УДК 510.52 DOI 10.17223/20710410/70/6

О ГЕНЕРИЧЕСКОЙ СЛОЖНОСТИ РЕШЕНИЯ УРАВНЕНИЙ В КОНЕЧНЫХ ПРЕДИКАТНЫХ АЛГЕБРАИЧЕСКИХ СИСТЕМАХ[1]

А. Н. Рыбалов

Институт математики им. С. Л. Соболева СО РАН, г. Омск, Россия

E-mail: alexander.rybalov@gmail.com

Изучается генерическая сложность двух вариантов проблемы решения уравнений без констант над конечными предикатными алгебраическими системами: распознавания разрешимости и поиска решения. Для обеих проблем во многих случаях неизвестно эффективных полиномиальных алгоритмов. Предлагается полиномиальный генерический алгоритм для проблемы распознавания разрешимости. С другой стороны, для проблемы поиска решения доказывается, что если для неё нет полиномиального вероятностного алгоритма, то существует подпроблема этой проблемы, для которой нет полиномиального генерического алгоритма. Полученный результат является теоретическим обоснованием возможных приложений проблемы поиска решения в криптографии, где нужно, чтобы проблема взлома криптоалгоритма была трудной для почти всех входов.

Ключевые слова: *генерическая сложность, конечные алгебраические системы, уравнения.*

ON GENERIC COMPLEXITY OF SOLVING OF EQUATIONS IN FINITE PREDICATE ALGEBRAIC STRUCTURES

A. N. Rybalov

Sobolev Institute of Mathematics, Omsk, Russia

In this paper, we study the generic complexity of two variants of the problem of solving equations without constants over finite predicate algebraic systems: the solvability recognition problem and the solution search problem. For both problems, efficient polynomial algorithms are not known in many cases. We propose a polynomial generic algorithm for the solvability recognition problem. On the other hand, for the solution search problem, we prove that if there is no polynomial probabilistic algorithm for it, then there is a subproblem of this problem for which there is no polynomial generic algorithm. The obtained result is a theoretical justification for possible applications of the solution search problem in cryptography, where the problem of breaking a cryptographic algorithm is required to be hard for almost all inputs. To prove this theorem, we use the method of generic amplification, which allows to construct generically hard problems from the problems hard in the classical sense. The main ingredient of this method is a technique of cloning, which unites inputs of the problem together in the large enough sets of equivalent inputs. Equivalence is understood in the sense that the problem is solved similarly for them.

Keywords: *generic complexity, finite algebraic structures, equations.*

[1]Работа поддержана грантом Российского научного фонда №25-11-20023.

Введение

Решение уравнений и систем уравнений над вещественными, комплексными, рациональными, целыми числами является классической темой исследований в различных областях математики в течение тысяч лет. В последние десятилетия фокус исследований перемещается на неклассические области, такие, как группы [1], полугруппы [2–5], графы [6], частичные порядки [7]. 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 задаётся некоторым набором уравнений, множество решений которых даёт нужный язык. К необходимости решения уравнений над графами приводят задачи проверки вложимости (совместимости) одной коммуникационной сети в другую.

Особый интерес представляет изучение вычислительной сложности проблемы решения уравнений над конечными алгебраическими системами. Очень часто здесь возникает так называемая дихотомия: для каких-то конечных систем данного класса эта проблема разрешима за полиномиальное время, для всех других является NP-полной. Это явление характерно для классов конечных групп [8], конечных полугрупп [9], конечных графов [6] (для систем уравнений без констант). Например, для конечных графов дихотомия зависит от хроматического числа графа, над которым решаются уравнения: если оно не превосходит двух, то проблема разрешима за полиномиальное время, иначе — NP-полна. Напомним, что хроматическое число графа — это минимальное число цветов, в которые можно раскрасить вершины так, чтобы любые вершины, соединённые ребром, были покрашены в разные цвета. Этот результат [6] был получен для систем уравнений без констант, однако аналогичный результат для систем с константами легко следует из работ [6, 10].

NP-полнота позволяет эффективно сводить другие практически важные NP-полные проблемы к проблеме решения уравнений и использовать мощные алгебраические методы для разработки более эффективных алгоритмов их решения. Кроме того, в случае NP-полноты проблемы решения уравнений актуальным является изучение её генерической сложности [11]. 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. Генерическая сложность проблем решения уравнений над конечными полями и полугруппами рассмотрена в [12].

