ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

УДК 510.52, 510.67, 519.17

АЛГОРИТМЫ РЕШЕНИЯ СИСТЕМ УРАВНЕНИЙ НАД РАЗЛИЧНЫМИ КЛАССАМИ КОНЕЧНЫХ ГРАФОВ[1]

А. В. Ильев*, В. П. Ильев**

**Институт математики им. С. Л. Соболева СО РАН, г. Омск, Россия,*
***Омский государственный университет им. Ф. М. Достоевского, г. Омск, Россия*

Целью работы является исследование и решение конечных систем уравнений над конечными неориентированными графами. Уравнениями над графами называются атомарные формулы языка L, состоящего из множества констант (вершин графа), бинарного предиката смежности вершин и предиката равенства. Доказано, что задача проверки совместности системы уравнений S от k переменных над произвольным обыкновенным n-вершинным графом Γ является $\mathcal{NP}$-полной. Подсчитана вычислительная сложность процедуры проверки совместности системы уравнений S над обыкновенным графом Γ и процедуры нахождения общего решения этой системы. Вычислительная сложность алгоритма решения системы уравнений S от k переменных над произвольным обыкновенным n-вершинным графом Γ, включающего эти процедуры, составляет $O(k^2n^{k/2+1}(k+n)^2)$ при $n \geqslant 3$. Выделены полиномиально разрешимые случаи: системы уравнений над деревьями, лесами, двудольными и полными двудольными графами, для решения которых предложены полиномиальные алгоритмы трудоёмкости $O(k^2n(k+n)^2)$.

Ключевые слова: *граф, система уравнений, вычислительная сложность.*

DOI 10.17223/20710410/53/6

ALGORITHMS FOR SOLVING SYSTEMS OF EQUATIONS OVER VARIOUS CLASSES OF FINITE GRAPHS

A. V. Il'ev*, V. P. Il'ev**

**Sobolev Institute of Mathematics SB RAS, Omsk, Russia*
***Dostoevsky Omsk State University, Omsk, Russia*

E-mail: artyom_iljev@mail.ru, iljev@mail.ru

The aim of the paper is to study and to solve finite systems of equations over finite undirected graphs. Equations over graphs are atomic formulas of the language L consisting of the set of constants (graph vertices), the binary vertex adjacency predicate and the equality predicate. It is proved that the problem of checking compatibility of a system of equations S with k variables over an arbitrary simple n-vertex graph Γ is $\mathcal{NP}$-complete. The computational complexity of the procedure for checking compatibility of a system of equations S over a simple graph Γ and the procedure for finding a general solution of this system is calculated. The computational complexity

[1]Работа первого автора поддержана грантом РНФ № 19-11-00209.

of the algorithm for solving a system of equations S with k variables over an arbitrary simple n-vertex graph Γ involving these procedures is $O(k^2 n^{k/2+1}(k+n)^2)$ for $n \geqslant 3$. Polynomially solvable cases are distinguished: systems of equations over trees, forests, bipartite and complete bipartite graphs. Polynomial time algorithms for solving these systems with running time $O(k^2 n(k+n)^2)$ are proposed.

Keywords: *graph, system of equations, computational complexity.*

Введение

Традиционным направлением исследований в алгебраической геометрии является решение систем уравнений над вещественными, комплексными, рациональными и целыми числами. Различным алгоритмическим проблемам, связанным с решением систем уравнений, посвящены многочисленные теоретические и практические исследования. Как правило, такие проблемы оказываются либо неразрешимыми, либо разрешимыми, но вычислительно трудными. К числу неразрешимых проблем относится проблема решения полиномиальных уравнений над кольцом целых чисел [1]. Среди разрешимых проблем, имеющих экспоненциальную сложность, можно выделить арифметику Пресбургера [2], проблему решения систем полиномиальных уравнений над полями комплексных и вещественных чисел [3] и проблему решения систем полиномиальных уравнений над конечными полями, которая эквивалентна $\mathcal{NP}$-полной проблеме выполнимости булевых формул [4].

В последние годы в алгебраической геометрии активно развивается направление исследований, в рамках которого изучаются системы уравнений над произвольными алгебраическими системами, где уравнениями являются атомарные формулы языка алгебраической системы [5]. И здесь также естественным образом возникают вопросы разрешимости и вычислительной сложности алгоритмических проблем. Например, доказано, что проблема совместности систем уравнений над конечными частично упорядоченными множествами является $\mathcal{NP}$-полной в том случае, когда проверяется существование решения, состоящего из попарно различных элементов частично упорядоченного множества [6], или что экзистенциальная теория класса всех конечных полей является $\mathcal{NP}$-трудной, а универсальная теория этого класса — co-$\mathcal{NP}$-трудной [7].

Многие алгоритмические проблемы связаны с проблемами разрешимости универсальных теорий различных классов алгебраических систем, изучение которых представляет значительный интерес в алгебраической геометрии. Для обыкновенных графов, являющихся важнейшими математическими объектами и используемых при решении большого числа практически важных задач оптимизации, вопросы разрешимости универсальных теорий подробно изучены в [8]. В свою очередь, эти вопросы имеют тесную связь с решением систем уравнений над конечными графами. Проверка принадлежности графа некоторому классу входит в качестве процедуры в алгоритмы, устанавливающие разрешимость универсальных теорий наследственных классов графов. Такая процедура может быть оформлена в виде решения конечной серии систем уравнений над графами.

В [9] предложены процедуры решения следующих задач:

1) проверка совместности системы уравнений S над конечным графом;
2) вычисление радикала системы S;
3) построение координатного графа системы S.

Отметим, что роль координатного графа системы уравнений S над графом аналогична роли общего решения системы линейных уравнений над полем в линейной алгебре. При этом координатный граф однозначно определяется радикалом системы S.

В настоящей работе доказано, что задача проверки совместности системы уравнений над произвольным конечным обыкновенным графом является $\mathcal{NP}$-полной. Выделены полиномиально разрешимые случаи: системы уравнений над деревьями, лесами, двудольными и полными двудольными графами. Конструктивно улучшена процедура проверки совместности системы уравнений S от k переменных над произвольным n-вершинным графом Γ и подобраны структуры данных, позволяющие реализовать все процедуры алгоритма решения системы уравнений S над графом Γ на ЭВМ. Подсчитана вычислительная сложность этого алгоритма, которая составляет $O(k^2 n^{k/2+1}(k+n)^2)$ при $n \geqslant 3$. Кроме того, предложены полиномиальные процедуры проверки совместности систем уравнений и нахождения их общего решения для класса двудольных графов, в который входят также деревья и леса, и класса полных двудольных графов. Доказано, что вычислительная сложность алгоритмов решения систем уравнений над двудольными и полными двудольными графами составляет $O(k^2 n(k+n)^2)$.

