S©MR

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ
Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru
ISSN 1813-3304

Том 23, № 1, стр. 244–256 (2026) УДК 510.652
https://doi.org/10.33048/semi.2026.23.015 MSC 11U99

О ГЕНЕРИЧЕСКОЙ СЛОЖНОСТИ ПРОБЛЕМЫ РЕШЕНИЯ УРАВНЕНИЙ В ФОРМЕ СКОЛЕМА НАД МОНОИДАМИ

А.Н. Рыбалов, А.Н. Шевляков

Представлено С.В. Судоплатовым

Abstract: We study the generic complexity of the problem of solving systems of equations in the Skolem form over finitely generated monoids. Three variants of this problem are considered. The first variant is the problem of recognizing the solvability of dense systems of equations, where the number of equations is bounded cubically in the number of variables. For this problem, we prove its strongly generic decidability in polynomial time. The second variant is a similar problem for sparse systems of equations, where the number of equations is bounded linearly in the number of variables. For this problem, we prove that, given its worst-case intractability, no strongly generic polynomial algorithm exists. This means that for any generic polynomial algorithm, there exists an efficient method for randomly generating inputs on which this algorithm cannot solve the problem. Finally, the third variant is the problem of searching solutions to systems of equations. For this problem, the input is a system of equations for which a solution is known to exist and at least one solution must be found. Search problems find application in cryptography, where a solution is always known to exist and this solution must be found. For the

Rybalov, A.N., Shevlyakov, A.N., On generic complexity of the problem of solvability of equations in the Skolem form over monoids.
© 2026 Рыбалов А.Н..
© 2026 Шевляков А.Н..
Работа поддержана грантом Российского Научного Фонда №25-11-20023.
Поступила 12 января 2026 г., опубликована 25 марта 2026 г.

search problem, it is proven that, given its intractability, there exists a subproblem of this problem for which there is no generic polynomial algorithm.

Keywords: generic complexity, monoids, solvability of equations.

1 Введение

Решение уравнений является классической темой исследований в различных областях математики в течение тысяч лет. Классическая алгебраическая геометрия изучает множества решений алгебраических уравнений над полями вещественных и комплексных чисел. В рамках диофантовой геометрии и диофантова анализа изучаются решения алгебраических уравнений над целыми и рациональными числами. В XX веке большую роль начали играть вычислительные аспекты этих теорий. Изучение алгоритмических проблем, связанных с определением наличия решения у систем уравнений, а также с нахождением и описанием множества решений, является темой многочисленных теоретических и практических исследований.

В последние десятилетия внимание ученых перемещается на неклассические области, такие как группы [1, 6, 7], полугруппы [4], графы [8], частичные порядки [5], булевы алгебры [9]. Потребность решения уравнений в этих системах возникает при рассмотрении различных практических проблем информатики, криптографии, теории языков программирования. Например, свободные полугруппы являются базисом для описания важнейших классов формальных языков и грамматик: регулярных, контекстно свободных. Часто при этом изучаемый формальный язык задается некоторым набором уравнений, множество решений которых дает нужный нам язык. К необходимости решения уравнений над графами приводят задачи проверки вложимости (совместимости) одной коммуникационной сети в другую сеть.

Как правило, проблема решения систем уравнений над различными алгебраическими системами является либо неразрешимой, либо имеет большую вычислительную сложность. Даже над конечными алгебраическими системами эта проблема часто оказывается NP-полной. Это означает, что, при условии $\mathrm{P} \neq \mathrm{NP}$, для нее не существует полиномиальных алгоритмов. Поэтому актуальным является изучение генерической сложности [3] данных проблем. В рамках генерического подхода алгоритмическая проблема рассматривается не на всём множестве входов, а на некотором подмножестве «почти всех» входов. С одной стороны, положительные результаты о возможности эффективного решения каких-либо трудных задач для почти всех входов полезны для практики. С другой стороны, негативные результаты о генерической трудности некоторых проблем дают надежду на возможное их использование в криптографии, где как раз важно чтобы проблема взлома криптосистемы была трудной для почти всех входов.

