УДК 510.51 Вестник СПбГУ. Математика. Механика. Астрономия. 2019. Т. 6 (64). Вып. 3
MSC 05A05, 68T27

Полиномиальная эквивалентность задач ИЗОМОРФИЗМ ПРЕДИКАТНЫХ ФОРМУЛ и ИЗОМОРФИЗМ ГРАФОВ

Т. М. Косовская, Н. Н. Косовский

Санкт-Петербургский государственный университет,
Российская Федерация, 199034, Санкт-Петербург, Университетская наб., 7–9

Для цитирования: *Косовская Т. М., Косовский Н. Н.* Полиномиальная эквивалентность задач ИЗОМОРФИЗМ ПРЕДИКАТНЫХ ФОРМУЛ и ИЗОМОРФИЗМ ГРАФОВ // Вестник Санкт-Петербургского университета. Математика. Механика. Астрономия. 2019. Т. 6 (64). Вып. 3. С. 430–439. https://doi.org/10.21638/11701/spbu01.2019.308

В работе рассматривается задача проверки изоморфности двух элементарных конъюнкций предикатных формул, возникающая при решении ряда задач искусственного интеллекта, допускающих формализацию средствами языка исчисления предикатов, и ее связь с задачей проверки изоморфизма графов. Точное определение понятия изоморфности таких формул приведено в тексте статьи. Однако, неформально говоря, изоморфные элементарные конъюнкции предикатных формул — это формулы, которые при некоторой подстановке переменных вместо их аргументов совпадают с точностью до порядка записи литералов. Описаны задачи, при решении которых возникает необходимость проверки формул на изоморфность. Доказана полиномиальная эквивалентность задач проверки изоморфности предикатных формул и проверки изоморфности графов.

Ключевые слова: изоморфизм графов, изоморфизм предикатных формул, GI-полнота.

Введение. Задача проверки графов на изоморфизм вызывает особое внимание у математиков, занимающихся теорией сложности алгоритмов. Эта задача относится к так называемым «открытым» задачам, для которых не доказана их NP-полнота и не доказана их принадлежность классу **P**, т. е. не найден алгоритм, решающий их за полиномиальное время. Ласло Бабай в 2017 г. [1] предложил квази-полиномиальный алгоритм ее решения. Полученная им оценка $2^{O((\log\ n)^3)}$ до сих пор проверяется.

Задача изоморфизма графов считается «универсальной», то есть к ней можно свести любую задачу, где ставится вопрос об изоморфизме комбинаторных структур. В настоящее время доказана полиномиальная сводимость к ней многих задач проверки изоморфизма структур, тесно связанных с графами, например, 2-раскрашенные графы, мультиграфы, гиперграфы, конечные автоматы, контекстно-свободные грамматики [2]. По аналогии с понятием NP-полной задачи было даже введено понятие GI-полной задачи, т. е. задачи, полиномиально эквивалентной задаче проверки изоморфизма графов.

В [3] рассматриваются открытые вопросы, связанные с задачами изоморфизма графов, колец и групп.

В [4] (регулярно обновляемом издании, посвященном положению с открытыми задачами из [5]) обсуждается, в частности, положение дел с задачей изоморфизма графов (ИГ).

При решении некоторых задач искусственного интеллекта (ИИ), допускающих формализацию средствами исчисления предикатов, встает вопрос о проверке двух элементарных конъюнкций предикатных формул на их изоморфизм, т. е. на их совпадение с точностью до имен переменных и порядка литералов. Это отношение не является отношением графического равенства (даже если ввести условие на порядок следования литералов), так как переменные (или константы), входящие в формулы, могут иметь произвольные имена. Оно также не является отношением равносильности. Фактически изоморфные формулы задают одно и то же отношение между своими аргументами.

В настоящей работе доказано, что задачи «проверка двух элементарных конъюнкций предикатных формул на их изоморфизм» и «изоморфизм графов» полиномиально эквивалентны.

1. Основные определения и формулировки задач.

Определение. Две элементарные конъюнкции атомарных формул исчисления предикатов P и Q называются *изоморфными*, если существуют такая элементарная конъюнкция R и подстановки аргументов $a_1, \dots, a_m$ и $b_1, \dots, b_m$ формул P и Q соответственно вместо всех вхождений переменных $x_1, \dots, x_m$ формулы R, что результаты этих подстановок в R совпадают с формулами P и Q соответственно с точностью до порядка литералов.