1. Предварительные сведения

Граф — это пара $\Gamma = (V, E)$, где V — непустое множество элементов, называемых *вершинами*; E — множество неупорядоченных пар различных элементов из V, называемых *рёбрами*. Если $(u, v) \in E$, то вершины u и v называются *смежными*. Граф *конечен*, если множество его вершин конечно, и *счётно бесконечен*, если множество его вершин счётно бесконечно. В данной работе рассматриваются только конечные обыкновенные графы, т. е. графы, не содержащие петель и кратных рёбер.

Граф называется *деревом*, если он не содержит циклов и является связным, и *лесом*, если он не содержит циклов и может быть несвязным. Граф называется *двудольным*, если множество его вершин можно разбить на два непересекающихся подмножества V_1 и V_2 так, что каждое ребро графа соединяет вершины из разных долей. Двудольный граф называется *полным двудольным*, если любая вершина из одной его доли смежна со всеми вершинами из другой доли.

Теперь дадим определение графа как алгебраической системы.

Граф — это алгебраическая система $\Gamma = \langle V, \mathrm{L} \rangle$, носитель которой V — непустое не более чем счётное множество, а язык $\mathrm{L} = \{E(x,y) \cup (x = y) \cup \mathrm{C}\}$ состоит из множества констант $\mathrm{C} \subseteq V$, бинарного предиката смежности вершин и предиката равенства, причём предикат смежности $E(x,y)$ является *иррефлексивным и симметричным*, т. е. удовлетворяет условиям:

1) $\forall x \left(\neg E(x,x)\right)$ (иррефлексивность);
2) $\forall x \forall y \left(E(x,y) \to E(y,x)\right)$ (симметричность).

Поскольку язык L обыкновенных графов не содержит функциональных символов, то здесь и далее рассматриваются только предикатные системы, а все определения адаптированы под предикатный случай.

Пусть $\Gamma = \langle V, \mathrm{L} \rangle$ — фиксированный обыкновенный граф, X — конечное множество переменных. Множество $T_{\mathrm{L}}(X)$ *термов* языка L от переменных из множества X состоит из всех переменных $x \in X$ и всех констант C. Множество $At_{\mathrm{L}}(X)$ *атомарных формул* языка L от переменных из множества X состоит из всех формул вида $t_1 = t_2$ и $E(t_1, t_2)$, где $t_1, t_2 \in T_{\mathrm{L}}(X)$. Атомарные формулы называются *уравнениями языка*, а произвольные подмножества $S \subseteq At_{\mathrm{L}}(X)$ — *системами уравнений* языка L. В данной работе рассматриваются только конечные системы уравнений *диофантовых языков*, т. е. таких языков, в которых множество констант совпадает с множеством вершин графа.

Любая система уравнений S от k переменных над фиксированным обыкновенным графом определяет в аффинном k-мерном пространстве V^k множество своих решений $V_\Gamma(S) = \{\overline{v} \in V^k : \Gamma \models S(\overline{v})\}$, которое называется *алгебраическим множеством*. Если $V_\Gamma(S) = \varnothing$, то система уравнений S *несовместна* над Γ; иначе она является *совместной*. Две системы уравнений S_1 и S_2 называются *эквивалентными* над Γ, если $V_\Gamma(S_1) = V_\Gamma(S_2)$. Для любой системы уравнений S над Γ существует единственная эквивалентная ей максимальная система уравнений над Γ, которая называется *радикалом* системы S и обозначается $\mathrm{Rad}_\Gamma(S)$. Если система S несовместна над Γ, то $\mathrm{Rad}_\Gamma(S) = At_{\mathrm{L}}(X)$.

Роль координатного графа $CG_\Gamma(S)$ системы уравнений S аналогична роли общего решения системы линейных уравнений над полем в линейной алгебре. Координатный граф однозначно определяется радикалом $\mathrm{Rad}_\Gamma(S)$ следующим образом.

Отношение θ_S на множестве термов $T_{\mathrm{L}}(X)$, заданное по правилу

$$t_1 \sim_{\theta_S} t_2 \;\Leftrightarrow\; (t_1 = t_2) \in \mathrm{Rad}_\Gamma(S),\; t_1, t_2 \in T_{\mathrm{L}}(X),$$

является отношением эквивалентности, а константные и предикатные символы языка L интерпретируются на фактор-множестве $T_{\mathrm{L}}(X)/\theta_S$ следующим образом:

1) $c/\theta_S = c$ для любого символа $c \in \mathrm{C}$;
2) $E(t_1/\theta_S, t_2/\theta_S) = \texttt{и} \;\Leftrightarrow\; E(t_1, t_2) \in \mathrm{Rad}_\Gamma(S)$.

Построенный на фактор-множестве $T_{\mathrm{L}}(X)/\theta_S$ граф $CG_\Gamma(S)$ называется *координатным графом* алгебраического множества $V_\Gamma(S)$. Если система S несовместна над Γ, то $CG_\Gamma(S)$ является тривиальной системой $\mathcal{E}$, т. е. графом, состоящим из единственной вершины и петли.

Следует отметить, что введённые отношение эквивалентности θ_S и определение координатного графа являются частными случаями обобщённых понятий конгруэнции и фактор-системы, предложенных В. А. Горбуновым и В. И. Тумановым [10].

2. Процедура проверки совместности системы уравнений в общем случае

Пусть $\Gamma = \big(V(\Gamma), E(\Gamma)\big)$ — обыкновенный граф с n вершинами $v_1, \ldots, v_n$, S — конечная система уравнений над Γ от переменных из множества $X = \{x_1, \ldots, x_k\}$.

Теорема 1. Задача проверки совместности системы уравнений S над произвольным конечным графом Γ является $\mathcal{NP}$-полной задачей.

Доказательство. Достаточно показать, что к задаче проверки совместности системы уравнений S над конечным графом Γ полиномиально сводится какая-нибудь другая $\mathcal{NP}$-полная задача распознавания [11], например распознавательный вариант задачи о клике.

