S©MR

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

Том 23, № 1, стр. 266–288 (2026)
https://doi.org/10.33048/semi.2026.23.017

УДК 510.522, 519.178
MSC 03D15, 05C85

СЛОЖНОСТЬ ПРОБЛЕМЫ СОВМЕСТНОСТИ СИСТЕМ ДИОФАНТОВЫХ УРАВНЕНИЙ НАД КОНЕЧНЫМИ ДВУДОЛЬНЫМИ ГРАФАМИ

Ф.Д. Кобзев, Н.Т. Когабаев

Представлено С.В. Судоплатовым

Abstract: We study finite systems of Diophantine equations over finite bipartite graphs. In 2021, A.V. Il'ev and V.P. Il'ev proposed a deterministic polynomial-time procedure for verifying the consistency of such systems. We construct a counterexample to show that their procedure works incorrectly and prove that the consistency problem for such systems is in fact NP-complete.

Keywords: bipartite graph, system of equations, nondeterministic Turing machine, NP-complete problem.

1 Введение

Классическая алгебраическая геометрия традиционно занимается изучением систем уравнений над числовыми полями и кольцами. Однако, в последние годы активно развивается обобщённый подход, в рамках которого исследуются системы уравнений над произвольными алгебраическими структурами [3]. Вместе с этим усилился интерес к вопросам

Kobzev, F.D., Kogabaev, N.T. Complexity of the consistency problem for systems of Diophantine equations over finite bipartite graphs.

Работа второго автора выполнена в рамках государственного задания ИМ СО РАН (проект FWNF-2026-0032).

Поступила 19 июня 2025 г., опубликована 31 марта 2026 г.

разрешимости и вычислительной сложности алгоритмов поиска решений таких систем, особенно в случае предикатных структур. В [8] доказано, что для систем уравнений над конечными частичными порядками проблема существования решения, состоящего из попарно различных элементов, является NP-полной. В [7] представлен полиномиальный алгоритм построения радикала и координатного частичного порядка для систем уравнений над конечными частичными порядками в языке без констант. В [4] установлена NP-полнота проблемы совместности систем уравнений над конечными структурами в таких классах как полные p-дольные графы ($p \geqslant 3$), матроиды, ранг которых не превосходит k ($k \geqslant 2$), k-однородные матроиды ($k \geqslant 2$) и матроиды разбиения ранга, не превосходящего k ($k \geqslant 3$).

Особое внимание уделяется исследованию систем уравнений над конечными графами. Так, в [6] были предложены алгоритмы проверки совместности, вычисления радикала и нахождения координатного графа для систем уравнений над произвольными конечными графами. В дальнейшем, в [5] были найдены оценки трудоемкости алгоритмов проверки совместности и нахождения общего решения систем уравнений в различных классах конечных графов. В частности, в [5] было во-первых доказано, что проблема совместности конечных систем диофантовых уравнений над конечными симметричными иррефлексивными графами является NP-полной, а во-вторых приведён результат о том, что аналогичная проблема для систем уравнений над конечными двудольными графами является полиномиально разрешимой, т.е. лежит в классе P.

Однако, было выявлено, что второй из упомянутых выше результатов не верен, а именно, процедура 1.1 [5] проверки совместности систем уравнений над конечными двудольными графами работает некорректно. Данная процедура делает вывод о совместности системы, опираясь на необходимый, но не достаточный признак, что может приводить к ложным положительным ответам.

Таким образом, вопрос о том, какова всё-таки вычислительная сложность проблемы совместности конечных систем диофантовых уравнений над конечными двудольными графами, остаётся актуальным. В настоящей статье мы строим контрпример, показывающий, что процедура 1.1 [5] работает некорректно, и доказываем, что указанная проблема на самом деле является NP-полной.

Теперь перейдём к формальной постановке задачи и предварительным сведениям, необходимым для дальнейшего изложения.

В данной работе мы будем рассматривать только конечные симметричные иррефлексивные графы. Таким образом, *графом* мы называем пару $\Gamma = \langle V, E\rangle$, где V — непустое конечное множество, элементы которого называют *вершинами*, а $E \subseteq V^2$ — симметричное иррефлексивное бинарное отношение на V, элементы которого называют *рёбрами*. Граф является *двудольным*, если множество его вершин можно разбить на два непересекающихся подмножества V_1 и V_2 так, что $E \subseteq (V_1 \times V_2) \cup (V_2 \times V_1)$.

Следуя [3, гл. 2], введём основные понятия универсальной алгебраической геометрии над произвольным графом.

Пусть $\Gamma = \langle V, E\rangle$ — симметричный иррефлексивный граф, $C \subseteq V$ — некоторое множество его вершин. Определим сигнатуру $\sigma_C = \langle E, c\rangle_{c\in C}$, состоящую из предикатного символа E смежности вершин и константных символов из множества C, при этом мы считаем, что каждый символ $c \in C$ всегда интерпретируется в графе Γ как одноименная вершина c.

Пусть $X = \{x_1, \ldots, x_k\}$ — конечное множество переменных. Множество термов сигнатуры σ_C над переменными из X совпадает с $C \cup X$. Обозначим через $\mathrm{At}_{\sigma_C}(X)$ множество всех атомарных формул сигнатуры σ_C над переменными из X.

Уравнением сигнатуры σ_C над X будем называть любую формулу $\varphi(x_1, \ldots, x_k) \in \mathrm{At}_{\sigma_C}(X)$. Таким образом, уравнениями сигнатуры σ_C над X являются формулы двух видов: $t = t'$ или $E(t, t')$, где t и t' — термы сигнатуры σ_C.

Системой уравнений сигнатуры σ_C над X будем называть любое подмножество $S \subseteq \mathrm{At}_{\sigma_C}(X)$. *Множеством решений системы уравнений S над Γ* будем называть множество

$$V_\Gamma(S) = \{\langle v_1, \ldots, v_k\rangle \in V^k \mid \Gamma \models \varphi(v_1, \ldots, v_k) \text{ для всех } \varphi \in S\}.$$

Система S *совместна над* Γ, если $V_\Gamma(S) \neq \varnothing$, иначе она является *несовместной над* Γ. Системы уравнений S_1 и S_2 называются *эквивалентными над* Γ, если $V_\Gamma(S_1) = V_\Gamma(S_2)$.

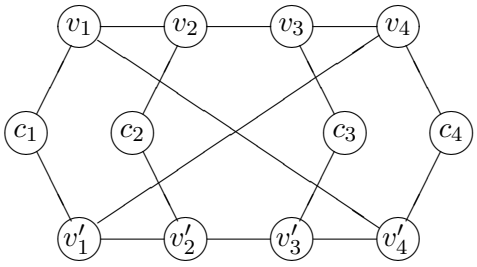
В данной работе мы будем рассматривать только конечные системы *диофантовых* уравнений, то есть уравнений, в которых множество констант C совпадает с множеством V вершин графа.

В § 2 настоящей статьи приведён пример симметричного иррефлексивного двудольного графа и соответствующей системы уравнений, для которых процедура 1.1 [5] работает некорректно. В § 3 описана недетерминированная полиномиальная процедура для распознавания совместных конечных систем диофантовых уравнений над конечными двудольными графами. В § 4 построено полиномиальное сведение NP-полной проблемы совместности конечных систем диофантовых уравнений над конечными симметричными иррефлексивными графами к рассматриваемой проблеме, ограниченной двудольными графами, и тем самым установлена NP-полнота последней.

2 Контрпример к результату А.В. Ильева и В.П. Ильева

Как было отмечено во введении, в [5] была предложена процедура 1.1 проверки совместности конечных систем уравнений над конечными двудольными графами, при этом утверждалась её корректность и полиномиальная временна́я сложность $\mathcal{O}(k^2 n(k+n)^2)$, где n — число вершин в графе, а k — число переменных в системе.

Рассмотрим граф Γ и систему уравнений S, изображённые на рисунке ниже.



$$E(x_1, x_2), \quad E(x_1, c_1),$$
$$E(x_2, x_3), \quad E(x_2, c_2),$$
$$E(x_3, x_4), \quad E(x_3, c_3),$$
$$E(x_4, x_1), \quad E(x_4, c_4).$$

Нетрудно проверить, что граф Γ является двудольным, а система уравнений S не совместна над этим графом, поскольку G не содержит циклов длины 4.

Описание процедуры 1.1 и точные определения используемых в ней данных можно найти в [5]. Отметим лишь, что состоящая из шагов 0–4 процедура 1.1 преобразует исходную систему S в эквивалентную ей систему, и если $\overline{S}$ — текущая система уравнений, то на каждом шаге исполнения процедуры 1.1 множество W_i состоит из вершин v графа Γ, обладающих свойством $E(x_i, v) \in \overline{S}$, а множество $W_i^{\perp}$ состоит из всех вершин графа Γ, смежных с каждой вершиной из W_i.

Если подать на вход процедуры 1.1 наши граф Γ и систему S, то после шага 0 система останется без изменений и для всех $1 \leqslant i \leqslant 4$ будут определены множества $W_i = \{c_i\}$ и $W_i^{\perp} = \{v_i, v'_i\}$.

Шаги 1–3 процедуры 1.1 исполняются до тех пор, пока из текущей системы не будут удалены уравнения вида $t = t'$. Поэтому шаги 1–3 этой процедуры не изменят множества W_i, $W_i^{\perp}$ и систему S.

