МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М. В. ЛОМОНОСОВА

МАТЕРИАЛЫ
XX Международной научной конференции

ПРОБЛЕМЫ ТЕОРЕТИЧЕСКОЙ КИБЕРНЕТИКИ

Москва
5-8 декабря 2024 г.

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М. В. Ломоносова

МАТЕРИАЛЫ
XX МЕЖДУНАРОДНОЙ НАУЧНОЙ КОНФЕРЕНЦИИ

ПРОБЛЕМЫ ТЕОРЕТИЧЕСКОЙ КИБЕРНЕТИКИ

Москва
5–8 декабря 2024 г.

МАКС ПРЕСС

МОСКВА – 2025

УДК 519.7
ББК 22.18
П78

https://elibrary.ru/brukip

Редакторы:
С.А. Ложкин, Д.С. Романов, В.В. Подымов

П78 **Проблемы теоретической кибернетики** : материалы XX Международной научной конференции (Москва, 5–8 декабря 2024 г.) / редакторы С.А. Ложкин, Д.С. Романов, В.В. Подымов. – М. : МАКС Пресс, 2025. – 200 с.
ISBN 978-5-317-07402-9
https://doi.org/10.29003/m4678.978-5-317-07402-9

В сборнике представлены труды XX Международной научной конференции «Проблемы теоретической кибернетики» (Москва, 5–8 декабря 2024г.), посвященной 270-летию МГУ имени М. В. Ломоносова и 100-летию со дня рождения чл.-корр. РАН С. В. Яблонского. Тематика конференции включает следующие направления: дискретные функциональные системы, свойства дискретных функций, сложность алгоритмов, синтез, сложность, надёжность, контроль и диагностика управляющих систем, автоматы, теория графов, комбинаторика, теория кодирования, математические методы защиты информации, теория распознавания образов, математическая теория интеллектуальных систем, прикладная математическая логика, приложения дискретной математики и математической кибернетики в естествознании и технике.

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

УДК 519.7
ББК 22.18

Problems of theoretical cybernetics : XX International Scientific Conference (Moscow, December 5–8, 2024) : Proceedings / S.A. Lozhkin, D.S. Romanov, V.V. Podymov (Eds.). – Moscow : MAKS Press, 2025. – 200 p.

The collection represents proceedings of the XX International Scientific Conference “Problems of Theoretical Cybernetics” (Moscow, December 5–8, 2024) dedicated to 270th anniversary of Lomonosov Moscow State University and 100th anniversary of S. V. Yablonsky, corresponding member of Russian Academy of Sciences. The conference subject area includes: discrete functional systems; properties of discrete functions; complexity of algorithms; synthesis, complexity, reliability, control and diagnostics of control systems; automata; graph theory; combinatorics; coding theory; mathematical methods of information security; theory of pattern recognition; mathematical theory of intelligence systems; applied mathematical logic; applications of discrete mathematics and mathematical cybernetics to natural sciences and engineering. For scientists and specialists in areas of mathematical cybernetics, discrete mathematics, computer science an their applications.

ISBN 978-5-317-07402-9

Научное издание

Напечатано с готового оригинал-макета

Подписано в печать 23.05.2025 г. Формат 60x90 1/16. Усл.печ.л. 12,5. Тираж 100 экз. Заказ 069.
Издательство ООО “МАКС Пресс”. Лицензия ИД N 00510 от 01.12.99 г. 119992, ГСП-2, Москва, Ленинские горы, МГУ им. М.В. Ломоносова, 2-й учебный корпус, 527 к. Тел. 8(495)939-3890/91. Тел./Факс 8(495)939-3891.

Отпечатано в полном соответствии с качеством предоставленных материалов в ООО «Фотоэксперт»
109316, г. Москва, Волгоградский проспект, д. 42, корп. 5, эт. 1, пом. I, ком. 6.3-23Н

Содержание

О проблеме существования билинейного алгоритма сложности 17 для перемножения матриц размеров 5×2 и 2×2

Алексеев Валерий Борисович, Назаров Андрей Александрович
Московский государственный университет имени М. В. Ломоносова;
vbalekseev@rambler.ru, nazarovandry2@mail.ru

Введение

Рассматривается задача о сложности умножения матрицы размера $m \times n$ на матрицу размера $n \times p$ (задача $\langle m, n, p \rangle$).

Умножая все строки на все столбцы, получаем, что сложность стандартного алгоритма имеет кубическую зависимость от порядка матрицы (m^3 умножений для матриц размера $m \times m$). В 1969 г. В. Штрассен [1] придумывает алгоритм для умножения матриц 2×2 с 7 умножениями (вместо 8), благодаря чему сложность умножения матриц порядка m понижается с $O(m^3)$ до $O(m^{\log_2 7})$. В 1986 г. Д. Копперсмит и С. Виноград [2] снизили сложность до $O(m^{2.38})$, после чего значительных улучшений не было.

С тем, чтобы лучше понять, как могут быть устроены быстрые алгоритмы для умножения матриц порядка m, изучаются различные вопросы об оптимальных алгоритмах умножения матриц. Данная статья посвящена исследованию билинейной сложности умножения матриц размеров 5×2 и 2×2.

Постановка задачи

Определение. *Билинейный алгоритм* [1, 3] *для задачи умножения матрицы* $||a_{ij}||_{m \times n}$ *на матрицу* $||b_{kh}||_{n \times p}$ *над полем* F *состоит в вычислении* l *выражений вида*

$$D_t = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^t x_{ij} \right) \left(\sum_{k=1}^{n} \sum_{h=1}^{p} b_{kh}^t y_{kh} \right), \quad t = \overline{1, l}, \tag{1}$$

таких, чтобы из них линейными комбинациями (с коэффициентами γ_{ih}^t*) можно было получить все билинейные формы*

$$\sum_{j=1}^{n} x_{ij} y_{jh} = \sum_{t=1}^{l} \gamma_{ih}^t D_t, \quad i = \overline{1, m}, \quad h = \overline{1, p},$$

т. е. все элементы матрицы $X \cdot Y$*. Здесь* x_{ij} *и* y_{kh} *рассматриваются как независимые переменные, а коэффициенты* a_{ij}^t, b_{kh}^t *и* γ_{uw}^t *(*$u = \overline{1, m}$*,* $w = \overline{1, p}$*) берутся из* F*. Число* l *называется сложностью билинейного алгоритма,*

а минимально возможное l по всем билинейным алгоритмам для задачи называется ее билинейной сложностью.

Известно, что билинейная сложность задачи $\langle m, n, p\rangle$ не меняется при любой перестановке чисел m, n и p [4, 5]. Точное значение билинейной сложности задачи $\langle m, n, p\rangle$ известно для очень малого числа параметров. Легко показать, что для задачи $\langle m, 1, p\rangle$ она равна mp. Над произвольным полем доказано только, что для задачи $\langle 2, 2, 2\rangle$ она равна 7 [1, 3], для задачи $\langle 2, 2, 3\rangle$ она равна 11 [6], для задачи $\langle 2, 2, 4\rangle$ она равна 14 [7], для задачи $\langle 2, 3, 3\rangle$ она равна 15 [8].

Итак, $\langle 5, 2, 2\rangle = \langle 2, 2, 5\rangle$. Задачу умножения матрицы размера 2×2 на матрицу размера 2×5 можно рассматривать как два умножения матрицы размера 2×2 на матрицу размера 2×2 и одно умножение матрицы размера 2×2 на матрицу размера 2×1. Поскольку для перемножения двух матриц размера 2×2 алгоритм Штрасена (он билинейный) имеет билинейную сложность 7, а умножение матрицы размера 2×2 на матрицу размера 2×1 стандартным алгоритмом имеет билинейную сложность 4, то для задачи $\langle 2, 2, 5\rangle$ получаем билинейный алгоритм сложности 18. С другой стороны, в работе [9] доказано, что для задачи $\langle 2, 2, 5\rangle$ не существует билинейного алгоритма сложности менее 17. Возникает вопрос: существует ли билинейный алгоритм для $\langle 2, 2, 5\rangle$ сложности 17?

Формулировка задачи в другом виде

Из (1) следует, что каждое D_t — билинейная форма от двух множеств переменных $\{x_{ij}\}$ и $\{y_{kh}\}$, причем эта форма разложима в произведение двух линейных форм: первая от переменных $\{x_{ij}\}$, вторая от переменных $\{y_{kh}\}$. Нетрудно доказывается следующее утверждение.

Утверждение. *Ненулевая билинейная форма $\sum_{ik} x_i y_k$ представима в форме $(\sum_i x_i) \cdot (\sum_k y_k)$ тогда и только тогда, когда матрица этой билинейной формы имеет ранг 1.*

Определим пространство матриц $M_0 = M_{mn \times np} = M_{n \times n}(K)$, где блоки K являются матрицами из $M_{m \times p}$. Заметим, что билинейный алгоритм задается коэфициентами $a^t_{ij}, b^t_{kh}, \gamma^t_{uw}$ и только ими. Тогда вопрос о билинейной сложности задачи $\langle m, n, p\rangle$ равносилен следующей задаче.

Задача. *Найти минимально возможное число l таких матриц ранга 1 из M_0, что их линейными комбинациями можно получить mp всевозможных матриц из M_0, таких что все их блоки нулевые, кроме блоков главной диагонали, которые все одинаковы и имеют следующий вид: все элементы нулевые, кроме одного единичного.*

В случае $\langle m, n, p\rangle = \langle 2, 2, 5\rangle$ матрицы, которые нужно получить в результате линейных комюбинаций матриц ранга 1, будут выглядеть так:

$$\left(\begin{array}{ccccc|ccccc} 1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0 \end{array}\right), \ldots, \left(\begin{array}{ccccc|ccccc} 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ \hline 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1 \end{array}\right).$$

Таким образом, поиск билинейного алгоритма сложности l для задачи $\langle 2, 2, 5\rangle$ равносилен поиску l матриц $\{X_1, \ldots X_l\}$ ранга 1, линейными комбинациями которых можно получить все 10 матриц, указанных выше. Как уже отмечено, для этой задачи существует решение с $l = 18$ и не существует с $l \leqslant 16$. Мы исследуем вопрос о существовании решения с $l = 17$. Обозначим проекции решения на правый верхний блок за $\varphi_{12}(X_i)$, и пусть d — размерность линейного подпространства в 10-мерном пространстве матриц размера 2×5, порожденного матрицами $\varphi_{12}(X_i)$. Доказано следующее утверждение.

Теорема. *При $d \leqslant 4$ и $d \geqslant 8$ решения $\langle 2, 2, 5\rangle$ с $l = 17$ не существует.*

Остается исследовать $d = 5, 6, 7$. Случай $d = 5$ исследован частично.

Список литературы

[1] Штрассен В. Алгоритм Гаусса не оптимален // Кибернетический сборник. Т. 7. М. : Мир, 1970. С. 67–70.

[2] Coppersmith D., Winograd S. Matrix multiplication via arithmetic progressions // Journal of Symbolic Computation. 1990. Vol. 9. P. 251–280.

[3] Winograd S. On multiplication of 2×2 matrices // Linear Algebra and its Applications. 1971. Vol. 4. P. 381–388.

[4] Hopcroft J. E., Musinski J. Duality applied to the complexity of matrix multiplication and other bilinear forms // SIAM Journal on Computing. 1973. Vol. 2, no. 3. P. 159–173.

[5] Алексеев В. Б. Сложность умножения матриц. Обзор // Кибернетический сборник. Новая серия. Вып. 25. М. : Мир, 1988. С. 189–236.

[6] Alekseyev V. B. On the complexity of some algorithms of matrix multiplication // Journal of Algorithms. 1985. Vol. 6, no. 1. P. 71–85.

[7] Алексеев В. Б., Смирнов А. В. О точной и приближенной билинейных сложностях умножения матриц размеров 4×2 и 2×2 // Современные проблемы математики. 2013. Т. 17. С. 6–23.

[8] Бориченко В. П. О билинейной сложности умножения 3×2 матрицы на 2×3 матрицу // Дискретная математика. 2024. Т. 36, вып. 1. С 15–45.

[9] Алексеев В. Б. О билинейной сложности умножения матриц размеров 5×2 и 2×2 // Ученые записки Казанского университета. Серия Физико-математические науки. 2014. Т. 156, кн. 3. С. 19–29.

Новый подход к оценке сумм граничных функционалов для случая квазирегулярных структур

Андреева Татьяна Владимировна
Московский государственный университет имени М. В. Ломоносова; andreevatv@cs.msu.ru

В работах А. А. Сапоженко (см., например, [1]) для получения нижней оценки числа антицепей в ранжированных частично упорядоченных множествах применяется метод граничных функционалов, суть которого заключается в сведении исходной задачи к вычислению сумм граничных функционалов по связным множествам малой мощности. А. А. Сапоженко ввел понятие «ординарность» как совокупность условий, выполнение которых позволяет получать оценки этих сумм. Такой подход оказался эффективным только для случая частично упорядоченных множеств, имеющих регулярную структуру. В случае множеств с нерегулярной структурой оценки оказываются слишком грубыми, а необходимые вычисления — слишком громоздкими.

В настоящей работе предложено понятие «квазирегулярность», которое в некоторых случаях позволяет получать более точные оценки сумм граничных функционалов с помощью менее громоздких выкладок.

Рассмотрим применение метода граничных функционалов на примере задачи о числе независимых множеств в двудольных графах.

Пусть $\Gamma = (X, Z; E)$ — двудольный граф с долями вершин X, Z и множеством ребер E. Границей множества $A \subseteq X$ называется множество

$$\partial(A) = \big\{v \in Z : \ \exists\, u \in A \ \ \{u, v\} \in E\big\}.$$

Рассмотрим граф $G_\Gamma = (X; E_\Gamma)$, в котором $E_\Gamma = \big\{\{u, v\} : \partial\{v\} \cap \partial\{u\} \neq \varnothing\big\}$. Множество $A \subseteq X$ связно (в Γ), если связен подграф графа G_Γ, порожденный множеством A.

Обозначим через $\Phi(\Gamma)$ число независимых множеств в графе Γ, тогда

$$\Phi(\Gamma) = 2^{|Z|} \sum_{A \subseteq X} 2^{-|\partial(A)|}.$$

Функционал $f(A) = 2^{-|\partial(A)|}$ называется граничным функционалом.

Обозначим через $\mathcal{A}^{(1)}(\Gamma) = \mathcal{A}(\Gamma)$ семейство всех связных подмножеств множества X. Для $\mathcal{B} \subseteq \mathcal{A}(\Gamma)$ положим $\mathcal{B} = \mathcal{B}^{(1)}$.

Пусть теперь $s \geqslant 2$, $A_1, \ldots, A_s$ — различные непустые подмножества множества X. Семейство $F = \{A_1, \ldots, A_s\}$ называется связным семейством ранга 2 над X, если $A_i \in \mathcal{A}(\Gamma)$, $i = 1, \ldots, s$, и $A_1 \cup \ldots \cup A_s \in \mathcal{A}(\Gamma)$. Положим

$$f(F) = \prod_{i=1}^{s} f(A_i).$$

Множество всех связных семейств ранга 2 обозначается через $\mathcal{A}^{(2)}(\Gamma)$. Для произвольного $\mathcal{B} \subseteq \mathcal{A}(\Gamma)$ определим семейство $\mathcal{B}^{(2)} = 2^{\mathcal{B}} \cap \mathcal{A}^{(2)}(\Gamma)$.

Пусть $r \in \{1, 2\}$, ν — натуральное число. Положим

$$\alpha^{\nu}(\mathcal{B}^{(r)}) = \sum_{A \in \mathcal{B}^{(r)}} \big(f(A)\big)^{\nu}.$$

Существенной частью метода граничных функционалов является выражение $\Phi(\Gamma)$ через суммы типа $\alpha^{\nu}(\mathcal{B}^{(r)})$ по связным множествам малой мощности.

Граф $\Gamma = (X, Z; E)$ назовем (κ, p, q, t)-квазирегулярным, если выполнены следующие условия:

1) $\kappa \leqslant |\partial\{v\}| \leqslant p\kappa$ для всякого $v \in X$;

2) $\big|\big\{v \in X : \{v, w\} \in E\big\}\big| \leqslant p\kappa$ для всякого $w \in Z$;

3) $|\partial\{u\} \cap \partial\{v\}| \leqslant q$ для любых $u, v \subseteq X$;

4) $\big||\partial\{u\}| - |\partial\{v\}|\big| \leqslant t$ для всех $u, v \in X$ таких, что $\partial\{u\} \cap \partial\{v\} \neq \varnothing$.

Везде в дальнейшем κ, q, t — натуральные числа, $p \geqslant 1$.

Для графа $\Gamma = (X, Z; E)$ и $\mathcal{B} \subseteq \mathcal{A}(\Gamma)$ положим

$$\mathcal{B}_{[j]} = \big\{A \in \mathcal{B} : |A| = j\big\}, \qquad \mathcal{A}_{\widehat{m}}(\Gamma) = \bigcup_{j=1}^{m} \mathcal{A}_{[j]}(\Gamma),$$

$$\mathcal{B}^{(2)}_{[j]} = \big\{F = \{A_1, \ldots, A_s\} \in \mathcal{B}^{(2)} : |A_1| + \ldots + |A_s| = j\big\}.$$

Утверждение 1. *Для любого (κ, p, q, t)-квазирегулярного двудольного графа Γ и любых $\mathcal{B} \subseteq \mathcal{A}(\Gamma)$ и $j, \nu \in \mathbb{N}$ справедливо*

$$\alpha^{\nu}(\mathcal{B}_{[j]}) \leqslant \frac{(p\kappa)^{2(j-1)}}{4j} 2^{j(\nu(q+t)(j-1)/2+2)} \cdot \alpha^{j\nu}\big(\mathcal{B}_{[1]}\big).$$

Утверждение 2. *Для любого (κ, p, q, t)-квазирегулярного двудольного графа Γ и любых $m, j \in \mathbb{N}$ и $\mathcal{B} \subseteq \mathcal{A}_{\widehat{m}}(\Gamma)$ справедливо*

$$\alpha^{1}(\mathcal{B}^{(2)}_{[j]}) \leqslant \frac{(p\kappa)^{2(j-1)}}{4j} 2^{j(q(m-1)/2+t(j-1)+5)} \cdot \alpha^{j}(\mathcal{B}_{[1]}).$$

Пусть S — ранжированное частично упорядоченное множество. Обозначим через $\Psi(S)$ число антицепей в S. Пусть S_n — n-й слой множества S, i^* — номер слоя, имеющего максимальную мощность. Рассмотрим частично упорядоченное множество $P = S_{i^*-1} \cup S_{i^*}$. Определим граф $\Gamma(S) = (S_{i^*-1}, S_{i^*}; E)$, в котором $E = \big\{\{\tilde{\beta}, \tilde{\gamma}\} \colon \tilde{\beta} \in S_{i^*-1},\ \tilde{\gamma} \in S_{i^*},\ \tilde{\beta} \leqslant \tilde{\gamma}\big\}$. Тогда

$$\Psi(S) \geqslant \Psi(P) = \Phi\big(\Gamma(S)\big).$$

Положим $\mathcal{B} = \mathcal{A}_{\widehat{2}}\big(\Gamma(S)\big)$. Если граф $\Gamma(S)$ квазирегулярный, то для оценки сумм граничных функционалов можно применить утверждения 1 и 2. Суммы типа $\alpha^{\nu}(\mathcal{B}_{[1]})$ могут быть вычислены как коэффициенты некоторых производящих функций.

Пример 1. При $b, c \in \mathbb{N}$ пусть $E_{b,c} = \{e_1^-, \ldots, e_c^-, n_1, \ldots, n_b, e_1^+, \ldots, e_c^+\}$. Элементы e_i^- будем называть отрицательными, элементы n_j — нейтральными, элементы e_k^+ — положительными. На множестве $E_{b,c}$ введем отношение порядка: $e_i^- < n_j < e_k^+$ при любых $i, j \in \{1, \ldots, c\}$, $k \in \{1, \ldots, b\}$, элементы одного знака будем считать несравнимыми.

На множестве $E_{b,c}^n$ определим функцию ранга $r(\tilde{\beta}) = \sum_{i=1}^n \operatorname{sgn}(\beta_i)$. Через $F_{b,c}(n, m)$ обозначим m-й слой множества $E_{b,c}^n$, $-n \leqslant m \leqslant n$. При этом $i^* = 0$.

Положим $X = F_{b,c}(n, -1)$, $Z = F_{b,c}(n, 0)$. Граф $\Gamma(E_{b,c}^n)$ является (κ, p, q, t)-квазирегулярным при $\kappa = \lfloor n \cdot \min(b/2, c) \rfloor$, $p = \max\left(\frac{b}{2c}, \frac{2c}{b}\right)$ и $q = \max(b, c)$, $t = |b - 2c|$.

Рассмотрим производящую функцию

$$G_{b,c}(z) = (2^{-\nu b}c + 2^{-\nu c}bz + cz^2)^n = \sum_{r=0}^{2n} G_{b,c}^{(r)} z^r.$$

Можно показать, что $\alpha^{\nu}(\mathcal{B}_{[1]}) = G_{b,c}^{(n-1)}$.

Для вычисления $G_{b,c}^{(n-1)}$ можно воспользоваться результатами работы [2].

Пример 2. Пусть $k \in \mathbb{N}$, рассмотрим множество $E_k = \{0, 1, \ldots, k-1\}$ с отношением порядка $0 < 1 < \cdots < k - 1$.

На множестве E_k^n функцию ранга определим как $r(\tilde{\alpha}) = \sum_{j=1}^n \alpha_j$. Через $F(n, r, k)$ обозначим r-й слой множества E_k^n, $0 \leqslant r \leqslant n(k-1)$. При этом $i^* = \left\lfloor \frac{n(k-1)}{2} \right\rfloor$.

Положим $X = F\left(n, \left\lfloor \frac{n(k-1)}{2} \right\rfloor - 1, k\right)$, $Z = F\left(n, \left\lfloor \frac{n(k-1)}{2} \right\rfloor, k\right)$. Граф $\Gamma(E_k^n)$ является $(\lceil n/2 \rceil, 2, 1, 1)$-квазирегулярным.

Рассмотрим производящую функцию

$$G_k(z) = (2^{-\nu} + 2^{-\nu}z + \cdots + 2^{-\nu}z^{k-2} + z^{k-1})^n = \sum_{r=0}^{(k-1)n} G_k^{(r)} z^r.$$

Можно показать, что $\alpha^{\nu}(\mathcal{B}_{[1]}) = G_k^{(\lfloor (k-1)n/2 \rfloor - 1)}$.

Список литературы

[1] Сапоженко А. А. Проблема Дедекинда и метод граничных функционалов. М. : Физматлит, 2009. 152 с.

[2] Андреева Т. В. Асимптотика значений коэффициентов некоторых производящих функций // Материалы XIV Международного семинара «Дискретная математика и ее приложения» имени академика О. Б. Лупанова

(Москва, МГУ, 20–25 июня 2022 г.). М. : ИПМ им. М. В. Келдыша, 2022. С. 145–147.

Об энтропийном сжатии видео

Бабин Дмитрий Николаевич
Московский государственный университет имени М. В. Ломоносова,
механико-математический факультет; d.n.babin@mail.ru

Основным алгоритмом сжатия фильмов является поиск одинаковых с предыдущими кадрами прямоугольников на новых кадрах фильма и запоминание ссылок на них. При этом объём этих ссылок может оказаться довольно большим. Идея предлагаемого метода заключается в небольшом числе ссылок, но не на прямоугольники, а на произвольные множества пикселов.

Предполагается, что кадр K_s в нашем распоряжении имеется, например, получен раньше. Пусть кадр K_t фильма — это матрица размера $m \times n$. В другом кадре K_s, сдвинутом на вектор (I, J) относительно кадра K_t, найдём совпадающие (с погрешностью ε) элементы. Среди всевозможных сдвигов (I, J) выберем несколько тех, у которых указанных элементов достаточно много.

При сжатии кадра K_t мы укажем номер сдвига (I, J) вместо самого пикселя, а для пикселей, которых не нашлось в кадре K_s, укажем номер кластера таких пикселей. Число кластеров является параметром алгоритма. Проверка показала, что эффективность метода лучше, чем у сжатия методом JPEG, и в отдельных случаях близка в кодеку H264. Эффективность сжатия измерялась как отношение сигнала к шуму ($PSNR$). Этот подход был обобщён на блоки 2×2 или 3×3 пикселей кадра, при этом эффективность сжатия увеличилась.

Интересный эффект даёт использование в качестве блоков строк матрицы кадра для простых черно-белых изображений. Здесь возможно точное сжатие, которое плохо достигается кодеками H264–H266. Метод был опробован на базе изображений DAVIS, https://davischallenge.org/. В среднем получилось точно сжать некоторые фильмы более чем в два раза лучше, чем кодеком H264.

Список литературы

[1] Яблонский С. В. Введение в дискретную математику. Учебное пособие для вузов. М. : Высшая школа, 2010. 384 с.

[2] Бабин Д. Н., Пархоменко Д. В., Кириллов И. В. Об одном методе сжатия некоторых фильмов // Тезисы докладов международной конференции «Математика в созвездии наук». М. : Издательство Московского университета, 2024. 306 с.

[3] Hierarchical B-frame video coding for long group of pictures / I. Kirillov, D. Parkhomenko, K. Chernyshev, A. Pletnev, Yibo Shi, Kai Lin, D. Babin // arXiv preprint arXiv:2406.16544. 2024. (available at https://arxiv.org/abs/2406.16544).

Новый компромисс между временем работы и количеством используемой памяти алгоритмов k-просеивания для решения задачи нахождения кратчайшего вектора в решётке

Бахарев Александр Олегович
Новосибирский государственный университет; a.bakharev@g.nsu.ru

Задача поиска кратчайшего вектора в решётке (SVP) заключается в поиске кратчайшего ненулевого вектора в решётке и является одной из основных задач, к которой сводится стойкость большинства криптосистем, построенных на решётках. Например, вариации задач обучения с ошибками (LWE) и нахождения короткого целочисленного решения (SIS), на сложности решения которых основываются современные схемы инкапсуляции ключей и подписи, могут быть представлены как аппроксимационный вариант задачи SVP или сводиться к ней. Одними из основных алгоритмов, решающих SVP, являются семейство алгоритмов просеивания [1–4]. Алгоритмы просеивания имеют экспоненциальное время работы и используют экспоненциальное количество памяти в зависимости от размерности решётки. В настоящей работе предложен новый компромисс между временем работы и используемой памятью алгоритма 8-просеивания для решения задачи SVP.

Решётки. Термин «решётка» появляется в различных областях математики, например, алгебра, геометрия, теория графов и другие. Мы будем рассматривать следующее определение. Пусть векторы $\vec{v}_1, \ldots, \vec{v}_n \in \mathbb{R}^d$ линейно независимы. *Решёткой, порождённой векторами* $\vec{v}_1, \ldots, \vec{v}_n$, называется набор линейных целочисленных комбинаций векторов $\vec{v}_1, \ldots, \vec{v}_n$,

$$\mathcal{L} = \{a_1\vec{v}_1 + a_2\vec{v}_2 + \ldots + a_n\vec{v}_n : a_1, a_2, \ldots, a_n \in \mathbb{Z}\}.$$

Будем называть число n *рангом решётки*, а d — *размерностью решётки*. В случае, когда $n = d$, решётка $\mathcal{L}$ называется *решёткой полного ранга*. В настоящей работе рассматриваются решётки полного ранга. *Базисом* для $\mathcal{L}$ является любой линейно независимый набор векторов, порождающий $\mathcal{L}$. Базисные векторы могут быть представлены в виде базисной матрицы $\mathbf{B} = [\vec{v}_1|\vec{v}_2|\ldots|\vec{v}_n]$. Любые два базиса $\mathcal{L}$ связаны преобразованием с целочисленной матрицей, определитель которой равен ± 1 (унимодулярная матрица).

Минимальным расстоянием λ_1 называется норма кратчайшего ненулевого вектора в решетке $\mathcal{L}$, то есть $\lambda_1(\mathcal{L}) = \min_{\vec{x}\in\mathcal{L}\setminus\{\mathbf{0}\}} \|\vec{x}\|$. Определим *задачу поиска кратчайшего вектора (Shortest Vector Problem, SVP).* Пусть дан базис $\mathbf{B}$, который задает решетку $\mathcal{L}$, требуется найти ненулевой вектор $\vec{v} \in \mathcal{L}$ такой, что $\|\vec{v}\| = \lambda_1(\mathcal{L})$. SVP является NP-трудной задачей [5].

Алгоритмы просеивания. Основная идея k-просеивания состоит в нахождении в заданном списке L кортежей $(\vec{x}_1, \ldots, \vec{x}_k) \in L^k$, удовлетворяющих условию $\|\vec{x}_1 \pm \cdots \pm \vec{x}_k\| \leqslant 1$, в отличие от рассматриваемых ранее алгоритмов, в которых искались пары $(\vec{x}_1, \vec{x}_2) \in L^2$, удовлетворяющие условию $\|\vec{x}_1 \pm \vec{x}_2\| \leqslant 1$ (условие $\|\vec{x}_1 \pm \vec{x}_2\| \leqslant 1$ означает, что $\|\vec{x}_1 + \vec{x}_2\| \leqslant 1$ или $\|\vec{x}_1 - \vec{x}_2\| \leqslant 1$, для $k > 2$ аналогично). Задачу нахождения всех таких кортежей будем называть *задачей k-просеивания.* Данное расширение на $k > 2$ позволяет уменьшить количество используемой памяти алгоритмом, т. е. входного списка L, но вместе с этим увеличивается время работы. Данный компромисс между используемой памятью и временем работы алгоритма может быть полезен при вычислении кратчайшего времени в решётке.

Будем говорить, что кортеж $(\vec{x}_1, \ldots, \vec{x}_k) \in L^k$ *удовлетворяет конфигурации* $C \in \mathbb{R}^{k\times k}$, если и только если $\forall\, i, j\ \langle\vec{x}_i|\vec{x}_j\rangle \leqslant C_{ij}$, где $\langle\vec{x}_i|\vec{x}_j\rangle$ — скалярное произведение векторов $\vec{x}_i$ и $\vec{x}_j$. В [3] показано, что задача k-просеивания сводится к задаче поиска кортежа $(\vec{x}_1, \ldots, \vec{x}_k) \in L^k$, удовлетворяющего некоторой конфигурации. Для конфигурации $C \in \mathbb{R}^k$ и угла α введём обозначения

$$C'_{ij}(\alpha) = \frac{1}{\sin^2\alpha}\left(C_{ij} + \frac{\cos^2\alpha}{k-1}\right) \quad \text{и} \quad \mathcal{V}(\alpha) = poly(d)\cdot\sin^n\alpha.$$

Также для улучшения временных характеристик алгоритма используются локально-чувствительные фильтры [1, 2]. В [2] авторами была представлена новая концепция алгоритма k-просеивания, на основе которой были построены алгоритмы 3- и 4-просеивания, предлагающие новый компромисс между временем работы и используемой памятью алгоритмов.

8-просеивание. Используя концепцию из [2], в настоящей работе предложен новый алгоритм 8-просеивания. Для данного алгоритма получены выражения для времени работы и количества используемой памяти.

Теорема. *Пусть* $\mathcal{V}(\alpha) = \frac{1}{|L_1|}$, $Y = \frac{1}{\sin^2(\alpha)\cdot(8+12C_{12}+8C_{23}+4C_{34})} - 1$, $Y_{23} = \frac{C_{23}-C_{12}^2}{(1-C_{12}^2)}$, $Y_{234} = \frac{C_{34}-C_{12}^2-(1-C_{12}^2)Y_{23}}{1-C_{12}^2-(1-C_{12}^2)Y_{23}^2}$. *Тогда время выполнения предложенного алгоритма равно* $T = 6T_{12} + 4T_{123} + 2T_{1234} + T_{1\ldots8}$, *где*

$$T_{12} = \mathcal{O}\left(|L_1|^2\frac{(1-C_{12}^2)^{n/2}}{(1-C'_{12}(\alpha)^2)^{n/2}}\right), \qquad L_{12} = |L_1|^2(1-C_{12}^2)^{n/2},$$

$$T_{123} = \mathcal{O}\left(|L_1||L_2(\vec{x}_1)|^2\frac{(1-Y_{23}^2)^{n/2}}{(1-Y_{23}'^2)^{n/2}}\right), \qquad L_{123} = |L_1||L_2(\vec{x}_1)|^2(1-Y_{23}^2)^{n/2},$$

$$T_{1234} = \mathcal{O}\left(|L_1||L_2(\vec{x}_1)||L_3(\vec{x}_1,\vec{x}_2)|^2\frac{(1-Y_{234}^2)^{n/2}}{(1-Y_{234}'(\alpha)^2)^{n/2}}\right),$$

$$L_{1234} = |L_1||L_2(\vec{x}_1)||L_3(\vec{x}_1,\vec{x}_2)|^2(1-Y_{234}^2)^{n/2},$$

$$T_{1\ldots8} = \mathcal{O}\left(|L_{1234}|^2\frac{(1-Y^2)^{n/2}}{(1-Y'(\alpha)^2)^{n/2}}\right), \qquad L_{1\ldots8} = |L_{1234}|^2(1-Y^2)^{n/2}.$$

Количество используемой памяти равно

$$M = \max\{|L_1|, |L_{12}|, |L_{123}|, |L_{1234}|, |L_{1\ldots8}|\}.$$

Для получения численных значений времени работы и количества используемой памяти предложенного алгоритма 8-просеивания написана программа на языке Sage. Отметим, что при количестве используемой памяти равном $2^{0.2075n}$ предлагаемый алгоритм совпадает по времени работы и используемой памяти с алгоритмом из [1], который является оптимальным по времени работы алгоритмом на сегодняшний день. Для количества используемой памяти, близкого к минимальным значениям, предлагаемый алгоритм работает дольше, что совпадает с поведением алгоритма 4-просеивания из [2]. На отрезке $(2^{0.157n}, 2^{0.189n})$ используемой памяти предложенный алгоритм показывает минимальное время работы для известных алгоритмов k-просеивания.

Работа выполнена при поддержке Математического Центра в Академгородке, соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2022-282.

Список литературы

[1] Becker A., Ducas L., Gama G., Laarhoven T. New directions in nearest neighbor searching with applications to lattice sieving // Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA, USA, Jan. 10–12, 2016). Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 2016. P. 10–24.

[2] Chailloux A., Loyer J. Classical and quantum 3 and 4-sieves to solve SVP with low memory // Post-Quantum Cryptography. PQCrypto 2023. Lecture Notes in Computer Science. 2023. Vol. 14154. P. 225–255.

[3] Herold G., Kirshanova E. Improved algorithms for the approximate k-list problem in Euclidean norm // Public-Key Cryptography — PKC 2017. Lecture Notes in Computer Science. 2017. Vol. 10174. P. 16–40.

[4] Herold G., Kirshanova E., Laarhoven T. Speed-ups and time–memory tradeoffs for tuple lattice sieving // Public-Key Cryptography — PKC 2018. Lecture Notes in Computer Science. 2018. Vol. 10769. P. 407–436.

[5] Ajtai M. The shortest vector problem in L_2 is NP-hard for randomized reductions // STOC '98: Proceedings of the thirtieth annual ACM symposium

on Theory of computing. New York, NY, USA : Association for Computing Machinery, 1998. P. 10–19.

Оценка длин тестов в базисе Жегалкина при константных неисправностях типа «1» на выходах элементов

Бородина Юлия Владиславовна
Институт прикладной математики имени М. В. Келдыша РАН; jborodina@inbox.ru

Будем рассматривать схемы из функциональных элементов в базисе Жегалкина $B = \{\oplus, \&, 0, 1\}$. В качестве неисправностей предполагаем константные неисправности типа «1» на выходах конъюнкторов и сумматоров.

Пусть S — некоторая схема из функциональных элементов, реализующая булеву функцию $f(\tilde{x})$, $\tilde{x} = (x_1, x_2, \ldots, x_n)$, в базисе B.

Функция, реализуемая на выходе схемы при наличии в последней неисправных элементов, называется *функцией неисправности*. Всякое множество T входных наборов схемы S называется *полным проверяющим тестом* для этой схемы, если для любой функции неисправности $g(\tilde{x})$, не равной тождественно $f(\tilde{x})$, в T найдется хотя бы один такой набор $\tilde{\sigma}$, что $f(\tilde{\sigma}) \neq g(\tilde{\sigma})$. Число наборов, составляющих этот тест, называется *длиной* теста. Введем обозначения: $D(f)$ — минимум длин проверяющих тестов по всем схемам S, реализующим функцию f; $D(n) = \max D(f)$, где максимум берется по всем булевым функциям f от n переменных.

В работе [1] было доказано, что в случае константных неисправностей типа «0» на выходах элементов всякую булеву функцию можно реализовать схемой из функциональных элементов в базисе B, допускающей полный проверяющий тест длины 1.

В случае константных неисправностей типа «1» такого рода результат невозможен. Именно, в [2] был описан достаточно узкий класс булевых функций f, для которых $D(f) = 1$. В [3] выделены некоторые классы функций, допускающих легкотестируемые схемы.

В докладе представлена оценка $D(f)$ для функций f, у которых многочлен Жегалкина имеет ограниченную степень. При этом удается улучшить известные оценки $D(n)$ при малых n.

Теорема 1. *Справедливы следующие оценки функций Шеннона длин полных проверяющих тестов для схем из функциональных элементов в базисе Жегалкина при константных неисправностях типа* «1» *на выходах элементов для классов булевых функций:*

1) $D(2) = 2, D(3) \leqslant 4, D(4) \leqslant 8, D(5) \leqslant 16, D(6) \leqslant 32$*;*

2) для функций f от n переменных, представимых линейным многочленом Жегалкина, $D(f) \leqslant 1$;

3) для функций f от n переменных, у которых многочлен Жегалкина имеет степень не выше 2, $D(f) \leqslant 2n - 2$;

4) для функций f от n переменных, у которых многочлен Жегалкина имеет степень не выше 3, $D(f) \leqslant n^2 - 3n + 4$;

5) для функций f от $n \geqslant k$ переменных, у которых многочлен Жегалкина имеет степень не выше k, $D(f) \leqslant \frac{n^{k-1}}{(k-2)!} + 1$.

В последнее время получен ряд оценок функции Шеннона длин тестов для схем в базисе Жегалкина в предположении наличия только одного неисправного элемента. В [4] найдено точное значение 1 функции Шеннона длины единичного проверяющего теста при константных неисправностях типа «1» на выходах элементов. Д. С. Романов [5] нашёл точное значение 1 функции Шеннона длины единичного диагностического теста при инверсных неисправностях на выходах элементов, Д. С. Романов и Е. Ю. Романова [6] получили верхнюю оценку 16 функции Шеннона длины единичного проверяющего теста при произвольных константных неисправностях на входах и выходах элементов, К. А. Попков — для функции Шеннона длины единичного диагностического теста точное значение 2 (при $n \geqslant 2$) при однотипных константных неисправностях типа «0» на выходах элементов [7] и верхнюю оценку 3 при однотипных константных неисправностях типа «1» на выходах элементов и при инверсных неисправностях на входах и выходах элементов [8] .

Список литературы

[1] Бородина Ю. В., Бородин П. А. Синтез легкотестируемых схем в базисе Жегалкина при константных неисправностях типа “0” на выходах элементов // Дискретная математика. 2010. Т. 22, № 3. С. 127–133.

[2] Бородина Ю. В. Легкотестируемые схемы в базисе Жегалкина при константных неисправностях типа “1” на выходах элементов // Дискретная математика. 2019. Т. 31, № 2. С. 14–19.

[3] Бородина Ю. В. Некоторые классы легкотестируемых схем в базисе Жегалкина // Дискретная математика. 2021. Т. 33, № 4. С. 3–10.

[4] Бородина Ю. В. О схемах, допускающих единичные тесты длины 1 при константных неисправностях на выходах элементов // Вестник Московского университета. Серия 1. Математика. Механика. 2008. № 5. С. 49–52.

[5] Романов Д. С. Метод синтеза неизбыточных схем в базисе Жегалкина, допускающих единичные диагностические тесты длины один // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2015. № 4 (36). С. 38–54.

[6] Романов Д. С., Романова Е. Ю. Метод синтеза неизбыточных схем, допускающих единичные проверяющие тесты константной длины // Дискретная математика. 2017. Т. 29, № 4. С. 87–105.

[7] Попков К. А. О единичных диагностических тестах для схем из функциональных элементов в базисе Жегалкина // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2016. № 3 (39). С. 3–18.

[8] Попков К. А. Метод построения легко диагностируемых схем из функциональных элементов относительно единичных неисправностей // Прикладная дискретная математика. 2019. Т. 46. С. 38–57.

Сложность нахождения представлений булевых функций в классе расширенных полиномиальных форм

Винокуров Сергей Федорович
Иркутский государственный университет; servin38@gmail.com

В работе [1] предложен алгоритм перехода между операторными полиномиальными формами булевых функций в классе двупорожденных форм, а также найдена верхняя граница его сложности. Дальнейшие исследования позволили модифицировать этот алгоритм до применения в более широких классах форм, не являющихся двупорожденными.

Как и в указанной работе, далее в изложении предполагается, что булевы функции n аргументов являются векторами линейного векторного пространства F_n размерности 2^n.

Пусть заданы базисы

$$G = \{g_1(x_1, \ldots, x_n), \ldots, g_{2^n}(x_1, \ldots, x_n)\} \text{ и}$$

$$H = \{h_1(x_1, \ldots, x_n), \ldots, h_{2^n}(x_1, \ldots, x_n)\}.$$

Для функции $f(x_1, \ldots, x_n)$ известны коэффициенты разложения или вектор функции $(\alpha_1, \ldots, \alpha_k, \ldots, \alpha_{2^n})$ в базисе G. Задача заключается в построении алгоритма (имеющего в некотором смысле минимальную сложность), вычисляющего компоненты вектора $(\beta_1, \ldots, \beta_k, \ldots, \beta_{2^n})$ этой функции в базисе H.

Под сложностью $L_j(n)$ алгоритма будем понимать количество операций умножения $\cdot$ и сложения $\oplus$, которые нужно применить к коэффициентам разложения α_i для нахождения коэффициента β_j, и $L(n)$ — количество перестановок и операций умножения $\cdot$ и сложения $\oplus$, которые нужно применить к коэффициентам разложения α_i для нахождения всех β_j.

Для произвольных базисов существует универсальный алгоритм, использующий матрицу перехода. Однако он имеет большую асимптотическую сложность даже для нахождения одной компоненты: $L_j(n) \asymp 2 \cdot 2^n$, соответственно для всех компонент: $L(n) \asymp 2 \cdot 4^n$.

Алгоритм из работы [1] имеет верхнюю оценку сложности $L(n) \leqslant \dfrac{n}{2} \cdot 2^n$ для класса базисов, каждый из которых состоит из функций, являющихся операторными образами определенного класса функций. Для конкретного базиса выбираются образы одной функции.

Для дальнейшего изложения потребуются несколько определений. Более детальное описание используемых операторов и их свойств приведено в [2].

Оператор $\mathbf{t} : F_n \to F_n$ представляется в виде последовательности $t_1 \dots t_n$, $t_i \in \{d, e, p\}$. Компонента t_i оператора $\mathbf{t}$ действует на функцию $f(x_1, \dots, x_n)$ по переменной x_i следующим образом:

$$t_i f(x_1, \dots, x_n) = \begin{cases} f(x_1, \dots, x_n), \text{ если } t_i = e, \\ f(x_1, \dots, \bar{x}_i, \dots, x_n), \text{ если } t_i = p, \\ f'_{x_i}(x_1, \dots, x_n), \text{ если } t_i = d, \end{cases}$$

где $f'_{x_i}(x_1, \dots, x_n)$ — производная функции f по переменой x_i.

Оператор $\mathbf{t}$ действует на функцию так:

$$\mathbf{t}(f(x_1, \dots, x_n)) = t_1(t_2 \dots t_n(f(x_1, \dots, x_n))).$$

Пучком операторов называется упорядоченная последовательность 2^n операторов. Пучок $\mathbf{T} = (\mathbf{t}^0, \dots, \mathbf{t}^{2^n-1})$ называется базисным, если существует такая функция $g(x_1, \dots, x_n)$, что $\{\mathbf{t}^0(g(x_1, \dots, x_n)), \dots, \mathbf{t}^{2^n-1}(g(x_1, \dots, x_n))\}$ — базис F_n. Функция g в этом случае называется базисной.

Пучок $\mathbf{T}$ будет называться двупорожденным, если существуют операторы $\mathbf{a} = a_1 \dots a_n$ и $\mathbf{b} = b_1 \dots b_n$, в которых для любого i выполняется $a_i \neq b_i$,

1) $\mathbf{t}^0 = \mathbf{a}$,
2) $\mathbf{t}^{2^n-1} = \mathbf{b}$ и
3) $\mathbf{t}^i = t_1 \dots t_n$, где $t_k = a_k$, если в двоичном разложении числа $i = j_1 \dots j_k \dots j_n$ цифра $j_k = 0$, и $t_k = b_k$, если $j_k = 1$.

Пусть $\mathbf{a} = a_1 \dots a_n$ и $\mathbf{b} = b_1 \dots b_n$ — порождающие операторы двупорожденного пучка $\mathbf{T}$, а компоненты оператора $\mathbf{c} = c_1 \dots c_n$ удовлетворяют условию: $c_i \neq a_i$ и $c_i \neq b_i$ для всех $1 \leqslant i \leqslant n$.

Класс пучков $\mathbf{E_T}$ построен по пучку $\mathbf{T}$ следующим образом: $\mathbf{E}_j \in \mathbf{E_T}$, если оператор $\mathbf{o}^i$ пучка $\mathbf{E}_j$ совпадает с оператором $\mathbf{t}^i$ при $i \neq j$ и $\mathbf{o}^i = \mathbf{c}$ при $i = j$. Полагаем, что $\mathbf{T} \in \mathbf{E_T}$. Класс $\mathbf{E_T}$ называется расширением для $\mathbf{T}$, пучки из этого класса для краткости будем назвать расширенными.

Двупорожденные и расширенные пучки являются базисными [2]. Операторной формой функции будет называться разложение по базису, построенному по базисному операторному пучку и базисной функции.

Через $\mathbf{H_t}$ будет обозначаться класс пучков, в которых оператор с индексом 0 совпадает с $\mathbf{t}$; $\mathbf{H}$ — класс всех двупорожденных пучков; $\mathbf{EH_t}$ — класс расширений всех пучков из $\mathbf{H_t}$; $\mathbf{EH}$ — класс расширений всех пучков из $\mathbf{H}$.

Теорема 1. *Для любой функции $f(x_1, \ldots, x_n)$ для любой пары базисов, порожденных пучками из класса $\mathbf{EH}$, существует алгоритм перехода со следующей верхней границей сложности $L(n)$:*

1. *Если пучки из одного класса $\mathbf{H_t}$, то*

$$L(n) \leqslant \frac{n}{2} \cdot 2^n.$$

2. *Если пучки из класса $\mathbf{H}$ или из одного класса $\mathbf{EH_t}$, но не принадлежат одному классу $\mathbf{H_t}$ для любого $\mathbf{t}$ и не принадлежат одному классу $\mathbf{E_T}$ для любого $\mathbf{T}$, то*

$$L(n) \leqslant \left(\frac{n}{2} + 2\right) \cdot 2^n.$$

3. *Если пучки из класса $\mathbf{EH}$, но не принадлежат одному классу $\mathbf{EH_t}$ для любого $\mathbf{t}$ и не принадлежат классу $\mathbf{H}$, то*

$$L(n) < \left(\frac{n}{2} + 4\right) \cdot 2^n.$$

Замечание. *Верхняя оценка для класса $\mathbf{H}$ всех двупорожденных пучков, приведенная в* [1]*, не совпадает с оценкой пункта 2 теоремы ввиду разного определения сложности алгоритма. В теореме используется определенный порядок операторных образов функции, порождающей базис.*

Известный алгоритм перехода от совершенной полиномиальной нормальной формы (СПНФ) к полиному Жегалкина [3] *имеет сложность (без учета перестановок) $L(n) = \frac{n}{2} \cdot 2^n$ и неявно полагает присутствия определенного порядка функций в базисах. При таком порядке эти полиномиальные формы порождаются пучками, входящими в класс $\mathbf{H_{e\ldots e}}$ и, следовательно, для них верхняя граница определяется пунктом 1 теоремы. Можно заметить, что эти два пучка, порождающие базисы СПНФ и полинома Жегалкина, дают верхнюю границу сложности алгоритма перехода.*

Список литературы

[1] Винокуров С. Ф. Сложность алгоритмов построения полиномиальных форм булевых функций // Материалы 8-й Всероссийской конференции «Синтаксис и семантика логических систем» (Аршан, Республика Бурятия, 20–24 августа 2024 г.). Иркутск : Издательство ИГУ, 2024. С. 25–27.

[2] Избранные вопросы теории булевых функций / А. С. Балюк, С. Ф. Винокуров, А. И. Гайдуков, О. В. Зубков, К. Д. Кириченко, В. И. Пантелеев, Н. А. Перязев, Ю. В. Перязева. Под редакцией С. Ф. Винокурова и Н. А. Перязева. М. : ФИЗМАТЛИТ, 2001. 192 с.

[3] Шоломов Л. А. Основы теории дискретных логических и вычислительных устройств. СПб. : «Лань», 2011. 432 с.

Кривая Пеано — Гильберта, ее реализация конечным автоматом и ее свойства

Власов Андрей Владимирович, Гашков Сергей Борисович
Московский государственный университет имени М. В. Ломоносова;
andrei.vlasov@math.msu.ru, sbgashkov@gmail.com

Непрерывные отображения, которые переводят отрезок в квадрат, известны давно. В 1891 году немецкий математик Давид Гильберт опубликовал [1] свой вариант кривой, заполняющей пространство. Это отображение называется кривая Гильберта. В работе был построен конечный автомат с 4 состояниями, реализующий данное отображение.

Известно, что обе функции $P_i : [0;1] \to [0;1]$, являющиеся компонентами отображения Гильберта, непрерывны и удовлетворяют условию Гёльдера с показателем $\frac{1}{2}$ и не удовлетворяют этому условию с большим показателем [2–4]. Также былла известна [5] оценка сверху на константу в условии Гёльдера с показателем $\frac{1}{2}$:

$$|P_i(x) - P_i(y)| \leqslant \sqrt{6} * \sqrt{|x-y|}. \tag{1}$$

В данной работе было доказано, что улучшить оценку нельзя, то есть даже если выбирать x и y сколь угодно близкими друг к другу, модуль непрерывности

$$\omega(P_i, h) = \max_{x,y|x-y|\leqslant h} |P_i(x) - P_i(y)| \tag{2}$$

не может быть меньше $\sqrt{6h}$ при $h = (2/3)/4^{2n+i-1}$.

В работе также исследовались арифметические свойства отображения Гильберта. В частности, изучалось, как функции P_i преобразуют правильные рациональные дроби p/q. Было доказано, что период раациональной дроби не может увеличиться больше, чем в 2 раза. Также при помощи обратного автомата был построен пример уменьшения периода в 3 раза.

Также была доказана следующая теорема о свойствах автомата, реализующего кривую Гильберта.

Теорема 1. *У автомата, реализующего кривую Гильберта, не может быть меньше* 4 *состояний.*

Список литературы

[1] Hilbert D. Ueber die stetige Abbildung einer Linie auf an Flächenstück // Mathematische Annalen. 1891. Vol. 38. P. 459–460.

[2] Besicovitch A. S., Schoenberg J. J. On Jordan arcs and Lipschitz classes of functions defined on them // Acta Mathematica. 1961. Vol. 106. P. 113–136.

[3] Щепин Е. В. О фрактальных кривых Пеано // Труды Математического института имени В. А. Стеклова. 2004. Т. 247. С. 204–303.

[4] Щепин Е. В., Мычка Е. Ю. О нижних оценках квадратно-линейного отношения плоских кривых Пеано // Математические заметки. 2021. Т. 110, вып. 2. С. 289–296.

[5] Бауман К. Е. Коэффициент растяжения кривой Пеано — Гильберта // Математические заметки. 2021. Т. 110, вып. 2. С. 289–296.

Верхняя оценка переключательной мощности плоской автоматной схемы для автоматов с ограничениями на диаграммы Мура

Воротников Алексей Сергеевич
Московский государственный университет имени М. В. Ломоносова; vorotnikov.lexa@yandex.ru

В данной работе рассматривается понятие плоской автоматной схемы, являющееся расширением понятия схемы из клеточных элементов, введённого в работе Кравцова С. С. [1]. В работах [2, 3] Калачев Г. В. показал, что порядок потенциала и переключательной мощности схемы из клеточных элементов, реализующей булеву функцию от n переменных, составляет $2^{n/2}$.

Определения

В данной работе мы будем опираться на определение, введённое автором в работе [4]. Ниже вводятся меры сложности, отличные от мер сложности из [4], так как теперь мы рассматриваем схемы со входами.

Определение переключательной мощности. Состоянием схемы K на такте t при подаче на вход строки $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_l)$ длины l назовём вектор $s_K(\alpha, t) := (g_1(t), \ldots, g_h(t))$, где g_i — автоматная функция, реализуемая в i-м узле схемы K. Величину $c_K(t) := |s_K(\alpha, t) \oplus s_K(\alpha, t+1)|$ назовём затратой энергии на переключение схемы с такта t на $t+1$.

Переключательной мощностью схемы K на последовательности α назовём $W(K, \alpha) = \dfrac{1}{l} \sum\limits_{t=0}^{l-1} c_K(t, \alpha)$. Переключательной мощностью схемы K на

последовательностях длины s назовём $W(K,s) = \dfrac{1}{2^s}\sum\limits_{\alpha\in E^s} W(K,\alpha)$. Переключательной мощностью автомата A на последовательностях длины s назовём $W(A,s) = \min\limits_{A_K=A} W(K,s)$, где A_K — автомат, реализуемый схемой K.

Функцией Шеннона для переключательной мощности автоматов из класса $\mathcal{A}$ на последовательностях длины s назовём $W(\mathcal{A},s) = \max\limits_{A\in\mathcal{A}} W(A,s)$.

Определение рассматриваемого класса автоматов. Определим множество троек $\Gamma = (g,R,p)$, где первый элемент — граф, второй — множество выделенных вершин «корней», последний — число выделенных рёбер, называемое *число переключений*, следующим образом. Определим некоторый элемент из Γ и две операции, сохраняющие Γ:

1. Граф g содержит единственную вершину v и не имеет рёбер. Тогда $(g,\{v\},0)\in\Gamma$.
2. Операция «Образование петли». Пусть тройка (g,R,p) принадлежит Γ, $g=(V,E)$. Тогда для произвольной висячей вершины v графа g и произвольных $r,r'\in R; r\neq r'$ тройки $(g',R,p+1)$ и $(g'',R,p+2)$ также принадлежат Γ. Здесь $g'=(V,E\cup\{(v,r)\})$, $g''=(V,E\cup\{(v,r),(v,r')\})$.
3. Операция «Добавление дерева». Пусть тройка (g,R,p) принадлежит Γ. Тогда для любых $\mathcal{U},\mathcal{U}'\in D(n,d)$ и произвольной висячей вершины v графа g тройки $(g'\cup\mathcal{U},R',p+1)$ и $(g''\cup\mathcal{U}\cup\mathcal{U}',R'',p+2)$ также принадлежат Γ. Пусть ν и ν' — корни $\mathcal{U}$ и $\mathcal{U}'$ соответственно. Здесь $g' := (V,E\cup\{(v,\nu)\})$, $g'' := (V,E\cup\{(v,\nu),(v,\nu')\})$, $R' := R\cup\{\nu\}$, $R'' := R\cup\{\nu,\nu'\}$. Объединение графов понимается как объединение множеств вершин и множеств рёбер в предположении, что эти множества изначально не пересекаются.

Каждый граф из троек из множества Γ содержит в качестве подграфов деревья, из листьев которых произвольно проведены рёбра в корни других деревьев или же в корень самого подграфа. Деревья для построения выбираются из множества $D(n,d)$. Из каждой вершины выходит не более двух рёбер. Операция «Добавление дерева» позволяет построить тройку с большим числом вершин в графе, а операция «Образование петли» позволяет провести рёбра из листа некоторого поддерева в корень некоторого, возможно совпадающего, поддерева.

Определим $\Gamma(2^n)\subseteq\Gamma$ как множество всех троек, в которых граф g построен на 2^n вершинах и не имеет висячих вершин. Положим $\Gamma(2^n,s) := \{(g,R,p)\mid (g,R,p)\in\Gamma(2^n), p\leqslant s\}$. Графы этого множества не имеют висячих вершин, при этом рёбер, соединяющих корни и листья поддеревьев, не более s.

Определим $\mathbb{A}(2^n,s,d(n))$ — множество диаграмм Мура, полученных путём определения нагрузки на графах g из тройки $(g,R,p)\in\Gamma(2^n,s)$ следующим образом:

1. Если из вершины выходит только одно ребро, то нагрузим его символом $(0, 1)$, если два ребра, то одно — символом 0, другое — 1.
2. Выберем произвольную вершину и отметим её как начальную.
3. Нагрузим каждую вершину одним из символов $\{0, 1, x, \bar{x}\}$.

В данной работе представлена верхняя оценка для реализации плоскими автоматными схемами произвольного автомата из данного класса. В дальнейшем полагаем s по порядку не более $\frac{2^n}{n}$.

Теорема о верхней оценке переключательной мощности

Теорема 1.

$$W(\mathbb{A}(2^n, s, d(n)), l) \preceq \frac{2^{n/2}}{d(n)} \text{ при } n \to \infty,\ l \geqslant d(n),\ s \preceq \frac{2^n}{n},$$

причём для каждой пары соседних тактов затраты энергии на переключение схемы, реализующей автомат из класса $\mathbb{A}(2^n, s, d(n))$*, не больше по порядку* $2^{n/2}$*.*

Для доказательства теоремы строится схема, содержащая большее количество задержек относительно тривиальной реализации, однако именно избыточность по числу задержек позволяет уменьшить переключательную мощность. Результат достигается за счёт того, что отдельное дерево в рассматриваемом автомате при переходе автомата в состояние из него передаётся в относительно маленькую по площади область схемы, расположенную близко к выходу. Тем самым, пока автомат находится в состояниях внутри конкретного поддерева, схема выделяет очень мало мощности, и это продолжается не менее $d(n)$ тактов — хотя бы столько тактов нужно после попадания автомата в корень поддерева, чтобы дойти до листа. Переключение между автоматами, наоборот, требует активации значительной площади схемы, что приводит к скачку затрат энергии на переключение. Существенный результат достигается при растущем с ростом n параметре $d(n)$.

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. Вып. 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. Обобщение оценок мощности плоских схем, реализующих частичные булевы операторы. // Вестник Московского Университета. Серия 1. Математика. Механика. 2018. № 3. С. 60–64.

[4] Воротников А. С. О верхних оценках сложности синтеза автономных автоматных плоских схем // Интеллектуальные системы. Теория и приложения. 2023. Т. 27, вып. 2. С. 84–110.

О некоторых новых классах правильных семейств

Галатенко Алексей Владимирович,
Носов Валентин Александрович, Панкратьев Антон Евгеньевич,
Царегородцев Кирилл Денисович
Московский государственный университет имени М. В. Ломоносова;
agalat@msu.ru, vnosov40@mail.ru, apankrat@intsys.msu.ru, kirill94_12@mail.ru

Введение

Правильные семейства функций были введены В. А. Носовым в работе [1].

Определение 1. *Пусть* $k, n \in \mathbb{N}$, $k \geqslant 2$. *Семейство функций* $(f_1, \ldots, f_n)$, $f_i \in P_k^n$, *называется правильным, если для любых* $\alpha, \beta \in E_k^n$, $\alpha = (a_1, \ldots, a_n)$, $\beta = (b_1, \ldots, b_n)$, $\alpha \neq \beta$, *найдется индекс* i, $1 \leqslant i \leqslant n$, *такой что* $a_i \neq b_i$, *но* $f_i(\alpha) = f_i(\beta)$.

К настоящему моменту известен ряд примеров правильных семейств. В частности, несложно показать, что правильными являются треугольные семейства, для которых f_i может существенно зависеть только от переменных с меньшими номерами (с точностью до согласованной перенумерации переменных и функций; см., например, [2]). Одной из важных характеристик правильного семейства является мощность образа; в работе [3] показано, что в множестве треугольных семейств содержатся представители, мощность образа которых максимально возможная в классе правильных семейств.

Еще один интересный пример правильных семейств приведен в работе [2]. Пусть значность логики k является простым числом, n нечетно, φ есть некоторый перестановочный многочлен, $f_i = \varphi(x_{i+1}+1)\cdot\ldots\cdot\varphi(x_{i+1}+k-1)\cdot\varphi(x_{i+2})$, где индексы переменных зацикливаются естественным образом. Достоинствами таких семейств являются компактность спецификации и возможность задания функций с требуемыми ограничениями на степень многочлена.

Отметим, что доля известных примеров правильных семейств среди всех семейств порядка n стремится к 0 при $n \to \infty$. Таким образом, задача поиска новых примеров представляет несомненный интерес.

Рассмотрим обобщения приведенных классов правильных семейств.

Рекурсивно и локально треугольные семейства

Определение 2. *Семейство* $(f_1, \ldots, f_n)$ *является рекурсивно треугольным, если найдется индекс* i, $1 \leqslant i \leqslant n$, *такой что* f_i *является константой и каждое непустое семейство, полученное из* $(f_1, \ldots, f_n)$ *подстановкой некоторой константы вместо переменной* x_i *и исключением функции* f_i, *также является рекурсивно треугольным.*

Правильность рекурсивно треугольных семейств будет установлена как следствие теоремы 2.

Теорема 1. *Для числа рекурсивно треугольных семейств размера* n *справедлива формула:* $\Delta_k^{\mathsf{rec}}(n) = \sum_{j=1}^{n}(-1)^{j+1} \cdot k^j \cdot \binom{n}{j} \left(\Delta_k^{\mathsf{rec}}(n-j)\right)^{k^j}$, *где* $\Delta_k^{\mathsf{rec}}(0) = 1$. *При* $k = 2$ *доля рекурсивно треугольных семейств в классе всех правильных семейств стремится к* 0 *при* $n \to \infty$.

Интересно, что при $k = 2$ число рекурсивно треугольных семейств размера n совпадает с числом рекурсивных ориентаций куба $G(E_2^n)$ (см. последовательность A141770). Для обобщения второго утверждения теоремы на случай произвольного k требуется получение верхней оценки на число правильных семейств в общем случае, что является задачей дальнейших исследований.

В работе [4] для случая $k = 2$ был введен локальный граф существенной зависимости семейства функций. Он строится для некоторого входного набора $\alpha = (a_1, \ldots, a_n)$. Множество вершин есть $\{1, \ldots, n\}$, дуга (i, j) проводится если и только если найдется $a' \in E_k$, такое что $f_j(\alpha) \neq f_j(a_1, \ldots, a_{i-1}, a', a_{i+1}, \ldots, a_n)$ (обозначим это через $\partial_i f_j(\alpha) \neq 0$; фактически, это отсутствие существенной зависимости f_j от x_i в точке α). В работе [4] по сути было показано, что если для каждого $\alpha \in E_2^n$ локальный граф существенной зависимости булева семейства $(f_1, \ldots, f_n)$ ациклический, то семейство является правильным. Обобщим этот результат на случай $k \geqslant 3$.

Определение 3. *Семейство* $(f_1, \ldots, f_n)$ *называется локально треугольным в точке* $\alpha \in E_k^n$, *если с точностью до согласованной перенумерации переменных и функций* $\partial_i f_j(\alpha) = 0$ *для всех* $1 \leqslant j < i \leqslant n$. *Если это свойство выполнено для всех* α, *семейство называется локально треугольным.*

Несложно заметить, что локальная треугольность эквивалентна ацикличности графа существенной зависимости.

Теорема 2. *Если семейство является рекурсивно треугольным, то оно локально треугольное. Если семейство является локально треугольным, то оно правильное.*

Несложно увидеть, что треугольные семейства являются рекурсивно треугольными, но существуют рекурсивно треугольные семейства, не являющиеся треугольными. Кроме того, существуют локально треугольные семейства, не являющиеся рекурсивно треугольными. Из этих наблюдений следует, что мощность образа локально треугольного семейства может быть равна любому целому числу из интервала от 1 до k^{n-1}, то есть полностью покрывает весь диапазон мощностей образа правильных семейств.