Задача о клике. Дан n-вершинный граф $\Gamma = (V, E)$ и натуральное число $k \leqslant n$. Существует ли в графе Γ *клика* размера, не меньшего k, т. е. такое множество $V' \subseteq V$, что $(u, v) \in E$ для любых $u, v \in V'$ и $|V'| \geqslant k$?

Для ответа на поставленный вопрос нужно определить, совместна ли над графом Γ система уравнений $S = \big\{E(x_i, x_j) : i, j \in \{1, \ldots, k\},\; i < j\big\}$.

Таким образом, задача проверки совместности системы уравнений S над конечным графом Γ является $\mathcal{NP}$-полной. ■

Следствие 1. Если $\mathcal{P} \neq \mathcal{NP}$, то не существует полиномиального алгоритма проверки совместности системы уравнений S над произвольным конечным графом Γ.

2.1. Процедура проверки совместности системы уравнений S над произвольным конечным графом

Ранее А. В. Ильевым и В. Н. Ремесленниковым была предложена процедура проверки совместности системы уравнений S над конечным графом Γ [9]. Мы приведём её конструктивно улучшенный вариант без каких-либо существенных и структурных изменений. Важным предварительным шагом для работы этой процедуры является нахождение информационной базы системы уравнений S.

Информационная база системы S состоит из набора конечных множеств и натуральных чисел, определяемых по группам:

1) $X = \{x_1, \ldots, x_k\}$ — множество переменных, $S = \{s_1, \ldots, s_l\}$ — множество уравнений с переменными из X; k, l — числовые параметры.
2) $W_1, \ldots, W_k$ — подмножества $V(\Gamma)$; W_i состоит из вершин графа Γ, которые содержатся в записи уравнений вида $E(x_i, v_j)$ системы S; $\alpha_i = |W_i|$ — числовые параметры, $i = 1, \ldots, k$.
3) $W_1^\perp, \ldots, W_k^\perp$ — подмножества $V(\Gamma)$; $W_i^\perp$ состоит из вершин графа Γ, которые смежны с каждой из вершин множества W_i (если $W_i = \varnothing$, то по определению полагаем $W_i^\perp = V(\Gamma)$); $\beta_i = |W_i^\perp|$ — числовые параметры, $i = 1, \ldots, k$.
4) $W_1^{\perp\perp}, \ldots, W_k^{\perp\perp}$ — подмножества $V(\Gamma)$; $W_i^{\perp\perp} = (W_i^\perp)^\perp$; $\gamma_i = |W_i^{\perp\perp}|$ — числовые параметры, причём $\gamma_i \geqslant \alpha_i$ для любых $i = 1, \ldots, k$.

Если в информационной базе хотя бы одно из чисел β_i равно нулю при $\alpha_i \neq 0$, то информационная база является *несогласованной*, а система S несовместна над Γ.

Поскольку случай $n = 1$ тривиальный, а при $n = 2$ обыкновенный граф Γ является двудольным и для проверки совместности системы уравнений S над ним уместно использовать полиномиальную процедуру 1.1 из п. 3.1, то всюду далее предполагается, что $n \geqslant 3$.

Процедура 1. Данная процедура строит классы эквивалентности $Y(t_i)$, где t_i — переменная x_i либо константа v_i, и преобразует систему уравнений S в эквивалентную систему $\overline{S}$ на графе Γ. В начале работы процедуры каждое множество $Y(t_i)$ состоит только из одного терма t_i, а система $\overline{S}$ совпадает с S.

В процессе работы процедура использует следующие структуры данных:

— Граф Γ представлен в виде матрицы смежности $A = \big(a_{ij}\big)$ размера $n \times n$, состоящей из нулей и единиц.
— Каждому типу уравнений отвечает своя матрица, состоящая из нулей и единиц, в которой единицами отмечены соответствующие уравнения. Равенства вида $x_i = x_j$ записаны в виде матрицы размера $k \times k$, равенства $x_i = v_j$ — в виде матрицы размера $k \times n$, а равенства $v_i = v_j$ — в виде матрицы размера $n \times n$. Матрицы, соответствующие уравнениям вида $E(t_i, t_j)$, выглядят аналогично.
— Семейства множеств $\{W_i\}$ и $\{W_i^\perp\}$, также записаны в виде матриц W и $W^\perp$, состоящих из нулей и единиц, размера $k \times n$. Строками этих матриц являются характеристические векторы множеств W_i и $W_i^\perp$.
— Классы эквивалентности $Y(t_i)$ хранятся в виде списков термов. Изначально количество таких списков $k + n$, а терм t_i — первый и единственный элемент в каждом списке. В дальнейшем при работе процедуры количество списков может сокращаться, при этом длина каждого из них ограничена значением $k + n$.

Шаг 0. Производится заполнение матриц W и $W^{\perp}$ и построение классов эквивалентности $Y(t_i)$. Матрица W сначала совпадает с матрицей, соответствующей уравнениям вида $E(x_i, v_j)$. Чтобы заполнить матрицу $W^{\perp}$, нужно для каждого её элемента перемножить элементы матрицы A графа Γ из тех строк, которые соответствуют вершинам из множеств W_i, т. е. в матрице $W^{\perp}$ каждый элемент определяется по формуле $w_{ij}^{\perp} = \prod\limits_{m\colon v_m \in W_i} a_{mj}$. Если хотя бы одна из строк матрицы $W^{\perp}$ состоит только из нулей, то исходная система S несовместна над Γ и процедура завершает свою работу.

Шаг 1. Для каждого равенства $t_i = t_j$ из $\overline{S}$ процедура объединяет множества $Y(t_p)$ и $Y(t_q)$, содержащие термы t_i и t_j. Полученное множество обозначается $Y(t_m)$, где t_m — константа с наименьшим номером, а при её отсутствии — переменная с наименьшим номером из множества $Y(t_p) \cup Y(t_q)$. Равенство $t_i = t_j$ при этом удаляется из $\overline{S}$.

Шаг 2. Процедура последовательно просматривает все множества $Y(t)$.

1) Если существует множество $Y(t)$, в котором содержатся две различные константы v_i и v_j, то исходная система S несовместна над Γ.
2) Если в множестве $Y(t)$ содержится только одна константа v_j и при этом $\bigcap\limits_{i\colon x_i \in Y(t)} W_i^{\perp} \cap \{v_j\} = \varnothing$, то исходная система S несовместна над Γ.
3) Если множество $Y(t)$ не содержит констант v_j, но при этом $\bigcap\limits_{i\colon x_i \in Y(t)} W_i^{\perp} = \varnothing$, то исходная система S несовместна над Γ.