При этом полученные подстановки $(a_1 \to x_{i_1}, \dots, a_m \to x_{i_m})$ и $(b_1 \to x_{j_1}, \dots, b_m \to x_{j_m})$ называются унификаторами формул P и Q с формулой R соответственно.

Приведем два примера пар изоморфных и неизоморфных формул.

1. Пусть $A(a,b,c,d) = p(a,b,c)\ \&\ p(d,a,b)\ \&\ p(c,d,a)$ и $B(a,b,c,d) = p(d,a,c)\ \&\ p(b,d,a)\ \&\ p(c,b,d)$. Формула $R_{AB}(x,y,z,u) = p(u,x,z)\ \&\ p(y,u,x)\ \&\ p(z,y,u)$ при подстановках $(a \to u, b \to x, c \to z, d \to y)$ и $(a \to x, b \to y, c \to z, d \to u)$ совпадает с $A(a,b,c,d)$ и $B(a,b,c,d)$ соответственно: $R_{AB}(b,d,c,a) = A(a,b,c,d)$, а $R_{AB}(a,b,c,d) = B(a,b,c,d)$. Следовательно, формулы $A(a,b,c,d)$ и $B(a,b,c,d)$ изоморфны.

2. Рассмотрим формулу $C(a,b,c,d) = p(a,b,c)\ \&\ p(d,c,b)\ \&\ p(c,d,a)$ (она отличается от $A(a,b,c,d)$ только вторым аргументом второго литерала). Какова бы ни была формула $R_{AC}(x,y,z,u)$, при любых подстановках аргументов формул $A(a,b,c,d)$ и $C(a,b,c,d)$ вместо переменных x, y, z и u, таких что совпадают первый и третий литералы формул, во втором литерале вторые аргументы окажутся различными.

Отметим, что аргументами элементарных конъюнкций P и Q могут быть как предметные переменные, так и предметные константы. Кроме того, понятие изоморфизма элементарных конъюнкций атомарных формул исчисления предикатов отличается от понятия равносильности этих формул, так как они могут иметь существенно различные аргументы. По сути дела для изоморфных формул существуют такие перестановки их аргументов, то они задают одно и то же отношение между этими аргументами.

В работе рассматривается следующая задача.

ИЗОМОРФИЗМ ПРЕДИКАТНЫХ ФОРМУЛ (ИПФ)

Дано: $\{a_1, \dots, a_m\}$, $\{b_1, \dots, b_m\}$ — множества предметных переменных или констант, $\{p_1, \dots, p_n\}$ — множество предикатных символов, P и Q — элементарные конъюнкции атомарных формул с аргументами $(a_1, \dots, a_m)$ и $(b_1, \dots, b_m)$ соответственно и предикатами из $\{p_1, \ldots, p_n\}$, у которых в каждом литерале все аргументы различны. Вопрос. Изоморфны ли P и Q?

2. Задачи ИИ, в которых возникает проблема проверки изоморфизма предикатных формул. При решении задач ИИ иногда возникают задачи, в которых удобным языком их формализации является язык исчисления предикатов.

Подробное изложение логико-предметного подхода к решению задач ИИ имеется в [6]. Здесь сформулируем только общую постановку задач, а также подзадачи, при решении которых возникает задача проверки на изоморфизм двух элементарных конъюнкций атомарных формул.

Пусть исследуемый объект представлен как множество своих элементов $\omega = \{\omega_1, \ldots, \omega_t\}$. На ω задан набор предикатов $p_1, \ldots, p_n$, характеризующих свойства элементов ω и отношения между ними. Логическим описанием $S(\omega)$ объекта ω называется множество всех атомарных формул или их отрицаний, истинных на ω. Множество всех объектов разбито на классы $\Omega = \cup_{k=1}^{K} \Omega_k$. Логическим описанием класса Ω_k называется формула $A_k(\overline{x})$, заданная в виде дизъюнкции элементарных конъюнкций, такая что если $A_k(\overline{\omega})$ истинна, то $\omega \in \Omega_k$.[1]

С помощью построенных описаний объектов и классов в [8] предлагается решать следующие задачи.