В данной работе изучается генерическая сложность двух вариантов проблемы решения уравнений без констант в конечных предикатных алгебраических системах. Первый вариант — проблема распознавания разрешимости систем уравнений. Здесь входом является произвольная система уравнений без констант, необходимо определить, существует ли у неё решение. Второй вариант — проблема поиска решения системы уравнений. Для этой проблемы входом является система уравнений без констант, для которой заведомо существует решение, нужно найти хотя бы одно её решение. Проблемы поиска, в отличие от проблем распознавания, находят применения в криптографии, где всегда известно, что решение есть и надо его найти. В работе

предлагается полиномиальный генерический алгоритм для проблемы распознавания разрешимости систем уравнений. С другой стороны, для проблемы поиска решения доказывается, что если для неё не существует полиномиального вероятностного алгоритма, то существует подпроблема этой проблемы, для которой нет полиномиального генерического алгоритма. Вероятностные алгоритмы в процессе своей работы могут использовать датчик случайных чисел, что позволяет ускорять вычисления. Однако считается, что любой полиномиальный вероятностный алгоритм можно эффективно дерандомизировать, построив полиномиальный алгоритм, не использующий генератор случайных чисел и решающий ту же самую проблему. Хотя этот факт до сих пор не доказан, имеются веские основания в пользу него [13].

1. Предварительные сведения

На протяжении всей работы будем рассматривать системы уравнений без констант. Пусть $\mathfrak{A} = \langle A, \sigma \rangle$ — алгебраическая система с предикатной сигнатурой $\sigma = \{P_i^{(k_i)} : i = 1, \ldots, m\}$. *Уравнением* над $\mathfrak{A}$ называется формула одного из двух типов:

1) $(x_i = x_j)$;
2) $P_i(x_1, \ldots, x_{k_i})$, $P_i \in \sigma$, $i = 1, \ldots, m$.

Системой уравнений над $\mathfrak{A}$ называется конечный набор уравнений. *Решение* системы уравнений S от переменных $x_1, \ldots, x_t$ — это такой набор $a_1, \ldots, a_t$ элементов из A, который при подстановке в каждое уравнение системы S даёт истинную над $\mathfrak{A}$ формулу. Легко видеть, что для любой системы S над $\mathfrak{A}$ существует эквивалентная ей система S', в которой отсутствуют уравнения вида $(x_i = x_j)$. Действительно, для удаления таких уравнений достаточно во всех остальных уравнениях заменить переменную x_j на переменную x_i. Поэтому в дальнейшем будем рассматривать системы, в которых все уравнения имеют тип 2, то есть являются предикатами от переменных.

Проблема распознавания разрешимости систем уравнений над $\mathfrak{A}$ формулируется следующим образом. По произвольной заданной системе уравнений S определить, существует ли у неё решение в $\mathfrak{A}$. *Проблема поиска решения* систем уравнений над $\mathfrak{A}$ формулируется немного иначе. По произвольной заданной разрешимой системе уравнений S найти хотя бы одно её решение в $\mathfrak{A}$.

Напомним основные определения генерического подхода [11]. Пусть I — некоторое множество входов, а I_n — подмножество входов размера n. Для подмножества $S_n \subseteq I$ определим последовательность

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

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

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

Множество S называется *пренебрежимым*, если его асимптотическая плотность $\rho(S) = 0$.

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

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

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

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

Проблема распознавания множества $A \subseteq I$ *генерически разрешима за полиномиальное время*, если существует полиномиальный генерический алгоритм, вычисляющий характеристическую функцию множества A. Напомним, что *характеристической функцией* множества $A \subseteq I$ называется функция $\chi_A : I \to \{0, 1\}$, определённая следующим образом:

$$\chi_A(x) = \begin{cases} 1, & \text{если } x \in A, \\ 0, & \text{если } x \notin A. \end{cases}$$

Напомним также некоторые понятия классической теории сложности вычислений [14]. *Время работы* $t_M(x)$ машины Тьюринга M на входе $x \in I$ — это число шагов машины от начала работы до остановки. Машина Тьюринга M *полиномиальна*, если существует полином $p(n)$, такой, что для любого $x \in I$ имеет место $t_M(x) < p(\text{size}(x))$.

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 $t_M(x, \tau) < p(\text{size}(x))$.

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

$$\big(f(x) = y\big) \ \Rightarrow \Pr[M(x) = y] > 2/3.$$

Вероятностные машины Тьюринга формализуют понятие алгоритма, использующего генератор случайных чисел.