В данной статье изучается генерическая сложность проблемы решения систем уравнений в форме Сколема в конечно порожденных моноидах. Уравнения в форме Сколема имеют вид $\alpha = \beta\gamma$, где α, β, γ – переменные или порождающие. Легко показать, что любая система уравнений над моноидом эквивалентна некоторой системе уравнений в форме Сколема. При этом число переменных и количество уравнений в новой системе по сравнению с исходной увеличивается не более чем полиномиально. Кроме того, сама процедура построения системы в форме Сколема является полиномиальной по времени.

Рассматриваются три варианта этой проблемы. Первый вариант – проблема распознавания разрешимости *плотных* систем уравнений, когда число уравнений ограничено кубически от числа переменных. Для этой проблемы доказывается ее сильно генерическая разрешимость за полиномиальное время. Второй вариант проблемы – аналогичная проблема для *разреженных* систем уравнений, когда число уравнений ограничено линейно от числа переменных. Для этой проблемы доказывается, что, при условии ее трудноразрешимости в худшем случае, не существует сильно генерического полиномиального алгоритма. Это означает, что для любого генерического полиномиального алгоритма существует эффективный метод случайной генерации входов, на которых этот алгоритм не может решить рассматриваемую проблему. Наконец, третий вариант – проблема поиска решения систем уравнений. Для этой проблемы входом является система уравнений, для которой заведомо существует решение, нужно найти хотя бы одно её решение. Проблемы поиска часто находят применения в криптографии, где всегда известно, что решение есть и надо найти это решение. Для проблемы поиска решения доказывается, что, при условии ее трудноразрешимости, существует подпроблема этой проблемы, для которой нет полиномиального генерического алгоритма.

2 Предварительные сведения

Пусть I – некоторое множество входов, а I_n — подмножество входов размера n. Для подмножества $S_n \subseteq I$ определим последовательность

$$\rho_n(S) = \frac{|S_n|}{|I_n|}, n = 1, 2, 3, \ldots,$$

где $S_n = S \cap I_n$ — множество входов из S размера n. *Асимптотической плотностью* S назовем предел

$$\rho(S) = \lim_{n\to\infty} \rho_n(S).$$

Множество S называется *пренебрежимым*, если его асимптотическая плотность $\rho(S) = 0$. Следуя [3], назовём множество S *сильно пренебрежимым*, если последовательность $\rho_n(S)$ экспоненциально быстро сходится к 0, т. е. существуют константы σ, $0 < \sigma < 1$, и $C > 0$, такие, что

для любого n

$$\rho_n(S) < C\sigma^n.$$

Теперь S называется *сильно генерическим*, если его дополнение $I \setminus S$ сильно пренебрежимо.

Алгоритм $\mathcal{A}$ с множеством входов I и множеством выходов $J \cup \{?\} (? \notin J)$ называется *(сильно) генерическим*, если

(1) $\mathcal{A}$ останавливается на всех входах из I;
(2) множество $\{x \in I : \mathcal{A}(x) = ?\}$ является (сильно) пренебрежимым.

Здесь символ «?» обозначает неопределенный ответ. Генерический алгоритм $\mathcal{A}$ *вычисляет* функцию $f : I \to \mathbb{N}$, если для всех $x \in I$ выполнено

$$(\mathcal{A}(x) \neq ?) \Rightarrow (f(x) = \mathcal{A}(x)).$$

Проблема распознавания множества $A \subseteq I$ *(сильно) генерически разрешима за полиномиальное время*, если существует полиномиальный (сильно) генерический алгоритм для вычисления характеристической функции множества A.

Имеется существенное различие между генерически алгоритмами и сильно генерически алгоритмами. Допустим, имеется проблема S, разрешимая на некотором разрешимом за полиномиальное время генерическом множестве G, для которого

$$\frac{|G \cap I_n|}{|I_n|} = \frac{n-1}{n}.$$

Таким образом G – генерическое, но не сильно генерическое множество. Теперь хоть и проблема S разрешима для почти всех входов, тем не менее, есть эффективный способ получить «плохой» вход, на котором генерический алгоритм не работает. Полиномиальный алгоритм для генерации плохих входов следующий.

(1) Сгенерировать равномерно случайный вход x размера n.
(2) Если $x \in G$, повторить шаг 1, иначе закончить.