Задача идентификации. *Проверить, удовлетворяет ли объект ω или его часть описанию класса $A_k(\overline{x})$ и предъявить эту часть объекта.*

Задача классификации. *Найти все такие номера k, что верна формула $A_k(\overline{\omega})$.*

Задача анализа. *Найти и классифицировать все части τ объекта ω, для которых $A_k(\overline{\tau})$.*

Решение задач идентификации, классификации и анализа для распознавания сложного объекта сведено к доказательству соответственно логических следований[2]

$$S(\omega) \Rightarrow \exists \overline{x}_{\neq}\ A_k(\overline{x}), \tag{1}$$

$$S(\omega) \Rightarrow \bigvee_{k=1}^{M} A_k(\overline{\omega}), \tag{2}$$

$$S(\omega) \Rightarrow \bigvee_{k=1}^{M} \exists \overline{x}_{\neq}\ A_k(\overline{x}). \tag{3}$$

В [8] доказано, что задачи (1) и (3) NP-полны.

---

[1] Здесь и далее посредством $\overline{x}$ обозначается список элементов конечного множества x, соответствующий некоторой перестановке номеров его элементов. Тот факт, что элементами списка $\overline{x}$ являются элементы множества y, будем записывать в виде $x \subseteq y$.

[2] Для того чтобы записать, что значения для переменных списка $\overline{x}$, удовлетворяющие формуле $A(\overline{x})$, различны, вместо формулы

$$\exists x_1 \ldots \exists x_m (\&_{i=1}^{m} \&_{j=i+1}^{m} (x_i \neq x_j)\ \&\ A(x_1, \ldots, x_m))$$

будет использоваться обозначение

$$\exists \overline{x}_{\neq} A(\overline{x}).$$

Заметим, что для того, чтобы уметь доказывать (1) и (3), достаточно уметь доказывать логическое следование

$$S(\omega) \Rightarrow \exists \overline{x}_{\neq}\ A(\overline{x}), \tag{4}$$

где $A(\overline{x})$ — элементарная конъюнкция атомарных формул и их отрицаний. В [7, 8] доказаны оценки числа шагов алгоритмов, решающих задачу (4), а также задачи (1) и (3). Эти оценки имеют экспоненциальный от длины записи формулы $A(\overline{x})$ вид.

Задача (2) сводится к последовательной проверке при $k = 1, \ldots, M$ изоморфности элементарной конъюнкции $A_k(\overline{\omega})$ и конъюнкции литералов из $S(\omega)$.

Для уменьшения числа шагов работы алгоритмов, решающих описанные задачи, в [9] предложено многоуровневое описание классов распознаваемых объектов, по сути своей являющееся иерархическим описанием классов. Оно основывается на выделении из описаний классов подформул, изоморфных друг другу и задающих обобщенные характеристики объектов, присущие объектам одного класса [10].

Предложенное многоуровневое описание можно применить к решению задачи из [5].

КОНЪЮНКТИВНЫЙ БУЛЕВСКИЙ ЗАПРОС (КБЗ)

Дано: конечное множество D; совокупность предикатов $R = \{R_1, \ldots, R_n\}$, где R_i задает d_i-местное отношение между элементами из D; множество $S(D)$ всех литералов с предикатами из R, истинных на D; конъюнктивный булевский запрос Q с литералами вида $R_j(u_1, \ldots, u_{d_j})$ с переменными и константами из D в качестве аргументов.

Вопрос. Существует ли набор значений из D для переменных, при которых запрос Q истинен?

$$S(D) \Rightarrow \exists y_1 y_2 \ldots y_l (A_1\ \&\ A_2\ \&\ \ldots\ \&\ A_r).$$

Формулировка этой задачи почти полностью совпадает с NP-полной задачей выделения объекта на сложной сцене при логико-предметном подходе к решению задач искусственного интеллекта.

Существенная разница в применении этих задач заключается в следующем:

—база данных может не изменяться вообще или меняться достаточно редко, а запросы каждый раз могут возникать различные;

—при распознавании образов множество целевых формул (описаний классов объектов) может не изменяться вообще или меняться достаточно редко, а распознаваемые объекты каждый раз могут возникать новые.