В заключение раздела в таблице 1 приведем численные значения мощности для $k = 2$ и небольших n. Заполнение пустой ячейки является задачей дальнейших исследований.

Размер n	$\Delta(n)$	$\Delta^{\mathsf{rec}}(n)$	$\Delta^{\mathsf{loc}}(n)$	$T(n)$
$n = 1$	2	2	2	2
$n = 2$	12	12	12	12
$n = 3$	488	680	680	744
$n = 4$	481776	3209712	3349488	5541744
$n = 5$	157549032992	94504354122272	...	638560878292512

Табл. 1: мощности классов при небольших n. Символом $\Delta(n)$ обозначено число треугольных, $\Delta^{\mathsf{loc}}(n)$ — локально треугольных, $T(n)$ — число правильных семейств размера n при $k = 2$.

Обобщение конструкции с перестановочным многочленом

Определение 4. *Пусть $c, d \in E_k$, $h \in P_k^k$. Функция h обладает (c, d)-свойством, если на любом входном наборе, содержащем значение c, она принимает значение d.*

Например, функции min и умножение по модулю k обладают $(0, 0)$-свойством, а функция max обладает $(k-1, k-1)$-свойством.

Будем считать, что индексы $1, 2, \ldots, n$ «скручены в кольцо»: за n следует 1, перед единицей идет n. Пусть функции $h_i \in P_k^k$, $i = 1, \ldots, n$, обладают (c, d)-свойством, $g \in P_k^1$ принимает значение c хотя бы на одном входном наборе, $I_1 = (c_1, \ldots, c_t)$, $I_2 = (c_{t+1}, \ldots, c_k)$ — две непересекающиеся подпоследовательности элементов E_k, в объединении дающие все E_k. Рассмотрим семейство $(f_1, \ldots, f_n)$, определенное следующим соотношением:

$$f_i = h_i\left(g(x_{i+1} + c_1), \ldots, g(x_{i+1} + c_t), g(x_{i+2} + c_{t+1}), \ldots, g(x_{i+2} + c_k)\right). \quad (1)$$

Теорема 3. *Семейство* (1) *является правильным при нечетном n.*

Список литературы

[1] Носов В. А. Критерий регулярности булевского неавтономного автомата с разделенным входом // Интеллектуальные системы. 1998. Т. 3, № 3–4. С. 269–280.

[2] Носов В. А. Построение параметрического семейства латинских квадратов в векторной базе данных // Интеллектуальные системы. 2006. Т. 8, № 1–4. С. 517–529.

[3] О порождении n-квазигрупп с помощью правильных семейств функций / А. В. Галатенко, В. А. Носов, А. Е. Панкратьев, К. Д. Царегородцев // Дискретная математика. 2023. Т. 35, № 1. С. 35–53.

[4] Shih Mau-Hsiang, Dong Jian-Lang. A combinatorial analogue of the Jacobian problem in automata networks // Advances in Applied Mathematics. 2005. Vol. 34. P. 30–46.

Умножение и деление натуральных чисел клеточными автоматами с локаторами

Гасанов Эльяр Эльдарович[1], Хайбуллин Бакир Фаридович[2]

1 Московский государственный университет имени М. В. Ломоносова; el_gasanov@mail.ru

2 ООО «Elius», г. Ташкент, Узбекистан; bakir_k@mail.ru

Пусть a и b — два натуральных числа, двоичная запись которых содержит по порядку n разрядов. Наиболее известный и быстрый алгоритм умножения таких чисел был предложен А. А. Карацубой [1], и он имеет сложность $O(n^{\log_2 3})$. Более быстрым по порядку алгоритмом умножения является алгоритм Шёнхаге — Штрассена [2]. Его сложность $O(n \cdot \log n \cdot \log\log n)$. Но на практике алгоритм Шёнхаге — Штрассена быстрее алгоритма Карацубы, только если значность числа более 10 тысяч. Еще более быстрым по порядку является алгоритм Фюрера [3], но его преимущество может проявиться при значности чисел более 10^{13}. Относительно недавно появился алгоритм Харви — ван дер Хувена [4] со сложностью $O(n \log n)$.

Для деления натуральных чисел с остатком известен алгоритм Бурникеля — Циглера [5]. Он использует внутри себя алгоритм умножения. Если в качестве алгоритма умножения взять алгоритм Карацубы, то вычислительная сложность алгоритма Бурникеля — Циглера будет $O(n^{\log_2 3})$, а если использовать алгоритм умножения Шёнхаге — Штрассена, то сложность алгоритма Бурникеля — Циглера будет $O(n \cdot \log^2 n \cdot \log\log n)$.

В данной работе предлагаются алгоритмы решения задач умножения и деления с остатком n-значных натуральных чисел с помощью клеточных автоматов с локаторами.

Приведем неформальное описание двумерного клеточного автомата с локаторами.

Расположим в каждой клетке плоской решетки $\mathbb{Z}^2$ один и тот же автомат с локаторами. Понятие локатора определим чуть позже, сейчас важно, что каждый локатор в каждый момент принимает некоторое значение. Автомат имеет функцию перехода, которая по состоянию соседей автомата и по значениям локаторов в текущей момент определяет состояние автомата в следующий момент. Кроме того, у автомата есть функция вещания, которая по состояниям соседей автомата и по значениям локаторов вычисляет сигнал вещания, который передается в эфир. Сигналы вещания образуют конечную аддитивную коммутативную полугруппу, а эфир представляет собой потенциально бесконечный сумматор сигналов элементарных автоматов, где в качестве оператора суммы выступает определяющая операция данной полугруппы. Каждый локатор представляет собой некоторый телесный угол с вершиной в позиции автомата, а значением локатора в текущий момент является сумма сигналов вещания всех автоматов, попадающих в этот телесный угол. Отметим, что в область суммирования локатора не входит вершина телесного угла. т. е. мы сигнал вещания, посылаемый данным автоматом, не включаем в сумму.

Строгое определение клеточного автомата с локаторами можно найти в [6].

В наших алгоритмах будут использоваться один полный локатор, который представляет собой двумерную плоскость с выколотым началом координат, и 8 локаторов, представляющих собой лучи, направленные на север, северо-восток, восток, юго-восток, юг, юго-запад, запад и северо-запад.

Определим задачу умножения чисел a и b для клеточного автомата с локаторами. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами $(0, 0)$ находится в состоянии, которое можно назвать «начало координат», ячейка с координатами $(a, 0)$ находится в состоянии, которое можно назвать «первый сомножитель», а ячейка с координатами $(0, b)$ находится в состоянии, которое можно назвать «второй сомножитель». Клеточный автомат решает задачу умножения чисел, если в финальной конфигурации ячейка с координатами $(a \cdot b, 0)$ перейдет в состояние «результат умножения», а все остальные ячейки, кроме $(0, 0)$, перейдут в состояние покоя.

Справедлива следующая теорема, доказанная Э. Э. Гасановым.

Теорема 1. *Существует двумерный клеточный автомат с* 9 *локаторами, который решает задачу умножения чисел* a *и* b *за время* $2\lceil \log_2 a \rceil + 2$.

Здесь если x — вещественное число, то $]x[$ — это наименьшее целое, не меньшее x.

Определим задачу деления чисел a и b с остатком для клеточного автомата с локаторами. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами $(0, 0)$ находится в состоянии,

которое можно назвать «начало координат», ячейка с координатами $(a, 0)$ находится в состоянии, которое можно назвать «делимое», а ячейка с координатами $(0, b)$ находится в состоянии, которое можно назвать «делитель». Пусть $c = \lfloor a/b \rfloor$ — целая часть от деления a на b, $d = a \mod b$ — отстаток от деления a на b. Клеточный автомат решает задачу деления чисел, если в финальной конфигурации ячейка с координатами $(c, 0)$ перейдет в состояние «частное», ячейка с координатами $(0, d)$ перейдет в состояние «остаток», а все остальные ячейки, кроме $(0, 0)$, перейдут в состояние покоя.

Справедлива следующая теорема, доказанная Б. Ф. Хайбуллиным.

Теорема 2. *Существует двумерный клеточный автомат с* 9 *локаторами, который решает задачу деления чисел* a *и* b *с остатком за время* $3\lceil \log_2(a/b) \rceil + 8$.

Список литературы

[1] Карацуба А., Офман Ю. Умножение многозначных чисел на автоматах // Доклады Академии наук СССР. 1962. Т. 145, № 2. С. 293–294.

[2] Schönhage A., Strassen V. Schnelle Multiplikation großer Zahlen // Computing. 1971. No. 7. P. 33–47.

[3] Fürer M. Faster integer multiplication // Proceedings of the thirty-ninth annual ACM symposium on Theory of computing. New York, NY, USA : Association for Computing Machinery, 2007. P. 57–66.

[4] Harvey D., van der Hoeven J. Integer multiplication in time $O(n \log n)$ // Annals of Mathematics. 2021. Vol. 193, no. 2. P. 563–617.

[5] Burnikel C., Ziegler J. Fast recursive division. Research Report MPI-I-98-1-022. Saarbrücken, Germany : Max-Planck-Institut für Informatik, 1998.

[6] Гасанов Э. Э. Клеточные автоматы с локаторами как модель устройств с беспроводной связью // Математические вопросы кибернетики. 2023. Т. 21. С. 5–51.

О диаметре начала натурального ряда в одной арифметической модели

Дергач Пётр Сергеевич[1], Дускаев Рифат Ринатович[2]

1 Московский государственный университет имени М. В. Ломоносова; dergachpes@mail.ru

2 Московский государственный университет имени М. В. Ломоносова, филиал в городе Ташкенте; duskaevrifat2904@mail.ru

Аннотация

Данная статья является тезисами к докладу автора на XX Международной конференции «Проблемы теоретической кибернетики». В докладе излагают-

ся результаты об асимптотике диаметра начала натурального ряда в одной арифметической модели. Более подробная постановка задачи приводится во введении. Ознакомиться с аналогичными задачами по той же тематике можно в работах [1, 2].

Введение

Приведем аккуратную постановку исследуемой задачи. Пусть $n \in \mathbf{N}$ — натуральное число. Множество натуральных чисел от 1 до n обозначаем через $[n]$. Расстоянием между произвольными двумя числами a, b из $[n]$ называем минимальное количество $d(a, b)$ арифметических операций $+1, -1, *2, /2$ достаточное для того, чтобы из числа a получить число b. При этом можно в ходе операций выходить за границы множества $[n]$, но все промежуточные вычисления должны быть целочисленны, то есть операция $/2$ применима лишь к четным числам. Диаметром $d(n)$ называем самое большое из попарных расстояний между элементами из $[n]$. Необходимо получить как можно более полную информацию о поведении данной функции. В рамках данной статьи приводится результат об ее асимптотике. В дальнейшем представляется возможным получить ее точное значение, но это потребует дополнительной проработки и аккуратного исследования.

Вспомогательные результаты

Определение. *Называем* каноническим путем *между числами* $a, b \in [n]$ *последовательность операций вида*

$$a : a : \ldots a : b * a * a \ldots * a,$$

где вместо каждого вхождения a *подставляются (независимо друг от друга) или одна из операций* $+1$, -1, *или отсутствие операции, через* b *обозначено или некоторое (возможно нулевое) количество операций* $+1$, *или некоторое (возможно нулевое) количество операций* -1, *через* $*$ *обозначена операция* $*2$, *через* $:$ *обозначена операция* $/2$. *При этом как количество умножений, так и количество делений в этой последовательности может быть равно* 0. *Если в каноническом пути* k *умножений и* l *делений, то для краткости называем такой путь* k, l-каноническим путем.

Лемма 1. *Для произвольных* $a, b \in [n]$ *расстояние* $d(a, b)$ *реализуется на одном из канонических путей между* a, b.

Доказательство. Утверждение тривиально следует из четырех наблюдений. Первое — если где-то в последовательности операций после умножения идет деление, то можно взять две такие соседние операции, между которыми будут

только операции сложения и вычитания. Но в этом случае цепочку $t * \pm^{2k} :$ (степень обязана быть четной из-за операции деления) можно заменить на $t\pm^k$, тем самым уменьшив ее длину. Второе — если где-то в блоке из идущих подряд операций сложения и вычитания есть обе операции, то соседнюю пару противоположных операций можно было бы убрать. Третье — непосредственно перед операцией деления не может идти больше одной операции $\pm$, так как в этом случае цепочка $t \pm \pm :$ заменяется на более короткую $t : \pm$. И четвертое — непосредственно после операции умножения не может идти больше одной операции $\pm$, так как в этом случае цепочка $t * \pm\pm$ заменяется на более короткую $t \pm *$. □

Лемма 2. *Примененная к числу* $t \in \mathbf{N}$ *последовательность операций*

$$a_1 : a_2 : \ldots : a_k :,$$

где через a_i, $i \in [k]$, *обозначены или одна из операций* $+1$, -1, *или отсутствие операции, переводит это число или в целую часть снизу или в целую часть сверху от* $\frac{t}{2^k}$.

Доказательство. Доказательство тривиально следует из того факта, что получаемое число равно $\frac{t}{2^k} + \frac{a_1}{2^k} + \frac{a_2}{2^{k-1}} + \ldots + \frac{a_k}{2}$, где $a_i \in \{-1, 0, 1\}$. □

Основные результаты

Теорема.

$$d(n) = 3\log_2 n \cdot (1 + o(1)).$$

Доказательство. Ввиду ограниченности объема тезисов приведем здесь лишь общую схему доказательства.

Для обоснования верхней оценки достаточно заметить, что если число записать в бинарном виде (в системе исчисления по модулю 2), то для того, чтобы стереть 2 бита на конце числа, всегда хватает 3 операции. В самом деле, для стирания 00 достаточно провести ::, для $01 — -$ ::, для $10 — : -$:, для $11 — +$ ::. Таким образом, для преобразования числа n к 1 асимптотически достаточно $\frac{3}{2}\log_2 n$ операций, откуда тривиально получаем требуемую верхнюю оценку на произвольную пару чисел из $[n]$.

Для обоснования нижней оценки предъявим конкретную пару чисел с двоичными представлениями $a = 10(01)^s$ и $b = 11(01)^s$, где параметр s выбирается так, чтобы эти числа были как можно ближе к n, но не превосходили его. Покажем, что расстояние между этой парой чисел асимптотически не меньше чем $3\log_2 n$. Из леммы 1 следует, что расстояние между ними достигается на каком-то k, l-каноническом пути. Из леммы 2 также следует, что выбор такого пути полностью определяется его параметрами k, l и выбором

того, в какую сторону идет округление — вниз или вверх. Другими словами, мы можем поделить одно число на 2^k с округлением вниз или вверх, а второе число — на 2^l с округлением вниз или вверх (основной этап). И для полученных чисел уже с помощью операции $+1$ одно преобразуется в другое (средний этап). Обсудим более детально, как правильно выбирать параметры k, l и сторону округления. Интуитивно понятно, что сложность перехода от числа a к числу $\frac{a}{2^k}$ с округлением вниз образуется из k делений и примерно (асимптотически) $\frac{k}{2}$ вычитаний — количество 1 среди последних k битов числа a. Аккуратно это можно обосновать, применив лемму 2 и ее разложение в виде суммы со знаком некоторых отрицательных степеней двойки. Здесь крайне важно, что в двоичном представлении числа a чередуются 0 и 1, ведь иначе несколько подряд идущих 1 могли бы нам сэкономить количество операций ± 1. То же самое можно сказать и про округление вверх, но там уже важно будет не количество 1 среди последних k битов, а количество 0. В любом случае, асимптотически опять получаем $\frac{3}{2}k$ операций. С параметром l ситуация полностью аналогична. Далее можно заметить, что если параметры k, l далеки друг от друга, то нам выгодно сблизить их, прибавив к меньшему 1, а из большего отняв 1. Тем самым можно считать, что k, l отличаются друг от друга не больше чем на маленькую константу. Наконец, замечаем, что если параметры k, l уменьшить одновременно на 1, то на основном этапе мы потеряем примерно 3 операции, а вот выигрыш от преобразований сложения в среднем этапе будет больше. Тем самым оптимальной будет ситуация, когда параметры k, l примерно равны $2s$. В этом случае на среднем этапе мы почти ничего не потеряем (числа почти одинаковы и равны маленьким значениям), а основной этап даст $\frac{3}{2}(2s) + \frac{3}{2}(2s) = 6s = 3\log_2 n \cdot (1 + o(1))$ операций. □

Список литературы

[1] Conway J. H. On numbers and games. London, England : Academic Press Inc., 1976.

[2] Деорнуа П. Комбинаторная теория игр. М. : МЦНМО, 2017. 40 с.

О классах сверхфункций, замкнутых относительно операции отрицания

Дудакова Ольга Сергеевна
Московский государственный университет имени М. В. Ломоносова; olga.dudakova@gmail.com

Работа относится к теории функциональных систем. Рассматриваются обобщения булевых функций, называемые сверхфункциями. Сверхфункция — это произвольное непустое множество булевых функций, зависящих от одних и тех же переменных. На множестве сверхфункций можно определить

операции суперпозиции (как объединение всевозможных суперпозиций булевых функций, входящих во все сверхфункции), введения и удаления фиктивных переменных и другие. Отметим, что сверхфункции можно также считать обобщением понятия гиперфункций — функций, определенных на наборах из нулей и единиц и принимающих значения из множества $\{0, 1, \{0, 1\}\}$ (см., например, [1, 2]). Семейство классов сверхфункций, замкнутых относительно операций суперпозиции, добавления и удаления фиктивных переменных и операции перехода к подмножеству, изучалось в работах [3, 4]. В данной работе рассматривается семейство сверхфункций с теоретико-множественными операциями и операцией отрицания.

Множества всех булевых функций и всех сверхфункций обозначаются через P_2 и $\mathbf{P}_2$ соответственно; через $P_2(n)$ и $\mathbf{P}_2(n)$ обозначаются множества булевых функций и сверхфункций соответственно, зависящих от n фиксированных переменных. Пусть $F = \{f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)\}$ — сверхфункция, $f_1, \ldots, f_m \in P_2(n)$. Функции $f_1, \ldots, f_m$ будем называть компонентами сверхфункции F и писать $f_i \in F$. Результатом применения операции отрицания к сверхфункции F будем называть сверхфункцию $\overline{F} = \{\overline{f}_1, \ldots, \overline{f}_m\}$. Дополнением сверхфункции F будем называть сверхфункцию $\neg F$, состоящую из всех функций из $P_2(n) \setminus F$. Далее, пусть $F(x_1, \ldots, x_n)$ и $G(x_1, \ldots, x_n)$ — две сверхфункции, зависящие от одних и тех же переменных. Объединением $F \cup G$ сверхфункций F и G будем называть объединение множеств функций, входящих в F и G, пересечением $F \cap G$ сверхфункций — пересечение множеств входящих в них функций.

На множестве $\mathbf{P}_2$ всех сверхфункций определим операцию замыкания: пусть $\mathbf{F} \subseteq \mathbf{P}_2$, тогда замыкание $[\mathbf{F}]$ множества $\mathbf{F}$ — это множество всех сверхфункций, которые получаются из $\mathbf{F}$ применением операций объединения, пересечения, дополнения и отрицания. Легко видеть, что для так определенной операции замыкания выполняются основные свойства замыкания, в частности, для любого $\mathbf{F} \subseteq \mathbf{P}_2$ выполняется $[[\mathbf{F}]] = [\mathbf{F}]$. Множество $\mathbf{F}$ сверхфункций называется замкнутым, если $[\mathbf{F}] = \mathbf{F}$.

Пусть $\mathbf{F}$ — множество сверхфункций, зависящих от переменных $x_1, \ldots, x_n$. По множеству $\mathbf{F}$ построим разбиение $\Delta_{\mathbf{F}}$ множества $P_2(n)$ на непересекающиеся непустые подмножества. Будем говорить, что $f(x_1, \ldots, x_n)$ — функция 1-го типа, если для каждой сверхфункции $F \in \mathbf{F}$ выполняется одно из двух соотношений: $f, \overline{f} \in F$ или $f, \overline{f} \notin F$. Остальные функции из $P_2(n)$ будем называть функциями 2-го типа. Далее, пусть f и g — функции 1-го типа. Будем говорить, что пары функций $f, \overline{f}$ и $g, \overline{g}$ эквивалентны, если для каждой сверхфункции $F \in \mathbf{F}$ эти пары одновременно входят или не входят в F. Указанное отношение эквивалентности разбивает все функции 1-го типа на подмножества. Пусть теперь f и g — функции 2-го типа. Будем говорить, что функции f и g эквивалентны, если для каждой сверхфункции $F \in \mathbf{F}$ функции f и g

входят или не входят в F одновременно и функции $\overline{f}$ и $\overline{g}$ также входят или не входят в F одновременно. Таким образом, получили разбиение множества всех функций 2-го типа на подмножества. Все построенные таким образом подмножества функций из $P_2(n)$ образуют семейство $\Delta_{\mathbf{F}}$.

Утверждение. *Пусть* $\mathbf{F} \subseteq \mathbf{P}_2(n)$. *Замыкание множества* $\mathbf{F}$ *состоит из всех сверхфункций, получающихся объединением некоторых подмножеств булевых функций из семейства* $\Delta_{\mathbf{F}}$.

Систему $\mathbf{F}$ сверхфункций из $\mathbf{P}_2(n)$ назовем n-полной, если $[\mathbf{F}] = \mathbf{P}_2(n)$.

Следствие 1. *Система сверхфункций* $\mathbf{F} = \{F_1, \ldots, F_k\}$ *из* $\mathbf{P}_2(n)$ *является* n*-полной тогда и только тогда, когда выполняются два условия: 1) для любой булевой функции* $f \in P_2(n)$ *в* $\mathbf{F}$ *найдется сверхфункция, содержащая ровно одну из функций* $f, \overline{f}$, *2) для любых двух функций* $g, h \in P_2(n)$, *таких что* $g \neq \overline{h}$, *в* $\mathbf{F}$ *найдется сверхфункция, содержащая ровно одну из функций* g, h *или ровно одну из функций* $\overline{g}, \overline{h}$.

Пусть Δ — разбиение множества $P_2(n)$ на непустые непересекающиеся подмножества. Будем говорить, что семейство $\mathbf{F}$ сверхфункций из $\mathbf{P}_2(n)$ порождается разбиением Δ, если в $\mathbf{F}$ содержатся всевозможные сверхфункции, которые получаются объединением некоторых подмножеств из Δ.

Разбиение Δ множества $P_2(n)$ на непустые непересекающиеся подмножества $A_1, \ldots, A_k$ будем называть правильным, если каждое подмножество можно отнести к одному из двух типов: 1) если $f \in A_i$, то $\overline{f} \in A_i$, 2) если $f \in A_j$, то $\overline{f} \notin A_j$ и при этом если функции $f_1, \ldots, f_l$ лежат в одном и том же подмножестве, то функции $\overline{f}_1, \ldots, \overline{f}_l$ также лежат в одном и том же подмножестве. Заметим, что если разбиение Δ правильное, то порожденное им множество сверхфункций является замкнутым классом.

Следствие 2. *Множество замкнутых классов сверхфункций в* $\mathbf{P}_2(n)$ *совпадает с множеством классов, порожденных правильными разбиениями множества* $P_2(n)$.

Работа выполнена при финансовой поддержке Минобрнауки России в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2022-284.

Список литературы

[1] Machida H. Hyperclones on a two-element set // Multiple-Valued Logic. 2002. Vol. 8, no. 4. P. 495–501.

[2] Čolić Oravec J., Machida H., Pantović J., Vojvodić G. From clones to hyperclones // Selected Topic in Logic in Computer Science. 2. Zbornik radova. 2015. Vol. 18, no. 26. P. 111–144.

[3] Маслова И. И. О классах сверхфункций на двухэлементном множестве // Вестник Московского университета. Серия 1: Математика. Механика. 2021. № 5. С. 60–64.

[4] Маслова И. И. Описание всех классов сверхфункций, состоящих из дизъюнкций // Вестник Московского университета. Серия 1: Математика. Механика. 2022. № 4. С. 21–27.

Интервальное кодирование дискретных структур, сохраняющее метрические свойства отделимости

Евдокимов Александр Андреевич
Институт математики имени С. Л. Соболева СО РАН;
Новосибирский государственный университет; evdok@math.nsc.ru

При решении задач кодирования информации для последующей работы с ней в компьютере бывает полезно, а иногда и необходимо, иметь описание объекта исследования в хорошо структурированном виде, которое позволяет воспроизвести в кодовом пространстве структуру кодируемого объекта и использовать сохраняемые данные для их эффективной обработки [1]. Отметим, что структурированное кодирование, предполагая сохранение кодирующим отображением различного типа свойств, позволяет распознавать и корректировать «ошибки» искажения при передаче или хранении информации, анализируя выполнение или невыполнение сохраняемых отображением свойств. Другой прикладной аспект такого подхода состоит в том, что при полном или частичном (например, локально изометрическое кодирование) воспроизведении в кодовом пространстве структуры исходного кодируемого множества появляется возможность работы с данными уже в их машинных кодах с использованием операций, взаимосвязанных с архитектурой и спецификой компьютера. Это бывает удобно и может ускорить процесс обработки данных.

Рассмотрим сказанное на примере вложений графов в n-мерные булевы кубы, в частности, кодирования сеточного табло (двумерной целочисленной решётки конечного размера) и вариантов его вложения в гиперкуб. Мы дополняем исследованное ранее локально изометрическое кодирование табло и кодирование в классе отображений ограниченного искажения [2, 3] рассмотрением интервальных вложений специального вида и кодирования, сохраняющего структуру и отделимость элементов метрических структур: кодируемого множества (табло) и кодового пространства (гиперкуб с метрикой Хемминга).

Пусть G и H — связные графы с метрикой пути, $e = (u, v)$ — ребро в G, и инъективное отображение $f : G \to H$ множества вершин $V(G)$ в $V(H)$ удовлетворяет двум условиям:

1) для любых двух инцидентных ребер $e_1 = (u, v_1)$, $e_2 = (u, v_2)$ в G имеем $f(e_1) \cap f(e_2) = f(u)$,

2) для любых не инцидентных ребер e_i и e_j выполняется $f(e_i) \cap f(e_j) = \varnothing$,

где $f(e)$ — метрический отрезок вершин $f(u)$ и $f(v)$ в H, то есть множество всех вершин w графа H, для которых $\rho_H(f(u), w) + \rho_H(w, f(v)) = \rho_H(f(u), f(v))$. Если H — булев гиперкуб I^n и для вложения $f : G \to I^n$ выполнены условия (1), (2), а все метрические отрезки $f(e)$ в I^n (то есть подкубы) имеют одинаковую размерность, равную k, $k < n$, то вложение называется *k-интервальным*.

Проиллюстрируем метод построения k-интервального вложения $f : N_m^2 \to I^n$ двумерной целочисленной решётки $N_m^2 = N_m \times N_m$ размера $m \times m$ с расстоянием $\rho_{N^2}(u, v) = |x_1 - x_2| + |y_1 - y_2|$, где $N_m = \{0, 1, \ldots, m-1\}$. В основе метода лежит интерпретация задачи в терминах комбинаторики слов, которую автор применял для решения задач структурированного кодирования, конструкций вложений графов в гиперкубы, описания динамики функционирования дискретных моделей генных сетей и др. Выберем значения параметров $m = 8$, $k = 2$, $n = 10$, заметив, что конструкция будет понятна и в общем случае, и что она также обобщается на решётки размерности большей двух. Определим условия, которым должна удовлетворять последовательность в алфавите $\langle 1, 2, 3, 4, 5\rangle$ ортов пятимерного куба, которая определяет 2-интервальное вложение $f : N_8 \to I^5$ цепи N_8. Нетрудно понять, что условие (1) будет выполняться, поскольку отображение f инъективно. Метрические отрезки ребер цепи N_8 будут подкубами размерности $k = 2$, то есть квадратами. По определению интервального вложения эти квадраты должны иметь общими лишь вершины, являющиеся образами узлов решетки N_8^2. Поэтому 2-интервальное вложение $f : N_8 \to I^5$ удобно задать словом длины 14, которое разделено на блоки длины 2: $X = 1\,2.3\,4.1\,2.3\,5.1\,2.3\,4.1\,2$. Легко проверяется, что в любом подслове-отрезке слова X есть буква, входящая в это подслово нечетное число раз, то есть в пятимерном кубе цепь, определяемая словом X, не имеет самопересечений. Перестановке двух букв внутри любого блока соответствует 2 варианта прохождения квадрата, и легко проверить, что действие этих перестановок на слово X не приводит к самопересечению. Это означает, что свойства (1) и (2) 2-интервальности вложения $f : N_8 \to I^5$ выполнены. По слову X кодирующее отображение $f : N_8^2 \to I^{10}$ определяется стандартным образом. Однако надо учесть, что в последовательности кодовых слов длины 5 $\tilde{\alpha}_0, \tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_{14}$, $\tilde{\alpha}_0 = (00000)$, нужно выбрать подпоследовательность слов с четными индексами. Тогда 2-интервальное вложение $f : N_8^2 \to I^{10}$, определяющее кодирование решётки, задаётся отображением $f(x, y) = \tilde{\alpha}_{2x}\tilde{\alpha}_{2y}$ вершин (x, y) двумерной целочисленной решётки в кодовые слова длины 10, образованные конкатенацией кодовых слов длины 5. Если для работы с табло нужны не вершины клеток

его решётки, а сами клеточные области, то их кодирование осуществляется переходом к двойственному планарному графу и аналогично описанному.

В заключение о задачах, в которых сохранение структуры кодируемых объектов должно сочетаться со свойствами их отделимости и помехоустойчивости. Это связано с двумя различными понятиями отделимости и окрестности. Шаровая окрестность и интервальное замыкание. На примере табло и его кодирования с шаровой окрестностью, например, радиуса 1, эта задача решается в [3]. В общем случае задача трудна, поскольку содержит в качестве подзадачи результаты по оценкам мощности цепных кодов и их конструкции. Если в аналогичной постановке задачи достаточно использования метрической интервальной окрестности, используя вместо шаров, например, 2-интервальное замыкание, то задача решается с помощью несложной конструкции и построения соответствующего алгоритма вложения.

Список литературы

[1] Евдокимов А. А. Кодирование структурированной информации и вложения дискретных пространств // Дискретный анализ и исследование операций. Серия 1. 2000. Т. 7, № 4. С. 48–58.

[2] Евдокимов А. А. Вложения в классе параметрических отображений ограниченного искажения // Ученые записки Казанского государственного университета. Серия Физико-математические науки. 2009. Т. 151, кн. 2. С. 72–79.

[3] Евдокимов А. А. Кодирование конечной целочисленной решетки в классе отображений ограниченного искажения // Прикладная дискретная математика. Приложение. 2011. № 4, С. 8–9.

Анализ решений задачи составления расписания выполнения заказов клиентов с двумя критериями

Захаров Алексей Олегович, Захарова Юлия Викторовна
Институт математики имени С. Л. Соболева СО РАН, Омский филиал;
{azakharov,yzakharova}@ofim.oscsbras.ru

Постановка задачи

Рассматривается задача составления расписания выполнения заказов m клиентов, заданных множеством $M = \{1, \ldots, m\}$, каждый из который характеризуется своим (одним) заказом. Обозначим множество различных продуктов через $N = \{1, \ldots, n\}$. Заказ каждого клиента состоит из некоторого подмножества множества продуктов. Заданы следующие входные данные задачи: длительность производства $p_{ij} \geqslant 0$ продукта j для клиента i, величина

начальной переналадки $s_j \geqslant 0$ для производства продукта j, длительность переналадки $s_{jj'} \geqslant 0$ с производства продукта j на продукт j', где $j \neq j'$, $\forall j, j' \in N$, $\forall i \in M$.

Такая задача ранее исследовалась в [1, 2], где в качестве критериев построения оптимального расписания ставились: минимизация суммы моментов завершения выполнения заказов клиентов; максимизация суммы весов заказов, выполненных в срок. Задачи рассматривались как однокритериальные (несколько задач с одной моделью, но различными критериями). Решение задается в виде перестановки операций $o = (i, j)$, где пара (i, j) означает производство продукта j для клиента i. Тогда множество всех операций обозначим через $O = \{(i, j) |\ \forall i \in M,\ \forall j \in N\}$.

В данной работе предлагается рассмотреть двухкритериальную задачу с критериями минимизации суммы моментов завершения выполнения заказов и минимизации момента завершения выполнения последнего заказа и исследовать её множество Парето [3, 4].

Математическая модель и множество Парето

Будем предполагать, что в заказе каждого клиента присутствует n продуктов. Описанная выше модель задается следующей моделью целочисленного линейного программирования, в которой расписание кодируется в виде булева вектора x длины nm с компонентами

$$x_{ok} = \begin{cases} 1, & \text{если операция } o \in O \text{ находится в позиции } k \in K, \\ 0 & \text{иначе;} \end{cases}$$

$$\sum_{k \in K} x_{ok} = 1 \quad \forall o \in O, \tag{1}$$

$$\sum_{o \in O} x_{ok} = 1 \quad \forall k \in K, \tag{2}$$

$$t_1^f \geqslant \sum_{o \in O} x_{o1}(p_o + s'_o), \tag{3}$$

$$t_k^f \geqslant t_{k-1}^f + p_o + \sum_{o' \in O} x_{o',k-1} s_{o'o} - H(1 - x_{ok})\ \forall k \in \{2, \ldots, nm\},\ \forall o \in O, \tag{4}$$

$$T_i \geqslant t_k^f - H(1 - x_{ok}) \quad \forall k \in K,\ \forall o \in O, \tag{5}$$

$$T_i \geqslant 0,\ t_k^f \geqslant 0 \quad \forall i \in M,\ \forall k \in K, \tag{6}$$

где: t_k^f — момент завершения операции, находящейся в позиции $k \in K$; T_i — момент завершения выполнения заказа клиента $i \in M$; H — некоторая большая константа; $K = \{1, \ldots, nm\}$. Здесь длительности и переналадки приведены с индексами операции $o = (i, j)$, их значения естественным образом получаются из длительностей и переналадок соответствующего продукта. Критерий $f = (f_1, f_2)$ будет иметь компоненты, каждая из которых рассматривается на минимум

$$f_1 = \sum_{i \in M} T_i, \quad f_2 = \max_{i \in M} T_i. \tag{7}$$

Обозначим множество допустимых решений $X = \{x_{ok} \in \{0, 1\} \,|\, o \in O, k \in K : (1)\text{–}(6)\}$. Пусть множество $Y = f(X)$. Возьмем задачу, в которой все допустимые моменты завершения заказов $T(x)$ $\forall x \in X$ являются целочисленными векторами. Исследуем множество Парето $P(Y)$ такой задачи.

Рассмотрим m-мерное пространство моментов завершения выполнения заказов $(T_1, \ldots, T_m)$, вектор из данного пространства будем обозначать через T. Введем следующие множества векторов $\mathbf{T}' = \{\arg\min_{x \in X} f_1(T(x))\}$, $\mathbf{T}'' = \{\arg\min_{x \in X} f_2(T(x))\}$. Если $\mathbf{T}' \cap \mathbf{T}'' \neq \varnothing$, то во множестве Парето $P(Y)$ будет единственная точка. В случае $\mathbf{T}' \cap \mathbf{T}'' = \varnothing$, возьмем максимальную компоненту по каждому вектору из $\mathbf{T}'$, далее выберем среди этих значений минимальное: $t' = \min\{\max_{i \in M} T_i \,|\, \forall T \in \mathbf{T}'\}$. Вектор T, на котором было получено значение t', даст парето-оптимальную точку в пространстве критериев (f_1, f_2). Далее, возьмем границу отрицательного ортанта в пространстве $(T_1, \ldots, T_m)$, перенесенного в точку $(t' - 1, t' - 1, \ldots, t' - 1)$. Отрицательный ортант является поверхностью уровня функции f_2. Такой ортант с вершиной в точке $(t' - 1, t' - 1, \ldots, t' - 1)$ может дать парето-оптимальную точку, у которой $f_1(T) > \min f_1$. Далее, возьмем ортант с вершиной в точке $(t' - 2, t' - 2, \ldots, t' - 2)$, который потенциально может содержать еще одну парето-оптимальную точку. Такую процедуру будем продолжать, пока не придём к ортанту с вершиной в точке $(t'', t'', \ldots, t'') \in \mathbf{T}''$. На каждой такой итерации существует не более одной парето-оптимальной точки в смысле критерия $f = (f_1, f_2)$. Таким образом, пришли к следующему утверждению.

Утверждение. *Мощность множества Парето $P(Y)$ задачи* (1)–(7) *имеет следующую оценку сверху:* $|P(Y)| \leqslant t' - \min_{x \in X} f_2 + 1$.

Возьмем задачу, в которой все переналадки s_j, $s_{jj'}$ равны между собой. Тогда в силу критерия f их можно считать равными нулю. В таком случае любая перестановка $x \in X$ дает одно и тоже значение критерия f_2, т. е. по факту задача является однокритериальной в смысле критерия f_1. Значит, множество Парето $P(Y)$ содержит одну точку. В таком оптимальном расписании производства продуктов для каждого клиента должны быть выполнены без переключения на производство продукта другого клиента, т. е.

сначала все продукты одного клиента, затем другого и так далее. Продукты должны быть упорядочены по возрастанию сумм длительностей производства заказов. Поиск такого расписания является полиномиально разрешимой задачей.

В общем случае, когда переналадки и длительности являются произвольными значениями, задача поиска какого-либо парето-оптимального решения является NP-трудной. Можно искать приближенное решение с гарантированной точностью, например, следующим образом. Пусть имеется ρ_1-приближенное решение $x^{(1)}$ для критерия f_1 и ρ_2-приближенное решение $x^{(2)}$ для критерия f_2. Рассмотрим комбинацию расписаний, когда сначала до момента $f_2(x^{(2)})$ берется часть расписания $x^{(2)}$ (заказы, которые завершают обслуживание до момента $f_2(x^{(2)})$), а завершающая часть расписания берется из $x^{(1)}$. Такая комбинация гарантирует $(2\rho_1, 2\rho_2)$-аппроксимацию парето-оптимального решения.

Исследование выполнено за счет гранта Российского научного фонда № 22-71-10015, https://rscf.ru/project/22-71-10015/.

Список литературы

[1] Hazır Ö., Günalay Y., Erel E. Customer order scheduling problem: a comparative metaheuristics study // The International Journal of Advanced Manufacturing Technology. 2008. Vol. 37. P. 589–598.

[2] Zakharova Yu., Zakharov A. Integer programming models and metaheuristics for customer order scheduling // Mathematical Optimization Theory and Operations Research: Recent Trends. MOTOR 2024. Communications in Computer and Information Science. 2024. Vol. 2239. P. 276–290.

[3] Петровский А. Б. Теория принятия решений. М. : Издательский центр «Академия», 2009. 400 с.

[4] Подиновский В. В., Ногин В. Д. Парето-оптимальные решения многокритериальных задач. М. : Физматлит, 2007. 255 с.

Вычислительная сложность задачи составления расписаний с дополнительными ограничениями на размещение операций и потребление ресурсов

Захарова Юлия Викторовна
Институт математики имени С. Л. Соболева СО РАН, Омский филиал; yzakharova@ofim.oscsbras.ru

Введение

В работе рассматривается задача составления расписаний на нескольких машинах и исследуется ее вычислительная сложность в случае дополнительных

ограничений. Здесь исследуются варианты задачи с предписаниями операций в позициях машин и учетом ресурсов невозобновимого типа. Выделяются NP-трудные частные случаи и строятся алгоритмы с гарантированными оценками точности на основе подходов списочного типа для критерия минимизации длины расписания. Устанавливаются новые свойства дополнительных ограничений, позволяющие обобщать известные ранее результаты при выполнении условий на нижнюю границу целевой функции.

Постановка задачи

Рассматривается задача составления расписаний, где независимое множество работ $J = \{j_1, \ldots, j_n\}$ планируется к выполнению на m машинах. Длительности работ зависят от потребления ресурсов по правилу

$$p_j(r_j) = \left(\frac{W_j}{r_j}\right)^{\kappa},$$

где W_j — требуемый объем работы $j \in J$, r_j — потребляемый объем ресурса работой $j \in J$, $0 < \kappa \leqslant 1$ — заданная константа. Общее ограничение на объем потребляемого ресурса обозначим через R. В качестве критерия рассматривается длина расписания, то есть момент окончания последней работы.

Также рассматривается вариант задачи, когда имеют место предписания в позициях машин на выполнение работ: X_{ik} — множество работ, которые могут выполняться в позиции k машины $i = 1, \ldots, m$.

Алгоритм для ограничений по ресурсу

Для задачи предлагается следующий алгоритм с гарантированной оценкой точности. Строим модель выпуклого программирования, обозначая переменную, отвечающую за длительность работы $j \in J$, через p_j:

$$C_{\max} \to \min, \tag{1}$$

$$p_j \leqslant C_{\max}, \ j \in J, \tag{2}$$

$$\frac{1}{m}\sum_{j \in J} p_j \leqslant C_{\max}, \tag{3}$$

$$\sum_{j \in J} W_j p_j^{-1/\kappa} \leqslant R, \tag{4}$$

$$p_j \geqslant 0, \ j \in J. \tag{5}$$

Далее, используя полученные длительности работ, строим расписание по следующему правилу:

1. Работы упорядочиваются по невозрастанию длительностей. Обозначим полученную последовательность через π.
2. Все машины запускаются на выполнение первых m работ из последовательности π.
3. Как только некоторая работа завершает свое выполнение, на соответствующей машине запускается очередная работа из последовательности π.

Утверждение 1. *Алгоритм позволяет получить* $\left(\frac{4}{3} - \frac{4}{3m}\right)$*-приближенное решение задачи с ограничением по невозобновимому ресурсу.*

Доказательство основано на адаптации подходов из работ [1, 2].

Утверждение 2. *Задача с ограничением по невозобновимому ресурсу является NP-трудной.*

Доказательство основано на сводимости задачи Разбиение [3] к случаю двух машин и общего объема ресурса равного общему объему работ.

Алгоритм для предписаний работ

Предположим, что общее число позиций, где доступна загрузка машин, совпадает с общим числом работ. Перенумеруем позиции по общему правилу увеличения номера индекса на машине и увеличения индекса машины. Пусть X_i обозначает множество работ, которые могут выполняться в позиции $i = 1, \ldots, n$.

Построим двудольный граф $G = (J_n, J, E)$. Вершины левой доли $J_n = \{1. \ldots, n\}$ соответствуют позициям, а вершины правой доли J — работам. Есть ребро между левой и правой долями, если соответствующая работа может выполняться в позиции. Пример представлен на рис. 1.

Перебор допустимых расписаний соответствует перебору совершенных паросочетаний [4] в графе $G = (J_n, J, E)$ и приводит к трудоемкости $O(n^{2.5} + n^2 n_{pm})$ с учетом времени $O(n)$ на вычисление целевой функции, где n_{pm} — число совершенных паросочетаний. Здесь время $O(n^{2.5})$ соответствует вычислению первого совершенного паросочетания, а время $O(n^2 n_{pm})$ используется для перехода от текущего совершенного паросочетания к следующему и расчета требуемых метрик.

Случай, когда все предписания содержат не более двух элементов и имеется одна машина, исследовался в работе [5].

Утверждение 3. *Задача с предписаниями работ является NP-трудной.*

Доказательство основано на сводимости задачи Упорядоченное Разбиение [3] к случаю двух машин.

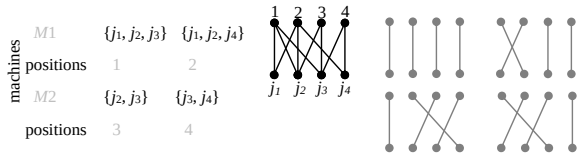

Рис. 1: Пример задачи с двумя машинами и четырьмя работами. Представлены двудольный граф $G = (J_4, J, E)$ и его совершенные паросочетания.

Исследование выполнено за счет гранта Российского научного фонда № 22-71-10015, https://rscf.ru/project/22-71-10015/.

Список литературы

[1] Graham R. L. Bounds on multiprocessing timing anomalies // SIAM Journal on Applied Mathematics. 1969. Vol. 17, no. 2. P. 416–429.

[2] Kononov A., Zakharova Yu. Minimizing makespan for parallelizable jobs with energy constraint // Siberian Electronic Mathematical Reports. 2022. Vol. 19, no. 2. P. 586–600.

[3] Garey M. R., Johnson D. S. Computers and intractability. A guide to the theory of NP-completeness. New York, NY, USA : W. H. Freeman and Company, 1979.

[4] Uno T. Algorithms for enumerating all perfect, maximum and maximal matchings in bipartite graphs // Algorithms and Computation. ISAAC 1997. Lecture Notes in Computer Science. 1997. Vol. 1350. P. 92–101.

[5] Захарова Ю. В. Точные алгоритмы для задачи составления расписаний с предписаниями работ на одной машине // Дискретные модели в теории управляющих систем : XI Международная конференция, Москва и Подмосковье, 26–29 мая 2023 г. : Труды. М. : МАКС Пресс, 2023. С. 45–50.

О предикатном определении минимальных клонов трехзначной логики

Зданович Артем Иванович
Московский государственный университет имени М. В. Ломоносова; artsemzdanovich@mail.ru

Минимальные клоны

Пусть $\mathbb{N} = \{1, 2, \dots\}$ — множество натуральных чисел, $E_k = \{0, 1, \dots, k-1\}$. Пусть $E_k^n \rightleftharpoons \{0, 1, \dots, k-1\}^n, P_k^n \rightleftharpoons \{f \mid f : E_k^n \to E_k\}, P_k \rightleftharpoons \bigcup_{n \geqslant 1} P_k^n$. Элементы P_k назовем функциями k-значной логики. Стандартным образом на

множестве P_k определяем оператор замыкания $[\,]$ относительно операций суперпозиции. Множество $M \subseteq P_k$ называется замкнутым (замкнутым классом), если $[M] = M$. Клоном называется всякое замкнутое множество функций, содержащее тождественную функцию x. Клон называется минимальным, если он не содержит никакого собственного клона, отличного от $[\{x\}]$.

Как показал Emil Post [1], в P_2 существует 7 минимальных клонов: $[\{x, 0\}]$, $[\{x, 1\}]$, $[\{\overline{x}\}]$, $[\{x \vee y\}]$, $[\{x \wedge y\}]$, $[\{x \oplus y \oplus z\}]$, $[\{xy \vee yz \vee xz\}]$. Однако отметим, что на текущий момент нет полного описания минимальных клонов P_k, при этом имеет место [2] следующая

Теорема 1 (Ivo Rosenberg)**.** *Пусть A — минимальный клон в P_k, тогда существует функция $f \in P_k^n$, такая что $A = [\{x\} \cup \{f\}]$, а также выполняется одно из следующих условий:*

1. $n = 1$.
2. $n = 2$ *и* f — *идемпотентная функция, то есть* $f(a, a) = a$ *для любого* $a \in E_k$.
3. $n = 3$ *и* f — *функция минорирования (minority function), то есть*
$$\forall a, b \in E_k : \; f(a, a, b) = f(a, b, a) = f(b, a, a) = b.$$
4. $n = 3$ *и* f — *функция голосования или мажорирования (majority function), то есть*
$$\forall a, b \in E_k : \; f(a, a, b) = f(a, b, a) = f(b, a, a) = a.$$
5. $n \in \{3, 4, \ldots, k\}$ *и* f — *полуселектор (semiprojection), то есть существует* i, *такое что для любых* $a_1, a_2, \ldots, a_n \in E_k$, *удовлетворяющих* $|\{a_1, a_2, \ldots, a_n\}| \leqslant n - 1$, *выполняется* $f(a_1, a_2, \ldots, a_n) = a_i$.

Существует только конечное количество минимальных клонов в P_k. Добавим, что с помощью компьютера удалось полностью описать минимальные клоны в случае $k = 3$, $k = 4$. Результат для $k = 3$ представлен ниже [3].

Теорема 2 (Béla Csákány)**.** *В P_3 есть ровно 84 минимальных клона. Они могут быть получены с помощью внутренних автоморфизмов в P_3 из следующих клонов:*

— $[\{c_0\}]$, $[\{c_1\}]$, $[\{c_2\}]$, $[\{c_3\}]$, *где c_0, c_1, c_2, c_3 — унарные функции, определённые следующей таблицей:*

x	c_0	c_1	c_2	c_3
0	1	1	0	1
1	1	0	1	2
2	1	2	1	0

— $[\{b_i\}]$ *для* $i \in \{1, 2, \ldots, 12\}$*, где* $b_1, b_2, \ldots, b_{12}$ — *идемпотентные бинарные функции, определённые следующей таблицей:*

x	y	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}
0	1	1	0	1	1	1	1	0	0	0	0	1	2
1	0	1	1	1	1	1	1	1	1	1	1	1	2
0	2	1	0	0	0	0	0	0	1	0	1	0	1
2	0	1	1	0	2	0	2	0	1	1	2	0	1
1	2	1	1	1	1	1	1	1	1	1	0	2	0
2	1	1	1	1	1	2	2	1	2	2	2	2	0

— $[\{m_i\}]$, $[\{s_j\}]$ *для* $i \in \{1, 2, 3\}$, $j \in \{1, 2, 3, 4, 5\}$*, где* m_1, m_2, m_3 — *функции мажорирования, определённые следующей таблицей, и* s_1, s_2, s_3, s_4, s_5 — *полуселекторы на первую переменную, определённые следующей таблицей:*

x	y	z	m_1	m_2	m_3	s_1	s_2	s_3	s_4	s_5
0	1	2	1	0	0	1	0	0	1	1
0	2	1	1	1	0	1	0	0	1	2
1	0	2	1	1	1	1	1	1	0	0
1	2	0	1	0	1	1	1	1	0	2
2	0	1	1	0	2	1	1	0	2	0
2	1	0	1	1	2	1	1	1	2	1

Предикатная описуемость

Пусть $R_k^n \rightleftharpoons \{\rho \mid \rho : E_k^n \to \{0, 1\}\}$, $R_k = \bigcup_{n \geqslant 1} R_k^n$. Назовем элементы R_k предикатами k-значной логики. Функция $f \in P_k^n$ сохраняет предикат $\rho \in R_k^h$ если для любых $(a_{i,1}, a_{i,2}, \ldots, a_{i,h}) \in R_k^h, i \in \{1, \ldots, n\}$ набор $(b_1, b_2, \ldots, b_h) \in R_k^h$, где b_i определены как $f(a_{1,i}, a_{2,i}, \ldots, a_{n,i})$.

Обозначим за $\mathrm{Pol}(\rho)$ клон, состоящий из $f \in P_k$, сохраняющих $\rho \in R_k$. Обозначим для $S \subseteq R_k$, $\mathrm{Pol}(S) = \cap_{\rho \in S} \mathrm{Pol}(\rho)$. Известно, что для любого клона $F \subseteq P_k$ существует $S \subseteq R_k$ такое, что $F = \mathrm{Pol}(S)$. Если существует конечное такое S, клон называется предикатно-описуемым.

К настоящему моменту известно, что все минимальные клоны, удовлетворяющие условиям 1), 3), 4) теоремы Розенберга, являются предикатно-описуемыми. Также была доказана предикатная описуемость и найдено предикатное задание для $[\{b_1\}]$, $[\{b_3\}]$, $[\{b_4\}]$, $[\{b_{11}\}]$, $[\{b_{12}\}]$, $[\{s_5\}]$.

Определим множество предикатов, считая $a, b, c \in E_3$:

— $\rho_{1in3}^{0,1} \rightleftharpoons \{(x_1, x_2, x_3) \in \{0, 1\}^3 \mid \exists!\, i : x_i = 1\}$;

— $\rho_{1in3,2\to 2}^{0,1} \rightleftharpoons (2, 2, 2) \cup \rho_{sel\,0,1}$;

— $\rho_{same\to a;b} \rightleftharpoons \{(x_1, x_2, x_3) : ((x_1 = x_2) \vee (x_3 = a)) \wedge (x_3 \in \{a, b\})\}$;

— $\rho_{a\leqslant b\leqslant c} \rightleftharpoons \{(a,a),(a,b),(a,c),(b,b),(b,c),(c,c)\}$;
— $\rho_{0\to a;1\to b;2\to c} \rightleftharpoons \{(0,a),(1,b),(2,c)\}$;
— $\rho_{0\to a;1\to\forall} \rightleftharpoons \{(0,a),(1,0),(1,1),(1,2)\}$;
— $\rho_{0\to a,b;1\to\forall} \rightleftharpoons \{(0,a),(0,b),(1,0),(1,1),(1,2)\}$.

Пусть $\sigma_2 \rightleftharpoons \rho_{0\to1;1\to1;2\to2}$, $\sigma_8 \rightleftharpoons \rho_{0\to2;1\to1;2\to0}$. Автор установил следующие теоремы.

Теорема 3. $[\{b_2\}] = \mathrm{Pol}(\rho^{0,1}_{1in3}, \rho_{0\to1;1\to\forall}, \rho_{0\to2;1\to\forall}, \rho_{0\to1,2;1\to\forall}, \rho_{same\to1;2}, \rho_{\sigma_2}, \{2\})$.

Теорема 4. $[\{b_7\}] = \mathrm{Pol}(\rho^{0,1}_{1in3}, \rho^{0,1}_{1in3,2\to2}, \rho_{same\to0;2}, \rho_{0\leqslant2\leqslant1}, \{2\})$.

Теорема 5. $[\{b_8\}] = \mathrm{Pol}(\rho^{0,1}_{1in3}, \rho_{0\leqslant1\leqslant2}, \rho_{\sigma_8})$.

Из данного предикатного задания, в частности, следует, что минимальные клоны $[\{b_2\}], [\{b_7\}], [\{b_8\}]$ предикатно описуемы.

Список литературы

[1] Post E. L. The two-valued iterative systems of mathematical logic // Annals of Mathematics Studies. 1941. Princeton, New Jersey : Princeton University Press.

[2] Rosenberg I. G. Minimal clones I: the five types // Lectures in universal algebra. Amsterdam : North-Holland, 1986. P. 405–427.

[3] Csákány B. All minimal clones on the three-element set // Acta Cybernetica. 1983. Vol. 6, no. 3. P. 227–238.

Эффективная реализация квантового хеширования

Зиннатуллин Илнар Гумарович, Хадиев Камиль Равилевич
Казанский (Приволжский) федеральный университет; Казанский физико-технический институт имени Е. К. Завойского ФИЦ Казанский научный центр РАН;
IlnGZinnatullin@kpfu.ru, kamilhadi@gmail.com

В данной работе рассматривается эффективная реализация квантового хеширования. Квантовое хеширование позволяет проектировать эффективные по памяти квантовые алгоритмы и строить защищенные коммуникационные протоколы. Мы предлагаем алгоритм, позволяющий балансировать между числом CNOT-гейтов (глубиной схемы) и точностью углов поворота. Современные квантовые вычислители являются устройствами NISQ (Noisy Intermediate-Scale Quantum) эры и чувствительны к точности углов.

Квантовое хеширование

Квантовое хеширование впервые было определено в [1]. В данной работе рассматриваются амплитудная [1] и фазовая [2] формы квантового хеширования. В общем случае для $x \in \mathbb{Z}_q$ n-кубитный хеш определяется как $|\psi(x)\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} |j\rangle \left(R_a\left(\theta_j\right)|q_n\rangle\right)$, где $\theta_j = \frac{4\pi s_j x}{q}$ и $S = \{s_0, \ldots, s_{d-1}\} \subseteq \mathbb{Z}_q$ — набор параметров такой, что $\frac{1}{d}\left|\sum_{j=0}^{d-1} e^{i\frac{2\pi s_j x}{q}}\right| \leqslant \varepsilon$. Заметим, что $n = \log d + 1$ и $d = O\left(\frac{\log q}{\varepsilon^2}\right)$. Для амплитудной формы $a = y$, т. е. используются повороты вокруг оси y, и $|q_n\rangle = |0\rangle$. Для фазовой формы $a = z$, т. е. используются повороты вокруг оси z, и $|q_n\rangle = |1\rangle$.

Схема для реализации квантового хеширования

Схема для реализации квантового хеширования представлена на рисунке 1. Кроме всего прочего, она состоит из n-кубитных контролируемых поворотов целевого n-го кубита вокруг оси a, в которых первые $n-1$ кубитов задействованы в качестве контролирующих. Структура схемы такова, что повороты осуществляются, используя всевозможные состояния контролирующих кубитов. Такая группа гейтов называется оператором равномерно контролируемого поворота UCR_a^{n-1} (uniformly controlled rotation). Наша задача сводится к эффективному разложению гейта UCR_a^{n-1}.

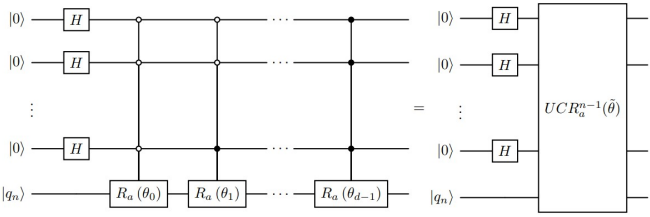

Рис. 1: Схема для реализации квантового хеширования.

Оптимизация схемы

Существует эффективная декомпозиция UCR_a^{n-1}, изложенная в работе [3]. Она может быть получена путём рекурсивного применения схемы, представленной на рисунке 2. В этом случае для декомпозиции требуется d CNOT-гейтов и d R_a-гейтов с точностью углов поворота $O\left(\frac{1}{d2^d}\right)$. Видим, что здесь фигурируют более чувствительные углы, так как изначальная точность была $O\left(\frac{1}{2^d}\right)$.

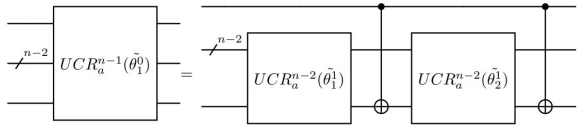

Рис. 2: Шаг рекурсивной декомпозиции UCR_a^{n-1}.

Рассмотрим вспомогательную конструкцию, которая пригодится нам дальше. Используя разложение [4, лемма 7.9], осуществляем декомпозицию контролируемых поворотов, представленную на рисунке 3. Стоит отметить, что эта схема симметрична относительно вертикальной оси. Данная декомпозиция примечательна тем, что для дальнейшего разложения контролируемых отрицаний $C^{n-2}(X)$ кубит с номером $n-1$ можно использовать в качестве анциллы. Известно [4], что в этом случае требуется $24l-52$ CNOT-гейтов, где l — число контролирующих кубитов.

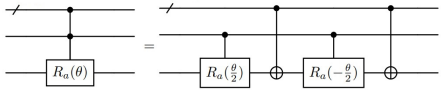

Рис. 3: Декомпозиция контролируемого поворота вокруг оси a.

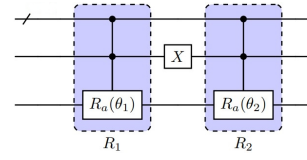

Рис. 4: Фрагмент схемы, реализующей UCR_a^{n-1}.

Нами предлагается применить к исходному гейту UCR_a^{n-1} рекурсивно k раз схему, изображенную на рисунке 2. После k итераций получаем схему, содержащую 2^k CNOT-гейтов и 2^k UCR_a^{n-k-1} гейтов. Точность углов поворота при этом возрастает до $O\left(\frac{1}{2^{d+k}}\right)$. Далее для гейта UCR_a^{n-k-1} строим разложение, в котором осуществляем перебор контролируемых поворотов, используя код Грея. Использование кода Грея удобно тем, что соседние кодовые слова отличаются ровно в одной позиции, поэтому переход из одного состояния контролирующих кубитов в другой осуществляется путём применения одного отрицания. Различающаяся позиция определяет номер кубита, к которому применяется отрицание.

Далее в получившейся схеме можно выделить 2^{n-k-2} фрагментов, изображенных на рисунке 4. Здесь мы для гейта R_1 применяем декомпозицию, представленную на рисунке 3, а для гейта R_2 зеркальное отображение этой же декомпозиции. В итоге получаем схему, которая представлена на рисунке 5. Легко заметить, что обрамленные в рамку контролируемые отрицания гасят друг друга.

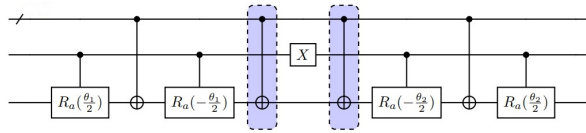

Рис. 5: Декомпозиция схемы на рис. 4.

Таким образом, общее число CNOT-гейтов равно $2^k + 2^{n-1}(24(n-k)-97)$ или $2^k + d(24(\log d - k) - 73)$. Отметим, что $k \leqslant n-5 = \log d - 4$. При увеличении k глубина схемы уменьшается от $O(\log q \log\log q)$ до $O(\log q)$, однако точность углов повышается от $O(1/q)$ до $O(1/(q\log q))$.

Исследование выполнено за счет гранта Российского научного фонда № 24-21-00406, https://rscf.ru/project/24-21-00406/.

Список литературы

[1] Ablayev F. M., Vasiliev A. V. Cryptographic quantum hashing // Laser Physics Letters. 2013. Vol. 11, no. 2. P. 753–757.

[2] Vasiliev A. Quantum hashing for finite abelian groups // Lobachevskii Journal of Mathematics. 2016. Vol. 37, no. 6. P. 753–757.

[3] Quantum circuits for general multiqubit gates / M. Möttönen, J. J. Vartiainen, V. Bergholm, M. M. Salomaa // Physical Review Letters. 2004. Vol. 93, no. 13. P. 130502.

[4] Elementary gates for quantum computation / A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, H. Weinfurter // Physical Review A. 1995. Vol. 52, no. 5. P. 3457.

Выбор оптимального корня для корневых ориентированных деревьев

Иорданский Михаил Анатольевич
Нижегородский государственный педагогический университет имени Козьмы Минина;
Нижегородский государственный университет имени Н. И. Лобачевского; iordanski@mail.ru

Постановка задачи

Пусть $t(V,E)$ — дерево, содержащее n вершин, $A = \{1, 2, \ldots, n\}$ — множество из n натуральных чисел. Взаимно однозначное отображение $\varphi : V(t) \to A$ называется *нумерацией* вершин дерева $t(V,E)$. При этом каждой вершине $v_i \in V(t)$ ставится в соответствие номер $\varphi(v_i) \in A$, каждому ребру $e = (v_i, v_j)$ — число $\Delta_e^{\varphi} = \mid \varphi(v_i) - \varphi(v_j) \mid$, а всему дереву $t(V,E)$ соответствует сумма $\Delta^{\varphi}(t) = \sum_{(v_i,v_j)\in E(t)} \mid \varphi(v_i) - \varphi(v_j) \mid$, где суммирование производится по всем ребрам дерева $t(V,E)$. Величина $\Delta^{\varphi}(t)$ задает *длину* дерева $t(V,E)$ на

нумерации φ. Любая нумерация, на которой достигается $\min_\varphi \Delta^\varphi(t) = \Delta(t)$, называется *минимальной* нумерацией дерева $t(V, E)$.

Задача построения минимальных нумераций для произвольных деревьев достаточно сложна и решается с помощью нелинейных алгоритов [1–3]. Рассматривались также её приближенные решения при различных ограничениях на множество допустимых нумераций [4, 5]. Одно из таких ограничений соответствует представлению $t(V, E)$ в виде корневых ориентированных дереьев $\overrightarrow{t}(V, E)$, в которых выделяется некоторая вершина v_k — *корень*, с номером $\varphi(v_k) = 1$, из которой все остальные вершины дерева $\overrightarrow{t}(V, E)$ достижимы по простым путям. Номера вершин мототонно растут вдоль всех путей. Такие нумерации называются *монотонными*.

В [4] был предложен алгоритм построения минимальной монотонной нумерации от заданного корня и было показано, что минимум длины дерева в классе монотонных нумераций может превосходить минимум длины в классе всех нумераций не более чем в 2 раза.

В доказательстве этой оценки использовалась минимальная монотонная нумерация дерева $\overrightarrow{t}(V, E)$ от корня, совпадающего с вершиной, имевшей первый номер при минимальной нумерации дерева $t(V, E)$. В общем случае эти вершины не всегда совпадают и возможно построение минимальной монотонной нумерации меньшей длины.