Действительно, вероятность получить только хорошие входы за n^2 раундов:

$$\left(\frac{n-1}{n}\right)^{n^2} = \left(\left(1 - \frac{1}{n}\right)^n\right)^n \to e^{-n}.$$

Поэтому с вероятностью, очень близкой к 1, будет получен плохой вход. С другой стороны, легко видеть, что если проблема разрешима на сильно генерическом множестве, то такой простой алгоритм генерации потребует экспоненциального числа раундов и будет неэффективным. Для приложений к криптографии, это означает, что просто генерическая легкоразрешимость проблемы не делает эту проблему бесполезной для создания на ее основе криптосистемы, так как для нее существует эффективная процедура генерации трудных входов. В то же время, сильно генерически легкоразрешимые проблемы в этом смысле бесполезны для криптографии.

Напомним также некоторые понятия классической теории сложности вычислений.

Вероятностная машина Тьюринга – это машина Тьюринга, в программе которой допускаются пары недетерминированных правил, которые одновременно применимы в данной ситуации. В процессе работы такой машины с вероятностью $1/2$ выбирается первое правило и с вероятностью $1/2$ второе. Время работы $t_M(x, \tau)$ вероятностной машины Тьюринга на входе x зависит от вычислительного пути (последовательности выполненных команд) τ. Вероятностная машина Тьюринга M называется *полиномиальной*, если существует полином $p(n)$ такой, что для любого x и для любого вычислительного пути τ машины M на x имеет место $t_M(x, \tau) < p(size(x))$.

Обозначим через $P(M(x) = y)$ вероятность того, что машина M на входе x выдает ответ y. Вероятностная машина M *вычисляет* функцию $f : I \to J$, если для любого $x \in I$ имеет место

$$\Big(f(x) = y\Big) \Rightarrow P(M(x) = y) > 2/3.$$

Множество принадлежит классу **BPP**, если существует полиномиальная вероятностная машина, вычисляющая характеристическую функцию этого множества.

Вероятностные машины Тьюринга формализуют понятие алгоритма, использующего генератор случайных чисел. Большинство специалистов по теоретической информатике сейчас считает, что любой полиномиальный вероятностный алгоритм можно эффективно дерандомизировать, т.е. построить полиномиальный детерминированный алгоритм, решающий ту же задачу. В частности, считается, что **BPP** = **P**. Хотя этот факт еще не доказан, имеются серьезные результаты в пользу него [2].

3 Плотные системы уравнений

Пусть $\mathfrak{M}$ – конечно порожденный моноид с множеством порождающих $A = \{a_1, \ldots, a_m\}$. Любую систему уравнений над $\mathfrak{M}$ можно преобразовать в эквивалентную ей систему в *форме Сколема*, в которой каждое уравнение имеет вид $\alpha = \beta\gamma$, где α, β, γ – переменные или порождающие. В данном разделе будем представлять системы уравнений над $\mathfrak{M}$ следующим образом. Во-первых, зафиксируем переменные системы $x_1, \ldots, x_n$. Для единообразия обозначим набор порождающих и переменных в порядке возрастания индексов $a_1, \ldots, a_m, x_1, \ldots, x_n$ через $\alpha_1, \ldots, \alpha_{n+m}$. Далее рассмотрим так называемый *куб включения* – это куб с $n + m$ позициями по каждой размерности. На месте с координатами (i, j, k) записываем 1, если в системе есть уравнение $\alpha_i = \alpha_j\alpha_k$, и 0 если нет. Будем отождествлять систему уравнений и ее куб включения.

Заметим, что такой способ представления систем уравнений аналогичен классическому представлению графов с помощью матриц смежности. У нас же куб включения задает гиперграф, в котором тройки вершин соединяются гиперребрами, соответствующими уравнениям. При случайной равновероятной генерации кубов включения размера $n+m$ типичными системами будут системы с числом уравнений больше $C(n+m)^3$, с константой $C > 0$. Это следует из центральной предельной теоремы для равномерного распределения (схема Бернулли с вероятностью успеха $p = \frac{1}{2}$ для $(n+m)^3$ независимых испытаний). Таким образом, соответствующие типичные гиперграфы получаются достаточно «плотными» с большим числом гиперребер, лишь в константу раз отличающимся от максимального $(n+m)^3$ числа гиперребер. Поэтому естественно называть системы уравнений в этой модели *плотными*.

Число переменных n плотной системы уравнений назовем *размером системы*. В этом разделе будем рассматривать только плотные системы уравнений. Обозначим через $\mathcal{D}$ множество плотных систем уравнений над $\mathfrak{M}$, представленных таким образом.