При многократном решении задачи КБЗ в распоряжении исследователя нет совокупности элементарных конъюнкций. При создании базы данных мы не можем сказать с уверенностью, какие запросы могут возникнуть у пользователя. Однако сама база данных при всех запросах остается практически неизменной. Поэтому закономерности следует искать в самой базе (множестве постоянных литералов $S(D)$).

Построение многоуровневой базы данных основывается на выделении из множества литералов серий таких подмножеств, что конъюнкции литералов из любых двух подмножеств серии изоморфны друг другу [11].

3. Полиномиальная эквивалентность задач ИГ и ИПФ.

Теорема 1. *Задача ИГ полиномиально сводится к задаче ИПФ.*

Доказательство. Задача ИГ является сужением задачи ИПФ (и, тем самым, ИГ полиномиально сводится к ИПФ). Сужение заключается в том, что в условии ИПФ имеется единственный 2-местный предикат p и элементарные конъюнкции P и Q не содержат отрицания. При этом множество аргументов каждой из элементарных конъюнкций P и Q — это множество вершин в соответствующих графах, а множество пар аргументов каждой из элементарных конъюнкций можно рассматривать как множество ребер графа. ■

Для доказательства полиномиальной эквивалентности задач введем в рассмотрение понятие *графа, ассоциированного с элементарной конъюнкцией* предикатных формул, у которой ни один литерал не содержит одинаковых аргументов.

Упорядочим литералы так, чтобы литералы с одинаковыми предикатными символами шли подряд, причем внутри каждой такой группы сначала шли литералы без отрицания, а затем с отрицанием. Предикатные символы подгруппы без отрицаний обозначим w_{2i-1}, а предикатные символы подгруппы с отрицаниями обозначим w_{2i} $(i = 1, \ldots, n)$. Отметим, что общее количество введенных обозначений может быть меньше, чем $2n$, так как некоторые предикатные символы могут входить только без отрицаний, а некоторые — только с отрицаниями.

Рассмотрим следующие можества вершин.

Множество вершин $Pred$ — это множество предикатных символов w_i, входящих в литералы.

Множество вершин $Index$ — для каждого символа w_i в нем имеется $i + 2$ вершины. Эти висячие вершины, смежные с w_i, будут задавать номер i, записанный в унарной системе счисления, увеличенный на 2.

Множество вершин $Arity$ — для каждого вхождения символа w_i в нем имеется $n_i k_i$ вершин, где n_i — количество вхождений символа w_i, k_i — количество аргументов символа w_i.

Множество вершин $Label$ — для каждого символа w_i в нем имеется n_i вершин. Каждая из этих вершин является висячей и смежна с вершиной из $Arity$, соответствующей месту первого аргумента литерала.

Множество вершин Arg — множество аргументов конъюнкции.

Каждому вхождению литерала с предикатом w_i $(1 \le i \le 2n)$ поставим в соответствие подграф, полученный последовательным соединением ребрами k_i вершин из $Arity$ между собой (первую вершину последовательности «пометим» присоединением висячей вершины из множества $Label$), а также с вершиной w_i. Кроме того, вершину w_i соединим ребрами с $i + 2$ висячими вершинами из $Index$.

Кроме того, каждую вершину из $Arity$ соединим ребром с соответствующей вершиной из Arg.

Так, например, литералу $\neg p_i(a, b, c)$ соответствует следующий подграф, изображенный на рис. 1.

Отметим, что все такие подграфы, соответствующие различным литералам с одним и тем же предикатом, имеют общую вершину из множества $Pred$ и $i + 2$ смежные вершины из множества $Index$.

Например, элементарной конъюнкции $p(a)$ & $q(x, y)$ & $q(x, a)$ & $\neg q(a, y)$ соответствует граф, представленный на рис. 2. При этом предикатному символу p соответствует вершина w_1; предикатному символу q, входящему без отрицания, соот-

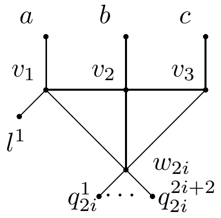

Рис. 1. Подграф, соответствующий литералу $\neg p_i(a, b, c)$.

