

Полиномиальный алгоритм тестирования графов на изоморфизм

М.И.Трофимов

18 июня 2013 г.

email: mt2@comtv.ru

(авторский перевод с английского оригинала препринта на <http://arxiv.org/abs/1004.1808v6>)

Аннотация

В статье предложен новый полиномиальный алгоритм тестирования графов на изоморфизм.

Введение

Согласно определению Харари, два графа изоморфны, если между множествами их вершин существует взаимнооднозначное соответствие, сохраняющее смежность. То же самое на языке матричной алгебры звучит следующим образом: два графа с матрицами смежности A и A' изоморфны тогда и только тогда, когда существует такая матрица перестановки P , что $A = P^{-1} \cdot A' \cdot P$. [9]

Несмотря на непрекращающиеся усилия многих исследователей, вопрос, является ли задача изоморфизма графов NP-трудной, до сих пор остается открытым. Интересное объяснение этому факту приведено в книге [6]. Авторы отмечают, что для доказательства NP-полноты требуется определенная избыточность задачи, которой не обладает задача изоморфизма графов. Так, например, в случае задачи изоморфного вложения добавление или удаление некоторых ребер графа (не принадлежащих к образу отображения) не влияет на результат, в задаче изоморфизма добавление или удаление любого ребра одного из графов изменяет результат на противоположный. Таким образом, задача изоморфизма графов не является типичной NP-задачей.

Также важно отметить, что для отдельных классов графов были предложены эффективные полиномиальные алгоритмы тестирования на изоморфизм. В частности, такой алгоритм был доказан для деревьев [1, 3, 5].

Основные принципы предлагаемого подхода

Не нарушая общности, будем искать решение задачи изоморфизма для неориентированных связных графов без кратных ребер и петель [7, 8]:

$$G_s = (V, E_s),$$

где V - множество вершин, $|V| = n$;

E_s - множество ребер, $|E_s| = t$.

Назовем такие графы G_s *исходными* или *S-графами*.

Во избежание терминологической путаницы повторим также некоторые широко известные определения, которые понадобятся в дальнейшем. *Объединением графов* $G \cup G'$ является граф с множеством вершин $V \cup V'$ и множеством ребер $E \cup E'$ [9]. Аналогично, *пересечением графов* $G \cap G'$ является граф с множеством вершин $V \cap V'$ и множеством ребер $E \cap E'$. Отметим, что такое определение допускает "пустой граф" с $V = \emptyset$, $E = \emptyset$.

В соответствии с определением Харари: "Две вершины u и v графа G называются *подобными*, если для некоторого автоморфизма α этого графа $\alpha(u) = v$ " [9]. Расширим это определение:

Определение 1. Вершина v графа G и вершина v' графа G' *подобны*, если существует изоморфизм π графа G на граф G' такой, что $\pi(v) = v'$. \square

Аналогичное определение возможно и для ребер:

Определение 2. Ребро e графа G и ребро e' графа G' *подобны*, если существует изоморфизм π графа G на граф G' такой, что $\pi(e) = e'$. \square

Было также замечено, что: "если α – автоморфизм графа G , то графы $G - u$ и $G - \alpha(u)$ изоморфны. Поэтому, если вершины v и u подобны, то $G - v \cong G - u$ " [9]. Из этого замечания, с учетом приведенных выше расширительных определений 1, 2, следует:

Вывод 1. Если $G \cong G'$, и вершины v, v' подобны, то $G - v \cong G' - v'$. \square

Вывод 2. Если $G \cong G'$, π – такой изоморфизм, что $e = \pi(e')$, $e \in E$, $e' \in E'$, то $G - e \cong G' - e'$ и π будет изоморфизмом и для этих графов. \square

Лемма 1. Если $G \cong G'$, вершины $v \in V$, $v' \in V'$ подобны, ребро $(v, w) \in E$, то существует такая вершина $w' \in V'$, смежная вершине v' , что ребра $(v', w') \in E'$, (v, w) и вершины w, w' соответственно подобны. \square

Доказательство. Пусть графы G и G' изоморфны, вершины $v \in V$, $v' \in V'$ подобны. Тогда существует изоморфизм $v' = \pi(v)$. Пусть $w_1, w_2, \dots, w_k \in V$ смежны с v , любая другая вершина не смежна с v . Т.к. v и v' подобны, то их степени равны [4]. Пусть вершины $w'_1, w'_2, \dots, w'_k \in V'$ смежны с v' , любая другая вершина не смежна с v' . Предположим, что для вершины w_i , $1 \leq i \leq k$ не найдется подобной вершины w'_j , $1 \leq j \leq k$. Тогда

$x' = \pi(w_i)$, $x' \in V'$, $x' \notin \{w'_1, w'_2, \dots, w'_k\}$. Таким образом вершина x' не смежная с v' , в то время как вершина w_i смежная с v , т.о. смежность при отображении π нарушается, а значит π не изоморфизм. Полученное противоречие показывает, что сделанное нами предположение неверно. Значит, для вершины w_i всегда найдется подобная вершина w'_j , смежная вершине v' . \square

Обозначим расстояние между вершинами $v, u \in V$ графа как $\text{dist}(v, u)$ и будем считать расстояние между любой парой смежных вершин равным единице. Из леммы 1 следует:

Вывод 3. Пусть $G \cong G'$, а вершины $v, w \in V$ подобны соответственно вершинам $v', w' \in V'$, тогда $\text{dist}(v, w) = \text{dist}(v', w')$. \square

Лемма 2. Пусть $G \cong G'$, причем возможен такой изоморфизм π , что для несмежных вершин $v, w \in V$, $v', w' \in V'$: $v = \pi(v')$, $w = \pi(w')$, тогда, если вершины v, w соединить ребром и вершины v', w' соединить ребром, то полученные графы G_2, G'_2 будут изоморфны и π – возможный изоморфизм. \square

Доказательство. Т.к. $\pi : G \cong G'$, то $\pi : \bar{G} \cong \bar{G}'$, где $\bar{G} = (V, \bar{E})$, $\bar{G}' = (V', \bar{E}')$ – дополнительные графы. По условию вершины v, w (и v', w') не смежны, значит в дополнительных графах существуют ребра $(v, w) \in \bar{E}$ и $(v', w') \in \bar{E}'$. Из вывода 2 следует, что $\pi : \bar{G} - (v, w) \cong \bar{G}' - (v', w')$. Обозначив $\bar{G} - (v, w) = H$, $\bar{G}' - (v', w') = H'$, получим $\pi : H \cong H'$. Отсюда $\pi : \bar{H} \cong \bar{H}'$. Но $\bar{H} = (V, E \cup (v, w)) = G_2$, $\bar{H}' = (V', E' \cup (v', w')) = G'_2$, следовательно, $\pi : G_2 \cong G'_2$. \square

Определим процедуру P_B . Каждое ребро (i, j) исходного графа G_s заменим *дополнительной вершиной* k и ребрами (i, k) , (k, j) . Раскрасим оранжевым дополнительные вершины и черным вершины исходного графа. \square

Очевидно, что преобразованный таким образом граф будет двудольным графом, где оранжевые вершины будут составлять первую долю, а черные вершины - вторую:

$$G_B = (U, V, E),$$

где U - подмножество вершин первой доли (дополнительные оранжевые вершины), $|U| = m$;

V - подмножество вершин второй доли (черные вершины исходного графа), $|V| = n$;

E - множество ребер, $|E| = 2m$.