Лемма 1. *Число плотных систем размера n над $\mathfrak{M}$ есть*

$$|\mathcal{D}_n| = 2^{(n+m)^3}.$$

Доказательство. В каждой из $(n+m)^3$ позиций куба включения может стоять либо 0, либо 1. Всего получается $2^{(n+m)^3}$ вариантов. □

Будем называть систему уравнений *нетривиальной*, если в ней отсутствуют уравнения вида $\alpha = \beta\gamma$, где α, β, γ – только порождающие. Назовем моноид $\mathfrak{M}$ *нетривиальным*, если существует нетривиальная система уравнений, которая не имеет решения над $\mathfrak{M}$. В противном случае, моноид $\mathfrak{M}$ *тривиальный*. Очевидно, что для тривиальных моноидов проблема распознавания разрешимости систем уравнений разрешима за полиномиальное время.

Теорема 1. *Проблема распознавания разрешимости плотных систем уравнений над конечно порожденным нетривиальным моноидом $\mathfrak{M}$ сильно генерически разрешима за полиномиальное время.*

Доказательство. Пусть S' – какая-то фиксированная нетривиальная система уравнений из s уравнений размера t, неразрешимая над $\mathfrak{M}$. Таким образом, в системе S' участвуют t переменных. Алгоритм для распознавания разрешимости систем уравнений над $\mathfrak{M}$ будет работать на системе S размера n следующим образом.

(1) Ищет в системе S подсистему, эквивалентную S'. Это делается следующим образом. Перебираем все выборки по t переменных из n переменных системы S. Для каждой выборки ищем в S все уравнения из системы S' с учетом замены переменных S' соответствующими переменными из выборки.
(2) Если такая подсистема нашлась, то выдает ответ «НЕТ».

(3) Если нет, выдает ответ «?».

Для доказательства полиномиальности построенного алгоритма заметим, что на шаге 1 число проверяемых выборок $C_n^t = O(n^t)$ полиномиально от размера входа n, и проверка каждой выборки делается за полиномиальное от n время.

Для доказательства сильной генеричности этого алгоритма покажем, что множество систем уравнений, не содержащих подсистемы, эквивалентной S' (обозначим это множество A), является сильно пренебрежимым. Рассмотрим множество систем B, в которых на переменных $\{x_1, \ldots, x_n\}$ запрещена подсистема S' на переменных $\{x_1, \ldots, x_t\}$, на переменных $\{x_{t+1}, \ldots, x_{2t}\}$, ..., на переменных $\{x_{t([n/t]-1)+1}, \ldots, x_{t[n/t]}\}$. Через $[a]$ здесь обозначена целая часть числа a. Так как для систем из B запретов меньше, чем для систем из A, то $A \subseteq B$.

Можно подсчитать, что

$$|B_n| = 2^{(n+m)^3 - s[n/t]} \Big(2^s - 1\Big)^{[n/t]}.$$

Это следует из того, что в кубе включения B «запрещены» $[n/t]$ расстановок единиц на множествах мест размера s, соответствующих системе S' на переменных $\{x_1, \ldots, x_t\}$, на переменных $\{x_{t+1}, \ldots, x_{2t}\}$, ..., на переменных $\{x_{t([n/t]-1)+1}, \ldots, x_{t[n/t]}\}$. Заметим, что эти множества мест не пересекаются. Остальные места в кубе можно заполнять нулями и единицами произвольно.

Теперь

$$\rho(B) = \lim_{n\to\infty} \frac{|B_n|}{|\mathcal{D}_n|} = \lim_{n\to\infty} \frac{2^{(n+m)^3 - s[n/t]} \Big(2^s - 1\Big)^{[n/t]}}{2^{(n+m)^3}} =$$

$$= \lim_{n\to\infty} \frac{\Big(2^s - 1\Big)^{[n/t]}}{2^{s[n/t]}} = \lim_{n\to\infty} \Big(1 - \frac{1}{2^s}\Big)^{[n/t]} = 0.$$

Это доказывает, что множество B является сильно пренебрежимым, а, значит, и его подмножество A также сильно пренебрежимо. □

4 Разреженные системы уравнений

В этом разделе будем также рассматривать системы уравнений в форме Сколема над моноидом $\mathfrak{M}$. Но под *размером системы* будем подразумевать число уравнений в ней. Также системы будут предполагаться *нормализованными*. Это означает, что в k-м уравнении системы могут встречаться только порождающие и переменные x_i, где $i \leq 3k$. Это соответствует естественной нумерации переменных в системе: в первом уравнении переменные естественно обозначать x_1, x_2, x_3, но не сразу x_{100}, во втором – либо «старые» переменные из первого уравнения, либо x_4, x_5, x_6, и т.д. Очевидно, что любую систему в форме Сколема можно нормализовать с помощью подходящей перенумерации переменных.