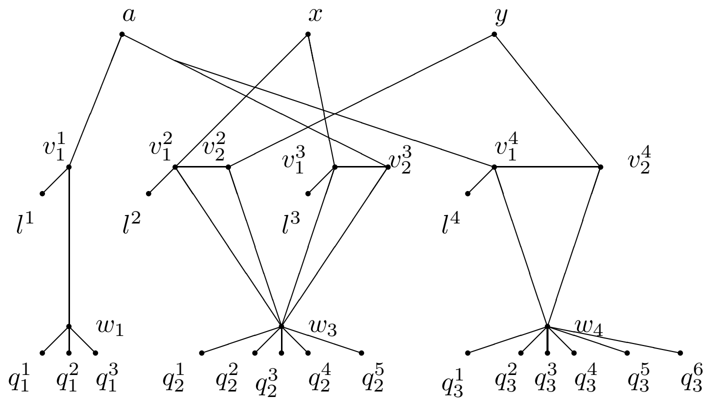

Рис. 2. Граф, соответствующий элементарной конъюнкции $p(a)\ \&\ q(x, y)\ \&\ q(x, a)\ \&\ \neg q(a, y)$.

ветствует вершина w_3; предикатному символу q, входящему с отрицанием, соответствует вершина w_4.

Построенный таким образом граф, ассоциированный с элементарной конъюнкцией предикатных формул, обладает следующими свойствами:

— каждая вершина w_i из множества $Pred$ смежна не менее чем с двумя висячими вершинами;

— каждая вершина w_i из множества $Pred$, соответствующая n_i-местному ($n_i > 1$) предикату, входит в k_i подграфов, каждый из которых состоит из цикла, содержащего $n_i - 1$ треугольников (циклов из трех вершин), в каждом из которых две другие вершины принадлежат множеству $Arity$, и смежна с $k_i + 2$ висячими вершинами из множества $Index$;

— каждая вершина w_i из множества $Pred$, соответствующая 1-местному предикату, смежна ровно с одной вершиной из $Arity$ и смежна с тремя висячими вершинами из множества $Index$;

— все вершины из множества $Index$ являются висячими и смежны с некоторой вершиной из множества $Pred$;

— каждая вершина из множества $Label$ является висячей и смежна с ровно одной вершиной из множества $Arity$;

— каждая вершина из множества $Arity$ (кроме смежных с вершинами, соответствующими 1-местному предикату) входит хоть в один треугольник и смежна с ровно одной вершиной из множества $Pred$, входящей в этот треугольник, и по крайней мере с одной вершиной множества Arg;

— каждая вершина из множества $Arity$, соответствующая 1-местному предикату, смежна ровно с одной вершиной из множества $Label$, ровно с одной вершиной из множества $Pred$ и ровно с одной вершиной из множества Arg;

— ни одна вершина из множества Arg не принадлежит ни одному треугольнику;

— каждая висячая вершина принадлежит либо множеству $Index$, либо множеству $Label$, либо множеству Arg. В последнем случае такая вершина соответствует первому аргументу литерала и этот аргумент входит в формулу единственный раз.

Теорема 2. *Задача ИПФ полиномиально сводится к задаче ИГ.*

Доказательство. Если элементарные конъюнкции P и Q имеют различный набор (например, кратность вхождений без отрицания и кратность вхождений с отрицанием) предикатных символов, то они не изоморфны и им можно поставить в соответствие любые два неизоморфных графа, например, пустой с одной вершиной и пустой с двумя вершинами. В дальнейшем изложении будем предполагать, что они имеют одинаковый набор предикатных символов, включая их кратность вхождений без отрицания и кратность вхождений с отрицанием.

По исходным данным P и Q с одинаковым набором предикатных символов для задачи ИПФ построим исходные данные G_P и G_Q для задачи ИГ в виде ассоциированных с ними графов. При этом множества вершин $Pred$, $Index$, $Arity$ и $Label$ у этих графов совпадают.