В работе рассматривается алгоритм выбора оптимального корня для дерева $\overrightarrow{t}(V, E)$, обеспечивающего его минимальную длину на множестве всевозможных корневых ориентированных деревьев, сопоставляемых дереву $t(V, E)$.

Свойства минимальных монотонных нумераций деревьев

Пусть φ минимальная монотонная нумерация дерева $\overrightarrow{t}(V, E)$. Выделим в $\overrightarrow{t}(V, E)$ путь σ_1 из вершины $\varphi^{-1}(1)$ в вершину $\varphi^{-1}(n)$. При этом из дуг, не принадлежащих пути σ_1, образуются поддеревья разложения $t_i^\sigma, i = \overline{1, k}, k < n$, представляющие собой корневые ориентированные деревья, «подвешенные» к вершинам пути σ_1 за свои корневые вершины. Применяя к ним предыдущее представление, получаем

Утверждение 1. *Любой минимальной монотонной нумерации вершин корневого ориентированного дерева соответствует его разложение на последовательность непересекающихся по дугам путей σ_i, $i = \overline{1, l}$, таких, что:*

1) *каждый путь начинается в корне того поддерева, в котором он выделяется, и заканчивается в некотором его листе;*

2) номера вершин вдоль каждого пути монотонно растут, а нумерующие последовательности всех поддеревьев разложения, образующихся в процессе выделения путей, сплошные.

Выделим в дереве $t(V, E)$ произвольную вершину $v_i \in V$ степени $s(v_i) = p$ и все инцидентные ей ребра (v_i, v_r), $r = \overline{1, p}$. Любая вершина $v_q \neq v_i$ связана с вершиной v_i единственной цепью. Будем говорить, что вершина v_q принадлежит *ветке*, соединенной с вершиной v_i по ребру (v_i, v_r), $r \in \overline{1, p}$, если в v_q можно попасть из v_i по цепи, содержащей ребро (v_i, v_r). Число вершин в ветке определяет её *вес*. В корневом ориентированном дереве ветки делятся на *выходящие*, в них ведут дуги (v_i, v_r), $r \in \overline{1, p}$, из вершины v_i, и одну *входящую*, из вершины v_r которой можно попасть в вершину v_i по дуге (v_r, v_i), $r \in \overline{1, p}$. Справедлива [6]

Теорема 1. *Монотонная нумерация произвольного корневого ориентированного дерева является минимальной тогда и только тогда, когда пути σ_i, $i = \overline{1, l}$, выходят из вершин дерева по дугам, ведущим в ветки с наибольшим весом.*

Рассмотрим *обход в глубину* вершин корневого ориентированного дерева, начиная с корня, в соответствии с ориентацией дуг. Перемещение по каждому пути заканчивается в листе. Затем осуществляется возврат к предыдущей вершине, из которой исходит хотя бы одна ещё не пройденная дуга, по которой продолжается обход вершин до тех пор, пока не вернемся в корень, все исходящие дуги которого пройдены.

При использовани обхода вершин дерева в глубину из теоремы 1 получаем краткую формулировку алгоритма построения минимальной монотонной нумерации коневого ориентированного дерева от заданного корня.

Следствие 1. *Нумерация корневого ориентированного дерева минимальна тогда и только тогда, когда она проводится в соответствии с обходом в глубину вершин дерева, начиная от корня, в порядке неубывания весов веток.*

В каждом дереве $t(V, E)$ можно выделить вершину (две смежные вершины), веса всех веток к которой (которым) не превосходят половины от общего числа вершин. Эти вершины образуют *центроид* дерева. Входящие в него вершины называются *центроидными*.

Следствие 2. *При минимальной монотонной нумерации от любого корня все ветки к центроидной вершине имеют сплошные нумерующие последовательности.*

Следствие 3. *Оптимальный корень принадлежит одной из двух веток к центроидной вершине с наибольшим весом.*

Алгоритм выбора оптимального корня

1. Выделить в дереве центроидную вершину.
2. Выбрать одну из двух веток к центроидной вершине с наибольшим весом.
3. В выбранной ветке построить минимальную монотонную нумерацию с корнем в центроидной вершине.
4. Вершину, получившую наибольший номер, взять в качестве оптимального корня.

Обоснование алгоритма

Рассмотрим минимальную монотонную нумерацию ориентированного дерева с корнем в вершине, выбранной по алгоритму. Разобьем дуги дерева на три непересекающихся подмножества: дуги ветки, содержащей корень, дуги, исходящие из центроидной вершины, и внутренние дуги веток, исходящих из центроидной вершины.

Длина ветки, содержащей корень, минимальна, так как она равна её длине при минимальной монотонной нумерации от центроидного корня. Это так, поскольку, учитывая теорему 1, у обеих нумераций совпадают цепи первых путей разложения и, следовательно, все поддеревья разложения.

Сумма длин дуг, исходящих из центроидной вершины, минимальна, учитыва следствия 2 и 3. Внутренние дуги веток, исходящих из центроидной вершины, разобьем на непересекающиеся подможества дуг, относящихся к разным веткам, выходящим из центроидной вершины. Минимальность длины каждой такой ветки следует из сплошности нумерующей последовательности (следствие 2) и того, что по следствию 1 на каждой из них строится минимальная монотонная нумерация с корнем в вершине, смежной с центроидной вершиной.

Трудоемкость алгоритма не превосходит по порядку $O(n \log n)$.

Список литературы

[1] Гольдберг М. К., Клипкер И. А. Алгоритм минимальной нумерации вершин дерева // Сообщения Академии наук Грузинской ССР. 1976. Т. 81, № 3. С. 553–556.

[2] Shiloach Y. A minimum linear arrangement algorithm for undirected trees // SIAM Journal on Computing. 1979. Vol. 8, no. 1. P. 15–32.

[3] Chung Fan Rong King. On optimal linear arrangements of trees // Computers & Mathematics with Applications. 1984. Vol. 10, no. 1. P. 43–60.

[4] Шейдвассер М. А. О длине и ширине размещений графов в решетках // Проблемы кибернетики. М. : Наука, 1974. Вып. 29. С. 63–102.

[5] Иорданский М. А. Минимальные плоские размещения деревьев // Методы дискретного анализа в решении экстремальных задач. Вып. 33. Новосибирск : Институт математики СО АН СССР, 1979. С. 3–30.

[6] Иорданский М. А. Кодирование комбинаторных объектов. СПб. : Лань, 2018. 92 с.

Об эффективных алгоритмах в задаче о максиминных путях

Каймаков Кирилл Владимирович

Национальный исследовательский университет «Высшая школа экономики»;
kirill.kaymakov@mail.ru

Задача о максиминном пути (кратко, ЗМП) — для заданных связного графа $G = (V, E)$, пропускных способностей его ребер $c : E \longrightarrow \mathbb{R}_{\geqslant 0}$ и вершин $s, t \in V$ найти $b(s,t) = \max\limits_{P \in \mathcal{P}_{st}} \min\limits_{e \in P} c(e)$, где $\mathcal{P}_{st}$ обозначает множество всех путей между s и t.

ЗМП возникает в качестве отдельной подзадачи в алгоритме Эдмондса и Карпа [1] для вычисления максимального потока в сети, а также в алгоритме решения задачи о k-расщепляемом потоке [2]. Задача о максимальном потоке в сети возникает, например, в математических моделях передачи электроэнергии, планирования полетов, сетях связи. Вариант ЗМП, когда фиксируется s, а t пробегает все множество вершин, возникает в качестве подзадачи в планировании расписания движения железнодорожного транспорта, см., например, [3].

ЗМП для графа с n вершинами и m ребрами может быть решена алгоритмом Камерини [4] за время $O(m)$. В этой работе автором рассматриваются две разновидности задачи ЗМП и предлагаются эффективные алгоритмы для их решения. В *онлайновой задаче о максиминном пути* (кратко, ОЗМП) требуется так предобработать заранее известный граф (G, c), чтобы по результату предобработки вычислять $b(s,t)$ для любых задаваемых $s, t \in V$. В *задаче о k максиминных путях* (кратко, k-ЗМП) для заданных связного графа $G = (V, E)$, пропускных способностей его ребер $c : E \longrightarrow \mathbb{R}_{\geqslant 0}$ и вершин $s_1, t_1, s_2, t_2, \ldots, s_k, t_k \in V$ требуется найти $b(s_1, t_1), \ldots, b(s_k, t_k)$. Нам понадобится следующая хорошо известная связь между оптимальными решениями ЗМП и задачи о максимальном остовном дереве:

Утверждение 1. *Пусть T — произвольное максимальное остовное дерево графа $(G = (V, E), c)$. Тогда, для любых $s \in V, t \in V$ минимальная из пропускных способностей ребер на st-пути дерева T равна $b(s,t)$.*

Для решения онлайновой версии ЗМП воспользуемся модификацией jump-pointers алгоритма [5], используемого для поиска наибольшего общего предка

двух задаваемых вершин корневого дерева. В каждой вершине дерева будем хранить, кроме вершин для прыжков, минимальное значение на этом прыжке. Таким образом, во время подъема по дереву мы сможем агрегировать значение минимума на нашем пути.

Наш алгоритм будет выглядеть следующим образом (подчеркиванием выделена предобработка):

Шаг 1. Найти максимальное остовное дерево T графа (G, c) алгоритмом Прима с использованием Фибоначчиевых куч.
Шаг 2. Применить к T модифицированный jump-pointers алгоритм.
Шаг 3. Вычислить $b(s, t)$ и вернуть его значение.

Корректность алгоритма следует из утверждения 1. Сложность Шага 1 есть $O(m + n \log n)$, сложность Шага 2 есть $O(n \log n)$, а сложность Шага 3 есть $O(\log n)$. Поэтому итоговая сложность есть $O(m + n \log n)$.

Для эффективного решения задачи k-ЗМП воспользуемся структурой данных «система непересекающихся множеств» (кратко, СНМ) для хранения разбиения множества на подмножества и поддержки операций на подмножествах (см., например, книгу [6]). Этот алгоритм использует следующие обозначения:

— $ind[v]$ — множество тех i, что s_i или t_i принадлежит элементу СНМ, который содержит v;
— $Find(v)$ — поиск канонического элемента подмножества, которое содержит v;
— $Join(x, y)$ — замена двух подмножеств, имеющих канонические элементы x и y, их объединением и назначение x в качестве канонического элемента нового подмножества, а также транспонирование x и y в $ind[x]$ и $ind[y]$, если $\big|ind[y]\big| > \big|ind[x]\big|$;
— $answer[i] = b(s_i, t_i)$ для любого $i \in [k]$.

Для заданных (G, c) и $(s_1, t_1), \ldots, (s_k, t_k)$ наш алгоритм выглядит следующим образом:

Шаг 1. Инициализировать n-элементный массив ind пустыми множествами.
Шаг 2. Инициализировать СНМ n синглетонами, каждый из которых соответствует вершине G.
Шаг 3. Для каждого $i \in [k]$ выполнить:
$b(s_i, t_i) := -\infty, ind[s_i] := ind[s_i] \cup \{i\}, ind[t_i] := ind[t_i] \cup \{i\}$.
Шаг 4. Упорядочить E по невозрастанию их пропускных способностей.
Шаг 5. В цикле по $e = ab \in E$:
Шаг 5.1. Вычислить $x := Find(a), y := Find(b)$.

Шаг 5.2. Если $x \neq y$, то:
Шаг 5.2.1. Выполнить $Join(x, y)$.
Шаг 5.2.2. В цикле по $z \in ind[x] \cap ind[y]$ присвоить $answer[z] := c(e)$.
Шаг 5.3. Присвоить $ind[x] := ind[x] \otimes ind[y]$.
Шаг 6. Вернуть $answer[]$.

Корректность предложенного алгоритма следует из утверждения 1 и алгоритма Краскалла поиска максимального остовного дерева. Действительно, в соответствии с ними значение $b(s_i, t_i)$ определяется именно в тот момент, когда возникает путь между s_i и t_i в частичном оптимальном решении. Иными словами, когда e соединяет вершины a и b из разных компонент связности, a, s_i принадлежат одной из них и b, t_i принадлежат другой. Следовательно, $i \in ind[x]$ и $i \in ind[y]$. В этот момент $answer[i] = c(e) = b(s_i, t_i)$ и это же верно для всего множества $ind[x] \cap ind[y]$. Присваивание $ind[x] := ind[x] \otimes ind[y]$ гарантирует, что подмножество T_x с каноническим элементом x будет хранить только те s_i или t_i, что $s_i \in T_x$, $t_i \notin T_x$ или $s_i \notin T_x$, $t_i \in T_x$.

Можно показать, что предположении того, что $ind[]$ хранится хэш-множеством, ожидаемое время работы предложенного алгоритма есть

$$O(m + (n + k) \log n).$$

Автор выражает благодарность своему научному руководителю, д.ф.-м.н., проф. Малышеву Д. С. за постоянное внимание к работе, полезные советы и замечания.

Список литературы

[1] Edmonds J., Karp R. Theoretical improvements in algorithmic efficiency for network flow problems // Journal of the ACM. 1972. Vol. 19, no. 2. P. 284–264.

[2] Baier G., Köhler E., Skutella M. On the k-splittable flow problem // Algorithms — ESA 2002. ESA 2002. Lecture Notes in Computer Science. 2002. Vol. 2461. P. 101–113.

[3] Railway timetabling: a maximum bottleneck path algorithm for finding an additional train path / F. Ljunggren, K. Persson, A. Peterson, C. Schmidt // Public Transport. 2021. Vol. 13. P. 597–623.

[4] Camerini P. The min-max spanning tree problem and some extensions // Information Processing Letters. 1978. Vol. 7, no. 1. P. 10–14.

[5] Bender M., Farach-Colton M. The level ancestor problem simplified // Theoretical Computer Science. 2004. Vol. 321, no. 1. P. 5–12.

[6] Tarjan R. E. Data structures and network agorithms. Philadelphia, PA : Society for Industrial and Applied Mathematics, 1983. 131 p.

О спектре бумеранговой равномерности квадратичных подстановок

Калинин Юрий Сергеевич
ООО «Центр сертификационных исследований»; shzikarev1703@gmail.com

Одним из разностных методов атаки на блочные шифры является метод бумеранга, впервые опубликованный в работе [1]. Его преимущество заключаются в том, что даже при наличии невысокой дифференциальной равномерности, шифр всё равно может быть уязвим для разностных атак. В работе [1] рассматривались шифрсистема COCONUT98 и другие блочные шифры, а позднее метод был применен к 5-раундовому и 6-раундовому AES и другим шифрам. Недавно была предложена атака методом бумеранга на 4-раундовый алгоритм шифрования LILLIPUT-TBC-II-256 [2]. В настоящей работе исследуются свойства квадратичных подстановок относительно такого параметра, как бумеранговая равномерность, который характеризует стойкость функции к данному методу. Приведены оценки её значений при различных условиях. Представлен экспериментально полученный класс триномиальных квадратичных подстановок, описаны его характеристики.

Основные определения и обозначения

Все упомянутые, но не отмеченные определения и обозначения можно уточнить в работе [3].

Пусть $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ — векторная булева функция, заданная многочленом над полем $\mathbb{F}_{2^n}$ из 2^n элементов, «+» — операция сложения в поле.

Определение 1. *Дифференциальной характеристикой векторной булевой функции F называется величина, определенная следующим образом:*

$$\delta_F(a,b) = \left|\{x \in \mathbb{F}_{2^n} : F(x) + F(x+a) = b\}\right|,$$

где a, b — элементы поля $\mathbb{F}_{2^n}$. Дифференциальной равномерностью, соответственно, величина

$$\delta_F = \max_{a,b \in \mathbb{F}_{2^n}^*} \delta_F(a,b).$$

Пусть F — подстановка на $\mathbb{F}_{2^n}$.

Определение 2. *Бумеранговой характеристикой называется величина, заданная следующим образом:*

$$\beta_F(a,b) = \left|\left\{x \in \mathbb{F}_{2^n} \mid F^{-1}(F(x)+b) + F^{-1}(F(x+a)+b) = a\right\}\right|,$$

где a, b — элементы поля $\mathbb{F}_{2^n}$. Бумеранговая равномерность для функции F есть максимальное значение из $\beta_F(a,b)$, за исключением случаев $a=0$ или $b=0$:

$$\beta_F = \max_{a,b\in\mathbb{F}_{2^n}^*} \beta_F(a,b).$$

Для любых $a, b \in \mathbb{F}_{2^n}$ и подстановки F справедливо

$$\beta_F(a,b) \geqslant \delta_F(a,b).$$

Другие свойства бумеранговой равномерности хорошо описаны в статье [4].

Квадратичную функцию, представленную полиномом над $\mathbb{F}_{2^n}$, без линейной и константной частей будем называть *однородной квадратичной функцией*. Пусть $\overline{\alpha} = (\alpha_1, \ldots, \alpha_n)$ — базис $\mathbb{F}_{2^n}$ над $\mathbb{F}_2$, обозначим $\overrightarrow{x} = (x_1, \ldots, x_n)$ — разложение $x \in \mathbb{F}_{2^n}$ по базису $\overline{\alpha}$. Положим, $M_\alpha \in \mathbb{F}_{2^n}^{n\times n}$ — такая матрица, что $M_\alpha[i,j] = \alpha_j^{2^{i-1}}$ для $1 \leqslant i, j \leqslant n$. Тогда каждой однородной квадратичной функции

$$F(x) = \sum_{1\leqslant j<i\leqslant n} c_{ij} x^{2^{i-1}+2^{j-1}} \in \mathbb{F}_{2^n}[x]$$

соответствует матрица $H = M^T C_F M$, где $M = M_\alpha$, C_F — матрица $n \times n$ такая, что $C_F[j,i] = C_F[i,j] = c_{ij}$ для $1 \leqslant j < i \leqslant n$ и $C_F[i,i] = 0$ для $i \in \overline{1,n}$.

Результаты

В работе [3] исследовалась взаимосвязь показателя бумеранговой равномерности со значениями дифференциальной характеристики. Было показано, что если $F + A$ — квадратичная подстановка, A — её аффинная часть, то

$$\beta_{F+A}(a,b) = \delta_{F+A}(a,b) \; + \sum_{z\in U^*_{a,F(a)}(F)} \delta_{F+A}(z,b), \tag{1}$$

где множество $U^*_{a,b}(F)$ определено следующим образом:

$$U_{a,b}(F) = \{x \in \mathbb{F}_{2^n} : F(x) + F(x+a) = b\}, \quad U^*_{a,b}(F) = U_{a,b}(F)\backslash\{0, a\}.$$

С использованием формулы (1) были получены следующие результаты.

Утверждение 1. *Пусть F — однородная квадратичная функция, A — аффинная, тогда для подстановки $F + A$ и для любого $a \in \mathbb{F}_{2^n}^*$ такого, что $\delta_F(a, F(a)) = \delta_{F+A}$, выполнено:*

$$\beta_{F+A}(a, F(b) + L(b)) \geqslant \delta_{F+A} + \delta_{F+A}(a, F(b) + L(b)),$$

где $b \in U^*_{a,b}(F)$, L — линейная часть A. При этом

$$\beta_{F+A}(a, F(b) + L(b)) \geqslant \delta_{F+A} + \delta_{F+A}(a, F(b) + L(b)) + 2^k,$$

если $rank(\Lambda, \gamma^\downarrow) = rank(\Lambda) = n - k$, где Λ и $\gamma^\downarrow$ определены соотношениями: $\overline{\alpha} \cdot \Lambda = (\overrightarrow{a} + \overrightarrow{b}) \cdot H^T$ и $\overline{\alpha} \cdot \gamma^\downarrow = F(a) + L(a)$, $\overline{\alpha}$ — базис $\mathbb{F}_{2^n}$.

Оценка значений показателя бумеранговой равномерности была приведена в работе [3], далее представлены достаточные условия для её улучшения.

Утверждение 2. *Пусть F — квадратичная подстановка, $\delta_F = 2^k$. Если существуют такие $a, b \in \mathbb{F}_{2^n}^*$, что $\delta_F(a,b) < \delta_F$ и $\beta_F(a,b) = \beta_F$, то справедлива верхняя оценка:*

$$\beta_F \leqslant \beta, \text{ где } \beta = \begin{cases} 2^{2k} - 2^{k+1}, \text{ при } \delta_F(a,b) = 0 \text{ и } \delta_F(a, F(a)) = \delta_F; \\ 2^{2k-t} - 2^{k+1} + 2^{k-t}, \text{ при } \delta_F(a,b) = \frac{\delta_F}{2^t}, \ t \in \overline{1, k-1}. \end{cases}$$

Также исследовались некоторые конструкции квадратичных подстановок, а именно триномиальные однородные квадратичные подстановки. Был получен следующий класс функций:

Утверждение 3. *Пусть $n > 1$ — натуральное и нечетное число, α — примитивный элемент поля $\mathbb{F}_{2^n}$, заданы параметры $t \in \overline{1, n}$, $\mathrm{wt}_2(t) = 2$, $0 \leqslant k_1 < k_2 < k_3 \leqslant n-1$, $m \in \overline{0, 2^n - 2}$ и функция $C(m, t, k_1, k_2, k_3)$ принимает множество из $\frac{2^n+1}{3}$ значений в поле $\mathbb{F}_{2^n}$ при фиксированных параметрах. Тогда функции вида*

$$x^{2^{k_3}t} + \alpha^{2m+1} \cdot x^{2^{k_2}t} + \alpha^{3m} \cdot C(m, t, k_1, k_2, k_3) \cdot x^{2^{k_1}t}$$

являются триномиальными квадратичными однородными подстановками и APN- и AB-отображениями и имеют следующие характеристики:

$\deg F$	$\deg F^{-1}$	δ_F	β_F	Nl_F
2	$\frac{n+1}{2}$	2	2	$2^{n-1} - 2^{\frac{n-1}{2}}$

Для дальнейшего исследования можно поставить следующие вопросы:

1. Когда для квадратичной подстановки F существуют такие $a, b \in \mathbb{F}_{2^n}^*$, что
   - $\delta_F(a,b) < \delta_F$ и $\beta_F(a,b) = \beta_F$;
   - $\delta_F(a,b) = \delta_F$ и $\beta_F(a,b) = \beta_F$?
2. При каких условиях на квадратичную подстановку F выполнено $\delta_F = \beta_F$ и $\delta_F \neq 2$?

Список литературы

[1] Wagner D. The boomerang attack // Fast Software Encryption. FSE 1999. Lecture Notes in Computer Science. 1999. Vol. 1636. P. 156–170.

[2] Пудовкина М. А., Смирнов А. М. Анализ методом бумеранга 4-раундового алгоритма шифрования LILLIPUT-TBC-II-256 // Прикладная дискретная математика. Приложение. 2023. № 16. С. 81–84.

[3] Калинин Ю. С. Исследование бумеранговой равномерности квадратичных взаимно однозначных векторных булевых функций // Прикладная дискретная математика. Приложение. 2024. № 17. С. 28–34.

[4] Boura C., Canteaut A. On the boomerang uniformity of cryptographic S-boxes // IACR Transactions on Symmetric Cryptology. 2018. Vol. 2018, no. 3. P. 290–310.

О структурных графах теории механизмов

Ковалёв Михаил Дмитриевич
Московский государственный университет имени М. В. Ломоносова,
механико-математический факультет, кафедра дискретной математики; mkov@rambler.ru

В теории механизмов в настоящее время для описания строения механизмов, как правило, используют граф $\mathcal{G}$ [1–3]. Его получают, сопоставляя звеньям шарнирного механизма вершины графа $\mathcal{G}$, а соединяющим звенья между собой кинематическим парам — рёбра графа $\mathcal{G}$. Однако, как было показано в работе [4], описание строения механизма графом $\mathcal{G}$ не всегда адекватно и удобно. Чтобы проанализировать этот вопрос, достаточно рассмотреть какую-либо определённую математическую модель механизмов.

Мы ограничимся рассмотрением модели плоских шарнирно-рычажных механизмов [5, 6]. Под ними мы понимаем конструкции, составленные из прямолинейных жёстких стержней (*рычагов*), соединённых шарнирами в их концах. У нас каждый рычаг несёт по *шарниру* на своих концах. При этом если в шарнире соединены лишь два рычага, то этому шарниру отвечает обычная вращательная пара, допускающая произвольное проворачивание в плоскости одного из рычагов относительно другого. Будем называть такой шарнир 1-*шарниром*. Если в шарнире соединены $k > 2$ рычагов, то это так называемый совмещённый или сложный шарнир с одним общим центром вращения для всех k рычагов. Его мы назовём $(k-1)$-*шарниром*. В этом шарнире каждый из k рычагов допускает проворачивание, независимое от остальных рычагов. Если же конец рычага не соединён ни с каким другим рычагом, то в нём нет кинематической пары, и мы его называем 0-*шарниром*. До сих пор мы говорили о незакреплённых конструкциях. В теории механизмов обычно рассматривают закреплённые в плоскости (стойке) конструкции. Закрепление будем производить шарнирами, которые назовём закреплёнными, и будем обозначать крестиками — в отличие от кружочков, отвечающих свободным (незакреплённым) шарнирам (рис. 1). В закреплённом шарнире непременно имеется хотя бы одна кинематическая пара, и он не может быть 0-шарниром.

В этой модели естественно использовать граф G, вершины которого отвечают шарнирам, а рёбра — рычагам конструкции. Конфигурационное пространство механизма является компонентой связности положительной размерности множества решений системы уравнений, накладывающих условия на расстояния между шарнирами и сразу выписывающихся по графу G:

$$(x_i - x_j)^2 + (y_i - y_j)^2 = d_{ij},$$

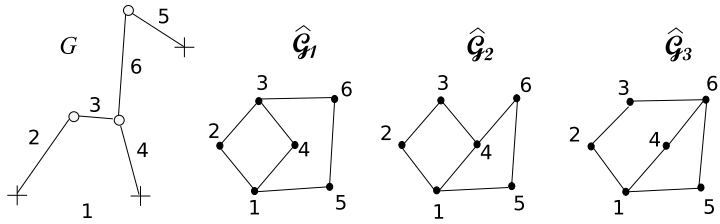

Рис. 1: Звенья механизма обозначены цифрами, стойка — цифрой 1. Граф G конструкции с 2-шарниром не восстанавливается ни по одному из отвечающих ему трёх графов $\hat{\mathcal{G}}_i$, изображённых справа, взятому в отдельности.

где (x_i, y_i) — координаты i-го шарнира p_i, а d_{ij} — квадрат длины рычага, несущего на концах шарниры p_i и p_j. Число уравнений равно числу r рычагов, а число неизвестных — удвоенному числу незакреплённых шарниров.

При наличии совмещённых шарниров по графу $\mathcal{G}$ невозможно восстановить строение механизма. В некоторых источниках, например, в [2], предлагалось в этом случае в качестве структурного графа использовать граф $\hat{\mathcal{G}}$, отличающийся от графа $\mathcal{G}$ при наличии совмещённого k-кратного ($k > 1$) шарнира тем, что из $\mathcal{G}$ удалялось определённое число рёбер, входящих в отвечающий этому совмещённому шарниру полный подграф $K_{k+1} \subset \mathcal{G}$. А именно, из полного графа K_{k+1} удалялось $\dfrac{k(k-1)}{2}$ рёбер. Скажем, при наличии 2-шарнира из подграфа K_3 удалялось одно из рёбер (рис. 1). Такой структурный граф $\hat{\mathcal{G}}$ строится по конструкции неоднозначно, на рисунке 1 тремя способами. Да и структуру механизма восстановить по нему невозможно.

Чтобы не терять информацию при наличии совмещённых шарниров, достаточно сопоставить конструкции взвешенный граф $\mathcal{G}^*$. Взвешенный граф $\mathcal{G}^*$ — это граф $\mathcal{G}$, каждому отвечающему k-кратному шарниру ($k > 0$) ребру которого приписан вес k, где $k = 1, 2, \ldots$. Можно, изображая взвешенный граф $\mathcal{G}^*$, писать на его рёбрах вес $k = 2, 3, \ldots$ и не записывать вес, равный 1. Тогда если механизм содержит лишь 1-шарниры, то его взвешенный граф $\mathcal{G}^*$ совпадает с обычным графом $\mathcal{G}$. Однако, как показано в работе [4], и в этом случае удобнее пользоваться графом G.

Таким образом, использование графа G в модели плоских шарнирно-рычажных механизмов предпочтительнее использования традиционных графов $\mathcal{G}$ и $\hat{\mathcal{G}}$. Для описания структуры в других моделях плоских механизмов с вращательными парами целесообразно перейти к эквивалентному шарнирно-рычажному механизму, и использовать его граф G [7].

Список литературы

[1] Dong Kaijie, Li Duanling, Kong Xianwen. Representation of planar kinematic chains with multiple joints based on a modified graph and isomorphism identification // Mechanism and Machine Theory. 2022. Vol. 172. P. 104793.

[2] Пейсах Э. Е., Нестеров В. А. Система проектирования плоских рычажных механизмов. М. : Машиностроение. 1988.

[3] Диденко Е. В. Разработка и анализ плоских многоконтурных механизмов на основе теории графов : специальность 05.02.18 «Теория механизмов и машин» : диссертация на соискание ученой степени кандидата технических наук ; Институт машиноведения им. А. А. Благонравова РАН. Москва, 2019. 125 с.

[4] Ковалёв М. Д. О структурных графах теории механизмов // Проблемы машиностроения и надёжности машин. 2023. № 2. С. 44–49.

[5] Ковалёв М. Д. Геометрическая теория шарнирных устройств // Известия Российской академии наук. Серия математическая. 1994. Т. 58, № 1. С. 45–70.

[6] Ковалёв М. Д. Геометрические вопросы кинематики и статики. М. : Ленанд, URSS, 2019.

[7] Ковалёв М. Д. О графах и структурных формулах теории механизмов // Сибирский журнал индустриальной математики. 2023. Т. 26, № 3. С. 42–55.

О бесквадратных свойствах формальных слов специального вида

Колпаков Роман Максимович
Московский государственный университет имени М. В. Ломоносова;
Федеральный исследовательский центр «Информатика и управление» РАН;
foroman@mail.ru, roman.kolpakov@math.msu.ru

Пусть $w = w[1]w[2]\dots w[n]$ — произвольное формальное слово длины n, обозначаемой через $|w|$. Фрагмент $w[i]w[i+1]\dots w[j]$ слова w, где $1 \leqslant i \leqslant j \leqslant n$, называется *фактором* слова w и обозначается через $w[i..j]$. Два фактора $w[i'..j']$ и $w[i''..j'']$ такие, что $i' \leqslant i''$, будем называть *смыкающимися*, если $i'' \leqslant j' + 1$. Под *пересечением* данных факторов понимается фактор $w[i''..j']$ (если $i'' = j' + 1$, то пересечение данных факторов полагается пустым). Если некоторое слово u совпадает с некоторым фактором v слова w, то v называется *вхождением* слова u в w. Два фактора в слове полагаются равными, если они являются вхождениями одного и того же слова. Мы обозначаем через $p(w)$ минимальный период слова w и через $e(w)$ отношение $|w|/p(w)$, которое называется *порядком* слова w. Слово называется *периодическим*, если его порядок не меньше, чем 2. Вхождения периодических слов в некотором слове называются *периодичностями* в этом слове.

Самым простым и наиболее известным примером периодичностей являются факторы вида uu, где u — непустое слово. Такие факторы называются *квадратами*. Под *периодом* квадрата uu понимается длина слова u. Квадраты в словах являются классическим объектом исследований в словарной комбинаторике [1] и используются в многочисленных комбинаторных алгоритмах на словах [2]. Поэтому представляет большой интерес как с комбинаторной, так и с алгоритмической точки зрения задача определения всех квадратов в заданных словах специального вида.

Пусть r — некоторая периодичность. Любой фактор длины $p(r)$ в r называется *циклическим корнем* периодичности r. Две периодичности с одинаковым минимальным периодом называются *периодичностями с одинаковыми корнями*, если они имеют одинаковые множества различных циклических корней.

Периодичность в некотором слове называется *максимальной*, если эта периодичность не может быть расширена в этом слове ни на один символ ни вправо, ни влево с сохранением ее минимального периода. Более строго, периодичность $r \equiv w[i..j]$ в w называется *максимальной*, если она удовлетворяет следующим условиям:

1. Если $i > 1$, то $w[i-1] \neq w[i-1+p(r)]$.
2. Если $j < n$, то $w[j+1-p(r)] \neq w[j+1]$.

Множество всех максимальных периодичностей в слове является компактным представлением всех периодичностей в слове, имеющим много полезных приложений (см., например, [3]). Для максимальных периодичностей с одинаковыми корнями имеют место следующие факты (см., например, [4]).

Утверждение 1. *Длина пересечения двух смыкающихся максимальных периодичностей с одинаковыми корнями меньше минимального периода данных периодичностей.*

Утверждение 2. *Пусть r', r'' — две смыкающиеся максимальные периодичности с одинаковыми корнями и минимальным периодом p. Тогда существует целое σ такое, что $0 < \sigma < p$ и для любых равных циклических корней $w[i'..i'+p-1]$, $w[i''..i''+p-1]$ периодичностей r' и r'' соответственно выполняется $i'' - i' \equiv \sigma \pmod{p}$.*

Мы обозначаем число σ, удовлетворяющее утверждению 2, через $\sigma(r', r'')$.

Заметим, что в максимальной периодичности r с минимальным периодом $p(r)$ любой фактор длины $2kp(r)$, где $k \in \mathbb{N}$, являются квадратами с периодом $kp(r)$. Мы называем такие квадраты *порожденными периодичностью* r. Очевидно, что все квадраты, порожденные любой максимальной периодичностью, могут быть легко определены и описаны.

Пусть для слова w найдутся i, j, где $1 \leqslant i \leqslant j+1 \leqslant n$, такие, что факторы $w[1..j]$ и $w[i..n]$ являются максимальными периодичностями с одинаковыми корнями, т. е. w состоит из двух смыкающихся максимальных периодичностей с одинаковыми корнями. В таком случае мы будем говорить, что слово w образовано смыкающимися периодичностями $r' = w[1..j]$ и $r'' = w[i..n]$ с одинаковыми корнями. Положим $p' = p(r') = p(r'')$ и заметим, что в данном слове w любой фактор $w[s..s+2p-1]$ длины $2p$, где $p = \sigma(r', r'')+kp'$ для некоторого $k \in \mathbb{N}$, такой, что $i \leqslant s+p \leqslant j+1$, является квадратом. Мы называем такие квадраты *порожденными парой смыкающихся периодичностей* r', r''. Очевидно, что все квадраты, порожденные любой парой смыкающихся периодичностей, могут быть легко определены и описаны. Основным результатом данной работы является следующее утверждение.

Теорема 1. *Пусть слово w образовано некоторыми смыкающимися периодичностями r' и r'' с одинаковыми корнями, $p(r') = p(r'') = p'$. Тогда в слове w любой квадрат с периодом не меньшим, чем p', является либо квадратом, порожденным периодичностями r' или r'', либо квадратом, порожденным парой смыкающихся периодичностей r', r''.*

Таким образом, в теореме 1 получено полное описание всех квадратов с относительно большим периодом в словах, образованных смыкающимися периодичностями с одинаковыми корнями.

Работа выполнена при финансовой поддержке Минобрнауки России в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2022-284.

Список литературы

[1] Crochemore M., Rytter W. Squares, cubes, and time-space efficient string searching // Algorithmica. 1995. Vol. 13. P. 405–425.

[2] Крошемор М., Лекрок Т., Риттер В. Алгоритмы обработки текста. 125 задач с решениями. М. : ДМК-Пресс, 2021. 312 с.

[3] Extracting powers and periods in a string from its runs structure / M. Crochemore, C. Iliopoulos, M. Kubica, J. Radoszewski, W. Rytter, T. Walen // String Processing and Information Retrieval. SPIRE 2010. Lecture Notes in Computer Science. 2010. Vol. 6393. P. 258–269.

[4] Kolpakov R. On primary and secondary repetitions in words // Theoretical computer science. 2012. Vol. 418. P. 71–81.

Об инвариантах 5-конфигураций

Комягин Максим Михайлович
НКО «Фонд содействия развитию безопасных информационных технологий»; komyagin.mm@mail.ru

Следуя работе [1], рассматриваем конфигурации как совокупности из v подмножеств одинаковой мощности $k < v$ в множестве мощности v, для которых матрица инциденций имеет специальный вид.

Определение 1. *Матрицу $L \in GL(v, 2)$ называем (v, k)-матрицей, если у неё и у матрицы L^{-1} в каждой строке и каждом столбце k единиц и $v - k$ нулей.*

Для любых подстановочных матриц $P, Q \in GL(v, 2)$ матрица PLQ, получающаяся из L независимыми перестановками строк и столбцов, — тоже (v, k)-матрица. Матрицы PLQ и L будем называть *комбинаторно эквивалентными.*

Сумму подмножеств $A, B \subseteq X$ множества X мощности v определим равенством $A + B = (A \cup B) \setminus (A \cap B)$. Тогда множество 2^X всех подмножеств множества X с заданной на нём операцией сложения становится элементарной абелевой группой, изоморфной группе $GF(2)^v$ по сложению. Иногда для рассматриваемых конфигураций удобнее пользоваться эквивалентным определением.

Определение 2. *Совокупность $\mathcal{X} \subset 2^X$ из v подмножеств мощности k множества X мощности v называем k-конфигурацией, если:*

1) *каждый элемент $x \in X$ принадлежит ровно k подмножествам из $\mathcal{X}$;*
2) *каждый элемент $x \in X$ (как одноэлементное подмножество $\{x\}$) является суммой ровно k подмножеств из $\mathcal{X}$, причём каждое подмножество из $\mathcal{X}$ участвует в качестве слагаемого ровно в k суммах.*

Предполагается, что соответствующий гиперграф является связным.

Матрицу инциденций для k-конфигурации $\mathcal{X}$ будем обозначать $L_{\mathcal{X}} = ||l_{(x,A)}||_{x \in X, A \in \mathcal{X}}$, $l_{(x,A)} = 1 \Leftrightarrow x \in A$. k-конфигурацию с матрицей инциденций L будем обозначать $\mathcal{X}_L$.

Данная работа посвящена изучению инвариантов класса комбинаторной эквивалентности 5-конфигураций.

Канонический набор представлений 5-матрицы. В работе автора [2] для различения комбинаторно неэквивалентных 5-матриц используется понятие *канонического набора представлений* 5-матрицы, введённое Ф. М. Малышевым. Для данного инварианта доказано утверждение [2].

Утверждение 1. *$(v, 5)$-матрицы A, B комбинаторно эквивалентны тогда и только тогда, когда их канонические наборы представлений совпадают.*

Данный результат показывает, что канонический набор представлений 5-матрицы является инвариантом класса комбинаторной эквивалентности. Более того, с помощью этого инварианта удалось различить все комбинаторно неэквивалентные $(v, 5)$-матрицы при $v \leqslant 12$. Однако при проверке комбинаторной эквивалентности двух $(v, 5)$-матриц с помощью канонического набора представлений этих $(v, 5)$-матриц в худшем случае потребуется порядка $(v-5)!v!$ операций перестановки строк и столбцов в $(v, 5)$-матрице. Поэтому хотелось бы найти и другие инварианты, позволяющие более быстро различать комбинаторно неэквивалентные $(v, 5)$-матрицы.

Набор типов вершин 5-конфигурации. Рассмотрим ещё один инвариант класса комбинаторной эквивалентности 5-конфигураций, предложенный Ф. М. Малышевым.

Определение 3. *Пусть вершина $x \in X$ содержится в подмножествах X_1, X_2, X_3, X_4, X_5 $(v, 5)$-конфигурации $\mathcal{X}$. Тогда типом вершины $x \in X$ будем называть следующий структурированный набор чисел, содержащий всевозможные мощности пересечений четырёх, трёх, двух 5-подмножеств, содержащих x:*

$$type_{\mathcal{X}}(x) = (|X_1 \cap X_2 \cap X_3 \cap X_4|, \ldots, |X_2 \cap X_3 \cap X_4 \cap X_5|,\ |X_1 \cap X_2 \cap X_3|, \ldots$$
$$\ldots, |X_3 \cap X_4 \cap X_5|,\ |X_1 \cap X_2|, \ldots, |X_4 \cap X_5|).$$

Считаем, что наборы, получающиеся при перенумерации подмножеств $X_1, \ldots, X_5$, эквивалентны.

Определение 4. *Пусть $\mathcal{X}$ — $(v, 5)$-конфигурация на множестве X. Тогда мультимножество $type(\mathcal{X}) = \{type_{\mathcal{X}}(x) | x \in X\}$, составленное из всех типов вершин $(v, 5)$-конфигурации $\mathcal{X}$, будем называть набором типов вершин $(v, 5)$-конфигурации $\mathcal{X}$.*

Утверждение 2. *Набор типов вершин является инвариантом для всего класса комбинаторной эквивалентности с представителем $\mathcal{X}$.*

С помощью компьютерных вычислений был получен следующий результат.

Утверждение 3. *При $v \leqslant 11$ наборы типов вершин однозначно определяют $(v, 5)$-конфигурации. При $v = 12$ имеется 8 пар и одна тройка комбинаторно неэквивалентных $(v, 5)$-конфигураций, обладающих одинаковым набором типов вершин.*

Перманент 5-матрицы. В заключение работы рассмотрим такую характеристику 5-матриц, как перманент (см., например, [3]).

По определению 1 5-матрица является обратимой матрицей над полем $GF(2)$, поэтому её перманент равен единице. Также 5-матрица является матрицей инциденций соответствующей 5-конфигурации, поэтому её можно рассматривать как (0,1)-матрицу над кольцом целых чисел. И тогда такая характеристика, как перманент 5-матрицы, становится более содержательной.

Далее в этом разделе 5-матрицу L рассматриваем как $(0,1)$-матрицу над кольцом целых чисел. И под записью $\mathrm{Per}(L)$ понимаем перманент 5-матрицы L.

Важным для нас свойством перманента матрицы является то, что при перестановке строк и столбцов матрицы он не изменяется. Следовательно, перманент является инвариантом класса комбинаторной эквивалентности 5-матриц. Однако для квадратной матрицы L также верно $\mathrm{Per}(L) = \mathrm{Per}(L^T)$. Значит, с помощью данной характеристики мы не можем различить такие матрицы. Но так как мы знаем, каким преобразованием могут отличаться 5-матрицы с одинаковым перманентом, то мы не потеряем комбинаторно неэквивалентные.

Отметим, что с помощью формулы Райзера (см., например, [4]) и обхода n-мерного двоичного куба по гамильтонову циклу перманент квадратной матрицы порядка n вычисляется за порядка $n2^n$ операций сложения и умножения в рассматриваемом коммутативном кольце.

При вычислении перманентов для $(10,5)$-матриц оказалось, что имеется 13 пар $(10,5)$-матриц L, L', для которых выполнено $\mathrm{Per}(L) = \mathrm{Per}(L')$, но L, L' комбинаторно неэквивалентны и $L' \neq L^T$. Таким образом, данный инвариант класса комбинаторной эквивалентности плохо различает $(v,5)$-конфигурации даже при малых значениях v.

Далее с 5-матрицей L свяжем матрицу $L(x)$ над кольцом $\mathbb{Z}[x]$, где $L(x)$ получена из L заменой «0» на «x». Рассмотрение такой характеристики обусловлено тем, что она будет давать больше информации, чем $\mathrm{Per}(L)$. Для комбинаторно неэквивалентных $(v,5)$-матриц при $v = 10, 11, 12$ получили следующие результаты.

Утверждение 4. *Для $v = 10, 11$ имеется всего по 2 пары, а для $v = 12$ — всего 85 пар $(v,5)$-матриц L, L', для которых выполнено $Per(L(x)) = Per(L'(x))$, но L, L' комбинаторно неэквивалентны и $L' \neq L^T$.*

Автор выражает благодарность Ф. М. Малышеву за постановку задач.

Список литературы

[1] Малышев Ф. М. k-Конфигурации // Труды Математического института имени В. А. Стеклова. 2022. Т. 316. С. 248–269.

[2] Комягин М. М. Классификация $(v,5)$-конфигураций для $v \leqslant 11$ // Дискретная математика. 2024. Т. 36, вып. 1. С. 46–66.

[3] Минк Х. Перманенты. М. : Мир, 1982. 213 с.

[4] Сачков В. Н. Введение в комбинаторные методы дискретной математики. М. : МЦНМО, 2004. 424 с.

Сложность задачи о существовании сюръективного гомоморфизма для рефлексивных циклов

Корчагин Никита Павлович
Московский государственный университет имени М. В. Ломоносова; kolkor92@gmail.com

Рассматривается задача о существовании сюръективного гомоморфизма на рефлексивные циклы различной длины. Рассмотрим граф $\mathcal{G} = (V, E)$ с множеством вершин V и множеством ребер $E \subseteq V \times V$ и граф $\mathcal{H} = (V', E')$ с вершинами V' и ребрами $E' \subseteq V' \times V'$. *Гомоморфизм* графа $\mathcal{H}$ на граф $\mathcal{G}$ — это отображение $f : V' \to V$ такое, что $\forall (a_i, a_j) \in E'$ верно, что $(f(a_i), f(a_j)) \in E$. Для фиксированного графа $\mathcal{H}$ задача о существовании сюръективного гомоморфизма Surj-Hom($\mathcal{H}$) — это массовая задача, в которой по данному графу $\mathcal{G}$ требуется проверить, существует ли сюръективный гомоморфизм из $\mathcal{G}$ на $\mathcal{H}$.

Петлёй в графе $\mathcal{G} = (V, E)$ называется ребро вида (v, v), где $v \in V$. Граф называется *рефлексивным*, если для каждой $v \in V$ верно $(v, v) \in E$. Назовем граф *неориентированным*, если для каждого ребра вида (v, w) верно $(w, v) \in E$. Назовем граф *строго ориентированным*, если для каждого ребра вида (v, w), $v \neq w$, верно $(w, v) \notin E$. *Циклом длины* n, $n \geqslant 3$, будем называть граф с вершинами $V = \{0, 1, \ldots, n-1\}$, в котором для каждого $v \in V$ верно $(v, v+1 \pmod n) \in E$ или $(v+1 \pmod n, v) \in E$.

Задача о существовании сюръективного гомоморфизма была сформулирована ещё в прошлом веке [1], но до сих пор не является полностью решённой. Известно, что задача либо лежит в P, либо является NP-трудной [2, 3]. Предпринималось множество способов связать сложность Surj-Hom($\mathcal{G}$) с различными свойствами графа $\mathcal{G}$. Так, известна сложность задачи в случае, если $\mathcal{G}$ — звезда (то есть граф с ровно одной вершиной, смежной всем остальным вершинам) без петель или полный двудольный граф без петель [4], связный граф с ровно двумя петлями [5], цикл длины 6 без петель [6]. Для строго ориентированных циклов сложность задачи известна только для $n \leqslant 3$ [7]. Так, известен граф C_3, для которого задача решается за полиномиальное время.

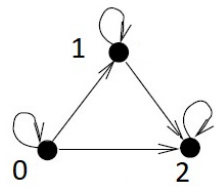

Рис. 1: Граф C_3.

Для неориентированных циклов сложность задачи известна для $n < 5$ [8].

Ключевым результатом является доказательство сложности задачи для всех строго ориентированных рефлексивных циклов, неориентированных рефлексивных циклов длины $n > 6$ [9].

Теорема 1. *Пусть C — строго ориентированный цикл, содержащий n вершин, $n \geqslant 3$. Если C совпадает с C_3, то* Surj-Hom(C) *лежит в P. Иначе* Surj-Hom(C) *является NP-трудной.*

Теорема 2. *Пусть C — неориентированный цикл, содержащий n вершин, $n > 6$. Тогда* Surj-Hom(C) *является NP-трудной.*

Для доказательства этих теорем используются полиморфизмы циклов — функции, сохраняющие их отношение смежности.

Функция f местности k называется *полиморфизмом* m-местного отношения R, если для любых $(a_1^1, \ldots, a_1^m), \ldots, (a_k^1, \ldots, a_k^m)$ из R верно, что $(f(a_1^1, \ldots, a_k^1), \ldots, f(a_1^m, \ldots, a_k^m))$ тоже из R. Функция f от n переменных называется *существенно-унарной*, если она существенно зависит не более чем от одной переменной. Будем отождествлять граф и отношение смежности на его вершинах.

Теорема 3. *Пусть C — неориентировнный цикл, содержащий n вершин, $n > 3$, и f — сюръективный полиморфизм C. Тогда f существенно унарна.*

Теорема 4. *Пусть C — строго ориентированный цикл, содержащий n вершин, $n > 3$, и f — сюръективный полиморфизм C. Тогда f существенно унарна.*

Список литературы

[1] Vikas N. Computational complexity of compaction to irreflexive cycles // Journal of Computer and System Sciences. 2004. Vol. 68, no. 1. P. 473-496.

[2] Zhuk D. A proof of the CSP dichotomy conjecture // Journal of the ACM. 2020. Vol. 67, no. 5. P. 1–78.

[3] Bulatov A. A. A dichotomy theorem for nonuniform CSPs // 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS). Washington, DC, USA : IEEE Computer Society, 2017. P. 319–330.

[4] Focke J., Goldberg L. A., Živný S. The complexity of counting surjective homomorphisms and compactions // SIAM Journal on Discrete Mathematics. 2019. Vol. 33, no. 2. P. 1006–1043.

[5] Surjective H-colouring: new hardness results / P. A. Golovach, M. Johnson, B. Martin, D. Paulusma, A. Stewart // Computability. 2019. Vol. 8, no. 1. P. 27–42.

[6] Vikas N. Computational complexity of graph partition under vertex-compaction to an irreflexive hexagon // 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017). Leibniz International Proceedings in Informatics. 2017. Vol. 83. P. 69:1–69:14.

[7] Larose B., Martin B., Paulusma D. Surjective H-colouring over reflexive digraphs // ACM Transactions on Computation Theory. 2018. Vol. 11, no. 1. P. 1–21.

[8] Martin B., Paulusma D. The computational complexity of disconnected cut and $2K_2$-partition // Journal of Combinatorial Theory, Series B. 2015. Vol. 111. P. 17–37.

[9] Корчагин Н. П. Сложность задачи о существовании сюръективного гомоморфизма на рефлексивные циклы // Интеллектуальные системы. Теория и приложения. 2023. Т. 27, вып. 4. С. 40–61.

Приближенные алгоритмы решения для соразмерной задачи open shop с маршрутизацией

Кривоногова Ольга Сергеевна, Черных Илья Дмитриевич
Институт математики имени С. Л. Соболева СО РАН, Омский филиал;
krivonogova.olga@gmail.com, idchern@gmail.com

Введение

В данной работе исследуется задача open shop с маршрутизацией, являющаяся обобщением задачи open shop и метрической задачи коммивояжёра, которая может быть описана следующим образом. Задано множество работ $\mathcal{J} = \{J_1, \ldots, J_n\}$ и множество машин $\mathcal{M} = \{M_1, M_2\}$, каждая машина M_i выполняет операции каждой работы J_j с заданной длительностью p_{ji} в произвольном порядке. Работы расположены в вершинах транспортной сети, заданной реберно-взевешенным графом $G = \langle V; E \rangle$.

Вес ребра соответствует времени перемещения машин из одной вершины в другую. Машины изначально находятся в выделенном узле, называемом базой, и должны вернуться в него после выполнения всех операций. Машина может выполнять операции только тех работ, которые расположены в том же узле, где в данное время находится машина. Длиной расписания $R_{\max}$ в этой задаче является момент возвращения последней машины на базу после выполнения всех операций. Требуется составить расписание выполнения всех операций и перемещения машин с минимальным значением $R_{\max}$. Задача с m машинами обозначается $ROm||R_{\max}$, а также $ROm|G = X|R_{\max}$, если хотим уточнить структуру транспортной сети. Задача с независимыми временами

перемещения машин, когда расстояния между вершинами для каждой машины индивидуальны, обозначается $ROm|Rtt|R_{\max}$.

Задача $ROm||R_{\max}$ была впервые сформулирована в статьях [1, 2]. В работе [1] доказано, что задача $RO2|G = K_2|R_{\max}$ является NP-трудной. Для этой постановки в [3] описан FPTAS.

Одним из частных случаев является соразмерная задача open shop с маршрутизацией: в этой постановке длительности операций одной работы совпадают, а задача обозначается $ROm|j\text{-}prpt|R_{\max}$. В статье [4] показано, что задача остаётся NP-трудной даже в соразмерном случае $RO2|G = K_2, j\text{-}prpt|R_{\max}$.

В [2] была введена стандартная нижняя оценка длины расписания для задачи $ROm||R_{\max}$:

$$R^*_{\max} \geqslant \bar{R} = \max\left\{\ell_{\max} + T^*, \max_{v\in V}(d_{\max}(v) + 2dist(v_0, v))\right\}.$$

Здесь $\ell_{\max} = \max\limits_i \sum\limits_{j=1}^{n} p_{ji}$ означает максимальную нагрузку машины, $d_{\max}(v)$ — максимальная длина работы в вершине v, T^* — оптимум задачи коммивояжёра, а $dist(u, v)$ время перемещения между вершинами u и v.

Результаты

Основным объектом исследования в данной работе является соразмерная задача open shop с маршрутизацией ($ROm|j\text{-}prpt|R_{\max}$).

Одним из направлений исследования этой задачи является поиск интервала локализации оптимумов, который заключается в следующем: найти минимальное значение параметра ρ такое, что $\forall I$ выполняется $R^*_{\max}(I) \in [\bar{R}, \rho\bar{R}]$, а также в описании примера, на котором оценка достигается.

Наш подход к нахождению таких интервалов базируется на двух процедурах упрощения примера, сохраняющих стандартную нижнюю оценку: склеивание работ (замена подмножества работ одной) и стягивание висячих вершин (перемещение работы из висячей вершины в смежную ей). Использование процедуры упрощения исходного примера позволяет сократить количество работ в примере, а также упростить структуру транспортной сети с сохранением стандартной нижней оценки.

Данный подход для соразмерной задачи open shop с двумя машинами ($RO2|j\text{-}prpt|R_{\max}$) использовался для нахождения интервалов локализации оптимумов для транспортной сети с двумя и тремя вершинами [4] и дерева [5] в случае идентичных времен перемещения. Во всех этих случаях интервал локализации оптимумов равен $[\bar{R}, \frac{7}{6}\bar{R}]$.

В [5] также рассматривалась задача с независимыми временами перемещения ($ROm|Rtt, j\text{-}prpt|R_{\max}$): было доказано, что для случая с двумя вер-

шинами интервалы локализации оптимумов для задачи с идентичными временами перемещения и независимыми временами перемещения совпадают. Основным результатом данной работы является обобщение этого результата для некоторых частных случаев соразмерной задачи open shop с независимыми временами перемещения на дереве.

Также была рассмотрена соразмерная задача с m машинами и произвольной транспортной сетью $ROm|j\text{-}prpt|R_{\max}$. Для этой задачи была доказана следующая теорема.

Теорема 1. *Пусть I — пример задачи $ROm|j\text{-}prpt|R_{\max}$. Тогда за время, линейное от числа работ, можно построить расписание S, длина которого принадлежит интервалу $[\bar{R}, \frac{5}{2}\bar{R}]$.*

Доказательство использует результат для задачи $Fm|prpt|C_{max}$, представленный в [6], о том, что любое перестановочное расписание для задачи $Fm|prpt|C_{max}$ является оптимальным.

Исследование выполнено за счет гранта Российского научного фонда № 22-71-10015, https://rscf.ru/project/22-71-10015/.

Список литературы

[1] Averbakh I., Berman O., Chernykh I. The routing open-shop problem on a network: complexity and approximation // European Journal of Operational Research. 2006. Vol. 173, no. 2, P. 521–539.

[2] Averbakh I., Berman O., Chernykh I. A $\frac{6}{5}$-approximation algorithm for the two-machine routing open-shop problem on a two-node network // European Journal of Operational Research. 2005. Vol. 166, no. 1. P. 3–24.

[3] Kononov A. $O(\log m)$-approximation for the routing open shop problem // RAIRO-Operations Research. 2015. Vol. 49, no. 2. P. 383–391.

[4] Pyatkin A. V., Chernykh I. D. On complexity of two-machine routing proportionate open shop // Siberian Electronic Mathematical Reports. 2022. Vol. 19, no. 1. P. 273—284.

[5] Chernykh I., Krivonogova O., Shmyrina A. Approximation algorithms for two-machine proportionate routing open shop on a tree // Mathematical Optimization Theory and Operations Research. MOTOR 2023. Lecture Notes in Computer Science. 2023. Vol. 13930. P. 197–211.

[6] Chin F. Y., Tsai Long-Lieh. On J-maximal and J-minimal flow-shop schedules // Journal of the ACM. 1981. Vol. 28, no. 3. P. 462–476.

О действии отображения дуальности на один класс обобщённых бент-функций

Куценко Александр Владимирович
Новосибирский государственный университет; alexandrkutsenko@bk.ru

Обозначим через $\mathbb{F}_2^n$ пространство двоичных векторов с n координатами. *Обобщённой булевой функцией* от n переменных называется произвольное отображение из пространства $\mathbb{F}_2^n$ в кольцо $\mathbb{Z}_q$ (см. [1]). В случае $q = 2$ функция называется *булевой* функцией. Каждая обобщённая булева функция от n переменных единственным образом представима в виде многочлена Жегалкина (алгебраической нормальной формы, АНФ) над кольцом $\mathbb{Z}_q$:

$$f(x_1, x_2, \ldots, x_n) = a_0 + \sum_{k=1}^{n} \sum_{1 \leqslant i_1 < i_2 < \ldots < i_k \leqslant n} a_{i_1 i_2 \ldots i_k} x_{i_1} x_{i_2} \ldots x_{i_k},$$

где $a_z \in \mathbb{Z}_q$ для всех z. *Алгебраической степенью* функции f называется максимальная из степеней одночленов, входящих в её многочлен Жегалкина с ненулевыми коэффициентами.

Весом Ли элемента $x \in \mathbb{Z}_q$ называется число $\mathrm{wt}_L(x) = \min\{x, q - x\}$. *Весом Ли* обобщённой булевой функции от n переменных называется сумма весов Ли всех её значений:

$$\mathrm{wt}_L(f) = \sum_{x \in \mathbb{F}_2^n} \mathrm{wt}_L(f(x)).$$

Расстояние Ли $\mathrm{dist}_L(f, g)$ между обобщёнными булевыми функциями f, g от n переменных есть число $\mathrm{wt}_L(f - g)$. Заметим, что в булевом случае $q = 2$ вес, а также расстояние Ли совпадают с весом и, соответственно, расстоянием Хэмминга.

Пусть $\omega = e^{2\pi i/q}$ — примитивный корень из 1. Для $x, y \in \mathbb{F}_2^n$ через $\langle x, y \rangle$ обозначим выражение $\bigoplus_{i=1}^{n} x_i y_i$, где знак $\oplus$ есть операция сложения по модулю 2. *Преобразованием Уолша — Адамара* обобщённой булевой функции f от n переменных называется комплекснозначная функция

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

Особое внимание уделяется классу (обобщённых) булевых функций, обладающих равномерным спектром Уолша — Адамара. Такие функции называются *(обобщёнными) бент-функциями*, они нашли ряд приложений в криптографии, обработке сигналов, а также теории кодирования.

Определение 1. *Обобщённая булева функция f от n переменных называется* обобщённой бент-функцией, *если*

$$|H_f(y)| = 2^{n/2}$$

для всех $y \in \mathbb{F}_2^n$.

Свойствам и конструкциям обобщённых бент-функций посвящён ряд работ, в частности, [2–5].

Если существует булева функция $\widetilde{f}$ от n переменных такая, что $H_f(y) = \omega^{\widetilde{f}(y)} 2^{n/2}$ для всех $y \in \mathbb{F}_2^n$, то f называется *регулярной*, а $\widetilde{f}$ — её *дуальной*. Отметим, что дуальная функция $\widetilde{f}$ также является обобщённой бент-функцией. Известно, что при $q = 2^k, k \geqslant 2$, все обобщённые бент-функции являются регулярными, как для чётного, так и для нечётного n, за исключением единственного случая, когда $k = 2$, а n — нечётное число [4].

Многочлены Жегалкина вида

$$f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} \lambda_j x_j + \lambda_0, \quad x \in \mathbb{F}_2^n, \tag{1}$$

где $\lambda_0, \lambda_1, \lambda_2, \ldots, \lambda_n \in \mathbb{Z}_q$, задают множество обобщённых булевых функций степени 1 от n переменных. Векторы значений данных функций образуют обобщённый код Рида — Маллера $\mathrm{RM}_q(r, n)$ [1]. Хорошо известно, что булева бент-функция не может иметь степень 1, а также не может зависеть от нечётного числа переменных, но в случае с обобщёнными бент-функциями ситуация меняется, — в работе [5] были приведены достаточные условия того, что при $q \equiv 0 \pmod 4$ обобщённая булева функция вида (1) является обобщённой бент-функцией, а также представлены примеры таких функций. Следующий результат обобщает известные данные.

Теорема 1. *Обобщённая булева функция вида* (1) *является обобщённой бент-функцией тогда и только тогда, когда $q \equiv 0 \pmod 4$ и $\lambda_j \in \left\{\frac{q}{4}, \frac{3q}{4}\right\}$ для всех $j = 1, 2, \ldots, n$. При этом она является регулярной, если и только если выполнено по крайней мере одно из условий:*

1) число n — чётное;

2) $q \equiv 0 \pmod 8$.

Её дуальная в этом случае имеет вид

$$\widetilde{f}(x) = \sum_{j=1}^{n} (q - \lambda_j) x_j + \left(\lambda_0 + \frac{3q}{4} n + \frac{3}{2} \sum_{k=1}^{n} \lambda_k \right).$$

Отображение называется *изометричным*, если оно сохраняет расстояние между каждой парой функций. С обобщёнными бент-функциями связан ряд

открытых вопросов, одним из них является описание изометричных отображений, оставляющих множество обобщённых бент-функций от n переменных на месте. Данный вопрос тесно связан с задачей исследования группы автоморфизмов данного класса функций, что подразумевает изучение его структурных и метрических свойств. Стоит отметить, что в данном случае дополнительно ставятся вопросы выбора метрики и исследования группы автоморфизмов в различных метриках.

Отображение $f \to \widetilde{f}$, определённое на множестве регулярных обобщённых бент-функций от n переменных и ставящее в соответствие каждой регулярной обобщённой бент-функции от n переменных дуальную к ней функцию, называется *отображением дуальности*. Известно, что для $q = 2$ отображение дуальности является единственным известным отображением, действующим на множестве бент-функций от n переменных изометрично и не являющимся элементом группы автоморфизмов множества бент-функций от n переменных. При этом неизвестно, обладает ли данное отображение таким свойством по отношению к обобщённым бент-функциям. В настоящей работе доказана следующая

Теорема 2. *На множестве регулярных обобщённых бент-функций степени* 1 *отображение дуальности является изометрией относительно метрики Ли.*

Открытым остаётся вопрос, является ли отображение дуальности изометричным на множестве обобщённых бент-функций большей степени. Данный вопрос также можно рассматривать по отношению к известным классам обобщённых бент-функций относительно различных метрик.

Список литературы

[1] Davis J. A., Jedwab J. Peak-to-mean power control in OFDM, Golay complementary sequences, and Reed-Muller codes // IEEE Transactions on Information Theory. 1999. Vol. 45, no. 7. P. 2397–2417.

[2] Schmidt K.-U. Quaternary constant-amplitude codes for multicode CDMA // IEEE Transactions on Information Theory. 2009. Vol. 55, no. 4. P. 1824–1832.

[3] Bent and generalized bent Boolean functions / P. Stănică, T. Martinsen, S. Gangopadhyay, B. K. Singh // Designs, Codes and Cryptography. 2013. Vol. 69. P. 77–94.

[4] Martinsen T., Meidl W., Stănică P. Generalized bent functions and their Gray images // Arithmetic of Finite Fields. WAIFI 2016. Lecture Notes in Computer Science. 2016. Vol. 10064. P. 160–173.