Можно показать, что при случайной генерации таких систем размера n, когда для каждого последующего уравнения переменные равновероятно выбираются из предыдущих и трех новых, типичными будут системы, в которых число переменных $m > Cn$, с некоторой константой $C > 0$. Другими словами число уравнений n меньше $\frac{m}{C}$, где m – число переменных. Таким образом, сравнивая с плотными системами из предыдущего раздела, такие системы можно назвать *разреженными*.

В этом разделе будем рассматривать только разреженные системы уравнений. Обозначим через $\mathcal{S}$ множество всех разреженных систем уравнений.

Лемма 2. *Число разреженных систем уравнений в форме Сколема размера n есть*

$$|\mathcal{S}_n| = \prod_{k=1}^{n} (3k+m)^3.$$

Доказательство. В k-м уравнении $\alpha = \beta\gamma$ в системе $S \in \mathcal{S}_n$ вместо α есть $3k$ вариантов выбрать переменную x_i, $i \leq 3k$ и m вариантов выбрать порождающий. То же самое для β и γ. Итого для k-го уравнения есть $(3k+m)^3$ вариантов. А для всей системы из n уравнений имеем

$$|\mathcal{S}_n| = \prod_{k=1}^{n} (3k+m)^3$$

вариантов. □

Для произвольной разреженной системы уравнений $S = \{S_1, \ldots, S_k\}$ рассмотрим множество разреженных систем $eq(S)_n$, которые получаются добавлением к системе S произвольных «однородных» уравнений $S_{k+1}, \ldots, S_n$, где l-е уравнение имеет вид $x_i = x_j x_t$, причем $3k < i, j, t \leq 3(l+k)$. Легко видеть, что любая система из $eq(S)_n$ совместна над моноидом $\mathfrak{M}$ тогда и только тогда, когда совместна над $\mathfrak{M}$ система S. Действительно, единичные значения новых переменных удовлетворяют всем новым уравнениям и переменные из новых уравнений никак не участвуют в старых уравнениях.

Лемма 3. *Для любой разреженной системы S размера k и любого $n > k$ имеет место оценка*

$$\rho_n(eq(S)_n) = \frac{|eq(S)_n|}{|\mathcal{S}_n|} > \frac{(m!)^3}{(n-k+m)^{3m}(3n+m)^{3k}}.$$

Доказательство. Пусть $n > 2k$. Для t-го добавленного к S уравнения вида $x_i = x_j x_l$, где $3k < i, j, l \leq 3(t+k)$, имеется $(3t)^3 = 27t^3$ вариантов. Поэтому

$$|eq(S)_n| = \prod_{t=1}^{n-k} (3t)^3.$$

Теперь по лемме 2

$$\rho_n(eq(S)_n) = \frac{|eq(S)_n|}{|\mathcal{S}_n|} = \frac{\prod_{t=1}^{n-k}(3t)^3}{\prod_{t=1}^{n}(3t+m)^3} = \prod_{t=1}^{n-k}\Big(\frac{3t}{3t+m}\Big)^3 \prod_{t=n-k+1}^{n}\frac{1}{(3t+m)^3}.$$

Оценим снизу сначала первое произведение:

$$\prod_{t=1}^{n-k}\Big(\frac{3t}{3t+m}\Big)^3 = \Big(\prod_{t=1}^{n-k}\frac{3t}{3t+m}\Big)^3 = \Big(\prod_{t=1}^{n-k}(1-\frac{m}{3t+m})\Big)^3 >$$

$$> \Big(\prod_{t=1}^{n-k}(1-\frac{m}{t+m})\Big)^3 = \Big(\prod_{t=1}^{n-k}\frac{t}{t+m}\Big)^3 =$$

$$= \Big(\frac{1}{1+m}\cdot\frac{2}{2+m}\cdot\ldots\cdot\frac{1+m}{1+2m}\cdot\frac{2+m}{2+2m}\cdot\ldots\cdot\frac{n-k-m}{n-k}\cdot\ldots\cdot\frac{n-k}{n-k+m}\Big)^3 =$$

$$= \Big(\frac{1\cdot 2\cdot\ldots\cdot m}{(n-k+1)(n-k+2)\cdot(n-k+m)}\Big)^3 > \frac{(m!)^3}{(n-k+m)^{3m}}.$$

