ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 59 **2023** **Вып. 1**

УДК 621.391 : 519.16

© 2023 г. **М.Н. Вялый**

О ПРОВЕРКЕ ВЫПОЛНИМОСТИ АЛГЕБРАИЧЕСКИХ ФОРМУЛ НАД ПОЛЕМ ИЗ ДВУХ ЭЛЕМЕНТОВ[1]

Построен вероятностный полиномиальный алгоритм проверки выполнимости алгебраических формул глубины 3 над полем из двух элементов, верхней операцией в которых является сложение. Алгоритм с теми же характеристиками существует для проверки равенства нулю многочлена (задача PIT), задаваемого формулами указанного вида. Однако эти задачи и алгоритмы их решения существенно отличаются. Вероятностный алгоритм для задачи PIT основан на лемме Шварца – Зиппеля, а предложенный в этой статье алгоритм проверки выполнимости основан на лемме Вельянта – Вазирани.

Ключевые слова: выполнимость булевых формул, вероятностный алгоритм, алгебраические формулы.

DOI: 10.31857/S0555292323010059, **EDN:** RMKBBO

§ 1. Введение

Задача PIT (проверка равенства нулю многочлена, задаваемого алгебраической формулой или схемой) активно изучалась в последние десятилетия. Интерес к этой задаче во многом связан с проблемой дерандомизации. Для задачи PIT существуют вероятностные полиномиальные алгоритмы, но неизвестны детерминированные полиномиальные алгоритмы. Известно, что построение таких алгоритмов приводит к хорошим нижним оценкам в алгебраической сложности [1]. Для многих частных видов формул были найдены полиномиальные детерминированные алгоритмы (см. обзоры [2,3]). Однако уже случай формул глубины 3 оказывается очень трудным. Как было доказано в [4], для любой схемы над полем $\mathbb{C}$ комплексных чисел существует эквивалентная формула глубины 3 субэкспоненциального размера.

С точки зрения теории вычислительной сложности естественно рассмотреть другую задачу: выполнимости алгебраической формулы (или схемы). Над бесконечным или достаточно большим (по сравнению со степенью формулы) конечным полем разницы между этими задачами нет. Однако для малых конечных полей эти задачи различаются, и нет прямой сводимости одной из них к другой.

В этой короткой заметке мы обсуждаем задачу проверки выполнимости алгебраических формул над наименьшим конечным полем $\mathbb{F}_2$ из двух элементов. Эта задача оказывается NP-полной уже для формул глубины 3. Однако, как это принято в исследованиях задачи PIT, если различать формулы по расположению функциональных элементов в слоях, ситуация оказывается более интересной. NP-полнота имеет место для формул вида ΠΣΠ (верхний уровень – умножение). При этом для таких формул задача PIT решается за детерминированное полиномиальное время

[1] Работа выполнена в рамках Программы фундаментальных исследований НИУ ВШЭ, а также частично финансировалась в рамках госзадания 0063-2019-0003.

(см. замечание 1 в § 2). Для второго класса схем глубины 3 – формул вида ΣΠΣ (верхний уровень – сложение) – существование детерминированного полиномиального алгоритма для задачи PIT проблематично в силу упомянутого выше результата о представлении произвольных схем ΣΠΣ-формулами (в [4] используются именно они). Проверка выполнимости ΣΠΣ-формулы над $\mathbb{F}_2$ также выглядит нетривиальной задачей.

Основной результат этой статьи – вероятностный полиномиальный алгоритм проверки выполнимости для ΣΠΣ-формул над $\mathbb{F}_2$. Интересно отметить, что основан этот алгоритм не на лемме Шварца – Зиппеля (см., например, [2]), как вероятностные алгоритмы для задачи PIT, а на лемме Вельянта – Вазирани, изолирующей одну из единиц булевой формулы (см. [5]).

§ 2. Задача выполнимости для алгебраических формул

Здесь мы рассматриваем алгебраические формулы малой глубины над полем $\mathbb{F}_2$ из двух элементов. Входная степень функциональных элементов – сложения и умножения – предполагается неограниченной.

Формулы глубины 1 бывают двух видов: Π-формулы, т.е. произведения переменных, и Σ-формулы, которые задают линейные функции.

Среди формул глубины 2 мы выделяем ΣΠ-формулы (суммы произведений переменных) и ΠΣ-формулы (произведения линейных функций).

Аналогично, ΠΣΠ-формулы – это формулы глубины 3, которые являются произведениями ΣΠ-формул, а ΣΠΣ-формулы – это формулы глубины 3, которые являются суммами ΠΣ-формул.