[5] Singh B. K. On cross-correlation spectrum of generalized bent functions in generalized Maiorana–McFarland class // Information Sciences Letters. 2013. Vol. 2, no. 3. P. 139–145.

О сложности реализации универсального клеточного многополюсника для класса самодвойственных функций

Ложкин Сергей Андреевич, Зизов Вадим Сергеевич
Московский государственный университет имени М. В. Ломоносова,
факультет вычислительной математики и кибернетики; lozhkin@cs.msu.ru, alvadia@cs.msu.ru

Введение

Модель клеточных схем (КС) впервые была предложена в 1967 году С. С. Кравцовым в работе [1], в которой для неё был получен порядок функции Шеннона. Функция Шеннона характеризует сложность самой «сложной» функции алгебры логики (ФАЛ) от n переменных. Модель 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 изученных классов схем из функциональных элементов (СФЭ).

Аналогичная математическая модель в зарубежных источниках была описана в 1980 году К. Д. Томпсоном в работе [2]. Дальнейшие продвижения в работах [3, 4] показали асимптотические оценки высокой степени точности (АОВСТ) для функции Шеннона клеточных схем, реализующих отдельные классы функций.

В работе [3] были установлены асимптотически точные верхние и нижние оценки для площади схем над базисом $Б_0'$ (см. рис. 1 и ср. с [1]), реализующих дешифратор порядка n, которые имеют вид $n2^{n-1}(1 \pm O(\frac{1}{n}))$.

В работе [4] были установлены верхние и нижние АОВСТ для сложности $A_{Б_0'}(\vec{P}_{2n}(n))$, то есть для площади универсального многополюсника порядка n в модели клеточных схем над базисом $Б_0'$, имеющие вид

$$n \cdot 2^{2^n-1} - O(n^2) \leqslant A_{Б_0'}(\vec{P}_2(n)) \leqslant (n+6)2^{2^n-1} + \frac{3n}{2^n}2^{2^n-1}. \qquad (1)$$

В настоящей работе аналогичные АОВСТ получаются для системы всех самодвойственных функций.

Основные результаты

Утверждение 1. *Для площади клеточной схемы из функциональных и коммутационных элементов Σ_n над базисом $Б_0'$, реализующей систему всех самодвойственных функций $S(n)$, верна верхняя оценка*

$$A(\Sigma_n) \leqslant 2^{2^{n-1}-1}(n+8) + O(2^{2^{n-1}}).$$

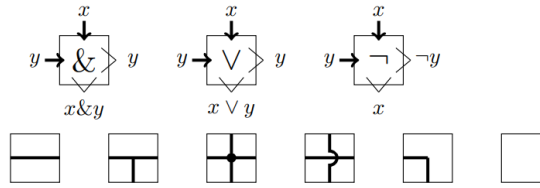

Рис. 1: Базис $Б_0'$: *функциональные элементы* — конъюнкция (&), дизъюнкция ($\vee$) и отрицание ($\neg$); *коммутационные элементы* (слева направо) — проводник, Т-образный разветвитель, разветвитель, пересечение без соединения, поворот, изолятор.

Утверждение 2. *Для площади системы всех самодвойственных функций $S(n)$ верна нижняя оценка:*

$$A_{Б_0'}(S(n)) \geqslant 2^{2^{n-1}-1}n - O(2^{2^{n-1}}).$$

Таким образом, основным результатом работы является следующая теорема

Теорема. *Для системы всех самодвойственных функций $S(n)$ выполняется равенство*

$$A_{Б_0'}(S(n)) = 2^{2^{n-1}-1}(n \pm O(1)).$$

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. Вып. 19. С. 285–292.

[2] Thompson C. D. A complexity theory for VLSI : Doctoral Dissertation ; Carnegie-Mellon University, Computer Science Department. Pittsburgh, PA, USA. 1980. 132 p.

[3] Ложкин С. А., Зизов В. С. Уточненные оценки сложности дешифратора в модели клеточных схем из функциональных и коммутационных элементов // Ученые записки Казанского университета. Серия физико-математические науки. 2020. Т. 162, № 3. С. 322–334.

[4] Ложкин С. А., Зизов В. С. Уточненные оценки сложности универсального многополюсника в модели клеточных схем // Материалы XIV Международного семинара «Дискретная математика и ее приложения» имени академика О. Б. Лупанова. 2022. М. : издательство ИПМ им. М. В. Келдыша РАН. С. 80–82.

О сложности линейной функции алгебры логики в некоторых классах обобщенных контактных схем

Ложкин Сергей Андреевич, Михалев Евгений Константинович
Московский государственный университет имени М. В. Ломоносова;
lozhkin@cs.msu.ru, genya.mikhalev@gmail.com

В данной работе рассматривается задача индивидуального синтеза, а именно синтеза обобщенных контактных схем (КС), реализующих линейную функцию алгебры логики (ФАЛ). В ней изучается класс обобщенных КС ранга не более r, то есть класс КС в базисе, состоящем из контактов, реализующих все различные ФАЛ от r булевых переменных (БП). Исследуется сложность реализации линейной ФАЛ в данном классе схем, а также в некоторых его подклассах.

Ранее было установлено, что сложность линейной ФАЛ от n БП в классе контактных схем в стандартном базисе (базисе, содержащем замыкающий и размыкающий контакты) равна $4n-4$. При этом верхняя оценка указанного вида получается с помощью метода каскадов (см., например, [1]), а нижняя была доказана в [2] (более простое доказательство, предложенное С. А. Ложкиным, см., например, в [3]).

Пусть $B=\{0,1\}$, а $B^n=\underbrace{B\times B\times\cdots\times B}_{n}$ — его n-я декартова степень или, иначе, единичный n-мерный куб, то есть множество наборов $\alpha=(\alpha_1,\ldots,\alpha_n)$, где $\alpha_i\in B$ при всех $i, i\in[1,n]$.

Пусть X — множество, элементами которого являются БП, а $|X|$ — его мощность, то есть число БП, которые в нём содержатся. Множество БП $\{x_1,x_2,\ldots,x_n\}$ будем обозначать через $X(n)$, а множество всех ФАЛ от БП из $X(n)$ — через $P_2(n)$.

Набор $\alpha=(\alpha_1,\ldots,\alpha_n)\in B^n$ будем называть *четным*, если в нем содержится четное число единиц, то есть число α_i, таких, что $\alpha_i=1$, является четным, $i=1,\ldots,n$. В противном случае набор α будем называть *нечетным*. Через $B^n_{\text{чет}}$ обозначим множество всех четных наборов от n БП. Функция l_r (соответственно $\overline{l_n}$) называется *линейной нечетной* (*линейной четной*) ФАЛ ранга r, если она существенно зависит от r БП и принимает значение 1 только на всех нечетных (четных) наборах. Контакт, управляемый такой ФАЛ, называется *линейным нечетным* (*линейным четным*) контактом ранга r.

В работе рассматривается специальный подкласс U_r^{Kl} класса обобщенных КС, включающий в себя все контактные схемы, которые состоят только из линейных контактов ранга r. Заметим, что класс U_r^{Kl} является полным при любом r, $r\geqslant 1$, то есть в нем можно реализовать любую ФАЛ.

Сложностью $L(\Sigma)$ *контактной схемы* Σ будем, как обычно, называть число её контактов, а *сложностью* $L_r^{Kl}(f)$ ФАЛ f в классе КС U_r^{Kl} — минимальную сложность КС из U_r^{Kl}, реализующую данную ФАЛ f.

Функцию f из $P_2(n)$ будем называть α-сферической, где $\alpha \in B^n$, если для любых наборов β и γ из B^n, отличающихся от α ровно в одном и ровно в двух разрядах соответственно, $f(\beta) = 0$ и $f(\gamma) = 1$. При этом $\widetilde{0}$-сферическую ФАЛ будем называть просто сферической.

В данной работе исследована сложность реализации линейной ФАЛ l_n в классе U_r^{Kl}. Получены точная верхняя и асимптотически точная нижняя оценки сложности линейной ФАЛ l_n в данном классе.

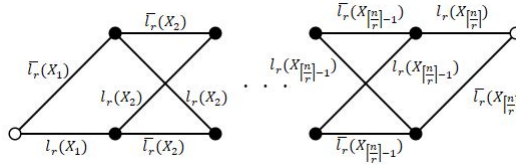

Рис. 1: Схема Кардо

Для получения верхней оценки сложности используется структура так называемой схемы Кардо (см., например, [1]). Разбив множество всех БП $X(n)$ на $\lceil \frac{n}{r} \rceil$ непересекающихся подмножеств $X_1, X_2, \ldots, X_{\lceil \frac{n}{r} \rceil}$, мощность каждого из которых, кроме, быть может, одного, равна r, можно построить схему типа схемы Кардо (см. рис. 1), контактами которой управляют линейные ФАЛ $l_r(X_i)$, $\overline{l_r}(X_i)$ от полученных множеств БП X_i, где $i = 1, 2, \ldots, \lceil \frac{n}{r} \rceil$. Таким образом может быть установлена следующая верхняя оценка исследуемой сложности:

$$L_r^{Kl}(l_n) \leqslant 4 \left\lceil \frac{n}{r} \right\rceil - 4.$$

Для получения нижней оценки данной сложности сначала исследуется сложность сферической ФАЛ в классе U_r^{Kl}. Было доказано, что если КС Σ, $\Sigma \in U_r^{Kl}$, реализует сферическую ФАЛ f из $P_2(n)$ и $L(\Sigma) \leqslant 4\lceil \frac{n}{r} \rceil$, то в КС Σ содержится не менее $2\lceil \frac{n}{r} \rceil - c_0 \frac{\sqrt{n}}{r}$ нечетных контактов, где c_0 — некоторая константа, зависящая от r и не зависящая от n. Так как ФАЛ l_n является α-сферической для любого набора $\alpha \in B^n_{\text{нечет}}$, то далее для доказательства нижней оценки сложности линейной ФАЛ l_n в классе U_r^{Kl}, следуя [4], используется средняя проводимость КС. В результате получена следующая оценка:

$$L_r^{Kl}(l_n) \geqslant 4 \left\lceil \frac{n}{r} \right\rceil - 2c_0 \frac{\sqrt{n}}{r}.$$

Из предыдущей оценки следует асимптотически точная нижняя оценка исследуемой сложности*:

$$L_r^{Kl}(l_n) \gtrsim 4\left\lceil \frac{n}{r} \right\rceil .$$

В итоге получена следующая асимптотически точная оценка сложности линейной ФАЛ l_n в классе U_r^{Kl}:

$$L_r^{Kl}(l_n) \sim 4\left\lceil \frac{n}{r} \right\rceil .$$

Список литературы

[1] Ложкин С. А. Лекции по основам кибернетики. М. : Издательский отдел ф-та ВМиК МГУ, 2004.

[2] Cardot C. Quelques résultats sur l'application de l'algèbre de Boole a la synthèse des circuits a relais // Annales Des Télécommunications. 1952. Vol. 7, no. 2. P. 75–84.

[3] Яблонский С. В. Элементы математической кибернетики. М. : Высшая школа, 2007.

[4] Ложкин С. А., Кошкин Н. А. О сложности реализации некоторых систем функций алгебры логики контактными многополюсниками // Доклады Академии наук СССР. 1988. Т. 298, № 4. С. 807–811.

Построение оптимальных двусторонних вложений полных двоичных и троичных деревьев в прямоугольные решетки

Ложкин Сергей Андреевич, Мо Ди

Московский государственный университет имени М. В. Ломоносова;
lozhkin@cs.msu.ru, xdmodi1991@163.com

Из задачи организации взаимодействия и моделирования вычислений возникает задача оптимального вложения деревьев в прямоугольные решетки (ПР), т. е. графы, вершинами которых являются точки плоскости с координатами (x, y), где x и y — целые числа такие, что $x \in [a, a + \lambda)$, $y \in [b, b + h)$ для заданных целых a, b и натуральных h, λ. При этом ребра соединяют все пары точек (x_1, y_1) и (x_2, y_2) таких, что $|x_1 - x_2| + |y_1 - y_2| = 1$. В узлах ПР можно размещать вычислительные узлы, а по ребрам проводить соединяющие их проводники. Вложения могут быть описаны отображениями вершин

*Используются следующие обозначения асимптотических неравенств и равенств:
$a(n) \gtrsim b(n)$, если $a(n) \geqslant b(n)(1 + \varepsilon(n))$;
$a(n) \sim b(n)$, если $a(n) = b(n)(1 + \varepsilon(n))$,
для некоторой последовательности $\varepsilon(n) \to 0$ при $n \to \infty$.

дерева в узлы ПР, а ребер — в цепи решетки. Задача оптимизации вложения сводится к нахождению при определенных условиях минимальной высоты h, длины λ, $\lambda \geqslant h$, и площади $h \cdot \lambda$ допускающей его ПР.

Следуя [1], для графа G через $V(G)$, $X(G)$ и $C(G)$ будем обозначать множество его вершин, ребер и простых цепей соответсвенно, а сам граф G будем записывать в виде $G = (V(G), X(G))$. Будем говорить, что упорядоченная пара отображений (ϕ, ψ) определяет вложение графа G в граф F, если $\phi : V(G) \to V(F)$, $\psi : X(G) \to C(F)$ и для любого ребра $x = (u, v)$, $x \in X(G)$, цепь $\phi(x)$ соединяет вершины $\phi(u)$ и $\phi(v)$. Вершины графа F, которые являются образами вершин графа G, бутем называть основными вершинами вложения, цепи графа F, которые соответствуют ребрам графа G, — его транзитными цепями, внутренние вершины и ребра транзитных цепей — транзитными вершинами и ребрами.

Пусть B^4 — множество всех ненулевых двоичных наборов длины 4 с обычным отношением частичного порядка $\leqslant$, которое имеет место, если аналогичное неравенство выполняется в каждом разряде, а $\hat{B} = B^4 \cup \{*\}$, причем для всех δ, $\delta \in B^4$, справедливо $\delta < *$. Будем предполагать, что стороны граничного прямоугольника решетки A пронумерованы числами от 1 до 4, считая от левой вертикальной стороны по часовой стрелке. Пусть $\delta(A) = (\delta_1\delta_2\delta_3\delta_4) \in B^4$, где $\delta_i = 1$ тогда и только тогда, когда полюса (листья дерева) могут располагаться на границе решетки и на её стороне с номером i, и пусть $\delta(A) = *$, если полюса могут располагаться в решетке A произвольным образом. Решетку A будем при этом называть δ-решеткой, где $\delta = \delta(A)$. Если через каждую транзитную вершину проходит не более ν, $\nu = 1, 2$, различных транзитных цепей, то соответствующее вложение называется ν-вложением. Для дерева D обозначим через $k(D)$ число его ярусов, а через $H_\nu^\delta(D)$ и $S_\nu^\delta(D)$ ($\delta \in \hat{B}$, $\nu \in \{1, 2\}$) — минимальное значение наименьшего линейного размера (высоты) и числа вершин решетки A соответственно, где минимум берется по всем δ-решеткам A, в которые возможно ν-вложение дерева D. Очевидно, что величины $H_\nu^\delta(D)$ и $S_\nu^\delta(D)$ монотонно не возрастают по ν, $\nu = 1, 2$, и δ, $\delta \in \hat{B}$, а также не изменяются при перестановке координат одной четности в наборе δ, $\delta \in B^4$.

Теорема 1 ([1]). *Для любого d-ичного, $d \in \{2, 3\}$, дерева $D_d(k)$, $k = 1, 2, \ldots$, и всех $\nu \in \{1, 2\}$, $\delta \in \hat{B}$, выполняется равенство*

$$H_\nu^\delta(D_d(k)) = \left\lceil \frac{k+1}{4-d} \right\rceil. \tag{1}$$

Задача об оптимизации площади ПР при дополнительном ограничении на расположение листьев на границе решетки была рассмотрена в [1, 2]. Там для оптимальной площади 1-вложения полного k-ярусного d-ичного, где $d = 2, 3$,

дерева была получена асимптотически точная при $k = 1, 2, \ldots$ оценка порядка kd^k, в то время как высота построенного вложения не больше чем на константу отличалась от минимально возможного значения высоты ПР, допускающей указанное вложение, равного $\lceil \frac{k+1}{4-d} \rceil$.

Теорема 2 ([1]). *При всех $d \in \{2, 3\}$, $\delta \in B^4$, $\nu \in \{1, 2\}$ и $k = 1, 2, \ldots$ имеет место соотношение*

$$S^{\delta}_{\nu}(D_d(k)) = \left(\frac{1}{s(\delta)} + o(1) \right) d^k H(d, k), \tag{2}$$

где $\delta \neq *$ и $s(\delta) = s(\delta_1, \delta_2, \delta_3, \delta_4) = \max\{\delta_1 + \delta_3, \delta_2 + \delta_4\} \leqslant \nu$.

Теорема 3 (следует из [1, 3]). *Для случая $\delta \neq *$ и $s(\delta) = 2$ при $k = 1, 2, \ldots$ имеет место соотношение*

$$S^{\delta}_{1}(D_2(k)) = \frac{k}{3} \cdot 2^k \cdot (1 + o(1)). \tag{3}$$

В данной работе продолжаются исследования [1–3] и рассматривается задача оптимальных по высоте (длине, т. е. «второму» линейному размеру решетки) двусторонних вложений полных двоичных и троичных k-ярусных деревьев в ПР минимальной длины (высоты) и связь между высотой и длиной решетки в указанных выше условиях. Эта связь исследуется на уровне т. н. асимптотических оценок высокой степени точности, когда поведение этих параметров при $k = 1, 2, \ldots$ устанавливается с точностью до слагаемого вида $O(1)$.

Теорема 4. *Для любого полного двоичного дерева $D_2(k)$, $k = 1, 2, \ldots$, и $\delta = (0101)$, $\nu = 2$ существует 2-вложение этого дерева в δ-решетку высоты h и длины λ, где*

$$h \leqslant \left\lfloor \frac{k}{2} \right\rfloor + 4, \lambda = 2^{k-1}. \tag{4}$$

Теорема 4 получена индукцией по $k = 2, 3, \ldots$. На рис. 1 показаны искомые вложения полных двоичных деревьев в ПР, когда k равно 2, 3, 4 и 5 соответственно.

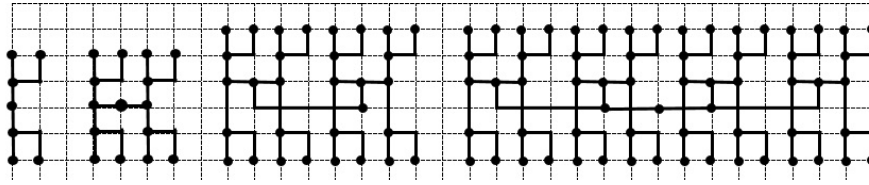

Рис. 1: Вложения двоичных деревьев.

Теорема 5. *Для любого полного троичного дерева $D_3(k)$, $k = 1, 2, \ldots$, и $\delta = (0101)$, $\nu = 2$ существует 2-вложение этого дерева в δ-решетку высоты h и длины λ, где*

$$h \leqslant k + 3, \lambda = \left\lceil \frac{3^k}{2} \right\rceil. \tag{5}$$

Теорема 5 получена индукцией по $k = 3, 4, \ldots$. На рис. 2 показано искомое вложение полного троичного 3-ярусного дерева в ПР.

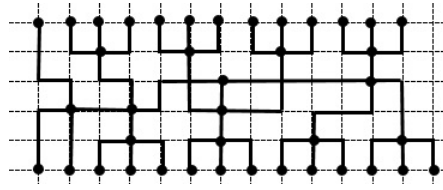

Рис. 2: Вложение троичного дерева.

Список литературы

[1] Ложкин С. А., Ли Да Мин. О некоторых оптимальных вложениях двоичных и троичных деревьев в плоские прямоугольные решетки // Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. 1995. № 4. С. 49–55.

[2] Ли Да Мин. Некоторые оптимальные вложения древовидных графов в плоские прямоугольные решетки : специальность 01.01.09 «Дискретная математика и математическая кибернетика» : диссертация на соискание ученой степени кандидата физико-математических наук ; МГУ имени М. В. Ломоносова. Москва, 1994.

[3] Высоцкий Л. И., Ложкин С. А. Оптимальные двусторонние вложения полных двоичных деревьев в прямоугольные решетки // Прикладная математика и информатика : Труды факультета ВМК МГУ имени М. В. Ломоносова. М. : МАКС Пресс, 2018. Т. 59. С. 25–39.

О сложности задачи о вершинной 3-раскраске для некоторых пар 6-вершинных порожденных запретов

Малышев Дмитрий Сергеевич

Национальный исследовательский университет «Высшая школа экономики»;
Московский физико-технический институт; dsmalyshev@rambler.ru

В настоящей работе рассматриваются только *обыкновенные графы* — неориентированные графы без петель и кратных ребер. *Наследственный*

класс графов — множество графов, замкнутое относительно удаления вершин. Каждый наследственный класс графов $\mathcal{X}$ задается множеством своих *запрещенных порожденных подграфов* $\mathcal{Y}$, при этом принята запись $\mathcal{X} = Free(\mathcal{Y})$. Например, класс лесов задается запрещением всех порожденных циклов, а класс двудольных графов задается запрещением всех порожденных нечетных циклов.

Любое отображение, назначающее каждой вершине графа G элемент множества $\{1, 2, \ldots, k\}$ так, что любым соседним вершинам назначаются различные цвета, называется *вершинной k-раскраской* графа G. *Задача о вершинной k-раскраске* (кратко, *задача* k-ВР) состоит в том, чтобы для заданного графа проверить, допускает ли он вершинную k-раскраску или нет. При любом $k > 2$ задача k-ВР является классической NP-полной задачей на графах [1].

Для некоторых k сложностной статус задачи k-ВР остается открытым даже при запрещении небольших порожденных фрагментов. Так, сложность задачи k-ВР открыта для класса $Free(\{P_8\})$ и $k = 3$, а также для класса $Free(\{P_7\})$ и $k = 4$, где через P_n обозначен простой путь на n вершинах. Вместе с тем, относительно семейств наследственных классов, определяемых порожденными запрещенными подграфами, известен ряд следующих результатов об алгоритмической сложности задачи k-ВР:

— установлена (см. работу [2]) алгоритмическая сложность задачи 3-ВР для всех классов вида $Free(\{H\})$, где $|V(H)| \leqslant 6$;

— установлена (см. работу [3]) алгоритмическая сложность задачи 4-ВР для всех классов вида $Free(\{H\})$, где $|V(H)| \leqslant 5$;

— для каждого k задача k-ВР разрешима за полиномиальное время в классе $Free(\{P_5\})$ [4];

— задача 3-ВР полиномиально разрешима в классе $Free(\{P_7\})$ [5];

— задача 4-ВР полиномиально разрешима в классе $Free(\{P_6\})$ [6];

— установлена (см. работу [7]) алгоритмическая сложность задачи 3-ВР для всех классов вида $Free(\{H\})$, где $|V(H)| \leqslant 7$;

— задача 4-ВР является NP-полной в классе $Free(\{P_7\})$, но для каждого $k \geqslant 5$ задача k-ВР является NP-полной в классе $Free(\{P_6\})$ [8];

— для любого множества запрещенных порожденных подграфов, каждый не более чем с 5 вершинами, получена алгоритмическая сложность задачи 3-ВР на соответствующем наследственном классе графов [9–12].

В данной работе рассматривается задача 3-ВР для пар (H_1, H_2), состоящих из 6-вершинных запрещенных порожденных подграфов. Задача 3-ВР будет NP-полной (см., например, работу [9]), если класс $\mathcal{X} = Free(\{H_1, H_2\})$ включает один из двух классов: $\mathcal{X}_1$ — класс лесов, а $\mathcal{X}_2$ — класс реберных графов лесов со степенями всех вершин не более чем 3. По-видимому, этих двух

классов достаточно для полной классификации алгоритмической сложности задачи 3-ВР для пар 6-вершинных запрещенных порожденных фрагментов:

Предположение. *Пусть* $\mathcal{X} = Free(\{H_1, H_2\})$, *где* H_1 *и* H_2 *содержат не более 6 вершин каждый. Тогда задача* 3*-ВР является NP-полной в классе* $\mathcal{X}$, *если* $\mathcal{X}_1 \subseteq \mathcal{X}$ *или* $\mathcal{X}_2 \subseteq \mathcal{X}$, *а иначе она является полиномиально разрешимой в классе* $\mathcal{X}$.

В этой работе мы частично подтверждаем это предположение. Согласно [7, 9] можно считать, что $H_1 \in \mathcal{X}_1$, причем H_1 отличен от *линейного леса* (т. е. дизъюнктной суммы путей), и что $H_2 \in \mathcal{X}_2 \setminus \mathcal{X}_1$. В этой работе рассматриваются пары вида $(K_{1,3}^{++}, H)$ и $(K_{1,4}^{+}, H)$, где $|V(H)| \leqslant 6$, $K_{1,n}$ — звезда с n листьями, $K_{1,3}^{++}$ — результат добавления к $K_{1,3}$ двух изолированных вершин, $K_{1,4}^{+}$ — результат добавления к $K_{1,4}$ изолированной вершины. Справедливо следующее

Утверждение. *Пусть* $\mathcal{X} = Free(\{K_{1,3}^{++}, H\})$ *или* $\mathcal{X} = Free(\{K_{1,4}^{+}, H\})$, *где* $|V(H)| \leqslant 6$. *Тогда задача* 3*-ВР полиномиально разрешима в классе* $\mathcal{X}$, *если* $H \in \mathcal{X}_2$, *а иначе она является NP-полной для* $\mathcal{X}$.

Работа выполнена при финансовой поддержке Министерства Образования и Науки РФ (проект FSMG-2024-0025).

Список литературы

[1] Garey M. R., Johnson D. S. Computers and intractability: a guide to the theory of NP-completeness. New York, NY, USA : W. H. Freeman and Co., 1979. 338 p.

[2] Updating the complexity status of coloring graphs without a fixed induced linear forest / H. J. Broersma, P. A. Golovach, D. Paulusma, Jian Song // Theoretical Computer Science. 2012. Vol. 414, no. 1. P. 9–19.

[3] Golovach P. A., Paulusma D., Song Jian. 4-coloring H-free graphs when H is small // Discrete Applied Mathematics. 2013. Vol. 161, no. 1–2. P. 140–150.

[4] Hoàng C., Kamiński M., Lozin V. V., Sawada J., Shu Xiao. Deciding k-colorability of P_5-free graphs in polynomial time // Algorithmica. 2010. Vol. 57, no. 1. P. 74–81.

[5] Three-coloring and list three-coloring of graphs without induced paths on seven vertices / F. Bonomo, M. Chudnovsky, P. Maceli, O. Schaudt, M. Stein, Mingxian Zhong // Combinatorica. 2018. Vol. 38. P. 1–23.

[6] Colouring ($P_r + P_s$)-free graphs / T. Klimosová, J. Malík, T. Masarík, J. Novotná, D. Paulusma, V. Slívová // Algorithmica. 2020. Vol. 82, no. 7. P. 1833–1858.

[7] Spirkl S., Chudnovsky M., Zhong Mingxian. Four-coloring P_6-free graphs // ACM-SIAM Symposium on Discrete Algorithms. Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA, USA, Jan. 10–12, 2016). Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 2019. P. 1239–1256.

[8] Huang Shenwei. Improved complexity results on k-coloring P_t-free graphs // European Journal of Combinatorics. 2016. Vol. 51. P. 336–346.

[9] Malyshev D. S. 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.

[10] Malyshev D. S. The complexity of the vertex 3-colorability problem for some hereditary classes defined by 5-vertex forbidden induced subgraphs // Graphs and Combinatorics. 2017. Vol. 33, no. 4. P. 1009–1022.

[11] Сироткин Д. В., Малышев Д. С. О сложности задачи вершинной 3-раскраски для наследственных классов графов, определённых запретами небольшого размера // Дискретный анализ и исследование операций. 2018. Т. 25, № 4. С. 112–130.

[12] Малышев Д. С. Полная классификация сложности задачи о вершинной 3-раскраске для четверок порожденных 5-вершинных запретов // Журнал Средневолжского математического общества. 2020. Т. 22, № 1. С. 38–47.

Комбинаторные конфигурации для линейной среды алгоритмов шифрования

Малышев Фёдор Михайлович
Математический институт имени В. А. Стеклова РАН; malyshevfm@mi-ras.ru

Основная идея построения современных блочных шифраторов восходит к квадрату Полибия (III век до н. э.). Шеннон называл его дробным шифром [1]. Квадрат Полибия оказался идеальной конструкцией для воплощения Шенноном его идей по обеспечению в шифрах рассеивания и перемешивания. Перемешивание нацелено на обеспечение существенной зависимости каждой компоненты шифрблока от всех компонент соответствующего открытого блока, а за счёт рассеивания исключается возможность применения в вероятностных методах криптографического анализа статистик от небольшого числа компонент открытых и соответствующих шифрованных блоков.

Требования Шеннона по перемешиванию и рассеиванию обеспечиваются повторными произведениями двух простых не коммутирующих операций, при этом для одной операции можно ограничиться локальными перемешиваниями, а в качестве второй операции первое время использовалась даже перестановка бит шифруемого блока, отвечающая умножению на подстановочную матрицу $P \in GL(v, 2)$, где v — длина одного блока в битах.

С появлением в начале 90-х годов прошлого века разностного и линейного методов криптографического анализа (см. [2]) использование подстановочных матриц P стало нежелательным, предпочтительней использование матриц $L \in GL(v, 2)$ общего вида. Наличие у матрицы L или у L^{-1} строк либо столбцов веса 1 (с одной единицей) наследует выявленные криптографические слабости подстановочных матриц P. В то же время представляют интерес разреженные матрицы L как наиболее просто реализуемые, при этом матрицы L^{-1} (с учётом расшифрования) тоже должны быть разреженными. В этой связи возникают следующие понятия (см. [3]).

Определение 1. *Матрицу $L \in GL(v, 2)$ (рассматриваемую с точностью до независимых перестановок строк и столбцов) называем k-матрицей, если у неё и у матрицы L^{-1} в каждой строке и в каждом столбце k единиц и $v - k$ нулей. Соответствующее семейство из v подмножеств мощности k в множестве $X = \{1, \ldots, v\}$ с такой матрицей инциденций называем k-конфигурацией.*

Здесь k нечётное, иначе матрица L была бы вырожденной. Подстановочные матрицы являются 1-матрицами. При чётном v инвертирование элементов k-матрицы предоставляет $(v - k)$-матрицу. Операция сложения подмножеств $A, B \subseteq X$ по правилу $A + B = (A \cup B) \setminus (A \cap B)$ позволяет сформулировать определение эквивалентного понятию k-матрицы понятия k-конфигурации.

Определение 2. *Совокупность $\mathcal{X} \subset 2^X$ из v подмножеств мощности k в множестве X, $|X| = v$, называем k-конфигурацией, если:*

i) каждый элемент $x \in X$ принадлежит ровно k подмножествам из $\mathcal{X}$;

ii) каждый элемент $x \in X$ является суммой (как подмножество $\{x\}$) ровно k подмножеств из $\mathcal{X}$, причём каждое подмножество из $\mathcal{X}$ участвует в качестве слагаемого ровно в k таких суммах.

Результаты многих авторов о строении неразложимых k-конфигураций (когда соответствующие гиперграфы связны) имеются в [3], где, в частности, доказываются следующие две теоремы.

Теорема 1. *При любых чётном v и нечётном k, $0 < k < v$, существует неразложимая k-конфигурация. Если при нечётных v и k существует k-конфигурация, то $v \geqslant k + (1 + \sqrt{4k - 3})/2$. Для $k \leqslant 17$ и всех*

$v \geqslant k + (1 + \sqrt{4k-3})/2$ существует k-конфигурация за исключением при $k = 3$ значения $v = 7$, когда её не существует.

Теорема 2. *При каждом $v = 2w$, $w \geqslant 2$, неразложимая 3-конфигурация состоит из подмножеств в группе вычетов $\mathbb{Z}_v$ вида $\{2i, 2i+1, 2i+2\}$, $\{2i, 2i+1, 2i+3\}$, $i = 0, 1, \ldots, w-1$. Для нечётных $v \geqslant 7$ не существует неразложимых 3-матриц. Для $v = 5$ 3-конфигурация отвечает триангуляции листа Мёбиуса.*

Класс 5-конфигураций оказался существенно богаче [4]. С помощью компьютерных вычислений Комягин М. М. показал, что с точностью до комбинаторной эквивалентности среди неразложимых 5-конфигураций для $v = 9, 10, 11, 12$ имеется соответственно ровно 1, 34, 386, 71355 5-конфигураций. Неразложимые 5-конфигурации для $v = 6$ и $v = 8$ состоят из дополнений к подмножествам соответственно 1- и 3-конфигураций. Для $v = 8$ 3-конфигурация может быть как неразложимой, так и разложимой, разбиваемой на две 3-конфигурации с $v = 4$. Для $v = 7$ 5-конфигураций не существует по теореме 1. К настоящему времени усилиями Тришина А. Е. [5] и Фролова А. А. [6] получены все 5-конфигурации в виде 5-подмножеств абелевых групп, получающихся из одного параллельными сдвигами.

В известных примерах 5-конфигураций задействован весь спектр средств, привлекавшихся ранее для построения k-конфигураций [3], включая правильные многогранники, регулярные и симметрические графы, квадратичные вычеты по простому модулю, конечные группы, (v, k, λ)-конфигурации, включая конфигурации, которые отвечают совершенным разностным множествам, конечным проективным плоскостям и матрицам Адамара. В построении k-конфигураций в разное время принимали участие Брославский М. В., Зубков А. М., Комягин М. М., Красулина Е. Г., Малышев Ф. М., Сачков В. Н., Тараканов В. Е., Тришин А. Е., Фролов А. А.

Ориентируясь на максимально разреженные матрицы $L, L^{-1} \in GL(v, 2)$, возникает понятие $\{2, 3\}$-конфигурации.

Определение 3. *Совокупность $\mathcal{X} \subset 2^X$ из v подмножеств мощности либо 2 либо 3 в множестве X, $|X| = v$, называем $\{2, 3\}$-конфигурацией, если:*

i) каждый элемент $x \in X$ принадлежит либо 2, либо 3 подмножествам из $\mathcal{X}$;

ii) каждый элемент $x \in X$ является суммой (как подмножество $\{x\}$) либо 2, либо 3 подмножеств из $\mathcal{X}$, причём каждое подмножество из $\mathcal{X}$ участвует в качестве слагаемого либо в 2, либо в 3 таких суммах.

Теорема 3. *Любая неразложимая $\{2, 3\}$-конфигурация либо является одной из 3-конфигураций из теоремы 2, либо имеет матрицу инциден-*

цией $\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ *размера* 3×3 *или* $\begin{pmatrix} J & K & O & \cdots & O & O & O \\ O & J & K & \cdots & O & O & O \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ O & O & O & \cdots & O & J & K \\ K & O & O & \cdots & O & O & J \end{pmatrix}$ *размера* $2w \times 2w$, $w \geqslant 2$, *где* $J = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, $K = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$, $O = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$.

Список литературы

[1] Шеннон К. Работы по теории информации и кибернетике. М. : Издательство иностранной литературы, 1963.

[2] Малышев Ф. М. Методы линейных и разностных соотношений в криптографии // Дискретная математика. 2022. Т. 23, вып. 1. С. 36–63.

[3] Малышев Ф. М. k-конфигурации // Труды Математического института имени В. А. Стеклова. 2022. Т. 316. С. 248–269.

[4] Комягин М. М. Классификация $(v, 5)$-конфигураций для $v \leqslant 11$ // Дискретная математика. 2024. Т. 36, вып. 1. С. 46–66.

[5] Тришин А. Е. Классификация циркулянтных $(v, 5)$-матриц // Обозрение прикладной и промышленной математики. 2004. Т. 11, вып. 2. С. 258–259.

[6] Фролов А. А. Классификация неразложимых абелевых $(v, 5)$-групп // Дискретная математика. 2008. Т. 20, вып. 1. С. 94–108.

Упаковки путей в пороговых графах

Мокеев Дмитрий Борисович

Нижегородский государственный университет имени Н. И. Лобачевского; Национальный исследовательский университет «Высшая школа экономики», Нижегородский филиал; MokeevDB@gmail.com

Рассматриваются конечные обыкновенные графы, т. е. неориентированные графы $G = (V, E)$ с множеством вершин $V = V(G)$ и множеством рёбер $E = E(G)$, не содержащим петель и кратных рёбер.

Задача об упаковке k-путей в графе (о P_k-упаковке) заключается в следующем. Дан граф G. Требуется в графе G найти максимальное число путей размера k, попарно не содержащих общих вершин. Такое число является инвариантом графа и обозначается $\mu_{P_k}(G)$.

Задачи о P_k-упаковке возникают при проектировании электронных плат с помощью компьютера [1]. Известно, что задача является NP-полной для $k \geqslant 3$ для графов общего вида [2] и для субкубических графов [3]. Доказана полиномиальная разрешимость данной задачи в некоторых классах графов для частных случаев k [4, 5] и для произвольного k [3, 6].

Определение. *Граф называется* пороговым*, если может быть построен из одновершинного графа последовательным добавлением в граф одной изолированной вершины или доминирующей вершины, т. е. отдельной вершины, связанной со всеми остальными вершинами.*

Мы рассматриваем задачу о P_k-упаковке для фиксированного k в пороговых графах и их соединениях.

Упаковки путей в пороговых графах

Каждый пороговый граф является расщепляемым графом. Это значит, что его вершины можно разделить на клику C и независимое множество I. В случае, если граф G пороговый, множества I и C могут быть упорядочены $v_1, v_2, \ldots, v_q$ и $u_1, u_2, \ldots, u_p$ соответственно так, что $N(v_{i-1}) \subseteq N(v_i)$ для всех $i \in \{2, 3, \ldots, q\}$ и $N(v_i)$ состоит из последовательных вершин $u_1, u_2, \ldots, u_{p_i}$ для всех $i \in \{1, 2, \ldots, p\}$. Последовательности $v_1, v_2, \ldots, v_q$ и $u_1, u_2, \ldots, u_p$ называются совершенным упорядочением множества вершин.

Пусть G — пороговый граф с совершенно упорядоченными независимым множеством $I = (v_1, v_2, \ldots, v_q)$ и кликой $C = (u_1, u_2, \ldots, u_p)$, причём $p \geqslant k$.

Пусть k чётно и равно $2s$. Рассмотрим два случая:

1. Существуют такие $i_1, i_2, \ldots, i_s$, что $i_1 < i_2 < \cdots < i_s$ для всех $j \in \{1, 2, \ldots, q\}$ и $deg(v_x) < j$ для всех $x < i_j$. Обозначим $G^{'}$ граф, полученный из G удалением вершин $v_1, v_2, \ldots, v_{i_k}, u_1, u_2, \ldots, u_s$. Обозначим w простой путь в графе G, построенный на вершинах $v_{i_1}, u_1, v_{i_2}, u_2, \ldots, v_{i_s}, u_s$ в указанной последовательности.
2. Для некоторого t, $0 \leqslant t < s$, существуют такие $i_1, i_2, \ldots, i_t$, что $i_1 < i_2 < \cdots < i_t$, $\forall j \in \{1, 2, \ldots, t\}$: $deg(v_{i_j}) \geqslant j$ и $deg(v_x) < j$ для всех $i_{j-1} < x < i_j$, $\forall x \in \{i_t + 1, i_t + 2, \ldots, p\}$: $deg(v_x) \leqslant t$ и $p \geqslant 2s - t$. Обозначим w простой путь в графе G, построенный на вершинах $v_{i_1}, u_1, v_{i_2}, u_2, \ldots, v_{i_t}, u_t, u_{t+1}, \ldots, u_{2s-t}$ в указанной последовательности. Обозначим $G^{'}$ граф, порождённый вершинами $u_{2s-t+1}, u_{2s-t+2}, \ldots, u_r$ графа G.

Теорема 1. *Пусть $M^{'}$ — наибольшая P_{2s}-упаковка графа $G^{'}$. Тогда $M^{'} \cup \{w\}$ — наибольшая P_{2s}-упаковка графа G.*

Пусть теперь k нечётно и равно $2s+1$. Рассмотрим аналогично два случая:

1. Существуют такие $i_1, i_2, \ldots, i_s$, что $i_1 < i_2 < \cdots < i_s$ для всех $j \in \{1, 2, \ldots, q\}$ и $deg(v_x) < j$ для всех $x < i_j$. Обозначим $G^{'}$ граф, полученный из G удалением вершин $v_1, v_2, \ldots, v_{i_s}, v_{i_s+1}, u_1, u_2, \ldots, u_s$. Обозначим w простой путь в графе G, построенный на вершинах $v_{i_1}, u_1, v_{i_2}, u_2, \ldots, v_{i_s}, u_s, v_{i_s+1}$ в указанной последовательности.

2. Для некоторого t, $0 \leqslant t < s$, существуют такие $i_1, i_2, \ldots, i_t$, что $i_1 < i_2 < \cdots < i_t$, $\forall j \in \{1, 2, \ldots, t\}$: $deg(v_{i_j}) \geqslant j$ и $deg(v_x) < j$ для всех $i_{j-1} < x < i_j$, $\forall x \in \{i_t + 1, i_t + 2, \ldots, p\}$: $deg(v_x) \leqslant t$ и $p \geqslant 2s - t$ или $t = s$, $i_s = q$ и $p \geqslant s + 1$. Обозначим $G^{'}$ граф, полученный из G удалением вершин $v_1, v_2, \ldots, v_{i_s}, v_{i_s+1}, u_1, u_2, \ldots, u_s$. Обозначим w простой путь в графе G, построенный на вершинах $v_{i_1}, u_1, v_{i_2}, u_2, \ldots, v_{i_s}, u_s, v_{i_s+1}$ в указанной последовательности.

Теорема 2. *Пусть $M^{'}$ — наибольшая P_{2s+1}-упаковка графа $G^{'}$. Тогда $M^{'} \cup \{w\}$ — наибольшая P_{2s+1}-упаковка графа G.*

Отметим, что граф $G^{'}$ полный или также пороговый. Таким образом, задача о P_k-упаковке в пороговом графе сводится к задаче нахождения его совершенного упорядочения, то есть к задаче сортировки вершин множеств I и C по их степеням, после чего задача нахождения очередного элемента упаковки решается последовательным просмотром вершин.

Теорема 3. *Задача о P_k-упаковке в пороговых графах может быть решена за время $O(|G|^2)$, где $|G|$ — число вершин графа.*

Упаковки путей в соединениях пороговых графов

Соединением двух графов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$, где $V_1 \cap V_2 = \varnothing$, называется граф $G_1 \circ G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup (V_1 \times V_2))$.

Пусть G_1 и G_2 — пороговые графы с совершенно упорядоченными независимыми множествами $I_1 = (v_{1,1}, v_{1,2}, \ldots, v_{1,q_1})$ и $I_2 = (v_{2,1}, v_{2,2}, \ldots, v_{2,q_2})$ и кликами $C_1 = (u_{1,1}, u_{1,2}, \ldots, u_{1,p_1})$ и $C_2 = (u_{2,1}, u_{2,2}, \ldots, u_{2,p_2})$ соответственно.

Рассмотрим граф $G = G_1 \circ G_2$. Легко заметить, что вершины графа G можно разделить на клику $C = C_1 \cup C_2$ и биклику $B = I_1 \cup I_2$.

Без ограничений общности предположим, что $q_1 \leqslant q_2$. Если $q_1 + p_1 \geqslant q_2 - 1$, то граф G содержит гамильтонов путь. Тогда задача о P_k-упаковке решается простым разбиением этого пути на пути порядка k и, может быть, один путь порядка, меньшего k.

Иначе рассмотрим путь

$$(v_{2,1}, v_{1,1}, v_{2,2}, v_{1,2}, \ldots, v_{1,q_1}, v_{2,q_1+1}, u_{1,1}, v_{2,q_1+2}, u_{1,2}, \ldots, u_{1,p_1}, v_{2,q_1+p_1+1}).$$

Выделим в нём последовательно пути порядка k, начиная с вершины $v_{2,1}$ до тех пор, пока это возможно. Полученное множество путей обозначим M_1. Оставшиеся вершины множества B образуют независимое множество $I_2^{'} = (v_{2,q_1+p_1-t+2}, \ldots, v_{2,q_2})$, где $2t$ или $2t+1$ — остаток от деления $2(q_1+p_1)+1$ на k.

Граф $G^{'}$, полученный удалением всех вершин найденных путей, является пороговым. По теореме 3, в нём можно найти наибольшую P_k-упаковку M_2 за время $O(|G^{'}|^2)$.

Теорема 4. *$M_1 \cup M_2$ является наибольшей P_k-упаковкой графа G.*

Теорема 5. *Задача о P_k-упаковке в соединениях пороговых графов может быть решена за время $O(|G|^2)$, где $|G|$ — число вершин графа.*

Список литературы

[1] Hope A. Component placement through graph partitioning in computer-aided printed-wiring-board design // Electronics Letters. 1972. Vol. 8, no. 4. P. 87–88.

[2] Hell P., Kirkpatrick D. G. On the complexity of general graph factor problems // SIAM Journal on Computing. 1983. Vol. 12. P. 601–609.

[3] Masuyama S., Ibaraki T. Chain packing in graphs // Algorithmica. 1991. Vol. 6, no. 1. P. 826–839.

[4] Kosowski A., Malafiejski M., Żylinski P. Tighter bounds on the size of a maximum P_3-matching in a cubic graph // Graphs and Combinatorics. 2008. Vol. 24, no. 5. P. 461–468.

[5] Alekseev V. E., Mokeev D. König graphs for 3-paths and 3-cycles // Discrete Applied Mathematics. 2016. Vol. 204, P. 1–5.

[6] Malyshev D., Mokeev D. A polynomial-time algorithm of finding a minimum k-path vertex cover and a maximum k-path packing in some graphs // Optimization Letters. 2019, Vol. 14, no. 6. P. 1317–1322.

О графовых задачах, возникающих для ИИ-ассистентов редакторов схем печатных плат

Никитин Андрей Анатольевич[1], Энтина Елена Львовна[2]

1 Russian Research Institute; nika19751975@gmail.com

2 Coleman Group; elena.entina@h-partners.com

Введение

В настоящее время создание всевозможных ассистентов (или ко-пилотов), позволяющих облегчить инженерам процесс дизайна печатных плат, — одна из актуальных задач САПР. Данная публикация посвящена ассистенту для проектирования схематики печатных плат. Определяются инженерная задача, а также математические проблемы из области теории графов и области ИИ, к решению которых сводится решение исходной инженерной задачи. Целью данной публикации является приглашение заинтересованных исследовательских команд к кооперации для совместного решения представленных математических проблем.

Описание задачи

Одним из основных этапов процесса проектирования печатных плат (ПП) является редактирование схем. Работая над схемой ПП, инженер имеет дело со списком компонент ПП и соединениями между контактами разных компонент. Последовательное добавление компонент и соединение их контактов может занять значительное время, и это процесс, крайне подверженный ошибкам. Существуют отдельные группы инженеров, занимающиеся просмотром таблицы компонент и поиском соответствующих компонент. Используя технологии искусственного интеллекта (ИИ), а также схемы, созданные ранее, можно помочь инженерам разрабатывать новые схемы, предложив одну или несколько наиболее вероятных компонент и соединений контактов компонент. Подобный помощник (или ко-пилот) позволяет сократить время работы инженеров и снижает вероятность ошибок.

Наиболее естественная математическая форма для описания схемы печатной платы — гиперграфы специального вида, в которых каждое ребро соответствует некоторому соединению в схеме, и оно помечено названиями контактов компонент, участвующих в данном соединении. Предположим, что любой гиперграф нетлиста можно разбить на набор гиперграфов размера не более заданного. Тогда один из вариантов схемы работы рекомендательной системы ко-пилота: предварительно нарезать базу данных (БД) из гиперграфов ограниченного размера и ограниченного диаметра, а затем в реальном времени по каждому запросу (текущей схеме пользователя) находить в базе похожие гиперграфы.

В первом приближении задача поиска похожего гиперграфа может быть сведена к задаче нахождения похожего ориентированного псевдомультиграфа или ориентированного псевдографа, в котором каждая дуга помечена некоторым натуральным числом, означающим кратность этой дуги. Каждая вершина такого псевдографа соответствует некоторой компоненте ПП, наличие дуги от одной вершины до другой определяется существованием или отсутствием соединения между соответствующими компонентами ПП, а кратность дуги — количеством таких соединений.

Также в первом приближении можно считать, что задача определения сходства ставится для двух ориентированных псевдографов с равным количеством вершин. В более общем случае, когда один из графов имеет большее количество вершин, требуется находить максимально близкий подграф этого графа с числом вершин, равным числу вершин второго графа.

Итак, пусть $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ — два ориентированных псевдографа, которые имеют одинаковое число вершин $n : |V_1| = |V_2| = n$. Пусть дана мера близости между вершинами двух графов $d : V_1 \times V_2 \to [0; 1]$. Пусть c — это функция кратности рёбер из E_1 и E_2, то есть $c : E_1 \cup E_2 \to \mathbb{N}$, где $\mathbb{N}$ —

множество натуральных чисел. Без ограничения общности будем считать, что оба графа являются полными, а функция c принимает целые неотрицательные значения, то есть $c : E_1 \cup E_2 \to \mathbb{Z}^+$, где значение 0 означает отсутствие дуги в оригинальном псевдографе.

Произвольное взаимно однозначное отображение $m : V_1 \leftrightarrow V_2$ назовем непротиворечивым тогда и только тогда, когда $c(v_1, v_2) \leqslant c(m(v_1), m(v_2))$, $\forall v_1, v_2 \in V_1$. Пусть $M(G_1, G_2)$ — это множество всех непротиворечивых отображений $m : V_1 \leftrightarrow V_2$.

Ценой отображения m назовем следующую величину:

$$dist(G_1, G_2, m) = \sum_{v \in V_1} d(v, m(v)).$$

В качестве меры близости двух псевдографов G_1 и G_2 выберем

$$dist(G_1, G_2) = \max_{m \in M(G_1, G_2)} dist(G_1, G_2, m).$$

Задача вычисления значения $dist(G_1, G_2)$ и нахождения непротиворечивого отображения m, на котором достигается максимум суммы, является NP-трудной задачей. На практике требуется решать аппроксимационную задачу, где необходимо построить алгоритм, работающий с минимально возможной (например, полиномиальной) сложностью и находящий хорошее решение с заданной вероятностью p, $p \in [0.85; 0.95]$, и заданной точностью ∂, $\partial \in [0.9; 0.95]$.

Определяется набор исходных данных $PAIRS = \{(G_1, G_2)\}$, состоящий из большого числа пар, для которого необходимо построить такой алгоритм ALG минимально возможной сложности, который для пары $(G_1, G_2) \in PAIRS$ находит такое непротиворечивое отображение $ALG(G_1, G_2)$, что

$$\frac{|(G_1, G_2) \in PAIRS,\ dist(G_1, G_2, ALG(G_1, G_2))) \geqslant \partial \times dist(G_1, G_2)|}{|PAIRS|} \geqslant p,$$

где запись $|X|$ обозначает мощность множества X.

Данная проблема сталкивается с другими известными проблемами теории графов.

Связанные проблемы

Поиск изоморфного подграфа. Если при построении алгоритма ALG не учитывать информацию о вершинах, то задача сведётся к известной NP-трудной проблеме изоморфизма подграфов. В нашем случае выбор диаметра графов в $PAIRS$ и других гиперпараметров, возможно, позволяет построить алгоритм, решающий задачу за время $O(n^3)$. У известного алгоритма VF2 [1] время работы $\Theta(n^2)$ в лучшем случае и $\Theta(n! \cdot n)$ в худшем. Случаи n^2

удовлетворяют практическим нуждам, но случаи с $n! \cdot n$ могут радикально увеличить среднее время работы алгоритма на $PAIRS$. Если процент таких сложных пар не больше $(1 - p)$, можно пропускать часть кандидатов, модифицировав VF2 для задачи inexact subgraph matching'a.

Существует много прикладных подходов к решению задачи изоморфизма подграфов, гарантии для таких алгоритмов считают отдельно на разных классах входных графов. Один из них, PathLAD+ [2], использует зондирующий поиск и сложную стадию фильтрации. Дают ли эти стадии или особенности других подходов преимущество в описанной выше задаче и можно ли построить на их основе удовлетворяющие практическим нуждам алгоритмы — вопрос исследования.

Задача о назначениях с ограничениями. Другой случай, когда поначалу можно целиком игнорировать информацию о связях, но максимально полно учесть информацию о степени похожести вершин графов. Задача сведётся к задаче о назначениях. Линейная задача о назначениях обычно решается одной из модификаций венгерского алгоритма за время $O(n^3)$ или даже за время $O(n^{2.5} \cdot \log_2(W))$, когда стоимость назначения получается задать целыми весами в отрезке $[0, W]$ [3]. Описанная выше задача может быть сведена к такому случаю преобразованием, сопоставляющим d с набором дискретных значений. Кроме того, показано, что для некоторых графов можно найти решение за линейное время [4], однако это требует очень жёстких (и непрактичных) ограничений на функцию стоимости назначения.

Для учета связей в таком подходе можно, например, устанавливать, что некоторые вершины графа G_1 не могут быть назначены в часть вершин графа G_2, то есть решать задачу о назначениях с ограничениями. Ограничения можно вводить итеративно. В пределе, когда все связи таким образом будут учтены, задача снова сведётся к поиску изоморфизма графов, теперь уже с учетом стоимости замены вершин. Аналогично предыдущему пункту, для практических нужд подходит решение, возможно пропускающее часть правильных ответов в угоду скорости при заданных выше ограничениях на качество.

Расстояние редактирования графа. Ещё один способ подходить к поиску «похожих» графов — введение некоторой метрики близости. Первым естественным решением будет редакторское расстояние (GED, см., например, [5]) — сколько операций удаления, замены и добавления (с разными стоимостями) вершин/рёбер нужно сделать, чтобы преобразовать один граф в другой. Вычисление GED предполагает нахождение пути редактирования с минимальной суммарной стоимостью. Задачу вычисления оптимального пути можно свести к задаче поиска кратчайшего пути, зачастую она решается алгоритмами типа A* (использует эвристику). Кроме того, для GED возможно построить аппроксимации, вычисляемые за линейное время [6]. Удовлетво-

ряют ли аппроксимации конкретным практическим требованиям и можно ли построить другие удобные метрики близости — также вопрос изучения.

Приглашаем к сотрудничеству все заинтересованные организации для решения вышеописанной задачи.

Список литературы

[1] A (sub)graph isomorphism algorithm for matching large graphs / L. P. Cordella, P. Foggia, C. Sansone, M. Vento // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. Vol. 26. P. 1367–1372.

[2] PathLAD+: An improved exact algorithm for subgraph isomorphism problem / Yiyuan Wang, Chenghou Jin, Shaowei Cai, Qingwei Lin // Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23). California, USA : International Joint Conferences on Artificial Intelligence Organization, 2023. P. 5639–5647.

[3] Duan Ran, Su Hsin-Hao. A scaling algorithm for maximum weight matching in bipartite graphs // Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete algorithms. Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 2012. P. 1413–1424.

[4] Computing optimal assignments in linear time for approximate graph matching / N. M. Kriege, P.-L. Giscard, F. Bause, R. C. Wilson // 2019 IEEE International Conference on Data Mining (ICDM). Washington, DC, USA : IEEE Computer Society, 2019. P. 349–358.

[5] An efficient algorithm for graph edit distance computation / Xiaoyang Chen, Hongwei Huo, Jun Huan, J. S. Vitter // Knowledge-Based Systems. 2019. Vol. 163. P. 762–775.

[6] Santacruz P., Serratosa F. Error-tolerant graph matching in linear computational cost using an initial small partial matching // Pattern Recognition Letters. 2020. Vol. 134. P. 10–19.

Импликативное замыкание на множестве мультиопераций

Пантелеев Владимир Иннокентьевич

Иркутский государственный университет;
Бурятский государственный университете имени Доржи Банзарова; vl.panteleyev@gmail.com

В теории дискретных операций достаточно давно изучаются частичные, гипер- и мультиоперации — операции, заданные на конечном множестве A и принимающие в качестве своих значений подмножества (с некоторыми ограничениями или без) этого множества.

Одно из важнейших направлений в исследовании дискретных операций — классификация. Широко распространенным при этом является подход, ос-

нованный на операторах замыкания. Классы, замкнутые относительно рассматриваемых операторов, образуют соответствующую классификацию рассматриваемого множества операций.

Наиболее известной является классификация, основанная на операторе суперпозиции. Однако классификация, основанная только на суперпозиции, даже для частичных, гипер- и мультиопераций заданных на двухэлементном множестве приводит к континууму замкнутых множеств.

Попытки сократить континуальные классификации приводят к необходимости изучения более сильных операторов замыкания.

Одним из таких является оператор параметрического замыкания, предложенный А. В. Кузнецовым [1]. В [2] дан обзор операторов, являющихся расширением оператора параметрического замыкания. В рамках этого подхода можно выделить оператор импликативного замыкания. Действие оператора импликативного замыкания на множестве частичных функций (операций) рассматривается в [3]. Мы рассматриваем оператор мультипликативного замыкания на множестве мультиопераций.

Пусть $E_k = \{0, 1, \ldots, k-1\}$. Для произвольного конечного множества A через $|A|$ обозначим мощность, а через $\mathcal{B}(A)$ — множество всех подмножеств.

Для целого положительного n отображение $f : E_k^n \to \mathcal{B}(E_k)$ назовем n-местной мультиоперацией ранга k. Множество всех n-местных мультиопераций обозначим как $\mathcal{M}^n$, а множество всех мультиопераций ранга k как $\mathcal{M}_k$. Множество всех мультиопераций ранга k содержит в себе множество операций $\mathcal{O}_k$ и множество частичных операций $\mathcal{O}_k^*$.

Пусть f_0 — n-местная мультиоперация, $f_1, \ldots, f_n$ — m-местные мультиоперации. Суперпозиция с внешней операцией f_0 и внутренними операциями $f_1, \ldots, f_n$ определяет m-местную мультиоперацию $s(f_0, f_1, \ldots, f_n)$ следующим образом: для набора $(\alpha_1, \ldots, \alpha_m) \in E_k^m$ по определению*

$$s(f_0, f_1, \ldots, f_n)(\alpha_1, \ldots, \alpha_m) = \bigcup_{\beta_i \in f_i(\alpha_1, \ldots, \alpha_m)} f_0(\beta_1, \ldots, \beta_n).$$

Если в последовательности $\beta_1, \ldots, \beta_n$ некоторое $\beta_i = \varnothing$, то $f_0(\beta_1, \ldots, \beta_n)$ тоже равно $\varnothing$.

Пусть A — некоторое множество мультиопераций. Определим понятие «мультиоперация над A»:

— если $f \in A$, то f — мультиоперация над A;

— если f — мультиоперация над A, то операция, полученная из f перестановкой или отождествлением аргументов, является мультиоперацией над A;

*Определение суперпозиции мультиопераций позволяет находить значение мультиопераций не только на двоичных наборах. Подробно можно посмотреть в [4].

— если f_0 — n-местная и f_i — m-местная мультиоперации над A или мультиоперации-переменные ($i \in \{1, \ldots, n\}$), то мультиоперация $s(f_0, f_1, \ldots, f_n)$ является мультиоперацией над A.

Язык импликативного замыкания Imp использует формулы логики предикатов, построенные с помощью логических связок конъюнкции, импликации и квантора существования, а также отношения равенства термов. С точными определениями можно ознакомиться, например, в [3]. Для изучения мультиопераций вместо отношения равенства мы будем использовать отношение включения: элементарной формулой будем называть выражение $(t_1 \subseteq t_2)$, где t_1, t_2 — термы. При этом, для упрощения записи, мы не будем разделять одноэлементное множество и элемент этого множества, если это не вызывает недоразумений.

Пусть $Q \subseteq M_k$, $f(x_1, \ldots, x_n) \in M_k$, $\Phi(x_1, \ldots x_n, y)$ — формула языка Imp со свободными переменными $x_1, \ldots x_n, y$, все функциональные символы которой являются обозначениями мультиопераций над Q. Будем говорить, что формула Φ импликативно выражает операцию $f(x_1, \ldots, x_n)$ через операции множества Q, если множества истинности формулы Φ и отношения $y \subseteq f(x_1, \ldots, x_n)$ совпадают. Множество всех операций, импликативно выразимых через операции множества Q, назовем импликативным замыканием множества Q и обозначим $\mathrm{Imp}[Q]$. Множество Q, которое совпадает со своим импликативным замыканием, называется импликативно замкнутым классом. Через $[Q]$ обозначаем замыкание с оператором суперпозиции.

Утверждение 1. *Любой имплликативно замкнутый класс мультиопераций замкнут относительно операции суперпозиции.*

Утверждение 2. *Для любого множества $Q \subseteq \mathcal{O}_k$ справедливо: если $[Q] = \mathcal{O}_k$, то $\mathrm{Imp}[Q] = \mathcal{M}_k$.*

Утверждение 3. *Мультиоперации $f_1(x) = x$, $f_2(x) \equiv E_k$, $f_3(x) \equiv \varnothing$ и*

$$p(x_1, x_2, x_3) = \begin{cases} x_3, & \text{если } x_1 = x_2; \\ x_1 & \text{иначе}; \end{cases}$$

содержатся в каждом импликативно замкнутом классе мультиопераций.

Пусть π — подстановка на множестве E_k, положим также, что $\pi(\varnothing) = \varnothing$. Через S_π обозначим множество всех мультиопераций для которых π — эндоморфизм, т. е. $\pi f(x_1, \ldots, x_n) = f(\pi(x_1), \ldots, \pi(x_n))$.

Теорема 1. *В M_3 все импликативно предполные классы исчерпываются классами вида S_π, где π — нетождественная подстановка на E_3.*

Исследование выполнено за счет гранта Российского научного фонда № 24-21-00011 в Бурятском государственном университете им. Д. Банзарова, https://rscf.ru/project/24-21-00011.

Список литературы

[1] Кузнецов А. В. О средствах для обнаружения невыводимости и невыразимости // Логический вывод. М. : Наука, 1979. С. 5–33.

[2] Марченков С. С. Логические расширения оператора параметрического замыкания // Дискретная математика. 2022. Т. 34, вып. 3. С. 52–62.

[3] Марченков С. С. О действии оператора импликативного замыкания на множестве частичных функций многозначной логики // Дискретная математика. 2020. Т. 32, вып. 1. С. 60–73

[4] Пантелеев В. И., Тагласов Э. С. ES_I-замыкание мультиопераций ранга 2: критерий полноты, классификация и типы базисов // Интеллектуальные системы. Теория и приложения. 2021. Т. 25, вып. 2. С. 55–80.

О некоторых SI^*-замкнутых классах мультиопераций ранга 2

Пантелеев Владимир Иннокентьевич[1], Фомина Ирина Владимировна[2]

1 Иркутский государственный университет;
Бурятский государственный университете имени Доржи Банзарова; vl.panteleyev@gmail.com
2 Бурятский государственный университете имени Доржи Банзарова; fomina-irina0104@yandex.ru

Рассматриваем мультиоперации ранга 2, т. е. мультиоперации, заданные на множестве $E = \{0, 1\}$. Множество $\mathcal{M}_2$ мультиопераций ранга 2 содержит в себе множество операций, частичных и гипероперации. Оператор суперпозиции для мультиопераций можно определить неоднозначно [1]. Основные подходы в определениях основаны на теоретико-множественных операциях объединения и пересечения.

Основной проблемой при классификации мультиопераций относительно суперпозиции является континуальность множества классов, замкнутых относительно суперпозиции. В связи с этим традиционной является задача описания некоторой части решетки замкнутых классов. Интерес вызывают интервалы, начинающиеся с множества операций [2].

Будем считать известными такие понятия, как клон, мультиклон, сохранение предиката функцией (операцией). Предикат будем задавать матрицей, в которой столбцами являются наборы из предиката.