Назовем такие графы G_B *B-графами*. Отметим, что по построению степень любой оранжевой вершины будет 2. Очевидно, что

$$G_B \cong G'_B \iff G_s \cong G'_s.$$

Выберем произвольно вершину $t \in V$, которую назовем *стартовой*, и расположим вершины по уровням так, что на каждом уровне d будут находиться все вершины, удаленные от стартовой на расстояние d (рис.1).

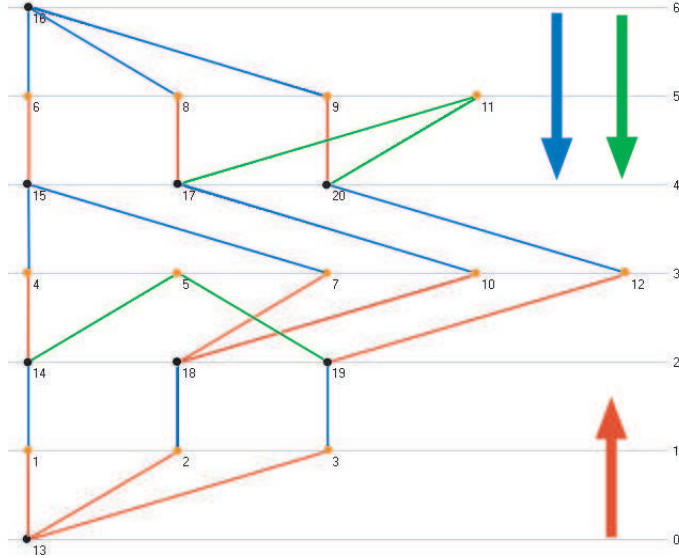


Рис. 1: $D_\alpha(t)$.

Отметим, что на нулевом уровне всегда будет располагаться единственная вершина t . На уровнях $d = 2k$, $k = 0, 1, 2, \dots$ будут располагаться вершины из V (черные), на уровнях $d = 2k + 1$ будут располагаться вершины из U (оранжевые). При этом из черных вершин четного уровня $d = 2k$ (за исключением последнего) к оранжевым вершинам следующего уровня $d = 2k + 1$ будут расти деревья высотой 1 (раскрасим ребра этих деревьев красным), и из каждого уровня $d = 2k$ (за исключением нулевого) к оранжевым вершинам предыдущего уровня $d = 2k - 1$ будут расти деревья высотой 1 (раскрасим ребра этих деревьев синим). Т.о. каждое из таких деревьев будет иметь один черный корень и один или более оранжевых листьев. Особо отметим деревья, состоящие из двух черных листьев, лежащих на одном уровне, и одного оранжевого корня. Ребра таких деревьев раскрасим зеленым. Деревья, состоящие из красных ребер, назовем *красными деревьями*, из синих - *синими деревьями* и из зеленых - *зелеными деревьями*, соответственно. Назовем такие деревья α -деревьями. Направление роста α -деревьев (от их корней к листьям) на рис. 1 показано стрелками соответствующих цветов. Процедуру выделения α -деревьев с выбранной стартовой вершиной t назовем α -декомпозицией B -графа и обозначим $D_\alpha(t)$.

Лемма 3. В результате D_α все ребра B -графа будут раскрашены. \square

Доказательство. Т.к. B -граф двудольный, то каждое его ребро (i, j) инцидентно одной черной i и одной оранжевой вершине j . Если i принадлежит уровню, расположенному выше j , то D_α раскрасит это ребро синим, если ниже, то красным или зеленым. Других вариантов быть не может. \square

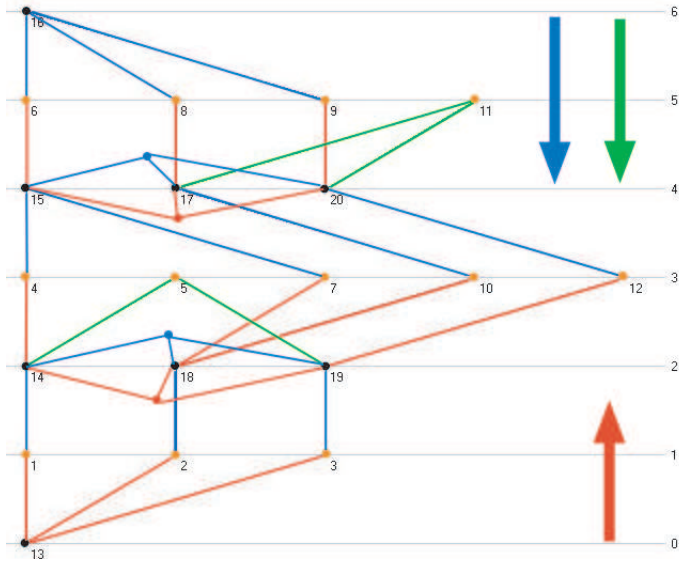


Рис. 2: $D_\beta(t)$.

Если на каком-либо четном уровне d растет больше одного α -дерева одного цвета, то эти деревья можно объединить в одно, соединив их корни с *дополнительной вершиной* того же цвета (рис. 2). Такие деревья будем называть β -деревьями и ставить им в соответствие уровень d . При этом если на уровне d растет только одно α -дерево какого-либо цвета, то такое дерево также назовем β -деревом. Прделанное построение назовем β -декомпозицией B -графа и обозначим $D_\beta(t)$, где t – стартовая вершина этой декомпозиции.

