

УДК 519.171 + 519.175.1 + 519.17:5

ОБ АЛЬТЕРНАТИВНОМ СПОСОБЕ ЗАДАНИЯ КОНЕЧНЫХ ГРАФОВ

М. Н. Назаров

Национальный исследовательский университет «МИЭТ», г. Москва, Россия

Рассматривается линейная нотация — полный инвариант графов, который позиционируется как альтернатива для описания конечных графов. Данный инвариант строится с помощью алгоритма, близкого к алгоритму поиска канонических форм графов. Хранение линейной нотации в памяти вместо обычного графа позволяет проще решать две основные задачи: построение иллюстраций для графов и сравнение графов на изоморфизм. Для полученного описания дополнительно демонстрируется переносимость таких понятий теории графов, как раскраски и пути, непосредственно на линейные нотации.

Ключевые слова: *изоморфизм графов, классы автоморфизма вершин и рёбер, инварианты графов.*

DOI 10.17223/20710410/29/7

AN ALTERNATIVE WAY OF DEFINING FINITE GRAPHS

M. N. Nazarov

*National Research University of Electronic Technology, Moscow, Russia***E-mail:** Nazarov-Maximilian@yandex.ru

In this paper, we consider the graph linear notation — a complete graph invariant, which is positioned as an alternative to description of finite graphs. This invariant is constructed using an algorithm which is close to the search algorithm for canonical forms of graphs. The storage in a memory of a graph linear notation instead of the graph itself simplifies the procedures for constructing graph illustrations and testing two graphs for isomorphism. We demonstrate how the main graph theory concepts including colouring and graph paths can be defined in terms of graph linear notations.

Keywords: *graph isomorphism, automorphism classes of vertices and edges, graph invariants.*

Введение

В работе [1] введена линейная нотация абстрактного графа $I[G]$. Для нотации $I[G]$ продемонстрировано, что она является полным инвариантом и по своим свойствам во многом аналогична обычным графам G . В частности, для $I[G]$ введены понятия абстрактных вершин и абстрактных рёбер, а также подграфы, раскраски, пути и некоторые элементарные операции на графах. Отдельно следует отметить, что алгоритм перехода от $I[G]$ к обычным графам G является полиномиальным. Поскольку $I[G]$ является полным инвариантом, данный факт служит основанием для использования нотации $I[G]$ в качестве альтернативного способа хранения графов.

Реализована линейная нотация $I[G]$ с помощью номеров $I(\bar{v})$ и $I(\bar{u}, \bar{v})$ классов автоморфизма вершин и рёбер графов. Для данных номеров доказано, что они являются однозначными идентификаторами для классов автоморфизма вершин и рёбер и позволяют ввести на множестве этих классов линейные порядки: $\bar{v}_1 \leq \bar{v}_2 \Leftrightarrow I(\bar{v}_1) \leq I(\bar{v}_2)$ и $(\bar{v}_1, \bar{u}_1) \leq (\bar{v}_2, \bar{u}_2) \Leftrightarrow I(\bar{v}_1, \bar{u}_1) \leq I(\bar{v}_2, \bar{u}_2)$. Это означает, что можно пронумеровать классы вершин числами $1, \dots, m$, а классы рёбер — числами $1, \dots, k$, если у графа m классов автоморфных вершин и k классов автоморфных рёбер. Нужно отметить, что в работе [1] была допущена неточность: во всех иллюстрациях и примерах вершины и рёбра помечены именно такими порядковыми индексами, а не $I(\bar{v})$ и $I(\bar{u}, \bar{v})$, как утверждается в подписях к ним. В данной работе мы в первую очередь проведём разграничение между индексами и номерами классов автоморфизма вершин и рёбер и переформулируем все основные результаты из [1] в приложении к индексам классов автоморфизма.

Для демонстрации преимуществ линейной нотации $I[G]$ по отношению к классическому описанию графов дополнительно рассмотрим примеры задания стандартных графов, а также приведём алгоритмы для построения путей и реализации раскрасок на $I[G]$.

1. Построение линейных нотаций для классов изоморфных графов

Мы исключаем из рассмотрения ориентированные графы, графы с петлями, кратными рёбрами и бесконечные графы [2]. Отметим, что построить обобщение нашей методики будет относительно просто только для классов ориентированных графов и графов с петлями.

Определение 1. Графом будем называть пару $G = (V, E)$, где множество вершин V — это любое конечное множество, а множество рёбер $E \subseteq V \times V$ — это бинарное отношение на V , для которого выполняются следующие условия:

- 1) $\forall a, b \in V \quad ((a, b) \in E \Rightarrow (b, a) \in E)$ — отношение симметрично;
- 2) $\forall a \in V \quad ((a, a) \notin E)$ — отношение антирефлексивно.

Определение 2. Пусть дан произвольный граф $G = (V, E)$, у которого $|V| = n$. Тогда, если выбрать некоторый порядок $\alpha = (v_1, \dots, v_n)$ на множестве вершин V , то можно поставить в соответствие графу G матрицу смежности A по следующему правилу:

$$A(i, j) = 1 \Leftrightarrow (v_i, v_j) \in E \quad \& \quad A(i, j) = 0 \Leftrightarrow (v_i, v_j) \notin E.$$

На рис. 1 представлен пример двух матриц смежности, которые построены для разных порядков α_1 и α_2 на множестве вершин V графа G .

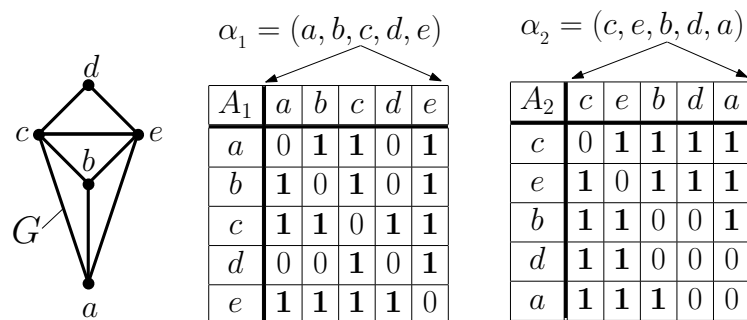


Рис. 1. Пример двух разных матриц смежности A_1 и A_2 для одного графа

Определение 3. Кодом матрицы смежности A конечного графа G будем называть число $\mu(A) = A(n, n) + A(n, n-1) \cdot 2 + \dots + A(n, 1) \cdot 2^{n-1} + A(n-1, n) \cdot 2^n + \dots + A(1, 1) \cdot 2^{n^2-1}$, где $A(i, j)$ — элементы матрицы смежности A .