Далее на шаге 4 рассматривается граф H, вершинами которого являются переменные из текущей системы уравнений, а рёбра определяются уравнениями вида $E(x_i, x_j) \in S$. В нашем случае граф H изоморфен графу C_4. Исполнение инструкций шага 4 вновь не приводит к изменениям в W_i, $W_i^{\perp}$ и S. При этом критерием совместности системы на шаге 4 является отсутствие циклов нечётной длины в графе H и отсутствие пустых множеств $W_i^{\perp}$.

Таким образом, процедура 1.1 завершает свою работу с положительным результатом о совместности системы S, что противоречит исходному утверждению о несовместности S.

3 Принадлежность классу NP

В [5] построена детерминированная процедура 1, которая осуществляет проверку совместности конечных систем диофантовых уравнений над произвольными конечными графами за экспоненциальное время. В данном параграфе мы приведём модификацию данной процедуры для

распознавания совместных систем уравнений над конечными двудольными графами. Модификация коснётся лишь шага 4, в котором полный перебор всех кортежей фиксированной длины из допустимых элементов графа заменён на недетерминированное угадывание одного такого кортежа, которое будет осуществляться за полиномиальное время. Остальные шаги процедуры повторяют аналогичные шаги процедуры 1 [5].

В качестве математической модели для реализации процедур мы используем недетерминированную машину Тьюринга с несколькими рабочими лентами [1, § 10.1]. Мы также будем придерживаться определений класса NP и понятия NP-полной проблемы, принятых в [1, § 10.2]. Машины Тьюринга будут интересовать нас прежде всего как устройства для распознавания языков. Напомним, что недетерминированная машина Тьюринга T *распознаёт* слово w над некоторым фиксированным алфавитом, если хотя бы одна ветвь вычислений, запущенных на машине T с входным словом w, достигает заключительного состояния машины. Таким образом, нам необходимо представить интересующую нас алгоритмическую проблему в виде формального языка над некоторым конечным алфавитом.

Пусть даны произвольный конечный симметричный иррефлексивный граф $\Gamma = \langle V, E\rangle$, конечное множество переменных X и система уравнений S сигнатуры $\sigma_V = \langle E, v\rangle_{v\in V}$ над X. Будем считать, что $|V| = n$, $|X| = k$, $V = \{v_1, \ldots, v_n\}$, $X = \{x_1, \ldots, x_k\}$. Занумеруем все термы сигнатуры σ_V над переменными из X следующим образом:

$$t_1 = v_1, \ldots, t_n = v_n, \quad t_{n+1} = x_1, \ldots, t_{n+k} = x_k.$$

В процессе работы наши процедуры будут использовать следующие данные, которые хранятся на лентах машины Тьюринга в виде слова над алфавитом $\{0, 1, \#, *, \lhd, \rhd\}$:

1-я лента содержит граф Γ, представленный матрицей смежности размера $n \times n$, записанной в ячейках ленты в виде слова

$$\mathcal{A} = \lhd \mathcal{A}_{1,1} \ldots \mathcal{A}_{1,n} \# \ldots \# \mathcal{A}_{n,1} \ldots \mathcal{A}_{n,n} \rhd,$$

где $\mathcal{A}_{i,j} \in \{0, 1\}$ и $\mathcal{A}_{i,j} = 1 \Leftrightarrow \Gamma \models E(v_i, v_j)$ для всех $1 \leqslant i, j \leqslant n$.

2-я лента хранит уравнения вида $E(t_i, t_j)$, представленные матрицей размера $(n+k) \times (n+k)$, записанной в ячейках ленты в виде слова

$$\mathcal{R} = \lhd \mathcal{R}_{1,1} \ldots \mathcal{R}_{1,n+k} \# \ldots \# \mathcal{R}_{n+k,1} \ldots \mathcal{R}_{n+k,n+k} \rhd,$$

где $\mathcal{R}_{i,j} \in \{0, 1\}$ и $\mathcal{R}_{i,j} = 1 \Leftrightarrow E(t_i, t_j) \in S$ для всех $1 \leqslant i, j \leqslant n+k$. Поскольку Γ — симметричный граф, можно считать, что матрица $\mathcal{R}$ тоже симметрична. В силу такого соглашения, мы не будем различать уравнения $E(t_i, t_j)$ и $E(t_j, t_i)$.

3-я лента содержит уравнения вида $t_i = t_j$ в виде матрицы размера $(n+k) \times (n+k)$, записанной в ячейках ленты в виде слова

$$\mathcal{E} = \lhd \mathcal{E}_{1,1} \ldots \mathcal{E}_{1,n+k} \# \ldots \# \mathcal{E}_{n+k,1} \ldots \mathcal{E}_{n+k,n+k} \rhd,$$

где $\mathcal{E}_{i,j} \in \{0,1\}$ и $\mathcal{E}_{i,j} = 1 \Leftrightarrow t_i = t_j \in S$ для всех $1 \leqslant i,j \leqslant n{+}k$. Поскольку равенство — симметричное отношение, можно считать, что матрица $\mathcal{E}$ также симметрична. В соответствии с этим, мы не будем различать уравнения $t_i = t_j$ и $t_j = t_i$.

4-я лента кодирует семейство подмножеств $W_i \subseteq V$, где $1 \leqslant i \leqslant k$. Каждое множество W_i состоит из таких вершин v_j графа Γ, для которых в текущей системе уравнений присутствует уравнение вида $E(x_i, v_j)$. Множества W_i представлены построчно матрицей размера $k \times n$, которая будет записана в ячейках ленты в виде слова

$$\mathcal{W} = \triangleleft \mathcal{W}_{1,1} \ldots \mathcal{W}_{1,n} \# \ldots \# \mathcal{W}_{k,1} \ldots \mathcal{W}_{k,n} \triangleright,$$

где $\mathcal{W}_{i,j} \in \{0,1\}$ и $\mathcal{W}_{i,j} = 1 \Leftrightarrow v_j \in W_i$ для всех $1 \leqslant i \leqslant k$ и $1 \leqslant j \leqslant n$.

5-я лента кодирует семейство подмножеств $W_i^{\perp} \subseteq V$, где $1 \leqslant i \leqslant k$. Каждое множество $W_i^{\perp}$ состоит из вершин графа Γ, смежных с каждой вершиной множества W_i. Если $W_i = \varnothing$, то по определению полагаем $W_i^{\perp} = V$. Множества $W_i^{\perp}$ представлены построчно матрицей размера $k \times n$, которая будет записана в ячейках ленты в виде слова

$$\mathcal{W}^{\perp} = \triangleleft \mathcal{W}_{1,1}^{\perp} \ldots \mathcal{W}_{1,n}^{\perp} \# \ldots \# \mathcal{W}_{k,1}^{\perp} \ldots \mathcal{W}_{k,n}^{\perp} \triangleright,$$

где $\mathcal{W}_{i,j}^{\perp} \in \{0,1\}$ и $\mathcal{W}_{i,j}^{\perp} = 1 \Leftrightarrow v_j \in W_i^{\perp}$ для всех $1 \leqslant i \leqslant k$ и $1 \leqslant j \leqslant n$. Неформально, $W_i^{\perp}$ можно интерпретировать как множество допустимых значений, которые может принимать переменная x_i в рамках текущей системы уравнений.

6-я лента хранит семейство классов $Y(t_j)$, где $1 \leqslant j \leqslant n{+}k$, которые образуют разбиение множества $V \cup X$. В ходе исполнения процедуры некоторые классы могут стать пустыми, при этом элементы таких классов переносятся в другие классы. Если класс $Y(t_j)$ не пуст, то процедура будет отождествлять все его элементы друг с другом, при этом j будет наименьшим индексом термов из $Y(t_j)$. Классы представлены построчно матрицей размера $(n{+}k) \times (n{+}k)$, которая будет записана в ячейках ленты в виде слова

$$\mathcal{Y} = \triangleleft \mathcal{Y}_{1,1} \ldots \mathcal{Y}_{1,n+k} \# \ldots \# \mathcal{Y}_{n+k,1} \ldots \mathcal{Y}_{n+k,n+k} \triangleright,$$

где $\mathcal{Y}_{i,j} \in \{0,1\}$ и $\mathcal{Y}_{i,j} = 1 \Leftrightarrow t_j \in Y(t_i)$ для всех $1 \leqslant i,j \leqslant n{+}k$.

Ленты 7–10 являются вспомогательными.

Для нашей основной процедуры, распознающей совместные системы уравнений над конечными двудольными графами, входными данными будут описанные выше слова $\mathcal{A}$, $\mathcal{R}$, $\mathcal{E}$. Таким образом, наша основная процедура, запущенная на тройке входных слов $\mathcal{A}$, $\mathcal{R}$, $\mathcal{E}$, будет переходить в заключительное состояние тогда и только тогда, когда слова $\mathcal{A}$, $\mathcal{R}$, $\mathcal{E}$ соответствуют совместной системе уравнений. Напомним, что недетерминированная машина Тьюринга T имеет *временну́ю сложность* $f(l)$, если для любого распознаваемого машиной T входного слова w длины l существует ветвь вычислений на машине T, вдоль которой слово w распознаётся за не более чем $f(l)$ шагов. В нашем случае размер входных

данных определяется как сумма их длин и, соответственно, функцию временной сложности основной процедуры мы будем рассматривать как функцию от натурального аргумента $l = |\mathcal{A}| + |\mathcal{R}| + |\mathcal{E}|$. В частности, $l = n^2 + 2(n+k)^2 + 3n + 2k + 3$.

Описание процедур будем излагать на естественном языке, т.е. без подробного изложения программ на языке машин Тьюринга. Иногда наши процедуры будут останавливаться и выдавать сообщение о несовместности системы — в этом случае мы считаем, что остановка произошла в состоянии, отличном от заключительного. Объекты, указанные в описаниях наших процедур, будут использоваться как динамические переменные, т.е. данные объекты могут переопределяться в ходе исполнения процедуры, но при этом для них используются те же имена, которые были и до переопределения.