Теперь оценим второе произведение:

$$\prod_{t=n-k+1}^{n}\frac{1}{(3t+m)^3} > \frac{1}{(3n+m)^{3k}}.$$

Итого получаем

$$\rho_n(eq(S)) = \frac{|eq(S)_n|}{|\mathcal{S}_n|} > \frac{(m!)^3}{(n-k+m)^{3m}(3n+m)^{3k}}.$$

□

Теорема 2. *Пусть для проблемы распознавания разрешимости разреженных систем уравнений над конечно порожденным моноидом $\mathfrak{M}$ не существует полиномиального алгоритма и $\mathbf{P} = \mathbf{BPP}$. Тогда для этой проблемы не существует сильно генерического полиномиального алгоритма.*

Доказательство. Допустим, что существует сильно генерический полиномиальный алгоритм $\mathcal{A}$, определяющий разрешимость разреженных систем уравнений над $\mathfrak{M}$. Построим вероятностный полиномиальный алгоритм $\mathcal{B}$, решающий эту проблему на всем множестве входов. На системе S размера n алгоритм $\mathcal{B}$ будет работать следующим образом.

(1) Генерирует случайно и равновероятно систему S' из множества $eq(S)$ размера n^2.
(2) Запускает алгоритм $\mathcal{A}$ на системе S'.
(3) Если $\mathcal{A}(S') \neq ?$, то алгоритм правильно определяет, разрешима ли система S', а вместе с ней и система S.
(4) Если $\mathcal{A}(S') = ?$, то выдает ответ «НЕТ» – возможно неправильный.

Заметим, что полиномиальный вероятностный алгоритм $\mathcal{B}$ выдает правильный ответ на шаге 3, а на шаге 4 может выдать неправильный ответ. Надо доказать, что вероятность того, что ответ выдается на шаге 4, меньше $1/3$.

Оценим вероятность выдачи ответа на шаге 4. Вероятность того, что для S' имеет место $\mathcal{A}(S') = ?$ не больше

$$\frac{|\{S' \in \mathcal{S} : \mathcal{A}(S') = ?\}_{n^2}|}{|eq(S)_{n^2}|} = \frac{|\{S' \in \mathcal{S} : \mathcal{A}(S') = ?\}_{n^2}|}{|\mathcal{S}_{n^2}|} \times \frac{|\mathcal{S}_{n^2}|}{|eq(S)_{n^2}|}.$$

Так как множество $\{S' \in \mathcal{S} : \mathcal{A}(S') = ?\}$ сильно пренебрежимое, то существует константа $\alpha > 0$ такая, что

$$\frac{|\{S' \in \mathcal{S} : \mathcal{A}(S') = ?\}_{n^2}|}{|\mathcal{S}_{n^2}|} < \frac{1}{2^{\alpha n^2}}$$

для любого n. По лемме 3

$$\frac{|\mathcal{S}_{n^2}|}{|eq(S)_{n^2}|} < \frac{(n^2 - n + m)^{3m}(3n^2 + m)^{3n}}{(m!)^3}.$$

Поэтому искомая вероятность ответа на шаге 4 не больше

$$\frac{(n^2 - n + m)^{3m}(3n^2 + m)^{3n}}{2^{\alpha n^2}(m!)^3} = \frac{2^{3m \log (n^2 - n + m) + 3n \log (3n^2 + m)}}{2^{\alpha n^2}(m!)^3} <$$

$$< \frac{1}{2^{\alpha n^2 - 3m \log (n^2 - n + m) - 3n \log (3n^2 + m)}} < \frac{1}{3}$$

при достаточно больших n.