Покажем, что графы G_P и G_Q выписываются за число шагов, ограниченное полиномом от длины записи P и Q. Пусть N — количество предикатных символов в формулах, сосчитанных по отдельности с отрицанием и без отрицания. Тогда для каждого из этих графов количество элементов в каждом из множеств $Pred$, $Index$, $Arity$, $Lable$ и Arg не превосходит соответственно

— N (и, следовательно, длины записи P или Q);

— $\frac{N(N-1)}{2}$ (и, следовательно, длины записи P или Q в квадрате);

— количества вхождений аргументов в P или Q (и, следовательно, длины записи P или Q);

— количества литералов в P или Q (и, следовательно, длины записи P или Q);

— количества различных аргументов в P или Q (и, следовательно, длины записи P или Q).

Возведя в квадрат число вершин графа получим верхнюю (очень грубую) оценку числа ребер.

Умножив суммарное количество вершин и ребер графа на логарифм числа вершин получим, что длина записи графов не превосходит полинома от длины записи P и Q.

Покажем, что элементарные конъюнкции P и Q изоморфны тогда и только тогда, когда изоморфны графы G_P и G_Q.

Пусть две элементарные конъюнкции P и Q с аргументами $(a_1, \ldots, a_m)$ и $(b_1, \ldots, b_m)$ соответственно изоморфны, т. е. существует такая элементарная конъюнкция R и подстановки аргументов $a_1, \ldots, a_m$ и $b_1, \ldots, b_m$ формул P и Q соответственно вместо всех вхождений переменных $x_1, \ldots, x_m$ формулы R, что результаты этих подстановок в R совпадают с формулами P и Q соответственно с точностью до порядка литералов.

Из их изоморфности следует, что они имеют одинаковый набор (включая кратность вхождений без отрицания и кратность вхождений с отрицанием) предикат-

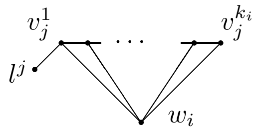

Рис. 3. Подграф, в который входит вершина w_i.

ных символов. Следовательно, подграфы, индуцированные множеством вершин $Pred \cup \mathrm{Index} \cup Arity \cup Label$, изоморфны при тождественном отображении вершин из этих множеств.

Биекция между остальными вершинами графов G_P и G_Q — это биекция между множествами $\{a_1, \ldots, a_n\}$ и $\{b_1, \ldots, b_n\}$, определяемая как $(a_{i_1} \to b_{j_1}, \ldots, a_{i_n} \to b_{j_n})$.

Пусть два графа G_P и G_Q, ассоциированные с элементарными конъюнкциями P и Q, изоморфны.

Покажем, что при любом изоморфизме графов G_P и G_Q вершины из $Pred \cup Index \cup Arity$ отобразятся в одноименные вершины другого графа. Более того, этот изоморфизм можно изменить так, что вершины из $Label$ одного графа отобразятся в одноименные вершины из $Label$ другого графа.

Действительно, только вершины из множества $Pred$ имеют более двух висячих смежных вершин. При этом для любых двух вершин из $Pred$ количества смежных с ними висячих вершин различны. Следовательно, при любом изоморфизме графов G_P и G_Q вершины из $Pred_P$ отображаются в одноименные вершины из $Pred_Q$.

Кроме того, каждая вершина w_i из $Pred_P$ и $Pred_Q$ входит в n_i подграфов вида, изображенного на рис. 3, где n_i — количество вхождений литералов с предикатом, соответствующим вершине w_i, k_i — количество аргументов соответствующего предикатного символа, l^j — висячая вершина, указывающая на первый аргумент литерала.

При произвольном изоморфизме графов G_P и G_Q такие подграфы могут отображаться не обязательно тождественно, т. е. вместо индекса j на рис. 3 могут быть индексы j_1 и j_2 ($j_1 \neq j_2$). Это соответствует перестановке литералов в элементарных конъюнкциях.

Кроме того, вершина l^j может отображаться в висячую вершину множества Arg, соответствующую первому аргументу литерала, если этот аргумент входит в формулу единственный раз. Тогда можно изменить изоморфизм таким образом, чтобы вершина l^j отображалась в одноименную вершину.

Таким образом, изоморфизм можно изменить так, чтобы l^j отображалась в l^j.