Опишем сначала детерминированную процедуру INIT для инициализации описанных выше данных. Процедура INIT эквивалентна шагу 0 процедуры 1 из [5].

Процедура INIT.

Входными данными процедуры являются двудольный граф Γ и система уравнений S. Выходными данными являются множества W_i, $W_i^{\perp}$ и классы эквивалентности $Y(t_i)$. Работа процедуры состоит из следующих трёх пунктов:

(1) Обходим уравнения системы S и для каждого уравнения вида $E(x_i, v_j)$ устанавливаем $\mathcal{W}_{i,j} = 1$. Для остальных ячеек матрицы $\mathcal{W}$ значение приравниваем к нулю.

(2) Для построения матрицы $\mathcal{W}^{\perp}$ обходим множества W_i. Если $W_i \neq \varnothing$, вычисляем элементы $\mathcal{W}^{\perp}_{i,j}$ по формуле

$$\mathcal{W}^{\perp}_{i,j} = \prod\{\mathcal{A}_{m,j} \mid 1 \leqslant m \leqslant n, \mathcal{W}_{i,m} = 1\}.$$

Если же $W_i = \varnothing$, полагаем $W_i^{\perp} = V$. Если оказывается, что хотя бы одна строка матрицы $\mathcal{W}^{\perp}$ содержит только нули, то процедура останавливается и выдаёт сообщение о несовместности системы.

(3) Для каждого терма t_i устанавливаем $Y(t_i) = \{t_i\}$.

Легко видеть, что после исполнения процедуры INIT система S эквивалентна системе $S \cup \{t = t' \mid \exists t_i \in V \cup X (t, t' \in Y(t_i))\}$. Корректность работы процедуры очевидна. Оценим время её работы.

Предложение 1. *Время работы процедуры INIT составляет* $\mathcal{O}(l^2)$.

Доказательство. Пункт (1) выполняется за время $\mathcal{O}(l)$. В пункте (2) для каждого фиксированного $i \in \{1, \dots k\}$ нахождение всех элементов множества $W_i^{\perp}$ требует время $\mathcal{O}(l)$, следовательно, суммарно пункт (2) исполняется за время $\mathcal{O}(l^2)$. Время исполнения пункта (3) составляет $\mathcal{O}(l)$. Таким образом, время работы процедуры INIT составляет $\mathcal{O}(l^2)$. □

Опишем следующую детерминированную процедуру ELIM, которая преобразует систему уравнений S, устраняя из неё уравнения с равенствами, но при этом сохраняя множество решений системы $S \cup \{t = t' \mid \exists t_i \in V \cup X (t, t' \in Y(t_i))\}$. Процедура ELIM эквивалентна шагам 1–3 процедуры 1 из [5].

Процедура ELIM.

Входными данными процедуры являются двудольный граф Γ, система уравнений S, а также множества W_i, $W_i^\perp$ и классы эквивалентности $Y(t_i)$, удовлетворяющие следующим условиям:

(а) Множества $Y(t_i)$ образуют разбиение множества $V \cup X$;

(б) Если $Y(x_i) \neq \varnothing$, то $W_i = \{v_j \in V \mid E(x_i, v_j) \in S\}$;

(в) Если $Y(x_i) \neq \varnothing$ и $W_i \neq \varnothing$, то $W_i^\perp = \{v_p \in V \mid \forall v_j \in W_i (\langle v_p, v_j \rangle \in E)\}$;

(г) Если $Y(x_i) \neq \varnothing$ и $W_i = \varnothing$, то $W_i^\perp = V$.

Выходными данными являются обновлённые система уравнений S, множества W_i, $W_i^\perp$ и классы $Y(t_i)$. Работа процедуры заключается в исполнении циклической структуры, состоящей из шагов 1–3, при этом проверка условия выхода из цикла осуществляется в конце шага 3.

Шаг 1. Устранение равенств и объединение классов.

Последовательно просматриваем в системе S все уравнения, содержащие равенства. Для каждого такого уравнения $t_i = t_j$ объединяем классы эквивалентности, содержащие термы t_i и t_j, следующим образом. Пусть $t_i \in Y(t_p)$, $t_j \in Y(t_q)$, $m = \min\{p, q\}$ и $s = \max\{p, q\}$. Тогда полагаем $Y(t_m) = Y(t_p) \cup Y(t_q)$ и $Y(t_s) = \varnothing$. После этого уравнение $t_i = t_j$ удаляется из системы.

Шаг 2. Проверка на несовместность и замена переменных.

Последовательно рассматриваем все непустые классы $Y(t)$ и для каждого из них исполняем следующие пункты (1)–(6):

(1) Если в $Y(t)$ содержатся две различные константы v_i и v_j, то процедура останавливается и выдаёт сообщение о несовместности системы.

(2) Если в $Y(t)$ содержится только одна константа v_j и хотя бы одна переменная, при этом выполняется условие $\left(\bigcap\{W_i^\perp \mid x_i \in Y(t)\}\right) \cap \{v_j\} = \varnothing$, то процедура останавливается и выдаёт сообщение о несовместности системы.

(3) Если $Y(t)$ не содержит констант из V и $\bigcap\{W_i^\perp \mid x_i \in Y(t)\} = \varnothing$, то процедура также останавливается и выдаёт сообщение о несовместности системы.

(4) Если в $Y(t)$ содержится только одна константа v_j и хотя бы одна переменная, и при этом $\left(\bigcap\{W_i^\perp \mid x_i \in Y(t)\}\right) \cap \{v_j\} = \{v_j\}$, то во всех уравнениях из S каждую переменную $x_i \in Y(t)$ заменяем на v_j.

(5) Если $Y(t)$ не содержит констант из V, но при этом $\bigcap\{W_i^\perp \mid x_i \in Y(t)\} = \{v_j\}$ для некоторого v_j, то во всех уравнениях из S каждую переменную $x_i \in Y(t) \setminus \{t\}$ заменяем на v_j и добавляем в S уравнение $t = v_j$.

(6) Если $Y(t)$ не содержит констант из V и при этом $\left|\bigcap\{W_i^{\perp} \mid x_i \in Y(t)\}\right| > 1$, то во всех уравнениях из S каждую переменную $x_i \in Y(t)\backslash\{t\}$ заменяем на t.

Шаг 3. Обработка уравнений и выход из цикла.

Сначала последовательно просматриваем в системе S все уравнения вида $E(t_p, t_q)$ и выполняем для каждого из них пункты (1)–(2), затем переходим к пунктам (3)–(4).

(1) Если уравнение имеет вид $E(x_i, x_i)$ или $E(v_i, v_j)$, где $\langle v_i, v_j\rangle \notin E$, то процедура останавливается и выдаёт сообщение о несовместности системы.

(2) Если уравнение имеет вид $E(x_i, v_j)$ и при этом $v_j \notin W_i$ (то есть уравнение $E(x_i, v_j)$ получено на шаге 2), множества W_i и $W_i^{\perp}$ переопределяются следующим образом:

$$W_i = W_i \cup \{v_j\}, \quad W_i^{\perp} = W_i^{\perp} \cap \{v_j\}^{\perp},$$

где $\{v_j\}^{\perp}$ — множество вершин графа Γ, смежных с v_j. Если после переопределения оказалось, что $W_i^{\perp} = \varnothing$, то процедура останавливается и выдаёт сообщение о несовместности системы. Если же $W_i^{\perp} = \{v_m\}$ для некоторого v_m, то добавляем в систему S уравнение $x_i = v_m$.

(3) Для каждой переменной x_i такой, что $Y(x_i) = \varnothing$, находим терм t, для которого $x_i \in Y(t)$. Если t — это константа v_j, полагаем $W_i^{\perp} = \{v_j\}$. Если же t — это переменная x_m, то устанавливаем $W_i^{\perp} = W_m^{\perp}$.

(4) Проверяем, содержит ли система S уравнения с равенством. Если в S осталось хотя бы одно уравнение с равенством, процедура возвращается на шаг 1. Если равенств в S больше нет, процедура завершает работу, возвращая обновлённую систему уравнений S, множества W_i, $W_i^{\perp}$ и классы $Y(t_i)$. Если при этом в системе S не осталось уравнений с равенством и нет уравнений вида $E(x_i, x_j)$, то процедура переходит в заключительное состояние и выдаёт сообщение «система совместна».

Покажем, что условиям (а)–(г) удовлетворяют не только входные данные процедуры ELIM, но и её текущие данные в начале каждой итерации состоящего из шагов 1–3 цикла.

Лемма 1. *На шаге 3 процедуры ELIM в момент проверки условия из пункта (4) для текущих данных, используемых процедурой, справедливы указанные в ее описании условия (а)–(г).*

Доказательство. Справедливость условия (а) очевидно следует из определений на шаге 1.

Докажем справедливость условия (б). Перед запуском процедуры данное условие выполняется в силу свойств входных данных. Допустим, перед началом шага 1 условие верно. Тогда после шага 1 оно тоже останется справедливым, поскольку шаг 1 не изменяет множества вида W_i и уравнения вида $E(t_i, t_j)$. Шаг 2 тоже не изменяет множества вида W_i, но делает замены переменных в уравнениях в пунктах (4)–(6). Пусть после

шага 3 для переменной x_i справедливо $Y(x_i) \neq \varnothing$ и пусть $v_j \in W_i$. Если условие $v_j \in W_i$ было верно еще перед началом шага 3, то, поскольку шаг 2 не изменяет W_i, заключаем, что после шага 1 имеет место $E(x_i, v_j) \in S$. Следовательно, так как после шага 1 $Y(x_i) \neq \varnothing$ и $v_j \in W_i$, уравнение $E(x_i, v_j)$ не должно измениться на шагах 2 и 3, т.е. после шага 3 оно по-прежнему будет лежать в системе. Если элемент v_j попал в W_i непосредственно на шаге 3, то по построению в пункте (2) шага 3 заключаем, что после шага 3 верно $E(x_i, v_j) \in S$. Справедливость включения $\{v_j \mid E(x_i, v_j) \in S\} \subseteq W_i$ после шага 3 вытекает непосредственно из пункта (2) этого шага.