Суперпозиция $s(f_0, f_1, \ldots, f_m)$ с внешней f_0 и внутренними мультиоперациями $f_1, \ldots, f_m$ (SI-суперпозиция), основанная на пересечении, определяется

следующим образом. Если $(a_1, \ldots, a_n) \in E^n$, то по определению

$$s(f_0, f_1, \ldots, f_m) = \begin{cases} \varnothing, \text{если найдется } i \text{ из } \{1, \ldots, m\} : f_i(a_1, \ldots, a_n) = \varnothing; \\ \bigcap\limits_{b_i \in f_i(a_1, \ldots, a_n)} f(b_1, \ldots, b_m), \text{ если оно не пусто;} \\ \bigcup\limits_{b_i \in f_i(a_1, \ldots, a_n)} f(b_1, \ldots, b_m) \quad \text{иначе.} \end{cases}$$

Пусть S — множество самодвойственных операций (множество самодвойственных булевых функций). Справедлива

Теорема 1 ([1]). *Интервал $I(S, \mathcal{M}_2)$ содержит ровно* 17 *различных мультиклонов.*

Будем рассматривать суперпозицию, основанную на пересечении, но интерпретируя при этом пустое множество как «поломку» (SI^*-суперпозицию). Если $(a_1, \ldots, a_n) \in E^n$, то по определению

$$s(f_0, f_1, \ldots, f_m) = \begin{cases} \varnothing, \text{если найдется } i \text{ из } \{1, \ldots, n\} : f_i(a_1, \ldots, a_n) = \varnothing \\ \text{или найдется набор } (b_1, \ldots, b_m), \text{ где } b_i \in f_i(a_1, \ldots, a_n) \\ \text{для которого } f_0(b_1, \ldots, b_m) = \varnothing; \\ \bigcap\limits_{b_i \in f_i(a_1, \ldots, a_n)} f(b_1, \ldots, b_m), \text{ если оно не пусто;} \\ \bigcup\limits_{b_i \in f_i(a_1, \ldots, a_n)} f(b_1, \ldots, b_m) \quad \text{иначе.} \end{cases}$$

Это определение позволяет вычислять значение мультиоперации на любом наборе $(a_1, \ldots, a_m) \in (2^E)^m$.

В [3, 4] описаны классы S_1, S_2, S', K_5. Определим дополнительно следующие множества мультиопераций:

— S_2^- — класс мультиопераций, сохраняющих предикат $\begin{pmatrix} 0 & 1 & - \\ 1 & 0 & - \end{pmatrix}$;

— S_2^{*-} — класс мультиопераций, сохраняющих предикат $\begin{pmatrix} 0 & 1 & - & * \\ 1 & 0 & - & * \end{pmatrix}$;

— K_1 — множество, состоящее из всех мультиопераций f таких, что для любого двоичного набора $\tilde{\alpha}$ мультиоперация f возвращает $\varnothing$ на этом наборе или противоположном;

— K_2 — класс мультиопераций, сохраняющих следующий предикат:

$$\begin{pmatrix} 0 & 1 & - & * & * & 0 & * & 1 & * & - \\ 1 & 0 & - & * & 0 & * & 1 & * & - & * \end{pmatrix}.$$

Справедливы следующие утверждения.

Утверждение 1. *Для любой мультиоперации* $f \in \bar{S^*} \setminus (S^- \cup \{*\})$ *справедливо* $\bar{S^*} = [S^- \bigcup \{f\}]$, *здесь* $\{*\}$ — *множество всех операций от любого числа переменных, тождественно равных* $\varnothing$.

Утверждение 2. *Класс* K_2 *является предполным в* M_2.

Утверждение 3. *Для любой операции* $f \in K_2 \setminus (\bar{S^*} \cup K_1)$ *справедливо* $K_2 = [\bar{S^*} \bigcup K_1 \bigcup \{f\}]$.

Утверждение 4. *Для любой операции* $f \in K_2 \setminus K_5$ *справедливо* $K_2 = [K_5 \bigcup \{f\}]$.

Утверждение 5. *Для любой операции* $f \in K_5 \setminus S'$ *справедливо* $K_5 = [S' \bigcup \{f\}]$.

Теорема 2. *Интервал* $I(S, \mathcal{M}_2)$ *содержит ровно* 19 *различных мультиклонов, а именно мультиклоны, представленные на рис. 1.*

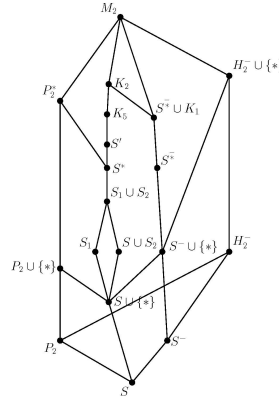

Рис. 1: Интервал $I(S, \mathcal{M}_2)$

Исследование выполнено за счет гранта Российского научного фонда № 24-21-00011 в Бурятском государственном университете им. Д. Банзарова, https://rscf.ru/project/24-21-00011.

Список литературы

[1] Пантелеев В. И., Тагласов Э. С. ES_I-замыкание мультиопераций ранга 2: критерий полноты, классификация и типы базисов // Интеллектуальные системы. Теория и приложения. 2021. Т. 25, вып. 2. С. 55–80.

[2] A solution to a problem of D. Lau: complete classification of intervals in the lattice of partial Boolean clones /M. Lamsade, K. Sholzel, L. Haddad, T. Waldhauzer // 2013 IEEE 43rd International Symposium on Multiple-Valued Logic. Washington, DC, USA : IEEE Computer Society, 2013. P. 123–128.

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

[4] О двух интервалах в решетке частичных ультраклонов ранга 2 / С. А. Бадмаев, А. Е. Дугаров, И. В. Фомина, И. К. Шаранхаев // Сибирские электронные математические известия. 2023. Т. 20, № 1. С. 262–274.

Об исчислении мультиопераций

Перязев Николай Алексеевич

Иркутский государственный университет; Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» имени В. И Ульянова (Ленина); nikolai.baikal@gmail.com

Язык мультиопераций

Пусть A — множество, $B(A)$ — множество всех подмножеств A. Тогда $g^n : A^n \to B(A)$ — n-местная мультиоперация.

Алфавит языка мультиопераций определяется следующим образом. Сигнатура F — это множество мультифункциональных символов с зафиксированной размерностью (местностью); при этом выделенными нульместными символами будут: o, c_i; $\neg, \&, \vee, \to$ — логические символы; $\subseteq$ — внелогический символ; $(,)$ — технические символы.

Определение терма:

— любой нульместный символ сигнатуры F является термом;

— $f^n(t_1, \ldots, t_n)$ — терм, где $f^n \in F$ и $t_1, \ldots, t_n$ термы.

Определение формулы:

— $(t_1 \subseteq t_2)$ — формула, где t_1, t_2 термы;

— $\neg\Phi, (\Phi\&\Psi), (\Phi \vee \Psi), (\Phi \to \Psi)$ — формулы, где Φ, Ψ формулы.

Введем семантику языка мультиопераций. Интерпретация сигнатуры F в множество A — это отображение $\gamma : F \to M$ в множество M мультиопераций на A, сохраняющее местность мультифункционального символа и мультиоперации, при этом $\gamma(o) = \varnothing$, $\gamma(c_i)$ — одноэлементные множества.

Значение $\gamma[t]$ терма t при интерпретации γ:

— если $t \equiv f^0$, то $\gamma[t] = \gamma(f^0)$;

— если $t \equiv f^n(t_1, \ldots, t_n)$, то $\gamma[t] = \bigcup\limits_{a_i \in \gamma[t_i]} \{a \mid a \in \gamma(f^n)(a_1, \ldots, a_n)\}$.

В дальнейшем ограничимся интерпретациями на конечные множества.

Двузначная семантика языка мультиопераций:

— если $\Phi \equiv t_1 \subseteq t_2$, то $\delta[\Phi] = 1$ при $\gamma[t_1] \subseteq \gamma[t_2]$,
$\delta[\Phi] = 0$ в противном случае;

— если $\Phi \equiv \neg\Psi$, то $\delta[\Phi] = 1 - \delta[\Psi]$;

— если $\Phi \equiv \Psi_1 \,\&\, \Psi_2$, то $\delta[\Phi] = \min\{\delta[\Psi_1], \delta[\Psi_2]\}$;

— если $\Phi \equiv \Psi_1 \vee \Psi_2$, то $\delta[\Phi] = \max\{\delta[\Psi_1], \delta[\Psi_2]\}$;

— если $\Phi \equiv \Psi_1 \to \Psi_2$, то $\delta[\Phi] = \max\{1 - \delta[\Psi_1], \delta[\Psi_2]\}$.

Алгоритм для вычислений в этой семантике разработан в [1].

Первый пункт приведенного определения можно обобщить следующим образом. Обобщенная семантика языка мультиопераций:

— если $\Phi \equiv t_1 \subseteq t_2$, то $\delta[\Phi] = \dfrac{|\gamma[t_1] \cap \gamma[t_2]|}{|\gamma[t_1]|}$ при $|\gamma[t_1]| > 0$,
$\delta[\Phi] = 1$ при $|\gamma[t_1]| = 0$.

Пусть $\lesseqgtr \in \{\leqslant, <, \geqslant, >\}$. Тогда если при любой интерпретации верно $\delta[\Phi] \lesseqgtr \alpha$, то говорим, что $\Phi \lesseqgtr \alpha$ тождественно выполняется в обобщенной семантике.

Исчисление мультиопераций табличного типа

Отмеченные формулы — это выражения вида $\Phi \leqslant \alpha$, $\Phi < \alpha$, $\Phi \geqslant \alpha$, $\Phi > \alpha$, где α — рациональное число от 0 до 1. Пусть $\lesssim \in \{\leqslant, <\}$; $\gtrsim \in \{\geqslant, >\}$.

Правила построения таблиц:

$(\neg\gtrsim)$ $\dfrac{\neg\Phi \gtrsim \alpha}{\Phi \lesssim 1 - \alpha}$ $\qquad$ $(\neg\lesssim)$ $\dfrac{\neg\Phi \lesssim \alpha}{\Phi \gtrsim 1 - \alpha}$

$(\&\gtrsim)$ $\dfrac{\Phi \,\&\, \Psi \gtrsim \alpha}{\begin{array}{c}\Phi \gtrsim \alpha \\ \Psi \gtrsim \alpha\end{array}}$ $\qquad$ $(\&\lesssim)$ $\dfrac{\Phi \,\&\, \Psi \lesssim \alpha}{\Phi \lesssim \alpha \mid \Psi \lesssim \alpha}$

$(\vee\gtrsim)$ $\dfrac{\Phi \vee \Psi \gtrsim \alpha}{\Phi \gtrsim \alpha \mid \Psi \gtrsim \alpha}$ $\qquad$ $(\vee\lesssim)$ $\dfrac{\Phi \vee \Psi \lesssim \alpha}{\begin{array}{c}\Phi \lesssim \alpha \\ \Psi \lesssim \alpha\end{array}}$

$(\to\gtrsim)$ $\dfrac{\Phi \to \Psi \gtrsim \alpha}{\Phi \lesssim 1 - \alpha \mid \Psi \gtrsim \alpha}$ $\qquad$ $(\to\lesssim)$ $\dfrac{\Phi \to \Psi \lesssim \alpha}{\begin{array}{c}\Phi \gtrsim 1 - \alpha \\ \Psi \lesssim \alpha\end{array}}$

$(\gtrsim)$ $\dfrac{t_1 \subseteq t_2 \gtrsim \alpha}{t_1 \subseteq t_2 > \beta}$ при $\alpha > \beta$ $\qquad$ $(\lesssim)$ $\dfrac{t_1 \subseteq t_2 \lesssim \alpha}{t_1 \subseteq t_2 < \beta}$ при $\alpha < \beta$

$$(>0)\ \frac{t_1 \subseteq t_2 > 0}{t_2 \subseteq t_1 > 0} \qquad (\leqslant 0)\ \frac{t_1 \subseteq t_2 \leqslant 0}{t_2 \subseteq t_1 \leqslant 0}$$

$$(\subseteq\gtrsim)\ \frac{t_i \subseteq s_i \geqslant 1}{t \subseteq f(t_1, \ldots, t_i, \ldots, t_n) \gtrsim \alpha \over t \subseteq f(t_1, \ldots, s_i, \ldots, t_n) \gtrsim \alpha} \qquad (\subseteq\lesssim)\ \frac{s_i \subseteq t_i \geqslant 1}{t \subseteq f(t_1, \ldots, t_i, \ldots, t_n) \lesssim \alpha \over t \subseteq f(t_1, \ldots, s_i, \ldots, t_n) \lesssim \alpha}$$

Разобьем все введенные правила на три группы:

1) $(\gtrsim)$, $(\lesssim)$;
2) $(\&\lesssim)$, $(\vee\gtrsim)$, $(\to\gtrsim)$;
3) все остальные правила.

Для множество отмеченных формул Σ построение таблицы (дерева) определяется по индукции:

1. Φ_0 — корень дерева, где $\Phi_0 \in \Sigma$.
2. D — дерево, $\Phi_0, \ldots, \Phi_i, \ldots, \Phi_n$ — ветка дерева:
   - если Ψ_i из Σ, то к ветке добавляется одна последовательная вершина, нумерованная этой формулой;
   - если к Φ_i применяется правило из первой группы, то к ветке добавляется одна последовательная вершина, нумерованная согласно примененному правилу, где β встречается в качестве отметки в формулах этой ветки;
   - если к Φ_i применяется правило из второй группы, то к ветке добавляются две вершины одного уровня, нумерованные согласно примененному правилу.
   - если к Φ_i применяется правило из третьей группы, то к ветке добавляются одна или две последовательные вершины, нумерованные согласно примененному правилу.

Ветка *замкнутая*, если содержит либо $\Phi \lesssim \alpha$ и $\Phi > \alpha$, либо $\Phi < \alpha$ и $\Phi \gtrsim \alpha$, либо $\Phi > 1$, либо $\Phi < 0$ хоть для одной формулы Φ, либо $t \subseteq t < 1$, либо $t \subseteq o > 0$ хоть для одного терма t. Применение правила является *избыточным*, если хоть в одной из полученных веток нет новых формул. Ветка *финальная*, если либо она замкнута, либо применение правил ко всем ее формулам избыточно. Если все ветки таблицы финальные, то таблица *финальная*, а если замкнутые, то таблица *замкнутая*. Множество отмеченных формул Σ является *опровержимым* в исчислении мультиопераций табличного типа, если существует замкнутая таблица для множества Σ. Отмеченная формула $\Phi \lesssim \alpha$ *выводима* из множества отмеченных формул Σ в исчислении мультиопераций, если множество $\Sigma \cup \{\Phi \not\lesssim \alpha\}$ является опровержимым.

Доказано, что приведенное исчисление является корректным и полным для обобщенной семантике языка мультиопераций.

Теорема 1. *Для формулы* Φ *языка мультиопераций тождественно выполняется* $\Phi \leqslant \alpha$ *в обобщенной семантике тогда и только тогда, когда отмеченная формула* $\Phi \leqslant \alpha$ *выводима в исчислении мультиопераций.*

Список литературы

[1] Peryazev N. A. Systems of inclusions with unknowns in multioperations // Известия Иркутского государственного университета. Серия «Математика». 2021. Т. 38. С. 112–123.

О единичных тестах для схем в базисе Жегалкина при произвольных константных неисправностях элементов

Попков Кирилл Андреевич
Институт прикладной математики имени М. В. Келдыша РАН; kirill-formulist@mail.ru

Рассматривается задача синтеза легкотестируемых схем, реализующих заданные булевы функции (см. [1]). Пусть имеется схема из функциональных элементов S с одним выходом, реализующая булеву функцию $f(\tilde{x}^n)$, где $\tilde{x}^n = (x_1, \ldots, x_n)$. Под воздействием некоторого источника неисправностей один или несколько элементов схемы S могут перейти в неисправное состояние. В результате данная схема вместо исходной функции $f(\tilde{x}^n)$ будет реализовывать некоторую булеву функцию $g(\tilde{x}^n)$, вообще говоря, отличную от f. Все такие функции $g(\tilde{x}^n)$ называются *функциями неисправности* схемы S.

Введём следующие определения [1]. *Проверяющим тестом* для схемы S называется такое множество T наборов значений переменных $x_1, \ldots, x_n$, что для любой отличной от $f(\tilde{x}^n)$ функции неисправности $g(\tilde{x}^n)$ схемы S в T найдётся набор $\tilde{\sigma}$, на котором $f(\tilde{\sigma}) \neq g(\tilde{\sigma})$. *Диагностическим тестом* для схемы S называется такое множество T наборов значений переменных $x_1, \ldots, x_n$, что T является проверяющим тестом и, кроме того, для любых двух различных функций неисправности $g_1(\tilde{x}^n)$ и $g_2(\tilde{x}^n)$ схемы S в T найдётся набор $\tilde{\sigma}$, на котором $g_1(\tilde{\sigma}) \neq g_2(\tilde{\sigma})$. Число наборов в T называется *длиной* теста. В качестве тривиального диагностического (и проверяющего) теста длины 2^n для схемы S всегда можно взять множество, состоящее из всех двоичных n-разрядных наборов. Тест называется *полным*, если в схеме могут быть неисправны сколько угодно элементов, и *единичным*, если в схеме может быть неисправен только один элемент. Единичные тесты обычно рассматривают для *неизбыточных схем* (см. [1, с. 110–111]), в которых любая допустимая неисправность любого одного элемента приводит к функции неисправности, отличной от исходной функции, реализуемой данной схемой.

Пусть зафиксирован вид неисправностей элементов, B — произвольный функционально полный базис и T — единичный проверяющий тест (ЕПТ) для

некоторой схемы S в базисе B. Введём следующие обозначения: $D^B_{\text{ЕП}}(T)$ — длина теста T; $D^B_{\text{ЕП}}(S) = \min D^B_{\text{ЕП}}(T)$, где минимум берётся по всем ЕПТ T для схемы S; $D^B_{\text{ЕП}}(f) = \min D^B_{\text{ЕП}}(S)$, где минимум берётся по всем неизбыточным схемам S в базисе B, реализующим функцию f; $D^B_{\text{ЕП}}(n) = \max D^B_{\text{ЕП}}(f)$, где максимум берётся по всем булевым функциям f от n переменных, для которых определено значение $D^B_{\text{ЕП}}(f)$. По аналогии с функциями $D^B_{\text{ЕП}}$ можно ввести функции $D^B_{\text{ЕД}}$ для единичного диагностического теста (ЕДТ), зависящие от T, от S, от f и от n. Величины $D^B_{\text{ЕП}}(n)$ и $D^B_{\text{ЕД}}(n)$ называются *функциями Шеннона* длины ЕПТ и ЕДТ соответственно.

Класс допустимых неисправностей функциональных элементов ограничим константными неисправностями на выходах элементов, при которых значение на выходе любого неисправного элемента становится равно некоторой булевой константе. Неисправности на выходах элементов называются *однотипными константными типа p*, если эта константа одна и та же для каждого неисправного элемента и равна p, и *произвольными константными*, если эта константа может быть равна как 0, так и 1 для каждого неисправного элемента независимо от неисправностей других элементов. Вполне разумно предполагать, что если в базисе содержится булева константа α, то у элемента, её реализующего, нет входов и не может быть константной неисправности типа α на его выходе.

При рассмотрении произвольных константных неисправностей на выходах функциональных элементов ранее были получены следующие результаты. В [1, с. 116, теорема 10] с использованием метода синтеза схем, предложенного С. М. Редди [2], для базиса Жегалкина $B_1 = \{\&, \oplus, 1, 0\}$ установлено, что $D^{B_1}_{\text{ЕП}}(n) \leqslant n + 3$ при $n \geqslant 0$. Д. С. Романов в [3] для любого функционально полного базиса B получил оценку $D^B_{\text{ЕП}}(n) \leqslant 4$ (правда, в указанной работе использовалось несколько другое определение неизбыточных схем). Им же совместно с Е. Ю. Романовой в [4] установлено неравенство $D^{B_1}_{\text{ЕД}}(n) \leqslant 22$, а также доказано существование базиса B_2, состоящего из булевых функций от не более чем девяти переменных, для которого $D^{B_2}_{\text{ЕД}}(n) \leqslant 6$. В работе [5], в частности, получены нижние оценки $D^B_{\text{ЕП}}(n) \geqslant 3$ при $n \geqslant 3$ для любого полного базиса B, состоящего из булевых функций от не более чем двух переменных, а также, возможно, из некоторых других булевых функций специального вида и не содержащего констант, и $D^B_{\text{ЕД}}(n) \geqslant 3$ для любого полного конечного базиса B при n, большем максимального числа существенных переменных у функций из B. В [6] установлены равенства $D^{B_3}_{\text{ЕП}}(n) = 2$ при $n \geqslant 1$ для базиса $B_3 = \{x\&y, \overline{x}, x \oplus y \oplus z\}$ и $D^{B_4}_{\text{ЕД}}(n) = 3$ при $n \geqslant 2$ для некоторого базиса B_4, состоящего из одной булевой функции от шести переменных; в [7] для того же базиса B_3 доказано, что $D^{B_3}_{\text{ЕД}}(n) \leqslant 4$ при $n \geqslant 1$.

При рассмотрении однотипных константных неисправностей типа p на выходах элементов Ю. В. Бородина нашла точное значение функции Шеннона

$D^{B_1}_{\text{ЕП}}(n) = 1$ в случаях $p = 1$ [8] и $p = 0$ [9] (совместно с П. А. Бородиным), где $n \in \mathbb{N}$; в работах [10, 7] установлены соотношения $D^{B_1}_{\text{ЕД}}(n) = 2$ при $n \geqslant 2$ в случае $p = 0$ и $D^{B_1}_{\text{ЕД}}(n) \leqslant 3$ при $n \geqslant 0$ в случае $p = 1$ соответственно.

Введём обозначения $(\tilde{0}^n) = (\underbrace{0, \ldots, 0}_{n})$ и $(\tilde{1}^n) = (\underbrace{1, \ldots, 1}_{n})$, где $n \in \mathbb{N}$.

Теорема 1. *Любую булеву функцию $f(\tilde{x}^n)$, $n \geqslant 2$, не принадлежащую множеству $\{0, 1, x_1\& \ldots \&x_n, x_1\& \ldots \&x_n\}$, можно реализовать неизбыточной схемой в базисе Жегалкина B_1, допускающей ЕПТ $\{(\tilde{0}^n), \tilde{\sigma}, (\tilde{1}^n)\}$, где $\tilde{\sigma}$ — произвольный двоичный набор длины n с наименьшим числом единиц, удовлетворяющий условию $f(\tilde{\sigma}) \neq f(\tilde{0}^n)$.*

Теорема 2. *Для любого $n \geqslant 0$ справедливо неравенство $D^{B_1}_{\text{ЕП}}(n) \leqslant 3$.*

Теорема 3. *Для любого $n \geqslant 0$ справедливо неравенство $D^{B_1}_{\text{ЕД}}(n) \leqslant 5$.*

Последний результат улучшает неравенство $D^{B_1}_{\text{ЕД}}(n) \leqslant 22$ из [4].

Список литературы

[1] Редькин Н. П. Надежность и диагностика схем. М. : Издательство Московского университета, 1992. 192 с.

[2] Reddy S. M. Easily testable realizations for logic functions // IEEE Transactions on Computers. 1972. Vol. C-21, no. 11. P. 1183–1188.

[3] Романов Д. С. Метод синтеза легкотестируемых схем, допускающих единичные проверяющие тесты константной длины // Дискретная математика. 2014. Т. 26, вып. 2. С. 100–130.

[4] Романов Д. С., Романова Е. Ю. Метод синтеза неизбыточных схем, допускающих короткие единичные диагностические тесты при константных неисправностях на выходах элементов // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2016. № 2 (38). С. 87–102.

[5] Попков К. А. Нижние оценки длин единичных тестов для схем из функциональных элементов // Дискретная математика. 2017. Т. 29, вып. 2. С. 53–69.

[6] Попков К. А. Короткие единичные тесты для схем при произвольных константных неисправностях на выходах элементов // Дискретная математика. 2018. Т. 30, вып. 3. С. 99–116.

[7] Попков К. А. Метод построения легко диагностируемых схем из функциональных элементов относительно единичных неисправностей // Прикладная дискретная математика. 2019. № 46. С. 38–57.

[8] Бородина Ю. В. О схемах, допускающих единичные тесты длины 1 при константных неисправностях на выходах элементов // Вестник Московского университета. Серия 1. Математика. Механика. 2008. № 5. С. 49–52.

[9] Бородина Ю. В., Бородин П. А. Синтез легкотестируемых схем в базисе Жегалкина при константных неисправностях типа 0 на выходах элементов // Дискретная математика. 2010. Т. 22, вып. 3. С. 127–133.

[10] Попков К. А. О единичных диагностических тестах для схем из функциональных элементов в базисе Жегалкина // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2016. № 3 (39). С. 3–18.

Использование преобразования Фурье для исследования нелинейности векторных функций над конечными полями

Рябов Владимир Геннадьевич

НП «ГСТ»; 4vryabov@gmail.com

Пусть $\mathbf{F}_q$ означает поле из q элементов, где $q = p^m$ и p — простое число, а $\mathbf{F}_q^n$ — n-мерное векторное пространство над полем $\mathbf{F}_q$. Обозначим $P_q^{n,k}$ множество отображений n-мерного пространства $\mathbf{F}_q^n$ в k-мерное пространство $\mathbf{F}_q^k$. В дальнейшем такие отображения будем называть векторными функциями. Всякая векторная функция $F \in P_q^{n,k}$ однозначно определяется упорядоченным набором своих k координатных функций. В свою очередь, каждая координатная функция может быть задана многочленом над полем $\mathbf{F}_q$. Для векторной функции F алгебраическая степень нелинейности $degF$ определяется как максимальная из степеней многочленов ее координатных функций. При выполнении условия $deg\ F \leqslant 1$ отображение F является аффинным. Обозначим через $A_q^{n,k}$ подмножества аффинных отображений из множества $P_q^{n,k}$.

Всякой векторной функции $F \in P_q^{n,k}$ можно поставить в соответствие вектор пространства $\mathbf{F}_{q^k}^{q^n}$ и определить нелинейность отображения F по формуле

$$N_F = \min_{A \in A_q^{n,k}} \rho(F, A), \tag{1}$$

где $\rho(F, A)$ — расстояние Хемминга в пространстве $\mathbf{F}_{q^k}^{q^n}$. Изучение поведения величины N_F представляет интерес для различных областей кибернетики.

При исследовании нелинейности (1) оказывается удобным использовать следующий набор параметров векторной функции. Обозначим $\mathbf{M}_q^{k,n}$ множество матриц с k строками и n столбцами над полем $\mathbf{F}_q$. Всякое аффинное отображение $A \in A_q^{n,k}$ может быть представлено в виде $\alpha_0 \oplus \mathbf{A}\boldsymbol{x}$, где $\alpha_0 = (a_{1,0}, \ldots, a_{k,0})^T \in \mathbf{F}_q^k$, $\mathbf{A} \in \mathbf{M}_q^{k,n}$, а $\boldsymbol{x} = (x_1, \ldots, x_n)^T \in \mathbf{F}_q^n$. Поставим ему

в соответствие матрицу $[\alpha_0, \mathbf{A}] \in \mathbf{M}_q^{k,n+1}$, полученную добавлением α_0 левым столбцом к матрице $\mathbf{A}$. Для векторной функции $F \in P_q^{n,k}$ и аффинного отображения $A \in A_q^{n,k}$ с матрицей $[\alpha_0, \mathbf{A}]$ зададим параметр вида

$$\mathfrak{d}_F^{[\alpha_0,\mathbf{A}]} = (q^k - 1)q^{-k} - \rho(F, A)q^{-n}. \tag{2}$$

Нелинейность связана с набором $q^{k(n+1)}$ параметров вида (2) соотношением

$$N_F = (q^k - 1)q^{n-k} - q^n \max_{[\alpha_0,\mathbf{A}]\in\mathbf{M}_q^{k,n+1}} \mathfrak{d}_F^{[\alpha_0,\mathbf{A}]}. \tag{3}$$

Введем теперь характеры некоторых абелевых групп. В отличие от распространенной практики определения аддитивных характеров поля через функцию следа, представим поле $\mathbf{F}_q$ как векторное пространство $\mathbf{F}_p^m$ над простым полем $\mathbf{F}_p$ и зададим отображение $\mathbf{F}_q$ в $\mathbf{F}_p$ через скалярное произведение $\langle a, x\rangle_p$ в пространстве $\mathbf{F}_p^m$, где x выступает в роли переменной. Определим характер аддитивной группы поля $\mathbf{F}_q$ по формуле $\chi_a(x) = e^{\frac{2\pi i}{p}\langle a,x\rangle_p}$. Пусть $\phi_\alpha(\boldsymbol{x}) = \prod_{t=1}^n \chi_{a_t}(x_t) = e^{\frac{2\pi i}{p}\Sigma_{t=1}^n\langle a_t,x_t\rangle_p}$, где $\alpha = (a_1, \ldots, a_n)$ и $\boldsymbol{x} = (x_1, \ldots, x_n)$ являются векторами пространства $\mathbf{F}_q^n$, а знак Σ означает сумму в поле $\mathbf{F}_p$. Из теории представлений следует, что множество $\{\phi_\alpha(\boldsymbol{x}) \mid \alpha \in \mathbf{F}_q^n\}$ является группой характеров аддитивной группы пространства $\mathbf{F}_q^n$ и образует ортонормированный базис унитарного пространства всех комплекснозначных отображений пространства $\mathbf{F}_q^n$.

Для отображения $F \in P_q^{n,k}$ с набором координат $(f_1, \ldots, f_k)$ и ненулевого вектора $\beta = (b_1, \ldots, b_k) \in \mathbf{F}_q^k$ зададим комплекснозначную функцию $\varphi_\beta(F) : \mathbf{F}_q^n \to \mathbf{C}$, положив $\varphi_\beta(F) = \prod_{s=1}^k \chi_{b_s}(f_s) = e^{\frac{2\pi i}{p}\Sigma_{s=1}^k\langle b_s,f_s\rangle_p}$, называя ее по аналогии с работой [1] характером F. Коэффициенты в разложении характера $\varphi_\beta(F)$ в ряд Фурье по базису $\{\phi_\alpha(\boldsymbol{x}) \mid \alpha \in \mathbf{F}_q^n\}$ имеют вид

$$c_F^\beta(\alpha) = \frac{1}{q^n}\sum_{\boldsymbol{x}\in\mathbf{F}_q^n} \varphi_\beta(F)\overline{\phi_\alpha(\boldsymbol{x})} = \frac{1}{q^n}\sum_{\boldsymbol{x}\in\mathbf{F}_q^n} e^{\frac{2\pi i}{p}\left(\Sigma_{s=1}^k\langle b_s,f_s\rangle_p \ominus \Sigma_{t=1}^n\langle a_t,x_t\rangle_p\right)}. \tag{4}$$

Для коэффициентов Фурье любого из характеров справедливы равенство Парсевала $\sum_{\alpha\in\mathbf{F}_q^n} |c_F^\beta(\alpha)|^2 = 1$ и неравенства $q^{-\frac{n}{2}} \leqslant \max_{\alpha\in\mathbf{F}_q^n} |c_F^\beta(\alpha)| \leqslant 1$. Отображения из множества $P_q^{n,k}$, у которых для любого из характеров все коэффициенты Фурье по модулю равны $q^{-\frac{n}{2}}$, носят название векторных бент-функций. Данное определение обобщает определение бент-функций над конечными полями из работы [2] на случай векторных функций.

Специфика определения характеров и вида коэффициентов Фурье позволила автору в статье [3] получить выражение параметров (2) векторной функции через коэффициенты Фурье ее характеров (4). Для векторной функции $F \in P_q^{n,k}$ и любой матрицы $[\alpha_0, \mathbf{A}] \in \mathbf{M}_q^{k,n+1}$, где вектор α_0 имеет вид

$(a_{1,0},\dots,a_{k,0})^T$, выполняется равенство

$$\mathfrak{d}_F^{[\alpha_0,\mathbf{A}]}=\frac{1}{q^k}\sum_{\beta\in\mathbf{F}_q^k\setminus\{\boldsymbol{0}\}}c_F^{\beta}(\beta\mathbf{A})\;e^{-\frac{2\pi i}{p}\Sigma_{s=1}^{k}\langle b_s,a_{s,0}\rangle_p}. \tag{5}$$

Там же были получены выражения коэффициентов Фурье через аналогичные коэффициенты составляющих векторной функции при декомпозициях вида

$$F(\boldsymbol{x})=G(\boldsymbol{x}')\oplus H(\boldsymbol{x}''), \tag{6}$$

где $F\in P_q^{n,k}$, $G\in P_q^{r,k}$, $H\in P_q^{n-r,k}$, $\boldsymbol{x}=[\boldsymbol{x}',\boldsymbol{x}'']\in\mathbf{F}_q^n$, $\boldsymbol{x}'=(x_1,\dots,x_r)\in\mathbf{F}_q^r$, $\boldsymbol{x}''=(x_{r+1},\dots,x_n)\in\mathbf{F}_q^{n-r}$, а также

$$F(\boldsymbol{x})=\big(G(\boldsymbol{x}),H(\boldsymbol{x})\big), \tag{7}$$

где $F\in P_q^{n,k}$, $G\in P_q^{n,v}$, $H\in P_q^{n,k-v}$ и для наборов координатных функций выполняются равенства $(g_1,\dots g_v)=(f_1,\dots,f_v)$ и $(h_1,\dots h_{k-v})=(f_{v+1},\dots,f_k)$. Полученные результаты с учетом соотношения (3) позволяют выразить нелинейность векторной функции при этих видах декомпозиции.

Утверждение 1. *Для $F\in P_q^{n,k}$ вида (6) выполняется равенство*

$$N_F=q^n-q^{n-k}\Big[1+\max_{[\alpha_0,\boldsymbol{A}]\in\boldsymbol{M}_q^{k,n+1}}\sum_{\beta\in\mathbf{F}_q^k\setminus\{\boldsymbol{0}\}}c_G^{\beta}(\beta\boldsymbol{A}')c_H^{\beta}(\beta\boldsymbol{A}'')\;e^{-\frac{2\pi i}{p}\Sigma_{s=1}^{k}\langle b_s,a_{s,0}\rangle_p}\Big],$$

где $\boldsymbol{A}=[\boldsymbol{A}',\boldsymbol{A}'']\in\boldsymbol{M}_q^{k,n}$, $\boldsymbol{A}'\in\boldsymbol{M}_q^{k,r}$, $\boldsymbol{A}''\in\boldsymbol{M}_q^{k,n-r}$, а для $F\in P_q^{n,k}$ вида (7) справедливо равенство

$$N_F=q^n-q^{n-k}\Big[1+\max_{[\alpha_0,\boldsymbol{A}]\in\boldsymbol{M}_q^{k,n+1}}\sum_{\beta\in\mathbf{F}_q^k\setminus\{\boldsymbol{0}\}}\Big(\sum_{\gamma\in\mathbf{F}_q^n}c_G^{\beta'}(\beta\boldsymbol{A}\ominus\gamma)c_H^{\beta''}(\gamma)\Big)\;e^{-\frac{2\pi i}{p}\Sigma_{s=1}^{k}\langle b_s,a_{s,0}\rangle_p}\Big],$$

где $\beta=[\beta',\beta'']\in\mathbf{F}_q^k$, $\beta'=(b_1,\dots,b_v)\in\mathbf{F}_q^v$ и $\beta''=(b_{v+1},\dots,b_k)\in\mathbf{F}_q^{k-v}$.

В статье [3] была также представлена нижняя граница нелинейности вида

$$N_F\geqslant q^n-q^{n-k}\big(1+(q^k-1)\max_{\beta\in\mathbf{F}_q^k\setminus\{\boldsymbol{0}\},\alpha\in\mathbf{F}_q^n}|c_F^{\beta}(\alpha)|\big), \tag{8}$$

которая для векторных бент-функций принимает максимальное значение[†], равное $q^n-q^{n-k}-g^{n/2}+q^{n/2-k}$. Используя (8), получим следующее.

Утверждение 2. *Для $F\in P_q^{n,k}$ вида (6) выполняется неравенство*

$$N_F\geqslant q^n-q^{n-k}-\frac{q^k}{q^k-1}\big(q^r-q^{r-k}-N_G\big)\big(q^{n-r}-q^{n-r-k}-N_H\big).$$

Замечание. *Для $F\in P_q^{n,k}$ вида (7) имеем $N_F\geqslant\max\{N_G,N_H\}$.*

[†]Приведенное здесь значение нижней границы нелинейности для векторных бент-функций вытекает также из результатов работы [4], полученных другим способом.

Выражение (8) позволяет также, используя известное значение максимального модуля коэффициентов Фурье характеров отображения, находить классы векторных функций с нелинейностью не менее заданной.

Список литературы

[1] Солодовников В. И. Бент-функции из конечной абелевой группы в конечную абелеву группу // Дискретная математика. 2002. Т. 14, вып. 1. С. 99–113.

[2] Амбросимов А. С. Свойства бент-функций q-значной логики над конечными полями // Дискретная математика. 1994. Т. 6, вып. 3. С. 50–60.

[3] Рябов В. Г. Нелинейность векторных функций над конечными полями // Дискретная математика. 2024. Т. 36, вып. 2. С. 50–70.

[4] Carlet C., Ding Cunsheng, Yuan Jin. Linear codes from perfect nonlinear mappings and their secret sharing schemes // IEEE Transactions on Information Theory. 2005. Vol. 51, no. 6. P. 2089–2102.

Операция GF(2)-shuffle над формальными языками

Сажнева Елизавета Александровна
Московский государственный университет имени М. В. Ломоносова; sazhneva.eliza@yandex.ru

В работе Бакиновой и др. [1] были определены $GF(2)$-операции над формальными языками. Эти операции являются вариантами классической конкатенации и звёздочки Клини. Дизъюнкция в определении этих операций заменяется на исключающее ИЛИ. Заменив в определении конкатенации дизъюнкцию на исключающее ИЛИ, получим новую операцию, называемую $GF(2)$-*конкатенацией*:

$$K \odot L = \{\, w \mid \#\text{ разбиений } w = uv, \text{ где } u \in K \text{ и } v \in L, \text{ нечетно} \,\}.$$

Формальные языки образуют кольцо, где $GF(2)$-конкатенация рассматривается как операция умножения, а симметрическая разность — как операция сложения [1]. Более того, каждый язык L, содержащий пустую строку, имеет *обратный* язык относительно операции $GF(2)$-конкатенация: язык L^{-1}, удовлетворяющий равенствам $L \odot L^{-1} = L^{-1} \odot L = \{\varepsilon\}$.

В статье [1] доказана замкнутость семейства регулярных языков относительно операций $GF(2)$-конкатенации и взятия $GF(2)$-обратного языка.

Кроме классических операций над формальными языками, также исследовалось немало других операций, например, циклический сдвиг [2], shuffle [3]. Алгебраические свойства операции shuffle и замкнутость различных классов языков относительно этой операции — тема ряда недавних теоретических исследований, краткий обзор которых можно найти в статье Пина [4].

Цель этой работы — определить $GF(2)$-вариант операции shuffle, исследовать алгебраические свойства новой операции и показать замкнутость регулярных языков относительно этой операции.

Операция $GF(2)$-shuffle и её основные свойства

Чтобы определить операцию $GF(2)$-shuffle, необходимо в определении операции shuffle учитывать только разбиения на непустые строки. Для этого перепишем определение операции shuffle следующим образом.

Определение 1. *Для любых языков K и L их shuffle, $K \sqcup L$, — это множество всех строк w, представимых в виде $w = u_1u_2 \dots u_k$, где $k \geqslant 1$, $u_1 \in \Sigma^*$, $u_2, \dots, u_k \in \Sigma^+$, $u_1u_3u_5 \dots \in K$ и $u_2u_4u_6 \dots \in L$.*

Заменив в определении 1 дизъюнкцию на исключающее ИЛИ, получим новую операцию $GF(2)$-shuffle.

Определение 2. *Для любых языков K и L их $GF(2)$-shuffle, обозначаемый $K \circledcirc L$, — это множество всех строк w, имеющих нечетное число представлений в виде $w = u_1u_2 \dots u_k$, где $k \geqslant 1$, $u_1 \in \Sigma^*$, $u_2, \dots, u_k \in \Sigma^+$, $u_1u_3u_5 \dots \in K$ и $u_2u_4u_6 \dots \in L$.*

$GF(2)$-shuffle состоит из всех строк w, имеющих *нечетное количество разбиений*, следовательно, $K \circledcirc L \subseteq K \sqcup L$.

Пример 1. $\{ab\} \sqcup \{bc\} = \{abbc, abcb, babc, bacb, bcab\}$. Строка $abbc$ не принадлежит $GF(2)$-shuffle, так как имеет два представления: $abbc = ab \cdot bc = a \cdot b \cdot b \cdot c$, которые исключают друг друга. Поэтому $\{ab\} \circledcirc \{bc\} = \{abcb, babc, bacb, bcab\}$.

Следующий пример показывает, что некоторые языки имеют обратный язык относительно операции $GF(2)$-*shuffle*.

Пример 2. $\{\varepsilon, ab\} \circledcirc \{\varepsilon, ab\} = \{\varepsilon\}$.

Оказывается, что каждый язык, содержащий пустую строку, имеет обратный язык относительно операции $GF(2)$-shuffle, и обратный язык равен себе.

Теорема 1. *Для любого языка $L \subseteq \Sigma^*$, такого что $\varepsilon \in L$, верно, что $L \circledcirc L = \{\varepsilon\}$. Для любых двух различных языков $K, L \in \Sigma^*$ верно, что $K \circledcirc L \neq \{\varepsilon\}$.*

Элементарные алгебраические свойства операции $GF(2)$-shuffle можно вывести из определения 2 и теоремы 1. Эти свойства кратко описаны ниже и могут быть установлены прямой проверкой.

Утверждение 1. *Для любого алфавита* Σ *множество всех языков над этим алфавитом, содержащих пустую строку, образует абелеву группу с бинарной операцией* $GF(2)$*-shuffle.*

Утверждение 2. *Для каждого алфавита* Σ *множество всех языков* 2^{Σ^*} *над алфавитом* Σ *образует* коммутативное кольцо *с операцией симметрической разности как суммой и с операцией* $GF(2)$*-shuffle как произведением.*

$GF(2)$-shuffle над регулярными языками

Важный вопрос для введенной операции на регулярных языках — сохранение класса регулярных языков и сложность описания этой операций, то есть насколько большой автомат необходим для представления операции на конечных автоматах заданного размера.

Детерминированный конечный автомат (ДКА) определяется как пятерка $\mathcal{A} = (\Sigma, Q, q_0, \delta, F)$, где: Σ — входной алфавит; Q — конечное непустое множество состояний; $q_0 \in Q$ — начальное состояние; $\delta : Q \times \Sigma \to Q$ — функция перехода; $F \subseteq Q$ — множество принимающих состояний. Вычисление $\mathcal{A}$ на строке $w = a_1 \dots a_n$, где $a_1, \dots, a_n \in \Sigma$, является однозначно определенной последовательностью состояний $r_0, \dots, r_n \in Q$, таких что $r_0 = q_0$ и $r_i = \delta(r_{i-1}, a_i)$ для $i \in \{1, \dots, n\}$. Если $r_n \in F$, говорят, что ДКА принимает строку w. Язык, распознаваемый ДКА, обозначаемый $L(\mathcal{A})$, представляет собой набор всех строк, которые принимает автомат $\mathcal{A}$. В следующей теореме показано, что $GF(2)$-shuffle сохраняет класс регулярных языков, и утверждается, что автомат, распознающий $GF(2)$-shuffle двух регулярных языков, может быть эффективно построен.

Теорема 2. *Для любых двух ДКА* $\mathcal{A} = (\Sigma, P, p_0, \eta, E)$ *и* $\mathcal{B} = (\Sigma, Q, q_0, \delta, F)$*, таких что* $|P| = m$ *и* $|Q| = n$*,* $GF(2)$*-shuffle* $L(\mathcal{A})$ *и* $L(\mathcal{B})$ *распознаётся ДКА* $\mathcal{C}$ *со множеством состояний* $2^{P \times Q}$*, где:*

- *начальное состояние равно* (p_0, q_0)*;*
- *функция переходов* $\pi : (2^{P \times Q}) \times \Sigma \to 2^{P \times Q}$ *определяется для каждого состояния* $S = \{ (p, q) \mid p \in P, q \in Q \}$ *и символа* $a \in \Sigma$ *следующим образом:*

  $$\pi(S, a) = S' = \{ (p', q') \mid \text{число состояний } (p, q) \in S, \text{ где } p \in P, q \in Q, \text{ таких что } p' = \eta(p, a) \text{ и } q' = q \text{ XOR } q' = \delta(q, a) \text{ и } p' = p, \text{ нечетно} \};$$

- *множество принимающих состояний* — $F' = \left\{ S \;\middle|\; |S \cap (E \times F)| \text{ нечетно} \right\}$.

Список литературы

[1] Formal languages over $GF(2)$ / E. Bakinova, A. Basharin, I. Batmanov, K. Lyubort, A. Okhotin, E. Sazhneva // Information and Computation. 2022. Vol. 283, P. 104672.

[2] Jirásková G., Okhotin A. State complexity of cyclic shift // RAIRO-Theoretical Informatics and Applications. 2008. Vol. 42, no. 2. P. 335–360.

[3] Câmpeanu C., Salomaa K., Yu Sheng. Tight lower bound for the state complexity of shuffle of regular languages // Journal of Automata, Languages and Combinatorics. 2002. Vol. 7, no. 3. P. 303–310.

[4] Pin J.-É. Shuffle product of regular languages: results and open problems // Algebraic Informatics. CAI 2022. Lecture Notes in Computer Science. 2022. Vol. 13706. P. 26–39.

Максимальные наборы, k-свободные от сумм, в абелевой группе

Саргсян Ваге Гнелович

Институт проблем информатики и автоматизации Национальной академии наук Республики Армения; vahe_sargsyan@ymail.com, vahesargsyan83@gmail.com

Введение

Пусть G — абелева группа порядка n, а $k \geqslant 2$ — целое число, и $A_1, \ldots, A_k$ — непустые подмножества G. Набор $(A_1, \ldots, A_k)$ называется *k-свободным от сумм* (сокращенно k-НСС), если не существует набора

$$(a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k,$$

являющегося решением уравнения

$$x_1 + \cdots + x_k = 0. \tag{1}$$

Семейство k-НСС в G обозначим через $S_k(G)$. Положим

$$\varrho_k(G) = \max_{(A_1,\ldots,A_k)\in S_k(G)} (|A_1| + \cdots + |A_k|).$$

В работе [1] была доказана следующая теорема

Теорема. *Пусть G — абелева группа порядка n, а k — натуральное число, $k \geqslant 3$. Тогда справедливо равенство*

$$\log |S_k(G)| = \varrho_k(G) + \bar{o}(n)$$

при $n \to \infty$.

Пусть $(A_1, \ldots, A_k)$ — набор, k-свободный от сумм, в группе G. Набор $(A_1, \ldots, A_k)$ назовем *максимальным по мощности,* если он максимальный по $\varrho_k(G)$, и *максимальным по включению,* если для любых $i \in \{1, \ldots, k\}$ и $x \in G \setminus A_i$, набор

$$(A_1, \ldots, A_{i-1}, A_i \cup \{x\}, A_{i+1}, \ldots, A_k)$$

не является k-свободным от сумм в группе G.

В этой работе рассматриваются следующие задачи.
Задача 1. Нахождение $\varrho_k(G)$.
Задача 2. Определение структуры максимального по мощности (по включению) k-НСС.

Нахождение $\varrho_k(G)$

Теорема 1. *Для любого простого числа p справедливо равенство*

$$\varrho_k(Z_p) = p + k - 2.$$

Теорема 2. *Пусть G — абелева группа порядка n и экспоненты ν. Тогда*

$$n + \frac{n}{p_1}(k-2) = \max_{d|\nu}\Big(\frac{n}{d}(d+k-2)\Big) \leqslant \varrho_k(G) \leqslant \max_{d|n}\Big(\frac{n}{d}(d+k-2)\Big) = n + \frac{n}{p_2}(k-2),$$

где p_1 — наименьший простой делитель ν, а p_2 — наименьший простой делитель n.

Есть предпосылки предположить, что верна следующая оценка.

Утверждение. *Пусть G — абелева группа порядка n и экспоненты ν. Тогда*

$$\varrho_k(G) = n + \frac{n}{p}(k-2),$$

где p — наименьший простой делитель ν.

Теорема 3. *Для любого n справедливо равенство*

$$\varrho_k(Z_n) = n + \frac{n}{p}\big(k-2\big),$$

где p — наименьший делитель n.

Теорема 4. *Пусть G — абелева группа порядка n и экспоненты ν. Тогда*

$$\varrho_k(G) \geqslant \max_{d|\nu}\Big(\frac{n}{d}\varrho_k(Z_d)\Big).$$

О структуре максимального по мощности набора, k-свободного от сумм, в циклической группе

Пусть A — подмножество абелевой группы G. Через $\overline{A}$ обозначаем дополнение подмножества A в абелевой группе G, то есть $\overline{A} = G \setminus A$.

Теорема 5. *Пусть $k \geqslant 2$, Z_p — циклическая группа простого порядка и $(A_1, \ldots, A_k)$ — максимальный по мощности набор, k-свободный от сумм, в Z_p. Тогда каждое множество набора с точностью до изоморфизма есть одно из следующих:*

i) $|A_i| = 1$;

ii) $A_i = \overline{-(A_1 + \cdots + A_{i-1} + A_{i+1} + \cdots + A_k)}$;

iii) A_i — *арифметическая прогрессия с разностью* 1;

где $i = 1, \ldots, k$.

Теорема 6. *Пусть* $k \geqslant 2$, p — *наименьший простой делитель натурального числа* n, H — *подгруппа группы* Z_n *порядка* n/p *и* $(A_1, \ldots, A_k)$ — *максимальный по мощности набор,* k*-свободный от сумм, в* Z_n. *Тогда каждое множество этого набора с точностью до изоморфизма есть одно из следующих:*

i) $|A_i| = n/p$, *то есть* A_i — *смежный класс* Z_n *по подгруппе* H;

ii) A_i — *объединение смежных классов* Z_n *по подгруппе* H *такое, что для множества представителей смежных классов как подмножества циклической группы* Z_p *справедливо соотношение*

$$A_i/H = \overline{-(A_1/H + \cdots + A_{i-1}/H + A_{i+1}/H + \cdots + A_k/H)};$$

iii) A_i — *объединение смежных классов* Z_n *по подгруппе* H *такое, что множество представителей смежных классов как подмножество циклической группы* Z_p *является арифметической прогрессией с разностью* 1;

где $i = 1, \ldots, k$.

Теорема 7. *Пусть* G — *абелева группа порядка* n *и* $(A_1, \ldots, A_k)$ — *максимальный по включению набор,* k*-свободный от сумм, в группе* G, *удовлетворяющий данному условию:*

$$|A_1| + \cdots + |A_k| \geqslant n + k - 2.$$

Тогда существует подгруппа H *группы* G *такая, что:*

i) $A_i + H = A_i$, *где* $i = 1, \ldots, k$;

ii) G/H — *циклическая группа;*

iii) $(A_1/H, \ldots, A_k/H)$ — *максимальный по включению набор,* k*-свободный от сумм, в фактор-группе* G/H.

Теорема 8. *Если в абелевой группе* G *существует максимальный по включению набор,* k*-свободный от сумм, с мощностью* k, *то группа* G *циклическая.*

Автор выражает благодарность профессору Сапоженко А. А. за постановку задачи.

Список литературы

[1] Сапоженко А. А., Саргсян В. Г. Асимптотика логарифма числа наборов, k-свободных от решений, в абелевых группах // Дискретная математика. 2018. Т. 30, вып. 3. С. 117–126.

Универсальные функции для пар линейных

Седова Анна Сергеевна
Московский государственный университет имени М. В. Ломоносова; okuneva-anna@mail.ru

Введение

Понятие универсальной функции было введено в работе [1]. Далее были исследованы задачи о существовании, мощности области определения и представления в простом виде универсальных функций для различных классов. В настоящей работе вводится понятие универсальной функции для пары линейных функций и рассматривается задача о ее существовании.

Постановка

Далее везде будем рассматривать булевы функции размерности n. Для функций $(g_0, \overline{g_0})$ не существует точек, на которых они совпадают. Пару функций $(g_0, \overline{g_0})$, будем называть недопустимой.

Определение. *Функция $f(x_1, \ldots, x_n)$ порождает допустимую пару линейных функций $g_0(x_1, \ldots, x_n)$ и $g_1(x_1, \ldots, x_n)$, если $g_0 \in L \cap T_0$, $g_1 \in L \cap \overline{T_0}$ и можно предъявить множество точек $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, такое, что g_0 и g_1 являются единственными функциями из соответствующих классов, что для любого $\mathbf{x} \in X$ выполняются соотношения $g_0(\mathbf{x}) = f(\mathbf{x}), g_1(\mathbf{x}) = f(\mathbf{x})$.*

Заметим, что если f не порождает пару (g_0, g_1), то существует пара допустимых функций (g_0', g_1'), такая, что не существует точки $\mathbf{x}$: $f(\mathbf{x}) = g_0(\mathbf{x}) = g_1(\mathbf{x})$ и при этом $f(\mathbf{x}) \neq g_0'(\mathbf{x})$ или $f(\mathbf{x}) \neq g_1'(\mathbf{x})$.

Определение. *Функция $f(x_1, \ldots, x_n)$ называется универсальной функцией для пар линейных функций, если она порождает любую пару допустимых линейных функций.*

Основная часть

Теорема 1. *Универсальная функция для пар функций существует при $n \geqslant 7$.*

Доказательство. Пусть $f(x_1, \ldots, x_n)$ порождает пару функций $g_0 \in L \cap T_0$ и $g_1 \in L \cap \overline{T_0}$. При этом g_0 и g_1 совпадают на половине куба.

Рассмотрим две пары допустимых функций: (g_0, g_1) и (g_0', g_1'). Обозначим через T часть пространства, где совпадают функции g_0 и g_1, а через T' — часть пространства, где совпадают функции g_0' и g_1'. T и T' имеют размер полпространства, то есть 2^{n-1}. При этом:

1. $T \neq T'$, так как только g_0 и g_1 совпадают на T.
2. $T \neq \overline{T'}$ ввиду значений функций в точке $(0, \ldots, 0)$.
3. $T \cap T'$ и T/T' имеют размерность четверти пространства, то есть 2^{n-2}.

Будем использовать вероятностный метод в комбинаторике [2]. Рассмотрим равномерное распределение булевых функций. Пусть $P(A)$ — вероятность того, что универсальная функция для пар функций существует. Оценим сверху вероятность дополнительного события $\overline{A}$ — любая $f(x_1, \ldots, x_n)$ не является универсальной функцией.

Для каждой точки из множества T/T' вероятность для случайной функции f не отличить пару (g_0, g_1) от пары (g_0', g_1') равна $\frac{1}{2}$. Число всевозможных допустимых пар функций (g_0, g_1) — $(2^n(2^n-1))$, а пар (g_0', g_1') — $(2^n(2^n-1)-1)$. Учитывая, что вероятность объединения не превосходит суммы вероятностей, получим:

$$P(\overline{A}) \leqslant \frac{1}{2^{2^{n-2}}} \times (2^n(2^n-1))(2^n(2^n-1)-1) \leqslant 2^{4n-2^{n-2}}.$$

Функция $4n - 2^{n-2}$ убывает при $n \geqslant 7$. При $n = 7$ получим: $P(\overline{A}) \leqslant 2^{-4} < 1$. □

Список литературы

[1] Вороненко А. А. Об универсальных частичных функциях для класса линейных функций // Дискретная математика. 2012. Т. 24, №. 3. С. 62–65.

[2] Райгородский А. М. Вероятность и алгебра в комбинаторике. М. : МЦНМО, 2008.

О замкнутом классе полиномиальных функций в k-значной логике

Селезнева Светлана Николаевна
Московский государственный университет имени М. В. Ломоносова,
факультет вычислительной математики и кибернетики; selezn@cs.msu.ru

Пусть $k \geqslant 2$ — целое число, $E_k = \{0, 1, \ldots, k-1\}$. Функцией k-значной логики f местности n называем отображение $f : E_k^n \to E_k$, $n \geqslant 1$. Множество всех функций k-значной логики обозначаем P_k. Рассматриваем представление функций k-значной логики полиномами по модулю k. Функцию из P_k называем полиномиальной, если ее можно представить каким-то полиномом над кольцом Z_k вычетов по модулю k. Множество всех полиномиальных функций k-значной логики обозначаем Pol_k. Известно, что $Pol_k = P_k$ тогда и только тогда, когда k — простое число [1, 2]. В случае, когда k — составное число, появляется вопрос о критериях полиномиальности функций из P_k. К настоящему времени разными авторами найден ряд таких критериев, основанных

на различных подходах. В частности, в [3] получены критерии полиномиальности, некоторые из них затем уточнены в [4, 5].

Множество P_k рассматриваем как функциональную систему с операциями суперпозиции [1]. Если $A \subseteq P_k$, то множество A называется замкнутым классом, если оно замкнуто относительно суперпозиции. Известно, что при $k \geqslant 2$ множество Pol_k является замкнутым классом. Отношением (предикатом) местности r на множестве E_k называем подмножество множества E_k^r, $r \geqslant 1$. Множество всех отношений на E_k обозначаем R_k. Пусть $f \in P_k$ и $\rho \in R_k$. Говорят, что функция f сохраняет отношение ρ, если для любых столбцов $\gamma^1, \ldots, \gamma^n \in \rho$ верно $f(\gamma^1, \ldots, \gamma^n) \in \rho$, где $f(\gamma^1, \ldots, \gamma^n) = \delta \in E_k^r$ и $\delta_i = f(\gamma_i^1, \ldots, \gamma_i^n)$ для всех $i = 1, \ldots, r$. Множество всех функций из P_k, сохраняющих отношение ρ, обозначаем $A_k(\rho)$. Известно, что для любого $\rho \in R_k$ множество $A_k(\rho)$ является замкнутым классом (см., например, [2]).

В [4, 5] автором настоящей работы начато исследование описания замкнутого класса Pol_k посредством отношений и получены эти описания для ряда значений числа k. Приведем здесь эти теоремы из [4, 5].

Теорема 1 ([4, 5]). *Пусть p — простое число. Класс Pol_{p^2} является множеством всех функций из P_{p^2}, сохраняющих отношение $\rho_{p,2,2}$, где*

$$\rho_{p,2,2} = \left\{ \begin{pmatrix} a \\ a+bp \\ a+cp \\ a+(b+c)p \end{pmatrix} \mid a, b, c \in E_{p^2} \right\}.$$

Теорема 2 ([5]). *Пусть p — простое число, $p \neq 2$. Класс Pol_{p^2} является множеством всех функций из P_{p^2}, сохраняющих отношение $\rho_{p,2,1}$, где*

$$\rho_{p,2,1} = \left\{ \begin{pmatrix} a \\ a+bp \\ a+2bp \end{pmatrix} \mid a, b \in E_{p^2} \right\}.$$

Теорема 3 ([4]). *Пусть p — простое число, $1 \leqslant m \leqslant p$. Класс Pol_{p^m} является множеством всех функций из P_{p^m}, сохраняющих отношение $\rho_{p,m,m}$, где*

$$\rho_{p,m,m} = \left\{ \gamma^b \in E_{p^m}^{2^m} \mid b \in E_{p^m}^{2^m} \right\},$$

$\gamma_r^b = \sum\limits_{j \in E_2^m, j \leqslant r} b_j \cdot p^{|j|}$ *для всех* $r \in E_2^m$.

Пример 1. Пусть $k = p^3$, где p — простое число и $p \neq 2$. Тогда $Pol_{p^3} = A_{p^3}(\rho_{p,3,3})$, где

$$\rho_{p,3,3} = \left\{ \begin{pmatrix} a \\ a + b_1 p \\ a + b_2 p \\ a + b_3 p \\ a + (b_1 + b_2)p + c_3 p^2 \\ a + (b_1 + b_3)p + c_2 p^2 \\ a + (b_2 + b_3)p + c_1 p^2 \\ a + (b_1 + b_2 + b_3)p + (c_1 + c_2 + c_3)p^2 \end{pmatrix} \mid a, b_i, c_i \in E_{p^3} \right\}.$$

В настоящей работе уточнен еще один критерий из [3]. На основе полученного уточнения найдено полное описание замкнутого класса Pol_k посредством отношений. А именно, для каждого простого числа p и каждого числа $m \geqslant 1$ найдены в явном виде отношения, описывающие замкнутый класс Pol_{p^m} (см. теорему 4).

Сначала введем необходимые определения. Пусть p — простое число, $m \geqslant 1$. Пусть N обозначает множество натуральных чисел с нулем. Если $s \in N$, то положим $c_{p,m}(s) = t$, где $t \in N$ — такое наибольшее число из чисел $0, 1, \ldots, m-1, m$, что факториал $s!$ числа s делится нацело на p^t. Если $s = (s_1, \ldots, s_n) \in N^n$, $n \geqslant 1$, то положим $c_{p,m}(s) = \min(m, \sum\limits_{i=1}^{n} c_{p,m}(s_i))$. Число $c_{p,m}(s)$ назовем составной характеристикой числа $s \in N$ или набора $s \in N^n$ по отношению к простому числу p и числу m. Если $s \in N^n$, то $sp = (s_1 p, \ldots, sp)$. Набор $s \in N^n$ назовем граничным (для свойства полиномиальности по модулю p^m), если $c_{p,m}(sp) = m$, но для любого такого набора $t \in N^n$, что $t < s$, верно $c_{p,m}(tp) < m$. Обозначим через $\Gamma_{p,m}(n)$ множество всех граничных наборов длины n. Если $s \in N^n$, то положим $\hat{T}(s) = \{r \in N^n \mid r > s\}$. Далее положим $N_{p,m}(n) = N^n \setminus \left(\bigcup\limits_{s \in \Gamma_{p,m}(n)} \hat{T}(s) \right)$. Множество $N_{p,m}(n)$ назовем множеством всех значащих наборов для свойства полиномиальности по модулю p^m. Пусть $N_{p,m} = N_{p,m}(m)$ и $n_{p,m} = |N_{p,m}|$. Наборы из N^n сравниваем поразрядно. Биномиальный коэффициент из r по j обозначаем C_r^j, $r, j \in N$.

Теорема 4. *Пусть p — простое число, $m \geqslant 1$. Класс Pol_{p^m} является множеством всех функций из P_{p^m}, сохраняющих отношение $\rho_{p,m,m}$, где*

$$\rho_{p,m,m} = \left\{ \gamma^b \in E_{p^m}^{n_{p,m}} \mid b \in E_{p^m}^{n_{p,m}} \right\},$$

для любого $r \in N_{p,m}$ верно $\gamma_r^b = \sum\limits_{j \in N_{p,m}, j \leqslant r} b_j \cdot C_{r_1}^{j_1} \cdot \ldots \cdot C_{r_m}^{j_m} \cdot p^{c_{p,m}(jp)}$.

Пример 2. Пусть $k = 8 = 2^3$. Тогда $Pol_8 = A_8(\rho_{2,3,3})$, где

$$\rho_{2,3,3} = \left\{ \begin{pmatrix} a \\ a+2b_1 \\ a+2b_2 \\ a+2b_3 \\ a+4b_1 \\ a+4b_2 \\ a+4b_3 \\ a+2(b_1+b_2)+4c_3 \\ a+2(b_1+b_3)+4c_2 \\ a+2(b_2+b_3)+4c_1 \\ a+2(b_1+b_2+b_3)+4(c_1+c_2+c_3) \end{pmatrix} \mid a, b_i, c_i \in E_8 \right\}.$$

Список литературы

[1] Яблонский С. В. Функциональные построения в k-значной логике // Труды Математического института имени В. А. Стеклова. 1958. Т. 51. С. 5–142.

[2] Марченков С. С. Функциональные системы с операцией суперпозиции. М. : Физматлит, 2004. 104 с.

[3] Carlitz L. Functions and polynomials (mod p^n) // Acta Arithmetica. 1964. Vol. 9, no. 1. P. 67–78.

[4] Селезнева С. Н. Описание замкнутого класса полиномиальных функций по модулю степени простого числа посредством отношения // Дискретная математика. 2023. Т. 35, вып. 4. С. 115–125.