Например, для двух матриц, изображённых на рис. 1, получим следующие коды: $\mu(A_1) = 110110101110110010111110$ и $\mu(A_2) = 111110111110011100011100$.

Определение 4. Наибольший из всех возможных кодов $\mu(A)$ матриц смежности A графа G назовём *макси-кодом* графа $\mu_{\max}(G)$. Если для некоторого порядка вершин α код матрицы смежности $\mu(A) = \mu_{\max}(G)$, то говорим, что α *соответствует* макси-коду.

В [1] доказано следующее утверждение о макси-коде $\mu_{\max}(G)$.

Утверждение 1 [1]. Если два порядка вершин $\alpha_1 = (v_1^1, \dots, v_n^1)$ и $\alpha_2 = (v_1^2, \dots, v_n^2)$ соответствуют макси-коду $\mu_{\max}(G)$ графа G , то вершины в этих порядках попарно автоморфны: $v_i^1 \sim v_i^2$ для всех $i = 1, \dots, n$.

Элементы последовательностей вершин $\alpha = (v_1, \dots, v_n)$ будем обозначать $v_i = \alpha(i)$.

Определение 5. Пусть макси-коду графа G соответствует некоторый порядок следования вершин α . Назовём *номером класса автоморфизма вершин* $\bar{v}$ натуральное число $N(\bar{v})$, которое равно первому вхождению в порядок α вершины из класса $\bar{v}$: $N(\bar{v}) = \min_{i: \alpha(i) \in \bar{v}} i$.

Определение 6. Пусть макси-коду графа G соответствует некоторый порядок следования вершин α . Назовём *номером класса автоморфизма рёбер* $(\bar{x}, \bar{y})$ натуральное число $N(\bar{x}, \bar{y})$, равное номеру первого вхождения ребра из класса $(\bar{x}, \bar{y})$ в матрицу смежности A графа G , построенную в соответствии с α (при нумерации элементов матрицы по строкам): $N(\bar{x}, \bar{y}) = \min_{\substack{A(i,j)=1 \\ (\alpha(i), \alpha(j)) \sim (x, y)}} j + (i-1) \cdot n$.

Утверждение 2. Номера $N(\bar{v})$ естественным образом порождают линейный порядок на классах автоморфных вершин: $\bar{v}_1 \leq \bar{v}_2 \Leftrightarrow N(\bar{v}_1) \leq N(\bar{v}_2)$.

Доказательство. По определению номера $N(\bar{v})$ являются натуральными числами; согласно утверждению 1, номера совпадают только у автоморфных вершин. Получаем, что множество номеров $N(\bar{v}_1), \dots, N(\bar{v}_k)$ линейно упорядочено и, следовательно, порядок $\bar{v}_1 \leq \bar{v}_2 \Leftrightarrow N(\bar{v}_1) \leq N(\bar{v}_2)$ линейный. ■

Утверждение 3. Номера $N(\bar{v}, \bar{u})$ естественным образом порождают линейный порядок на классах автоморфных ребер: $(\bar{v}_1, \bar{u}_1) \leq (\bar{v}_2, \bar{u}_2) \Leftrightarrow N(\bar{v}_1, \bar{u}_1) \leq N(\bar{v}_2, \bar{u}_2)$.

Доказательство. Доказательство аналогично доказательству утверждения 2. Если два порядка вершин α_1 и α_2 соответствуют макси-коду графа $\mu_{\max}(G)$, то матрицы смежности для этих порядков совпадают. В результате рёбра, которые соответствуют одинаковым позициям в этих матрицах, автоморфны. ■

Определение 7. Введём для графа G *индексы классов автоморфизма вершин* $I(\bar{v})$ через номера классов $N(\bar{v})$ с помощью следующих индуктивных правил:

- 1) $N(\bar{v}) = \min_{\bar{u}} N(\bar{u}) \Rightarrow I(\bar{v}) = 1$ — базис индукции для определения первого индекса;
- 2) $\left. \begin{aligned} & (N(\bar{v}) > N(\bar{u})) \& \\ & \& \forall u^* \neq u (N(\bar{v}) > N(\bar{u}^*) \Rightarrow N(\bar{u}) > N(\bar{u}^*)) \end{aligned} \right\} \Rightarrow I(\bar{v}) = I(\bar{u}) + 1$ — переход.

Определение 8. Аналогично определению для вершин введём понятие *индекса классов автоморфизма рёбер* $I(\bar{x}, \bar{y})$, заменив классы автоморфизма вершин на классы автоморфизма рёбер.

Теорема 1. Если $G \cong H$, то для $u \in V(G)$, $v \in V(H)$ равенство $I(\bar{u}) = I(\bar{v})$ выполняется тогда и только тогда, когда существует изоморфизм $\phi : V(G) \rightarrow V(H)$, такой, что $\phi(u) = v$.

Доказательство. Докажем сначала, что из $\phi(u) = v$, где ϕ — изоморфизм, следует совпадение индексов классов автоморфизма $I(\bar{u}) = I(\bar{v})$.

Если два графа изоморфны, то у них один и тот же макси-код $\mu_{\max}(G) = \mu_{\max}(H)$. В этом случае изоморфизм можно представить в виде $\phi = \begin{pmatrix} u_1 \dots u_n \\ v_1 \dots v_n \end{pmatrix}$, где $\alpha_1 = (u_1, \dots, u_n)$ — некоторый порядок, который соответствует макси-коду в G , а $\alpha_2 = (v_1, \dots, v_n)$ — порядок, соответствующий макси-коду в H . Поскольку номер $N(\bar{u})$ — натуральное число, то $\alpha_1(N(\bar{u}))$ задаёт некоторую вершину $u^* = \alpha_1(N(\bar{u}))$. Эта вершина u^* обладает двумя свойствами: $u \sim u^*$ и среди всех симметричных по отношению к u вершинам u^* имеет наименьший индекс в α_1 . Если допустим, что $N(\bar{u}) \neq N(\bar{v})$, то получим $\phi(u^*) \approx \phi(u)$. В результате получаем противоречие: $\phi(u^*) \approx \phi(u)$ и $u \sim u^*$. Таким образом, у всех изоморфных вершин ($\phi(u) = v$) номера классов автоморфизма обязательно совпадают ($N(\bar{u}) = N(\bar{v})$). Пользуясь утверждением 2 и учитывая определение для индексов $I(\bar{v})$, получаем, что $I(\bar{u}) = I(\bar{v})$.