Докажем справедливость условия (в). Перед запуском процедуры данное условие выполняется в силу свойств входных данных. Допустим, перед началом шага 1 условие справедливо. Тогда после последовательного исполнения шагов 1 и 2 оно тоже останется справедливым, поскольку эти шаги не изменяет множества W_i и $W_i^{\perp}$. Шаг 3 в случае, когда $Y(x_i) \neq \varnothing$ и $W_i \neq \varnothing$, может изменить множества W_i и $W_i^{\perp}$ только в пункте (2), поэтому будем доказывать требуемое свойство индукцией по числу переопределений множеств W_i и $W_i^{\perp}$ в пункте (2) на шаге 3. Пусть W_i и $W_i^{\perp}$ — это множества до переопределения, для которых требуемое свойство уже доказано, а $\overline{W}_i$ и $\overline{W}_i^{\perp}$ — это множества после переопределения в пункте (2). Тогда $\overline{W}_i = W_i \cup \{v_j\}$ и $\overline{W}_i^{\perp} = W_i^{\perp} \cap \{v_p \in V \mid \langle v_j, v_p \rangle \in E\}$. Нетрудно, используя индукционное предположение, понять, что $\overline{W}_i^{\perp} \subseteq \{v_q \in V \mid \forall v_p \in \overline{W}_i (\langle v_q, v_p \rangle \in E)\}$. С другой стороны, если $\langle v_q, v_j \rangle \in E$ и $\langle v_q, v_p \rangle \in E$ для всех $v_p \in W_i$, то по индукции $v_q \in W_i^{\perp}$ и, значит, $v_q \in W_i^{\perp} \cap \{v_p \in V \mid \langle v_j, v_p \rangle \in E\}$, то есть $v_q \in \overline{W}_i^{\perp}$.

Докажем справедливость условия (г). Пусть после шага 3 выполняется $Y(x_i) \neq \varnothing$ и $W_i = \varnothing$. Следовательно, утверждения $Y(x_i) \neq \varnothing$ и $W_i = \varnothing$ были справедливы всегда, начиная с запуска процедуры. Следовательно, в силу свойства (б), в системе не было уравнений вида $E(x_i, v_j)$ на всех шагах, начиная с запуска процедуры. Значит, в силу свойств входных данных и построений в пункте (2) на шаге 3, от момента запуска и до рассматриваемого момента было справедливо равенство $W_i^{\perp} = V$. □

Докажем, что множество решений системы $S \cup \{t = t' \mid \exists t_i \in V \cup X (t, t' \in Y(t_i))\}$ является инвариантом процедуры ELIM.

Лемма 2. *Пусть процедура ELIM была запущена на входных системе уравнений S_0 и классах эквивалентности $Y_0(t_i)$. Тогда в любой момент работы процедуры для текущих системы S и классов эквивалентности $Y(t_i)$ система уравнений $S \cup \{t = t' \mid \exists t_i \in V \cup X (t, t' \in Y(t_i))\}$ эквивалентна системе $S_0 \cup \{t = t' \mid \exists t_i \in V \cup X (t, t' \in Y_0(t_i))\}$.*

Доказательство. Покажем, что это свойство сохраняется на каждом шаге алгоритма.

Для шага 1 это очевидно: при каждом удалении равенства из системы S соответствующие классы эквивалентности объединяются и, значит, информация о равенстве этих термов сохраняется.

Во время выполнения шага 2 в пунктах (1)–(3), а также шага 3 в пунктах (1), (3)–(4) система и классы эквивалентности не изменяются.

В пунктах (4) и (6) шага 2, в силу нашей нумерации термов, после проверки соответствующих условий для $Y(t)$ во всех уравнениях происходит замена каждой переменной $x_i \in Y(t) \setminus \{t\}$ на терм t. Заметим, что в рассматриваемых пунктах текущая система не может содержать уравнения вида $x_i = t_m$. Поэтому каждая такая замена реализуется путём удаления из S уравнения вида $E(x_i, t_m)$ и добавления в S уравнения $E(t, t_m)$. Так как в процессе выполнения шага 2 классы эквивалентности не изменяются, множество $\{t = t' \mid \exists t_j \in V \cup X (t, t' \in Y(t_j))\}$ содержит уравнение $x_i = t$ как до, так и после указанной замены. Отсюда следует, что при любом означивании переменных подсистемы $\{E(x_i, t_m)\} \cup \{x_i = t\}$ и $\{E(t, t_m)\} \cup \{x_i = t\}$ могут выполняться только одновременно.

В пункте (5) шага 2 терм t является переменной, выполняется условие $\bigcap\{W_i^{\perp} \mid x_i \in Y(t)\} = \{v_j\}$ и во всех уравнениях происходит замена каждой переменной $x_i \in Y(t) \setminus \{t\}$ на v_j, при этом в систему добавляется уравнение $t = v_j$. В этом пункте система тоже не может содержать уравнения вида $x_i = t_m$. Поэтому каждое из указанных преобразований реализуется путём удаления из S уравнения вида $E(x_i, t_m)$ и добавления уравнений $E(v_j, t_m)$ и $t = v_j$. Можно считать, что означивания всех переменных из $Y(t)$ равны друг другу, в противном случае множество уравнений $\{x_i = x_p \mid x_i, x_p \in Y(t)\}$ будет невыполнимым как до, так и после исполнения пункта (5). Если означить все переменные $x_i \in Y(t)$ значением v_j, то подсистемы $\{E(x_i, t_m)\}$ и $\{E(v_j, t_m)\} \cup \{t = v_j\}$ могут выполняться только одновременно. Допустим, в качестве значения всех переменных из $Y(t)$ выбрана константа $v_s \neq v_j$. В силу леммы 1, из условия $\bigcap\{W_i^{\perp} \mid x_i \in Y(t)\} = \{v_j\}$ следует, что найдутся переменная $x_i \in Y(t)$ и константа v_p такие, что в системе содержится уравнение $E(x_i, v_p)$, но при этом вершина v_s не смежна вершине v_p в исходном графе. Тогда при означивании переменных t и x_i значением v_s уравнение $E(x_i, v_p)$ не будет выполняться, так же, как и уравнение $t = v_j$.

В пункте (2) шага 3 система может измениться только при условии $W_i^{\perp} = \{v_m\}$, при этом в систему добавляется уравнение $x_i = v_m$. Тогда при означивании переменной x_i значением v_m утверждение леммы очевидно. В противном случае, если выбрать в качестве значения константу $v_s \neq v_m$, то в силу леммы 1 из условия $W_i^{\perp} = \{v_m\}$ следует, что найдётся константа v_q такая, что в системе содержится уравнение $E(x_i, v_q)$, но при этом вершина v_s не смежна вершине v_q в исходном графе. Тогда уравнение $E(x_i, v_q)$ содержится в системе как до, так и после исполнения рассматриваемого пункта, и при этом оно будет ложным, если означить переменную x_i значением v_s. □

Лемма 3. *Если процедура ELIM возвращает сообщение о совместности или несовместности системы, то оно является корректным.*

Доказательство. Устанавливаемая в пункте (1) шага 2 несовместность системы $S' = S \cup \{t = t' \mid \exists t_i \in V \cup X (t, t' \in Y(t_i))\}$ следует из попадания v_i и v_j в один класс эквивалентности. Это приводит к появлению в S' уравнения $v_i = v_j$, что противоречит условию попарного различия всех констант.

В пунктах (2) и (3) шага 2 несовместность системы S' следует из того, что ни одна вершина графа Γ не может быть значением ни одной переменной $x_i \in Y(t)$.

Несовместность системы S' в пункте (1) шага 3 следует из противоречия с условиями на граф Γ, а в пункте (2) того же шага — из невозможности присвоить подходящее значение переменной x_i.

Если в пункте (4) шага 3 процедура переходит в заключительное состояние и сообщает о совместности системы, то в текущей системе отсутствуют равенства и уравнения вида $E(x_i, x_j)$, а все множества $W_i^{\perp}$ не пусты. Следовательно, выбор значения каждой из переменных x_i ограничен лишь множеством $W_i^{\perp}$ и не зависит от выбора значений для переменных, принадлежащих другим классам $Y(t)$. Таким образом, в качестве решения можно выбрать любой набор $\langle u_1, \ldots, u_k \rangle \in W_1^{\perp} \times \cdots \times W_k^{\perp}$, удовлетворяющий условию $\forall i \forall j \forall p (x_i, x_j \in Y(t_p) \longrightarrow u_i = u_j)$. □

Предложение 2. *Время работы процедуры ELIM составляет* $\mathcal{O}(l^4)$.

Доказательство. На шаге 1 для каждого равенства $t_i = t_j$, где $1 \leqslant i, j \leqslant n + k$, объединение классов эквивалентности $Y(t_i)$ и $Y(t_j)$ требует времени $\mathcal{O}(l)$, следовательно, общее время исполнения шага 1 оценивается как $\mathcal{O}(l^2)$.