Лемма 4. Пусть B -графы $G_B = (U, V, E)$ и $G'_B = (U', V', E')$ изоморфны и вершины $t \in V, t' \in V'$ подобны. Проведем $D_\beta(t)$ и $D_\beta(t')$. Тогда общее число уровней L графов G_B и G'_B будет одинаковым и каждое β -дерево уровня $d, 0 \leq d < L$ графа G_B будет изоморфно β -дереву того же цвета уровня d графа G'_B . \square

Доказательство. Из построения следует, что каждому уровню d всегда соответствует не больше одного β -дерева одного цвета. Таким образом, в результате D_β мы получаем лес красных, синих и зеленых деревьев. Ясно, что распределение вершин по уровням зависит только от выбора стартовой вершины t , т.е. от расстояния до t , и не зависит от нумерации

вершин графа. Предположим, что черная вершина $v \in V$, лежащая на уровне d_1 , не подобна ни одной из вершин уровня d_1 графа G'_B . Тогда, т.к. графы G_B и G'_B изоморфны, эта вершина будет подобна вершине $v' \in V'$, лежащей на другом уровне d_2 , $d_2 \neq d_1$. Но тогда получаем, что $\text{dist}(t, v) \neq \text{dist}(t', v')$, что противоречит выводу 3. Значит, наше предположение неверно, и для каждой вершины $v \in V$, лежащей на уровне d_i , найдется подобная вершина $v' \in V'$, лежащая на уровне d_i . По лемме 1 соседи этих вершин также должны быть попарно подобны, а значит и β -деревья будут изоморфны. \square

Для определения необходимых в дальнейшем структур данных будем использовать БНФ с традиционными метасимволами:

$$\langle \rangle ::= \{ \} |$$

и дополнительным метасимволом:

◇

означающим, что следующая за ним последовательность отсортирована в убывающем порядке. При этом имеется в виду последовательность подстрок (подпоследовательностей), каждая из которых интерпретируется как неразрывное целое. Для сравнения строк (массивов) будем использовать широко известное правило: “Если имеются два массива x и y , то отношение $x < y$ выполняется в том и только в том случае, если существует индекс k такой, что $x[k] < y[k]$ и $x[i] = y[i]$ для всякого $i < k$ ” [2]. Для сравнения массивов неравной длины в конец более короткого массива дописываются нулевые элементы (в случае сравнения числовых массивов) или пробелы (в случае сравнения строк) так, чтобы длина сравниваемых массивов стала одинаковой.

Замечание 1. Сравнение строк осуществляется за время, прямо пропорциональное длине строки. \square

Определение 3.

- $\langle \text{пусто} \rangle ::=$
- $\langle \text{цифра} \rangle ::= 0|1|2|3|4|5|6|7|8|9$
- $\langle \text{число} \rangle ::= \langle \text{цифра} \rangle \{ \langle \text{цифра} \rangle \}$
- $\langle \text{кортеж} \rangle ::= \langle \text{пусто} \rangle | \langle \text{число} \rangle \{ , \langle \text{число} \rangle \}$ (1)
- $\langle \text{метка rgb} \rangle ::= \langle \text{кортеж} \rangle ; \langle \text{кортеж} \rangle ; \langle \text{кортеж} \rangle$ (2)
- $\langle \text{уровень} \rangle ::= \langle \text{число} \rangle$ (3)
- $\langle \text{степень} \rangle ::= \langle \text{число} \rangle$ (4)
- $\langle \text{простой код вершины} \rangle ::= \langle \text{уровень} \rangle ; \langle \text{степень} \rangle ; \langle \text{метка rgb} \rangle$ (5)
- $\langle \text{код вершины} \rangle ::= \langle \text{простой код вершины} \rangle \diamond \{ * \langle \text{код ребра} \rangle \}$ (6)
- $\langle \text{код ребра} \rangle ::= (\diamond \langle \text{код вершины} \rangle . \langle \text{код вершины} \rangle)$ (7)
- $\langle \text{инвариант вершины} \rangle ::= \diamond \langle \text{код вершины} \rangle \{ \& \langle \text{код вершины} \rangle \}$ (8)

□

Описанные структуры данных позволяют использовать классическую технику кортежирования вершин, применяемую в алгоритмах установления изоморфизма деревьев [1, 3, 5]. Для этого вершины дерева располагаются по уровням в зависимости от их расстояния до корня. Двигаясь от листьев к корню, ставим в соответствие каждому листу кортеж (1), состоящий из одной единицы. Кортеж уровня d записывается в виде:

$$p, s_1, s_2, \dots, s_k,$$

где s_1, s_2, \dots, s_k - кортежи вершин предыдущего уровня, $s_i \geq s_{i+1}$, $i = 1, \dots, k - 1$;

$$p = 1 + \sum_{j=1}^k s_j[1];$$

$s_j[1]$ - первый элемент кортежа s_j .

Отметим, что данная техника применима и к помеченным деревьям. В этом случае в кортежи добавляются метки вершин [1]. Принимая во внимание замечание 1, видим правомерность следующей теоремы.

Теорема 1. *Время, необходимое на кортежирование вершин дерева, линейно зависит от числа вершин [1, 3].* □

Также было доказано, что “для изоморфизма деревьев необходимо и достаточно, чтобы совпадали их центральные кортежи” [5]. Т.к. мы не рассматриваем бисентральные деревья, то можем переформулировать эту теорему следующим образом:

Теорема 2. *Для изоморфизма корневых деревьев необходимо и достаточно, чтобы совпадали кортежи их корней.* □

Вывод 4. *Кортеж вершины однозначно задает дерево (поддерево), корнем которого является эта вершина.* □

Важно отметить разницу между уровнями деревьев при кортежировании вершин (от листьев к корню) и уровнями B -графа, в котором красные и синие деревья растут во встречных направлениях. В дальнейшем из контекста будет ясно, о каком типе уровней идет речь.

Лемма 5. *Вершины двух изоморфных деревьев подобны тогда и только тогда, когда кортежи этих вершин и кортежи их предков (вплоть до корня) попарно совпадают.* □

Доказательство. Рассмотрим деревья $S_0 = (V, E)$, $S'_0 = (V', E')$ с корнями r и r' соответственно (рис.3).

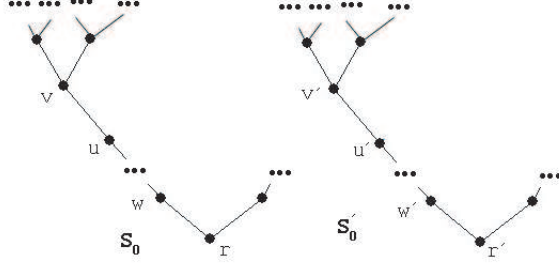


Рис. 3: Совпадающие кортежи предков.

Пусть у вершин v и v' и у всех их предков u, \dots, w, r и u', \dots, w', r' кортежи попарно совпадают, тогда из совпадения кортежей корней r и r' по теореме 2 следует, что $S_0 \cong S'_0$ и при этом $r \leftrightarrow r'$. Удалив r и r' , получаем поддеревья S_1 и S'_1 с корнями w и w' , кортежи которых совпадают по условию, следовательно, эти поддеревья изоморфны, и при этом $w \leftrightarrow w'$. Продолжая таким образом удалять корни, на k -ом удалении получим поддерева S_k и S'_k с корнями v и v' , кортежи которых совпадают, следовательно, эти поддеревья изоморфны, и при этом $v \leftrightarrow v'$. Рассмотрим полученные последовательности попарно изоморфных поддеревьев S_k, \dots, S_0 и S'_k, \dots, S'_0 как самостоятельные графы. Корневой кортеж деревьев S_{k-1} и S'_{k-1} может быть представлен как

$$s_u = (p + s_v[1], h_1, \dots, h_x, s_v, t_1, \dots, t_y),$$

где h_1, \dots, h_x – кортежи вершин предыдущего уровня, для которых $h_i > s_v$, $i = 1, \dots, x$, (могут отсутствовать);

s_v – кортеж вершины v ;

t_1, \dots, t_y – кортежи вершин предыдущего уровня, для которых $t_i \leq s_v$, $i = 1, \dots, y$, (могут отсутствовать);

$$p = 1 + \sum_{j=1}^x h_j[1] + \sum_{j=1}^y t_j[1];$$

$h_j[1]$ – первый элемент кортежа h_j ;

$t_j[1]$ – первый элемент кортежа t_j ;

$s_v[1]$ – первый элемент кортежа s_v .