Пусть $G \cong H$ и совпали индексы $I(\bar{u}) = I(\bar{v})$. Поскольку $G \cong H$, то существует некоторый изоморфизм $\phi_0 : V(G) \rightarrow V(H)$. Для образа $v^* = \phi_0(u)$ по первой части данного утверждения получим $I(\bar{v}^*) = I(\bar{u}) = I(\bar{v})$. Из этого можно заключить, что вершины v, v^* автоморфны. Обозначим автоморфизм, переводящий эти вершины одна в другую, через ψ : $\psi(v^*) = v$. Тогда искомый изоморфизм ϕ можно определить в виде композиции двух изоморфизмов $\phi = \phi_0 \circ \psi$. ■

Теорема 2. Если $G \cong H$, то для двух рёбер $(u_1, u_2) \in E(G)$ и $(v_1, v_2) \in E(H)$ равенство $I(\overline{u_1, u_2}) = I(\overline{v_1, v_2})$ выполняется тогда и только тогда, когда существует изоморфизм $\phi : V(G) \rightarrow V(H)$, такой, что $\phi(u_1) = v_1$ и $\phi(u_2) = v_2$, или $\phi(u_1) = v_2$ и $\phi(u_2) = v_1$.

Доказательство. Доказательство достаточности аналогично доказательству для вершин. Если допустить, что существует изоморфизм $\phi : V(G) \rightarrow V(H)$, такой, что $\phi(u_1) = v_1$ и $\phi(u_2) = v_2$, то совпадут номера $N(\overline{u_1, u_2}) = N(\overline{v_1, v_2})$. Из совпадения номеров для всех классов рёбер следует $I(\overline{u_1, u_2}) = I(\overline{v_1, v_2})$.

Необходимость также можно доказать по аналогии с теоремой 1. Пусть $G \cong H$ и $I(\overline{u_1, u_2}) = I(\overline{v_1, v_2})$. Для образов изоморфизма $v_1^* = \phi_0(u_1)$ и $v_2^* = \phi_0(u_2)$ по первой части данного утверждения получим $I(\overline{u_1, u_2}) = I(\overline{v_1, v_2}) = I(\overline{v_1^*, v_2^*})$. Из этого можно заключить, что рёбра автоморфны: $(v_1, v_2) \sim (v_1^*, v_2^*)$. Обозначим через ψ следующий автоморфизм: $\forall i (\psi(v_i^*) = v_i)$. Искомый изоморфизм — композиция $\phi = \phi_0 \circ \psi$. ■

Непосредственным следствием из теорем 1 и 2 является тот факт, что индексы $I(\bar{v})$ и $I(\overline{v_1, v_2})$ являются уникальными идентификаторами для классов автоморфизма. Таким образом, эти индексы можно использовать вместо самих вершин и рёбер при определении инварианта для класса изоморфных графов $[G]$. В частности, можно сначала представить граф G в виде строки символов с помощью алгоритма, похожего на алгоритм построения линейной нотации для молекулярных графов SMILES [3], а затем в данной строке заменить названия вершин v на их индексы $I(\bar{v})$ и получить инвариант для графа G .

Определение 9. Симметрическая линейная нотация $\mathcal{L}(G)$ для связного графа G — это строка символов, которая определяется на основе следующего алгоритма.

Шаг 0. Введём множества исключённых вершин V_E и временно заменённых вершин V_R . На начало работы алгоритма оба эти множества полагаются пустыми: $V_R = V_E = \emptyset$. Каждой новой вершине v , помещённой в множество заменённых V_R , будем присваивать специальный код $\#m$, где m — число элементов $|V_R|$ после добавления вершины v . Признаком окончания работы алгоритма является помещение всех вершин $V(G)$ графа G в множество исключённых вершин V_E .

Шаг 1. Выбираем любую вершину v_1 первого индекса ($I(\bar{v}_1) = 1$) и записываем её в нотацию $\mathfrak{L}(G)$, добавляя к ней открывающую скобку первого номера: $\mathfrak{L}(G) = v_1 \left[\dots \right]_1$. При этом помещаем v_1 в множество V_R и присваиваем ей первый специальный код $\#1$.

Шаг 2 (рекурсивный). Пусть линейная нотация доведена до некоторой вершины v_k . При этом часть из рассмотренных вершин $v_1, \dots, v_k$ помещена в множество исключённых V_E , а другая часть — в множество заменённых V_R . Вместо заменённых вершин можем использовать коды $\#1, \dots, \#m$ (код $\#m$ соответствует v_k). Среди всех вершин из окружения v_k , которые не были удалены или заменены на предыдущих этапах, выбираем любую v_{k+1} , которая удовлетворяет следующим условиям:

- 1) наибольший приоритет у вершины с минимальным индексом $I(\bar{v}_{k+1})$;
- 2) если у двух вершин индексы совпали ($I(\bar{v}_{k+1}) = I(\bar{v}_{k+1}^*)$), то предпочтение отдаётся вершине v_{k+1} , у которой меньше индекс ребра: $I(\bar{v}_k, v_{k+1}) < I(\bar{v}_k, v_{k+1}^*)$;
- 3) при совпадении индексов вершин и рёбер у v_{k+1} и v_{k+1}^* предпочтение отдаётся вершине, которая в графе G ближе к вершине кода $\#1$ (то есть может быть соединена с ней путём из меньшего числа вершин). Если эти расстояния совпадают для вершин v_{k+1} и v_{k+1}^* , то вершине, которая ближе к вершине кода $\#2$, и так далее до $\#(m-1)$;
- 4) если вершины v_{k+1} и v_{k+1}^* из окружения v_k не удалось различить с помощью условий 1–3 по индексам вершин, рёбер и расстояниям до вершин кодов $\#1, \dots, \#m$, то выбирается любая из них.