Каждая алгебраическая формула над $\mathbb{F}_2$ задает функцию $\mathbb{F}_2^n \to \mathbb{F}_2$, где n – количество переменных. Если эта функция не равна тождественно нулю, формула называется выполнимой. Задачу проверки выполнимости алгебраической формулы обозначим через AlgSAT. Она лежит в классе NP по очевидным причинам: вычисление значения формулы возможно за полиномиальное время.

Проверка выполнимости Σ- или Π-формул тривиальна. Почти столь же проста проверка выполнимости ΣΠ-формулы: нужно сократить одинаковые слагаемые (в поле $\mathbb{F}_2$ выполняется тождество $x + x = 0$), и формула выполнима, если и только если результат сокращения содержит хотя бы один ненулевой моном (многочлен Жегалкина для булевой функции единствен).

Проверка выполнимости ΠΣ-формул также лежит в классе P. Пусть

$$\varphi(x_1, \ldots, x_n) = \prod_{i=1}^{s} \Big(a_{i0} + \sum_{j=1}^{n} a_{ij} x_j \Big). \tag{1}$$

Выполнимость φ равносильна существованию решения системы линейных уравнений

$$\sum_{j=1}^{n} a_{ij} x_j = 1 + a_{i0}, \quad 1 \leqslant i \leqslant s, \tag{2}$$

что проверяется стандартными средствами линейной алгебры. Заметим, что множество тех $x \in \mathbb{F}_2^n$, для которых $\varphi(x) = 1$, образует аффинное подпространство координатного пространства $\mathbb{F}_2^n$.

Обозначим через ΠΣΠ-SAT задачу проверки выполнимости ΠΣΠ-формулы. Это уже трудная задача.

Предложение 1. *Задача* ΠΣΠ-SAT NP-*полна.*

Доказательство. Построим полиномиальную сводимость стандартной NP-полной задачи 3SAT (выполнимость 3-КНФ) к ΠΣΠ-SAT. Для этого заметим, что каждый дизъюнкт в КНФ представляется в виде многочлена Жегалкина, и этот многочлен имеет размер $O(1)$, так как в дизъюнкт входит только три литерала. Таким образом, по 3-КНФ получается эквивалентная ΠΣΠ-формула. ▲

Замечание 1. Задача PIT для ΠΣΠ-формул проще общего случая, для нее есть детерминированный полиномиальный алгоритм. В кольце многочленов над любым полем нет делителей нуля. Поэтому ΠΣΠ-формула задает нулевой многочлен тогда и только тогда, когда один из множителей верхнего произведения задает нулевой многочлен. Поэтому достаточно в каждом из этих множителей сократить подобные и проверить, остается ли после сокращения хотя бы один ненулевой моном.

Из предложения 1 следует NP-полнота проверки выполнимости любых формул глубины $O(1)$, за исключением случая ΣΠΣ-формул. Именно этот оставшийся случай и является главным предметом изучения в данной статье. Обозначим через ΣΠΣ-SAT задачу проверки выполнимости таких формул. Какова ее алгоритмическая сложность? В следующем параграфе мы приводим вероятностный полиномиальный алгоритм для задачи ΣΠΣ-SAT. Существование такого алгоритма делает крайне сомнительной полноту этой задачи в классе NP.

У задачи ΣΠΣ-SAT есть естественная геометрическая интерпретация. ΣΠΣ-формула обращается в единицу в точности в тех точках, в которых нечетное количество ее ΠΣ слагаемых обращается в единицу. Единицы ΠΣ-формулы – это аффинное подпространство, и любое аффинное подпространство представляется как множество единиц некоторой ΠΣ-формулы. Выбор такой формулы не однозначен. Один из возможных вариантов: если подпространство задано как множество решений системы линейных уравнений (2), то его точки совпадают с множеством единиц формулы (1).

Таким образом, равносильная формулировка задачи ΣΠΣ-SAT состоит в том, что дан набор аффинных подпространств, и нужно узнать, существует ли точка, покрытая нечетным количеством подпространств. Отсюда также следует, что общая задача ΣΠΣ-SAT сводится к случаю ΣΠΣ-формул, в которых каждое слагаемое имеет степень не более n (количество переменных в наших обозначениях).

§ 3. Вероятностный алгоритм для задачи ΣΠΣ-SAT

Напомним, что класс RP (вероятностные вычисления с односторонней ошибкой) состоит из тех языков L, для которых существует вероятностный полиномиальный алгоритм A, выдающий результаты из множества $\{0,1\}$, такой что

$$x \in L \iff \Pr[A(x)=1] \geqslant \frac{1}{2},$$
$$x \notin L \iff \Pr[A(x)=1] = 0.$$

Теорема 1. *Задача* ΣΠΣ-SAT *принадлежит классу* RP.