[5] Селезнева С. Н. Описание замкнутого класса полиномиальных функций по модулю квадрата простого числа посредством отношения // Международная конференция «Математика в созвездии наук». К юбилею ректора МГУ Виктора Антоновича Садовничего : Тезисы докладов. М. : Издательство Московского университета, 2024. С. 344–345.

Нижние оценки сложности линейных операторов над $GF(2)$

Сергеев Игорь Сергеевич

ФГУП «НИИ „Квант“»; isserg@gmail.com

Изучается сложность реализации линейных операторов над полем $GF(2)$ схемами из функциональных элементов сложения (*аддитивными схемами*). Сложность оператора с матрицей A обозначается через $\mathsf{L}(A)$ — эта величина сокращенно называется *сложностью матрицы* A.

Известно, что почти все булевы матрицы размера $n \times n$ имеют сложность асимптотически $n^2/(2\log_2 n)$ (фактически доказано в [1]). Однако трудно указать конкретную матрицу высокой сложности.

Для близких моделей аддитивных схем с операциями целочисленного сложения или дизъюнкции примеры матриц практически экстремальной сложности $n^{2-o(1)}$ построены в [2, 3]. Доказательство нелинейной нижней оценки $\mathsf{L}(A) = \omega(n)$ для явно заданной матрицы размера $n \times n$ является известной открытой проблемой.

По-видимому, самые высокие известные нижние оценки сложности конкретно заданных (последовательностей) матриц размера $n \times n$ имели величину $3n - o(n)$. Обобщая один из таких примеров, построенный Чашкиным в [4], мы укажем матрицу с нижней оценкой сложности $5n - o(n)$.

Далее через $GF(2)^{m\times n}$ обозначается множество булевых матриц размера $m \times n$ (m строк, n столбцов). Через $A_1 \boxplus A_2$ обозначим *прямую сумму* матриц A_1 и A_2, а именно $A_1 \boxplus A_2 = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$.

Сначала расширим модель вычислений. Рассмотрим вычисление оператора AX (где X — вектор переменных) аддитивными схемами, которые имеют произвольный набор Y дополнительных переменных в качестве входов. Если элемент такой расширенной схемы вычисляет сумму[‡] $\langle a, X\rangle + \langle b, Y\rangle$, то вектор b назовем *типом* элемента. *Приведенной сложностью* схемы назовем разность между общим числом элементов и числом типов элементов в схеме с весом не менее 2. Через $\mathsf{L}^*(A)$ обозначим минимум приведенной сложности расширенных схем, вычисляющих матрицу A. Очевидно, всегда выполнено $\mathsf{L}^*(A) \leqslant \mathsf{L}(A)$. Несложно доказывается

Лемма 1. *Для любой пары булевых матриц A_1, A_2 справедливо*

$$\mathsf{L}^*(A_1 \boxplus A_2) = \mathsf{L}^*(A_1) + \mathsf{L}^*(A_2), \qquad \mathsf{L}(A_1 \boxplus A_2) \geqslant \mathsf{L}(A_1) + \mathsf{L}^*(A_2).$$

К сожалению, высокие нижние оценки при помощи меры L^* получить нельзя. Приведенная сложность всегда не более чем линейна.

Теорема 1. *Для любой матрицы $A \in GF(2)^{m\times n}$ выполнено $\mathsf{L}^*(A) \leqslant 2m + n$.*

В отличие от меры L^*, исходная мера сложности L, вообще говоря, не обладает свойством аддитивности относительно операции прямой суммы. Это доказывает пример из работы [5]. Он опирается на простое наблюдение. Если $B \in GF(2)^{n\times n}$, то вычисление матрицы $A = B \boxplus \cdots \boxplus B$ (всего n слагаемых) соответствует умножению $B \cdot X$ матрицы B на матрицу переменных X. Вообще, используя известные результаты о быстром умножении матриц, легко

[‡]Через $\langle\cdot,\cdot\rangle$ обозначается скалярное произведение векторов над $GF(2)$.

оценить возможную экономию сложности при вычислении прямых сумм:

$$\rho = \sup_{\mathsf{L}(A)>0} \frac{\mathsf{L}(A)+\mathsf{L}(B)}{\mathsf{L}(A \boxplus B)} = 2.$$

Рассмотренный выше пример позволяет установить связь между мерой сложности L^* и сложностью билинейных алгоритмов умножения матриц. Напомним, что *билинейный алгоритм* — это схема в базисе операций умножения и сложения, причем для каждого элемента умножения одним аргументом является линейная комбинация коэффициентов одной перемножаемой матрицы, а другим — линейная комбинация коэффициентов второй матрицы. Пусть $\mathsf{bil}_+(n)$, $\mathsf{bil}_*(n)$, $\mathsf{bil}(n)$ означают соответственно минимальное число аддитивных операций, мультипликативных операций и общее число операций в билинейном алгоритме умножения матриц из $GF(2)^{n\times n}$. Через $\nu(B)$ обозначается вес булевой матрицы B.

Лемма 2. *Для любой матрицы $B \in GF(2)^{n\times n}$ справедливо*

$$\mathsf{bil}_+(n) \geqslant n\mathsf{L}^*(B) + n^2 - \nu(B) - O(n).$$

Индексом независимости матрицы B назовем максимальное число k, такое что любые k строк из B линейно независимы над $GF(2)$. Обозначим эту величину через $\mathrm{ind}(B)$.

С опорой на комбинаторный результат [6] (оценка границы Мура для нерегулярных графов) доказывается

Теорема 2. *Пусть $m \leqslant n$, матрица $B \in GF(2)^{n\times m}$ не имеет строк веса* 1 *и* $\mathrm{ind}(B) \geqslant 2k+2 \geqslant 6$. *Тогда*

$$\mathsf{L}^*(B) \geqslant n + \frac{2k-2}{2k+1} \cdot n^{\frac{k}{k+1}} - m.$$

Примеры матриц с высоким индексом независимости предоставляет теория линейных кодов. Известно, что если линейный код с проверочной матрицей H имеет расстояние d, то $\mathrm{ind}(H^\top) = d-1$.

Например, из строк вида $(\alpha, \alpha^2, \ldots, \alpha^s)$, где α — различные элементы поля $GF(2^p)$, записанные векторами коэффициентов в некотором базисе над $GF(2)$, можно составить матрицу с индексом независимости не менее s. Полагая $p = \lceil \log_2 n \rceil$, $s \sim \sqrt{n}$ и $m = ps$, составим матрицу U из $n-m$ таких строк. Используя теорему 2, лемму 1 и известное соотношение $\mathsf{L}(B)+m = \mathsf{L}(B^\top)+n$, справедливое для любой матрицы B размера $m\times n$ без нулевых строк и столбцов (принцип транспонирования), получаем

Следствие 1. *Для матрицы $A = U^\top \boxplus U \in GF(2)^{n\times n}$ выполнено* $\mathsf{L}(A) \geqslant 5n - o(n)$.

Замечая, что для единичного вектора $1_{1\times n}$ длины n имеет место $\mathsf{L}^*(1_{1\times n}) = \mathsf{L}(1_{1\times n}) = n-1$, также получаем

Следствие 2. *Для матрицы* $A = 1_{1\times(n-m)} \boxplus U \in GF(2)^{(n-m+1)\times n}$ *справедливо* $\mathsf{L}^*(A) \geqslant 3n - o(n)$.

В свете теоремы 1 матрица A имеет асимптотически максимально возможную для (почти) квадратных матриц приведенную сложность. Используя матрицу из следствия 2, с помощью леммы 2 и соотношения $\mathsf{bil}_*(n) \geqslant (3-o(1))n^2$ [7] извлекаем оценки сложности билинейных алгоритмов умножения матриц над $GF(2)$.

Следствие 3. $\mathsf{bil}_+(n) \geqslant (4-o(1))n^2$, $\mathsf{bil}(n) \geqslant (7-o(1))n^2$.

О существовании других нетривиальных нижних оценок аддитивной сложности умножения матриц при $n \to \infty$ автору неизвестно.

Список литературы

[1] Нечипорук Э. И. О вентильных схемах // Доклады Академии наук СССР. 1963. Т. 148, № 1. С. 50–53.

[2] Андреев А. Е. Об одном семействе булевых матриц // Вестник Московского университета. Серия 1. Математика. Механика. 1986. № 2. С. 97–100.

[3] Kóllar J., Rónyai L., Szabó T. Norm-graphs and bipartite Turán numbers // Combinatorica. 1996. Vol. 16, no. 3. P. 399–406.

[4] Чашкин А. В. О сложности булевых матриц, графов и соответствующих им булевых функций // Дискретная математика. 1994. Т. 6, вып. 2. С. 43–73.

[5] Paul W. J. Realizing Boolean functions on disjoint sets of variables // Theoretical Computer Science. 1976. Vol. 2., no. 3. P. 383–396.

[6] Alon N., Hoory S., Linial N. The Moore bound for irregular graphs // Graphs and Combinatorics. 2002. Vol. 18. P. 53–57.

[7] Shpilka A. Lower bounds for matrix product // SIAM Journal on Computing. 2003. Vol. 32, no. 5. P. 1185–1200.

Анализ работы нейронных сетей при решении задачи регрессии координат (Supervised Coordinate Regression)

Сидорчук Алексей Игоревич
Московский государственный университет имени М. В. Ломоносова, филиал в городе Ташкенте;
alexstelbu@mail.ru

Аннотация

В данной работе рассмотрено поведение нейронных сетей [1] (в частности, свёрточных [2]) при решении задачи регрессии координат [3]. Также в [3] показано, что свёрточные нейронные сети плохо справляются с решением данной задачи. Постановка задачи регрессии координат: дано входное изображение, содержащее один белый пиксель; требуется вывести его координаты. Для этой задачи предполагаем, что набор данных имеет следующую структуру: двумерный массив, заполненный числами a и содержащий ровно одно число b (a не равно b). Эти числа представляют черные и белый пиксели соответственно. Основной результат состоит в том, что любую детерминированную нейронную сеть, которая решает данную задачу с некоторой точностью, можно выразить через однослойную линейную сеть, состоящую из двух нейронов (для вывода координат x, y).

Основные результаты

Утверждение 1. *Для задачи регрессии координат и для однослойной сверточной нейронной сети, которая решает задачу с некоторой точностью, существует однослойная сверточная сеть с тождественными функциями активации, которая повторяет выход изначальной сети.*

Утверждение 2. *Для задачи регрессии координат и для нейронной сети, состоящей из сверточных слоев, которая решает задачу с некоторой точностью, существует однослойная сверточная сеть с тождественными функциями активации, которая повторяет выход изначальной сети.*

Теорема 1. *Для задачи регрессии координат и для любой детерминированной нейронной сети, которая решает задачу с некоторой точностью, существует однослойная нейронная сеть, состоящая из двух нейронов (один выдает координаты x, второй y), которая повторяет выход изначальной сети, и имеет тождественные функции активации.*

Следствие. *Для нейронной сети из теоремы 1 при решении задачи регрессии координат справедлива следующая оценка изменения выходных данных*

при изменении входных:

$$||\Delta output|| \leqslant \sqrt{2(a-b)^2} \cdot ||A||,$$

где $||\cdot||$ *— любая векторная норма и* A *— матрица, размерность которой* $2{\times}n$, $n \in \mathbb{N}$. *Строки данной матрицы заполняются весами нейронов, а именно, первая и вторая строка заполняются весами первого и второго нейронов соответственно (нейронов сети из теоремы 1). Если вектор весов одного нейрона меньше вектора весов другого, дополним первый нулями, чтобы их размерность совпадала.*

Автор выражает благодарность канд. физ.-мат. наук Иванову И. Е. за постановку задачи.

Список литературы

[1] Хайкин. С. Однослойный персептрон. Многослойный персептрон // Нейронныйе сети: полный курс, 2-е издание / С. Хайкин. М. : Вильямс, 2006. С. 172–341.

[2] LeCun Y., Bengio Y. Convolutional networks for images, speech, and time series // The handbook of brain theory and neural networks. Second edition / M. A. Arbib. Cambridge, MA, USA, London, England : The MIT Press, 2003. P. 276–279.

[3] An intriguing failing of convolutional neural networks and the CoordConv solution / R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, J. Yosinski // NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems. Red Hook, NY, USA : Curran Associates Inc., 2018. P. 9628–9639.

Об одном семействе неявно предполных классов, сохраняющих подмножества

Старостин Михаил Васильевич
Московский государственный университет имени М. В. Ломоносова; murmol@bk.ru

Введение

В 70-х годах прошлого века А. В. Кузнецов ввел понятия параметрической и неявной выразимости [1], которые обобщают понятие выразимости по суперпозиции. В той же работе Кузнецов описал множество всех параметрически замкнутых классов.

Позднее О. М. Касим-Заде доказал, что в двузначной логике оператор неявной выразимости эквивалентен оператору параметрического замыкания [2], и тем самым получил описание множества всех неявных расширений в двузначной логике, из которого можно непосредственно получить множество всех

неявно предполных (т. е. максимальных неполных) классов. Описание всех неявно предполных классов в трехзначной логике получено автором (см., например, [3]).

В работе рассматривается обобщение одного из семейств неявно предполных классов в трехзначной логике. Отметим, что некоторые из этих классов рассматривались Е. А. Ореховой [4].

Основные определения

Обозначим через E_k множество $\{0, 1, \ldots, k-1\}$, через P_k — множество всех функций k-значной логики, а через T_A — множество всех функций, сохраняющих подмножество A.

Говорят, что функция $f(x_1, \ldots, x_n) \in P_k$ *неявно выразима* над множеством функций $F \subseteq P_k$, если найдутся такие $A_i, B_i \in [F \cup \{x\}]$, что система уравнений

$$\begin{cases} A_1(x_1, \ldots, x_n, z) = B_1(x_1, \ldots, x_n, z), \\ \qquad\qquad \ldots \\ A_m(x_1, \ldots, x_n, z) = B_m(x_1, \ldots, x_n, z) \end{cases}$$

эквивалентна уравнению $z = f(x_1, \ldots, x_n)$.

Пусть $f(x_1, \ldots, x_n) \in T_{E_k} \subset P_l$. Через $\hat{f}(x_1, \ldots, x_n)$ обозначим такую функцию в P_k, что для любого набора $\tilde{\alpha} \in E_k^n$ выполнено $\hat{f}(\tilde{\alpha}) = f(\tilde{\alpha})$. Такую функцию будем называть *k-ограничением* функции f.

Пусть $W \subseteq P_k$. Через Σ_W будем обозначать множество всех функций в $T_{E_k} \subset P_l$, чье k-ограничение принадлежит W.

Результаты

Часть изложенных здесь результатов была опубликована в статье [5] для случая трехзначной логики.

Устройство классов Σ_W очень схоже с устройством классов W.

Утверждение 1. *Пусть $f(\tilde{x}) \in T_{E_k} \subset P_l$ и $\hat{f}$ — ее k-ограничение. Пусть, кроме того, $W \subseteq P_k$ — замкнутый класс, содержащий селекторную функцию и $[W \cup \{\hat{f}\}] = W_0$. Тогда $[\Sigma_W \cup \{f\}] = \Sigma_{W_0}$.*

Неявная полнота классов вида Σ_W в точности определяется неявной полнотой класса W.

Теорема 1. *Класс функций $W \subseteq P_k$ неявно полон тогда и только тогда, когда полон класс $\Sigma_W \subseteq P_l$.*

Однако с неявной предполнотой классов дела обстоят не столь просто.

Утверждение 2. *Пусть класс $W \subseteq P_k$ неявно предполон и не сохраняет ни одно собственное подмножество E_k. Тогда класс Σ_W будет неявно предполным для любого $l > k$.*

Утверждение 3. *Пусть класс $W \subseteq P_k$ неявно предполон и сохраняет подмножество $E_m \subset E_k$. Тогда если m-ограничение класса W неявно не полно в P_m, то класс Σ_W не будет неявно предполным для всех $l > k$.*

Таким образом, если класс W можно представить в виде $\Sigma_{W'}$ для некоторого класса W', то класс Σ_W не будет неявно предполным.

Работа выполнена при финансовой поддержке Минобрнауки России в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2022-284.

Список литературы

[1] Кузнецов А. В. О средствах для обнаружения невыводимости или невыразимости // Логический вывод. М. : Наука, 1979. С. 5–33.

[2] Касим-Заде О. М. О неявной выразимости булевых функций // Вестник Московского университета. Серия 1. Математика. Механика. 1995. №2. С. 44–49.

[3] Старостин М. В. Неявно предполные классы и критерий неявной полноты в трехзначной логике // Вестник Московского университета. Серия 1. Математика. Механика. 2018. № 2. С. 56–59.

[4] Орехова Е. А. О критерии неявной шефферовости в трёхзначной логике // Дискретный анализ и исследование операций. Серия 1. 2003. Т. 10, № 3. С. 82–105.

[5] Старостин М. В. Неявно предполные классы и критерий неявной полноты в трехзначной логике // Вестник Московского университета. Серия 1. Математика. Механика. 2018. № 6. С. 36–40.

О числе разбиений на большие подкубы

Таранников Юрий Валерьевич
Московский государственный университет имени М. В. Ломоносова; yutarann@gmail.com

Пусть q, m, n — целые числа, $q \geqslant 2$, $n \geqslant m \geqslant 0$. Подкубом размерности $n - m$ в $\mathbf{Z}_q^n$ называется такое подмножество наборов $\mathbf{Z}_q^n$, у которого некоторые m компонент фиксированы, а каждая из остальных $n - m$ компонент пробегает всевозможные значения из $\mathbf{Z}_q$.

При *разбиении на подкубы* каждый набор из $\mathbf{Z}_q^n$ должен попасть ровно в один подкуб. Разбиение на подкубы называется *A-примитивным*, если каждая компонента зафиксирована хотя бы в одном из подкубов разбиения.

Наиболее известна задача о разбиении на подкубы малой размерности. Так, если все подкубы разбиения булева куба имеют размерность 1, то эти подкубы являются ребрами, а разбиения называются совершенными паросочетаниями, и задача об их числе хорошо известна. В [1] рассматриваются задачи разбиения булева куба на подкубы, преимущественно малых размерностей, которые в том числе могут быть разными в составе одного разбиения.

Разбиения на подкубы (не обязательно одной размерности) с дополнительным условием неприводимости исследуются в [2].

Главным предметом изучения в [3] были разбиения на аффинные подпространства, а для разбиений на подкубы, являющихся частным случаем разбиений на аффинные подпространства, доказаны следующие утверждения, ориентированные на разбиения на подкубы одинаковой большой размерности.

Теорема 1 ([3]). *Пусть $q \geqslant 2$. Для любого натурального m существует наименьшее натуральное $N = N_q^{\mathrm{coord}}(m)$, что при $n > N$ не существует A-примитивных разбиений $\mathbf{Z}_q^n$ на q^m подкубов размерности $n - m$.*

Теорема 2 ([3]). *Справедлива формула*

$$c_q^{\mathrm{coord}}(n,m) = \sum_{h=m}^{N_q^{\mathrm{coord}}(m)} \binom{n}{h} c_q^{\mathrm{coord}*}(h,m), \tag{1}$$

где: $c_q^{\mathrm{coord}}(n,m)$ — число различных неупорядоченных разбиений $\mathbf{Z}_q^n$ на q^m подкубов размерности $n-m$; $c_q^{\mathrm{coord}*}(n,m)$ — число различных неупорядоченных A-примитивных разбиений $\mathbf{Z}_q^n$ на q^m подкубов размерности $n-m$; $\binom{n}{h}$ — обычный биномиальный коэффициент.

Теорема 3 ([3]). *Пусть q и m фиксированы, $n \to \infty$. Тогда имеет место асимптотика*

$$c_q^{\mathrm{coord}}(n,m) \sim C' n^{N_q^{\mathrm{coord}}(m)},$$

где $C' = \frac{c_q^{\mathrm{coord}*}(N_q^{\mathrm{coord}}(m),m)}{N_q^{\mathrm{coord}}(m)!}$.

Также в [3] получены оценки $\frac{q^m-1}{q-1} \leqslant N_q^{\mathrm{coord}}(m) \leqslant m \cdot q^{m-1}$ и установлены точные значения $N_q^{\mathrm{coord}}(2) = q+1$.

Новые результаты

Теорема 4. *Имеют место точные значения $N_2^{\mathrm{coord}}(4) = 15$, $N_2^{\mathrm{coord}}(5) = 31$, $N_q^{\mathrm{coord}}(3) = q^2+q+1$, $c_2^{\mathrm{coord}*}(15,4) = 15!$, $c_2^{\mathrm{coord}*}(31,5) = 31!$, $c_q^{\mathrm{coord}*}(q^2+q+1,3) = (q^2+q+1)!$.*

Идея доказательства. Звездным паттерном подкуба называется набор длины n над $\mathbf{Z}_q \cup \{*\}$, где элементы $\mathbf{Z}$ соответствуют зафиксированным компонентам, в то время как $*$ соответствует «свободной» компоненте.

Например, набор $(0,*,1,0,*)$ является звездным паттерном следующего подкуба в $\mathbf{Z}_2^5$:

$$\begin{array}{ll}\{ & (0,0,1,0,0),\\ & (0,0,1,0,1),\\ & (0,1,1,0,0),\\ & (0,1,1,0,1) \quad \}.\end{array}$$

Матрица, по строкам которой выписаны звездные паттерны всех подкубов разбиения, называется *звездной матрицей* разбиения.

Например, звездная матрица

$$\begin{pmatrix} 0, & 0, & * \\ 0, & 1, & * \\ 1, & *, & 0 \\ 1, & *, & 1 \end{pmatrix}$$

задает разбиение $\mathbf{Z}_2^3$ на 2^2 подкубов размерности $3-2=1$ каждый.

Легко видеть, что разбиение $\mathbf{Z}_q^n$ на q^m подкубов одинаковой размерности $n-m$ задается звездной матрицей размера $q^m \times n$ и является A-примитивным тогда и только тогда, когда его звездная матрица не содержит столбца из одних $*$.

Для установления значений, выписанных в формулировке теоремы, производится анализ звездных матриц размера $q^m \times N_q^{\mathrm{coord}}(m)$, задающих A-примитивное разбиение. Используются леммы, дадим формулировки наиболее важных из них.

Лемма 1. *В звездной матрице разбиения для любых двух строк найдется столбец, имеющий в этих строках разные значения из* $\mathbf{Z}_q$.

Указание к *доказательству* леммы 1. В противном случае можно заменить звездочки в этих строках на числа так, что обе строки станут одинаковыми, поэтому соответствующие подкубы пересекаются, чего быть не может.

Лемма 2. *В звездной матрице разбиения на подкубы одинаковой размерности в любом столбце все значения из* $\mathbf{Z}_q$ *встречаются одинаковое число раз.*

Указание к *доказательству* леммы 2. Рассмотрим i-й столбец. Если некоторая строка звездной матрицы имеет $*$ в i-м столбце, то соответствующий подкуб для каждого $a \in \mathbf{Z}_q^n$ содержит в точности q^{n-m-1} наборов со значением a в i-м столбце. Если некоторая строка звездной матрицы имеет a в i-м столбце, $a \in \mathbf{Z}_q$, то все q^{n-m} наборов соответствующего подкуба имеют a

в i-м столбце. Любой набор из $\mathbf{Z}_q^n$ принадлежит в точности одному подкубу разбиения. Отсюда вытекает утверждение леммы 2.

Ряд использованных лемм, формулировки которых мы здесь не приводим, являются в том или ином виде обобщениями леммы 2 на совокупности более чем из одного столбца.

Лемма 3. *Пусть звездная матрица A-примитивного разбиения $\mathbf{Z}_q^n$ на q^{n-m} подкубов размерности $n-m$ каждый имеет размер $q^m \times N_q^{\mathrm{coord}}(m)$ и содержит столбец без звездочек. Тогда $N_q^{\mathrm{coord}}(m) = qN_q^{\mathrm{coord}}(m-1)+1$.*

Из теорем 3 и 4 вытекают асимптотические формулы, выписанные в следующей теореме.

Теорема 5. *При $n \to \infty$ справедливы асимптотики*

$$c_2^{\mathrm{coord}}(n,4) \sim n^{15},$$
$$c_2^{\mathrm{coord}}(n,5) \sim n^{31},$$
$$c_q^{\mathrm{coord}}(n,3) \sim n^{q^2+q+1}.$$

Список литературы

[1] Noga A., Balogh J., Potapov V. N. Partitioning the hypercube into smaller hypercubes // arXiv preprint 2401.00299v2. 2024. (available at https://arxiv.org/abs/2401.00299v2).

[2] Irreducible subcube partitions / Y. Filmus, E. A. Hirsch, S. Kurz, F. Ihringer, A. Riazanov, A. V. Smal, M. Vinuals // The Electronic Journal of Combinatorics. 2023. Vol. 30, no. 3. Article P3.29.

[3] Таранников Ю. В. О существовании разбиений, примитивных по Агиевичу // Дискретный анализ и исследование операций. 2022. Т. 29, № 4. С. 104–123.

Вычисление некоторых характеристик всех неизоморфных строгих порядков на конечном множестве

Тензина Виктория Васильевна
Московский государственный университет имени М. В. Ломоносова; viktoria.tenzina@math.msu.ru

Бинарное отношение на множестве X — это любое подмножество множества $X^2 = X \times X$. Два отношения $R_1 \subseteq X \times X$ и $R_2 \subseteq Y \times Y$ называются изоморфными, если существует биективное отображение $f : X \mapsto Y$ такое, что $(x,y) \in R_1 \Leftrightarrow (f(x), f(y)) \in R_2$. Если $R_1 = R_2 = R$, то f называется

автоморфизмом. Если $R \subseteq X \times X$ и отображение $f : X \mapsto X$ таково, что $(x, y) \in R \Rightarrow (f(x), f(y)) \in R$, то f — *эндоморфизм.*

Каждому отношению R можно сопоставить множество всех его автоморфизмов, которое образует группу относительно композиции. Отношение изоморфизма разбивает все бинарные отношения заданного вида на классы эквивалентности. Можно перечислять все отношения заданного вида, а можно с точностью до изоморфизма, то есть учитывая только одного представителя из заданного класса изоморфизма.

Перечисление всех отношений заданного вида или с точностью до изоморфизма — хорошо известная задача (см. [1]). Например, этому посвящено много различных таблиц Слоэна (см. [2]). Нас будет интересовать число таких отношений с тривиальной группой автоморфизмов, то есть состоящей только из тождественного автоморфизма.

Теорема 1. *Пусть X — конечное множество с n элементами и пусть $\mathcal{K}$ — некоторое подмножество всех бинарных отношений на X такое, что если $\rho \in \mathcal{K}$, то и его изоморфный образ также из $\mathcal{K}$. Обозначим через S число всех бинарных отношений из $\mathcal{K}$, а через N число всех неизоморфных бинарных отношений также из $\mathcal{K}$. Тогда если A — количество неизоморфных бинарных отношений из $\mathcal{K}$ с тривиальной группой автоморфизмов, то*

$$\frac{2S}{n!} - N \leqslant A \leqslant \frac{S - N}{n! - 1}.$$

Каждому бинарному отношению R на конечном множестве из n элементов можно сопоставить ориентированный граф G и матрицу смежности A следующим образом: граф состоит из n вершин, а из i в j есть ребро тогда и только тогда, когда $(i, j) \in R$; матрица (a_{ij}) состоит из n строк и n столбцов, а элемент матрицы $a_{ij} = 1$ тогда и только тогда, когда $(i, j) \in R$, иначе $a_{ij} = 0$.

Теорема 2. *Доля асимметричных графов (чья группа автоморфизмов тривиальна) среди всех неизоморфных простых графов с конечным числом вершин n стремится к 1 при $n \longrightarrow \infty$.*

Если бинарное отношение антирефлексивно, транзитивно и антисимметрично, то оно называется строгим порядком. Частично упорядоченное множество называется линейным порядком, если все элементы в нём сравнимы.

Теорема 3. *Любое строго упорядоченное конечное множество, обладающее парой различных несравнимых элементов, можно нетождественно эндоморфно вложить в себя. Более того, образом такого эндоморфизма является линейно упорядоченное подмножество.*

Так как каждому строго упорядоченному множеству можно взаимооднозначно сопоставить конечное топологическое пространство с тем же числом элементов, то можно доказать следующее следствие.

Следствие. *Пусть X — конечное топологическое T_0-пространство. Тогда существует непрерывное отображение $f : X \mapsto X$ такое, что индуцированная топология на $f(X)$ совпадает с топологией линейного упорядочивания.*

Если отношение R является строгим порядком, то соответствующий граф не имеет петель, любые две различные вершины i, j соединены не более чем одним ребром, если есть рёбра из i в j и из j в k, то есть ребро из i в k, а для элементов матрицы $A = (a_{ij})$ выполняется: 1) на главной диагонали нули, 2) $a_{ij} + a_{ji} \leqslant 1$, 3) $a_{ij} = 1 \ \& \ a_{jk} = 1 \Rightarrow a_{ik} = 1$.

Пусть R — отношение строгого порядка. Так как граф такого отношения ацикличен, то, воспользовавшись топологической сортировкой, можем переупорядочить вершины графа так, чтобы булева матрица этого отношения $(a)_{ij}$ $(i, j \in \{1, \ldots, n\})$ была верхнетреугольной. Заметим, что на диагонали этой матрицы стоят нули. Сопоставим каждому такому R двоичное число $a_{n-1,n}a_{n-2,n}a_{n-2,n-1}a_{n-3,n}a_{n-3,n-1}a_{n-3,n-2}\ldots a_{1,n}a_{1,n-1}\ldots a_{1,2}$. Это код данной матрицы. Перебирая всевозможные отношения, изоморфные заданному R, с верхнетреугольными матрицами, найдём максимальный по значению код. Назовём его *максикодом*, а соответствующие матрицы *максикодными*.

Лемма 1. *Пусть для некоторого натурального n двоичный код $a_{n-1,n}a_{n-2,n}a_{n-2,n-1}a_{n-3,n}a_{n-3,n-1}a_{n-3,n-2}\ldots a_{1,n}a_{1,n-1}\ldots a_{1,2}$ является максикодом для некоторого строгого порядка на множестве из n элементов. Тогда $a_{n-1,n}a_{n-2,n}a_{n-2,n-1}a_{n-3,n}a_{n-3,n-1}a_{n-3,n-2}\ldots a_{2,n}a_{2,n-1}\ldots a_{2,3}$ (вычёркиваем младшие разряды в количестве $n-1$) будет максикодом для некоторого строгого порядка на множестве из $n-1$ элементов. Обратно: если к максикоду некоторого строгого порядка на множестве из n элементов приписать справа n нулей, то получится максикод для некоторого строгого порядка на множестве из $n+1$ элементов.*

Итак, если мы сможем перечислить все максикоды (назовём их нижними) для множества всех строгих порядков на множестве из n элементов, то, беря каждый максикод, преобразовав его в верхнетреугольную матрицу и приписав сверху строчки из нулей и единиц, сохраняющие транзитивность, а самый левый столбец заполнив нулями, получим множество кандидатов, содержащее всевозможные максикодные матрицы размера $n+1$ на $n+1$. В итоге получаем дерево максикодов.

На основе этой идеи создано программное обеспечение на языке C++, создающее на каждом шаге для заданного n файл со списком максикодов

и некоторыми его характеристиками (например, порядком группы автоморфизмов для заданного максикода) на основе файла для $n-1$. Сам алгоритм позволяет распараллеливать вычисления на несколько компьютеров при обработке очередного файла, так как каждый максикод читаемого файла обрабатывается независимо. При проверке кандидата на то, что он является максикодом, необходимо перечислить все топологические сортировки, сохраняющие соответствующий строгий порядок нижнего уровня, при этом про половину кандидатов заведомо известно, что они не подходят. Перечисление всех топологических сортировок осуществляется на основе алгоритма из [3] (с. 395). Если нижний максикод соответсвовал отношению с тривиальной группой автоморфизмов, то для некоторых кандидатов сразу очевидно, что им соответствует максикод также с тривиальной группой автоморфизмов. Заметим, что каждому автоморфизму соответствует некоторая топологическая сортировка, и поэтому все потенциальные автоморфизмы надо искать только среди них.

По каждому такому файлу строится суммарная статистика для n. Назовём максикод тривиальным, если он соответствует строгому порядку с тривиальной группой автоморфизмов, назовём связным, если соответствующий граф связен. Например, для $n = 11$ получено: число всех максикодов равно 46749427, тривиальных максикодов — 26554439, связных максикодов — 43944974, связных тривиальных — 25229911. Заметим, что сперва для небольших n были программно найдены все строгие порядки с тривиальной полугруппой эндоморфизмов, а потом доказана теорема 3.

Список литературы

[1] Харари Ф., Палмер Э. Перечисление графов // М. : Мир, 1977. 324 с.

[2] Sloane N. J. A., The OEIS Community. The on-line encyclopedia of integer sequences. The OEIS Foundation Inc. Retrieved 19.04.2025 from http://oeis.org.

[3] Кнут Д. Э. Искусство программирования, том 4, А. Комбинаторные алгоритмы, часть 1. М. : И. Д. Вильямс, 2013.

О типах деревьев с размером приведённой древесной колоды 2

Томилов Дмитрий Александрович, Абросимов Михаил Борисович
Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского; tomilov.d.a@mail.ru, mic@rambler.ru

В работе рассматриваются неориентированные графы. Основные определения следуют работам [1, 2].

Определение. *Дерево* — это связный граф, в котором нет циклов. Вершина степени 1 в дереве называется висячей или листом.

Определение. *Подграф* — граф, получающийся удалением произвольного количества вершин и всех инцидентных с ними рёбер из исходного графа.

Определение. *Максимальный подграф* — подграф, получающийся удалением одной произвольной вершины и всех её рёбер.

Определение. *Колодой графа* называется список его максимальных подграфов.

Определение. *Максимальное поддерево дерева* — дерево, получающееся удалением одной произвольной висячей вершины.

Определение. *Древесной колодой дерева* будем называть список его максимальных поддеревьев.

Определение. *Приведённая древесная колода дерева* — список попарно неизоморфных максимальных поддеревьев дерева.

Один из традиционных вопросов, рассматриваемых в различных разделах математики, касается связи между структурой объекта и его подструктурами. Большой интерес представляет то, в какой мере структура объекта определяется структурой его частей. Особое значение имеет вопрос о том, можно ли реконструировать объект по его частям.

Гипотеза реконструируемости Келли — Улама является одной из самых знаменитых открытых проблем в теории графов.

Гипотеза (Келли — Улама о реконструируемости, 1945)**.** *Каждый неориентированный граф на более чем двух вершинах реконструируем.*

Для деревьев гипотеза Келли — Улама была доказана Келли [3]. Харари и Палмер [4] доказали, что деревья реконструируемы и по максимальным поддеревьям. Также было доказано, что деревья реконструируемы и по приведённой древесной колоде [5].

В данной работе рассматривается задача описания деревьев с заданным размером колоды. Ранее были получены некоторые результаты о деревьях с размером приведённой древесной колоды 1 [6, 7]. В частности, в этих работах были описаны два типа звёзд: центральные SC и бицентральные SB. Это деревья, которые имеют размер приведённой древесной колоды 1, то есть все их максимальные поддеревья попарно изоморфны.

Рассмотрим деревья с размером приведённой древесной колоды 2. Пусть SC — какая-либо центральная звезда, а SB — какая-либо бицентральная звезда. Рассмотрим также цепь P отдельно, не рассматривая её в рамках SC и SB. Введём операцию объединения графов $\underline{\cup}$ такую, что звезды могут иметь общий центр (то есть SC с $SC - 1$ вершину, SB с SB — две вершины) и,

возможно, еще некоторое количество общих вершин, тогда как P может соединяться только за один из концов. Именно поэтому P не рассматривается здесь как частный случай центральной или бицентральной звезды. Основной результат работы:

Теорема. *Дерево имеет приведённую древесную колоду размера* 2 *тогда и только тогда, когда является одним из следующих четырёх типов деревьев:*

1. $SC \cup P$, *где цепь* P *крепится к вершине* SC.
2. $SC_1 \cup P \cup SC_2$, *где* SC_1 — *какая либо центральная звезда, неизоморфная другой произвольной центральной звезде* SC_2, *причем цепь* P *крепится одним своим концом к вершине* SC_1, *а другим своим концом — к вершине* SC_2.
3. $SC_1 \cup SC_2$ *с общим центром и, возможно, еще некоторым количеством общих вершин, где* SC_1 — *какая-либо центральная звезда, неизоморфная другой произвольной центральной звезде* SC_2.
4. $SB_1 \cup SB_2$ *с общим центром и, возможно, еще некоторым количеством общих вершин, где* SB_1 — *какая-либо бицентральная звезда, неизоморфная другой произвольной бицентральной звезде* SB_2.

Пример дерева с приведённой древесной колодой размера 2 и типа 3 приведен на рисунке 1.

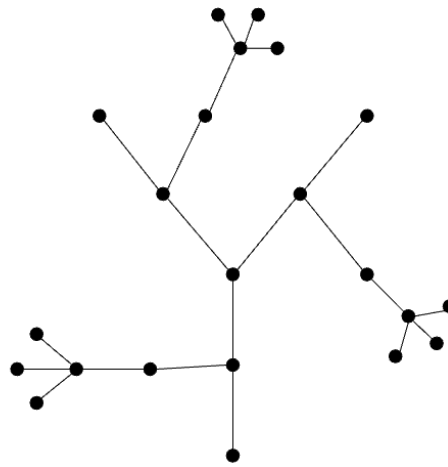

Рис. 1: $SC_{3,2} \cup SC_{3,3,3.1}$

Список литературы

[1] Богомолов А. М., Салий В. Н. Алгебраические основы теории дискретных систем. М. : Наука, Физматлит, 1997. 368 с.

[2] Харари Ф. Теория графов. М. : Мир, 1973. 300 с.

[3] Kelly P. J. A congruence theorem for trees // Pacific Journal of Mathematics. 1957. Vol. 7, no. 1. P. 961–968.

[4] Harary F., Palmer E. The reconstruction of a tree from its maximal subtrees // Canadian Journal of Mathematics. 1966. Vol. 18. P. 803–810.

[5] Manvel B. Reconstruction of trees // Canadian Journal of Mathematics. 1970. Vol. 22, no. 1. P. 55–60.

[6] Абросимов М. Б., Володина П. А. О некоторых свойствах деревьев с размером приведенной колоды 1 // Материалы XIV Международного семинара «Дискретная математика и ее приложения» имени академика О. Б. Лупанова (Москва, МГУ, 20–25 июня 2022 г.). М. : ИПМ им. М. В. Келдыша, 2022. С. 172–174.

[7] Абросимов М. Б., Томилов Д. А. Классификация деревьев, все максимальные поддеревья которых изоморфны // Прикладная дискретная математика. Приложение. 2024. № 17. С. 135–137.

О бесповторно замкнутых классах булевых функций и индуцированных преобразованиях рациональных вероятностей

Трифонова Екатерина Евгеньевна
Институт прикладной математики имени М. В. Келдыша РАН; etrifonova@keldysh.ru

Бесповторное замыкание для классов булевых функций упоминается в работе Ф. И. Салимова [1], вопросы существования конечно порождающих систем рассматривались в работах Р. Л. Схиртладзе, Ф. И. Салимова и Р. М. Колпакова (см. [2–4], а также обзор в [5]).

Данная работа продолжает исследования автора [6] в области преобразования бернуллиевских случайных величин с рациональными вероятностями посредством булевых функций и расширяет представление о бесповторно замкнутых классах булевых функций и их свойствах с точки зрения задачи конечного порождения рациональных вероятностей.

Пусть x — случайная величина, принимающая значение 1 и 0 с вероятностью $\widehat{x}$ и $1-\widehat{x}$ соответственно. Тогда распределение этой случайной величины однозначно определяется значением $\widehat{x} \in [0;1]$.

Будем рассматривать преобразования, осуществляемые в результате подстановки независимых в совокупности случайных величин со значениями 0 и 1 вместо переменных булевых функций. При этом в качестве преобразователей будем брать только булевы функции без фиктивных переменных.

Пусть задана булева функция $f(x_1, \ldots, x_n)\colon \{0,1\}^n \to \{0,1\}$, тогда *вероятностная функция* $\widehat{f}(\widehat{x}_1, \ldots, \widehat{x}_n)\colon [0;1]^n \to [0;1]$, *индуцированная булевой функцией* $f(x_1, \ldots, x_n)$, определяется соотношением:

$$\widehat{f}(\widehat{x}_1, \ldots, \widehat{x}_n) = \sum_{\substack{(x_1,\ldots,x_n):\\ f(x_1,\cdots,x_n)=1}} \prod_{i=1}^{n} (x_i\widehat{x}_i + (1-x_i)(1-\widehat{x}_i)).$$

Вероятностную функцию, индуцированную булевой функцией, можно также записать в виде суммы одночленов с целыми коэффициентами следующим образом:

$$\widehat{f}(\widehat{x}_1, \ldots, \widehat{x}_n) = \sum_{\kappa_1,\ldots,\kappa_n:\in\{0;1\}} \alpha_{\kappa_1\ldots\kappa_n} \widehat{x}_1^{\kappa_1} \ldots \widehat{x}_n^{\kappa_n},$$

где $\widehat{x}_i^0 = 1$, $\widehat{x}_i^1 = \widehat{x}_i$.

Тогда для простого p, $p \geqslant 5$, если:

1) $\alpha_{1\ldots1} = 0$, то функцию $\widehat{f}$ будем называть *p-сократимой первого типа*;
2) $\alpha_{1\ldots1} = p^t A$, где $t \geqslant 1$, $A \in \mathbb{Z}$, $A \mod p \neq 0$, то функцию $\widehat{f}$ будем называть *p-сократимой второго типа*;
3) $\alpha_{1\ldots1} = A$, где $A \in \mathbb{Z}$, $A \mod p \neq 0$, то функцию $\widehat{f}$ будем называть *p-несократимой.*

Заметим, что p-сократимые функции первого типа будут p-сократимыми для любого простого $p \geqslant 5$, а p-сократимые функции второго типа и p-несократимые будут являться таковыми для одних p и не будут для других. Например, индуцированная вероятностная функция $\widehat{f}$ с коэффициентом $\alpha_{1\ldots1} = 5$ является 5-сократимой 2-го типа и r-несократимой для любого простого $r \geqslant 7$, в частности, она является 7-несократимой. Оценки числа p-сократимых функций приведены в [7].

Будем классифицировать булевы функции с точки зрения индуцирования ими p-сократимых или p-несократимых вероятностных функций. Для каждой из булевых функций после удаления фиктивных переменных построим индуцированную функцию. И по тому, какая индуцированная функция получилась, отнесем исходную булеву функцию к одному из трех классов булевых функций, которые будем обозначать как $\mathcal{Z}$, $\mathcal{R}_p$, $\mathcal{N}_p$ — индуцирующие p-сократимые функции первого типа, p-сократимые функции второго типа, p-несократимые функции соответственно.

Будем называть *бесповторным замыканием* $[F]_0$ некоторого множества булевых функций F множество всех булевых функций, представимых над F

бесповторными формулами, т. е. формулами, в которых все переменные различны.

Теорема 1. *Классы $\mathcal{Z}, \mathcal{N}_p, \mathcal{R}_p$ бесповторно замкнуты.*

Теорема 2. *У классов булевых функций $\mathcal{R}_p$ и $\mathcal{N}_p$ нет конечного базиса относительно бесповторного замыкания.*

Теорема 3. *Для классов $\mathcal{R}_p$, $\mathcal{N}_p$, $\mathcal{Z}$ справедливо : $[\mathcal{R}_p \cup \mathcal{N}_p]_0 = \mathcal{R}_p \cup \mathcal{N}_p$, $[\mathcal{N}_p \cup \mathcal{Z}]_0 = \mathcal{N}_p \cup \mathcal{Z}$, $[\mathcal{R}_p \cup \mathcal{Z}]_0 = \mathcal{R}_p \cup \mathcal{Z}$.*

Теорема 4. *Класс всех булевых функций P_2 может быть разбит следующим образом: $P_2 = \mathcal{Z} \sqcup \mathcal{N}_p \sqcup \mathcal{R}_p \sqcup \{0; 1\}$ для p — простого, $p \geqslant 5$.*

Таким образом, поскольку для простых q, r, для которых $q \neq r$ и $q, r \geqslant 5$, имеем, что $\mathcal{N}_q \neq \mathcal{N}_r$ и $\mathcal{R}_q \neq \mathcal{R}_r$, то существует бесконечное множество разбиений всех булевых функций на непересекающиеся бесповторно замкнутые классы булевых функций. Естественным образом возник вопрос, каково место подобных классов в решетке замкнутых классов булевых функций. Ответ на этот вопрос дают следующие теоремы. Заметим, что замкнутые классы булевых функций являются также и бесповторно замкнутыми.

Теорема 5. *Классы булевых функций K_{01}, D_{01}, L_{01} лежат в классе $\mathcal{N}_p$.*

Теорема 6. *Классы булевых функций $MI_1^\infty, MO_0^\infty, SM$ не лежат ни в одном из классов $\mathcal{Z}, \mathcal{N}_p, \mathcal{R}_p$.*

Ранее был получен результат, определяющий некоторые свойства, которыми должно обладать множество булевых функций, индуцирующих p-несократимые функции, чтобы являться конечно порождающим [6]. При изучении класса $\mathcal{N}_5$ удалось дополнить этот результат, что отражено в теореме 7.

Пусть задано множество булевых функций F и множество правильных дробей G. Определим множество *выразимых вероятностей* $V_F(G)$ итерационно. Положим $V_F^1(G) = G$. Для $i \geqslant 1$ положим $V_F^{i+1}(G) = V_F^i(G) \cup \cup \{\widehat{f}(\widehat{x}_1, \ldots, \widehat{x}_n) | f \in F, \widehat{x}_j \in V_F^i(G)\}$. Тогда $V_F(G) = \bigcup_{i=1}^\infty V_F^i(G)$.

Будем говорить, что для простого $p \geqslant 5$ множество булевых функций F является *конечно порождающим в* $\Gamma[p]$, если найдётся такое конечное множество $G \subset \Gamma[p]$, что $V_F(G) = \Gamma[p]$.

Теорема 7. *Класс булевых функций $\mathcal{N}_5$ является конечно порождающим в $\Gamma[5]$, а именно $V_{\mathcal{N}_5}(A(5^2)) = \Gamma[5]$.*

Список литературы

[1] Салимов Ф. И. Об одной системе образующих для алгебр над случайными величинами // Известия высших учебных заведений. Математика. 1981. № 5. С. 78—82.

[2] Салимов Ф. И. Об одном семействе алгебр распределений // Известия высших учебных заведений. Математика. 1988. № 7. С. 64—72.

[3] Схиртладзе Р. Л. О синтезе р-схемы из контактов со случайными дискретными состояниями // Сообщения Академии наук Грузинской ССР. 1961. Т. 26, № 2. С. 181—186.

[4] Колпаков Р. М. Об оценках сложности порождения рациональных чисел вероятностными контактными π-сетями // Вестник Московского университета. Серия 1. Математика. Механика. 1992. № 6. С. 62—65.

[5] Яшунский А. Д. Алгебры вероятностных распределений на конечных множествах // Труды Математического института имени В. А. Стеклова. 2018. Т. 301. С. 320–335.

[6] Трифонова Е. Е. О некоторых свойствах конечно порождающих систем преобразователей p-ичных дробей // Дискретный анализ и исследование операций. 2022. Т. 19, № 4. С. 124–135.

[7] Трифонова Е. Е. О числе р-сократимых индуцированных вероятностных функций // Интеллектуальные системы. Теория и приложения. 2023. Т. 27, вып. 1. С. 134–142.

Квантовый алгоритм для задачи кратчайшей общей суперстроки с возможными ошибками

Хадиев Камиль Равилевич

Казанский (Приволжский) федеральный университет; Казанский физико-технический институт имени Е. К. Завойского; ФИЦ Казанский научный центр РАН; kamilhadi@gmail.com

В данной работе рассматривается квантовый алгоритм для задачи поиска кратчайшей общей суперстроки с возможными ошибками (SCST_k) с точки зрения временной сложности. Формально задача ставится следующим образом. Дан набор из n строк $S = (s^1, \ldots, s^n)$ суммарной длины $L = |s^1| + \cdots + |s^n|$, при этом максимальная длина $d = \max\{|s^1|, \ldots, |s^n|\}$. Данный набор назовем словарем. А также дан целочисленный параметр k. Необходимо составить строку t минимальной длины так, чтобы каждая строка из словаря была подстрокой строки t. Такую строку t назовем суперстрокой. Кроме того, мы разрешаем в одной из строк иметь не более чем k ошибок. Формально, если ошибки допускаются в строке s^j и она размещается в строке t начиная с позиции i, то количество индексов r таких, что $t[i+r-1] \neq s^j[r]$, не должно превышать k.

Данная задача является вариацией задачи биоинформатики, а именно задачи сборки длинной цепочки ДНК из коротких кусочков. На данный момент существуют две основные вариации задачи: De-Novo, в которой нет

примера длинной цепочки ДНК, которую нужно собрать, и Reference-guide, в которой такой пример существует. Задача построения кратчайшей общей суперстроки является возможным решением для вариации De-Novo. Квантовый алгоритм для этой задачи был предложен в работе [1]. Известно [2], что эта задача NP-полна и в классическом случае имеет сложность $O^*(n^2 2^n + L) = \tilde{O}(2^n + L)$, где O^* скрывает не только константный, но и логарифмический множитель, $\tilde{O}$ скрывает полиномиальный множитель относительно n и логарифмический относительно L. В то же время, для использования решения на практике необходимо разрешать допускать ошибки в строках. В связи с этим можно считать вариант задачи, приведенный в данной работе, ближе к практическому применению. Квантовый алгоритм, представленный в данной работе, имеет временную сложность $O^*(n^{7.5}1.728^n + n^3 d + n^2 dk + n^{4.5}\sqrt{L}) = \tilde{O}(1.728^n + dk + \sqrt{L})$.

Алгоритм. Для строки $u = (u_1, \ldots, u_{|u|})$ обозначим за $|u|$ длину строки, за $u[i : j] = (u_i, \ldots, u_j)$ — подстроку. Под сравнением строк мы подразумеваем сравнение в лексикографическом порядке.

Алгоритм состоит из трех последовательных частей.

Часть 1. В рамках этой части вычисляются позиции возможных ошибок для всех пар строк s^i и s^j. Пусть $\ell_i = |s^i|$, $\ell_j = |s^j|$ и $\ell = \min\{\ell_i, \ell_j\}$. Рассмотрим всевозможные пересечения $t \in \{0, \ldots, \ell\}$ этих двух строк в случае, если сначала идет s^i, затем s^j. Для каждой тройки (i, j, t) рассмотрим строку $u = s^i \circ s^j[t+1, \ell_j]$. Вычислим список позиций r, являющихся ошибками, т. е. $s^j_r \neq u_{\ell_i - t + r} = s^i_{\ell_i - t + r}$. В то же время если количество таких позиций более k, мы сохраняем только первые $k+1$ элементов, т. к. это уже означает, что данное пересечение недопустимо в рамках задачи. Формально, вычисляется список $K_{i,j,t} = \{r : s^j_r \neq s^i_{\ell_i - t + r}, 1 \leqslant r \leqslant t\}$, но если длина списка больше k, то сохраняются только первые $k+1$ элементов. Для этого мы рассматриваем всевозможные тройки $i, j \in \{1, \ldots, m\}$, $t \in \{1, \ldots, \ell\}$, а для зафиксированной тройки находим ошибки за $O(k \log t)$, используя технику префиксных хешей на базе вероятностного алгоритма отпечатков [3]. С ней можно ознакомиться, к примеру, в работе [4]. В результате временная сложность этой части — $O(n^2 dk \log d)$.

Часть 2. В рамках этой части мы вычисляем минимальную возможную строку, которую можно составить из трех последовательных строк s^i, s^j и s^b с пересечениями так, что в строке s^j может быть не более k ошибок. При этом используются уже вычисленные значения $K_{i,j,t}$. Пусть $\ell_i = |s^i|$, $\ell_j = |s^j|$, $\ell_b = |s^b|$, $\ell = \min\{\ell_i, \ell_j\}$ и $\ell' = \min\{\ell_j, \ell_b\}$. Рассмотрим всевозможные пересечения $t \in \{0, \ldots, \ell\}$ для строк s^i и s^j, а также $t' \in \{0, \ldots, \ell'\}$ для строк s^j и s^b. Минимальная возможная результирующая строка достигается при максимальной сумме $t + t'$. Для каждой тройки (i, j, b) вычислим соответствующую пару $k_{i,j,b} = (t, t')$. Для пятерки (i, j, b, t, t') имеется два случая. В

первом случае $t + t' \leqslant \ell_j$. Следовательно, эти три строки составляют строку $u = s^i \circ s^j[t+1, \ell_j - t'] \circ s^b$. В этом случае количество ошибок равно $|K_{i,j,t}| + |K_{j,b,t'}|$. Во втором случае $t + t' > \ell_j$. Тогда s^i и s^b тоже пересекаются. Другими словами, три строки составляют строку $u = s^i \circ s^b[t' + t - \ell_j + 1, \ell_b]$. Тогда должно выполняться условие $s^i[\ell_i - (t' + t - \ell_j) + 1, \ell_i] = s^b[1, t' + t - \ell_j,]$, т. к. ошибки могут быть только в строке s^j, а строки s^i и s^b должны пересекаться без ошибок. В этом случае количество ошибок равно сумме $K_{i,j,t}$ и числу ошибок при сопоставлении $s^j[t+1, \ell_j]$ и $s^b[t' + t - \ell_j + 1, \ell_b]$.

Для вычисления второй величины необходимо вычислить максимальный индекс q элемента списка $K_{j,b,t'}$, который меньше или равен $t + t' - \ell_j$. Это можно сделать с помощью бинарного поиска. Тогда второе значение в сумме равно $|K_{j,b,t'}| - q$, а в целом количество ошибок равно $|K_{i,j,t}| + |K_{j,b,t'}| - q$. Определим функцию $f_{i,j,b}(t, t')$, которая возвращает $t + t'$ в случае, если $s^i[\ell_i - (t' + t - \ell_j) + 1, \ell_i] = s^b[1, t' + t - \ell_j]$ и $|K_{i,j,t}| + |K_{j,b,t'}| - q \leqslant k$, иначе возвращает -1. Значение функции можно вычислить за $O(\log t')$ из-за сложности бинарного поиска. Равенство частей строк s^i и s^b можно определить за $O(1)$ с помощью префиксных хешей. В результате нашей целью становится вычисление максимума функции $f_{i,j,b}$, при этом размер пространства поиска не превышает d^2. Это можно сделать при помощи квантового алгоритма поиска максимума [5, 6] за время $O(d(\log d)^2)$.

Часть 3. Эта часть алгоритма является основной. В рамках нее перебираются все строки s^j, в которых мы предполагаем встретить ошибку, а также две строки s^i и s^b, находящиеся слева и справа от нее. Пусть $(t, t') = k_{i,j,b}$. Тогда вместо строк s^i, s^j и s^b будем рассматривать строку $u = s^i \circ s^j[t+1, \ell_j - t'] \circ s^b$ в случае, если $t + t' \leqslant \ell_j$, и $u = s^i \circ s^b[t' + t - \ell_j + 1, \ell_b]$ в случае, если $t + t' > \ell_j$, где $\ell_j = |s^j|$. Напомним, что u — строка с минимальной возможной длиной, составленной из s^i, s^j и s^b при условии, что количество ошибок в s^j не более k. Таким образом, у нас имееется новый словарь $S' = S \backslash \{s^i, s^j, s^b\} \cup \{u\}$ из $n - 2$ строк. Для данного словаря мы вычисляем кратчайшую общую суперстроку с помощью квантового алгоритма из работы [1]. Определим функцию $g(i, j, b)$, которая возвращает длину соответствующей кратчайшей общей суперстроки для тройки s^i, s^j и s^b. Наша цель — найти минимум этой функции при размере области поиска n^3. Здесь мы также можем воспользоваться квантовым алгоритмом поиска минимума [5, 6].

Теорема 1. *Представленный квантовый алгоритм решает задачу* $SCST_k$ *с временной сложностью* $O^*(n^{7.5}1.728^n + n^3d + n^2dk + n^{4.5}\sqrt{L}) = \tilde{O}(1.728^n + dk + \sqrt{L})$ *и вероятностью ошибки не более* 0.3.

Работа выполнена за счет средств Программы стратегического академического лидерства Казанского (Приволжского) федерального университета («ПРИОРИТЕТ-2030»).

Список литературы

[1] Quantum algorithms for the shortest common superstring and text assembling problems / K. Khadiev, C. M. B. Machado, Zeyu Chen, Junde Wu // Quantum Information and Computation. 2024. Vol. 24, no. 3&4. P. 0267–0294.

[2] Vassilevska V. Explicit inapproximability bounds for the shortest superstring problem // Mathematical Foundations of Computer Science 2005. MFCS 2005. Lecture Notes in Computer Science. 2005. Vol. 3618. P. 793–800.

[3] Freivalds R. Fast probabilistic algorithms // Mathematical Foundations of Computer Science 1979. MFCS 1979. Lecture Notes in Computer Science. 1979. Vol. 74. P. 57–69.

[4] Khadiev K., Remidovskii V. Classical and quantum algorithms for constructing text from dictionary problem // Natural Computing. 2021. Vol. 20. P. 713–724.

[5] Dürr C., Høyer P. A quantum algorithm for finding the minimum // arXiv preprint arXiv:quant-ph/9607014. 1996. (available at https://arxiv.org/abs/quant-ph/9607014)

[6] Quantum query complexity of some graph problems / C. Dürr, M. Heiligman, P. Høyer, M. Mhalla // SIAM Journal on Computing. 2006. Vol. 35, no. 6. P. 1310–1328.

Квантовый алгоритм для задачи поиска множества строк из словаря в тексте

Хадиев Камиль Равилевич, Серов Данил Юрьевич
Казанский (Приволжский) федеральный университет; kamilhadi@gmail.com, serovdanilru@gmail.com

В данной работе рассматривается квантовый алгоритм для задачи поиска множества строк из словаря в тексте с точки зрения запросной сложности. Формально задача ставится следующим образом. Даны строка t длины $|t| = n$, которую назовем текстом, и набор из m строк $S = (s^1, \ldots, s^m)$ суммарной длины $L = |s^1| + \cdots + |s^m|$. Данный набор назовем словарем. Для каждой строки s^j необходимо найти набор индексов $I_j = (i_{j,1}, \ldots, i_{j,k_j})$, такой что s^j является подстрокой t, начиная с символа $i_{j,k}$, для всех $1 \leqslant k \leqslant k_j$ для некоторого целого k_j. В классическом случае известен алгоритм Ахо — Корасик [1], который решает задачу и имеет временную и запросную сложность $O(n + L)$. В то же время эта сложность совпадает и с нижней оценкой $\Omega(n + L)$. Мы предлагаем квантовый алгоритм, который имеет запросную

сложность $O^*(n+\sqrt{mL})$, где O^* скрывает не только константный, но и логарифмический множитель. Кроме того, мы показали, что это совпадает и с нижней оценкой на квантовую запросную сложность $\Omega(n+\sqrt{mL})$. Временная сложность разработанного квантового алгоритма отличается от запросной на логарифмический множитель.

Использованные подходы и инструменты. Для строки $u = (u_1, \ldots, u_{|u|})$ обозначим за $|u|$ длину строки, за $u[i:j] = (u_i, \ldots, u_j)$ — подстроку. Сравнивая строки, мы подразумеваем сравнение в лексикографическом порядке. Пусть $suf = (suf_1, \ldots, suf_n)$ — перестановка из чисел от 1 до n, называемая суффиксным массивом для строки t. В нем указаны индексы суффиксов в отсортированном порядке, т. е. $t[suf_i : n] < t[suf_{i+1} : n]$. Согласно [2], его можно построить так, что временная и запросная сложность будет $O(n)$. Пусть $LCP(u, v)$ — длина наибольшего общего префикса для двух строк u и v. Согласно [3, 4], можно построить специальную структуру данных, которая позволит узнать $LCP(t[suf_i : n], t[suf_j : n])$ для произвольных i и j так, что временная и запросная сложность будет $O(1)$. Данный подход требует препроцессинг который работает за $O(n)$. Для произвольной пары строк u и v определим квантовую процедуру $QLCP(u, v, i)$, которая находит $LCP(u, v)$ в предположении, что $u[1:i] = v[1:i]$. Согласно [5], запросная сложность данной процедуры равна $O(\sqrt{b-i})$, где $b = LCP(u, v)$.

Алгоритм. Рассмотрим строку s^j. Если s^j является подстрокой t, начиная с индекса i, тогда s^j — префикс суффикса $t[suf_i : n]$, т. е. $t[suf_i : suf_i + |s^j| - 1] = s^j$. Так как суффиксы упорядочены, то все суффиксы, содержащие s^j в качестве префикса, находятся последовательно. Алгоритм находит два параметра $left_j$ и $right_j$ такие, что s^j является префиксом для всех суффиксов $t[suf_i : n]$, где $i \in \{suf_{left_j}, \ldots, suf_{right_j}\}$, т. е. $I_j = (suf_{left_j}, \ldots, suf_{right_j})$. Рассмотрим алгоритм для поиска $left_j$, параметр $right_j$ находится аналогично. Алгоритм основывается на бинарном поиске. Пусть Le будет левой границей отрезка, в котором мы ищем требуемый элемент, а Ri — правой. Обозначим за $St_i = t[suf_i : n]$ i-ый суффикс, где $i \in \{1, \ldots, n\}$.

Шаг 1. Присвоим $Le \leftarrow 1$ и $Ri \leftarrow n$. Пусть $Llcp \leftarrow QLCP(St_{Le}, s^j)$ будет LCP для первого суффикса и строки s^j. Пусть $Rlcp \leftarrow QLCP(St_{Ri}, s^j)$ будет LCP для последнего суффикса и строки s^j.

Шаг 2. Если $Llcp < |s^j|$ и $s^j < St_1$, т. е. $Llcp < |s^j|$ и $s^j[Llcp+1] < St_1[Llcp+1]$, то можно сказать, что s^j меньше любого суффикса t и не является префиксом любого из суффиксов t. В этом случае I_j — пустая и мы останавливаем алгоритм, иначе переходим к Шагу 3.

Шаг 3. Если $Rlcp < |s^j|$ и $s^j > St_n$, т. е. $Rlcp < |s^j|$ и $s^j[Rlcp+1] > St_n[Rlcp+1]$, то можно сказать, что s^j больше любого суф-

фикса t и не является префиксом любого из суффиксов t. В этом случае I_j — пустая и мы останавливаем алгоритм, иначе переходим к Шагу 4.

Шаг 4. Пока $Ri - Le > 1$, мы повторяем следующие шаги, иначе переходим к Шагу 9.

Шаг 5. Пусть $M \leftarrow \lfloor (Le + Ri)/2 \rfloor$.

Шаг 6. Если $Llcp \geqslant Rlcp$, тогда переходим к Шагу 7, а иначе к Шагу 8.

Шаг 7. Сравним $LCP(St_L, St_M)$ и $Llcp$. Напомним, что $LCP(St_{Le}, St_M)$ вычисляется за $O(1)$. Есть один из трех вариантов:

— Если $LCP(St_{Le}, St_M) > Llcp$, то все суффиксы с St_{Le} по St_M такие, что $St_M[Llcp + 1] = \cdots = St_{Le}[Llcp + 1] \neq s^j[Llcp + 1]$, и они не могут иметь s^j в качестве префикса. Тогда мы присваиваем $Le \leftarrow M$ и не изменяем $Llcp$.

— Если $LCP(St_{Le}, St_M) = Llcp$, то все суффиксы с St_{Le} по St_M имеют как минимум префикс длины $Llcp$, общий с s^j. Вычислим $Mlcp = LCP(St_M, s^j)$ с помощью $QLCP(St_M, s^j, Llcp + 1)$. Если $Mlcp = |s^j|$, то мы можем подвинуть правую границу в M и обновить $Rlcp$, т. к. мы ищем самое правое вхождение s^j: $Ri \leftarrow M$ и $Rlcp \leftarrow Mlcp$. Аналогично обновляем R и $Rlcp$ в случае, если $St_M[Mlcp + 1] > s^j[Mlcp + 1]$. Если $St_M[Mlcp + 1] < s^j[Mlcp + 1]$, то $Le \leftarrow M$ и $Llcp \leftarrow Mlcp$.