Аналогично корневой кортеж дерева S'_{k-1} может быть представлен как

$$s'_u = (p' + s'_v[1], h'_1, \dots, h'_x, s'_v, t'_1, \dots, t'_y).$$

Удалив в деревьях S_{k-1} и S'_{k-1} ребра (v, u) и (v', u') , получим два леса F_{k-1} и F'_{k-1} , в каждом из которых по два дерева: дерево S_k (или S'_k для другого леса) и дерево Q_{k-1} (или Q'_{k-1}) с корнем u (или u'). Тогда, учитывая, что $s_v = s'_v$ и $s_u = s'_u$, получим, что корневые кортежи деревьев Q_{k-1} и Q'_{k-1} совпадают, и, следовательно, $Q_{k-1} \cong Q'_{k-1}$, $F_{k-1} \cong F'_{k-1}$, при этом

возможен изоморфизм с отображениями $v \leftrightarrow v'$, $u \leftrightarrow u'$. Добавив к F_{k-1} и к F'_{k-1} ребра (v, u) и (v', u') , получаем (лемма 2), что для изоморфизма деревьев S_{k-1} и S'_{k-1} возможны те же отображения $v \leftrightarrow v'$, $u \leftrightarrow u'$. Далее повторяем аналогичные рассуждения для деревьев S_{k-2} и S'_{k-2} и т.д., пока не дойдем до деревьев S_0 и S'_0 : удалив ребра (w, r) и (w', r') , получим два леса F_0 и F'_0 , в каждом из которых по два дерева: дерево S_1 (или S'_1) и дерево Q_0 (или Q'_0) с корнем r (или r'). Корневые кортежи деревьев Q_0 и Q'_0 совпадают, и, следовательно, $Q_0 \cong Q'_0$, $F_0 \cong F'_0$, при этом возможен изоморфизм с отображениями $r \leftrightarrow r'$, $w \leftrightarrow w'$, ..., $v \leftrightarrow v'$, добавив к F_0 и к F'_0 ребра (w, r) и (w', r') , получим, что для изоморфизма деревьев S_0 и S'_0 возможны те же отображения, а значит вершины v и v' подобны.

Пусть теперь кортеж вершины u не совпадает с кортежем вершины u' (рис.4).

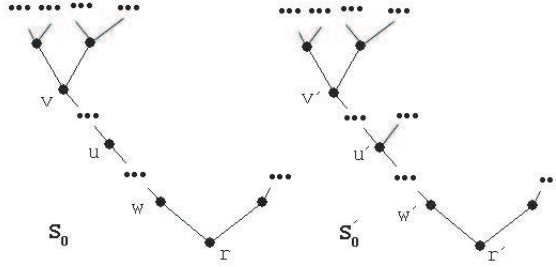


Рис. 4: Несовпадающие кортежи предков.

Из совпадения кортежей корней r и r' следует, что $S_0 \cong S'_0$ и $r = \pi_0(r')$, где π_0 – изоморфизм. Удалив r и r' , получаем поддеревья с корнями w и w' , кортежи которых также совпадают, следовательно, эти поддеревья изоморфны и $w = \pi_1(w')$, где π_1 – изоморфизм. Продолжая таким образом последовательно удалять корни, доходим до поддеревьев с корнями u и u' , кортежи которых не совпадают, следовательно, эти поддеревья не изоморфны, а вершины v и v' не подобны. \square

Рассмотрим графы, состоящие из двух подграфов H_1 и H_2 , эти подграфы имеют общие вершины v_1, v_2, \dots, v_k , причем не существует ребер (v_i, v_j) , $i, j = 1, 2, \dots, k$ (рис. 5). Докажем для таких графов следующую лемму.

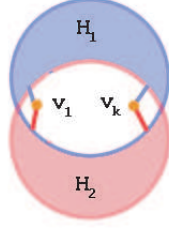


Рис. 5: Граф, образованный двумя подграфами с общими вершинами.

Лемма 6. *Если*

- 1) $G = (V, E)$, $G' = (V', E')$,
 - 2) $G = H_1 \cup H_2$, $G' = H'_1 \cup H'_2$, $H_1 = (V_1, E_1)$, $H_2 = (V_2, E_2)$, $H'_1 = (V'_1, E'_1)$, $H'_2 = (V'_2, E'_2)$,
 - 3) $H_1 \cap H_2 = (X, \emptyset)$, $X = \{v_1, v_2, \dots, v_k\}$, $v_i \in V$, $i = 1, 2, \dots, k$, $H'_1 \cap H'_2 = (X', \emptyset)$, $H'_1 \cap H'_2 = \{v'_1, v'_2, \dots, v'_k\} = X'$, $v'_i \in V'$,
 - 4) Не существует ребер (v_i, v_j) , (v'_i, v'_j) , $v_i, v_j \in X$, $v'_i, v'_j \in X'$, $i, j = 1, 2, \dots, k$,
 - 5) $H_1 \cong H'_1$, $H_2 \cong H'_2$, и Π_1, Π_2 - соответствующие множества изоморфизмов,
 - 6) $\exists \pi_1, \pi_2 : \pi_1 \in \Pi_1, \pi_2 \in \Pi_2$, такие, что $v_i = \pi_1(v'_i) = \pi_2(v'_i)$, $v_i \in X$, $v'_i \in X'$, $i, j = 1, 2, \dots, k$,
- то $G \cong G'$. □

Доказательство. Пусть $U_1 = V_1 \setminus X$, $U_2 = V_2 \setminus X$, $U'_1 = V'_1 \setminus X'$, $U'_2 = V'_2 \setminus X'$, $p = |U_1|$, $q = |U_2|$, $s = k + p + q$. Отметим, что т.к. по условию 5 $H_1 \cong H'_1$, то $|U'_1| = p$, и т.к. $H_2 \cong H'_2$, то $|U'_2| = q$. Не нарушая общности, пронумеруем вершины графов в следующем порядке. Граф G : вершины из X , U_1 , U_2 . Аналогично граф G' : вершины из X' , U'_1 , U'_2 . Тогда $s \times s$ матрица смежности графа G будут иметь следующий вид:

$$A = \begin{pmatrix} O_1 & B_1 & B_2 \\ B_1^T & C_1 & O_2 \\ B_2^T & O_3 & C_2 \end{pmatrix},$$

- где блок O_1 размера $k \times k$ нулевой по условию 4;
блок B_1 размера $k \times p$ соответствует ребрам (v, u) , $v \in X$, $u \in U_1$;
блок B_2 размера $k \times q$ соответствует ребрам (v, u) , $v \in X$, $u \in U_2$;
блок C_1 размера $p \times p$ соответствует ребрам (v, u) , $v, u \in U_1$;
блок C_2 размера $q \times q$ соответствует ребрам (v, u) , $v, u \in U_2$;
блоки O_2 размера $p \times q$ и O_3 размера $q \times p$ - нулевые, т.к. по условию 3 не существует ребер (v, u) таких, что $v \in U_1$, $u \in U_2$ (Табл. 1).