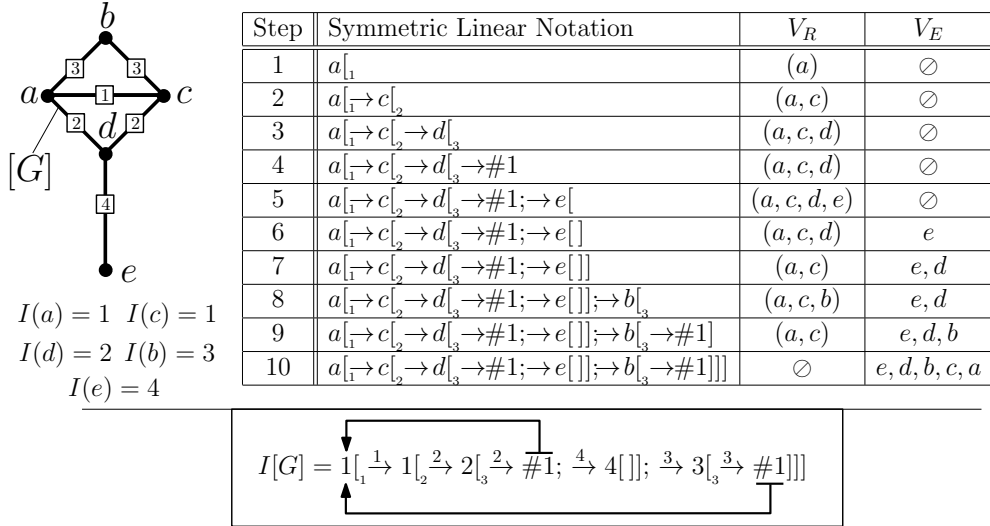
Если с помощью условий 1–4 удалось найти вершину v_{k+1} , то добавляем её в линейную нотацию $\mathfrak{L}(G) = \dots \xrightarrow{i_k} v_k \left[\dots ; \xrightarrow{i_{k+1}} v_{k+1} \left[\dots \right]_{m+1} \dots \right]_m$ и в множество V_R . После этого переходим к рекурсивному шагу 3 для v_{k+1} . Если вершину v_{k+1} не удалось найти, то удаляем v_k из множества V_R и переводим её в V_E , полностью исключая из рассмотрения. Затем закрываем скобку уровня m в линейной нотации $\mathfrak{L}(G)$ и повторяем шаг 2 для той вершины v_j , которая на текущий момент соответствует коду $\#(m-1)$.

Шаг 3 (рекурсивный). Пусть линейная нотация доведена до некоторой вершины v_k . В первую очередь проверяем, нет ли в графе ребра, которое соединяет вершину v_k и вершину кода $\#1$. Если такое ребро имеется и оно до сих пор не было учтено в нотации, то добавляем запись: $\mathfrak{L}(G) = \dots \xrightarrow{i_k} v_k \left[\dots \xrightarrow{i} \#1; \dots \right]_m$, где $i = I(\bar{v}, \bar{v}_k)$. Повторяем процесс для кода $\#2$ и так далее до последнего кода $\#(m-1)$. По завершении переходим к рекурсивному шагу 2 для вершины v_k .

Определение 10. *Линейной нотацией класса изоморфных графов $[G]$ назовём строку $I[G]$, которая получается из любой произвольной симметрической линейной нотации $\mathfrak{L}(G)$ путём замены всех вершин v на индексы классов вершин $I(\bar{v})$.*

Пример 1. Рассмотрим работу алгоритма для построения $\mathfrak{L}(G)$ и перехода к линейной нотации $I[G]$ с пошаговым выводом множеств V_R и V_E (рис. 2).

Согласно определению 9, одному графу G может соответствовать несколько различных симметрических нотаций $\mathfrak{L}(G)$. Для этого достаточно, чтобы у графа G хотя

Рис. 2. Пример построения линейной нотации $I[G]$

бы две вершины были автоморфными. Например, если в графе G есть две вершины первого индекса, то нотацию $\mathfrak{L}(G)$ можно начать с любой из них. Таким образом, необходимо доказать, что линейная нотация $I[G]$ определена корректно и не зависит от выбора $\mathfrak{L}(G)$.

Утверждение 4. Линейная нотация $I[G]$ определяется однозначным образом для любого конечного связного графа G .

Доказательство. Проведём доказательство однозначности $I[G]$ по индукции.

Б а з и с и н д у к ц и и. Рассмотрим шаг 1 алгоритма определения $\mathfrak{L}(G)$. После замены вершины v_1 на её индекс $I(\overline{v_1}) = 1$ имеем одну и ту же начальную строчку $I[G] = 1[1 \dots$ для любых $\mathfrak{L}(G)$.

И н д у к т и в н ы й п е р е х о д. Допустим, что неоднозначность в определении $I[G]$ не возникла до начала отдельной итерации рекурсивного шага 2 алгоритма определения $\mathfrak{L}(G)$. После замены вершин v_k и v_{k+1} на их индексы $I(\overline{v_k})$ и $I(\overline{v_{k+1}})$ получим одну и ту же строчку $I[G] = \dots \xrightarrow{i_k} I(\overline{v_k})[\dots ; \xrightarrow{i_{k+1}} I(\overline{v_{k+1}})[\dots$ для любых $\mathfrak{L}(G)$.

Допустим, что неоднозначность в определении $I[G]$ не возникла до начала отдельной итерации рекурсивного шага 3 алгоритма определения $\mathfrak{L}(G)$. Если предположим, что коды $\#j$, с которыми связана v_k , различаются для двух разных $\mathfrak{L}(G)$, то получим противоречие с условием 3 выбора новых вершин на рекурсивном шаге 2. Таким образом, переход к $I[G] = \dots \xrightarrow{i_k} I(\overline{v_k})[\dots \xrightarrow{i} \#1; \dots$ на шаге 3 также однозначный. ■

Замечание 1. Отметим, что для линейной нотации $I[G]$ можно ввести понятия абстрактных вершин и абстрактных рёбер, связав их с соответствующими индексами автоморфизма. При этом у $I[G]$ может быть несколько экземпляров абстрактных вершин, которые соответствуют одному и тому же индексу j , и то же самое для абстрактных рёбер.

Определение 11. Раскраской для линейной нотации $I[G]$ отдельной абстрактной вершины j в цвет α будем называть такую строку $I_{\langle j, \alpha \rangle}^*[G]$, которая получается из линейной нотации $I[G]$ путём замены первого вхождения j на пару $\langle j, \alpha \rangle$. Раскраску для большего количества вершин и большего количества цветов определим индуктивно как последовательную раскраску графа по одной отдельной абстрактной вершине.