— Если $LCP(St_{Le}, St_M) < Llcp$, то все суффиксы с St_M по St_R не могут быть префиксом s^j. Следовательно, мы присваиваем $Ri \leftarrow M$ и $Rlcp \leftarrow LCP(St_{Le}, St_M)$.

После этого шага мы переходим к Шагу 4.

Шаг 8. Данный шаг аналогичен Шагу 7, но сравнивается $LCP(St_M, St_{Ri})$ и $Rlcp$. Существует три варианта:

— Если $LCP(St_M, St_{Ri}) > Rlcp$, то $Ri \leftarrow M$ и $Rlcp$ не изменяется.

— Если $LCP(St_M, St_{Ri}) = Rlcp$, то вычисляем $Mlcp = LCP(St_M, s^j)$ с помощью $QLCP(St_M, s^j, Rlcp + 1)$. В этом случае обновляем переменные по тем же правилам, как и во втором варианте Шага 7.

— Если $LCP(St_M, St_{Ri}) < Rlcp$, то $L \leftarrow M$ и $Llcp \leftarrow LCP(St_M, St_{Ri})$.

После этого шага мы переходим к Шагу 4.

Шаг 9. Результатом поиска является Ri, и мы присваиваем $left_j \leftarrow Ri$.

В целом алгоритм состоит в том, чтобы определить $left_j$ и $right_j$ для каждой строки s^j, где $j \in \{1, \ldots, m\}$. Сложность рассмотренного алгоритма приведена далее.

Теорема 1. *Приведенный алгоритм решает задачу, имеет запросную сложность* $O(n + \sqrt{mL \log n} + m \log n)$ *и имеет вероятность ошибки не более* 0.1.

Нижняя оценка на запросную сложность алгоритма приведена в следующей теореме

Теорема 2. *Нижняя оценка на запросную сложность задачи поиска множества строк из словаря в тексте в классическом случае* $-\Omega(n + L)$, *в квантовом случае* $-\Omega(n + \sqrt{mL})$.

Работа выполнена за счет средств Программы стратегического академического лидерства Казанского (Приволжского) федерального университета («ПРИОРИТЕТ-2030»).

Список литературы

[1] Aho A. V., Corasick M. J. Efficient string matching: an aid to bibliographic search // Communications of the ACM. 1975. Vol. 18, no. 6. P. 333–340.

[2] Li Zhize, Li Jian, Huo Hongwei. Optimal in-place suffix sorting // Information and Computation. 2022. Vol. 285. Article 104818.

[3] Linear-time longest-common-prefix computation in suffix arrays and its applications / T. Kasai, Gunho Lee, H. Arimura, S. Arikawa, Kunsoo Park // Combinatorial Pattern Matching. CPM 2001. Lecture Notes in Computer Science. 2001. Vol. 2089. P. 181–192.

[4] Bender M. A., Farach-Colton M. The LCA problem revisited // LATIN 2000: Theoretical Informatics. LATIN 2000. Lecture Notes in Computer Science. 2000. Vol. 1776. P. 88–94.

[5] Khadiev K., Ilikaev A., Vihrovs J. Quantum algorithms for some strings problems based on quantum string comparator //Mathematics. 2022. Vol. 10, no. 3. Article 377.

О реализации классов шахматных позиций управляющими системами

Хелемендик Роман Викторович
Институт прикладной математики имени М. В. Келдыша РАН; romash@keldysh.ru

В классических работах С. В. Яблонского [1] и [2] (совместно с А.А.Ляпуновым) шахматы упоминаются многократно и рассматриваются как один из модельных объектов теории управляющих систем (УС). Согласно [2] в УС выделяются 4 составные части: схема, информация, координаты, функция; в частности, показывается их интерпретация для шахмат. В дальнейших исследованиях многие УС (формулы, схемы из функциональных элементов (СФЭ) и др.) представляются парой $U = \langle \Sigma, \Phi \rangle$, где Σ — структура,

а Φ — функционирование, что позволяет рассматривать их совместно, разрабатывать и применять общие методы анализа, синтеза. В настоящей работе предложены представление шахматных позиций в виде таких УС $U = \langle \Sigma, \Phi \rangle$, а также реализация некоторых классов шахматных позиций средствами УС.

Реализация шахматной позиции в виде УС

Мы будем задавать шахматную позицию булевой функцией (б. ф.) от 8 переменных с помощью содержательного и формального использования всех (не более чем) двухместных булевых функций. С содержательной точки зрения двухместные б. ф. соотвествуют шахматным фигурам на клетках доски, и тогда они называются *индексированными*, а с формальной являются составными элементами (в формулах, СФЭ) при реализации итоговой б. ф. Она строится путем получения по позиции набора индексированных функций, состоящего из следующих 4 этапов:

1. Расположение фигур на доске кодируем набором из 64 двухместных б. ф. $f(x,y)_i$, $1 \leqslant i \leqslant 64$, с возможным уточнением некоторых элементов набора на последующих этапах. Сопоставим белому королю (K), ферзю (Q), ладье (R), слону (B), коню (N), пешке (P) соответствующие двухместные функции: & (т. е. $x\&y$), $\not\to$ (антиимпликацию), $\not\leftarrow$ (левую антиимпликацию), x, y, $\oplus$; аналогичным образом черные фигуры k, q, r, b, n, p обозначаем противоположными функциями: $|$, $\to$, $\leftarrow$, $\bar{x}$, $\bar{y}$, $\sim$; пустое поле записываем константой 0.
2. В случае хода черных и нахождении их короля в поле i заменяем $|_i$ на 1_i.
3. При наличии возможности рокировки белых (черных) в короткую или длинную стороны и расположении соответствующих ладей на исходных позициях заменяем $\not\leftarrow$ ($\leftarrow$) на $\vee$.
4. Если в позиции последний ход был пешкой на два поля, то для пройденного ей поля константа 0 меняется на стрелку Пирса $\downarrow$.

В полученном наборе $\widetilde{f}^{64} = \langle f_1, \ldots, f_k, \ldots, f_{64} \rangle$ нули опускаем; функции f_k будем также обозначать f_{vg} и $f_{\underline{v}g}$, где для k-го поля $g = g(k) = \lfloor (k-1)/8 \rfloor + 1$ есть номер горизонтали, $v = v(k) = ((k-1)\ mod\ 8) + 1$ — номер вертикали, а $\underline{v}$ — буквенное обозначение v-й вертикали. Пусть $\sigma_j \in \{0,1\}$, $1 \leqslant j \leqslant 6$, $g(k) = \Sigma_{j=1}^{3} 2^{3-j}\sigma_j + 1$, $v(k) = \Sigma_{j=1}^{3} 2^{3-j}\sigma_{3+j} + 1$. Тогда $\phi_{g(k)} = x_1^{\sigma_1} x_2^{\sigma_2} x_3^{\sigma_3}$, $\chi_{\underline{v}(k)} = x_4^{\sigma_4} x_5^{\sigma_5} x_6^{\sigma_6}$, $\varphi_{v(k)g(k)} = f_k(x_7, x_8)$, а итоговая б. ф. $f(\widetilde{x}^8)$ имеет вид

$$f(x_1, \ldots, x_8) = \bigvee_{k=1}^{64} \chi_{\underline{v}(k)} \phi_{g(k)} \varphi_{v(k)g(k)}. \qquad (1)$$

Пример 1. Этюд Рети: Kh8, Pc6, ka6, ph5, ход белых; $\widetilde{f}^{64} = \langle \sim_{h5} |_{a6} \oplus_{c6} \&_{h8} \rangle$, $f(\widetilde{x}^8) = \chi_h \phi_5(x_7 \sim x_8) \vee \chi_a \phi_6(x_7 \mid x_8) \vee \chi_c \phi_6(x_7 \oplus x_8) \vee \chi_h \phi_8(x_7 \& x_8)$.

Теорема 1. *Всякая шахматная позиция представима в виде б. ф.* $f(\widetilde{x}^8)$.

Доказательство. Проводится по общей схеме, приведенной в доказательстве теоремы 1 из работы [3]. □

Пусть U_1, U_2, U_3 — виды УС: формулы, СФЭ, многополюсные СФЭ с 3 входами — и пусть $U_i = \langle \Sigma_i, \Phi_i \rangle$, $1 \leqslant i \leqslant 3$, $L_i = L(\Sigma_i)$, $D_i = D(\Sigma_i)$ — длина и глубина Σ_i соотвественно, $m = m_{f(\widetilde{x}^8)}$ — вес б.ф. $f(\widetilde{x}^8)$, т. е. количество ее единичных наборов и $i_1, \ldots, i_h, \ldots, i_m$ — номера всех таких наборов, $0 \leqslant i_h \leqslant 255$, $1 \leqslant h \leqslant m$.

Теорема 2. *Всякая шахматная позиция реализуема в виде* U_1, U_2, U_3.

Доказательство. В силу теоремы 1 и формулы 1 произвольная шахматная позиция может быть реализована как в виде формул, так и в виде СФЭ с 8 входами и одним выходом, реализующей ту же б. ф. В УС $U_3 = U_3(m)$ номер i_h каждого единичного набора является также номером некоторой функции от 3 переменных, поэтому соответствующая ему функция реализуется на выходе h в многополюсной СФЭ с 3 входами и $m = m_{f(\widetilde{x}^8)}$ выходами. □

Реализация классов шахматных позиций в виде УС

Заметим, что не всякая б. ф. от 8 переменных имеет своим прообразом шахматную позицию, а для однозначного определения $f(\widetilde{x}^8)$ необходимо согласованное задание позиции (см. выше этапы 3, 4).

Описание и примеры реализации классов позиций. Класс K_M: шахматно-математические задачи (см. [4]); его детализация: K_{MQ} — задачи о расстановке ферзей, K_{MQS} — задача Эйлера о мирных (безопасных) расстановках ферзей. Решения представимы в виде $\widetilde{f}^{64}$ и $f(\widetilde{x}^8)$, реализуемы в виде УС U_3 при $m = m_{f(\widetilde{x}^8)} = 8$, причем минимальной по глубине ($D_3 = 2$) в U_3 является расстановка $\langle \not\to_{a5} \not\to_{b3} \not\to_{c1} \not\to_{d7} \not\to_{e2} \not\to_{f8} \not\to_{g6} \not\to_{h4} \rangle$, $L_3 \leqslant 12$, хотя для ее отражений $D_3 > 2$, но для поворотов $D_3 = 2$, $L_3 \leqslant 15$.

Классы K_L, $K_\times$, $K_{1/2}$, K_1 (K_0) — *легальных*, *матовых*, *ничейных*, *выигранных* за белых (черных) позиций, соответственно; позиция легальна, если допускается правилами шахмат. Характеризуем эти классы на примере $K_{\{\&|^*\not\leftarrow\}}$ — позиций «король + белая ладья против короля» ($|^*$ есть $|$ или 1). Пусть $\Psi^1_1 = \{\chi_b \phi_6(x_7 \& x_8) \vee \chi_a \phi_8 \vee \chi_{\underline{r}(a)} \phi_8(x_7 \not\leftarrow x_8)\}$, $\underline{r}(a) \in \{c, d, e, f, g, h\}$, $\Psi^1_{\underline{v}} = \{\chi_{\underline{v}}(\phi_6(x_7 \& x_8) \vee \phi_8) \vee \chi_{\underline{r}(\underline{v})} \phi_8(x_7 \not\leftarrow x_8)\}$, $\underline{v} \in \{a, b, c, d\}$, $4 \leqslant r(b) \leqslant 8$, $r(c) \in \{1, 5..8\}$, $r(d) \in \{1..2, 6..8\}$, $\Psi^1 = \Psi^1_1 \cup (\cup^v_{1 \leqslant v \leqslant 4} \Psi^1_{\underline{v}})$, $|\Psi^1| = 27$. В виде U_2 класс Ψ^1 реализуем с параметрами $m = 27$, $L_2 \leqslant 63$, $D_2 \leqslant 5$. Тогда $K_{\{\&1\not\leftarrow\}\times} = (\cup^s_{1 \leqslant s \leqslant 8} \Psi^s)$ перечисляет все матовые позиции (с учетом 8 симметрий квадрата), $K_{\{\&|^*\not\leftarrow\}L} = K_L \cap K_{\{\&|^*\not\leftarrow\}}$, а позиции, в которых пат или черные

своим ходом съедают ладью, составляют класс $K_{\{\&|^*\not\leftarrow\}1/2} = K_{\{\&|^*\not\leftarrow\}L} \backslash K_{\{\&|^*\not\leftarrow\}1}$. При этом от любой позиции из $K_{\{\&|^*\not\leftarrow\}1}$ (выигранной) белые всегда могут перейти к позиции из $K_{\{\&|^*\not\leftarrow\}\times} = K_{\{\&1\not\leftarrow\}\times}$, поэтому $K_{\{\&1\not\leftarrow\}\times}$ является так называемым *выигрывающим базисом* для эндшпилей $K_{\{\&|^*\not\leftarrow\}}$.

Взаимосвязи УС и шахмат. Реализация классов шахматных позиций в терминах УС предоставляет новые возможности для взаимодействия математической кибернетики и шахмат. Это позволяет по табличной классификации из [2] работать с шахматными позициями как с конкретными объектами вида $U = \langle \Sigma, \Phi \rangle$, сочетая макроподход и микроподход, исследовать функционирование, алгоритмизацию, эквивалентные преобразования, эволюцию УС. С другой стороны, знания, технологии, оценки, анализы, накопленные в шахматах, переводимы в область УС, поэтому и взгляд на УС с точки зрения шахмат представляется интересным как в теории, так и на практике.

Список литературы

[1] Яблонский С. В. Основные понятия кибернетики // Проблемы кибернетики. М. : Государственное издательство физико-математической литературы, 1959. Вып. 2. С. 7–38.

[2] Ляпунов А. А., Яблонский С. В. Теоретические проблемы кибернетики // Проблемы кибернетики. М. : Государственное издательство физико-математической литературы, 1963. Вып. 9. С. 5–22.

[3] Хелемендик Р. В. О восьмимерности пространства шахматных позиций и их трансляции в управляющие системы // Научный сервис в сети Интернет: труды XXIV Всероссийской научной конференции (19–22 сентября 2022 г., онлайн). М. : ИПМ им. М. В. Келдыша, 2022. С. 508–525.

[4] Гик Е. Я. Шахматы. Математика. Компьютеры. М. : Издатель «Андрей Ельков», 2013. 336 с.

О единичных проверяющих тестах при константных неисправностях на выходах элементов для формул над базисом жегалкинского типа

Цуй Чжэнюй, Романов Дмитрий Сергеевич

Московский государственный университет имени М. В. Ломоносова;
ourobros1234@gmail.com, romanov@cs.msu.ru

Как известно [1, §§ 2–3 главы 2], формулы над множествами булевых функций могут рассматриваться как частный случай схем из функциональных элементов (СФЭ), а именно как СФЭ с одним выходом без ветвлений на выходах функциональных элементов (ФЭ). В рамках этой парадигмы вполне

корректно поставленной оказывается задача о тестировании формул. Будем считать, что в любой базис неявно входит тождественная функция.

Пусть на СФЭ или формулу S над полным базисом B, реализовывавшую булеву функцию $f(x_1, \ldots, x_n)$, мог подействовать источник неисправностей (ИН) U, способный преобразовать S в любую схему (формулу) из конечного содержащего S множества H схем (формул). Как правило, об ИН предполагается, что он не добавляет новые входы и выходы. Источник неисправностей обычно задается описанием тех поломок, которые он может вызывать в схеме. Тестовое исследование схемы состоит в анализе выходных значений, возникающих в качестве реакций схемы на подачу на входы схемы входных наборов (то есть наборов значений входных переменных).

Константные неисправности на выходах ФЭ заключаются в заменах неисправных ФЭ элементами, реализующими булевы константы. В настоящей работе рассматривается источник O_1^{c} одиночных произвольных константных неисправностей на выходах ФЭ (применительно к формулам над базисом $B_1' = \{x \,\&\, y,\ x \oplus y,\ x \sim y\}$ жегалкинского типа; будут также упоминаться базис Жегалкина $B_1 = \{x \,\&\, y,\ x \oplus y,\ 1\}$) и источник IO_1^{c} одиночных произвольных константных неисправностей на входах и выходах ФЭ.

Множество T входных наборов схемы (формулы) S является *проверяющим тестом* (ПТ) для схемы (формулы) S относительно ИН U тогда и только тогда, когда для любой схемы (формулы) S' из множества H имеет место импликация: если S' реализует булеву функцию $g(x_1, x_2, \ldots, x_n)$, не равную $f(x_1, x_2, \ldots, x_n)$, то в T найдется набор $\tilde{\alpha}$ такой, что $f(\tilde{\alpha}) \neq g(\tilde{\alpha})$.

Число наборов в тесте T именуется длиной теста и обозначается как $L(T)$. Минимальным тестом называется тест минимальной длины. Длина минимального ПТ для схемы или формулы S относительно источника неисправностей U обозначается через $L^{\mathrm{dt}}(U, S)$. Схема (формула) S называется неизбыточной относительно ИН U, если при любой меняющей хоть на каком-то входном наборе прохождение сигналов в схеме (соответственно формуле) одиночной поломке функционального элемента, вызванной источником U, полученная схема (формула) реализует функцию, не равную функции, реализуемой S в отсутствие неисправностей. Длиной ПТ для реализуемой СФЭ (формулами) над базисом B булевой функции f относительно источника неисправностей U называется величина $L^{\mathrm{dt}}_{\mathrm{C},B}(U, f)$ (соответственно $L^{\mathrm{dt}}_{\mathrm{F},B}(U, f)$), равная минимуму по всем неизбыточным реализующим f схемам (соответственно формулам) S над базисом B величин $L^{\mathrm{dt}}(U, S)$. *Функцией Шеннона* (ФШ) *длины проверяющего теста относительно источника неисправностей* U для СФЭ (формул) над базисом B называется величина $L^{\mathrm{dt}}_{\mathrm{C},B}(U, n) = \max\limits_{f \in P_2(n)} L^{\mathrm{dt}}_{\mathrm{C},B}(U, f)$ (соответственно $L^{\mathrm{dt}}_{\mathrm{F},B}(U, n) = \max\limits_{f \in P_2(n)} L^{\mathrm{dt}}_{\mathrm{F},B}(U, f)$).

Специально тесты для формул ранее не рассматривались, хотя некоторые результаты для СФЭ — например, константные верхние оценки для ФШ длины единичного проверяющего теста относительно инверсных неисправностей на выходах ФЭ в СФЭ над некоторыми базисами [2] — фактически являются и соответствующими результатами для формул. Однако константные верхние оценки ФШ длины единичного ПТ при произвольных константных неисправностях на выходах элементов неизвестны. Приведем обзор результатов, близких к изучаемой здесь задаче (речь пойдет об оценках функций Шеннона длины единичного ПТ относительно произвольных константных неисправностей на выходах ФЭ в СФЭ). По умолчанию оценки функций Шеннона справедливы для любого n, $n \in \mathbb{N}$.

В [3] фактически доказано, что в базисе B_1 любую булеву функцию n переменных можно смоделировать формулой с одним дополнительным аргументом, допускающей универсальный единичный ПТ длины не более $n+4$ относительно IO_1^{c}. В [4] эта оценка понижена до $n+3$, а в [5, с. 113–116] фактически доказано, что $L^{\mathrm{dt}}_{\mathrm{F},B_1}(IO_1^{\mathrm{c}}, n) \leqslant n+3$. В работах [6–8] установлено, что в произвольном полном базисе B имеет место верхняя оценка $L^{\mathrm{dt}}_{\mathrm{C},B}(O_1^{\mathrm{c}}, n) \leqslant n+3$ (в ряде базисов фактически используются формулы). В [9] доказано, что $L^{\mathrm{dt}}_{\mathrm{C},B_1'}(IO_1^{\mathrm{c}}, n) \leqslant 16$. В [10] доказано, что для произвольного конечного полного базиса B имеют место оценки $2 \leqslant L^{\mathrm{dt}}_{\mathrm{C},B}(O_1^{\mathrm{c}}, n) \leqslant 4$. В [11] доказано, что при $n \geqslant 3$ в любом полном базисе B, содержащемся в множестве элементарных конъюнкций с одинаковыми степенями переменных, линейных функций двух переменных и функций, представляющих собой конъюнкцию $x_1\bar{x}_2$ и некоторой функции, $L^{\mathrm{dt}}_{\mathrm{C},B}(O_1^{\mathrm{c}}, n) \geqslant 3$. В [12] доказано, что $L^{\mathrm{dt}}_{\mathrm{C},\{xy,\,\bar{x},\,x\oplus y\oplus z\}}(O_1^{\mathrm{c}}, n) = 2$. Здесь же установлен следующий результат.

Теорема 1. *При любом натуральном n имеет место оценка* $L^{\mathrm{dt}}_{\mathrm{F},B_1'}(O_1^{\mathrm{c}}, n) \leqslant 3$.

Идея доказательства основана на разложении функции по одной переменной с представлением компонент в виде полиномов Жегалкина.

В силу нижней оценки $L^{\mathrm{dt}}_{\mathrm{F},B_1'}(O_1^{\mathrm{c}}, x_1 \vee x_2) \geqslant 3$ выводится следующее утверждение.

Теорема 2. *При любом натуральном n, $n \geqslant 2$, имеет место равенство* $L^{\mathrm{dt}}_{\mathrm{F},B_1'}(O_1^{\mathrm{c}}, n) = 3$.

Работа выполнена при финансовой поддержке Минобрнауки в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2022-284.

Список литературы

[1] Ложкин С. А. Лекции по основам кибернетики. М. : Издательский отдел ф-та ВМиК МГУ, 2004. 256 с.

[2] Редькин Н. П. Единичные проверяющие тесты для схем при инверсных неисправностях элементов // Математические вопросы кибернетики. М. : Физматлит, 2003. Вып. 12. С. 217–230.

[3] Reddy S. M. Easily testable realization for logic functions // IEEE Transactions on Computers. 1972. Vol. C-21, no. 11. P. 124–141.

[4] Kodandapani K. L. A note on easily testable realizations for logic functions // IEEE Transactions on Computers. 1974. Vol. C-23, no. 3. P. 332–333.

[5] Редькин Н. П. Надежность и диагностика схем. М. : Издательство Московского университета, 1992. 192 с.

[6] Коляда С. С. О единичных проверяющих тестах для константных неисправностей на выходах функциональных элементов // Вестник Московского университета. Серия 1. Математика. Механика. 2011. № 6. С. 47–49.

[7] Коляда С. С. Единичные проверяющие тесты для схем из функциональных элементов в базисах из элементов, имеющих не более двух входов // Дискретный анализ и исследование операций. 2013. Т. 20, № 2. С. 58–74.

[8] Коляда С. С. Единичные проверяющие тесты для схем из функциональных элементов // Вестник Московского университета. Серия 1. Математика. Механика. 2013. № 4. С. 32–34.

[9] Романов Д. С., Романова Е. Ю. Метод синтеза неизбыточных схем, допускающих единичные проверяющие тесты константной длины // Дискретная математика. 2017. Т. 29, вып. 4. С. 87–105.

[10] Романов Д. С. Метод синтеза легкотестируемых схем, допускающих единичные проверяющие тесты константной длины // Дискретная математика. 2014. Т. 26, вып. 2. С. 100–130.

[11] Попков К. А. Нижние оценки длин единичных тестов для схем из функциональных элементов // Дискретная математика. 2017. Т. 29, вып. 2. С. 53–69.

[12] Попков К. А. Короткие единичные тесты для схем при произвольных константных неисправностях на выходах элементов // Дискретная математика. 2018. Т. 30, вып. 3. С. 99–116.

К вопросу о простоте регулярных турниров

Шабаркова Александра Олеговна, Абросимов Михаил Борисович
Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского; shabarkova_alex.andra@mail.ru, mic@rambler.ru

Введение

Напомним, что турниром называется полный направленный граф. Классом эквивалентности ϵ на множестве S, соответствующим элемен-

ту, называется множество $\epsilon(x) = y \in S : x \sim y$. Пусть ϵ — некоторое отношение эквивалентности на множестве вершин V орграфа $\overrightarrow{G}$. Факторграфом орграфа $\overrightarrow{G}$ по эквивалентности ϵ называется орграф $\overrightarrow{G}/\epsilon = (V/\epsilon, \alpha_\epsilon)$, где V/ϵ — множество классов эквивалентности ϵ; $\alpha_\epsilon = (\epsilon(v_1), \epsilon(v_2)) : \exists u_1 \in \epsilon(v_1), u_2 \in \epsilon(v_2)(u_1, u_2) \in \alpha)$.

Конгруэнция турнира $\overrightarrow{T} = (V, \alpha)$ — это такая эквивалентность на множестве его вершин, что факторграф по ней является турниром. То есть конгруэнция турнира $\overrightarrow{T} = (V, \alpha)$ — это такая эквивалентность $\theta \subseteq V \times V$, что никакие два различных θ-класса не имеют встречных дуг [1]. Con $\overrightarrow{T}$ — это совокупность всех конгруэнций турнира $\overrightarrow{T}$. Con $\overrightarrow{T}$ образует решётку [2]. Турнир $\overrightarrow{T} = (V, \alpha)$ называется простым, если решётка Con $\overrightarrow{T}$ двухэлементна, т. е. если $\overrightarrow{T}$ не содержит собственных нетождественных конгруэнций. Граф называется регулярным, если все его вершины имеют одинаковые степени захода и исхода.

Строению турниров посвящено много работ, см. например [3]. Простым турнирам посвящена работа [4]. Известно, что у каждого турнира имеется вершинное 1-расширение до простого турнира [5].

В данной работе мы рассмотрим регулярные турниры относительно свойства простоты, опишем их структуру и количество. Начнём с простого регулярного турнира.

Простые регулярные турниры

Теорема 1. *Для каждого нечётного n существует по крайней мере один простой регулярный турнир. При некоторой нумерации вершин список смежности такого турнира можно представить в следующем виде:*

$$v_i : v_{i+j} \mod n, j = \overline{1, \tfrac{n-1}{2}}, i = \overline{1, n}.$$

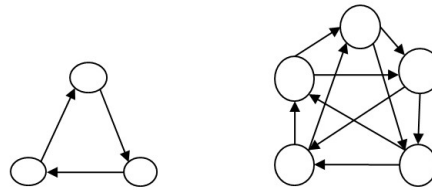

Рис. 1: Простые регулярные турниры размерности 3 и 5 со структурой, описанной в теореме 1.

На рисунке 1 демонстрируются простые регулярные турниры с описанной в теореме 1 структурой с количеством вершин 3 и 5.

Регулярный турнир также может простым не являться. Покажем это.

Регулярные турниры, не являющиеся простыми

Теорема 2. *Для турнира размерности $n = 3k$ существует не менее $\overline{C}^3_{|k|}$ регулярных турниров, не являющихся простыми, где $|k|$ — количество регулярных турниров размерности k.*

Обобщим полученный результат.

Теорема 3. *Для турнира размерности $n = mk$ существует не менее $\overline{C}^m_{|k|}$ регулярных турниров, не являющихся простыми, где $|k|$ — количество регулярных турниров размерности k, а m — нечётное.*

Список литературы

[1] Богомолов А. М., Салий В. Н. Алгебраические основы теории дискретных систем. М. : Наука, Физматлит, 1997. 368 с.

[2] Киреева А. В. Конгруэнции турниров // Студенты — ускорению научного прогресса: сборник студенческих научных работ. Саратов : Издательство Саратовского университета, 1990. С. 3–5.

[3] Moon J. W. Topics on tournaments. New York: Holt, Rinehart and Winston, 1968. 148 p.

[4] Some remarks on simple tournaments / P. Erdös, E. Fried, A. Hajnal, E. C. Milner // Algebra universalis. 1972. Vol. 2, no. 2. P. 238–245.

[5] Мун Дж. В. Вложение турниров в простые турниры // Теория графов. Покрытия, укладки, турниры. М. : Мир, 1974. С. 169–174.

Моделирование артериального барорефлекса линейными гибридными автоматами

Ширинян Маринэ Эдгаровна[1], Гасанов Эльяр Эльдарович[2]

1 Научно-технологический центр органической и фармацевтической химии Национальной академии наук Республики Армения; mshirinian@physiol.sci.am

2 Московский государственный университет имени М. В. Ломоносова; el_gasanov@mail.ru

Основной причиной развития сердечно-сосудистых патологий является нарушение механизмов регуляции артериального давления [1], среди которых

особое значение придается основному механизму нейрогенной регуляции артериального давления — артериальному барорецепторному рефлексу (или барорефлексу) [2, 3]. Дисфункция барорефлекса считается одним из основных факторов риска развития гипертонии и рассматривается в качестве предиктора в математических моделях прогнозирования и диагностики гипертонии и других сердечно-сосудистых заболеваний [4, 5]. На сегодняшний день с применением разных принципов и подходов создано множество математических моделей сердечно-сосудистой системы и механизмов ее контроля, в частности, барорецепторной регуляции артериального давления [6, 7, 8]. В работе [9] была описана автоматная модель барорефлекса с дискретными параметрами.

В нашей работе предложена математическая модель барорефлекса на основе линейных гибридных автоматов, показывающая сходимость к точке равновесия системы барорефлекса.

Пусть $f_1(p)$ — убывающая функция, отображающая зависимость изменения активности симпатической нервной системы (СНС) от изменения величины артериального давления (АД), $f_2(c)$ — возрастающая функция, отображающая зависимость величины АД от уровня активности СНС. В системе барорефлекса равновесие достигается в точке (c_0, p_0) пересечения функциональных кривых $f_1(p)$ и $f_2(c)$, координаты которой соответствуют уровню нормального АД и соответствующей активности СНС. При отклонении АД от своего нормального уровня p_0 активность СНС также отклонится от своего нормального уровня c_0 в соответствии с функцией $f_1(p)$, что повлечет за собой изменение значения АД согласно функции $f_2(c)$ и т. д. Взаимное влияние на значение друг друга прекратится только в том случае, когда текущее значение p станет равным p_0 и текущее значение c станет равным c_0.

Назовем *линейным гибридным автоматом* набор $H^\delta = (A, B, Z, \varphi, \psi, \delta, z_0)$ с выделенными начальным состоянием $z_0 \in Z$ и параметром $\delta \in \mathbb{R}$, где: A, B, Z — множества вещественных чисел $\mathbb{R}$, являющиеся соответственно входным алфавитом, выходным алфавитом и алфавитом состояний автомата H^δ; φ — функция перехода автомата H^δ, определенная на множестве $A \times Z$ и принимающая значение из множества Z, такая что $\varphi(a, z) = \varphi\, sign\,(a - z) + z$; ψ — функция выхода автомата H^δ, определенная на множестве $A \times Z$ и принимающая значение из множества B, такая что $\psi(a, z) = z$; δ — шаг автомата H^δ, определяющий величину изменения параметров автомата H^δ за один такт.

В данной работе, в отличие от классического описания гибридных систем [10], представлено упрощенное описание линейного гибридного автомата с одним входным и выходным множествами вещественных чисел. Понятие «времени» автомата H^δ носит дискретный характер, и отсчеты времени t принадлежат множеству натуральных чисел $\mathbb{N}$. Автомат H^δ при подаче на его вход последовательности $a(1), a(2), \ldots$ выдает на выходе последовательность

$b(1), b(2), \ldots$ в соответствии со следующими каноническими уравнениями:

$$\begin{cases} z(1) & = & z_0, \\ z(t+1) & = & \varphi(\, a(t),\, z(t)), \\ b(t) & = & \psi(a(t), z(t)). \end{cases}$$

Назовем *барорецепторным комплексом* (БР-комплексом) набор

$$K =< f_1(p), f_2(c), H_P^\delta, H_C^\delta, c_0, p_0, c_1, p_1 >,$$

где: $f_1(p)$ и $f_2(c)$ — функции зависимости реакции СНС в ответ на изменение величины АД и значений АД в ответ на изменение СНС соответственно; H_P^δ и H_C^δ — два экземпляра одного и того же автомата H^δ, формирующие состояние БР-комплекса в каждый момент времени t; c_0 и p_0 — равновесное состояние автоматов H_P^δ и H_C^δ, где $c_0 = f_1(p_0)$ и $p_0 = f_2(c_0)$; c_1 и p_1 — значения входных сигналов автоматов H_P^δ и H_C^δ соответственно в момент времени $t = 1$ (координаты отклонения БР-комплекса). Состояние БР-комплекса в момент времени t определяется парой $(c(t), p(t))$, где $c(1) = c_1$, $p(1) = p_1$ при $t = 1$ и $c(t) = \psi\,(\,z(t-1), f_1(p(t-1)))$, $p(t) = \psi\,(\,z(t-1), f_2(c(t-1)))$ при $t \geqslant 2$. В каждый момент времени t состояние БР-комплекса $(c(t), p(t))$ изменяется за счет изменения внутреннего состояния автоматов H_P^δ и H_C^δ и стремится к своему равновесному состоянию (c_0, p_0), что на плоскости $[\,C, P\,]$ отражается в перемещении точки $(c(t), p(t))$ в точку с координатами (c_0, p_0). В данной работе в качестве расстояния рассматривается расстояние по Манхэттену, т. е. расстояние между точками (c_1, p_1) и (c_2, p_2) равно $|c_1 - c_2| + |p_1 - p_2|$.

Теорема 1. *Пусть $f_1(p)$ — строго убывающая функция, $f_2(c)$ — строго возрастающая функция, тогда для любого вещественного числа $\varepsilon > 0$ существует вещественное число $\delta > 0$, такое что для любого отклоненного состояния (c_1, p_1) от состояния равновесия (c_0, p_0) существует момент времени T, такой что для любого момента времени $t \geqslant T$ состояние $(c(t), p(t))$ барорецепторного комплекса $< f_1(p), f_2(c), H_P^\delta, H_C^\delta, c_0, p_0, c_1, p_1, \delta >$ будет находится в ε-окрестности точки (c_0, p_0), т. е. будет выполнено неравенство $|c(t) - c_0| + |p(t) - p_0| \leqslant \varepsilon$.*

Биологической интерпретацией теоремы о сходимости БР-комплекса к точке равновесия является математически доказанное подтверждение теоретических рассуждений и экспериментальных данных о том, что система барорецепторного рефлекса, если она не подвержена никаким другим влияниям, при отклонении возвращается в состояние равновесия [3].

Вышесказанное позволяет заключить об адекватности предлагаемой модели БР-комплекса, построенной линейными гибридными автоматами, с системой барорецепторного рефлекса.

Список литературы

[1] 2018 ESC/ESH Guidelines for the management of arterial hypertension: The Task Force for the management of arterial hypertension of the European Society of Cardiology (ESC) and the European Society of Hypertension (ESH) / B. Williams, G. Mancia, W. Spiering, E. A. Rosei, M. Azizi, M. Burnier, D. L. Clement, A. Coca, G. de Simone, A. Dominiszak, T. Kahan, F. Mahfoud, J. Redon, L. Ruilope, A. Zanchetti, M. Kerins, S. E. Kjeldsen, R. Kreutz, S. Laurent, G. Y. H. Lip, R. McManus, K. Narkiewicz, F. Ruschitzka, R. E. Schmieder, E. Shlyakhto, C. Tsioufis, V. Aboyans, I. Desormais // European Heart Journal. 2018. Vol. 39, no. 33. P. 3021–3104.

[2] Parati G., Ochoa J. E. Prognostic value of baroreflex sensitivity in heart failure. A 2018 reappraisal // European Journal of Heart Failure. 2019. Vol. 21, no. 1. P. 59–62.

[3] Analytic and integrative framework for understanding human sympathetic arterial baroreflex function: equilibrium diagram of arterial pressure and plasma norepinephrine level / F. Yamasaki, T. Sato, K. Sato, A. Diedrich // Frontiers in Neuroscience. 2021. V 15. Article 707345.

[4] NCD Risk Factor Collaboration (NCD-RisC). Worldwide trends in blood pressure from 1975 to 2015: a pooled analysis of 1479 population-based measurement studies with 19.1 million participants // The Lancet. 2017. Vol. 389, no. 10064. P. 37–55.

[5] Molkov Y. Baroreflex models // Encyclopedia of computational neuroscience. New York, NY, USA : Springer, 2022. P. 345–353.

[6] Рубцова Е. Н. Персонализированная математическая модель сердечно-сосудистой системы с механизмом барорефлекса // Известия высших учебных заведений. Электроника. 2022. Т. 27, № 1. С. 89–105.

[7] Модель сердечно-сосудистой системы с регуляцией на основе нейронной сети / С. В. Фролов, А. А. Коробов, Д. Ш. Газизова, А. Ю. Потлов // Модели, системы, сети в экономике, технике, природе и обществе. 2021. № 2. С. 79–84.

[8] Modelling of long-term and short-term mechanisms of arterial pressure control in the cardiovascular system: an object-oriented approach / J. F. de Canete, J. Luque, J. Barbancho, V. Munoz // Computers in Biology and Medicine. 2014. Vol. 47. P. 104–112.

[9] Ширинян М. Э. Моделирование регуляции равновесия в системе артериальной барорецепции на основе гибридного автомата // Доклады Национальной академии наук Армении. 2013. Т. 113, № 1. С. 99-108.

[10] Сениченков Ю. Б. Численное моделирование гибридных систем. СПб. : Издательство Политехнического университета. 2004.

О генерации унографов с заданным числом вершин

Шкатов Владимир Михайлович, Абросимов Михаил Борисович
Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского; vmshkatov@gmail.com, mic@rambler.ru

В работе рассматривается задача быстрого перечисления унографов без необходимости проверок на изоморфизм и унографичность. Здесь и далее используются основные определения по теории графов, данные в [1]. Все рассматриваемые графы неориентированные.

Определение. *Вектором степеней графа называется невозрастающая последовательность степеней его вершин.*

Определение. *Будем называть граф унографом, если не существует никакого другого неизоморфного графа с таким же вектором степеней.*

Существует эффективный алгоритм ответа на вопрос, является ли заданный граф унографом (см. статью [2]). Напомним, что *кликой* графа называется любой полный подграф, содержащийся в данном графе, а *независимым множеством* графа называется любое множество попарно несмежных вершин графа. Для построения алгоритма перечисления существенно используются результаты Тышкевич из [2, 3] о распознавании унографов и их структуре, поэтому приведём их краткий обзор.

Определение. *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.*

Определение. *Расщепляемой тройкой называется тройка (G, A, B), где $G = (V, \alpha)$ — расщепляемый граф, A — клика, B — независимое множество, $A \cup B = V$ и $A \cap B = \varnothing$. 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 $\phi(A_1) = A_2$, $\phi(B_1) = B_2$.*

Определение. *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 1. *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.*

Теорема 2. *Граф F является унографом тогда и только тогда, когда все графы в его каноническом разложении являются унографами.*

К этой теореме в работе [3] также прилагается описание всех неразложимых унографов в виде нескольких параметризованных классов. Структура и вектор степеней этих графов в зависимости от параметров известен.

Алгоритм генерации унографов

Теоремы из предыдущего раздела позволяют разработать концептуально простой алгоритм генерации всех унографических векторов степеней для графов с заданным числом вершин, не требующий ни перебора всех графов, ни проверок на изоморфизм. Согласно теореме 1, граф определяется своим каноническим разложением с точностью до изоморфизма. Согласно теореме 2, унографами являются только те графы, у которых все члены их канонического разложения — унографы. Параметризованные классы неразложимых унографов приводятся, например, в [2, 3]. Таким образом, если для заданного числа вершин n сгенерировать список всех неразложимых унографов с числом вершин не более n, а потом перебирать их всевозможные композиции, можно найти среди них все унографы с числом вершин равным n.

Алгоритм перечисления унографических векторов степеней

Вход: число n.

Выход: унографические векторы степеней для графов с n вершинами.

Шаг 1. Создать списки *basicNonSplits*, *basicSplits*.

Шаг 2. Добавить в *basicNonSplits* пустой граф. Добавить в *basicNonSplits* все неразложимые нерасщепляемые графы с числом вершин не больше n.

Шаг 3. Добавить в *basicSplits* все неразложимые расщепляемые графы с числом вершин не больше n.

Шаг 4. Для каждого графа NS из *basicNonSplits* делать шаги 5–6. По окончании перейти к шагу 7.

Шаг 5. Если $|NS| = n$, то **выдать** NS и вернуться к шагу 4 для следующего графа.

Шаг 6. Для всех графов S из *basicSplits* запускать процедуру $recEnum(S \circ NS)$, если $|S \circ NS| = |S| + |NS| \leqslant n$

Шаг 7. Все унографы с числом вершин n перечислены, конец.

Процедура $recEnum(G)$

Вход: граф G.

Шаг 1. Если $|G| = n$, то **выдать** G и выйти из процедуры на уровень выше.

Шаг 2. Для всех графов S из *basicSplits* запускать процедуру $recEnum(S \circ G)$, если $|S \circ G| = |S| + |G| \leqslant n$.

Отметим два факта об этом алгоритме. Во-первых, содержательная часть данного алгоритма (шаги 4–7 и процедура $recEnum(G)$) концептуально похожа на перебор строк длины не более n некоторого алфавита и независимую проверку некоторых условий на каждой из строк, поэтому эта часть алгоритма может исполняться параллельно. Во-вторых, для унографов вектор степеней задаёт граф однозначно, поэтому операции алгоритма могут производится над векторами степеней. Это значительно экономит вычислительные ресурсы для расчёта композиции.

Предложенный алгоритм был реализован авторами на языке Go в однопоточном режиме и позволил найти количество унографов с 11–21 вершинами. До этого в OEIS A122423 [4] была доступна информация только об унографах с не более чем 10 вершинами. В таблице приведены найденные результаты и время работы программы на процессоре AMD® Ryzen 5 2600 (частота 3,4 ГГц).

Число вершин	Число унографов	Время работы генератора
11	5304	меньше секунды
12	12555	меньше секунды
13	29754	меньше секунды
14	70662	1 с.
15	167834	2 с.
16	398627	4,7 с.
17	946402	11,6 с.
18	2246294	31,7 с.
19	5330340	1 м. 19 с.
20	12647767	3 м. 3 с.
21	30010020	7 м. 26 с.

Список литературы

[1] Богомолов А. М., Салий В. Н. Алгебраические основы теории дискретных систем. М. : Наука, Физматлит, 1997. 368 с.

[2] Тышкевич Р. И., Суздаль С. В. Декомпозиция графов // Выбраныя навуковыя працы Беларускага дзяржаўнага універсітэта: У сямі тамах. Мінск : БДУ, 2001. Т. 6. С. 482–500.

[3] Tyshkevich R. Decomposition of graphical sequences and unigraphs // Discrete Mathematics. 2000. Vol. 220. P. 201–238.

[4] Sloane N. J. A., The OEIS Community. The on-line encyclopedia of integer sequences. The OEIS Foundation Inc. Retrieved 19.04.2025 from http://oeis.org.

Оценки динамической и статической активности схем контактного типа, реализующих функции, встречающиеся в приложениях

Шуплецов Михаил Сергеевич
Московский государственный университет имени М. В. Ломоносова; shupletsov@cs.msu.ru

Введение

При проектировании интегральных схем для современных портативных вычислительных устройств важной проблемой стала оценка и оптимизация энергопотребления схемы. Традиционно рассматривают два типа энергопотребления: статическое, связанное с рассеянием тепла и поддержанием заданного высокого потенциала в узлах схемы, подключенных к источнику питания, и динамическое, возникающее при изменении потенциалов в узлах схемы.

Первые подходы к анализу статического энергопотребления для модели схем из функциональных элементов (СФЭ) были предложены в работе [1], а основные теоретические результаты в этом направлении были получены О. М. Касим-Заде в работах [2, 3]. В указанных работах был введен и исследован функционал мощности СФЭ, характеризующий статическое энергопотребление, для которого был установлен порядок роста соответствующей функции Шеннона в произвольном конечном полном базисе. Оказалось, в частности, что существуют базисы как с линейным, так и с экспоненциальным поведением указанной функции Шеннона. Кроме того, была показана возможность построения для «типичной» функции алгебры логики (ФАЛ) такой реализующей ее СФЭ, сложность которой асимптотически оптимальна, а мощность оптимальна по порядку роста.

В работе [4] был введен функционал динамической (переключательной) активности для СФЭ. При этом было доказано, что в произвольном базисе порядок роста функции Шеннона для динамической активности СФЭ не превосходит некоторую линейную функцию, и были предложены методы синтеза, позволяющие для произвольной ФАЛ построить СФЭ в стандартном базисе, сложность которой асимптотически оптимальна, а мощность и динамическая активность оптимальны по порядку роста. Аналогичный функционал динамической активности был введен для модели ориентированных контакт-

ных схем в работе [5]. В последние годы появился ряд работ [6–8], в которых изучается связь рассматриваемых функционалов активности схем с другими сложностными характеристиками схем и свойствами ФАЛ, реализуемых этими схемами.

Основные определения

Пусть $\Sigma = \Sigma(x_1, \ldots, x_n; a', a'')$ — произвольная $(1, k)$-КС от булевых переменных (БП) $\widetilde{x} = (x_1, \ldots, x_n)$, имеющая вход a' и выходы $a'' = (a''_1, \ldots, a''_k)$, на которых реализуется оператор $F = (f_1, \ldots, f_k)$, состоящий из ФАЛ f_i проводимости от входа a' до выхода a''_i. Пусть $V(\Sigma)$ — множество вершин ОКС Σ, отличных от a' и a''. Тогда для каждой вершины v, $v \in V$, КС Σ определим функцию достижимости $g_v(\widetilde{x})$ от входа a' до вершины v, которая равна 1 на наборе $\widetilde{\alpha} = (\alpha_1, \ldots, \alpha_n) \in B^n$, где B^n — единичный n-мерный куб, тогда и только тогда, когда в Σ существует путь из вершины a' в вершину v, состоящий из проводящих контактов. *Статической активностью КС Σ на наборе* $\widetilde{\alpha}$ будем называть следующую величину:

$$E(\Sigma, \widetilde{\alpha}) = \sum_{v \in V(\Sigma)} g_v(\widetilde{\alpha}).$$

Динамической активностью КС Σ на паре наборов $\widetilde{\alpha}$ *и* $\widetilde{\beta}$ будем называть следующую величину:

$$S(\Sigma, \widetilde{\alpha}, \widetilde{\beta}) = \sum_{v \in V(\Sigma)} g_v(\widetilde{\alpha}) \oplus g_v(\widetilde{\beta}).$$

Статической активностью (*динамической активностью*) $E(\Sigma)$ $(S(\Sigma))$ КС Σ назовем максимальное значение величины $E(\Sigma, \widetilde{\alpha})$ $(S(\Sigma, \widetilde{\alpha}, \widetilde{\beta}))$, взятое по всем наборам $\widetilde{\alpha}$ из B^n (парам наборов $(\widetilde{\alpha}, \widetilde{\beta})$ из $B^n \times B^n$ соответственно).

Наконец, для произвольного булева оператора $F = (f_1, \ldots, f_k)$ *статической активностью* $E^{\mathrm{KC}}(F)$ (*динамической активностью* $S^{\mathrm{KC}}(F)$) этого оператора F в классе КС будем называть минимальную статическую (динамическую) активность $(1, k)$-КС, реализующих оператор F.

Пусть $\nu(\sigma) = \sum_{i=1}^{n} \sigma_i 2^{n-i}$ — номер набора $\sigma = (\sigma_1, \ldots, \sigma_n)$ при лексикографическом упорядочивании наборов куба B^n и $K_\sigma(\widetilde{x})$ — элементарная конъюнкция $x_1^{\sigma_1} x_2^{\sigma_2} \cdot \ldots \cdot x_n^{\sigma_n}$. Тогда $K_i = K_{\sigma^*}(\widetilde{x})$, где $i = \nu(\sigma^*)$. *Дешифратором порядка* n от БП $x_1, \ldots, x_n$ называется булевский оператор $Q_n = \{K_1, \ldots, K_{2^n}\}$.

Пусть $\widetilde{x} = (x_1, \ldots, x_n)$ и $\widetilde{y} = (y_1, \ldots, y_{2^n})$. *Мультплексорной функцией порядка* n называется ФАЛ $\mu_n = \mu_n(\widetilde{x}, \widetilde{y})$, зависящая от n адресных БП $\widetilde{x}$ и 2^n информационных БП $\widetilde{y}$, для которой верно следующее представление:

$$\mu_n(\widetilde{x}, \widetilde{y}) = \bigvee_{\sigma \in B^n} K_\sigma(\widetilde{x}) y_{\nu(\sigma)},$$

где $\sigma = (\sigma_1, \ldots, \sigma_n) \in B^n$ — произвольный набор значений переменных x.

Основные результаты

Теорема 1. *Существует неотрицательная и стремящееся к нулю последовательность действительных чисел* $\epsilon(1)$, $\epsilon(2), \ldots$ *такая, что для любого* n, $n = 1, 2, \ldots$, *дешифратор* Q_n *может быть реализован некоторой* $(1, 2^n)$-*КС* Σ_{Q_n}, *удовлетворяющей неравенствам*

$L^{KC}(\Sigma_{Q_n}) \leqslant 2^n(1+\epsilon(n))$, $E^{KC}(\Sigma_{Q_n}) \leqslant 4n(1+\epsilon(n))$, $S^{KC}(\Sigma_{Q_n}) \leqslant 8n(1+\epsilon(n))$.

Теорема 2. *Существует неотрицательная и стремящееся к нулю последовательность действительных чисел* $\epsilon(1)$, $\epsilon(2), \ldots$ *такая, что для любого* n, $n = 1, 2, \ldots$, *мультиплексор* μ_n *может быть реализована некоторой* $(1, 1)$-*КС* Σ_{μ_n}, *удовлетворяющей неравенствам*

$L^{KC}(\Sigma_{\mu_n}) \leqslant 2^{n+1}(1+\epsilon(n))$, $E^{KC}(\Sigma_{\mu_n}) \leqslant 5n(1+\epsilon(n))$, $S^{KC}(\Sigma_{\mu_n}) \leqslant 10n(1+\epsilon(n))$.

Работа выполнена при финансовой поддержке Минобрнауки в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2022-284.

Список литературы

[1] Вайнцвайг М. Н. О мощности схем из функциональных элементов // Доклады Академии наук СССР. 1961. Т. 139, № 2. С. 320–323.

[2] Касим-Заде О. М. Об одновременной минимизации сложности и мощности схем из функциональных элементов // Проблемы кибернетики. М. : Наука, 1978. Вып. 33. С. 215–220.

[3] Касим-Заде О. М. Об одной мере сложности схем из функциональных элементов // Проблемы кибернетики. М. : Наука, 1981. Вып. 38. С. 117–179.

[4] Ложкин С. А., Шуплецов М. С. О динамической активности схем из функциональных элементов и построении асимптотически оптимальных по сложности схем с линейной динамической активностью // Ученые записки Казанского университета. Серия физико-математические науки. 2014. Т. 156, кн. 3. С. 84–97.

[5] Шуплецов М. С. Оценки функции Шеннона для динамической активности ориентированных контактных схем // Проблемы теоретической кибернетики: XVIII Международная конференция (Пенза, 19–23 июня 2017 г.) : Материалы. М. : МАКС Пресс, 2017. С. 263-–266.

[6] Dinesh K., Otiv S., Sarma J. New bounds for energy complexity of Boolean functions // Theoretical Computer Science. 2020. Vol. 845. P. 59–75.

[7] Mestetskiy M. A., Shupletsov M. S. Relations between energy complexity measures of Boolean networks and positive sensitivity of Boolean functions // Discrete Mathematics and Applications. 2024. Vol. 34, no. 4. P. 211–219.

[8] Sun Xiaoming, Sun Yuan, Wu Kewen, Xia Zhiyu. On the relationship between energy complexity and other Boolean function measures // Computing and Combinatorics. COCOON 2019. Lecture Notes in Computer Science. 2019. Vol. 11653. P. 516–528.

Морфические слова с хорошо распределенными вхождениями подслов

Щавелев Владимир Эдуардович,
Пузынина Светлана Александровна
Санкт-Петербургский государственный университет; vovashavelev11@mail.ru, s.puzynina@gmail.com

В данной работе исследуется свойство бесконечных слов, называемое хорошо распределенными вхождениями факторов. Это свойство абелева типа, то есть свойство слов, в котором учитывается лишь число вхождений каждой буквы без учета их порядка. Изучение абелевых свойств слов — это одно из основных направлений современной комбинаторики слов. Помимо теоретического интереса в комбинаторике слов, свойство хорошо распределенных вхождений факторов применимо для генерирования псевдослучайных последовательностей.

Зафиксируем некоторое конечное множество Σ, будем называть его *алфавитом*, а его элементы — *буквами*. Всевозможные последовательности из букв будем называть *словами*, а множество всех конечных слов, включая пустое слово, обозначать как Σ^*. Подслово конечного или бесконечного слова из подряд идущих букв называется *фактором*, а если фактор начинается с начала слова, то будем его называть *префиксом*. Для конечного слова u и буквы a будем обозначать количество букв a, входящих в u, как $|u|_a$, а длину u как $|u|$. Фактор с позиции i до позиции j обозначается как $w[i, j + 1)$.

В наше время множество алгоритмов различной сложности основаны на случайных алгоритмах: ни одна симуляция не обходится без случайных последовательностей, некоторые даже простые алгоритмы, такие как Quick sort, генерируют случайные числа для оптимальной работы. И естественным образом появляется необходимость в построении случайных последовательностей, однако сделать это непросто, ведь любой алгоритм в каком-то смысле можно просчитать наперед. Одним из самых простых примеров генераторов случайных последовательностей являются линейные конгруэнтные генераторы — последовательности $Z_{n+1} = aZ_n + c \mod m$, для некоторых $a, c, m \in \mathbb{N}$.

Однако они имеют несвойственный случайным последовательностям дефект, называющийся решетчатой структурой: если рассмотреть множество

из всех n подряд идущих чисел из генератора как подмножество $\mathbb{Z}^n$, то они будут покрываться семейством параллельных плоскостей, не покрывающих все пространство.

В статье [1], чтобы избавиться от этого дефекта, предлагается генерировать некоторое бесконечное слово w над алфавитом Σ, а также $|\Sigma|$ линейных конгруэнтных генераторов $Z^{(i)}$ и далее рассматривать последовательность, в которой мы в w заменяем все вхождения i-ой буквы на числа в генераторе $Z^{(i)}$. Пусть $w = w_0 w_1 \cdots$ — бесконечное слово над алфавитом Σ и $Z^{(i)}$ — линейные конгруэнтные генераторы. Рассмотрим f — функцию такую, что $f(i) = |\{j < i | w_j = w_i\}|$, то есть считающую, сколько таких же букв, как и w_i, уже встретилось в w. Тогда новым генератором будет $Z(w)_n = Z^{(w_n)}_{f(n)}$. Выбором слова w можно получить последовательность, у которой не будет дефекта решетчатой структуры. В статье [2] приводится достаточное условие для отсутствия решетчатой структуры, называемое WELLDOC, или свойство хорошо распределенных вхождений факторов (well distributed occurrences).

Определение. *Будем говорить, что для слова w выполняется свойство* WELLDOC *над алфавитом* $\{0, 1, \ldots, n-1\}$*, если для любого фактора u и для любого натурального модуля m выполняется следующее. Пусть $i_0, i_1, \ldots$ — это позиции, с которых начинается каждое вхождение u в w. Тогда* $\{(|Pref_{i_j} w|_0, \ldots |Pref_{i_j} w|_{n-1}) \mod m \mid i_j \in \mathbb{N}\}$.

Каждый из факторов $w[i_j, i_{j+1})$ называется возвратом *u в w. Вектор* $(|v|_0, \ldots |v|_{n-1})$ *называется* вектором Парика *слова v.*

Другими словами, для бесконечного слова w выполняется свойство WELLDOC, если для любого фактора u и для любого натурального модуля m найдется префикс p перед u такой, что его вектор Парика совпадает по модулю m с любым наперед заданным вектором остатков.

В данной работе рассматривается свойство WELLDOC для слов, получающихся как предел применения некоторого морфизма последовательно к некоторой букве и последующим ее образам; такие слова называются морфическими. Преимущество использования таких слов для построения генераторов состоит, в частности, в том, что такие слова можно очень быстро генерировать.

Морфизмом ϕ называется отображение из Σ^* в Σ^* такое, что $\phi(uv) = \phi(u)\phi(v)$ для всех слов $u, v \in \Sigma^*$. Все рассматриваемые в этой работе морфизмы будут *нестирающими*: образ любого непустого слова не является пустым. Рассмотрим *неподвижные точки* морфизма ϕ, то есть бесконечные слова w такие, что $x = \phi(x)$. Говорят, что морфизм ϕ — *продолжаемый* буквой a, если образ $\phi(a)$ начинается с буквы a, другими словами, если $\phi(a) = as$ для некоторого непустого слова s. Тогда для каждой буквы a, на которой морфизм ϕ продолжаемый, у него есть неподвижная точка, так как $\phi^n(a)$

является префиксом $\phi^{n+1}(a)$ для всех $n \in \mathbb{N}$, и предел последовательности $(\phi^n(a))_{n\geqslant 0}$ будет являться бесконечным словом:

$$w = \lim_{n\to\infty} \phi^n(a).$$

Слова, порожденные таким образом, называются *морфическими*.

Определение. *Морфизм ϕ называется примитивным, если $\phi^k(x)$ для любой буквы x и некоторого k содержит каждую букву алфавита, и непримитивным иначе.*

Определение. *Матрицей морфизма ϕ называется следующая матрица порядка $\sigma = |\Sigma|$:*

$$A_\phi = \begin{pmatrix} |\phi(0)|_0 & |\phi(1)|_0 & \dots & |\phi(\sigma-1)|_0 \\ |\phi(0)|_1 & |\phi(1)|_1 & \dots & |\phi(\sigma-1)|_1 \\ \vdots & \vdots & \ddots & \vdots \\ |\phi(0)|_{\sigma-1} & |\phi(1)|_{\sigma-1} & \dots & |\phi(\sigma-1)|_{\sigma-1} \end{pmatrix}.$$

Основным результатом работы является критерий выполнения свойства WELLDOC для морфических слов в терминах матрицы соответствующего морфизма.

Теорема 1. *Пусть w — бесконечное бинарное слово, порожденное примитивным морфизмом ϕ. Тогда для w выполнено WELLDOC тогда и только тогда, когда* $\det A_\phi = \pm 1$.

Замечание. *Здесь и далее мы не считаем слова 0^∞ и 1^∞ бинарными. Для этих слов WELLDOC выполняется, и при этом их можно задать морфизмом $\phi(0) = 00$, $\phi(1) = 11$ с определителем* 4.

Для небинарных морфических слов, порожденных примитивными морфизмами, требуется дополнительное структурное условие на возвраты к первой букве.

Теорема 2. *Пусть w — бесконечное слово, порожденное примитивным морфизмом ϕ. Тогда для w выполнено WELLDOC тогда и только тогда, когда* $\det A_\phi = \pm 1$ *и вектора Парика всех возвратов к первой букве w порождают пространство $\mathbb{Z}^{|\Sigma|}$ как группа по сложению.*

Отметим, что дополнительное условие на векторы Парика префиксов несложно проверить алгоритмически для слов, порожденных примитивными морфизмами. С помощью этих теорем можно доказать, что для морфических слов Штурма и эпиштурмовых слов выполняется свойство WELLDOC.

Список литературы

[1] Guimond L.-S., Patera J., Patera J. Statistical properties and implementation of aperiodic pseudorandom number generators // Applied Numerical Mathematics. 2003. Vo. 46, no. 3–4. P. 295–318.

[2] Aperiodic pseudorandom number generators based on infinite words / L. Balková, M. Bucci, A. De Luca, J. Hladký, S. Puzynina // Theoretical Computer Science. 2016. Vol. 647. P. 85–100.

Применение тернарных L-квазигрупп для преобразования слов

Щучкин Николай Алексеевич, Веселова Александра Андреевна
Волгоградский государственный социально-педагогический университет;
nikolaj_shchuchkin@mail.ru, sachka-korzhova@mail.ru

В последнее время активно разрабатываются криптографические алгоритмы, основанные на неассоциативных алгебраических структурах [1]. Одной из наиболее подходящих алгебраических структур для таких целей является конечная квазигруппа. Известно широкое применение квазигрупп в криптографии (см., например, [2]). Обобщением квазигрупп являются левые квазигруппы, тернарные квазигруппы и L-квазигруппы. В работе [3] были рассмотрены применения тернарных квазигрупп для преобразования слов в заданном алфавите. Аналогичные преобразования с помощью тернарных L-квазигрупп будут приведены ниже.

Исследовательской проблемой является идентификация подходящих квазигрупп для криптографических целей. В работе [4] отмечалось, что с алгебраической точки зрения полиномиально полные квазигруппы подходят для криптографии. Наряду с полиномиально полными квазигруппами в криптографии можно использовать и такого же вида тернарные квазигруппы и L-квазигруппы. В работе [3] были исследованы алгебраические свойства тернарных квазигрупп, такие как полиномиальная полнота, отсутствие нетривиальных конгруэнций. Аналогичные исследования проведем ниже для тернарных L-квазигрупп. Эти свойства могут сыграть важную роль при анализе и проектировании криптографических схем на основе тернарных L-квазигрупп.

Предварительные сведения

Тернарный группоид $\langle Q, f\rangle$, в котором для любых элементов a, b, c из Q уравнение

$$f(x, b, c) = a \tag{1}$$

разрешимо однозначно, будем называть тернарной L-квазигруппой. На множестве Q имеется еще одна тернарная операция u, заданная по правилу

$$u(a,b,c)=d \Leftrightarrow f(d,b,c)=a.$$

Операции u и f связаны тождествами

$$u(f(x,y,z),y,z)=x=f(u(x,y,z),y,z). \tag{2}$$

Таким образом, на тернарную L-квазигруппу $\langle Q,f\rangle$ можно смотреть как на универсальную алгебру $\langle Q,f,u\rangle$ с набором тождеств (2).

Преобразования слов

Пусть множество Q конечно и $Q=\{1,2,\ldots,m\}$. Тернарной L-квазигруппе $\langle Q,f\rangle$ соответствует 3-мерная матрица $B=(b_{ijk}|i,j,k=1,2,\ldots,m)$ m-го порядка, где $b_{ijk}=f(i,j,k)$, причем, в силу однозначной разрешимости уравнения (1), в строках направления 1 стоят разные элементы из Q. Верно и обратное, любая 3-мерная матрица m-го порядка $B=(b_{ijk}|i,j,k=1,2,\ldots,m)$, у которой в строках направления 1 стоят разные элементы из Q, определяет тернарную L-квазигруппу $\langle Q,f\rangle$, где $f(i,j,k)=b_{ijk}$. Таким образом, между тернарными L-квазигруппами и 3-мерными матрицами указанного вида имеется взаимно однозначное соответствие.

Мы оцениваем число $L(m;3)$ тернарных L-квазигрупп порядка m:

$$L(m;3)=m!^{m^2}.$$

Эта оценка указывает на большое количество тернарных L-квазигрупп, построенных на конечном множестве. А значит, имеются перспективы использования тернарных L-квазигрупп в криптографии.

Каждая 3-мерная матрица B, построенная для тернарной L-квазигруппы $\langle Q,f\rangle$, где $Q=\{1,2,\ldots,m\}$, определяет набор из m латинских квадратов на множестве Q с умножением $i\circ_k j=f(i,j,k)$ $(k=1,2,\ldots,m)$. Таким образом, на 3-мерную матрицу B можно смотреть как на упорядоченный набор таблиц умножения левых квазигрупп в количестве, равном числу элементов множества Q.