2. Генерический алгоритм распознавания разрешимости систем уравнений

Пусть $\mathfrak{A} = \langle A, \sigma \rangle$ — конечная алгебраическая система с предикатной сигнатурой $\sigma = \{P_i^{(k_i)} : i = 1, \ldots, m\}$. Будем представлять системы уравнений над $\mathfrak{A}$ следующим образом. Во-первых, зафиксируем переменные системы $x_1, \ldots, x_n$. Число переменных n — *размер системы*. По каждому предикату $P_i^{(k_i)}$, $i = 1, \ldots, m$, из сигнатуры σ рассмотрим так называемую *таблицу включения* — это k_i-мерный куб с n позициями по каждой размерности. Итого получается n^{k_i} мест. На месте с координатами $(j_1, \ldots, j_{k_i})$ записываем 1, если в системе есть уравнение $P_i(x_{j_1}, \ldots, x_{j_{k_i}})$, и 0, если нет. *Представлением* системы уравнений является набор таблиц включения для каждого предиката сигнатуры, встречающегося в этой системе. Обозначим через $\mathcal{S}$ множество систем уравнений над $\mathfrak{A}$, представленных таким образом.

Лемма 1. Число систем размера n над $\mathfrak{A}$ равно

$$|\mathcal{S}_n| = \prod_{i=1}^{m} 2^{n^{k_i}}.$$

Доказательство. Прямой подсчёт. ■

Назовём алгебраическую систему $\mathfrak{A}$ *нетривиальной*, если существует система уравнений, которая не имеет решения над $\mathfrak{A}$. В противном случае $\mathfrak{A}$ *тривиальная*. Очевидно, что для тривиальных алгебраических систем проблема распознавания разрешимости систем уравнений разрешима за полиномиальное время.

Теорема 1. Проблема распознавания разрешимости систем уравнений над конечной нетривиальной алгебраической системой $\mathfrak{A}$ генерически разрешима за полиномиальное время.

Доказательство. Пусть S' — какая-то фиксированная система уравнений размера t, неразрешимая над $\mathfrak{A}$. Полиномиальный генерический алгоритм для распознавания разрешимости систем уравнений над $\mathfrak{A}$ работает на системе S размера n следующим образом:

1) Ищет в системе S подсистему, эквивалентную S': перебирает все выборки по t переменных из n переменных системы S; для каждой выборки ищет в S все предикаты из системы S' с учётом замены переменных S' соответствующими переменными из выборки. Число таких выборок $\mathrm{C}_n^t = O(n^t)$ полиномиально, и проверка каждой выборки делается за полиномиальное от n время.
2) Если эквивалентная подсистема нашлась, то выдаёт ответ «НЕТ».
3) Если нет, выдаёт ответ «НЕ ЗНАЮ».

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

Обозначим через I множество индексов тех предикатов из P_i, $i = 1, \ldots, m$, сигнатуры σ, которые встречаются в системе уравнений S'. Можно подсчитать, что

$$|B_n| = \prod_{i \notin I} 2^{n^{k_i}} \prod_{i \in I} 2^{n^{k_i} - t^{k_i}[n/t]} (2^{t^{k_i}} - 1)^{[n/t]}.$$

Это следует из того, что в таблицах включения для каждого предиката с индексом из I для систем из множества B «запрещены» $[n/t]$ подтаблиц размера t, соответствующих предикатам из системы S'. Эти подтаблицы имеют по $2^{t^{k_i}}$ мест для расстановки нулей и единиц.

Теперь запишем:

$$\rho(B) = \lim_{n\to\infty} \frac{|B_n|}{|\mathcal{S}_n|} = \lim_{n\to\infty} \frac{\prod\limits_{i \notin I} 2^{n^{k_i}} \prod\limits_{i \in I} 2^{n^{k_i} - t^{k_i}[n/t]} (2^{t^{k_i}} - 1)^{[n/t]}}{\prod\limits_{i=1}^{m} 2^{n^{k_i}}} =$$

$$= \lim_{n\to\infty} \frac{\prod\limits_{i \in I} (2^{t^{k_i}} - 1)^{[n/t]}}{\prod\limits_{i \in I} 2^{t^{k_i}[n/t]}} = \prod_{i \in I} \lim_{n\to\infty} \left(1 - 2^{-t^{k_i}}\right)^{[n/t]} = 0.$$

Это доказывает, что множество B является пренебрежимым, а значит, его подмножество A тем более пренебрежимо. ■

3. Проблема поиска решения систем уравнений

Напомним, что проблема поиска решения систем уравнений над алгебраической системой $\mathfrak{A}$ состоит в том, что по заданной произвольной разрешимой над $\mathfrak{A}$ системе уравнений требуется найти любое её решение. Обозначим эту проблему $\mathcal{SEP}_{\mathfrak{A}}$. Для неё также не известно полиномиальных алгоритмов.