   В каждом из случаев 1–3 процедура завершает работу.
4) Если в множестве $Y(t)$ содержится только одна константа v_j и при этом $\bigcap\limits_{i\colon x_i \in Y(t)} W_i^{\perp} \cap \{v_j\} = \{v_j\}$, то во всех уравнениях $\overline{S}$ каждая переменная $x_i \in Y(t)$ заменяется на v_j и для каждой из них множество $W_i^{\perp}$ переопределяется следующим образом: $W_i^{\perp} := \{v_j\}$.
5) Если множество $Y(t)$ не содержит констант, но при этом $\bigcap\limits_{i\colon x_i \in Y(t)} W_i^{\perp} = \{v_j\}$, то во всех уравнениях $\overline{S}$ каждая переменная $x_i \in Y(t)$ заменяется на v_j и в $\overline{S}$ добавляются равенства $x_i = v_j$.
6) Если множество $Y(t)$ не содержит констант и при этом $\Big|\bigcap\limits_{i\colon x_i \in Y(t)} W_i^{\perp}\Big| > 1$, то алгоритм выбирает переменную $x_j \in Y(t)$ с наименьшим номером и во всех уравнениях $\overline{S}$ заменяет все остальные переменные из $Y(t)$ на x_j.

Шаг 3. Процедура просматривает уравнения $E(t_i, t_j)$.

1) Если в $\overline{S}$ имеются уравнения $E(x_i, x_i)$ либо $E(v_i, v_j)$, такие, что $(v_i, v_j) \notin E(\Gamma)$, то исходная система S несовместна над Γ и процедура заканчивает работу. В противном случае уравнения $E(v_i, v_j)$ удаляются из системы $\overline{S}$.
2) Для каждой переменной, входящей в уравнения вида $E(x_i, v_j)$, процедура ищет элемент v_j в множестве W_i. Если такой элемент не найден, т. е. уравнение $E(x_i, v_j)$ было получено на шаге 2, то процедура переопределяет множества W_i и $W_i^{\perp}$ следующим образом: $W_i := W_i \cup \{v_j\}$ и $W_i^{\perp} := W_i^{\perp} \cap \{v_j\}^{\perp}$, где множество $\{v_j\}^{\perp}$ состоит из всех вершин графа Γ, смежных с v_j. Если получили $W_i^{\perp} = \varnothing$, то исходная система S несовместна над Γ; если $W_i^{\perp} = \{v_m\}$, то в $\overline{S}$ добавляется равенство $x_i = v_m$.

Если после выполнения шага 3 в $\overline{S}$ осталось хотя бы одно равенство, то процедура возвращается на шаг 1; если в $\overline{S}$ равенств нет, но есть уравнения вида $E(x_i, x_j)$, то процедура переходит на шаг 4. В противном случае процедура завершает работу.

Шаг 4. Процедура просматривает уравнения $E(x_i, x_j)$.

1) Процедура ищет минимальное доминирующее множество в графе $\mathrm{H} = (V_X, E_X)$, где V_X — множество переменных, входящих в запись уравнений $E(x_i, x_j)$, а множество рёбер E_X определяется этими уравнениями. Напомним, что подмножество $V' \subseteq V$ вершин графа (V, E) называется *доминирующим*, если каждая вершина из $V \setminus V'$ смежна по крайней мере с одной вершиной из V'. Для поиска минимального доминирующего множества можно использовать любые известные алгоритмы, имеющие трудоёмкость не хуже, чем $O(3^{|V_X|/2})$, например, предложенные в [12, 13]. Найденное доминирующее множество обозначим X'. Без ограничения общности можно считать, что $X' = \{x_1, \ldots, x_p\}$. Несложно заметить, что $p \leqslant k/2$.
2) Далее рассматриваются всевозможные наборы вида $(v_{1q}, v_{2q}, \ldots, v_{pq})$, где $v_{jq} \in W_j^{\perp}$. Всего таких наборов $|W_1^{\perp}| \cdot |W_2^{\perp}| \cdot \ldots \cdot |W_p^{\perp}| \leqslant n^p$. Для каждого такого набора составляется система уравнений
$$S_q = \overline{S} \cup \Big(\bigcup_{j=1}^{p} (x_j = v_{jq}) \Big)$$
с собственной информационной базой, включающей свои множества W_{iq} и $W_{iq}^{\perp}$. Каждая такая система проверяется на совместность, т. е. для неё выполняются шаги 1–3 процедуры. Если все системы S_q окажутся несовместными над Γ, то система $\overline{S}$ тоже несовместна над Γ и процедура завершает работу.
3) В случае, когда некоторые системы уравнений S_m из множества $\{S_q\}$ окажутся совместными над графом Γ, система уравнений $\overline{S} \cup \big(\bigcap\limits_m \overline{S}_m\big)$ эквивалентна исходной системе S, где $\overline{S}_m$ — системы уравнений, полученные после выполнения шагов 1–3 процедуры для совместных систем S_m. В этом случае происходит доопределение системы $\overline{S}$ вместе с множествами W_i и $W_i^{\perp}$ следующим образом: $\overline{S} := \overline{S} \cup \big(\bigcap\limits_m \overline{S}_m\big)$, $W_i := W_i \cup \big(\bigcap\limits_m W_{im}\big)$ и $W_i^{\perp} := W_i^{\perp} \cup \big(\bigcap\limits_m W_{im}^{\perp}\big)$. Заметим, что системы $\overline{S}_m$ могут состоять только из уравнений вида $E(x_i, v_j)$.
4) Кроме того, если некоторые классы эквивалентности $Y(t)$ объединялись в каждой совместной системе $\overline{S}_m$, то эти классы эквивалентности объединяются и в системе $\overline{S}$. Для этого составляются специальные матрицы $Y' = \big(y'_{ij}\big)$ размера $k \times k$ и $Y'' = \big(y''_{ij}\big)$ размера $k \times n$, изначально заполненные единицами. Затем для каждой системы $\overline{S}_m$ данные матрицы преобразуются следующим образом. Элемент y'_{ij} сохраняет свое текущее значение, если равенство $x_i = x_j$ имеет место в системе $\overline{S}_m$, и $y'_{ij} := 0$ в противном случае. Аналогично элемент y''_{ij} сохраняет свое текущее значение, если равенство $x_i = v_j$ имеет место в системе $\overline{S}_m$, и $y''_{ij} := 0$ в противном случае. В итоге матрицы Y' и Y'' будут содержать информацию обо всех равенствах вида $x_i = x_j$ и $x_i = v_j$, имевших место в любой совместной системе $\overline{S}_m$. Далее для каждого равенства, определяемого этими матрицами, происходит объединение соответствующих классов экивалентности системы $\overline{S}$.