Доказательство. Пусть ΣΠΣ-формула имеет вид

$$\varphi(x) = \sum_{i=1}^{s} p_i(x), \quad \text{где } p_i(x) \text{ – ΠΣ-формула.} \tag{3}$$

Обозначим через M матрицу размера $2^n \times s$, матричные элементы которой имеют вид $M_{xi} = p_i(x)$. Условие выполнимости формулы равносильно условию $M\mathbb{1} \neq 0$, так как $\varphi(x) = (M\mathbb{1})_x$. Здесь и далее через $\mathbb{1}$ обозначается вектор-столбец размерности s,

все координаты которого равны 1, а через 0 – любой нулевой вектор (размерность которого будет ясна из контекста).

В координатном пространстве векторов-строк $(\mathbb{F}_2^n)^*$ выберем случайный базис $b_1, \ldots, b_n$ по равномерному распределению. Для каждого $0 \leqslant i \leqslant n$ определим вектор-строку h_i размера 2^n, координаты которого индексированы векторами-столбцами $x \in \mathbb{F}_2^n$. Значения координат задаются следующим правилом: $(h_i)_x = 1$ тогда и только тогда, когда $b_1 x = b_2 x = \ldots = b_i x = 0$, если $i > 0$, и $(h_0)_x = 1$ для всех x. Вычислим $h_i M \mathbb{1}$, находя сначала $h_i M$ и умножая затем полученную строку размера s на вектор-столбец $\mathbb{1}$.

Если $h_i M \mathbb{1} = 1$, то выдаем ответ "$\varphi(x)$ выполнима". Если при всех проверках $h_i M \mathbb{1} = 0$, то выдаем ответ "$\varphi(x)$ невыполнима".

Оценим время работы алгоритма. В худшем случае он выполняет $n + 1$ вычислений $h_i M \mathbb{1}$. Вычисление $h_i M$ возможно за полиномиальное время, так как $(h_i M)_k$ равно четности количества точек в аффинном подпространстве

$$L_k = \{x : \, p_k(x) = 1\} \cap \{x : \, b_1 x = b_2 x = \ldots = b_i x = 0\}.$$

Количество точек в подпространстве размерности d равно 2^d, размерность L_k находится за полиномиальное время стандартными алгоритмами линейной алгебры. Заметим также, что s ограничено длиной входа (описание формулы $\varphi(x)$).

Докажем корректность алгоритма. Если $M\mathbb{1} = 0$, то $hM\mathbb{1} = 0$ для любого вектора-строки h. Поэтому вероятность ошибки алгоритма в случае невыполнимой формулы $\varphi(x)$ равна 0.

Предположим, что формула выполнима, т.е. $M\mathbb{1} \neq 0$. Обозначим через S множество тех $x \in \mathbb{F}_2^n$, для которых $(M\mathbb{1})_x = 1$ (другими словами, $\varphi(x) = 1$). Поскольку $\varphi(x)$ выполнима, $S \neq \varnothing$. Лемма Вельянта – Вазирани [5, теорема 2.4] утверждает, что в этом случае с вероятностью не менее $1/2$ для некоторого $0 \leqslant i \leqslant n$ выполняется

$$|S_i| = 1, \quad \text{где} \quad S_i = S \cap \{x : \, b_1 x = b_2 x = \ldots = b_i x = 0\}. \tag{4}$$

Заметим, что $h_i M \mathbb{1}$ равно четности $|S_i|$. Поэтому из (4) следует $h_i M \mathbb{1} = 1$. Значит, алгоритм с вероятностью не менее $1/2$ выдает ответ "$\varphi(x)$ выполнима". Таким образом, ΣΠΣ-SAT $\in$ RP. ▲

§ 4. О дерандомизации основной теоремы

Существует ли детерминированный полиномиальный алгоритм для задачи ΣΠΣ-SAT? Вопрос требует дальнейшего изучения, здесь мы ограничимся несколькими замечаниями.

Возможность дерандомизации леммы Вельянта – Вазирани представляется маловероятной (см. [6, 7]). Прямолинейная дерандомизация алгоритма из теоремы 1 заведомо невозможна по соображениям размерности (линейное отображение пространства размерности 2^n в пространство полиномиальной размерности всегда имеет ненулевое ядро). Заметим, что для задачи PIT возможность дерандомизации путем предъявления полиномиального по размеру количества точек, в которых достаточно вычислить значения формулы, считается вполне возможной, и существуют гипотезы о возможном виде таких множеств. Одна из таких гипотез предложена в [1]. Заметим также, что из τ-гипотезы Куарана [8] следует возможность дерандомизации указанного вида для формул над $\mathbb{R}$.