Рассмотрим бесконечную последовательность систем уравнений

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

такую, что S_n имеет размер n для $n = 1, 2, 3, \ldots$ Для каждой последовательности систем σ определим подпроблему поиска решения систем уравнений $\mathcal{SEP}_{\mathfrak{A}}(\sigma)$ как ограничение исходной проблемы $\mathcal{SEP}_{\mathfrak{A}}$ 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 S_1 получена из S_2 некоторой перестановкой переменных.

Лемма 2. Если не существует полиномиального вероятностного алгоритма для решения проблемы $\mathcal{SEP}_{\mathfrak{A}}$, то найдётся последовательность систем σ, такая, что не существует полиномиального вероятностного алгоритма для решения проблемы $\mathcal{SEP}_{\mathfrak{A}}(\sigma)$.

Доказательство. Пусть $P_1, P_2, \ldots$ — все полиномиальные вероятностные алгоритмы. Если не существует полиномиального вероятностного алгоритма для проблемы $\mathcal{SEP}_{\mathfrak{A}}$, то для любого вероятностного полиномиального алгоритма P_n найдётся бесконечно много систем, для которых алгоритм P_n не может решить $\mathcal{SEP}_{\mathfrak{A}}$. Поэтому можно выбрать такую последовательность систем $\sigma' = \{S_1, S_2, \ldots, S_n, \ldots\}$, что алгоритм P_n не может решить $\mathcal{SEP}_{\mathfrak{A}}$ 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 $\mathcal{SEP}_{\mathfrak{A}}(\sigma)$. ■

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

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

Лемма 3. Пусть σ — произвольная последовательность систем уравнений. Если существует генерический полиномиальный алгоритм, решающий проблему $\mathcal{SEP}_{\mathfrak{A}}(\sigma)$, 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 $\mathcal{SEP}_{\mathfrak{A}}(\sigma)$. Построим вероятностный полиномиальный алгоритм $\mathcal{B}$, решающий эту проблему на всём множестве входов. На системе S размера n алгоритм $\mathcal{B}$ работает следующим образом:

1) Запускает алгоритм $\mathcal{A}$ на S.
2) Если $\mathcal{A}(S) \neq \square$, то $\mathcal{B}$ выдаёт 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 $\mathcal{A}(S') = \square$, то выдаёт ответ $(a, \ldots, a)$, где $a \in A$, — возможно, неправильный.
6) Если $\mathcal{A}(S') = \{a_1, \ldots, a_n\}$ — решение системы S', то

$$\pi^{-1}(a_1, \ldots, a_n) = \left(a_{\pi^{-1}(1)}, \ldots, a_{\pi^{-1}(n)}\right)$$

является решением системы $S = \pi^{-1}(S')$.

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

Теорема 2. Если для проблемы поиска решения систем уравнений над алгебраической системой $\mathfrak{A}$ не существует вероятностного полиномиального алгоритма, то существует последовательность систем уравнений σ, такая, что для решения проблемы $\mathcal{SEP}_{\mathfrak{A}}(\sigma)$ не существует генерического полиномиального алгоритма.

Доказательство. Пусть для проблемы $\mathcal{SEP}_{\mathfrak{A}}$ нет вероятностного полиномиального алгоритма. По лемме 2 найдётся такая последовательность систем σ, что и для $\mathcal{SEP}_{\mathfrak{A}}(\sigma)$ нет полиномиального вероятностного алгоритма. Теперь если допустить, что для $\mathcal{SEP}_{\mathfrak{A}}(\sigma)$ существует полиномиальный генерический алгоритм, то по лемме 3 для $\mathcal{SEP}_{\mathfrak{A}}(\sigma)$ найдётся полиномиальный вероятностный алгоритм. Полученное противоречие доказывает теорему. ■