Подмножество вершин		X	U_1	U_2
	Мощность	k	p	q
X	k	O_1 $k \times k$	B_1 $k \times p$	B_2 $k \times q$
U_1	p	B_1^T $p \times k$	C_1 $p \times p$	O_2 $p \times q$
U_2	q	B_2^T $q \times k$	O_3 $q \times p$	C_2 $q \times q$

Таблица 1: Блоки матрицы A и их размеры.

Аналогичный вид будет иметь и матрица смежности графа G' :

$$A' = \begin{pmatrix} O_1 & B'_1 & B'_2 \\ B_1'^T & C'_1 & O_2 \\ B_2'^T & O_3 & C'_2 \end{pmatrix}.$$

Сохраняя принятую нами нумерацию вершин, матрицы смежности подграфов H_1 , H_2 можно представить в виде $s \times s$ матриц. Для этого к множеству вершин U_1 подграфа H_1 добавляем вершины U_2 , а к множеству вершин U_2 подграфа H_2 добавляем вершины U_1 . Очевидно, что от такого добавления изолированных вершин смежность в подграфах не изменится. Получим:

$$A_1 = \begin{pmatrix} O_1 & B_1 & O \\ B_1^T & C_1 & O_2 \\ O & O_3 & O \end{pmatrix},$$

$$A_2 = \begin{pmatrix} O_1 & O & B_2 \\ O & O & O_2 \\ B_2^T & O_3 & C_2 \end{pmatrix},$$

где O – нулевые блоки соответствующего размера.

Аналогичный вид будут иметь и матрицы смежности подграфов H'_1 , H'_2 после добавления изолированных вершин:

$$A'_1 = \begin{pmatrix} O_1 & B'_1 & O \\ B_1'^T & C'_1 & O_2 \\ O & O_3 & O \end{pmatrix},$$

$$A'_2 = \begin{pmatrix} O_1 & O & B'_2 \\ O & O & O_2 \\ B_2'^T & O_3 & C'_2 \end{pmatrix}.$$

Как можно видеть, $A = A_1 + A_2$ и $A' = A'_1 + A'_2$.

Пусть P – $s \times s$ матрица перестановки для изоморфизмов π_1 , π_2 : $A_1 = P^{-1} \cdot A'_1 \cdot P$ и $A_2 = P^{-1} \cdot A'_2 \cdot P$. Согласно условию 6 эта матрица нужным для сохранения смежности образом меняет местами столбцы и строки матриц

смежности подграфов, соответствующие общим вершинам X' , результатом перестановки для вершин из U'_2 в матрице A'_1 будет перестановка нулевых столбцов и нулевых строк, аналогичный результат будет для вершин U'_1 в матрице A'_2 .

В результате получаем:

$$\begin{aligned}
 A &= A_1 + A_2 = P^{-1} \cdot A'_1 \cdot P + P^{-1} \cdot A'_2 \cdot P = P^{-1} \cdot (A'_1 + A'_2) \cdot P = \\
 &P^{-1} \cdot \left(\left(\begin{array}{ccc} O_1 & B'_1 & O \\ B_1'^T & C'_1 & O_2 \\ O & O_3 & O \end{array} \right) + \left(\begin{array}{ccc} O_1 & O & B'_2 \\ O & O & O_2 \\ B_2'^T & O_3 & C'_2 \end{array} \right) \right) \cdot P = \\
 &P^{-1} \cdot \left(\begin{array}{ccc} 0 & B'_1 & B'_2 \\ B_1'^T & C'_1 & 0 \\ B_2'^T & 0 & C'_2 \end{array} \right) \cdot P = P^{-1} \cdot A' \cdot P.
 \end{aligned}$$

Следовательно, $G \cong G'$. □

По построению B -графа каждая черная вершина $v \in V$ может быть корнем не более трех α -деревьев разного цвета. Таким образом, записав в строку кортежи для красного (r), зеленого (g) и синего (b) α -деревьев с корнем v , получаем *метку rgb* (2) (см. определение 3). И, таким образом, каждой черной вершине графа мы можем поставить в соответствие метку rgb , однозначно характеризующую деревья (вывод 4). Далее, дописав в начало этой строки значение *уровня d* (3) и *степень вершины* (4), мы получаем *простой код* (5) каждой черной вершины. Для каждого ребра (i, j) исходного графа G_s записываем *код ребра* (7). И снова вычисляем *код каждой вершины* (6) уже с учетом кодов инцидентных ей ребер. С каждой итерацией "вычисление кода вершины - вычисление кода ребра" в кодах каждой вершины и каждого ребра отображается информация о все более и более удаленных вершинах и ребрах. Чтобы в кодах присутствовала информация обо всех вершинах и ребрах, число итераций в общем случае достаточно ограничить диаметром графа (обозначим его как $\text{dm}(G)$). Перебрав все черные вершины в качестве стартовых для декомпозиции и отсортировав решения, получаем *вершинные инварианты графа* (8), не зависящие от нумерации вершин. Этот процесс описывается на псевдокоде *процедурой P_C* (алгоритм 1).

Алгоритм 1 Процедура P_C .