После прохождения шагов 0–4 процедура завершает работу. В результате система $\overline{S}$ состоит только из уравнений вида $E(t_i, t_j)$.

2.2. Корректность процедуры проверки совместности системы уравнений S в общем случае

Доказательство следующей теоремы приведено в [9]. Для полноты картины воспроизведём его здесь.

Теорема 2. Процедура 1 корректно проверяет систему уравнений S на совместность.

Доказательство. Прежде всего докажем, что система уравнений $\overline{S}$, построенная процедурой 1, эквивалентна S.

По окончании шага 1 это очевидно.

Этапы 4 и 5 шага 2 означают, что только v_j является вершиной графа Γ, которая может быть значением переменных из класса $Y(t)$. Поэтому замена каждой из этих переменных на v_j сохраняет эквивалентность системы уравнений $\overline{S}$ исходной системе над Γ.

Этап 6 шага 2 означает, что множество значений переменных из класса $Y(t)$ непусто, и потому замена каждой из этих переменных на x_j сохраняет эквивалентность системы уравнений $\overline{S}$ исходной системе S над Γ.

На этапе 2 шага 3 уточняется множество значений переменной x_i на графе Γ, и если оно состоит только из одной вершины v_j, то добавление в $\overline{S}$ уравнения $x_i = v_j$ сохраняет эквивалентность системы уравнений $\overline{S}$ исходной системе S над Γ.

На шаге 4 осуществляется переход от системы уравнений $\overline{S}$, содержащей уравнения с двумя переменными $E(x_i, x_j)$, к системам уравнений S_q, таким, что $\overline{S}$ эквивалентна объединению $\bigcup\limits_q S_q$ над графом Γ. Если в каждую совместную систему S_m из множества $\{S_q\}$ после её преобразования в систему $\overline{S}_m$ попадут какие-либо уравнения, не содержащиеся в $\overline{S}$, то добавление этих уравнений в $\overline{S}$ сохранит эквивалентность системы уравнений $\overline{S}$ исходной системе S над графом Γ. Если в процессе проверки совместности каждой системы S_m происходило объединение одних и тех же классов эквивалентности $Y(t)$, то эти классы эквивалентности должны быть объединены и в системе $\overline{S}$.

Очевидно, что для любой переменной x_i множество её значений содержится в $W_i^{\perp}$. Несовместность системы $\overline{S}$, устанавливаемая на этапе 1 шага 2, следует из попадания двух констант v_i и v_j в один класс эквивалентности, что противоречит условию попарного различия всех вершин графа. Несовместность системы $\overline{S}$, устанавливаемая на этапах 2 и 3 шага 2, следует из того, что никакая вершина графа Γ не может быть значением переменной x_i из класса эквивалентности $Y(t)$.

Несовместность системы $\overline{S}$, устанавливаемая на этапе 1 шага 3, следует из отсутствия в графе Γ петель и возможного отсутствия рёбер (v_i, v_j), существование которых вытекает из уравнений системы $\overline{S}$. Несовместность системы $\overline{S}$, устанавливаемая на этапе 2 шага 3, следует из того, что переменной x_i не могут быть присвоены никакие значения на графе Γ.

Несовместность системы $\overline{S}$, устанавливаемая на шаге 4 процедуры, следует из того, что переменным из доминирующего множества X' не могут быть всем одновременно присвоены никакие конкретные значения на графе Γ. ■

2.3. Трудоёмкость процедуры проверки совместности системы уравнений S в общем случае

Теорема 3. Трудоёмкость процедуры 1 составляет $O(k^2 n^{k/2+1}(k+n)^2)$.

Доказательство. Распишем вычислительную сложность процедуры 1 по шагам:

Сложность шага 0 составляет $O(kn + kn^2)$, или, что то же, $O(kn^2)$.

Сложность шага 1 — $O(k^2 + n^2 + kn)$, или, что то же, $O((k + n)^2)$.

Сложность этапа 1 шага 2 в худшем случае составляет $O((k + n)^2)$.

Сложность этапов 2 и 3 шага 2 — $O((k+n)(k+n+kn))$, или, что то же, $O(kn(k+n))$.

Сложность этапов 4–6 шага 2 — $O((k+n)(k+n+kn+k^2)$, или, что то же, $O(k(k+n)^2)$.

Таким образом, трудоёмкость шага 2 составляет $O(k(k + n)^2)$.

Сложность этапа 1 шага 3 составляет $O(k + n^2)$.

Сложность этапа 2 шага 3 — $O(kn + kn^2)$, или, что то же, $O(kn^2)$.

Таким образом, трудоёмкость всех шагов 1–3 процедуры за одну итерацию составляет $O(k(k + n)^2)$.

Поскольку шаги 1–3 могут быть повторены для каждого из kn возможных равенств $x_i = v_j$, то общая вычислительная сложность их прохождения в худшем случае составляет $O(k^2n(k + n)^2)$.

Сложность этапа 1 шага 4 в худшем случае составляет $O(3^{k/2})$, что не хуже, чем $O(n^{k/2})$ при $n \geqslant 3$.

Поскольку на этапе 2 шага 4 количество всевозможных комбинаций $(v_{1q}, v_{2q}, \ldots, v_{pq})$ при $p \leqslant k/2$ и $q \leqslant n$ в худшем случае равно $n^{k/2}$ и для каждой полученной новой системы уравнений S_q осуществляется прохождение шагов 1–3, то сложность этапа 2 шага 4 составляет $O(k^2n^{k/2+1}(k + n)^2)$.

Сложность этапа 3 шага 4 в худшем случае — $O(kn^{k/2+1})$.

Сложность этапа 4 шага 4 в худшем случае — $O(k^2 + kn + n^{k/2}(k + n)^2)$, или, что то же, $O(n^{k/2}(k + n)^2)$.