Следовательно, если G_P и G_Q изоморфны, то можно построить изоморфизм, при котором вершины из $Pred_P \cup Index_P \cup Arity_P \cup Label_P$ переходят в одноименные вершины из $Pred_Q \cup Index_Q \cup Arity_Q \cup Label_Q$.

Для вершин множеств Arg_P и Arg_Q при полученном изоморфизме также имеется взаимно однозначное соответствие $(a_{i_1} \to b_{j_1}, \ldots, a_{i_n} \to b_{j_n})$. Поэтому можно записать элементарную конъюнкцию R с переменными $x_1, \ldots, x_n$ и тем же набором предикатных символов, из которой можно получить (с точностью до порядка литералов) формулы P и Q в результате подстановок $(x_1 \to a_{i_1}, \ldots, x_n \to a_{i_n})$ и $(x_1 \to b_{j_1}, \ldots, x_n \to b_{j_n})$. Следовательно, P и Q изоморфны. ■

Из доказанных теорем непосредственно следует полиномиальная эквивалентность задач ИПФ и ИГ.

Теорема 3. *Задача ИПФ полиномиально эквивалентна задаче ИГ.*

Заключение. В работе доказана полиномиальная эквивалентность задач ИЗОМОРФИЗМ ГРАФОВ и ИЗОМОРФИЗМ ПРЕДИКАТНЫХ ФОРМУЛ. Из этого следует, что последняя задача является GI-полной.

Литература

1. *Babai L.* Graph Isomorphism in Quasipolynomial Time (Version 2.1 Unfinished revision of Version 2 posted on arXiv May 23, 2017). URL: http://people.cs.uchicago.edu/~laci/17groups/version2.1.pdf (дата обращения: 21.03.19).

2. *Земляченко В. Н., Корнеенко Н. М. Тышкевич Р. И.* Проблема изоморфизма графов // Записки научных семинаров ЛОМИ. 1982. Т. 118. С. 83–158.

3. *Arvind V., Toran J.* Isomorphism testing: Perspectives and open problems // Bulletin of the European Association for Theoretical Computer Science. 2005. Vol. 86. P. 66–84.

4. *Johnson D. S.* The NP-Completeness Column // ACM Transactions on Algorithms. 2005. Vol. 1. P. 160–176.

5. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи: Пер. с англ. М.: Мир, 1982.

6. *Kosovskaya T.* Predicate Calculus as a Tool for AI Problems Solution: Algorithms and Their Complexity // In: Intelligent System. Open access peer-reviewed chapter. Edited by Chatchawal Wongchoosuk Kasetsart University. URL: https://www.intechopen.com/books/intelligent-system/predicate-calculus-as-a-tool-for-ai-problems-solution-algorithms-and-their-complexity (дата обращения: 21.03.19).

7. *Косовская Т. М.* Доказательства оценок числа шагов решения некоторых задач распознавания образов, имеющих логические описания // Вестн. С.-Петерб. ун-та. Сер. 1. Математика. Механика. Астрономия. 2007. Вып. 4. С. 82–90.

8. *Косовская Т. М.* Некоторые задачи искусственного интеллекта, допускающие формализацию на языке исчисления предикатов, и оценки числа шагов их решения // Труды СПИИРАН, 2010. Вып. 14. С. 58–75.

9. *Косовская Т. М.* Многоуровневые описания классов для уменьшения числа шагов решения задач распознавания образов, описываемых формулами исчисления предикатов // Вестн. С.-Петерб. ун-та. Сер. 10. Прикладная математика. Информатика. Процессы управления. 2008. Вып. 1. С. 64–72.

10. *Косовская Т. М.* Подход к решению задачи построения многоуровневого описания классов на языке исчисления предикатов // Труды СПИИРАН. 2014. № 3(34). С. 204–217.

11. *Косовская Т. М.* Построение многоуровневой базы для уменьшения вычислительной сложности решения задачи конъюнктивный булевский запрос // Материалы конференции «Информационные технологии в управлении» (ИТУ-2018), 2–4 октября 2018 г., Санкт-Петербург, 2018. С. 33–38.

Статья поступила в редакцию 28 ноября 2018 г.;
после доработки 11 марта 2019 г.;
рекомендована в печать 21 марта 2019 г.