На шаге 2 пункт (1) выполняется за время $\mathcal{O}(l)$. В пунктах (2) и (3) шага 2 для фиксированного класса $Y(t_i)$ проверка соответствующего условия выполняется за время $\mathcal{O}(l^{1.5})$. Поскольку количество таких проверок составляет $\mathcal{O}(n+k) = \mathcal{O}(l^{0.5})$, общее время работы данных пунктов оценивается как $\mathcal{O}(l^2)$. В пунктах (4)–(6) шага 2 для каждого класса $Y(t_i)$, помимо проверки аналогичного по сложности условия, выполняется замена переменных и добавление равенств, которые можно реализовать за время $\mathcal{O}(l^2)$. Это приводит к верхней оценке $\mathcal{O}(l^3)$ для данных пунктов. Таким образом, общая вычислительная сложность шага 2 составляет $\mathcal{O}(l^3)$.

На шаге 3 пункт (1) выполняется за время $\mathcal{O}(l)$. В пункте (2) шага 3 для каждого из уравнений вида $E(x_i, v_j)$ переопределение множеств W_i и $W_i^{\perp}$ осуществляется за время $\mathcal{O}(l)$. Поскольку количество таких уравнений не превосходит $\mathcal{O}(l)$, общее время работы пункта (2) оценивается как $\mathcal{O}(l^2)$. Пункт (3) шага 3 можно выполнить за время $\mathcal{O}(l^2)$. Проверка на наличие равенств в пункте (4) шага 2 выполняется за время $\mathcal{O}(l)$. Таким образом, суммарное время исполнения шага 3 составляет $\mathcal{O}(l^2)$.

Итак, время выполнения одной итерации цикла, состоящей из шагов 1–3, составляет $\mathcal{O}(l^2) + \mathcal{O}(l^3) + \mathcal{O}(l^2) = \mathcal{O}(l^3)$. Данная итерация может быть повторена в случае наличия равенства в полученной системе S и всегда приводит к невозможности получения этого равенства в конце следующих итераций. Поскольку общее количество равенств не превосходит $\mathcal{O}(l)$, в худшем случае итерация может быть повторена $\mathcal{O}(l)$ раз. Таким образом, суммарное время исполнения всей процедуры составляет $\mathcal{O}(l^4)$. □

Далее опишем нашу основную процедуру проверки совместности конечных систем диофантовых уравнений над конечными двудольными графами. Она объединяет приведённые ранее процедуры INIT и ELIM, а также включает дополнительный недетерминированный шаг, отвечающий непосредственно за проверку совместности.

Процедура CONSIST.

Входными данными процедуры являются двудольный граф Γ и система уравнений S. Результатом её работы является вывод о совместности или несовместности заданной системы уравнений. Работа процедуры заключается в исполнении следующих трёх последовательных этапов.

Этап 1. Инициализация.

На этом этапе выполняется запуск процедуры INIT на входных данных, в результате чего формируются множества W_i, $W_i^{\perp}$ и классы эквивалентности $Y(t_i)$. Если процедура INIT обнаруживает несовместность, она выводит соответствующее сообщение, которое переносится на основную процедуру.

Этап 2. Упрощение системы.

Легко видеть, что выходные данные процедуры INIT удовлетворяют условиям (а)–(г), предъявляемым к входным данным процедуры ELIM. Поэтому далее запускается процедура ELIM на данных, полученных на предыдущем этапе. В результате работы процедуры обновляются система S, множества W_i, $W_i^{\perp}$ и классы эквивалентности $Y(t_i)$. Если процедура ELIM сообщает о совместности или несовместности системы, это также переносится на основную процедуру, и она завершает свою работу с соответствующим сообщением.

Этап 3. Проверка совместности.

На этом этапе используется граф $H = \langle V_X, E_X \rangle$, где V_X — множество переменных, входящих в запись хотя бы одного уравнения в текущей системе S, а множество рёбер E_X определяется следующим образом

$$E_X = \{\langle x_i, x_j \rangle, \langle x_j, x_i \rangle \mid \text{уравнение } E(x_i, x_j) \text{ входит в систему } S\}.$$

(1) Представим граф H в виде матрицы смежности размера $k \times k$, записанной построчно на 7-й ленте в виде слова

$$\mathcal{H} = \triangleleft \mathcal{H}_{1,1} \dots \mathcal{H}_{1,k} \# \dots \# \mathcal{H}_{k,1} \dots \mathcal{H}_{k,k} \triangleright,$$

где $\mathcal{H}_{i,j} \in \{*, 0, 1\}$ для всех $1 \leqslant i, j \leqslant k$ и символ $\mathcal{H}_{i,j}$ определяется так

$$\mathcal{H}_{i,j} = \begin{cases} *, & \text{если } x_i \notin V_X \text{ или } x_j \notin V_X, \\ 1, & \text{если } x_i, x_j \in V_X \text{ и } E(x_i, x_j) \in S, \\ 0, & \text{если } x_i, x_j \in V_X \text{ и } E(x_i, x_j) \notin S. \end{cases}$$

Такое представление позволит обрабатывать граф, игнорируя неиспользуемые вершины. При выполнении стандартных алгоритмов, таких как поиск в ширину [2], строки и элементы, содержащие символ $*$, будут пропускаться, что исключает их влияние на процесс вычислений, но при этом позволяет в будущем восстановить номера переменных x_i.

(2) Применим к графу H стандартный поиск в ширину [2, § 22.2]. В процессе поиска в ширину граф H разбивается на компонены связности $H_1, H_2, \ldots, H_m$, и в каждой из компонент поиск начинается с произвольной стартовой вершины. На 8-й ленте изначально для каждой вершины графа H записывается символ $*$, обозначающий, что вершина ещё не была посещена. Очередь для обработки вершин графа H хранится на 9-й ленте в виде последовательности их номеров. В процессе обхода для каждой вершины x_i, помещаемой в очередь, на 8-й ленте мы сохраняем чётность длины пути от стартовой вершины до x_i, а именно: записываем на ленте символ 0, если длина пути чётная, и записываем символ 1, если длина пути нечётная. Таким образом, получаем разбиение V_X на два непустых подмножества V_0 и V_1, которые содержат вершины, находящиеся соотвественно на чётном и нечётном расстоянии от стартовой вершины в своей компоненте связности.

(3) Далее проверяем, является ли граф H двудольным относительно разбиения $V_X = V_0 \cup V_1$. Для этого убеждаемся в том, что в каждом из подмножеств V_0 и V_1 отсутствуют рёбра, то есть для любых $x_i, x_j \in V_0$ и любых $x_i, x_j \in V_1$ справедливо условие $\langle x_i, x_j \rangle \notin E_X$. Если такое ребро обнаруживается, то граф H не является двудольным, поэтому процедура останавливается и выдаёт сообщение о несовместности системы. В противном случае переходим к выполнению следующего пункта.

(4) Поскольку граф H двудольный, разбиение $V_X = V_0 \cup V_1$ позволяет выделить множество V_0 как доминирующее. Действительно, в виду построения каждая вершина из V_1 смежна в графе H хотя бы с одной вершиной из V_0, а все изолированные вершины изначально попадут в множество V_0. Без ограничения общности можно считать, что $V_0 = \{x_1, x_2, \ldots, x_p\}$. Будем называть переменные из V_0 *свободными*, а остальные — *зависимыми*. Далее алгоритм должен недетерминированно угадать значения свободных переменных, выбрав набор вида $\overline{u} = \langle u_1, \ldots, u_p \rangle \in W_1^{\perp} \times \cdots \times W_p^{\perp}$. Набор $\overline{u}$ будет представлен на 10-й ленте в виде слова

$$\Lambda = \triangleleft \Lambda_{1,1} \ldots \Lambda_{1,n} \# \ldots \# \Lambda_{p,1} \ldots \Lambda_{p,n} \triangleright,$$

где $\Lambda_{i,j} \in \{0, 1\}$ и $\Lambda_{i,j} = 1$ будет обозначать, что мы присваиваем переменной x_i значение v_j. Для выбора набора $\overline{u}$ процедура проходит по

слову $\mathcal{W}^{\perp}$ на 5-й ленте и для каждой переменной $x_i \in V_0$ недетерминированно выбирает ровно одно её значение $v_j \in W_i^{\perp}$. Для этого в соответствующем фрагменте $\Lambda_{i,1} \dots \Lambda_{i,n}$ на 10-й ленте устанавливается ровно одна единица, указывающая на выбранное значение v_j. Таким образом, машина Тьюринга порождает одну из нескольких ветвей вычислений, каждая ветвь соответствует ровно одному набору $\overline{u}$. Далее процедура переходит к следующему шагу, где необходимо выполнить проверку на совместность выбранных значений.