Таким образом, трудоёмкость всей процедуры 1 составляет $O(k^2n^{k/2+1}(k + n)^2)$. ■

3. Процедура проверки совместности системы уравнений над двудольными графами

Поскольку задача проверки совместности системы уравнений в общем случае $\mathcal{NP}$-полна, т. е. не является полиномиально разрешимой при $\mathcal{P} \neq \mathcal{NP}$, то возникает естественный вопрос о выделении её полиномиально разрешимых случаев. В число таких случаев входят задачи проверки совместности системы уравнений над деревьями, лесами, всеми двудольными графами и полными двудольными графами.

Нетрудно заметить, что деревья и леса являются двудольными графами, но не обязательно полными двудольными. Поэтому предложена единая процедура проверки совместности системы уравнений S над конечными двудольными графами, которая применима к деревьям и лесам, и отдельная процедура проверки совместности системы уравнений S над полными двудольными графами.

Пусть $\Gamma = \big(V(\Gamma), E(\Gamma)\big)$ — обыкновенный двудольный граф с n вершинами и S — конечная система уравнений над Γ от переменных из множества $X = \{x_1, \ldots, x_k\}$.

3.1. Процедура проверки совместности системы уравнений S над конечным двудольным графом

Процедура 1.1. Идея этой процедуры аналогична идее процедуры 1. В процессе работы используются те же структуры данных, что и в общем случае.

Шаги 0–3 в точности повторяют соответствующие шаги процедуры 1, а шаг 4 выглядит следующим образом:

Шаг 4. Процедура просматривает граф $\mathrm{H} = (V_X, E_X)$, где V_X — множество переменных, входящих в запись уравнений $E(x_i, x_j)$, а множество рёбер E_X определяется этими уравнениями.

1) В графе H ищутся циклы нечётной длины. Для этого производится расслоение каждой его компоненты связности, начиная с произвольной вершины, — создаётся специальный список, в котором каждый элемент содержит номер своего слоя и номер своей вершины. Если цикл нечётной длины найден, т. е. какие-то две вершины из одного слоя оказались смежны, то исходная система уравнений S несовместна над графом Γ.
2) Для каждого уравнения вида $E(x_i, x_j)$ множества $W_i^\perp$ и $W_j^\perp$ переопределяются следующим образом: $W_i^\perp := W_i^\perp \cap \Big(\bigcup\limits_{v_j \in W_j^\perp} \{v_j\}^\perp \Big)$ и $W_j^\perp := W_j^\perp \cap \Big(\bigcup\limits_{v_i \in W_i^\perp} \{v_i\}^\perp \Big)$.

   Если хотя бы одно множество $W_p^\perp = \varnothing$, то исходная система S несовместна над Γ; иначе — совместна. Если какое-либо множество $W_p^\perp = \{v_q\}$, то в $\overline{S}$ добавляется равенство $x_p = v_q$.
3) Для каждого равенства $x_p = v_q$ из $\overline{S}$, полученного на предыдущем этапе, процедура объединяет множества $Y(t_r)$ и $Y(v_q)$, содержащие переменную x_p и константу v_q, а равенства удаляются из системы. Полученное множество обозначается $Y(v_q)$. На этом процедура завершает свою работу.

Теорема 4. Процедура 1.1 корректно проверяет систему уравнений S на совместность.

Доказательство. Корректность шагов 1–3 процедуры доказывается аналогично тому, как это сделано в теореме 2.

На шаге 4 возможно уточнение множества значений переменной x_p на графе Γ, и если оно состоит только из одной вершины v_q, то добавление в $\overline{S}$ уравнения $x_p = v_q$ сохраняет эквивалентность системы уравнений $\overline{S}$ исходной системе S над Γ.

Несовместность системы $\overline{S}$, устанавливаемая на шаге 4, следует из того, что в графе Γ не существует циклов нечётной длины и при этом переменные, входящие в уравнения $E(x_i, x_j)$, могут иметь своими значениями только смежные вершины в графе Γ, т. е. находящиеся в разных его долях. ■

Теорема 5. Трудоёмкость процедуры 1.1 составляет $O(k^2 n(k+n)^2)$.

Доказательство. Распишем вычислительную сложность по шагам.

Сложность каждого из шагов 0–3 процедуры 1.1 такая же, как и у процедуры 1; общая вычислительная сложность их прохождения в худшем случае составляет $O(k^2 n(k+n)^2)$.

Сложность этапа 1 шага 4 составляет $O(k^3)$; этапа 2 — $O(k^2 n^2)$; этапа 3 — $O(kn)$.

Таким образом, трудоёмкость всей процедуры 1.1 составляет $O(k^2 n(k+n)^2)$. ■

3.2. Процедура проверки совместности системы уравнений S над конечным полным двудольным графом

Процедура 1.2. Идея этой процедуры полностью аналогична идее процедуры 1.1. Полный двудольный граф Γ хранится в виде матрицы размера $2 \times n$, состоящей из нулей и единиц, а в остальном в процессе работы используются те же структуры данных, что и в общем случае.

Шаги 0–3 в точности повторяют соответствующие шаги процедуры 1, а шаг 4 выглядит следующим образом:

Шаг 4. Процедура просматривает граф $H = (V_X, E_X)$, где V_X — множество переменных, входящих в запись уравнений $E(x_i, x_j)$, а множество рёбер E_X определяется этими уравнениями.

1) В графе H ищутся циклы нечётной длины. Для этого производится расслоение каждой его компоненты связности, начиная с произвольной вершины. Если цикл нечётной длины найден, то исходная система уравнений S несовместна над графом Г. Иначе граф H является двудольным и для каждой его компоненты связности формируются списки вершин долей.
2) Процедура просматривает списки долей графа H. Если две переменные x_i и x_j лежат в одной доле графа H, а вершины из множеств $W_i^{\perp}$ и $W_j^{\perp}$ принадлежат разным долям графа Г, то система уравнений S несовместна над Г. Если две переменные x_i и x_j лежат в разных долях графа H, но в одной его компоненте связности, а вершины из множеств $W_i^{\perp}$ и $W_j^{\perp}$ принадлежат одной доле графа Г, то система S также несовместна над Г. В противном случае система уравнений S является совместной над графом Г.

   На этом процедура завершает свою работу.

Теорема 6. Процедура 1.2 корректно проверяет систему уравнений S на совместность.

Доказательство полностью аналогично доказательству теоремы 4.

Теорема 7. Трудоёмкость процедуры 1.2 составляет $O(k^2n(k+n)^2)$.