Таким образом, проблема проверки разрешимости разреженных систем уравнений над $\mathfrak{M}$ принадлежит классу **BPP**. А так как **BPP** = **P**, то она принадлежит классу **P**. А это противоречит тому, что для нее нет полиномиального алгоритма. □

5 Проблема поиска решения систем уравнений

В этом разделе вернемся к рассмотрению плотных систем уравнений в форме Сколема над моноидом $\mathfrak{M}$. Напомним, что под размером плотной системы понимается число переменных, участвующих в ней. Проблема поиска решения систем уравнений над моноидом $\mathfrak{M}$ состоит в том, что по заданной произвольной разрешимой над $\mathfrak{M}$ системе уравнений требуется найти любое её решение. Обозначим эту проблему $\mathcal{SEP}_{\mathfrak{M}}$.

Рассмотрим бесконечную последовательность систем уравнений

$$\sigma = \{S_1, S_2, ..., S_n, \ldots\}$$

такую, что S_n имеет размер n для $n = 1, 2, 3, \ldots$. Для каждой последовательности систем σ определим подпроблему поиска решения систем уравнений $\mathcal{SEP}_{\mathfrak{M}}(\sigma)$ как ограничение исходной проблемы $\mathcal{SEP}_{\mathfrak{M}}$ на множество входов

$$\{S : S \cong S_n, S_n \in \sigma, n \in \mathbb{N}\}.$$

Здесь $S_1 \cong S_2$ означает, что системы S_1 и S_2 – это системы от одного множества переменных $\{x_1, \ldots, x_n\}$, и система S_1 получена из системы S_2 некоторой перестановкой переменных.

Лемма 4. *Если не существует полиномиального вероятностного алгоритма для решения проблемы $\mathcal{SEP}_{\mathfrak{M}}$, то найдется последовательность систем σ такая, что не существует полиномиального вероятностного алгоритма для решения проблемы $\mathcal{SEP}_{\mathfrak{M}}(\sigma)$.*

Доказательство. Пусть $P_1, P_2, \ldots$ – все полиномиальные вероятностные алгоритмы. Если не существует полиномиального вероятностного алгоритма для проблемы $\mathcal{SEP}_{\mathfrak{M}}$, то для любого вероятностного полиномиального алгоритма P_n найдётся бесконечно много систем, для которых алгоритм P_n не может решить $\mathcal{SEP}_{\mathfrak{M}}$. Поэтому можно выбрать такую последовательность систем $\sigma' = \{S_1, S_2, \ldots, S_n, \ldots\}$, что алгоритм P_n не может решить $\mathcal{SEP}_{\mathfrak{M}}$ для S_n для всех n. Более того, можно считать, что σ' упорядочена по возрастанию размеров. Теперь можно расширить последовательность σ' до последовательности σ с системами S_n для всех размеров n. Из построения σ следует, что не существует полиномиального вероятностного алгоритма для решения проблемы $\mathcal{SEP}_{\mathfrak{M}}(\sigma)$. □

Из определения видно, что множество всех входов размера n для проблемы $\mathcal{SEP}_{\mathfrak{M}}(\sigma)$ выглядет так:

$$I_n = \{S : S \cong S_n, S_n \in \sigma\}.$$

Лемма 5. *Пусть σ – произвольная последовательность систем уравнений. Если существует генерический полиномиальный алгоритм, решающий проблему $\mathcal{SEP}_{\mathfrak{M}}(\sigma)$, то существует вероятностный полиномиальный алгоритм, решающий эту проблему на всём множестве входов.*

Доказательство. Допустим, что существует генерический полиномиальный алгоритм $\mathcal{A}$, решающий проблему поиска решения систем уравнений $\mathcal{SEP}_{\mathfrak{M}}(\sigma)$. Построим вероятностный полиномиальный алгоритм $\mathcal{B}$, решающий эту проблему на всем множестве входов. На системе S размера n алгоритм $\mathcal{B}$ работает следующим образом.

(1) Запускает алгоритм $\mathcal{A}$ на S.
(2) Если $\mathcal{A}(S) \neq ?$, то $\mathcal{B}$ выдает ответ $\mathcal{A}(S)$ и останавливается, иначе идёт на шаг 3.
(3) Генерирует случайно и равномерно перестановку π на множестве номеров переменных $\{x_1, \ldots, x_n\}$ и вычисляет систему $S' = \pi(S)$.
(4) Запускает алгоритм $\mathcal{A}$ на S'.
(5) Если $\mathcal{A}(S') = ?$, то выдает ответ $(a, \ldots, a)$, где $a \in A$ – возможно неправильный.
(6) Если $\mathcal{A}(S') = \{a_1, \ldots, a_n\}$ — решение системы S', то
$$\pi^{-1}(a_1, \ldots, a_n) = (a_{\pi^{-1}(1)}, \ldots, a_{\pi^{-1}(n)})$$
является решением системы $S = \pi^{-1}(S')$.