1. вызвать процедуру P_B для исходного графа G_s , в результате получим граф G_B ;
 2. **for** $i := 1$ **to** n **do**
 3. **begin**
 4. провести $D_\alpha(v_i)$, $v_i \in V$;
 5. вычислить метку rgb (2) (см. определение 3) для каждой черной вершины графа G_B ;
 6. вычислить простой код (5) для каждой черной вершины графа G_B , присвоить эти коды кодам вершин (6);
 7. **for** $j := 1$ **to** $\text{dm}(G_s)$ **do**
 8. **begin**
 9. вычислить код ребра (7) для каждого ребра графа G_s ;
 10. для каждой вершины графа G_s вычислить код вершины (6), учитывая коды инцидентных этой вершине ребер;
 11. **end;**
 12. записать код каждой вершины : $C[j, i] := c(v_j)$, где $c(v_j)$ – код вершины $v_j \in V$;
 13. **end;**
 14. сортировать каждую строку матрицы C в отдельности;
 15. сортировать матрицу C по строкам;
 16. построить из каждой i -ой строки матрицы C инвариант вершины (8), записать его в i -ую координату вектора S ;
-

Рассмотрим подграфы B -графа, определяемые β -декомпозицией и образованные встречными β -деревьями с общими оранжевыми вершинами (рис. 6). Черные корни красных α -деревьев, составляющих такой подграф, лежат на уровне $2k$, $k = 0, 1, 2, \dots$, черные корни синих α -деревьев лежат на уровне $2k + 2$, оранжевые корни зеленых α -деревьев лежат на уровне $2k + 3$, и общие вершины - листья синего и красного β -деревьев лежат на уровне $2k + 1$. Назовем такие подграфы γ -графами.

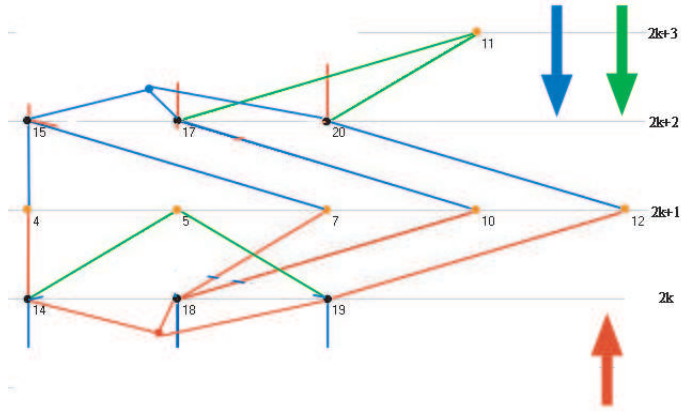


Рис. 6: γ -граф.

Очевидно, каждому четному уровню $2k$ может соответствовать не более двух γ -графов: один из них может быть образован красными деревьями, растущими вверх, а другой синими, растущими вниз. Т.е. нулевому уровню соответствует один γ -граф, образованный единственным красным α -деревом, а последнему верхнему уровню соответствует один γ -граф, образованный синим и/или зеленым β -деревьями. Каждому нечетному уровню $2k + 1$ всегда соответствует только один γ -граф (корни зеленых деревьев нижележащего γ -графа к данному γ -графу не относятся).

Лемма 7. Пусть в результате $D_\beta(t)$, $t \in V$, B -графа G_B получено множество γ -графов $\{\Gamma_1, \Gamma_2, \dots\}$, тогда $(\bigcup_{(i)} \Gamma_i) \cap G_B = G_B$. \square

Доказательство. По лемме 3 каждое ребро и инцидентные ему вершины B -графа G_B входят в какое-либо β -дерево. Каждое β -дерево входит в какой-нибудь γ -граф по определению. \square

Удалив из γ -графа оранжевые корни зеленых α -деревьев, получим граф, который назовем δ -графом (рис. 7). Корень синего и корень красного β -деревьев, образующих δ -граф, назовем *синим и красным корнями δ -графа*. Если корень δ -графа образован синей или красной дополнительной вершиной, то назовем его *дополнительным корнем*.

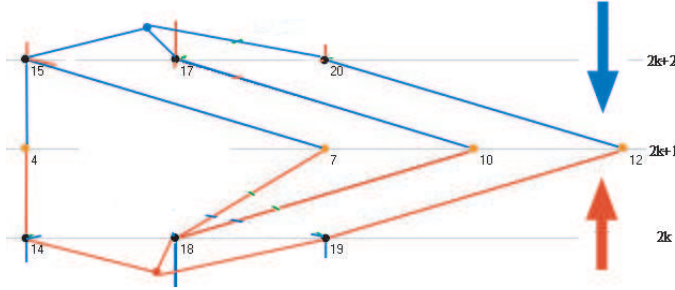


Рис. 7: δ -граф.

В следующей лемме мы учтем, что δ -графы – это подграфы, и будем использовать коды вершин (6) (см. определение 3), в которые входят уровни (3), т.о. речь будет идти о помеченных δ -графах, т.е. о δ -графах одного уровня.

Лемма 8. *Для изоморфизма δ -графов одного уровня необходимо и достаточно, чтобы коды вершин их синих и красных корней соответственно совпадали.* \square

Доказательство. Пусть графы $G = (V, E)$ и $G' = (V', E')$ являются δ -графами. Обозначим множество оранжевых вершин этих графов $X = \{v_1, v_2, \dots, v_k\}$, $X' = \{v'_1, v'_2, \dots, v'_k\}$ соответственно, синий и красный корни r_b, r_r, r'_b, r'_r соответственно, синее и красное деревья T_b, T_r, T'_b, T'_r соответственно.

Предположим, что коды синих и красных корней соответственно совпадают. Из этого следует, что совпадают синие кортежи r_b, r'_b и красные кортежи r_r, r'_r , из этого согласно теореме 2 следует, что $T_b \cong T'_b$ и $T_r \cong T'_r$. При этом для любой вершины $v_i \in X$ найдется подобная ей вершина $v'_j \in X'$, причем из леммы 5 следует, что кортежи этих вершин и кортежи их предков (вплоть до корня) совпадают как для синего, так и для красного деревьев. По лемме 6 $G \cong G'$.

Предположим теперь, что коды синих и/или красных корней не совпадают. Из этого следует, что синие и/или красные корневые кортежи не совпадают. Следовательно, красные и/или синие деревья будут неизоморфны (теорема 2). Очевидно, что если один из графов содержит подграф, который не будет изоморфен никакому подграфу второго графа, то такие графы не изоморфны. Т.о. графы G и G' в этом случае будут неизоморфны. \square

Зададим функцию

$$f(M) = K,$$

где $M = \{\Delta_1, \dots, \Delta_p\}$ – множество δ -графов Δ_i , $i = 1, \dots, p$;

K – мультимножество пар кортежей $(t_b(i), t_r(i))$, $|K| = |M| = p$;

$t_b(i)$ – кортеж синего корня Δ_i ;

$t_r(i)$ – кортеж красного корня Δ_i .