Доказательство. Как и у процедуры 1.1, общая трудоёмкость прохождения шагов 0–3 в худшем случае составляет $O(k^2n(k+n)^2)$.

Сложность этапа 1 шага 4 составляет $O(k^3)$; этапа 2 — $O(k^2n)$.

Таким образом, трудоёмкость всей процедуры 1.2 составляет $O(k^2n(k+n)^2)$. ■

4. Процедуры построения радикала и координатного графа

Обе эти процедуры, приведенные в [9] с полным доказательством их корректности, без изменений используются как при решении систем уравнений над двудольными графами, так и в общем случае.

4.1. Построение радикала системы уравнений S

Процедура 2. Если система S несовместна над Г, то $\mathrm{Rad}_\Gamma(S) = At_{\mathrm{L}}(X)$. В противном случае процедура работает с информационной базой системы уравнений S.

Шаг 0. Процедура рассматривает последние версии множеств $W_1^{\perp}, \ldots, W_k^{\perp}$, полученные при выполнении процедуры проверки совместности системы S. С их помощью заново определяются множества $W_1^{\perp\perp}, \ldots, W_k^{\perp\perp}$.

Далее радикал $\mathrm{Rad}_\Gamma(S)$ строится следующим образом.

Шаг 1. $\mathrm{Rad}_\Gamma(S) := \overline{S}$, для чего определяются новые матрицы уравнений.

Шаг 2. Уравнения вида $E(x_i, v_j)$ добавляются в $\mathrm{Rad}_\Gamma(S)$ для любых $v_j \in W_i^{\perp\perp}$.

Шаг 3. Если $t_1, t_2 \in Y(t)$, то уравнения $t_1 = t_2$ добавляются в $\mathrm{Rad}_\Gamma(S)$ для любых $t_1, t_2 \in T_{\mathrm{L}}(X)$.

Шаг 4. Уравнения $t = t$ добавляются в $\mathrm{Rad}_\Gamma(S)$ для любого $t \in T_{\mathrm{L}}(X)$.

Шаг 5. Если $(t_1 = t_2) \in \mathrm{Rad}_\Gamma(S)$, то уравнения $t_2 = t_1$ добавляются в $\mathrm{Rad}_\Gamma(S)$ для любых $t_1, t_2 \in T_{\mathrm{L}}(X)$. Поскольку матрица, содержащая уравнения вида $x_i = v_j$, не является квадратной, то для данного типа уравнений считаем действия на шаге 5 выполненными автоматически.

Шаг 6. Если $(t_1 = t_2) \in \mathrm{Rad}_\Gamma(S)$ и $(t_2 = t_3) \in \mathrm{Rad}_\Gamma(S)$, то уравнения $t_1 = t_3$ добавляются в $\mathrm{Rad}_\Gamma(S)$ для любых $t_1, t_2, t_3 \in T_{\mathrm{L}}(X)$.

Шаг 7. Если $E(t_1, t_2) \in \mathrm{Rad}_\Gamma(S)$, то уравнения $E(t_2, t_1)$ добавляются в $\mathrm{Rad}_\Gamma(S)$ для любых $t_1, t_2 \in T_{\mathrm{L}}(X)$. Поскольку матрица, содержащая уравнения вида $E(x_i, v_j)$, не является квадратной, то для данного типа уравнений считаем действия на шаге 7 выполненными автоматически.

Шаг 8. Если $(t_1 = t'_1), (t_2 = t'_2) \in \mathrm{Rad}_\Gamma(S)$ и $E(t_1, t_2) \in \mathrm{Rad}_\Gamma(S)$, то уравнения $E(t'_1, t'_2)$ добавляются в $\mathrm{Rad}_\Gamma(S)$ для любых $t_1, t_2, t'_1, t'_2 \in T_{\mathrm{L}}(X)$.

Шаг 9. Уравнения $E(v_i, v_j)$ добавляются в $\mathrm{Rad}_\Gamma(S)$ для любых $(v_i, v_j) \in E(\Gamma)$.

Теорема 8. Процедура 2 корректно строит радикал $\mathrm{Rad}_\Gamma(S)$.

Корректность процедуры 2 для построения радикала $\mathrm{Rad}_\Gamma(S)$ следует непосредственно из его определения.

Теорема 9. Трудоёмкость процедуры 2 составляет $O((k+n)^3)$.

Доказательство. Распишем вычислительную сложность процедуры 2 по шагам.

Переопределение множеств $W_1^{\perp\perp}, \ldots, W_k^{\perp\perp}$ на шаге 0 осуществляется за $O(kn^2)$ операций.

Сложность шага 1 составляет $O((k+n)^2)$; шага 2 — $O(kn)$; шага 3 — $O((k+n)^3)$; шага 4 — $O(k+n)$; шага 5 — $O(k^2+n^2)$; шага 6 — $O((k+n)^3)$; шага 7 — $O(k^2+n^2)$; шага 8 — $O((k+n)^2)$; шага 9 — $O(n^2)$.

Таким образом, трудоёмкость всей процедуры 2 составляет $O((k+n)^3)$. ■

4.2. Построение координатного графа системы уравнений S

Процедура 3. Множество вершин координатного графа $\Delta = CG_\Gamma(S)$ совпадает с множеством индикаторов t_i классов эквивалентности $Y(t_i)$ и является подмножеством $V(\Gamma) \cup X$, а множество рёбер выглядит следующим образом:

$$E(\Delta) = E(\Gamma) \cup E(x_i, x_j) \cup E(x_i, w_m), \text{ где } E(x_i, x_j), E(x_i, w_m) \in \mathrm{Rad}_\Gamma(S).$$

Если система S несовместна над Γ, то $\Delta = \mathcal{E}$, где $\mathcal{E}$ — граф, состоящий из одной вершины и петли. В противном случае процедура выполняет следующие построения:

Шаг 1. К множеству вершин графа Γ добавляются все вершины из множества X.

Шаг 2. К полученному графу добавляются всевозможные рёбра (x_i, x_j) и (x_i, w_m), для которых $E(x_i, x_j), E(x_i, w_m) \in \mathrm{Rad}_\Gamma(S)$.

Шаг 3. В полученном графе для каждого класса эквивалентности $Y(t_i)$ все вершины, находящиеся в этом классе, стягиваются в одну вершину, а кратные рёбра заменяются одним ребром.

В результате получается координатный граф Δ.