При этом достаточно потребовать, чтобы на каждом шаге раскраски выбирались только те вершины, которые не были раскрашены на предыдущих.

Теорема 3. Линейная нотация класса изоморфных графов $I[G]$ является полным инвариантом для любого конечного связного графа G .

Доказательство. Докажем необходимость, то есть что из изоморфизма графов $G_1 \cong G_2$ следует, что $I[G_1] = I[G_2]$. Из теорем 1 и 2 получаем, что у графов G_1 и G_2 один и тот же набор индексов для вершин и рёбер. Как следствие, с учётом утверждения 4, нотации $\mathfrak{L}(G_1)$ и $\mathfrak{L}(G_2)$ отличаются только названиями вершин. Таким образом, $I[G_1] = I[G_2]$.

Докажем достаточность, то есть что из совпадения линейных нотаций $I[G_1] = I[G_2]$ следует изоморфизм графов $G_1 \cong G_2$. Для этого с помощью раскраски названиями V_1 вершин $G_1 = (V_1, E_1)$ перейдём от $I[G_1]$ к $\mathfrak{L}(G_1)$. При этом остановимся на том варианте раскраски $\mathfrak{L}(G_1)$, который однозначно соответствует графу G_1 по связям E_1 между вершинами, потенциально перебрав все $|V_1|!$ вариантов. Затем повторим аналогичную процедуру для G_2 и получим $\mathfrak{L}(G_2)$. Построим изоморфизм в явном виде, отображив все подряд идущие вершины из $\mathfrak{L}(G_1)$ в соответствующие им по порядку вершины из $\mathfrak{L}(G_2)$. Если допустить для данного отображения, что условие изоморфизма не выполняется, то немедленно получим противоречие. Действительно, нарушение условия изоморфизма означает, что коды $\#1, \dots, \#m$ стоят на разных позициях в $\mathfrak{L}(G_1)$ и $\mathfrak{L}(G_2)$, что невозможно ввиду тождества $I[G_1] = I[G_2]$ и однозначного определения $I[G]$ на основе $\mathfrak{L}(G)$. ■

Для демонстрации изложенных результатов на рис. 3 приводятся примеры $I[G]$ для трёх графов G_1 , G_2 и G_3 , чьи вершины помечены индексами $I(\bar{v})$, а рёбра — $I(\bar{u}, \bar{v})$.

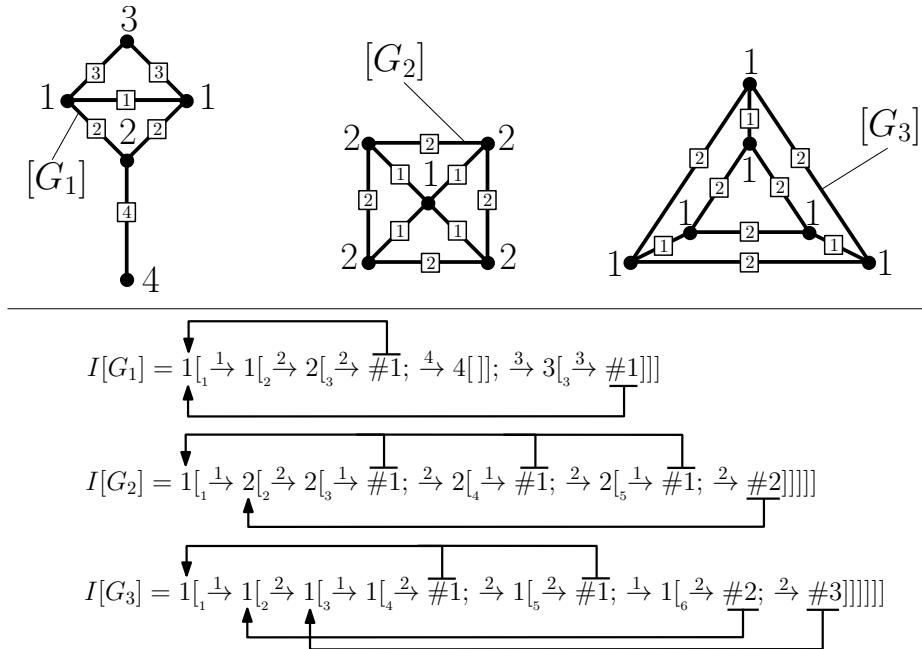


Рис. 3. Примеры линейных нотаций $I[G]$ для классов изоморфных графов $[G_1]$, $[G_2]$ и $[G_3]$

2. Практическое приложение линейных нотаций

Классический способ применения инвариантов графов сводится к вычислению и сравнению инвариантов каждый раз при проверке двух графов на изоморфизм.

Достаточно очевидно, что подобная схема совершенно неприменима для полных инвариантов, так как задача их построения относится к классу NP. В частности, для нотаций $I[G]$ имеем следующие оценки (параметр $n = |V(G)|$):

- 1) переход от G к $I[G]$ — задача эквивалентна задаче об изоморфизме графов;
- 2) проверка на совпадение для $I[G]$ — задача сложности $O(n^2)$;
- 3) переход от $I[G]$ к G на основе раскраски — задача сложности $O(n^2)$.

Общая идея заключается в том, чтобы *хранить в памяти компьютера* $I[G]$ вместо обычного графа G . При таком подходе упрощается проверка на изоморфизм, поскольку не требуется предварительно вычислять инвариант $I[G]$, а процедура сводится к полиномиальной задаче проверки на совпадение $I[G_1] = I[G_2]$. Для графов с небольшим числом вершин можно использовать базовый алгоритм перехода от G к $I[G]$, несмотря на его принадлежность к классу NP, поскольку потребуется применить его один раз перед сохранением линейной нотации $I[G]$. Однако для графов с большим числом вершин такой подход непрактичен. Фактически, построение линейных нотаций $I[G]$ для больших графов требует определения стандартных классов графов непосредственно на основе $I[G]$, а также введения некоторого набора операций полиномиальной сложности на $I[G]$ для получения более сложных классов изоморфных графов.

Рассмотрим, как можно с помощью линейной нотации $I[G]$ описать в общем виде следующие стандартные классы графов: пути P_n , циклы C_n , полные графы K_n и полные двудольные графы $K_{n,m}$ [2, 4].