Для доказательства корректности работы вероятностного алгоритма надо показать, что вероятность того, что $A(S') = ?$, меньше $1/3$. Заметим, что $\pi(S)$ при варьировании перестановки π пробегает всё множество входов размера n. Множество $\{S : A(S) = ?\}$ пренебрежимо, поэтому вероятность того, что $A(S') = ?$, стремится к 0 при увеличении n. □

Теорема 3. *Если для любой последовательности систем уравнений σ для решения проблемы $\mathcal{SEP}_{\mathfrak{M}}(\sigma)$ существует генерический полиномиальный алгоритм, то для проблемы поиска решения систем уравнений над моноидом $\mathfrak{M}$ существует вероятностный полиномиальный алгоритм.*

Доказательство. Докажем утверждение от противного. Пусть для любой последовательности систем уравнений σ для проблемы $\mathcal{SEP}_{\mathfrak{M}}(\sigma)$ существует генерический полиномиальный алгоритм, но нет вероятностного полиномиального алгоритма для проблемы поиска решения систем уравнений над моноидом $\mathfrak{M}$. Тогда, по лемме 4 найдется такая последовательность систем σ, что и для $\mathcal{SEP}_{\mathfrak{M}}(\sigma)$ нет полиномиального вероятностного алгоритма. Однако, по лемме 5, так как для $\mathcal{SEP}_{\mathfrak{M}}(\sigma)$ существует генерический полиномиальный алгоритм, то для нее существует и полиномиальный вероятностный алгоритм. Полученное противоречие доказывает теорему. □

Авторы выражают благодарность рецензенту за полезные замечания и предложения по улучшению текста статьи.

References

[1] G. Baumslag, A. Myasnikov, V. Remeslennikov, *Algebraic geometry over groups I. Algebraic sets and ideal theory*, Journal of Algebra, **219:1** (1999), 16–79. Zbl 0938.20020

[2] R. Impagliazzo, A. Wigderson, *P=BPP unless E has subexponential circuits: Derandomizing the XOR Lemma*, Proceedings of 29th STOC, (1997), 220–229. Zbl 0962.68058

[3] I. Kapovich, A. Miasnikov, P. Schupp, V. Shpilrain, *Generic-case complexity, decision problems in group theory and random walks*, Journal of Algebra, **264:2** (2003), 665–694. Zbl 1041.20021

[4] G.S. Makanin, *The problem of solvability of equations in a free semigroup*, Mathematics of the USSR-Sbornik, **32:2** (1977), 129–198. Zbl 0396.20037

[5] A.Yu. Nikitin, A.N. Rybalov, *On complexity of the problem of solvability of equations over posets*, Prikl. Diskr. Mat., **39** (2018), 94–98. Zbl 1515.68220

[6] V.A. Roman'kov, *Universal theory of nilpotent groups*, Mathematical Notes, **25:4** (1979), 253–258. Zbl 0425.20027

[7] V.A. Roman'kov, *Solvability of Independent Systems of Equations in Finitely Generated Nilpotent Groups*, Mathematical Notes, **110:4** (2021), 560–564.

[8] A.N. Rybalov, *On the complexity of solving of equations over graphs*, Siberian electronic mathematical reports, **21:1** (2024), 62–69. Zbl 1557.68079

[9] A.V. Seliverstov, *The length of an unsatisfiable subformula*, Algebra and Logic, **63:1** (2024), 65–72. Zbl 7983642

Alexander Nikolaevich Rybalov
Sobolev Institute of Mathematics,
Pevtsova 13,
Omsk, 644099, Russia
Email address: alexander.rybalov@gmail.com

Artem Nikolaevich Shevlyakov
Sobolev Institute of Mathematics,
Pevtsova 13,
Omsk, 644099, Russia,
Tyumen State University,
6 Volodarskogo Street,
Tyumen, 625003, Russia
Email address: art.shevlyakov@gmail.com