Лемма 9. Пусть в результате всех декомпозиций $D_\beta(t_i)$, $t_i \in V$, $i = 1, 2, \dots, n$ B -графа G_B получено множество δ -графов $M = \{\Delta_1, \Delta_2, \dots\}$, тогда $(\bigcup_{(j)} \Delta_j) \cap G_B = G_B$ и $f(M)$ однозначно задает G_B . \square

Доказательство. Для синих и красных ребер каждый δ -граф совпадает с γ -графом, из которого он был получен, поэтому справедливость доказываемого утверждения для этих ребер следует из леммы 7. Рассмотрим зеленые ребра. Пусть в результате $D_\beta(h)$ образовалось зеленое дерево с черными листьями $i, j \in V$ на уровне d и оранжевым корнем $k \in U$ на уровне $d + 1$. Т.о. ребра (i, k) и (k, j) не войдут в соответствующие уровню d δ -графы. Предположим, что эти ребра не войдут в δ -графы и для других стартовых вершин. Возьмем i в качестве стартовой вершины. Тогда в результате $D_\beta(i)$ вершина i окажется на уровне 0, вершина k окажется на уровне 1, а вершина j окажется на уровне 2, и ребро (i, k) будет раскрашено красным, а ребро (k, j) синим. Эти ребра войдут в соответствующий δ -граф. Следовательно, наше предположение неверно.

Пусть граф G_B изоморфен графу G'_B . Из леммы 4 следует, что при любой декомпозиции $D_\beta(t_i)$, $t_i \in V$, $i = 1, 2, \dots, n$ найдется такая декомпозиция $D_\beta(t')$, $t' \in V'$, что число уровней будет одинаковым, и β -деревья одного уровня будут изоморфны. Учитывая вывод 4 и лемму 8 видим, что $f(M) = f(M')$, где M' – множество всех δ -графов графа G'_B . \square

Удалив из δ -графа дополнительные корни, мы получим в общем случае несвязный граф, состоящий из синих и красных деревьев, который назовем σ -графом. При удалении дополнительных корней из нескольких δ -графов мы получаем один несвязный σ -граф. Удалив оранжевые корни зеленых деревьев из G_B , мы также получим σ -граф.

Лемма 10. Для изоморфизма σ -графов необходимо и достаточно, чтобы коды корней их синих и красных деревьев попарно совпадали. \square

Доказательство. Очевидно, что из σ -графа однозначно восстанавливается соответствующий ему δ -граф добавлением синего и/или красного корней/корня. При этом если коды корней синих и красных деревьев двух σ -графов попарно совпадают, то и коды корней соответствующих им δ -графов будут совпадать, из чего по лемме 8 следует изоморфизм этих δ -графов. Согласно выводу 1, если $G \cong G'$, вершины u, v' подобны, то $G - u \cong G' - v'$. Т.о. удаление синего и/или красного корней из изоморфных δ -графов приведет к изоморфным σ -графам. \square

Лемма 11. Для изоморфизма графов G_B и G'_B необходимо и достаточно, чтобы $S = S'$ (алгоритм 1). \square

Доказательство. Из каждого σ -графа однозначно восстанавливается соответствующий ему δ -граф добавлением синего и/или красного корней/корня. По лемме 9 совокупности δ -графов однозначно задают G_B и G'_B . Поэтому рассмотрим соответствующие σ -графы.

На i -ом шаге цикла 2 (алгоритм 1) для стартовой вершины i графа G_B получается σ -граф σ_i . Если $G \cong G'$, то аналогичный граф σ'_j получается на j -ом шаге цикла 2 для графа G'_B . По лемме 10 для изоморфизма σ_i и σ'_j необходимо и достаточно, чтобы коды корней этих графов попарно совпадали. Перебрав с помощью цикла 2 все возможные стартовые вершины $v_i \in V$, $i = 1, \dots, n$ для G_B (и для G'_B), мы получим все возможные σ -графы. Отсортировав коды вершин и записав полученные из них инварианты в векторы S и S' соответственно, мы получим, что если $S = S'$, то для любого графа σ_i найдется граф σ'_j с попарно совпадающими кодами корней, т.е. изоморфный. И наоборот, если $S \neq S'$, то для какого-то σ_i не найдется σ'_j с попарно совпадающими кодами корней.

□

Недостатком алгоритма 1 являются слишком длинные строки. Действительно, например, в случае регулярного графа со степенью вершины k и диаметром d длину l строки $c(v_j)$, представляющей код вершины на шаге 12, можно оценить как

$$l \approx (2ka)^d,$$

где a – длина строки, представляющей простой код вершины (т.к. речь идет о приближительной оценке, то выбираем вершину с максимальным a).

Т.е., если считать, что все коды имеют одинаковую длину a , то, пренебрегая терминальными символами, вклад которых в длину строки не столь существенен, получим, что каждая итерация цикла 7 увеличивает код ребра примерно в два раза и код вершины в k раз. Для преодоления этой проблемы используем простой трюк: добавим к исходному графу *общую вершину*, смежную со всеми прочими. Очевидно, что диаметр полученного графа не будет превосходить 2. По выводу 1, если такие графы изоморфны, то и графы с удаленной общей вершиной, т.е. исходные, также будут изоморфны. Т.о. общий алгоритм тестирования графов на изоморфизм будет выглядеть следующим образом (алгоритм 2).

Алгоритм 2 Тестирование графов G_s и G'_s на изоморфизм.

1. если число вершин графа G_s не равно числу вершин графа G'_s , то графы не изоморфны, выход;
 2. если число ребер графа G_s не равно числу ребер графа G'_s , то графы не изоморфны, выход;
 3. добавить общую вершину к графу G_s ;
 4. добавить общую вершину к графу G'_s ;
 5. с помощью процедуры P_C вычислить вектор S для графа G_s ;
 6. с помощью процедуры P_C вычислить вектор S' для графа G'_s ;
 7. если $S = S'$, то графы изоморфны, иначе графы не изоморфны.
-

Чтобы дать оценку вычислительной сложности алгоритма 2 для наихудшего случая, нужно рассмотреть процедуру P_C (алгоритм 1), как наиболее трудоемкую по времени. Согласно теореме 1, время на вычисление метки rgb (шаг 5) линейно зависит от n . Такая же зависимость будет и для шага 6. Наиболее трудоемкими являются операторы 9,10, которые повторяются циклами 2 и 7 не более чем $2n$ раз (т.к. $\text{dm}(G_s) \leq 2$). При этом оператор 9 является циклом, повторяемым m раз, а оператор 10 является циклом, повторяемым n раз. Однако в операторе 9 производится сравнение кодов двух вершин, в то время как в операторе 10 сортируются до n кодов вершин. В литературе приводится много эффективных методов сортировки, требующих не более чем n^2 сравнений сортируемых элементов. Т.о. оператором 9 производится $2m$ сравнений, а оператором 10 не более чем n^3 . Учитывая, что наибольшее количество ребер в графе - это число ребер полного графа $(n^2 - n)/2$, видим, что наиболее трудоемким по времени является оператор 10. С учетом циклов 2 и 7 получаем, что общее число сравнений не будет превосходить $2n^5$. Умножив эту величину на длину строки l , получим общее число посимвольных сравнений p (замечание 1):

$$p = 2n^5 l \approx 2n^5 (2ka)^2.$$

Полагая $k \leq n$ и $a < cn$, где c - константа, равная числу бит, необходимого для обозначения одного символа строки. Откуда:

$$p < 8cn^9.$$

Операторы 14,15,16 потребуют явно меньшего числа сравнений. Т.о. пренебрегая младшими членами и коэффициентами, получаем оценку для наихудшего случая $O(n^9)$.