Контактная информация:

Косовская Татьяна Матвеевна — д-р физ.-мат. наук, проф.; kosovtm@gmail.com
Косовский Николай Николаевич — канд. физ.-мат. наук, доц.; kosovnn@pdmi.ras.ru

Polynomial equivalence of the problems PREDICATE FORMULAS ISOMORPHISM and GRAPH ISOMORPHISM

T. M. Kosovskaya, N. N. Kosovskii

St. Petersburg State University, Universitetskaya nab., 7–9, St. Petersburg, 199034, Russian Federation

For citation: Kosovskaya T. M., Kosovskii N. N. Polynomial equivalence of the problems PREDICATE FORMULAS ISOMORPHISM and GRAPH ISOMORPHISM. *Vestnik of Saint Peters-*

burg University. Mathematics. Mechanics. Astronomy, 2019, vol. 6 (64), issue 3, pp. 430–439. https://doi.org/10.21638/11701/spbu01.2019.308 (In Russian)

A problem of isomorphism checking of two elementary conjunctions of predicate formulas is under consideration. Such a problem appears while solving some Artificial Intelligence problems, admitting formalization by means of predicate calculus language. Polynomial equivalence of this problem with the Graph Isomorphism (GI) problem is proved.

Keywords: graph isomorphism, predicate formulas isomorphism, GI-completeness.

References

1. Babai L., *Graph Isomorphism in Quasipolynomial Time* (Version 2.1 Unfinished revision of Version 2 posted on arXiv May 23, 2017). Available at http://people.cs.uchicago.edu/~laci/17groups/version2.1.pdf (accessed March 21, 2019).

2. Zemlyachenko V. N., Korneenko N. M., Tyshkevich R. I., "Graph isomorphism problem", *Journal of Mathematical Sciences* **29**(4), 1426–1481 (1985).

3. Arvind V., Toran J., "Isomorphism testing: Perspectives and open problems", *Bulletin of the European Association for Theoretical Computer Science* **86**, 66–84 (2005).

4. Johnson D. S., "The NP-Completeness Column", *ACM Transactions on Algorithms* **1**(1), 160–176 (2005).

5. Garey M. R., Johnson D. S., *Computers and Intractability: a Guide to the Theory of NP-completeness* (Freeman, New York, 1979).

6. Kosovskaya T., *Predicate Calculus as a Tool for AI Problems Solution: Algorithms and Their Complexity*, in *Intelligent System* (Open access peer-reviewed chapter, edited by Chatchawal Wongchoosuk Kasetsart University, 2018). Available at https://www.intechopen.com/books/intelligent-system/predicate-calculus-as-a-tool-for-ai-problems-solution-algorithms-and-their-complexity (accessed March 21, 2019).

7. Kosovskaya T. M., "Estimating the number of steps that it takes to solve some problems of pattern recognition which admit logical description", *Vestn. St. Petersburg Univ.: Math.* **40**, issue 4, 287–293 (2007). https://doi.org/10.3103/S1063454107040061

8. Kosovskaya T. M., "Some artificial intelligence problems permitting formalization by means of predicate calculus language and upper bounds of their solution steps", *SPIIRAS Proceedings* **14**, 58–75 (2010). (In Russian)

9. Kosovskaya T. M., "Level descriptions of classes for decreasing of step number of solving of a pattern recognition problem described by predicate calculus formulas", *Vestn. St. Petersburg Univ.: Seriya 10. Applied Mathematics. Computer Science. Control Processes*, issue 1, 64–72 (2008). (In Russian)

10. Kosovskaya T. M., "An approach to the construction of a level description of classes by means of a predicate calculus language", *SPIIRAS Proceedings* **3(34)**, 204–217 (2010). (In Russian)

11. Kosovskaya T. M., "A level base construction for computational complexity decreasing of Conjunctive Boolean Query problem", *Proceedings of conference «Information Technologies in Control» (ITC-2018), 2–4 October 2018, St. Petersburg*, 33–38 (2018). (In Russian)

Received: November 28, 2018
Revised: March 11, 2019
Accepted: March 21, 2019

Author's information:

Tatiana M. Kosovskaya — kosovtm@gmail.com
Nikolay N. Kosovskii — kosovnn@pdmi.ras.ru