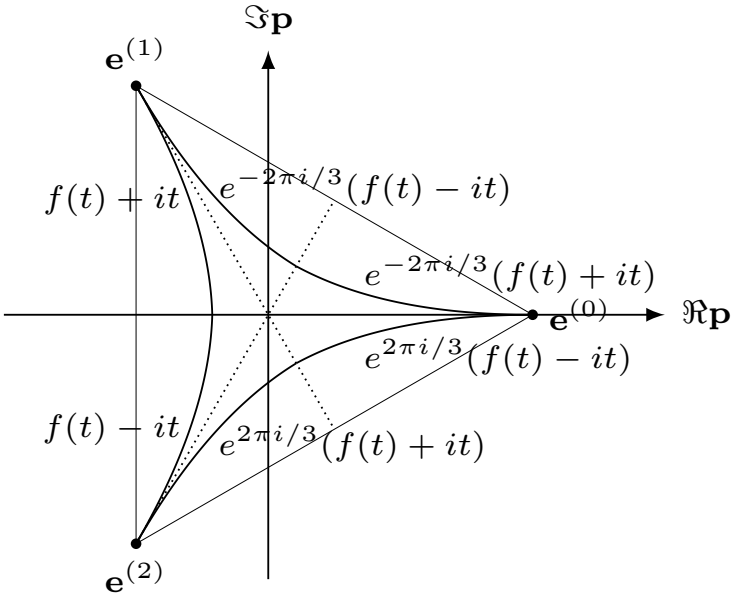


Рис. 1: Линия $\mathcal{A}(f)$, определяемая граничной функцией f.

В действительности множество $\mathbf{A}(f)$, фигурирующее в Теореме 2 и содержащее всевозможные распределения бесповторных над $\mathbb{Z}_3$ функций, допускает более простое описание.

Следствие 1. *Распределение* (p_0, p_1, p_2) *любой бесповторной над* $\mathbb{Z}_3$ *функции удовлетворяет неравенству* $\max p_i - \min p_i \leqslant (\max p_i + \min p_i)^3$.

Список литературы

[1] Кузнецов А. В. О бесповторных контактных схемах и бесповторных суперпозициях функций алгебры логики // Тр. МИАН СССР. — 1958. — Т. 51. — С. 186–225.

[2] Golumbic M. C., Gurvich V. A. Read-once functions // Boolean Functions: Theory, Algorithms and Applications / Crama Y., Hammer P. L. (eds.). — Cambridge: Cambridge University Press, 2011. P. 448–486.

[3] Черемушкин А. В. Бесповторная декомпозиция сильно зависимых функций // Дискретная математика. — 2004. — Т. 16, № 3. — С. 3–42.

[4] Volkovich I. Characterizing arithmetic read-once formulae // ACM Trans. Comput. Theory. — 2016. — V. 8, N. 1. — Art. 2. 19 p. doi:10.1145/2858783

[5] Яшунский А. Д. О бесповторных преобразованиях случайных величин над конечными полями // Дискретная математика. — 2015. — Т. 27, № 3. — С. 145–157.