Если φ – ΣΠΣ-формула, то и $1 + \varphi$ также ΣΠΣ-формула. Поэтому задача ΣΠΣ-SAT также эквивалентна проверке того, что хотя бы одна точка покрыта четным количеством аффинных подпространств из заданного набора. Заметим, что условие четности существенно слабее других условий на кратность покрытия. В [9] до-

казано, что проверка покрытия $\mathbb{F}_2^n$ заданным набором аффинных подпространств co-NP-полна. С учетом теоремы 1 это означает, что есть значительная разница в трудности проверки существования точки, покрытой четным количеством подпространств (лежит в RP), и существования точки, покрытой нулем подпространств (NP-полна).

Для вероятностного алгоритма из теоремы 1 существенна простота множеств единиц ΠΣ-формул. Действительно, теорема Вельянта – Вазирани [5] предоставляет вероятностную сводимость NP к классу ⊕P, полной задачей в которой является ⊕SAT: проверка того, что данная булева формула выполняется в нечетном количестве точек. Алгоритм из теоремы 1 неприменим в случае произвольных булевых формул, потому что необходимый для вычисления $h_i M$ подсчет количества единиц булевой формулы является #P-полной задачей.

Однако все эти замечания не исключают, конечно, существования детерминированного полиномиального алгоритма для ΣΠΣ-SAT.

В заключение рассмотрим несколько частных случаев ΣΠΣ-формул, для которых возможна проверка выполнимости за детерминированное полиномиальное время.

Самый простой пример – схемы ограниченной степени. Если все слагаемые в ΣΠΣ-формуле имеют степень $O(1)$, то раскрытием скобок и приведением подобных за полиномиальное время можно преобразовать ΣΠΣ-формулу в равносильную ΣΠ-формулу. Заметим, что в этом случае соответствующие аффинные подпространства имеют ограниченную коразмерность, так как количество уравнений в системе (2) для каждого слагаемого имеет величину $O(1)$.

Построить нетривиальные примеры формул, которые тождественно равны нулю на $\mathbb{F}_2^n$, можно следующим образом. Пусть $u_1, \ldots, u_t$ – набор точек в $\mathbb{F}_2^n$. Возьмем любой граф G на t вершинах, степень каждой из которых четна, и построим набор аффинных подпространств

$$A_{ij} = \{u_i, u_j\}, \quad \{i,j\} \in E(G)$$

(любая пара точек в $\mathbb{F}_2^n$ образует аффинное подпространство). Каждая точка $\mathbb{F}_2^n$ входит в четное (0 или 2) количество таких подпространств. Соответствующие формулы имеют вид

$$\varphi_G(x) = \sum_{\{i,j\}\in E(G)} q_{ij}(x), \quad \text{где} \quad q_{ij} = \prod_{k=1}^{n-1} \ell_k^{\{i,j\}}(x),$$

а множители $\ell_k^{\{i,j\}}(x)$ определяются через векторы $a_1^{\{i,j\}}, \ldots, a_{n-1}^{\{i,j\}}$ базиса ортогонального дополнения к $\mathbb{F}_2(u_j - u_i)$ следующим образом:

$$\ell_k^{\{i,j\}}(x) = 1 - \sum_{m=1}^{n} a_{km}^{\{i,j\}} u_{im} + \sum_{m=1}^{n} a_{km}^{\{i,j\}} x_m$$

(хотя $-1 = +1$ в $\mathbb{F}_2$, знаки расставлены для удобства чтения).

Обобщая этот пример, получаем другое семейство формул, для проверки выполнимости которых существует детерминированный полиномиальный алгоритм. Это такие формулы, в которых размерность аффинного пространства, отвечающего каждому слагаемому формулы, равна $d = O(1)$. Такое подпространство содержит лишь $2^d = O(1)$ точек $\mathbb{F}_2^n$, так что формулы из данного семейства принимают значение 1 лишь на некотором подмножестве объединения всех указанных подпространств. Мощность этого объединения не превосходит $2^d s$, где s – количество слагаемых верхнего уровня (как в (3)). Найти размерность решения системы (2) линейных уравнений и список всех ее решений можно стандартными методами линейной

алгебры, в данном случае это выполнимо за время, полиномиально ограниченное длиной входа (размером записи формулы).