Пример 2. Для путей P_n рассмотрим два случая: когда n чётное ($n = 2k$) и нечётное ($n = 2k - 1$):

$$\begin{aligned} I[P_{2k}] &= 1 \left[\begin{array}{c} \xrightarrow{1} \\ 1 \end{array} \xrightarrow{2} 1 \left[\begin{array}{c} \xrightarrow{2} \\ 2 \end{array} \xrightarrow{3} \dots \xrightarrow{n} n \left[\begin{array}{c} \xrightarrow{n} \\ n \end{array} \right] \dots \right] \dots \right]; \\ I[P_{2k-1}] &= 1 \left[\begin{array}{c} \xrightarrow{1} \\ 1 \end{array} \xrightarrow{2} 2 \left[\begin{array}{c} \xrightarrow{2} \\ 2 \end{array} \xrightarrow{3} \dots \xrightarrow{n-1} n \left[\begin{array}{c} \xrightarrow{n-1} \\ n \end{array} \right] \dots \right] \dots \right]; \end{aligned}$$

Пример 3. Линейная нотация для циклов C_n задана однозначно:

$$I[C_n] = 1 \left[\begin{array}{c} \xrightarrow{1} \\ 1 \end{array} \xrightarrow{2} \dots \xrightarrow{n} 1 \left[\begin{array}{c} \xrightarrow{1} \\ n \end{array} \xrightarrow{\#1} \dots \right] \dots \right].$$

Пример 4. Линейная нотация для полного графа K_n также задана однозначно:

$$I[K_n] = 1 \left[\begin{array}{c} \xrightarrow{1} \\ 1 \end{array} \xrightarrow{2} 1 \left[\begin{array}{c} \xrightarrow{1} \\ 2 \end{array} \xrightarrow{3} \dots \xrightarrow{n} 1 \left[\begin{array}{c} \xrightarrow{1} \\ n \end{array} \xrightarrow{\#1} \dots \right] \dots \right] \dots \right].$$

Пример 5. Для двудольных графов $K_{n,m}$ рассмотрим два случая: $n < m$ и $n = m$:

$$\begin{aligned} I[K_{n,m}] &= 1 \left[\dots \xrightarrow{1} 1 \left[\begin{array}{c} \xrightarrow{1} \\ 2m-1 \end{array} \xrightarrow{\#2} \dots \xrightarrow{\#2m-4} 2 \left[\begin{array}{c} \xrightarrow{1} \\ 2m \end{array} \xrightarrow{\#1} \dots \xrightarrow{\#2m-3} \dots \right] \dots \right] \dots \right]; \\ I[K_{n,n}] &= 1 \left[\dots \xrightarrow{1} 1 \left[\begin{array}{c} \xrightarrow{1} \\ 2n-1 \end{array} \xrightarrow{\#2} \dots \xrightarrow{\#2n-4} 1 \left[\begin{array}{c} \xrightarrow{1} \\ 2n \end{array} \xrightarrow{\#1} \dots \xrightarrow{\#2n-3} \dots \right] \dots \right] \dots \right]. \end{aligned}$$

Замечание 2. Как видно из примеров для $I[P_{2k-1}]$ и $I[K_{n,m}]$, линейную нотацию $I[G]$ легко можно адаптировать для сжатия данных, если использовать специальную нотацию $S \times n$ для подстрок, которые повторяются n раз подряд. В случае с $I[P_{2k-1}]$ получим $I^*[P_{2k-1}] = 1 \left[\begin{array}{c} \xrightarrow{1} \\ 1 \end{array} \xrightarrow{2} 2 \left[\begin{array}{c} \xrightarrow{2} \\ 2 \end{array} \xrightarrow{3} \dots \xrightarrow{n-1} n \left[\begin{array}{c} \xrightarrow{n-1} \\ n \end{array} \right] \dots \right] \dots \right]; \times 2$.

Перейдём к рассмотрению элементарных операций на графах. При этом произвольную операцию на классах изоморфных графов будем описывать в форме алгоритма строковой обработки для линейных нотаций $I[G]$.

Пример 6. Введём операцию удлинения цепочек, определив результирующий граф $\tilde{G}$ как полученный из исходного графа G путём присоединения ко всем вершинам первой степени ровно по одной новой вершине (реализацию операции см. на рис. 4).

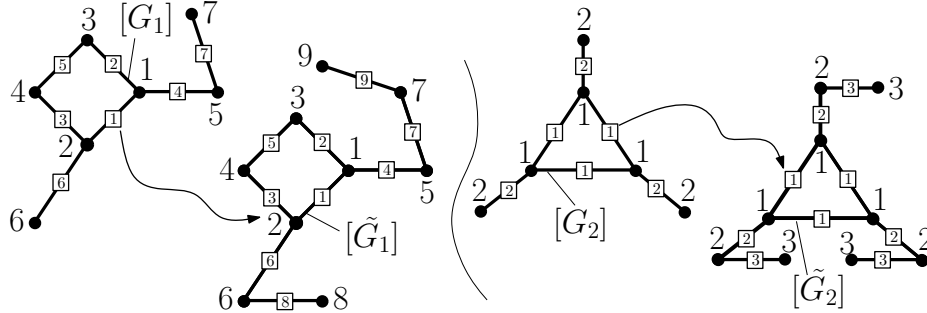


Рис. 4. Пример использования операции удлинения цепочек

Алгоритм построения линейной нотации $I[\tilde{G}]$ для результирующего графа на основе исходной нотации $I[G]$ можно свести к следующим трём шагам строковой обработки.

Шаг 1. Проводим поиск по строке $I[G]$ вершин, которые соответствуют шаблону $(\dots \xrightarrow{j} k[] \dots)$ для вершины первой степени. Среди всех этих вершин вычисляем минимальный индекс автоморфизма k и максимальный $k + m$, которые соответствуют данному шаблону.

Шаг 2. Вводим новые индексы классов $k + m + 1, k + m + 2, \dots, k + 2m$, которые будут соответствовать новым вершинам графа $\tilde{G}$.

Шаг 3. Получаем $I[\tilde{G}]$ из исходной линейной нотации $I[G]$ путём замены подстрок по шаблону: строку $(\dots \xrightarrow{j+i} k+i[] \dots)$ меняем на $(\dots \xrightarrow{j+i} k+i \left[\xrightarrow{j+i+m} k+m+i[] \right] \dots)$.