Заключение

По сути в данной работе предложен метод сведения общей задачи тестирования графов на изоморфизм к ранее решенной более частной задаче тестирования на изоморфизм деревьев с помеченными вершинами. Используемые при этом определения, алгоритмы и доказательства, возможно, выглядят несколько избыточными, а предлагаемые структуры данных слишком объемными. Однако целью нашей работы является прежде всего теоретический результат, для которого избыточность лучше, чем недостаточность. Предложенный алгоритм отвечает (да/нет) на вопрос изоморфны ли графы, но в случае утвердительного ответа не дает на выходе какого-либо возможного изоморфизма. Алгоритм был реализован в виде программы в среде программирования Delphi-7 для ОС MS Windows-XP. Zip-архив с исходными кодами и исполняемым файлом можно выгрузить по адресу:

<http://mt2.comtv.ru/>

пароль для разархивации:

hN758-kT402-N3D8a-961fQ-WJL24

По этому же адресу находится перевод данной статьи на русский.

Некоторые свойства B -графов, не понадобившиеся для приведенного доказательства, могут оказаться полезными для развития данного подхода. Поэтому они приведены в приложении 1.

Благодарности

Автор благодарен Г.М.Хитрову (СПб университет) за обсуждение этой статьи и М.П.Трофимовой за помощь в подготовке рукописи.

Приложение 1

Матрица смежности графа G_B имеет вид:

$$A = \begin{pmatrix} O & B \\ B^T & O \end{pmatrix},$$

где O - матрица, состоящая из нулей;

B - $(0,1)$ -матрица размера $m \times n$:

$$B = \|b_{ij}\|,$$

где b_{11} является смежностью первой вершины из U (т.е. дополнительной вершины 1) и первой вершины из V (т.е. вершины с номером $m + 1$) и т.д.

Исключим из последующего обсуждения тривиальные случаи, для которых $n \leq 3$ и будем рассматривать только связные графы.

Предложение 1. Если менять местами строки (столбцы) матрицы B , то смежность вершин не нарушается. \square

Доказательство. Поменяем местами строки i, j . Это приведет к тому, что $b_{ik} = b'_{jk}$ и $b_{jk} = b'_{ik}$, где b'_{jk}, b'_{ik} обозначают элементы матрицы до перемены мест, а b_{jk}, b_{ik} после. Это означает, что если до перестановки у графа было ребро $(i, k), i \in U, k \in V$, то после перестановки это ребро стало обозначаться $(j, k), j \in U$, а если до перестановки у графа было ребро $(j, k), j \in U, k \in V$, то после перестановки это ребро стало обозначаться $(i, k), i \in U$. То же будет наблюдаться для всех ребер, инцидентных вершинам i, j , соответственно. Т.о. в результате вершины i, j просто обменялись номерами и смежность не нарушилась.

Аналогично поменяем местами столбцы i, j . Это приведет к тому, что $b_{ki} = b'_{kj}$ и $b_{kj} = b'_{ki}$. Это означает, что если до перестановки у графа было ребро $(i, k), i \in V, k \in U$, то после перестановки это ребро стало обозначаться $(j, k), j \in V$, а если до перестановки у графа было ребро $(j, k), j \in V, k \in U$, то после перестановки это ребро стало обозначаться $(i, k), i \in V$. То же будет наблюдаться для всех ребер, инцидентных вершинам i, j , соответственно. Т.о. в результате вершины i, j просто обменялись номерами и смежность не нарушилась. \square

Предложение 2. *Никакие перестановки строк и столбцов матрицы B не могут привести к блоку:*

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \quad (1)$$

\square

Доказательство. Допустим, что такой блок возможен для вершин $i, j \in V, p, q \in U$. Это означает, что в исходном графе имеется два ребра (i, j) , что невозможно по условиям задачи (мультиграфы нами не рассматриваются). Полученное противоречие показывает, что наше предположение неверно. \square

Предложение 3. *В матрице B не может быть равных строк (столбцов).* \square

Доказательство. Предположим, что две строки матрицы B равны. Каждая дополнительная вершина имеет степень 2. Значит, строка состоит из двух единиц и остальные нули. Если строки равны, то перестановкой строк и столбцов можно получить блок (1), что противоречит предложению 2. Значит, предположение о равенстве строк неверно.

Проведем теперь доказательство для столбцов. Рассмотрим следующие случаи.

1) Два равных столбца состоят только из нулей. Этого быть не может, так как по условию задачи мы рассматриваем только связные графы.

2) Столбцы i, j равны и имеют каждый по одной единице, а остальные нули. В этом случае существует строка k такая, что $b_{ki} = b_{kj} = 1$. Т.к. мы не рассматриваем тривиальные случаи ($n \leq 3$), то это означает, что исходный граф содержит подграф, состоящий из одного ребра (i, j) и вершин i, j , не

связанных больше ни с какими другими вершинами. Но это противоречит условию задачи, по которому мы рассматриваем только связные графы.

3) Столбцы равны и имеют каждый не менее двух единиц. В этом случае перестановкой строк и столбцов можно получить блок (1), что противоречит предложению 2. Значит, наше предположение неверно. \square

Из предложения 1 следует предположение, что если сортировать матрицу B по строкам, по столбцам, потом снова по строкам и по столбцам и так до тех пор, пока матрица не перестанет изменяться, то получится "максимальная матрица", не зависящая от нумерации вершин. К сожалению, несложные контр-примеры показывают, что такое предположение неверно.

Список литературы

- [1] Дж. Ульман А. Ахо, Дж. Хопкрофт. *Построение и анализ вычислительных алгоритмов*. М.: Мир, 1979, С.102.
- [2] Н. Вирт. *Алгоритмы + Структуры данных = программы*. М.:Мир, 1985, С.27.
- [3] П.Е. Тарьян Дж.Е. Хопкрофт. Изоморфизм планарных графов. *Кибернетический сборник, Новая серия, Под ред. О.Б.Лупанова*, 12:39, М.: Мир, 1975.
- [4] G. Chartrand. *Introductory Graph Theory*. 1977.
- [5] А.А. Зыков. *Основы теории графов*. М.: Вузовская книга, 2004, С. 163.
- [6] Д.Джонсон М.Гери. *Вычислительные машины и труднорешаемые задачи*. М.: Мир, 1982, С.194.
- [7] J. Köbler, U. Schöning, and J. Torán. *The graph isomorphism problem - its structural complexity. Prog Theoret Comput Sci*. Birkhäuser, Boston, 1993.
- [8] G. L. Miller. Graph isomorphism, general remarks. *J. Comput. Syst. Sci.*, 18:128–142, 1979.
- [9] Ф. Харари. *Теория графов*. М.: УРСС, 2003.