(5) Для выбранного в предыдущем пункте набора $\overline{u} \in W_1^{\perp} \times \dots \times W_p^{\perp}$ составляется система уравнений

$$S_{\overline{u}} = S \cup \{x_1 = u_1, \dots, x_p = u_p\}.$$

Заметим, что в силу леммы 1 система $S_{\overline{u}}$ и текущие множества W_i, $W_i^{\perp}$, $Y(t_i)$ удовлетворяют требованиям, предъявляемым к входным данным процедуры ELIM. Поэтому далее для графа Г, системы $S_{\overline{u}}$, множеств W_i, $W_i^{\perp}$ и классов эквивалентности $Y(t_i)$ запускается процедура ELIM. Отметим, что после такого запуска процедуры ELIM в результирующей системе не останется уравнений вида $E(x_i, x_j)$, что гарантирует либо сообщение о совместности, либо сообщение о несовместности системы

$$S_{\overline{u}} \cup \{t = t' \mid \exists t \in V \cup X (t, t' \in Y(t_i))\}.$$

Если для рассматриваемого набора $\overline{u}$ вызов процедуры ELIM выдаст сообщение о совместности системы $S_{\overline{u}} \cup \{t = t' \mid \exists t \in V \cup X (t, t' \in Y(t_i))\}$, то наша основная процедура останавливается, переходит в заключительное состояние и выдает сообщение «система совместна». Если же подобный вызов процедуры ELIM выдаст сообщение о несовместности системы $S_{\overline{u}} \cup \{t = t' \mid \exists t \in V \cup X (t, t' \in Y(t_i))\}$, то наша основная процедура просто завершает свою работу вдоль данной ветви вычислений в некотором незаключительном состоянии. В частности, это означает, что если для каждого набора $\overline{u} \in W_1^{\perp} \times \dots \times W_p^{\perp}$ система $S_{\overline{u}} \cup \{t = t' \mid \exists t \in V \cup X (t, t' \in Y(t_i))\}$ оказалась несовместной, то несовместной будет и исходная система S.

Лемма 4. *Процедура CONSIST корректно проверяет систему уравнений на совместность.*

Доказательство. Если процедура CONSIST выдала сообщение о несовместности системы во время инициализации данных, то это может произойти только при условии $W_i^{\perp} = \varnothing$ для некоторого $1 \leqslant i \leqslant k$. В таком случае, очевидно, система действительно не является совместной.

Корректность сообщений о совместности или несоместности на этапе 2 следует из леммы 3.

Если процедура возвращает сообщение о несовместности системы во время исполнения пункта (3) этапа 3, это означает, что граф H оказался не двудольным, то есть в H существует цикл нечётной длины. Тогда для совместности текущей системы $S' = S \cup \{t = t' \mid \exists t_i \in V \cup X (t, t' \in Y(t_i))\}$

должен существовать набор значений переменных x_i, который удовлетворяет уравнениям, составляющим этот нечётный цикл. В силу свойств процедуры INIT и леммы 2, система S' эквивалентна исходной системе S. Следовательно, в ней также должен быть цикл нечётной длины, что противоречит условию о двудольности графа Г.

Во время исполнения пункта (5) этапа 3 свободным переменным присваивается определённый набор значений, таких что $x_i \in W_i^{\perp}$. Поскольку дальнейший вызов ELIM гарантированно приводит к окончанию работы процедуры с сообщением о совместности или несовместности, корректность финального решения следует из леммы 3. Таким образом, если для какого-либо набора значений свободных переменных процедура ELIM выдаст сообщение «система совместна», то существует такое означивание переменных, при котором все уравнения текущей системы выполняются. Тогда по лемме 2 это означивание приводит к выполнению уравнений исходной системы, что подтверждает её совместность.

В противном случае, если процедура ELIM выдаёт сообщение о несовместности для всех возможных наборов значений свободных переменных, то система не имеет такого означивания переменных, при котором все уравнения выполняются. Следовательно, исходная система несовместна. □

Предложение 3. *Время работы процедуры CONSIST составляет* $\mathcal{O}(l^4)$.

Доказательство. Согласно предложению 1, выполнение первого этапа занимает время $\mathcal{O}(l^2)$. В силу предложения 2, второй этап выполняется за время $\mathcal{O}(l^4)$.

Заполнение матрицы смежности графа H и 8-й ленты проводится за время $\mathcal{O}(l)$. Затем выполняется модифицированный поиск в ширину, сложность работы которого на машине Тьюринга можно оценить как $\mathcal{O}(l^2)$, после чего проверяется двудольность графа, что так же можно выполнить за время $\mathcal{O}(l^2)$. Таким образом, время выполнения пунктов (1)–(3) на этапе 3 укладывается в $\mathcal{O}(l^2)$.

Построение слова Λ в пункте (4) включает в себя обход слова $\mathcal{W}^{\perp}$, поэтому занимает время $\mathcal{O}(l)$. Построение системы S_q в пункте (5) производится за время $\mathcal{O}(l)$. Для выбранной конфигурации запускается процедура ELIM, время работы которой, как уже было показано, составляет $\mathcal{O}(l^4)$. Значит, совокупное время исполнения пунктов (4) и (5) на этапе 3 составляет $\mathcal{O}(l^4)$.

Таким образом, общее время работы процедуры CONSIST также оценивается как $\mathcal{O}(l^4)$. □

4 Доказательство NP-полноты проблемы

Чтобы установить NP-полноту проблемы совместности конечных систем диофантовых уравнений над конечными двудольными графами, воспользуемся тем, что проблема совместности аналогичной проблемы

для произвольных симметричных иррефлексивных графов является NP-полной [5], и построим её полиномиальное сведение к нашей проблеме.

Для этого опишем детерминированную полиномиальную процедуру, которая осуществляет требуемое сведение.

Процедура TRANSFORM.

Входными данными процедуры являются слова $\mathcal{A}$, $\mathcal{R}$ и $\mathcal{E}$, расположенными на 1-й, 2-й и 3-й лентах машины Тьюринга соответственно. Выходными данными являются слова $\mathcal{A}'$, $\mathcal{R}'$ и $\mathcal{E}'$, записываемые процедурой на 4-й, 5-й и 6-й лентах соответственно. Формат представления входных и выходных данных и их интерпретация описаны ранее в § 3.

Заметим, что входные данные могут иметь произвольный вид. Поэтому процедура сначала проверяет корректность входа. В случае обнаружения ошибки во входных данных процедура не выполняет преобразование, а прерывает свою работу и формирует на выходных лентах корректно заданную, но заведомо несовместную систему. Для этого она записывает граф, состоящий из одной вершины v_1, а также систему уравнений, содержащую единственное уравнение $E(v_1, v_1)$. Если же входные данные заданы корректно, то слова $\mathcal{A}$, $\mathcal{R}$ и $\mathcal{E}$ действительно кодируют некоторый граф Γ и систему S. В этом случае выходные данные $\mathcal{A}'$, $\mathcal{R}'$ и $\mathcal{E}'$ будут кодировать некоторый двудольный граф Γ' и новую систему S' такие, что совместность системы S над графом Γ будет эквивалентна совместности S' над Γ'.

Шаг 1. Проверка корректности входных данных.

Процедура последовательно проверяет, что слова $\mathcal{A}$, $\mathcal{R}$ и $\mathcal{E}$ действительно задают булевы квадратные матрицы. В частности, каждая матрица должна быть записана в виде строки, содержащей m блоков длины m, разделённых символами # и заключённых между символами $\lhd$ и $\rhd$, где $m = n$ для слова $\mathcal{A}$ и $m = n+k$ для слов $\mathcal{R}$ и $\mathcal{E}$. Значения n и k можно вычислить, исходя из длин первых блоков матриц $\mathcal{A}$ и $\mathcal{R}$, так как они однозначно задают размерность задачи. Процедура также проверяет, что каждый блок содержит только символы 0 и 1, общее количество блоков и символов-разделителей соответствует требуемому формату, а также что все три матрицы симметричны, причём матрица $\mathcal{A}$ имеет нулевую диагональ.

Если хотя бы одно из условий нарушено, выполнение процедуры прерывается, и на выходных лентах записывается несовместная система с одной петлёй, как было описано ранее. В случае успешного завершения проверки процедура переходит к следующему шагу.

Шаг 2. Преобразование графа.

В силу успешного завершения проверки на шаге 1, слово $\mathcal{A}$ кодирует некоторый симметричный иррефлексивный граф $\Gamma = \langle V, E\rangle$.

Для любой неупорядоченной пары $e = \{u, v\}$ вершин $u, v \in V$ таких, что $\langle u, v\rangle \in E$ введём новую вершину w_e, и заменим ребро $\langle u, v\rangle \in E$ исходного графа на пару новых рёбер $\langle u, w_e\rangle$ и $\langle w_e, v\rangle$.

Дополнительно вводятся две *индикаторные* вершины: вершина I_V, смежная со всеми вершинами из V, и вершина I_E, смежная со всеми вершинами вида w_e. Индикаторные вершины будут использоваться в качестве констант в новой системе уравнений и позволят явно различать исходные и добавленные вершины при проверке совместности.

Таким образом, процедура определяет граф $\Gamma' = \langle V', E' \rangle$ со следующими множествами вершин и ребёр:

$$V' = V \cup \{w_e \mid e = \{u, v\} \text{ и } \langle u, v\rangle \in E\} \cup \{I_V, I_E\},$$

$$E' = \{\langle u, w_e\rangle, \langle w_e, u\rangle, \langle v, w_e\rangle, \langle w_e, v\rangle \mid e = \{u, v\} \text{ и } \langle u, v\rangle \in E\} \cup$$
$$\cup \{\langle u, I_V\rangle, \langle I_V, u\rangle \mid u \in V\} \cup \{\langle w_e, I_E\rangle, \langle I_E, w_e\rangle \mid e = \{u, v\} \text{ и } \langle u, v\rangle \in E\}.$$

Каждое ребро графа Γ' соединяет только вершины из двух непересекающихся множеств $V \cup \{I_E\}$ и $\{w_e \mid e{=}\{u, v\} \text{ и } \langle u, v\rangle{\in}E\} \cup \{I_V\}$, поэтому Γ' является двудольным.