ЛИТЕРАТУРА

1. *Baumslag G., Myasnikov A., and Remeslennikov V.* Algebraic geometry over groups I. Algebraic sets and ideal theory // J. Algebra. 1999. V. 219. No. 1. P. 16–79.
2. *Shevlyakov A. N.* Equationally Noetherian varieties of semigroups and B. Plotkin's problem // Сиб. электрон. матем. изв. 2023. Т. 20. № 2. С. 724–734.
3. *Шевляков А. Н.* Сплетения полугрупп и проблема Б. И. Плоткина // Алгебра и логика. 2023. Т. 62. № 5. С. 665–691.
4. *Shevlyakov A. N.* On disjunctions of algebraic sets in completely simple semigroups // Communications in Algebra. 2017. V. 45. No. 9. P. 3757–3767.
5. *Шевляков А. Н.* Об объединении решений систем уравнений в полугруппах с конечным идеалом // Алгебра и логика. 2016. Т. 55. № 1. С. 87–105.
6. *Рыбалов А. Н.* О сложности решения уравнений над графами // Сиб. электрон. матем. изв. 2024. Т. 21. № 1. С. 62–69.
7. *Nikitin A. and Shevlyakov A.* On radicals over strict partial order sets // J. Phys.: Conf. Ser. 2021. V. 1791. No. 012080. 6 p.
8. *Goldmann M. and Russell A.* The complexity of solving equations over finite groups // Information and Computation. 2002. V. 178. No. 1. P. 253–262.
9. *Klima O., Tesson P., and Therien D.* Dichotomies in the complexity of solving systems of equations over finite semigroups // Theory Comput. Syst. 2007. V. 40. P. 263–297.
10. *Ильев А. В., Ильев В. П.* Алгоритмы для решения систем уравнений над различными классами конечных графов // Прикладная дискретная математика. 2021. № 53. С. 89–102.
11. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks // J. Algebra. 2003. V. 264. No. 2. P. 665–694.

12. *Rybalov A. and Shevlyakov A.* Generic complexity of solving of equations in finite groups, semigroups and fields // J. Phys.: Conf. Ser. 2021. V. 1901. No. 012047. 8 p.
13. *Impagliazzo R. and Wigderson A.* P = BPP unless E has subexponential circuits: Derandomizing the XOR Lemma. Proc. 29th STOC. El Paso: ACM, 1997. P. 220–229.
14. *Вялый М., Китаев А., Шень А.* Классические и квантовые вычисления. М.: МЦНМО, ЧеРо. 1999. 192 с.

REFERENCES

1. *Baumslag G., Myasnikov A., and Remeslennikov V.* Algebraic geometry over groups I. Algebraic sets and ideal theory. J. Algebra, 1999, vol. 219, no. 1, pp. 16–79.
2. *Shevlyakov A. N.* Equationally Noetherian varieties of semigroups and B. Plotkin's problem. Sib. Elektron. Mat. Izv., 2023, vol. 20, no. 2, pp. 724–734.
3. *Shevlyakov A. N.* Wreath products of semigroups and Plotkin's problem. Algebra and Logic, 2023, vol. 62, no. 5, pp. 448–467.
4. *Shevlyakov A. N.* On disjunctions of algebraic sets in completely simple semigroups. Communications in Algebra, 2017, vol. 45, no. 9, pp. 3757–3767.
5. *Shevlyakov A. N.* Combining solutions for systems equations in semigroups with finite ideal. Algebra and Logic, 2016, vol. 55, no. 1, pp. 58–71.
6. *Rybalov A. N.* O slozhnosti resheniya uravneniy nad grafami [On complexity of solving of equations over graphs]. Sib. Electr. Mat. Izv., 2024, vol. 21, no. 1, pp. 62–69. (in Russian)
7. *Nikitin A. and Shevlyakov A.* On radicals over strict partial order sets. J. Phys.: Conf. Ser., 2021, vol. 1791, no. 012080, 6 p.
8. *Goldmann M. and Russell A.* The complexity of solving equations over finite groups. Information and Computation, 2002, vol. 178, no. 1, pp. 253–262.
9. *Klima O., Tesson P., and Therien D.* Dichotomies in the complexity of solving systems of equations over finite semigroups. Theory Comput. Syst., 2007, vol. 40, pp. 263–297.
10. *Il'ev A. V. and Il'ev V. P.* Algoritmy dlya resheniya sistem uravneniy nad razlichnymi klassami konechnykh grafov [Algorithms for solving systems of equations over various classes of finite graphs]. Prikladnaya Diskretnaya Matematika, 2021, no. 53, pp. 89–102. (in Russian)
11. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks. J. Algebra, 2003, vol. 264, no. 2, pp. 665–694.
12. *Rybalov A. and Shevlyakov A.* Generic complexity of solving of equations in finite groups, semigroups and fields. J. Phys.: Conf. Ser., 2021, vol. 1901, no. 012047, 8 p.
13. *Impagliazzo R. and Wigderson A.* P = BPP unless E has subexponential circuits: Derandomizing the XOR Lemma. Proc. 29th STOC, El Paso, ACM, 1997, pp. 220–229.
14. *Vyalyy M., Kitaev A., and Shen' A.* Klassicheskie i kvantovye vychisleniya [Classical and Quantum Computations]. Moscow, MCCME Publ., 1999. 192 p. (in Russian)