Для преобразования слов в заданном алфавите используют квазигруппы [5]. Мы обобщаем преобразования слов из этой работы на тернарный случай, т. е. в работе [3] было указано преобразование слов с помощью тернарных квазигрупп, а здесь будем преобразовывать слова с помощью тернарных L-квазигрупп. Пусть $\langle Q,f\rangle$ — тернарная L-квазигруппа, где $Q=\{1,\ldots,m\}$. Множество всех слов в алфавите Q обозначим $Q^+=\{x_1\ldots x_s|x_i\in Q,s\geqslant 1\}$. Для пары элементов a,b из Q на множестве Q^+ определим отображение

$$F_{a,b}(x_1x_2\ldots x_s)=y_1y_2\ldots y_s=$$

$$= \begin{cases} y_1 = f(x_1, a, b), \\ y_2 = f(x_2, y_1, a), \\ y_{i+1} = f(x_{i+1}, y_i, y_{i-1}), i = 2, 3, \ldots, s-1. \end{cases} \quad (3)$$

Для той же пары элементов a, b из Q на множестве Q^+ строим еще одно отображение

$$G_{a,b}(y_1 y_2 \ldots y_s) = x_1 x_2 \ldots x_s =$$

$$= \begin{cases} x_1 = u(y_1, a, b), \\ x_2 = u(y_2, y_1, a), \\ x_{i+1} = u(y_{i+1}, y_i, y_{i-1}), i = 2, 3, \ldots, s-1. \end{cases} \quad (4)$$

Теорема 1. *Отображение $F_{a,b}$, построенное по правилу* (3)*, является биективным. Отображение $G_{a,b}$, построенное по правилу* (4)*, является обратным для отображения $F_{a,b}$.*

Тернарные L-квазигруппы с левым нейтральным элементом

Элемент e тернарной L-квазигруппы $\langle Q, f\rangle$ назовем левым нейтральным элементом, если верно равенство $f(e, a, a) = a$ для любого элемента $a \in Q$.

Теорема 2. *В тернарной L-квазигруппе с левым нейтральным элементом e верны тождества $u(x, x, x) = u(y, y, y) = e$. Верно и обратное, если в тернарной L-квазигруппе верно тождество $u(x, x, x) = u(y, y, y)$, то там есть левый нейтральный элемент $e = u(x, x, x)$.*

Теорема 3. *В тернарной L-квазигруппе с левым нейтральным элементом имеется терм Мальцева $m(x, y, z) = f(u(x, y, y), z, z)$.*

Теорема 4. ([6]). *Пусть A — конечная алгебра, содержащая по меньшей мере два элемента. Тогда следующие условия эквивалентны:*

i) A полиномиально полна;

ii) существует терм Мальцева на A и алгебра A является простой и неаффинной.

Следствие 1. *Конечная тернарная L-квазигруппа с левым нейтральным элементом, содержащая по меньшей мере два элемента, полиномиально полна тогда и только тогда, когда она является простой и неаффинной.*

Список литературы

[1] Марков В. Т., Михалёв А. В., Нечаев А. А. Неассоциативные алгебраические структуры в криптографии и кодировании // Фундаментальная и прикладная математика. 2016. Т. 21, № 4. С. 99–124.

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

[3] Щучкин Н. А. Применение тернарных квазигрупп к преобразованию слов // Дискретная математика. 2024. Т. 36, вып. 2. С. 132—143.

[4] Артамонов В. А. Полиномиально полные алгебры // Ученые записки Орловского государственного университета. 2012. №. 6, часть 2. С. 23–29.

[5] Markovski S., Gligoroski D, Bakeva V. Quasigroup string processing: part 1 // Contributions. Section of Mathematical and Technical Sciences. 1999. Vol. XX, no. 1–2. P. 13–28.

[6] Hagemann J., Herrmann C. Arithmetically locally equational classes and representation of partial functions // C Colloquia mathematica societatis János Bolyai (Ezstergom Conference on Universal Algebra). 1982. Vol. 29. P. 345–360.

On the invertability of finite state transducers and its applications in cryptography

Dai Yue, Zakharov Vladimir

Shenzhen MSU-BIT University; daiy0928@mail.ru, zakh@cs.msu.ru

In 1985 Tao Renji and Chen Shihua [1] proposed a public-key cryptosystem based on the Mealy automata, called Finite Automaton Public-Key Cryptography (FAPKC). The underlying idea is based on the difficulty of inverting composite finite automata. In the simplest version of this system, the public key consists of a composite of two finite automata, and the private key consists of their inverses. The security of the encryption is based on the assumption that it is difficult to invert the composite automata without knowing the private-key automata [2]. Therefore, the problem of inverting finite state machines requires close attention.

The study of this problem was first carried out in [3]. In this work, necessary and sufficient conditions for the one-to-one computations of finite transducers were established and an algorithm for constructing an inverse automaton was proposed. This solution is based on the language-theoretic technique for checking the uniquely decodability of alphabetic coding developed in [4]. The aim of our study is to modify the method proposed in [3] by using graph constructions similar to those used in A. A. Markov's algorithm [5].

Definition. *A* deterministic finite state transducer *(DFST) over an input alphabet A and an output alphabet B is 4-tuple $M = (S, s_0, \varphi, \psi)$, where S is a finite set of* states*, $s_0 \in S$ is an* initial state*, $\varphi : S \times A \to S$ is a* transition function*, and $\psi : S \times A \to B^*$ is an* output function*.*

A *run* of a DFST M on an input word $w = a_1 a_2 \dots a_n$ is a sequence of pairs $(s_0, \varepsilon), (s_1, u_1), \dots, (s_n, u_n)$ such that $s_i = \varphi(s_{i-1}, a_i)$ and $u_i = u_{i-1}\psi(s_{i-1}, a_i)$

hold for every $i, 1 \leqslant i \leqslant n$. A transducer M computes a function $F_M : A^* \to B^*$ such that for every $w \in A^*$ we have $F_M(w) = u$ iff there is a run of M on w which ends with a pair (s, u).

Without loss of generality we will assume that all states of M are reachable from the initial state s_0, i.e. for every $s \in S$ there exists a run of M which ends with a pair (s, u). By the *size* of a DFST M mean the size of tables by which transition and output functions are specified.

We say that a DFST M is *invertible* (one-to-one transducer) iff $w_1 \neq w_2$ implies $F_M(w_1) \neq F_M(w_2)$ for every pair of input words w_1 and w_2. To develop an algorithm for checking the invertability of DFSTs we follow the approach proposed by A. A. Markov (see [5]): given a DFST M, construct a finite directed labeled graph $G_M = (V_M, E_M)$ and then check certain properties of paths in G_M.

Let $M = (S, s_0, \varphi, \psi)$ be a DFST. For every state $s \in S$ define a set of output words

$$L_s = \{u : \exists\ s' \in S, a \in A,\ v \in B^*\ : \varphi(s', a) = s,\ \psi(s', a) = vu\}.$$

The vertices of graph G_M are all 4-tuples of the form (s_1, s_2, u, σ), where $s_1, s_2 \in S$, $\sigma \in \{1, 2\}$ and $u \in L_{s_2}$ if $\sigma = 1$ or $u \in L_{s_1}$ if $\sigma = 2$. Here σ indicates which of two states s_1 or s_2 is active in the vertex (s_1, s_2, u, σ).

Labeled arcs connect vertices of the graph $G_M = (V_M, E_M)$ according to the following rules.

1. An arc leads from a vertex $(s_1, s_2, u, 1)$ to a vertex $(s, s_2, v, 1)$ if there exists a letter $a \in A$ such that $s = \varphi(s_1, a)$ and $u = \psi(s_1, a)v$; this arc is labeled with a pair $(a, \psi(s_1, a))$ and it is depicted as
$$(s_1, s_2, u, 1) \overset{(a,\psi(s_1,a))}{\longrightarrow} (s, s_2, v, 1).$$
2. An arc leads from a vertex $(s_1, s_2, u, 1)$ to a vertex $(s, s_2, v, 2)$ if there exists a letter $a \in A$ such that $s = \varphi(s_1, a)$ and $\psi(s_1, a) = uv$; this arc is labeled with a pair $(a, \psi(s_1, a))$ and it is depicted as
$$(s_1, s_2, u, 1) \overset{(a,\psi(s_1,a))}{\longrightarrow} (s, s_2, v, 2).$$
3. An arc leads from a vertex $(s_1, s_2, u, 2)$ to a vertex $(s_1, s, v, 2)$ if there exists a letter $a \in A$ such that $s = \varphi(s_2, a)$ and $u = \psi(s_2, a)v$; this arc is labeled with a pair $(a, \psi(s_2, a))$ and it is depicted as
$$(s_1, s_2, u, 2) \overset{(a,\psi(s_2,a))}{\longrightarrow} (s_1, s, v, 2).$$
4. An arc leads from a vertex $(s_1, s_2, u, 2)$ to a vertex $(s_1, s, v, 1)$ if there exists a letter $a \in A$ such that $s = \varphi(s_2, a)$ and $\psi(s_2, a) = uv$; this arc is labeled with a pair $(a, \psi(s_2, a))$ and it is depicted as
$$(s_1, s_2, u, 2) \overset{(a,\psi(s_2,a))}{\longrightarrow} (s_1, s, v, 1).$$

There are no other arcs in the graph G_M.

Theorem 1. *A DFST $M = (S, s_0, \varphi, \psi)$ is not invertible iff there exists such a state $s \in S$ and such letters $a, b \in A, a \neq b$, that there exists a directed path in the graph G_M from the vertex $(s, s, \varepsilon, 1)$ to any node of the form $(s', s'', \varepsilon, \sigma)$, and this path begins with two following arcs*

$$(s, s, u, 1) \overset{(a,\psi(s,a))}{\longrightarrow} (\varphi(s,a), s, \psi(s,a), 2) \overset{(b,\psi(s,b))}{\longrightarrow} (\varphi(s,a), \varphi(s,b), u, \sigma).$$

It should be noticed that the number of vertices $|V_M|$ in the graph G_M is polynomial of the size n of M (actually, it is $O(n^4)$). Thus, we arrive at

Corollary 1. *The invertability checking problem for DFSTs is NL-complete.*

One may consider computations of DFSTs on infinite input words. Denote by A^ω the set of all infinite sequences whose elements are letters from A; such sequences are called *ω-words*. If $u_1, u_2, \ldots$ is an infinite sequence of finite words such that u_i is a proper prefix of u_{i+1} for every $i \geqslant 1$ then there exists the unique ω-word $u_\omega = \lim\limits_{i\to\infty} u_i$ such that every u_i is a prefix of u_ω. A transducer M computes together with F_M a function $F_{M,\omega} : A^\omega \to B^\omega$ such that for every $w_\omega \in A^\omega$ we have $F_{M,\omega}(w_\omega) = u_\omega$ iff there is an infinite run $(s_0, \varepsilon), (s_1, u_1), (s_2, u_2) \ldots$ of M on w_ω such that $u_\omega = \lim\limits_{i\to\infty} u_i$. We say that a DFST M is ω-invertible iff $w'_\omega \neq w''_\omega$ implies $F_{M,\omega}(w'_\omega) \neq F_{M,\omega}(w''_\omega)$ for every pair of input ω-words w'_ω and w'_ω.

Theorem 2. *A DFST $M = (S, s_0, \varphi, \psi)$ is not ω-invertible iff there exists such a state $s \in S$ and such letters $a, b \in A, a \neq b$, that there exists a directed path in the graph G_M from the vertex $(s, s, \varepsilon, 1)$ to any node of the form $(s', s'', \varepsilon, \sigma)$ which belongs to some strongly connected component of G_M, and this path begins with two following arcs*

$$(s, s, u, 1) \overset{(a,\psi(s,a))}{\longrightarrow} (\varphi(s,a), s, \psi(s,a), 2) \overset{(b,\psi(s,b))}{\longrightarrow} (\varphi(s,a), \varphi(s,b), u, \sigma).$$

Corollary 2. *The ω-invertability checking problem for DFSTs is NL-complete.*

Thus, we show that graph-theoretic approach proposed by A. A. Markov makes it possible to check efficiently unique decodability property not only for alphabetic codings but for automata codings as well.

References

[1] Tao Renji, Chen Shihua. On finite automaton public-key cryptosystem // Theoretical Computer Science. 1999. Vol. 226, no. i–2. P. 143–172.

[2] An overview of cryptosystems based on finite automata / G. Khaleel, S. Turaev, I. Al-Shaikhli, M. I. Mohd Tamrin // Journal of Advanced Review on Scientific Research. 2016. Vol. 27, no. 1. P. 1–7.

[3] Левенштейн В. И. Об обращении конечных автоматов // Доклады Академии наук СССР. 1962. Т. 147, № 6. С. 1300–1303.

[4] Sardinas A. A., Patterson G. W. A necessary and sufficient condition for the unique decomposition of coded messages // IRE International Convention Record. 1953. Part 8: Information Theory. P. 104–108.

[5] Марков А. А. Введение в теорию кодирования. М. : Наука, 1982.

On the equivalence checking problem for tree finite state automata

Deng Zhibo
Shenzhen MSU-BIT University; danel0727@outlook.com

Our research focuses on top-down finite state tree automata (FTA), whose syntax and semantics are defined formally in on-line textbook TATA [1] as follows.

Definition. *Let $\mathcal{F}$ be a finite set of functional symbols, and every $f \in \mathcal{F}$ has some arity $k \geqslant 0$. A top-down finite tree automaton (FTA) is a 4-tuple $\mathcal{A} = (Q, \mathcal{F}, I, \Delta)$, where Q is a finite set of states, $I \subseteq Q$ is a set of initial states, and Δ is a set of transition rules of the following type:*

$$(q, f) \to f(q_1, q_2, \ldots, q_k), \tag{1}$$

where $k \geqslant 0$, $f \in \mathcal{F}^k$, and $q, q_1, q_2, \ldots, q_k \in Q$.

We denote by $\Delta(\mathcal{A}, q)$ the set of all transition rules (1) from a state q of an FTA $\mathcal{A}$. An FTA $\mathcal{A}$ is *deterministic* (DFTA) if there is only one initial state and no two rules have the same left-hand side. By the size of an FTA $\mathcal{A}$ we mean the total number of letters in its transition rules.

FTAs operate on terms — finite trees whose nodes are marked with symbols in $\mathcal{F}$. A transition rule (1) means that whenever an FTA $\mathcal{A}$ at a state q observes a node marked with f, the copies of this FTA at the states $q_1, q_2, \ldots, q_k$ move to the successors of this node. When f is a constant symbol of arity $k = 0$, a transition rule $(q, f) \to f$ is called *terminating*: it fires when an FTA reaches a leaf node of an input tree marked with a constant f. An FTA A *accepts* a term t when it starts its run at the root of the tree and finally terminating rules fire at all leaves of t.

A collection of tree languages $L_{\mathcal{A}}(q)$ accepted at various states $q \in Q$ of an TFA $\mathcal{A}$ can be specified as the least solution of the following system of equations

$$L_{\mathcal{A}}(q) = \bigcup_{\delta \in \Delta(\mathcal{A}, q)} \{f(L_{\mathcal{A}}(q_1), L_{\mathcal{A}}(q_2), \ldots, L_{\mathcal{A}}(q_k)) \mid \delta : (q, f) \to f(q_1, q_2, \ldots, q_k)\}.$$

Two states q and p are called *equivalent* iff $L_{\mathcal{A}}(q) = L_{\mathcal{A}}(p)$. This paper presents an efficient equivalence checking algorithm for top-down DFTAs. This algorithm is

based on the approach proposed in [2, 3]; it allows one to build efficient equivalence checking algorithms for various types of finite state machines by reducing the problem of checking the equivalence of automata to checking the solvability of systems of language-theoretic equations.

The equivalence checking algorithm for top-down DFTAs presented in this article is divided into two main stages: constructing a system of equations which define the problem of testing the equivalence $L_{\mathcal{A}}(q) = L_{\mathcal{A}}(p)$, and checking the solvability of this system.

Stage 1. Constructing the system of equations. Suppose that we have a top-down DFTA $\mathcal{A} = (Q, \mathcal{F}, I, \Delta)$. Without loss of generality we will assume that $L_{\mathcal{A}}(q) \neq \varnothing$ for every state $q \in Q$. Our goal is to check the equivalence of a pair of states q' and q'' of $\mathcal{A}$. To this end we associate a variable X_q with each state $q \in Q$ of $\mathcal{A}$, and a term $t_\delta = f(X_{q_1}, \ldots, X_{q_k})$ with every transition rule δ of the form (1). Then the system of equations $\mathcal{E}_0$ required for our purpose is as follows:

$$\mathcal{E}_0 = \{X_q = \sum_{\delta \in \Delta(\mathcal{A},q)} t_\delta : q \in Q\} \cup \{X_{q'} = X_{q''}\}.$$

Stage 2. Checking the solvability of the equation system $\mathcal{E}_0$. The algorithm iteratively applies the following series of steps to the systems of equations, starting with $\mathcal{E}_0$ constructed at the Stage 1. Suppose that after the i-th iteration, $i \geqslant 0$, we have a system of equations $\mathcal{E}_i$.

1. *Termination detection.* If there are no equations of the form $X_p = X_q$ in $\mathcal{E}_i$ then the algorithm stops and outputs an answer: q' and q'' are equivalent states of $\mathcal{A}$.
2. *Substitution.* Otherwise, for every equation of the form $X_p = X_q$ in $\mathcal{E}_i$ replace all occurrences of the variable X_p in the equations of $\mathcal{E}_i$ with the variable X_q and remove the equation $X_p = X_q$ from $\mathcal{E}_i$.
3. *Conflict detection.* If there are two equations with the same left-hand side (say, X_q) such that one of them contains some functional symbol (say, f) in its right-hand side, whereas the other does not, then the algorithm stops and outputs an answer: q' and q'' are not equivalent states of $\mathcal{A}$.
4. *Restoration.* Otherwise, for every pair of equations with the same left-hand side (say, X_q) of the form

$$X_q = \sum f(X_{q_1}, \ldots, X_{q_k}), \tag{2}$$

$$X_q = \sum f(X_{p_1}, \ldots, X_{p_k}), \tag{3}$$

and for every functional symbol that appears in their right-hand sides (say, f) add equations

$$X_{q_1} = X_{p_1}, \ \ldots \ , X_{q_k} = X_{p_k}$$

to the system and afterwards remove from the system one of the equations (2) or (3). Denote by $\mathcal{E}_{i+1}$ the system of equations obtained thus, and this is where the iteration i of the algorithm ends.

Theorem 1. *For every top-down DFTA $\mathcal{A} = (Q, \mathcal{F}, I, \Delta)$ and a pair of its states q' and q'' the algorithm specified above always terminates and correctly recognizes the equivalence of q' and q'' in time $O(n^2)$ where n is the size of $\mathcal{A}$.*

Proof. (Sketch) The size of the initial system $\mathcal{E}_0$ is the same as the size of an DFTA $\mathcal{A}$. After every step of the algorithm the triple (n_1, n_2, n_3) lexicographically decreases, where n_1 is the number of variables, n_2 is the number of equations with non-variable right-hand side, and n_3 is the total number of equations in the system. Therefore, the algorithm always terminates.

The following four considerations confirm the correctness of the algorithm.

1. The system $\mathcal{E}_0$ has a solution (actually, it is $X_q = L_{\mathcal{A}}(q)$ for every $q \in Q$) iff the states q' and q'' are equivalent.
2. The transformations of the systems of equations at every step of the algorithm preserve their solvability.
3. A system of equation which satisfies the termination detection condition always has a solution.
4. A system of equation which satisfies the conflict detection condition does not have a solution.

Time complexity of the algorithm is estimated in the framework of computational model RAM with pointers. As it can be seen from the description of the algorithm, the number of its iterations does not exceed the number of states $|Q|$ of DFTA. The number of actions at each step of the algorithm does not exceed the size of the systems $\mathcal{E}_i$ which decreases monotonically. □

References

[1] Tree automata techniques and applications / H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, C. Löding, S. Tison, M. Tommasi // INRIA.HAL.SCIENCE Web Portal. 2008. Retrieved 19.04.2025 from https://inria.hal.science/hal-03367725.

[2] Zakharov V.À. Equivalence checking of prefix-free transducers and deterministic two-tape automata // Language and Automata Theory and Applications. LATA 2019. Lecture Notes in Computer Science. 2019. Vol. 11417. P. 146–158.

[3] Zakharov V.À. Efficient equivalence checking technique for some classes of finite-state machines // Automatic Control and Computer Sciences. 2021. Vol. 55, no. 7. P. 670–701.

On some specific features of set multicover problem

Li Ilin, Zakharov Vladimir
Shenzhen MSU-BIT University; 1120200005@smbu.edu.cn, zakh@cs.msu.ru

Set multicover problem is a natural generalization of the well-known set cover problem: given a finite set and a family of its subsets find a minimal subfamily of subsets which covers the set; in the case of set multicover some elements must be covered several times.

Let U be a finite set. A *multiset* is any function $M : U \to \{0, 1, 2, \dots\}$. A value $M(e)$ can be viewed as the number of occurrences (copies) of an element e in a multiset M. A *domain* D is such a multiset that $D(e) \leqslant 1$ holds for every $e \in U$. A *family* is any finite collection of domains $S = \{D_1, \dots, D_m\}$. Set-theoretic relations and operations are extended to multisets as follows: 1) $e \in M \iff M(e) > 0$, 2) $M_1 \subseteq M_2 \iff M_1(e) \leqslant M_2(e)$ for every $e \in U$, 3) $M_1 \cup M_2 = M_1 + M_2$, 4) $M_1 \cap M_2 = \min(M_1, M_2)$, 5) $M_1 \setminus M_2 = M_1 - M_2$, where $M_1(e) - M_2(e) = 0$ if $M_1(e) < M_2(e)$. A *set multicover problem* is a pair (M, S), where M is a multiset and S is a family. A family S' which is a subset of S is a *multicover* of (M, S) iff $M \subseteq \bigcup_{D \in S'} D$, and it is a *solution* to (M, S) if S' is a minimal multicover. A *set cover problem* is a variant of set multicover problem (D, S) when D is a domain.

Since set multucover problem is similar to set cover problem, many results, techniques and algorithms developed for the latter can be easily adapted to the former. Both problems are NP-complete [1], admit simple reduction to integer programming problem, and their approximate solutions can be obtained by means of greedy algorithms. Therefore, the study of set multicover problem has mainly been limited to analyzing efficiency and accuracy of various approximation algorithms [2–4]. Meanwhile, we show that the difference between these two covering problems is more substantial, and it affects considerably the applicability and efficiency of some common approaches to these problems.

Some distinctions between set cover and set multicover problems are obvious.

1. A domain D' in a family S is called *maximal* if it is not included in any other domain, i. e. $D'' \in S, D'' \neq D' \implies D' \not\subseteq D''$. It is well known that any set cover problem (D, S) has a solution S' which consists of maximal domains only. This consideration is not true in general for multicover problems, and we can not simplify (M, S) by deleting all non-maximal domains from S.
2. If D is a (set-theoretic) union of two domains D' and D'' then any solution S_0 to set cover problems (D', S) and (D'', S) is always a solution to (D, S). But this is not the case when multisets are concerned: a multicover S_0 of both (M', S) and (M'', S) is not necessarily a multicover of $(M' \cup M'', S)$.

3. A multicover S' of (M, S) is called *reduced* if any proper subfamily S'' of S' is not a cover of (M, S). Denote by $Red(M, S)$ the collection of reduced multicovers of (M, S). To solve covering problem, one can use the following approach, which originates from the solution of the DNF minimization problem. Given a multicover problem (M, S), one may compute two families $\Pi Red(M, S)$ and $\Sigma Red(M, S)$. The former includes the domains which appear in *all* reduced multicovers of (M, S), and the latter consists of the domains which occur in *at least* one reduced multicover of (M, S). Then any solution S' to a multicover problem $(M \setminus \bigcup_{D \in \Pi Red(M,S)} D, \Sigma Red(M, S) \setminus \Pi Red(M, S))$ gives a solution $S'' = S' \cup \Pi Red(M, S)$ to the multicover problem (M, S). If the families $\Pi Red(M, S)$ and $\Sigma Red(M, S)$ can be computed efficiently (as it is for set cover problem) then this approach can alleviate considerably the computation of solutions to (M, S).

For every family S denote by M_S a multiset $\bigcup_{D \in S} D$. The following statements hold for every multicover problem (M, S).

Theorem 1. *A multicover S' of (M, S) is reduced iff for every domain $D \in S'$ there exists such an element $e \in D$ that $M(e) = M_{S'}(e)$.*

Theorem 2. *A domain D belongs to $\Pi Red(M, S)$ iff $D \in S$ and there exists such an element $e \in D$ that $M(e) = M_S(e)$.*

Clearly, both reduction checking and membership checking for $\Pi Red(M, S)$ can be performed in linear time (when M and S are specified explicitly), and, therefore, the family $\Pi Red(M, S)$ can be computed efficiently (by local algorithms, in terms of [5]). However, this cannot be said about $\Sigma Red(M, S)$.

Consider a family $S = \{D_1, D_2, \ldots, D_n\}$ and let an element $e \in M$ is such that $e \in D_{i_1}, e \in D_{i_2}, \ldots, e \in D_{i_k}$ and $e \notin D_{i_{k+1}}, \ldots, e \notin D_{i_n}$. Let also $\widehat{S}_e = \{D_{i_2}, \ldots, D_{i_k}\}$ and $\widehat{M}_e = M \setminus (D_{i_1} \cup D_{i_{k+1}} \cup \ldots, \cup D_{i_n})$. Then we say that e is a *loose element of a domain D_{i_1} w.r.t.* (M, S) iff the multicover problem $(\widehat{M}_e, \widehat{S}_e)$ has a solution of the size greater than or equal to $M(e)$.

Theorem 3. *A domain D does not belong to $\Sigma Red(M, S)$ iff every element $e \in D$ is a loose element of D w.r.t. (M, S).*

As it can be seen from the definition above, checking the looseness property has the same complexity as a decision variant of multicover problem: check that the size of all multicovers of (M, S) are greater than or equal to d.

Corollary 1. *The membership problem $D \in \Sigma Red(M, S)$ is NP-complete.*

Thus, unlike the case of set cover problem, when multisets are concern we have no means for computing the family $\Sigma Red(M, S)$ in polynomial time, and this is another aspect, where set covering and set multicovering differ.

References

[1] Garey M. R., Johnson D. S. Computers and intractability. A guide to the theory of NP-completeness. New York, NY, USA : W. H. Freeman and Company, 1979.

[2] Exact Algorithms for set multicover and multiset multicover problems / Qiang-Sheng Hua, Dongxiao Yu, F. C. M. Lau, Yuexuan Wang // Algorithms and Computation. ISAAC 2009. Lecture Notes in Computer Science. 2009. Vol. 5878, P. 34–44.

[3] Berman P., DasGupta B., Sontag E. Randomized approximation algorithms for set multicover problems with applications to reverse engineering of protein and gene networks // Discrete Applied Mathematics. 2007. Vol. 155, no. 6–7. P. 733–749.

[4] Dynamic programming based algorithms for set multicover and multiset multicover problem / Qiang-Sheng Hua, Yuexuan Wang, Dongxiao Yu, F. C. M. Lau // Theoretical Computer Science. 2010. Vol. 411. P. 2467–2474.

[5] Журавлев Ю. И. Об алгоритмах упрощения дизъюнктивных нормальных форм // Доклады Академии наук СССР. 1960. Т. 132, № 2. С. 260–263.

Decision problems for parameterized weakly synchronous finite state transducers

Tang Tianxiang

Shenzhen MSU-BIT University; tangtx@smbu.edu.cn

The authors of [1] showed that the complexity of decision-making problems for regular expressions increases significantly when parameters are allowed to be used in such expressions along with letters. However, in regular expressions it is not possible to separate deterministic from non-deterministic computations. Therefore, in [2] we investigated a parameterized version of the synchronous finite state transducers, in which parameterization is allowed only for outputs. Synchronization implies that for each input signal the machine necessarily produces some response at the output. We found that for deterministic synchronous transducers, the presence of parameters does not have such a large impact on the complexity of decision problems. However, this computational model does not cover all applications where data parameterization can be used, and we continued our research by expanding the capabilities of the model. In the new computation model, transitions are allowed on which the automaton does not produce any output signals; in this case the length of the output word may be less than the length of the input word. We called this type of transducers weakly synchronous. This paper presents

the results of our study of the most important decision problems for deterministic and nondeterministic weakly synchronous automata.

A *Parameterized Weakly Synchronous Finite State Transducer* (PWFST for short) over an input alphabet Σ and an output alphabet Δ is such a transition system $\Pi = \langle Q, q_0, F, \mathcal{X}, \longrightarrow \rangle$, where Q is a finite set of states, q_0 is an initial state, $F \subseteq Q$ is a subset of final states, $\mathcal{X}$ is an an infinite set of variables disjoint with both alphabets Σ and Δ and $\longrightarrow$ is a transition relation of the type $Q \times \Sigma \times (\Delta \cup \mathcal{X} \cup \{\varepsilon\}) \times \mathcal{Q}$. Quadruples $(q, a, b, q') \in \longrightarrow$ are called transitions and depicted as $q \xrightarrow{a/b} q'$. A PWFST Π is *deterministic* (Det-PWFST) if for every state $q \in Q$ and a letter $a \in \Sigma$ there are no different transtions $q \xrightarrow{a/z_1} q'$ and $q \xrightarrow{a/z_2} q''$ in Π.

A *run* of Π is any finite sequence of transitions $q_0 \xrightarrow{a_1/b_1} q_1 \xrightarrow{a_2/b_2} \cdots \xrightarrow{a_n/b_n} q_n$, where $q_n \in F$. Such run is denoted as $q_0 \xrightarrow{u/w}_* q_n$, where $u = a_1 a_2 \dots a_n, w = b_1 b_2 \dots b_n$. Since the empty symbol ε may appear in transitions of PWFSTs, the length of an input word u in a run $q_0 \xrightarrow{u/w}_* q_n$ may exceed the length of an output string w.

When a transition relation $\longrightarrow$ is of the type $Q \times \Sigma \times (\Delta \cup \{\varepsilon\}) \times Q$ (there are no variables in transitions) such a transducer is called *Weakly Synchronous Finite State Transducer* (WFST for short). Any WFST π computes a *transduction relation* $TR(\pi) = \{(u, w) :$ there exists a run $q_0 \xrightarrow{u/w}_* q_n$ of $\pi\}$ on $\Sigma \times \Delta$. Most decision problems on transducers concern the properties of transduction relation.

A function $\theta : \mathcal{X} \to \Delta$ is called a ground substitution. The set of all ground substitutions is denoted by $GSubst$. Applying a substitution θ to a PWFST Π results in an WFST $\Pi\theta$ in which every occurrence of any variable x in the transitions of Π is replaced by an output letter $\theta(x)$. In this case a WFST $\pi = \Pi\theta$ is called an instance of Π. Thus, each PWFST Π generates a whole family of WFSTs $\mathcal{F}(\Pi) = \{\Pi\theta : \theta \in Gsubst\}$, and all typical decision problems for checking certain properties (like non-emptiness, functionality, 2-valuness) or relations (like equivalence, bisimilarity) on WFSTs can be quite naturally addressed to PWFSTs as well. These decision problems can be specified by unary or binary predicates of the form $P(\Pi)$ or $R(\Pi_1, \Pi_2)$ and they have dual presentation, but due to space limitation in this paper only existential variants of these problems are considered: given a PWFST Π or a pair of PWFSTs Π', Π'' check the predicates $\exists\theta \in GSubst : P(\Pi\theta)$ or $\exists\theta', \theta'' \in GSubst : R(\Pi'\theta', \Pi''\theta'')$. In what follows we denote by n the number of states $|Q|$ and by m the number of transitions $\longrightarrow$ in a PWFST.

1. **Membership**: given an WFST π and a pair of words $(u, w) \in \Sigma^* \times \Delta^*$, check $(u, w) \in TR(\pi)$.

Statement 1. *The ∃-membership problem is NP-complete for PWFSTs, L-complete and can be decided in time $O(n)$for Det-PWFSTs.*

2. **Nonemptiness**: given an WFST π, check $TR(\pi) \neq \varnothing$.

   Statement 2. *The ∃-nonemptiness problem is NL-complete for both PWFSTs and Det-PWFSTs, and can be decided in time $O(n \log n)$.*

3. **Functionality**: given an WFST π, check if $TR(\pi)$ is a (partial) function $TR(\pi) : \Sigma^* \to \Delta^*$.

   Statement 3. *The ∃-functionality is NL-complete for PWFSTs and can be decided in time $O(m^3 \log m)$.*

4. **k-valuedness**: given an FST π, check if for every input word $u \in \Sigma^*$ it is true that $|\{w : (u, w) \in TR(\pi)\}| \leqslant k$.

   Statement 4. *The ∃-k-valuedness is NP-complete for PWFSTs.*

5. **Equivalence**: given a pair of WFSTs π_1, π_2, check $TR(\pi_1) = TR(\pi_2)$.

   Statement 5. *The ∃-equivalence problem is PSPACE-complete for PWFSTs, NL-complete and can be decided in time $O(m^3 \log m)$ for Det-PWFSTs.*

6. **Minimization**: given an WFST π and an integer k, check if there exists such an FST $TR(\pi')$ that $TR(\pi) = TR(\pi')$ and $size(\pi') \leqslant k$.

   Statement 6. *The ∃-minimization problem is is PSPACE-complete for PWFSTs and NP-complete for Det-PWFSTs.*

7. **Problem Bisimulation**: given a pair of WFSTs π_1, π_2, check if π_1 and π_2 are bisimilar.

   Statement 7. *The ∃-bisimulation problem is is NP-complete for PWFSTs.*

The results obtained are collected in the table below. As comparing with the complexity of decision problems for WFSTs, one may see that parametrization affects significantly such problems as Membership, k-valuedness, and Bisimulation for nondeterministic transducers, and Minimization for deterministic transducers, but there are also cases when designing of efficient algorithms is possible even in the presence of parameters in transitions of automata.

References

[1] Barceló P., Reutter J., Libkin L. Parameterized regular expressions and their languages // Theoretical Computer Science. 2013. Vol. 474, P. 21–45.

[2] Tang T., Zakharov V. A. On the complexity of decision problems for parameterized finite state synchronous transducers // Implementation and Application of Automata. CIAA 2024. Lecture Notes in Computer Science. 2024. Vol. 15015. P. 332–346.

Table 1: Complexity of decision problems for PFSTs and PWFSTs

Problems	WFSTs			PWFSTs		
	Non-Det	Det	Time	Non-Det	Det	Time
Membership	L	NL	$O(n)$	L	NP	$O(n)$
Nonemptiness	NL	NL	$O(n \log n)$	NL	NL	$O(n \log n)$
Functionality	NL	—	$O(m^2 \log n)$	NL	—	$O(m^3 \log n)$
k-valuednes	NL	—	$O(m^2 \log n)$	NP	—	—
Equivalence	PSPACE	NL	$O(n \log n)$	PSPACE	NL	$O(n^3 \log n)$
Minimization	PSPACE	NL	$O(n \log n)$	PSPACE	NP	—
Bisimulation	P	—	$O(m^2 \log n)$	NP	—	—

An LTS-based semantics of improved variant of Real-Time Finite State Machines

Zhang Yao, Zakharov Vladimir
Shenzhen MSU-BIT University; 2120230008@smbu.edu.cn, zakh@cs.msu.ru

The concept of real-time finite state machines (TFSMs) appeared explicitly in [1] as an attempt to adapt timed automata [2] to modeling simple reactive systems and to avoid the undecidability of most decision problems inherent in general timed automata. TFSM can be viewed as a kind of Mealy automaton, in which the firing of a transition depends not only on an input signal, but also on the time of its appearance, and responses are not given immediately, but with some delay. Further research [3, 4] showed that the most important algorithmic problems for TFSMs can be reduced to those for conventional finite state automata and this turned out to be a significant advantage of the proposed model. Meanwhile, the original version of the TFSM model has a significant drawback: the output signals that the machine generates follow, regardless of the time of their appearance, in the same order in which the input signals were received. As a result, the computations of TFSM do not fully correspond to the observed behavior of the simulated system. To correct this deficiency, an improved version of the TFSM model was proposed in [5]. However, the operational semantics of the improved model was defined in [5] implicitly and did not allow the application of well-known model checking means to TFSMs of the new type. The purpose of

this paper is to eliminate this drawback and present an explicit description of the semantics of TFSM based on labeled transition systems.

Let $\Re^+$ denote the set of all positive reals, and $\Re_0^+ = \Re^+ \cup \{0\}$. A *timed letter* is any pair (x,t), where x is a letter, and $t \in \Re_0^+$. A *timed word* is any sequence $\gamma = (x_1,t_1),(x_2,t_2),\ldots,(x_n,t_n)$ of timed letters such that $t_1,t_2,\ldots,t_n$ is a non-decreasing sequence of reals. Given a set of timed letters P we denote by $order(P)$ a timed word which is composed of all elements of P. A *guard* is an interval of the type $(u,v]$, where $u,v \in \Re_0^+$ and $u < v$.

A Real-Time Finite State Machines (RTFSM) over an input alphabet A and an output alphabet B is a quadruple $M = (S, s_{in}, G, \rho)$, where S is a nonempty set of *control states*, $s_{in} \in S$ is an *initial state*, G is a set of guards, and $\rho \subseteq (S \times I \times O \times S \times G \times \Re^+)$ is a finite set of *actions*. Every action $(s,a,b,s',g,d) \in \rho$ means that whenever a RTFSM is at a control state s and receives an input signal a at time t assuming that the previous input signal has been received at time t' such that $t - t' \in g$ then RTFSM moves to a control state s' and outputs a response b at time $t+d$.

A run of RTFSM M is any sequence of moves

$$tr{=}(s_0,a_1,b_1,s_1,(u_1,v_1],d_1),(s_1,a_2,b_2,s_2,(u_2,v_2],d_2),\ldots,(s_{n-1},a_n,b_n,s_n,(u_n,v_n],d_n). \tag{1}$$

A run (1) *responds* to an input timed word $\alpha = (a_1,t_1),(a_2,t_2),\ldots,(a_n,t_n)$ with an output timed word $\beta = order(\{(b_1,t_1+d_1),(b_2,t_2+d_2),\ldots,(b_n,t_n+d_n)\})$ if the conditions $t_j - t_{j-1} \in (u_j,v_j]$ hold for all $j, 1 \leqslant j \leqslant n$, assuming that $t_0 = 0$.

For every RTFSM $M = (S, s_{in}, G, \rho)$ we define a Labeled Transition System (LTS) $L(M) = (Q, q_0, \rightarrow)$, where $Q = S \times \Re_0^+ \times \Re_0^+ \times 2^{O \times \Re_0^+}$ is a set of *configurations*, $(s_{in},0,0,\varnothing)$ is the *initial configuration*, and

$$\rightarrow \subseteq (Q \times \tau \times Q) \cup (Q \times (I \times \Re^+) \times Q) \cup (Q \times (O \times \Re^+) \times Q)$$

is a *transition relation*.

A transition relation $\rightarrow$ of an LTS $L(M)$ conforms 3 rules for every configuration $q = (s, T_0, T_1, B)$:

— **advancement of time**: for every $t' \in \Re^+$ such that $t' \leqslant \min(t : (y,t) \in B)$ there exists a transition $(s,T_0,T_1,B) \xrightarrow{\tau} (s, T_0+t', T_1+t', B-t')$;

— **input move**: for every action $(s,x,y,(u,v],d)$ in RTFSM M for which $u < T_1 \leqslant v$ holds there is a transition $(s,T_0,T_1,B) \xrightarrow{(x,T_0)} (s',T_0,0,B \cup \{(y,d)\})$;

— **output move**: for every timed output letter $(y,0) \in B$ there is a transition $(s,T_0,T_1,B) \xrightarrow{(y,T_0)} (s,T_0,T_1,B \setminus \{(y,0)\})$.

A *trace* of RTFSM M is any sequence of transitions in LTS $L(M)$:

$$tr = (s_{in},0,0,\varnothing) \xrightarrow{r_1} (s_1,t_1,t_1',B_1) \xrightarrow{r_2} (s_2,t_2,t_2',B_2) \xrightarrow{r_3} \ldots \xrightarrow{r_n} (s_n,t_n,t_n',B_n). \tag{2}$$

A sequence $h(tr) = r_1, r_2, r_3, \ldots, r_n$ of symbols τ and timed letters is a *history* of the trace (2). The maximal subsequence of $h(tr)$ which consists of timed input (output) letters is called an *input* (respectively, *output*) *history* of a trace (2). A trace (2) is called *exhaustive* if $B_n = \varnothing$. A configuration $q = (s, T, t, B)$ is called *reachable* in *LTS* $L(M)$, if there exists a trace which ends with q. If q is an exhaustive configuration then such a trace is also *exhaustive*.

Statement 1. *For every RTFSM M, for every configuration $q = (s, T_0, T_1, B)$ reachable in LTS $L(M)$, and for every timed letter $(y, t) \in B$ it is true that $T_1 \leqslant T_0$ and $0 \leqslant t$.*

Statement 2. *For every RTFSM M there exists such $T \in \Re^+$ that for every configuration $q = (s, T_0, T_1, B)$ reachable in LTS $L(M)$ and for every timed letter $(y, t) \in B$ it is true that $t \leqslant T$.*

Statement 3. *For every RTFSM M, for every timed input word α and output word β, if there is a run of RTFSM M on α which outputs β, then there is an exhaustive trace in LTS $L(M)$with an input history α and output history β.*

Statement 4. *For every RTFSM M, every timed input word α and output word β, if there is an exhaustive trace in LTS $L(M)$with an input history α and output history β, then there is a run of RTFSM M on α which outputs β.*

A configuration $q = (s, T_0, T_1, \{(y_1, t_1), \ldots, (y_k, t_k)\})$ is called *integral* if all numbers $T_0, T_1, t_1, \ldots, t_k$ are integers. Two configurations $q' = (s', T_0', T_1', B')$ and $q'' = (s'', T_0'', T_1'', B'')$ are called *equivalent* if $s' = s'', T_1' = T_1'', B' = B''$.

Statement 5. *For every RTFSM M the equivalence relation is a bisimulation relation on the set of configurations of LTS L(M), and the subset of reachable integral configurations can be partitioned on finitely many classes of equivalence.*

Statements 3 and 4 show that the operational semantics of RTFSM based on LTS captures the concept of computation of this model defined in [5] and can thus be used to reason about the behavior of RTFSMs. Moreover, as it follows from Statement 5, when timestamps in the input timed words and the delays in the actions of RTFSM M are integers (which is the case in many applications), an LTS $L(M)$ can be compressed to a finite state transition system which captures the runs of M. This consideration opens the way for developing efficient synthesis and analysis algorithms for an improved version of real-time finite state machines.

References

[1] Distinguing non-deterministic timed finite state machines / M. Gromov, K. El-Fakih, N. Shabaldina, N. Yevtushenko // Formal Techniques for Distributed Systems. FMOODS FORTE 2009. Lecture Notes in Computer Science. 2009. Vol. 5522, P. 137–151.

[2] Alur R., Dill D. L. A Theory of timed automata // Theoretical Computer Science. 1994. Vol. 126, no. 2. P. 183–235.

[3] Deterministic timed finite state machines: equivalence checking and expressive power / D. Bresolin, K. El-Fakih, T. Villa, N. Yevtushenko // Proceedings Fifth International Symposium on Games, Automata, Logics and Formal Verification. Electronic Proceedings in Theoretical Computer Science. 2014. Vol. 161. P. 203–216.

[4] Tvardovskii A. S., Yevtushenko N. V. Minimizing timed finite state machines // Tomsk State University Journal of Control and Computer Science. 2014. No. 4 (29). P. 77–83.

[5] Винарский Е. М., Захаров В. А. К проверке строго детерминированного поведения временных конечных автоматов // Труды Института системного программирования РАН. 2018. Т. 30, № 3. С. 325–340.

Тезисы постерных докладов

Алгоритм построения оптимальной ограниченной по диаметру и степеням вершин заполняющей топологии

Есипова Дарья Владимировна
Московский государственный университет имени М. В. Ломоносова; daria2002yes@gmail.com

Заполняющая топология является одним из элементов маршрутизации в компьютерных сетях. Она используется для определения пути передачи данных в сети, когда точные маршруты неизвестны или когда требуется доставка данных всем узлам в сети. Использование такой топологии позволяет значительно снизить нагрузку на сеть.

Пусть V — набор узлов, которые представляют маршрутизаторы исходной сети, $E = \{(u, v) | u \in V, v \in V, u \neq v\}$ — набор рёбер, которые представляют соединения между узлами сети, $w(e) : E \to \mathbb{R}$ — весовая характеристика. $G = (V, E)$ — неориентированный взвешенный связный граф, моделирующий исходную сеть. Тогда Δ-*заполняющей топологией* графа G является остовный подграф $G' = (V, E')$, $E' \subset E$, удовлетворяющий ограничениям:

1. $\forall v, v \in G', deg(v) \leqslant \Delta$, где $deg(v)$ — степень вершины.
2. G' — рёберно 2-связный граф.

Диаметр заполняющей топологии $D(G)$ — это максимальное значение кратчайшего расстояния между парами вершин в G. *Вес заполняющей топологии* $W(G)$ — сумма весов всех рёбер топологии. Заполняющая топология $G' \subseteq G$ называется *оптимальной*, если не существует другой топологии $G'' \subseteq G$, такой что $W(G'') \leqslant W(G')$ и $D(G'') \leqslant D(G')$, и при этом хотя бы одно из указанных неравенств выполняется строго.

В данной работе разработан алгоритм, основанный на алгоритме Краскала, находящий близкую к оптимальной Δ-заполняющую топологию, показавший лучшие результаты при сравнении на топологиях fat-tree и spine-leaf с алгоритмами из статьи [1], берущими за основу алгоритм Прима. В статье [2] доказана NP-полнота задачи о минимальном заполняющем дереве, что говорит об алгоритмической трудности задачи, даже в таких простых случаях, когда под топологией понимается дерево.

Список литературы

[1] Lim H., Kim C. Flooding in wireless ad hoc networks // Computer Communications. 2001. Vol. 24, no. 3–4. P. 353–363.

[2] Shupletsov M., Romanov D., Stepanov E. Flooding topology algorithms for computer networks // 2022 International Conference on Modern Network

Technologies (MoNeTec). Washington, DC, USA : IEEE Computer Society, 2022. P. 1–12.

Алгоритм распознавания эмоций на основе линейной регрессии

Ковалёва Елена Сергеевна
Московский государственный университет имени М. В. Ломоносова; e.kovaleva.msu@yandex.ru

В докладе предлагается эффективный метод распознавания эмоций на основе линейной регрессии с применением ключевых точек из двух нейронных сетей MediaPipe и Dlib, из которых были взяты наиболее точно расположенные ключевые точки.

Введение

Распознавание эмоций по лицевым изображениям является важной задачей в области искусственного интеллекта и компьютерного зрения. Разработанный алгоритм использует сочетание линейной регрессии для обработки признаков лица и нейронные сети из библиотек MediaPipe и Dlib для улучшения точности и эффективности распознавания эмоций.

Линейная регрессия для распознавания эмоций

Линейная регрессия используется для создания модели, которая предсказывает эмоциональные состояния на основе координат ключевых точек лица (глаза, рот, брови). Этот подход эффективен для базового распознавания эмоций, таких как счастье, грусть или удивление, благодаря своей простоте и быстроте вычислений.

Использование MediaPipe и Dlib

— MediaPipe: библиотека, предоставляющая предобученные модели для детектирования ключевых точек лица и анализа позы. Используется для быстрого и точного извлечения характеристик лица.

— Dlib: библиотека, которая применяется для точной локализации лицевых маркеров и дополнительных признаков. Сочетание Dlib с MediaPipe позволяет улучшить точность распознавания за счет интеграции данных с различных моделей.

Совмещение подходов

Предложенный алгоритм объединяет данные, полученные из MediaPipe и Dlib, с результатами линейной регрессии. Это позволяет использовать как простые математические модели для предсказания эмоций, так и более сложные нейронные сети для повышения точности и устойчивости к различным внешним факторам, таким как изменение освещения или угла наклона лица.

Преимущества предложенного алгоритма

— Быстрота обработки за счет применения линейной регрессии.

— Высокая точность распознавания эмоций благодаря использованию сетей MediaPipe и Dlib.

Заключение

Алгоритм распознавания эмоций, основанный на линейной регрессии и интеграции нейронных сетей MediaPipe и Dlib, демонстрирует высокую эффективность. Будущие исследования будут направлены на улучшение модели за счет расширения набора эмоциональных состояний и применения более сложных моделей регрессии.

Контурирование областей на изображениях лиц методом кластеризации пикселей

Ковалёва Елена Сергеевна
Московский государственный университет имени М. В. Ломоносова; e.kovaleva.msu@yandex.ru

В докладе предлагается эффективный метод кластеризации пикселей для контурирования областей на изображениях лиц, с помощью которого возможно более точно идентифицировать объекты лица (области глаз, носа, рта, овала лица, бровей и другие).

Введение

Контурирование областей основных объектов на изображениях лиц является важной задачей в областях искусственного интеллекта и компьютерного зрения. В докладе представлен новый детерминированный алгоритм кластеризации, позволяющий выделять области ключевых элементов лица на изображениях без применения нейронных сетей.

Идея алгоритма

Идея алгоритма основана на математическом анализе цветовых расстояний между соседними пикселями относительно малых и больших областей.

Преимущества предложенного алгоритма

— Высокая скорость обработки.
— Устойчивость к изменениям освещения.

Заключение

Алгоритм контурирования областей лица демонстрирует высокую точность. Будущие исследования будут направлены на детекцию ключевых точек на основных объектах лица.

О групповой сложности бесконечных слов

Лаунер Максим Вячеславович
Санкт-Петербургский государственный университет; mlauner_official@bk.ru

Словом называется последовательность символов — элементов конечного множества, называемого алфавитом. Фактором слова называется подпоследовательность подряд идущих символов. Комбинаторная сложность бесконечного слова w — это функция, которая сопоставляет натуральному числу n число различных факторов данного слова длины n. У этого понятия сложности существует ряд обобщений, например, абелева сложность. Два конечных слова называются абелево эквивалентными, если одно получается из другого какой-либо перестановкой символов. Соответственно, абелева сложность считает только число классов абелевой эквивалентности факторов длины n.

В работе [1] Шарлье, Пузынина и Замбони ввели понятие групповой сложности бесконечных слов. Рассмотрим последовательность $(G_1, G_2, G_3, \dots)$, подгрупп G_k симметрических групп S_k; группа G_k действует на строках длины k естественным образом — перестановкой символов. Групповая сложность, соответствующая этой последовательности — это функция, сопоставляющая длине фактора n число различных классов групповой эквивалентности; здесь два фактора u и v длины k считаются эквивалентными, если существует элемент $g \in G_k$, такой что $gu = v$. Понятие групповой сложности обобщает понятия комбинаторной и абелевой сложностей слов. Несложно понять, что групповая сложность не меньше абелевой сложности и не больше комбинаторной. Возникает естественный вопрос: для каких слов существует набор

групп, позволяющий получить все значения групповой сложности между абелевой сложностью и комбинаторной? В дипломной работе Е. А. Волошиновой показано, что для слов Штурма такое свойство выполнено.

В этой работе найден класс двоичных слов, связанный со словами Штурма, для которых также выполнено исследуемое свойство. Кроме того, найден класс тернарных слов, для которых можно получить все сложности для длины $n \geqslant 3$. Изучены возможные значения групповой сложности для тернарных слов минимальной сложности, классифицированных в работах [2, 3].

Список литературы

[1] Charlier É., Puzynina S. A., Zamboni L. Q. On a group theoretic generalization of the Morse-Hedlund theorem // Proceedings of the American Mathematical Society. 2017. Vol. 145, no. 8. P. 3381–3394.

[2] Cassaigne J. Sequences with grouped factors // Proceedings of the 3rd International Conference Developments in Language Theory, DLT 1997. Thessaloniki, Greece, 1997. P. 211–222.

[3] Kaboré I., Tapsoba T. Combinatoire de mots récurrents de complexité n+2 // RAIRO-Theoretical Informatics and Applications. 2007. Vol. 41, no. 4. P. 425–446.

Свойства многочленов двухполюсных вероятностных контактных схем

Порошин Богдан Алексеевич
Московский государственный университет имени М. В. Ломоносова; poroshin.bogdan@mail.ru

Данная работа посвящена изучению двухполюсных вероятностных контактных схем в качестве преобразователей дискретных вероятностных распределений. Отметим, что вероятностные контактные схемы задают преобразования случайных величин в виде полиномов от этих вероятностей. Мы называем эти полиномы многочленами вероятности схем. Одним из первых вероятностные контактные схемы рассматривал К. Шеннон [1] в задаче о надежных схемах из ненадежных элементов. Затем вероятностные контактные схемы рассматривались с точки зрения задачи выразимости вероятностных распределений в работах Р. Л. Схиртладзе [2, 3], Ф. И. Салимова (например, [4]) и Р. М. Колпакова (например, [5]).

В данной работе рассматриваются многочлены, выражающие вероятности, реализуемые вероятностными контактными схемами. Изучаются некоторые свойства многочленов вероятности схем. Получены утверждения, описывающие связь между некоторыми из коэффициентов многочлена вероятности схемы и топологической структурой схемы. В частности, первая производная

многочлена вероятности схемы в точке 0 равна числу рёбер, соединяющих полюса схемы напрямую, а в точке 1 — числу рёбер, размыкающих данную схему. Получено следствие из работы К. Шеннона [1], позволяющее при наличии информации о значениях производной многочлена вероятности схемы в точках 0 и 1 дать нижнюю и верхнюю оценку его значениям на интервале $(0, 1)$. Также исследуются многочлены вероятности параллельно-последовательных схем (далее π-схем). Мы называем π-подсхемой данной π-схемы любой подграф этой схемы, являющийся π-схемой с теми же полюсами. Для π-схем предложено их описание в терминах схем из функциональных элементов в базисе $\{\&, \vee\}$, а также получен результат, позволяющий дать полное определение всех коэффициентов многочлена вероятности такой схемы посредством множества всех π-подсхем данной схемы.

Список литературы

[1] Шеннон К. Работы по теории информации и кибернетике. М. : Издательство иностранной литературы, 1963.

[2] Схиртладзе Р. Л. О синтезе р-схемы из контактов со случайными дискретными состояниями // Сообщения Академии наук Грузинской ССР. 1961. Т. 26, № 2. С. 181—186.

[3] Схиртладзе Р. Л. О методе построения булевой величины с заданным распределением вероятностей // Дискретный анализ. 1966. Вып. 7. Новосибирск : Институт математики СО АН СССР, 1966. С. 71–80.

[4] Салимов Ф. И. К вопросу моделирования булевых случайных величин функциями алгебры логики // Вероятностные методы и кибернетика. Казань : Издательство Казанского университета, 1979. Вып. 15. С. 68–89.

[5] Колпаков Р. М. Дискретные преобразования вероятностных распределений // Современные проблемы математики и механики. М. : Издательство МГУ, 2009. Т. III. Математика, вып. 3. Дискретная математика. С. 35–50.

Метод сведения задачи логического синтеза оптимальных по динамической и статической активности схем к задаче выполнимости булевых формул

Фаизов Алексей Игоревич
Московский государственный университет имени М. В. Ломоносова; alexfaizov18@gmail.com

Одной из важных задач при моделировании интегральных схем является задача точного логического синтеза. Она сосредоточена на построении схем, оптимальных по некоторым функционалам сложности. В докладе рассматриваются статическая и динамическая активность, которые характеризуют

энергопотребление схемы, связанное с высокими значениями напряжения и изменением напряжения в узлах схемы соответственно.

В [1] были обобщены и систематизированы подходы к решению задачи точного синтеза. В работе рассматривается решение на основе задачи выполнимости булевых формул (ВЫП). Метод заключается в кодировании набора условий искомой схемы булевой формулой в виде КНФ. Популярность такого подхода обусловлена развитием алгоритмов решения задачи ВЫП [2] и гибкостью кодирования различных типов схем в виде булевых формул. В [3] был проведен обзор существующих способов выражения начального условия, определенного относительно параметра сложности схемы, в терминах КНФ.

В рамках доклада предложены модификации представленных в [3] методов для формулирования задачи синтеза относительно функционалов статической и динамической активности. Критерием для оценки эффективности подходов является значение длины КНФ, кодирующей необходимые условия синтеза схемы. Доказаны оценки длины КНФ, реализуемых исходными методами, а также их модификациями. Сформулирована задача многокритериальной оптимизации относительно функционалов размера и активности схемы. Изложенные модификации позволяют получить множество Парето-оптимальных решений задачи точного синтеза комбинационных схем.

Список литературы

[1] Ernst E. A. Optimal combinational multi-level logic synthesis : Doctoral Dissertation ; The University of Michigan. Ann Arbor, Michigan, USA. 2009.

[2] Éen N. Practical SAT — A tutorial on applied satisfiability solving [Conference presentation]. Formal Methods in Computer Aided Design, FMCAD 2007. Austin, TX, USA, 2007. Retrieved 19.04.2025 from https://www.cs.utexas.edu/ hunt/fmcad/2007/presentations/practicalsat.html.

[3] SAT-based exact synthesis: encodings, topology families, and parallelism / W. Haaswijk, M. Soeken, A. Mishchenko, G. De Micheli // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2019. Vol. 39, no. 4. P. 871–884.

Информация о прочитанных пленарных докладах

Галатенко Алексей Владимирович (Москва).
О некоторых свойствах конечных квазигрупп.

Конечные квазигруппы являются перспективной платформой для реализации различных криптоалгоритмов. Для обеспечения стойкости на квазигруппы накладывается ряд требований. В. А. Артамонов предложил использовать полиномиально полные квазигруппы (то есть такие, что квазигрупповая операция и множество всех констант порождают с помощью суперпозиции все функции) без собственных подквазигрупп. В докладе приведён критерий полиномиальной полноты, проанализирована типичность свойств полиномиальной полноты и отсутствия собственных подквазигрупп, рассмотрены методы построения полиномиально полных квазигрупп без подквазигрупп, оценена сложность распознавания соответствующих свойств; в заключение обсуждён перенос результатов на случай n-арных обобщений квазигрупп — конечных n-квазигрупп.

Евдокимов Александр Андреевич (Новосибирск).
Структурированное кодирование информации и вложения дискретных метрических пространств и графов в классе отображений ограниченного искажения.

В докладе рассказано и продолжено развитие идеи кодирования информации с сохранением в кодовом пространстве структурных свойств кодируемых объектов и возможности использования структурированного кодирования информации для её быстрой и эффективной обработки. Рассмотрены вариации свойств и типов отображений, в классе которых определяются вложения, сохраняющие структурные свойства дискретных метрический пространств, графов, упорядоченных множеств. Приведены задачи, в которых сохранение структуры кодируемых объектов сочетается со свойствами параметрической отделимости и помехоустойчивости кодирования. В частности, на примере кодирования табло (целочисленной решётки) и алгоритма его вложения в гиперкуб с помощью описания задачи в терминах комбинаторики слов с запретами.

Кочергин Вадим Васильевич, **Михайлович Анна Витальевна** (Москва).
Схемная сложность булевых функций над бесконечными базисами. Точное значение сложности для одного базиса.

В докладе дан обзор известных результатов о сложности реализации булевых функций схемами над бесконечными базисами и представлен результат

авторов, установивших для произвольной булевой функции точное значение схемной сложности при реализации над бесконечным базисом, состоящем из всех монотонных булевых функций и отрицания.

Малышев Дмитрий Сергеевич, **Каймаков Кирилл Владимирович** (Нижний Новгород).
Эффективный онлайновый анализ чувствительности в задаче о максиминном пути.

В докладе предложен эффективный алгоритм для анализа чувствительности задачи о максиминном пути. Эта задача для заданных графа и пропускных способностей его ребер, а также заданных его вершин s и t заключается в поиске st-пути, минимальная пропускная способность ребер которого принимает максимальное значение. Анализ чувствительности оптимальных решений задач комбинаторной оптимизации — это поиск предельных изменений стоимостей отдельных элементов задач, при которых рассматриваемое оптимальное решение остается оптимальным. Более конкретно, верхний допуск элемента — это максимальное увеличение его стоимости, так что текущее оптимальное решение остается таковым, а нижние допуски измеряют соответствующее уменьшение стоимости. В докладе предложен алгоритм со сложностью $O(m \cdot \alpha(m,n))$ для вычисления всех допусков, где $\alpha(,)$ — обратная функция Аккермана. Для разреженных графов он улучшает ранее известную сложность $O(m+n \cdot logn)$ алгоритма Рамасвами, Орлина и Чакраварти.

Перязев Николай Алексеевич (Иркутск).
Системы неравенств в теории мультиопераций.

Мультиоперации являются обобщением понятия операции и определяются как отображение декартового произведения множества во множество всех его подмножеств. В начале доклада введены все необходимые для понимания сведения из теории мультиопераций, включая понятия метаопераций над мультиоперациями и понятие терма на множестве мультиопераций над множеством метаопераций. В докладе дан обзор результатов автора методов решения систем неравенств с неизвестными в мультиоперациях на конечных множествах. Сначала рассмотрены простейшие неравенства с одним и многими предметными неизвестными и константными параметрами. Затем рассмотрены неравенства, обе части которых задаются термально (формульно) над множеством метаопераций суперпозиции и разрешимости. Для упрощения нахождения мультиопераций, реализующих термы, применена техника пространственных булевых матриц, которая осуществляет сведение неравенств в мультиоперациях к булевому уравнению. Более того, эта техника позволяет разработать методы решения наиболее общих типов неравенств, а именно функциональных, то есть где неизвестными являются мультиоперации и па-

раметры произвольной мерности. Все методы решения различных неравенств продемонстрированы на доступных для понимания примерах.

Попков Кирилл Андреевич, **Редькин Николай Петрович**, **Романов Дмитрий Сергеевич** (Москва).
Развитие теории тестовой сложности логических схем, реализующих произвольные булевы функции.

В докладе дан обзор результатов по таким характеристикам длин минимальных тестов для логических схем, реализующих произвольные булевы функции, как оценки функций Шеннона длин тестов, оценки длин минимальных тестов для почти всех булевых функций, оценки длин минимальных тестов для каждой булевой функции.

Саргсян Ваге Гнелович (Ереван).
Наборы, k-свободные от сумм, в абелевых группах.

Пусть G — абелева группа, а $k \geqslant 2$ — целое число, и $A_1, \ldots, A_k$ — непустые подмножества G. Набор $(A_1, \ldots, A_k)$ называется k-свободным от сумм (сокращенно k-НСС), если уравнение $x_1 + \cdots + x_k = 0$ не имеет решений в наборе $(A_1, \ldots, A_k)$, где $x_1 \in A_1, \ldots, x_k \in A_k$. Семейство k-НСС в G обозначим через $S_k(G)$. Положим $\varrho_k(G) = |A_1| + \cdots + |A_k|$.

С помощью техники, связанной с преобразованиями Фурье, получена асимптотика логарифма числа k-НСС в G. Доказано, что $\log |S_k(G)| \sim \varrho_k(G)$.

Набор $(A_1, \ldots, A_k) \in S_k(G)$ назовем максимальным по мощности, если он максимальный по сумме $|A_1| + \cdots + |A_k|$, и максимальным по включению, если для любых $i \in \{1, \ldots, k\}$ и $x \in G \setminus A_i$ набор $(A_1, \ldots, A_{i-1}, A_i \cup \{x\}, A_{i+1}, \ldots, A_k) \notin SFC_k(G)$. Изучена задача о максимальном значении $\varrho_k(G)$. В частности, определено максимальное значение $\varrho_k(G)$ для циклической группы $\mathbb{Z}_n$. Получены верхняя и нижняя оценки $\varrho_k(G)$ для абелевой группы G. Описана структура максимального по мощности (по включению) набора, k-свободного от сумм, для произвольной циклической группы.

Шуплецов Михаил Сергеевич (Москва).
О статической и динамической активности схем из разных классов.

В докладе рассмотрены функционалы сложности, оценивающие энергопотребление схем из различных классов функционального и проводящего типа. Наиболее известными функционалами такого типа являются статическая активность или мощность схемы, которая позволяет оценить статическое энергопотребление схемы, и динамическая или переключательная активность схемы, которая оценивает динамическое энергопотребление схемы, связанное с переходными процессами в схеме. В докладе представлен ряд результатов отечественных и зарубежных авторов, связанных с изучением указанных функционалов сложности. В том числе их известные верхние и нижние оценки

как для схем, реализующих функции, встречающиеся в приложениях, так и для соответствующих функций Шеннона. Кроме того, будут представлены результаты о связи данных функционалов с другими функционалами сложности схем и характеристиками булевых функций, реализуемых данными схемами.

Авторский указатель