Опишем способ кодирования графа Γ' на лентах машины Тьюринга. Пусть как и ранее вершины исходного графа Γ имеют фиксированную нумерацию $V = \{v_1, \ldots, v_n\}$. Процедура последовательно просматривает слово $\mathcal{A}$ на 1-й ленте и для каждой пары индексов $1{\leqslant}i{<}j{\leqslant}n$ проверяет наличие ребра $\langle v_i, v_j\rangle \in E$. Если $\mathcal{A}_{i,j} = 1$, то для пары $e = \{v_i, v_j\}$ на 7-й ленте создаётся кодирующая эту пару вспомогательная запись в виде слова

$$B = b_1 b_2 \ldots b_n,$$

где $b_s = 1$, если $s \in \{i, j\}$, и $b_s = 0$ в противном случае. После обхода всей матрицы смежности $\mathcal{A}$ процедура формирует список, хранящийся на 7-й ленте в виде слова

$$\triangleleft B^1 \# B^2 \# \ldots \# B^r \triangleright,$$

где $r = |E|/2$ — количество ребёр в графе Γ без учёта их ориентации. Обозначим через w_{e_s} добавленную вершину, которая соответствует ребру исходного графа, информация о котором закодирована в записи B^s.

Зафиксируем следующую нумерацию вершин нового графа Γ':

$$v'_1 = v_1, \ldots, v'_n = v_n,$$
$$v'_{n+1} = w_{e_1}, \ldots, v'_{n+r} = w_{e_r},$$
$$v'_{n+r+1} = I_V, \quad v'_{n+r+2} = I_E.$$

После этого на 4-й ленте построчно записывается матрица смежности $\mathcal{A}'$ размера $(n + r + 2) \times (n + r + 2)$. Элементы $\mathcal{A}'$ формируются по следующим правилам:

(1) Для каждой записи B^s на 7-й ленте, соответствующей паре $e_s = \{v_i, v_j\}$ смежных вершин исходного графа Γ, полагаем

$$\mathcal{A}'_{n+s,i} = \mathcal{A}'_{i,n+s} = \mathcal{A}'_{n+s,j} = \mathcal{A}'_{j,n+s} = 1.$$

(2) Для индикаторной вершины $I_V = v'_{n+r+1}$ и для каждого $1 \leqslant i \leqslant n$ полагаем

$$\mathcal{A}'_{n+r+1,i} = \mathcal{A}'_{i,n+r+1} = 1.$$

(3) Для индикаторной вершины $I_E = v'_{n+r+2}$ и каждого $1 \leqslant s \leqslant r$ полагаем

$$\mathcal{A}'_{n+r+2,n+s} = \mathcal{A}'_{n+s,n+r+2} = 1.$$

(4) Остальные элементы $\mathcal{A}'_{i,j}$ приравниваются нулю.

Шаг 3. Построение новой системы уравнений.

В силу успешного завершения проверки на шаге 1, слова $\mathcal{R}$ и $\mathcal{E}$ кодируют некоторую систему уравнений S сигнатуры σ_V над некоторым множеством переменных $X = \{x_1, \ldots, x_k\}$.

Для каждой неупорядоченной пары $c = \{t, t'\}$ термов из множества $V \cup X$ таких, что в системе S присутствуют уравнения $E(t, t')$ и $E(t', t)$, введём новую переменную y_c и заменим указанные уравнения на следующие четыре:

$$E(t, y_c), E(y_c, t'), E(t', y_c), E(y_c, t).$$

Такая замена обеспечивает согласованность системы S' со структурой преобразованного графа.

Далее для всех $1 \leqslant s \leqslant k$ и для каждой новой переменной y_c добавим в систему уравнения

$$E(x_s, I_V), E(y_c, I_E).$$

Эти уравнения связывают переменные с индикаторными вершинами I_V и I_E, что позволяет явно задать их роль в системе.

Уравнения с равенствами остаются без изменений. Таким образом, итоговая система уравнений имеет вид:

$$S' = \{E(t, y_c), E(y_c, t), E(t', y_c), E(y_c, t'), E(y_c, I_E), E(I_E, y_c) \mid c{=}\{t, t'\} \text{ и } E(t, t') \in S\} \cup \{E(x_s, I_V), E(I_V, x_s) \mid 1 \leqslant s \leqslant k\} \cup \{t{=}t' \mid t{=}t' \in S\}.$$

Опишем способ кодирования системы S' на лентах машины Тьюринга. Как и ранее термы из $V \cup X$ имеют следующую нумерацию: $t_i = v_i$ и $t_{n+j} = x_j$ для всех $1 \leqslant i \leqslant n$ и $1 \leqslant j \leqslant k$. Процедура последовательно просматривает слово $\mathcal{R}$ на 2-й ленте и для каждой пары индексов $1 \leqslant i < j \leqslant n+k$ проверяет, содержится ли уравнение $E(t_i, t_j)$ в системе S. Если $\mathcal{R}_{i,j} = 1$, то для пары $c = \{t_i, t_j\}$ на 8-й ленте создаётся кодирующая эту пару вспомогательныя запись в виде слова

$$B = b_1 b_2 \ldots b_{n+k},$$

где $b_s = 1$, если $s \in \{i, j\}$ и $b_s = 0$ в противном случае. По завершении обхода слова $\mathcal{R}$ процедура получит список, хранящийся на 8-й ленте в виде слова

$$\triangleleft B^1 \# B^2 \# \ldots \# B^m \triangleright,$$

где m — количество введённых новых переменных вида y_c. Обозначим через y_{c_s} новую переменную, которая соответствует уравнению, информация о котором закодирована в записи B^s.

Зафиксируем следующую нумерацию всех термов из множества $V' \cup X \cup \{y_{c_1}, \ldots, y_{c_m}\}$:

$$t'_1 = v'_1, \ldots, t'_{n+r+2} = v'_{n+r+2},$$
$$t'_{n+r+3} = x_1, \ldots, t'_{n+r+k+2} = x_k,$$
$$t'_{n+r+k+3} = y_{c_1}, \ldots, t'_{n+r+k+m+2} = y_{c_m}.$$

Далее процедура построчно формирует на 5-й ленте матрицу $\mathcal{R}'$ размера $(n+r+k+m+2) \times (n+r+k+m+2)$. Элементы $\mathcal{R}'$ определяются по следующим правилам:

(1) Для каждой записи B^s на 8-й ленте, соответствующей новой переменной y_{c_s} и уравнению $E(t_i, t_j)$ в исходной системе S, полагаем

$$\mathcal{R}'_{z,p} = \mathcal{R}'_{p,z} = \mathcal{R}'_{z,q} = \mathcal{R}'_{q,z} = 1,$$

где $t'_z = y_{c_s}$, $t'_p = t_i$ и $t'_q = t_j$.

(2) Для каждой исходной переменной x_s, $1 \leqslant s \leqslant k$, полагаем

$$\mathcal{R}'_{p,\alpha} = \mathcal{R}'_{\alpha,p} = 1,$$

где $t'_p = x_s$ и $t'_\alpha = I_V$.

(3) Для каждой новой переменной y_{c_s}, $1 \leqslant s \leqslant m$, полагаем

$$\mathcal{R}'_{z,\beta} = \mathcal{R}'_{\beta,z} = 1,$$

где $t'_z = y_{c_s}$ и $t'_\beta = I_E$.

(4) Остальные элементы $\mathcal{R}'_{i,j}$ приравниваются нулю.

Наконец, процедура построчно формирует на 6-й ленте матрицу $\mathcal{E}'$ размера $(n+r+k+m+2) \times (n+r+k+m+2)$. Так как уравнения с равенствами остаются неизменными, необходимо лишь адаптировать их к новой размерности и обновить индексы термов. Для этого процедура просматривает слово $\mathcal{E}$ на 3-й ленте и для каждого уравнения $t_i = t_j$ обновляет на 6-й ленте матрицу $\mathcal{E}'$, добавляя это уравнение в новую систему следующим образом:

$$\mathcal{E}'_{p,q} = \mathcal{E}'_{q,p} = 1,$$

где $t'_p = t_i$ и $t'_q = t_j$.

На этом процедура завершает свою работу.

Перейдём к доказательству свойств процедуры TRANSFORM.

Проблему L совместности конечных систем диофантовых уравнений над произвольными конечными графами мы формально представляем как множество всех троек $\langle \mathcal{A}, \mathcal{R}, \mathcal{E} \rangle$ слов в алфавите $\{0, 1, \lhd, \rhd, \#\}$, кодирующих совместные системы в соответствии с приведённым в § 3 определением. Аналогично определяется *проблема L' совместности конечных систем диофантовых уравнений над конечными двудольными графами.*

Лемма 5. *Пусть процедура TRANSFORM была запущена на тройке входных слов $\mathcal{A}$, $\mathcal{R}$, $\mathcal{E}$ и после остановки выдала тройку выходных слов $\mathcal{A}'$, $\mathcal{R}'$, $\mathcal{E}'$. Тогда $\langle \mathcal{A}, \mathcal{R}, \mathcal{E} \rangle \in L$, если и только если $\langle \mathcal{A}', \mathcal{R}', \mathcal{E}' \rangle \in L'$.*

Доказательство. Если тройка $\langle \mathcal{A}, \mathcal{R}, \mathcal{E} \rangle$ не удовлетворяет условиям корректности, проверяемым на шаге 1, то очевидно $\langle \mathcal{A}, \mathcal{R}, \mathcal{E} \rangle \notin L$ и по построению $\langle \mathcal{A}', \mathcal{R}', \mathcal{E}' \rangle \notin L'$. Поэтому далее будем считать, что тройке $\langle \mathcal{A}, \mathcal{R}, \mathcal{E} \rangle$ соответствуют граф $\Gamma = \langle V, E \rangle$ и система S уравнений сигнатуры σ_V над множеством переменных $X = \{x_1, \ldots, x_k\}$. Тогда по построению на шагах 2 и 3 тройке $\langle \mathcal{A}', \mathcal{R}', \mathcal{E}' \rangle$ соответствуют двудольный граф $\Gamma' = \langle V', E' \rangle$ и система S' уравнений сигнатуры $\sigma_{V'}$ над множеством переменных $X' = X \cup \{y_c \mid c = \{t, t'\} \text{ и } E(t, t') \in S\}$.