Пример 7. Рассмотрим операцию присоединения вершин первой степени к вершинам первого класса автоморфизма (реализацию операции для конкретных графов см. на рис. 5). Формально при отработке данной операции мы к каждой вершине v с индексом автоморфизма $I(\bar{v}) = 1$ присоединяем ровно одну вершину v^* степени 1.

Алгоритм построения линейной нотации $I[G^*]$ для результирующего графа на основе исходной нотации $I[G]$ можно свести к следующим трём шагам.

Шаг 1. Проводим поиск по строке $I[G]$ вершин, которые соответствуют шаблону $(\dots \xrightarrow{j} k[] \dots)$ и при этом связаны ребром с вершиной первого класса. Если таких вершин нет, то переходим к шагу 2. В противном случае добавляем в окружение каждой вершины первого класса одну дополнительную вершину класса k , осуществив строковую замену по шаблону: строки $(1[\dots \xrightarrow{j} k[]] \dots)$ меняем на $(1[\dots \xrightarrow{j} k[]; \xrightarrow{j} k[]] \dots)$; выход.

Шаг 2. Ищем вершины первой степени $(\dots \xrightarrow{j} k[] \dots)$. Среди всех этих вершин вычисляем минимальный индекс автоморфизма k и максимальный $k + m$.

Шаг 3. Производим строковую замену всех индексов вершин от k до $k + m$ по следующему шаблону: строки $(\dots \xrightarrow{j+i} k+i[] \dots)$ меняем на $(\dots \xrightarrow{j+i+1} k+i+1[] \dots)$. После этого добавляем в окружение каждой вершины класса 1 по одной новой вершине класса k в соответствии с шаблоном: $(1[\dots \xrightarrow{j} k[]] \dots)$.

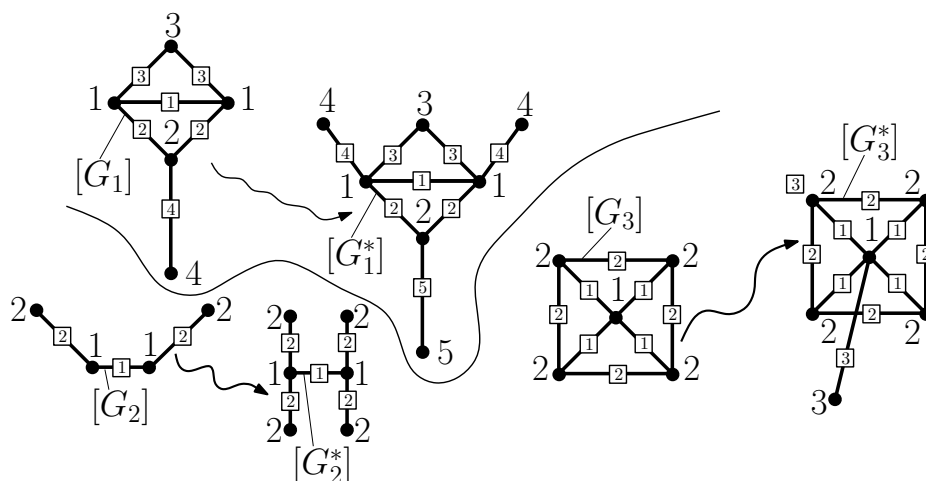
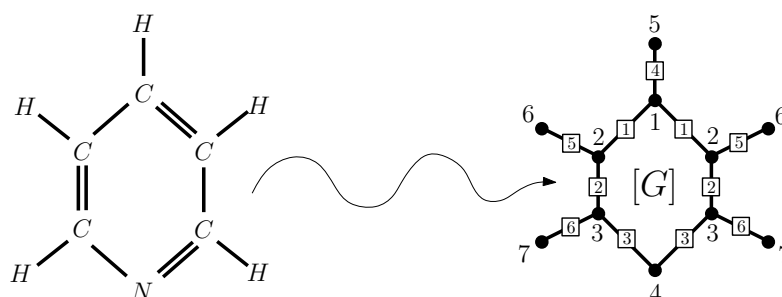


Рис. 5. Пример использования операции присоединения новой вершины к вершинам первого класса

Замечание 3. Для всех рассмотренных операций, а также для описания стандартных классов графов опустим доказательство корректности. При желании эти доказательства можно легко получить по аналогии с уже рассмотренными.

В определении 11 введено понятие раскраски для отдельных абстрактных вершин линейной нотации $I[G]$. По аналогии с этим определением можно ввести раскраску и для абстрактных рёбер на $I[G]$. Если у исходного графа G было n вершин, то, раскрасив итерационно все абстрактные вершины $I[G]$ в n разных цветов, получим $\mathfrak{L}(G^*)$ и восстановим исходный граф с точностью до изоморфизма $G^* \cong G$. Важно отметить, что подобная операция перехода от $I[G]$ к обычному графу $G^* \cong G$ в предельном случае потребует восстановить связи между каждым двумя вершинами графа и, как следствие, имеет максимальную сложность $O(n^2)$. Помимо восстановления обычного графа на основе $I[G]$, алгоритм раскраски можно использовать для построения молекулярных графов [5], если в качестве красок для вершин использовать названия химических элементов H, O, C, N , а в качестве красок для рёбер — типы связей.

Пример 8. На рис. 6 представлен молекулярный граф пиридина, а также маркированная линейная нотация, которая получена для этого графа с помощью алгоритма итерационной раскраски для вершин и рёбер (двойные связи показаны двойной стрелкой $\Rightarrow$).



$$I^*[G] = \langle 1, C \rangle \left[\xrightarrow{1} \langle 2, C \rangle \left[\xrightarrow{2} \langle 3, C \rangle \left[\xrightarrow{3} \langle 4, N \rangle \left[\xrightarrow{4} \langle 3, C \rangle \left[\xrightarrow{5} \langle 2, C \rangle \left[\xrightarrow{6} \#1; \xrightarrow{5} \langle 6, H \rangle; \xrightarrow{6} \langle 7, H \rangle \right] \right] \right] \right] \right] \right] \xrightarrow{6} \langle 7, H \rangle; \xrightarrow{5} \langle 6, H \rangle; \xrightarrow{4} \langle 5, H \rangle \right]$$

Рис. 6. Пример задания молекулярного графа в виде маркированной линейной нотации

При описании молекулярных графов на основе маркированных линейных нотаций $I^*[G]$ получаем три очевидных преимущества:

- проверка на изоморфизм двух графов имеет сложность $O(n^2)$;
- все возможные симметрии молекулы представлены в удобной форме через индексы классов автоморфизма вершин и рёбер;
- задание реакций между молекулами в виде операций на нотациях $I[G]$ сводится к процедурам строковой обработки, что упрощает их реализацию.