Более сложный алгоритм проверки выполнимости использует формулу для числа единиц суммы по модулю 2 булевых функций, аналогичную формуле включений и исключений. Пусть $f_1, \ldots, f_k$ – функции $\{0,1\}^n \to \{0,1\}$. Сейчас мы рассматриваем 0 и 1 как целые числа. Обозначим

$$f = f_1 \oplus f_2 \oplus \ldots \oplus f_k$$

(складываем значения функций по модулю 2). Тогда

$$(1-2f)(x) = \prod_{i=1}^{k}(1-2f_i(x)),$$

и если обозначить через $N(f)$ количество единиц функции f, то получаем

$$N(f) = \sum_{\varnothing \neq S \subset \{0,1\}^k} (-1)^{|S|+1} 2^{|S|-1} N\Big(\prod_{i \in S} f_i(x)\Big) \tag{5}$$

(см. аналогичное вычисление в [10]). Равенство (5) для формулы (3) включает $2^k - 1$ слагаемых. Однако часть этих слагаемых может равняться нулю. Например, пусть никакие $q = O(1)$ слагаемых в (3) не совместны, т.е. соответствующие аффинные подпространства не пересекаются. Тогда количество ненулевых слагаемых в (5) имеет величину $O(n^q)$, и вычисление по этой формуле занимает полиномиальное время (напомним, что подсчет числа точек в аффинном подпространстве возможен за полиномиальное время).

СПИСОК ЛИТЕРАТУРЫ

1. *Agrawal M.* Proving Lower Bounds via Pseudo-random Generators // FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science (Proc. 25th Int. Conf. Hyderabad, India. Dec. 15–18, 2005). Lect. Notes Comput. Sci. V. 3821. Berlin: Springer, 2005. P. 92–105. https://doi.org/10.1007/11590156_6
2. *Saxena N.* Progress on Polynomial Identity Testing // Bull. Eur. Assoc. Theor. Comput. Sci. 2009. № 90. P. 49–79.
3. *Saxena N.* Progress on Polynomial Identity Testing-II // Perspectives in Computational Complexity. Progr. Comput. Sci. Appl. Logic. V. 26. Cham: Birkhäuser, 2014. P. 131–146. https://doi.org/10.1007/978-3-319-05446-9_7
4. *Gupta A., Kamath P., Kayal N., Saptharishi R.* Arithmetic Circuits: A Chasm at Depth 3 // SIAM J. Comput. 2016. V. 45. № 3. P. 1064–1079. https://doi.org/10.1137/140957123
5. *Valiant L.G., Vazirani V.V.* NP Is as Easy as Detecting Unique Solutions // Theor. Comput. Sci. 1986. V. 47. № 1. P. 85–93. https://doi.org/10.1016/0304-3975(86)90135-0
6. *Hemaspaandra L.A., Naik A.V., Ogihara M., Selman A.L.* Computing Solutions Uniquely Collapses the Polynomial Hierarchy // SIAM J. Comput. 1996. V. 25. № 4. P. 697–708. https://doi.org/10.1137/S0097539794268315
7. *Dell H., Kabanets, V., van Melkebeek, D., Watanabe O.* Is Valiant–Vazirani's Isolation Probability Improvable? // Comput. Complexity. 2013. V. 22. № 2. P. 345–383. https://doi.org/10.1007/s00037-013-0059-7
8. *Grenet B., Koiran P., Portier N., Strozecki Y.* The Limited Power of Powering: Polynomial Identity Testing and a Depth-Four Lower Bound for the Permanent // Proc. 31st IARCS Annu. Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011). Mumbai, India. Dec. 12–14, 2011. Leibniz Int. Proc. Inform. (LIPIcs). V. 13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany: Dagstuhl Publ., 2011. P. 127–139. https://doi.org/10.4230/LIPIcs.FSTTCS.2011.127

9. *Arvind V., Guruswami V.* CNF Satisfiability in a Subspace and Related Problems // Algorithmica. 2022. V. 84. № 11. P. 3276–3299. https://doi.org/10.1007/s00453-022-00958-4
10. *Леонтьев В.К., Морено О.* О нулях булевых полиномов // Ж. вычисл. матем. и матем. физ. 1998. Т. 38. № 9. С. 1608–1615. https://www.mathnet.ru/rus/zvmmf1832

Вялый Михаил Николаевич
Федеральный исследовательский центр
"Информатика и управление" РАН, Москва
Национальный исследовательский университет
"Высшая школа экономики", Москва
Московский физико-технический институт
(государственный университет), Москва
vyalyi@gmail.com

Поступила в редакцию
18.12.2022
После доработки
12.02.2023
Принята к публикации
13.02.2023