Пусть $\langle \mathcal{A}, \mathcal{R}, \mathcal{E} \rangle \in L$, т.е. система S совместна над графом Γ. Следовательно, существует означивание $\gamma : X \to V$ переменных из X, при котором все уравнения из S истинны в Γ. Определим продолжение $\gamma' : X' \to V'$ означивания γ до означивания переменных из X' в графе Γ'. Рассмотрим произвольные термы t и t', связанные уравнением $E(t, t') \in S$. По построению неупорядоченной паре $c = \{t, t'\}$ соответствует новая переменная y_c, а уравнение $E(t, t')$ преобразуется (без учета симметричных уравнений) в два уравнения $E(t, y_c)$ и $E(y_c, t')$ в новой системе S'. Пусть $u = t[\gamma]$ и $v = t'[\gamma]$ — значения данных термов в графе Γ при означивании γ. Поскольку система совместна, вершины u и v соединены в графе Γ ребром $\langle u, v \rangle \in E$. Следовательно, для неупорядоченной пары $e = \{u, v\}$ в новом графе Γ' найдётся вершина w_e, которая смежна u и v уже в графе Γ'. В этом случае положим $\gamma'(y_c) = w_e$.

Тогда, если положить $\gamma'(x_i) = \gamma(x_i)$ для всех $1 \leqslant i \leqslant k$, то оба новых уравнения $E(t, y_c)$ и $E(y_c, t')$ будут истинны в Γ' при означивании γ'. Нетрудно видеть, что при определённом таким образом означивании γ' уравнения вида $E(x_i, I_V)$ для всех $1 \leqslant i \leqslant k$ и $E(y_c, I_E)$ для всех новых переменных y_c тоже будут истинны в Γ'. Уравнения вида $t = t'$ из исходной системы S, перенесённые в S' без изменений, также, очевидно, остаются истинными. Таким образом, система S' совместна над графом Γ', и значит, $\langle \mathcal{A}', \mathcal{R}', \mathcal{E}' \rangle \in L'$.

Пусть теперь $\langle \mathcal{A}', \mathcal{R}', \mathcal{E}' \rangle \in L'$, т.е. система S' совместна над графом Γ'. Следовательно, существует означивание $\gamma' : X' \to V'$ переменных из X', при котором все уравнения из S' истинны в Γ'. Рассмотрим произвольное уравнение $E(t, t') \in S$, которому соответствуют новая переменная y_c, введённая для неупорядоченной пары $c = \{t, t'\}$ термов из $V \cup X$. Пусть $u = t[\gamma']$, $v = t'[\gamma']$ и $w = y_c[\gamma']$ — значения термов t, t' и y_c в графе Γ' при означивании γ'. Поскольку в системе S' содержится уравнение $E(x_i, I_V)$ для любой переменной $x_i \in X$, а γ' обязанно действовать на множестве V тождественно, заключаем, что $u, v \in V$. Поскольку в системе S' есть уравнение $E(y_c, I_E)$, заключаем, что $w = w_e$ для некоторой неупорядоченной пары e смежных в графе Γ вершин. Однако, в системе S' также присутствуют уравнения $E(t, y_c)$, $E(y_c, t')$. Следовательно,

поскольку вершина w_e соотвествует единственной неупорядоченной паре вершин из V, заключаем, что $e = \{u, v\}$ и вершины u и v смежны в исходном графе Γ.

Отсюда следует, что $\gamma'(X) \subseteq V$ и если определить означивание $\gamma = \gamma' \restriction X : X \to V$, то все уравнения вида $E(t, t') \in S$ будут истинны в Γ при означивании γ. Так как уравнения вида $t = t'$ переносятся из S в S' без изменений, такое означивание также обеспечивает истинность всех уравнений с равенством. Таким образом, система S совместна над Γ, т.е. $\langle \mathcal{A}, \mathcal{R}, \mathcal{E} \rangle \in L$. □

Предложение 4. *Время работы процедуры TRANSFORM составляет* $\mathcal{O}(l^3)$.

Доказательство. На шаге 1 процедура проверяет корректность входных данных. Проверка длины слов и формата записи осуществляется за время $\mathcal{O}(l)$. Проверку на симметричность и иррефлексивность можно осуществить за время $O(l^2)$.

На шаге 2 процедура сначала строит список пар индексов $i < j$ таких, что вершины v_i и v_j смежны в графе Γ. Общее число таких пар не превышает $n(n-1)/2$, что оценивается как $\mathcal{O}(l)$. Длина одной записи B^s равна n, поэтому общая длина слова, кодирующего список пар индексов, оценивается как $\mathcal{O}(l^{1.5})$. Соответственно, время его построения тоже составляет $\mathcal{O}(l^{1.5})$.

Далее на шаге 2 алгоритм строит матрицу смежности $\mathcal{A}'$ размера $(n + r + 2) \times (n + r + 2)$, где $r = |E|/2$. Поскольку r ограничено сверху величиной $\mathcal{O}(l)$, общее число символов в записи слова $\mathcal{A}'$ не превосходит $\mathcal{O}(l^2)$. Вычисление одного символа $\mathcal{A}'_{i,j}$ в худшем случае требуется время $\mathcal{O}(l)$. Поэтому заполнение матрицы $\mathcal{A}'$ можно осуществить за время $\mathcal{O}(l^3)$.

На шаге 3 формируется список пар индексов $i < j$ таких, что термы t_i и t_j связаны уравнением $E(t_i, t_j)$ в системе S. Поскольку количество термов равно $n+k$, общее число таких пар не превышает $(n+k)(n+k-1)/2$. Длина одной записи B^s на этом шаге равна $n + k$, поэтому длина всего списка составляет $\mathcal{O}(l^{1.5})$ символов. Соответственно, его построение можно осуществить за время $\mathcal{O}(l^{1.5})$.

Затем на шаге 3 процедура строит две матрицы смежности $\mathcal{R}'$ и $\mathcal{E}'$ размера $(n+r+k+m+2) \times (n+r+k+m+2)$, где m — количество пар термов, связанных уравнением смежности в системе S. Обе величины r и m оцениваются как $\mathcal{O}(l)$, поэтому размер данных матриц составляет $\mathcal{O}(l^2)$ символов. Вычисление одного элемента для обеих матриц может потребовать в худшем случае время $\mathcal{O}(l)$. Поэтому общее время построения матриц $\mathcal{R}'$ и $\mathcal{E}'$ может быть выполнено за время $\mathcal{O}(l^3)$.

Таким образом, все этапы работы процедуры TRANSFORM выполняются за суммарное время, ограниченное сверху величиной $\mathcal{O}(l^3)$. □

Таким образом, процедура TRANSFORM выполняется за полиномиальное время. Это позволяет нам сформулировать и доказать следующий основной результат статьи.

Теорема 1. *Проблема совместности конечных систем диофантовых уравнений над конечными двудольными графами является NP-полной.*

Доказательство. Из леммы 4 и предложения 3 следует, что приведённая ранее недетерминированная процедура CONSIST распознаёт совместные системы за полиномиальное время. Следовательно, рассматриваемая нами проблема принадлежит классу NP.

NP-трудность задачи устанавливается с помощью полиномиального сведения проблемы совместности систем уравнений над произвольными графами, которая, как показано в [5], является NP-полной, к нашей проблеме. В силу леммы 5 и предложения 4 требуемое полиномиальное сведение осуществляется с помощью построенной нами процедуры TRANSFORM.

Таким образом, рассматриваемая проблема принадлежит классу NP, является NP-трудной и, следовательно, NP-полной. □

References

[1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley and Reading, London and Amsterdam, 1974. Zbl 0326.68005

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms*. 2nd ed., MIT and McGraw-Hill, Cambridge and London, 2001. Zbl 1047.68161

[3] E.Yu. Daniyarova, A.G. Myasnikov, V.N. Remeslennikov, *Algebraic Geometry over Algebraic Structures*, SB RAS, Novosibirsk, 2016.

[4] A.V. Il'ev, *Study of systems of equations over various classes of finite matroids*, Sib. Electr. Math. Reports, **19**:2 (2022), 1094–1102. Zbl 1573.68126

[5] A.V. Il'ev, V.P. Il'ev, *Algorithms for solving systems of equations over various classes of finite graphs*, Prikl. Diskretn. Mat., **54** (2021), 89–102. Zbl 1528.68298

[6] A.V. Il'ev, V.N. Remeslennikov, *Study of the compatibility of systems of equations over graphs and finding their general solutions*, Vestn. Omsk University, **4** (2017), 26–32.

[7] A.Yu. Nikitin, *Algebraic geometry and algorithms in the class of partially ordered sets*, Vestn. Omsk University, **29**:1 (2024), 23–32.

[8] A.Yu. Nikitin, A.N. Rybalov, *On complexity of the satisfiability problem of systems over finite posets*, Prikl. Diskretn. Mat., **39** (2018), 94–98. Zbl 1515.68220

Fedor Dmitrievich Kobzev
Novosibirs State University,
1, Pirogova st.,
Novosibirsk, 630090, Russia
Email address: f.kobzev@g.nsu.ru

Nurlan Talgatovich Kogabaev
Sobolev Institute of Mathematics,
4, Acad. Koptyug ave.,
Novosibirsk, 630090, Russia
Email address: kogabaev@math.nsc.ru