Алгоритм раскраски для $I[G]$, помимо уже рассмотренных приложений, может быть использован для построения путей на графе. В этом случае раскраска проводится натуральными числами $1, 2, \dots, n$, а две соседние вершины, получившие порядковые номера $i, i + 1$, должны быть соединены ребром в $I[G]$. Отметим, что отдельный интерес представляет поиск на линейной нотации $I[G]$ гамильтонова пути [6]. На основании перебора графов с числом вершин $|V| \leq 7$ была выдвинута следующая гипотеза относительно существования гамильтонова пути.

Гипотеза 1. Для того чтобы на $I[G]$ существовал гамильтонов путь, необходимо и достаточно, чтобы $I[G]$ удовлетворял одному из двух шаблонов: либо $I[G] = 1 \left[\begin{smallmatrix} j \\ 1 \end{smallmatrix} \right] \rightarrow k \left[\dots \right]$, либо $I[G] = 1 \left[\begin{smallmatrix} j \\ 1 \end{smallmatrix} \right] \rightarrow k \left[\begin{smallmatrix} \dots \\ 2 \end{smallmatrix} \right]; \xrightarrow{l} m \left[\begin{smallmatrix} \dots \\ 2 \end{smallmatrix} \right]$, где внутри скобки $\left[\begin{smallmatrix} \dots \\ 2 \end{smallmatrix} \right]$ число вложенных скобок в точности равно числу абстрактных вершин.

Заключение

Таким образом, можно утверждать, что линейная нотация $I[G]$ способна стать удобной альтернативой для классических графов G в приложении к целому ряду практических задач. Представляется актуальным продолжение исследований в данном направлении и построение новых алгоритмов на $I[G]$, в том числе и алгоритмов для реализации операций на графах. Более того, если удастся построить базис (набор операций, с помощью которых можно получить любой конечный граф, применяя их суперпозицию к некоторым простейшим графам) операций на $I[G]$ полиномиальной сложности, то нотацию $I[G]$ можно будет использовать на практике для хранения больших графов.

Сложность хранения в памяти $I[G]$ асимптотически равна $O(n^2)$, где n — число вершин графа. Можно повысить коэффициент сжатия данных для $I[G]$, если воспользоваться приёмом замечания 1, а также если отбросить индексы m для всех скобок $\left[\begin{smallmatrix} \dots \\ m \end{smallmatrix} \right]$ в записи $I[G]$ (данные индексы введены в определении $\mathfrak{L}(G)$ только для более наглядного представления нотации).

Помимо уже рассмотренных приложений для линейных нотаций $I[G]$, можно попробовать построить оптимизированный алгоритм для полного перебора графов из n вершин с отбрасыванием всех изоморфных копий. Фактически, если перебирать $I[G]$, а не матрицы смежности A графов G , то можно добиться многократного прироста производительности такого алгоритма, так как число классов изоморфных графов из n вершин существенно меньше числа таблиц из нулей и единиц размера $n \times n$ [7].

Как отмечено в работе [1], линейную нотацию $I[G]$ можно достаточно легко обобщить на случай упорядоченных графов, а также на гиперграфы [8]. С другой стороны, для классов автоморфизма вершин $\bar{v}$ и рёбер $(\bar{u}, \bar{v})$ можно ввести альтернативные индексы, если выбрать вместо макси-кода другой полный численный инвариант матрицы смежности. В частности, можно задействовать мини-код вместо макси-кода. С практической точки зрения такие альтернативные конструкции будут целесообразны, если

удастся показать, что с их помощью упрощается определение некоторых алгоритмов на классах изоморфных графов.

ЛИТЕРАТУРА

1. Назаров М. Н. Альтернативные подходы к описанию классов изоморфных графов // Прикладная дискретная математика. 2014. № 3. С. 86–97.
2. Зыков А. А. Основы теории графов. М.: Вузовская книга, 2004. 664 с.
3. Weininger D., Weininger A., and Weininger J. SMILES. 2. Algorithm for generation of unique SMILES notation // J. Chem. Inf. Comput. Sci. 1989. V. 29. No. 2. P. 97–101.
4. Белов В. В., Воробьев Е. М., Шаталов В. Е. Теория графов. М.: Высшая школа, 1976. 391 с.
5. Brecher J. Graphical representation of stereochemical configuration // Pure Appl. Chem. 2006. V. 78. No. 2. P. 1897–1970.
6. Харари Ф. Теория графов. М.: КомКнига, 2006. 296 с.
7. Harary F. and Palmer E. M. Graphical Enumeration. N.Y.: Academic Press, 1973. 240 p.
8. Zykov A. A. Hypergraphs // Russian Mathematical Surveys. 1974. V. 29. P. 89–154.

REFERENCES

1. Nazarov M. N. Al'ternativnye podkhody k opisaniyu klassov izomorfnykh grafov [Alternative approaches to the description of classes of isomorphic graphs]. Prikladnaya diskretnaya matematika, 2014, no. 3, pp. 86–97. (in Russian)
2. Zykov A. A. Osnovy teorii grafov [Basics of Graph Theory]. Moscow, Vuzovskaya Kniga Publ., 2004. 664 p. (in Russian)
3. Weininger D., Weininger A., and Weininger J. SMILES. 2. Algorithm for generation of unique SMILES notation. J. Chem. Inf. Comput. Sci., 1989, vol. 29, no. 2, pp. 97–101.
4. Belov V. V., Vorob'ev E. M., Shatalov V. E. Teoriya grafov [Graph Theory]. Moscow, Vysshaya Shkola Publ., 1976. 391 p. (in Russian)
5. Brecher J. Graphical representation of stereochemical configuration. Pure Appl. Chem., 2006, vol. 78, no. 2, pp. 1897–1970.
6. Kharari F. Teoriya grafov [Graph Theory]. Moscow, KomKniga Publ., 2006. 296 p. (in Russian)
7. Harary F. and Palmer E. M. Graphical Enumeration. N.Y., Academic Press, 1973. 240 p.
8. Zykov A. A. Hypergraphs. Russian Mathematical Surveys, 1974, vol. 29, pp. 89–154.