Теорема 10. Процедура 3 корректно строит координатный граф Δ.

Построение координатного графа процедурой 3 осуществляется непосредственно по определению.

Теорема 11. Трудоёмкость процедуры 3 составляет $O((k+n)^2)$.

Как следствия теорем 3, 5, 7, 9 и 11, можно получить трудоёмкости алгоритмов решения конечных систем уравнений над конечными обыкновенными графами, включающие процедуры проверки совместности этих систем, построения их радикала и общего решения — координатного графа.

Следствие 2. Трудоёмкость алгоритма решения систем уравнений над произвольными графами составляет $O(k^2 n^{k/2+1}(k+n)^2)$.

Следствие 3. Трудоёмкость алгоритма решения систем уравнений над двудольными графами, в том числе деревьями и лесами, составляет $O(k^2 n(k+n)^2)$.

Следствие 4. Трудоёмкость алгоритма решения систем уравнений над полными двудольными графами составляет $O(k^2 n(k+n)^2)$.

ЛИТЕРАТУРА

1. *Матиясевич Ю. В.* Диофантовость перечислимых множеств // Доклады Академии наук СССР. 1970. Т. 191. № 2. С. 279–282.
2. *Fischer M. J. and Rabin M. O.* Super-exponential complexity of Presburger arithmetic // Proc. SIAM-AMS Symp. Appl. Math. 1974. V. 7. P. 27–41.
3. *Mayr E. W. and Meyer A. R.* The complexity of the word problems for commutative semigroups and polynomial ideals // Adv. Math. 1982. V. 46. Iss. 3. P. 305–329.
4. *Cook S. A.* The complexity of theorem proving procedures // Proc. 3rd Ann. ACM Symp. on Theory of Computing. 1971. P. 151–158.
5. *Даниярова Э. Ю., Мясников А. Г., Ремесленников В. Н.* Алгебраическая геометрия над алгебраическими системами. Новосибирск: Изд-во СО РАН, 2016. 243 с.
6. *Никитин А. Ю., Рыбалов А. Н.* О сложности проблемы разрешимости систем уравнений над конечными частичными порядками // Прикладная дискретная математика. 2018. № 39. С. 94–98.
7. *Рыбалов А. Н.* О сложности экзистенциальной и универсальной теорий конечных полей // Прикладная дискретная математика. 2019. № 45. С. 85–89.
8. *Ильев А. В.* Разрешимость универсальных теорий и аксиоматизируемость наследственных классов графов // Труды института математики и механики УрО РАН. 2016. Т. 22. № 1. С. 100–111.
9. *Ильев А. В., Ремесленников В. Н.* Исследование совместности систем уравнений над графами и нахождение их общих решений // Вестник Омского университета. 2017. № 4(86). С. 26–32.
10. *Горбунов В. А.* Алгебраическая теория квазимногообразий. Новосибирск: Научная книга, 1999. 368+xii с.
11. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
12. *Fomin F. V., Grandoni F., and Kratsch D.* A measure & conquer approach for the analysis of exact algorithms // J. ACM. 2009. V. 56. Iss. 5. P. 25–32.
13. *Van Rooij J. M. M., Nederlof J., and van Dijk T. C.* Inclusion/exclusion meets measure and conquer: Exact algorithms for counting dominating sets // LNCS. 2009. V. 5757. P. 554–565.

REFERENCES

1. *Matiyasevich Y. V.* Diofantovost perechislimykh mnozhestv [Diophantineity of enumerable sets]. Doklady Akademii Nauk USSR, 1970, vol. 191, no. 2, pp. 279–282. (in Russian)
2. *Fischer M. J. and Rabin M. O.* Super-exponential complexity of Presburger arithmetic. Proc. SIAM-AMS Symp. Appl. Math., 1974, vol. 7, pp. 27–41.
3. *Mayr E. W. and Meyer A. R.* The complexity of the word problems for commutative semigroups and polynomial ideals. Adv. Math., 1982, vol. 46, iss. 3, pp. 305–329.
4. *Cook S. A.* The complexity of theorem proving procedures. Proc. 3rd Ann. ACM Symp. on Theory of Computing, 1971, pp. 151–158.
5. *Daniyarova E. Y., Myasnikov A. G., and Remeslennikov V. N.* Algebraicheskaya geometriya nad algebraicheskimi sistemami [Algebraic Geometry over Algebraic Structures]. Novosibirsk, SB RAS Publishing House, 2016. 243 p. (in Russian)
6. *Nikitin A. Y. and Rybalov A. N.* O slozhnosti problemy razreshimosti sistem uravneniy nad konechnymi chastichnymi poryadkami [On complexity of the satisfiability problem of systems over finite posets]. Prikladnaya Diskretnaya Matematika, 2018, no. 39, pp. 94–98. (in Russian)

7. *Rybalov A. N.* O slozhnosti ekzistentsialnoy i universalnoy teoriy konechnykh poley [On complexity of the existential and universal theories of finite fields]. Prikladnaya Diskretnaya Matematika, 2019, no. 45, pp. 85–89. (in Russian)
8. *Ilev A. V.* Razreshimost universalnykh teory i aksiomatiziruyemost nasledstvennykh klassov grafov [Decidability of universal theories and axiomatizability of hereditary classes of graphs]. Trudy Instituta Matematiki i Mekhaniki UrO RAN, 2016, vol. 22, no. 1, pp. 100–111. (in Russian)
9. *Ilev A. V. and Remeslennikov V. N.* Issledovaniye sovmestnosti sistem uravneny nad grafami i nakhozhdeniye ikh obshchikh resheny [Study of the compatibility of systems of equations over graphs and finding their general solutions]. Vestnik Omskogo Universiteta, 2017, no. 4(86), pp. 26–32. (in Russian)
10. *Gorbunov V. A.* Algebraic Theory of Quasivarieties. New York, Plenum, 1998. 298+xii p.
11. *Garey M. R. and Johnson D. S.* Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco, W. H. Freeman and Co, 1979. 338+x p.
12. *Fomin F. V., Grandoni F., and Kratsch D.* A measure & conquer approach for the analysis of exact algorithms. J. ACM, 2009, vol. 56, iss. 5, pp. 25–32.
13. *Van Rooij J. M. M., Nederlof J., and van Dijk T. C.* Inclusion/exclusion meets measure and conquer: Exact algorithms for counting dominating sets. LNCS, 2009, vol. 5757, pp. 554–565.